МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

О.Г. Аврунін, Є.В. Бодянський, М.В. Калашник, В.В. Семенець, В.О. Філатов

Сучасні інтелектуальні технології функціональної медичної діагностики

Монографія

Харків–2018

УДК 615.47: 616–072.7

Рекомендовано до друку рішенням Вченої ради Харківського національного університету радіоелектроніки (протокол №2/10-3 від 30.01.2018 р.)

Сучасні інтелектуальні технології функціональної медичної діагностики/ О.Г. Аврунін, Є.В. Бодянський, М.В. Калашник, В.В. Семенець, В.О. Філатов. – Харків: ХНУРЕ, 2018. – 236 с.

ISBN 978-966-659-234-0

Розглядаються інтелектуальні технології функціональної медичної діагностики. Наводяться основи інструментальних методів функціональних досліджень верхніх дихальних шляхів людини. Особлива увага приділяється сучасним методам обробки та інтелектуального аналізу результатів дослідження функції носового дихання. Обґрунтовуються принципи побудови апаратури та інформаційних систем функціональної діагностики порушень носового дихання. Розглядаються проблеми підвищення достовірності даних функціональної діагностики з урахуванням індивідуальної фізіологічної та анатомічної варіабельності.

Рекомендується для науковців, інженерів та медичних працівників – фахівців у галузі розробки та використання апаратури для функціональної діагностики в медицині.

The intellectual technologies of functional medical diagnostics are considered. The foundations of instrumental methods for functional studies of the upper respiratory tract of man are given. Main attention is paid to modern methods of processing and intellectual analysis of the results of the study of nasal breathing. The principles of apparatus and information systems for functional diagnostics of nasal breathing disorders are proposed. The problems of increasing the reliability of data of functional diagnostics with consideration of individual physiological and anatomical variability are described.

Recommended for scientists, engineers and doctors – specialists in the field of development and use of equipment for functional diagnostics in medicine.

УДК 615.47: 616–072.7

ISBN 978-966-659-234-0
DOI 10.30837/978-966-659-234-0

ЗМІСТ

ВСТУП

Сьогодні одним з найважливіших наукових напрямів, що інтенсивно розвиваються у всьому світі, є наука про дані (Data Science), невід'ємною частиною якої є інтелектуальний аналіз даних (Data Mining), що ставить своєю метою видобування з масивів апріорі необробленої інформації корисних з практичної точки зору знань, що можуть надалі бути використані для вирішення конкретних завдань. На базі Data Mining сьогодні сформувався цілий ряд напрямів, таких як: Text Mining, Web Mining, динамічний інтелектуальний аналіз даних (Dynamic Data Mining), аналіз потоків даних (Data Stream Mining) тощо. Особливе місце у цьому переліку займає інтелектуальний аналіз медичних даних (Medical Data Mining – MDM), що ставить своєю основною метою діагностування станів пацієнтів та вироблення конкретних рекомендацій щодо лікування з використовування апарата Data Mining, насамперед, класифікації, кластеризації, прогнозування, асоціації, ідентифікації та визначення змін (Chaujes and faults detection).

Слід зазначити, що у більшості реальних ситуацій, які виникають у MDM, неможливе використання вже напрацьованих результатів класичного інтелектуального аналізу даних. Це пояснюється високим рівнем апріорної та поточної невизначеності, що є притаманною саме завданням медичної діагностики. Вихідна інформація, яка має опрацьовуватися, може задаватися у формі текстових файлів, зображень, багатовимірних часових рядів, таблиць «об'єкт – властивість», що містять дані у різних шкалах, містять аномальні викиди та пропуски, збурення невідомої природи. Самі ж дані можуть міститися або у надвеликих базах даних (VLDB), або надходити у реальному часі в процесі обстежень, що висуває додаткові вимоги до швидкодії опрацювання інформації.

Необхідно також зазначити високу розмірність даних (що виключає можливість використання стандартних критеріїв розпізнавання образів – класифікації та кластеризації), виключення із ситуації, коли змінюється кількість факторів і діагнозів, нестаціонарність характеристик об'єкта дослідження, формованих класів – діагнозів, їх взаємне перетинання, що виключає можливість чітких (Crisp) методів опрацювання інформації.

У таких ситуаціях найбільш доцільним є використання апарата штучного інтелекту і, насамперед, гібридних систем обчислювального інтелекту (HSCI), таких як штучні нейронні мережі, системи нечіткого висловлювання.

Водночас складність задач MDM стає перепоною для використання відомих вже систем. Тому нами було розроблено, досліджено та використано у задачах медичної діагностики нові нейрон-фаззі, нео-фаззі, вейвлет-нейро-фаззі (включаючи fuzzy type-2) системи, що здатні навчатися (у традиційному сенсі та з позиції глибинного навчання – deep learning) і опрацьовувати інформацію у режимі реального часу, розв'язуючи задачі класифікації, кластеризації, прогнозування та діагностування за умов невизначеності, нестаціонарності, нелінійності, хаотичності та нечіткості (що особливо притаманні біологічним

системам), кількості та форми класів – кластерів, що утворюються цими даними. Особливістю введених систем є можливість роботи за високого рівня апріорної та поточної невизначеності: кількості та форми кластерів, значного їх перетинання, можливості виключення нових класів – діагнозів у процесі надходження інформації.

Як відомо, найбільш складними задачами у межах Data Mining є ті, що базуються на парадигмі самонавчання і, насамперед, задачі кластеризації. Найбільш складні ситуації виникають у випадку високого рівня перекриття класів (fuzzy clustering) дуже великих обсягів даних (Big Data) і особливо за високих розмірностей, векторів, ознак, коли виникає ефект «концентрації норм» (Concentration of Norms-CoN). Тому, нами було розроблено кластерувальні online HSCI, що здатні налаштовуватися у режимі самонавчання (self-learning), активного навчання (active learning) та лінивого навчання (lezi learning), формуючи класи довільної форми із суттєвим рівнем перетинання, не потерпаючи при цьому від «прокльону розмірності» та «концентрації норм».

Вирішення низки практичних реальних задач, включаючи задачі медичної діагностики, довело ефективність підходу, що розвиваються нами.

Результати щодо введених систем, надруковано у провідних світових журналах з проблематики штучного та обчислювального інтелекту: Neurocomputing, Soft Computing, Evolving Systems, Applied Soft Computing, clut.j.intelligent Systems and Applications (Web of Science, Scopus) та у розділах монографій, що надруковано у США, ФРН, Швейцарії.

У монографії наведено результати інтелектуального аналізу даних на прикладі результатів тестування носового дихання. Вибір оптимальної, за критерієм максимуму достовірності, системи інформаційних ознак є класичною задачею інтелектуального аналізу даних в умовах апріорної невизначеності в ході формування діагностичних рішень.

Сучасний розвиток технічних і методологічних засобів призвів до появи високоточних приладів вимірювання фізичних величин. Проте, сьогодні актуальною є проблема повторюваності даних під час вимірювання фізіологічних параметрів організму людини в умовах відсутності еталону. При цьому методи функціональної діагностики на сучасному етапі вимагають введення чітких і наочних критеріїв, необхідних для прийняття обґрунтованих діагностичних рішень, прогнозування та визначення ефективності функціональних оперативних втручань на доказовому рівні.

Збільшення кількості діагностичних параметрів без достатнього фізіологічного та статистичного обґрунтування, а також чіткої інтерпретації результатів обстеження тільки ускладнює прийняття діагностичних рішень. Тому актуальними є задачі застосування нових методів обчисленого інтелекту і відповідної вимірювальної апаратури для уточнення параметрів фізіологічних процесів, таких як зовнішнє дихання і проходження повітря через верхні дихальні шляхи, а також вивчення впливу певних анатомічних структур, наприклад, придаткових пазух і носового клапана на аеродинамічні процеси в носовій порожнині.

1 ПРОБЛЕМИ ТА ОСОБЛИВОСТІ МОДЕЛЮВАННЯ ПРОЦЕСІВ ОБРОБКИ ДАНИХ У РОЗПОДІЛЕНИХ СИСТЕМАХ ОБЧИСЛЮВАЛЬНОГО ІНТЕЛЕКТУ

1.1 Єдиний інформаційний простір: проблеми та методи керування інформацією

Сучасний рівень розвитку суспільства підняв індустрію інформаційних технологій до стратегічного напряму, в якому зосереджені великі інтелектуальні і фінансові ресурси. Інформація та інструменти керування інформацією – програмні продукти різного функціонального призначення − набули статусу інформаційних ресурсів [1]. Керування інформацією подібно до інших областей інформатики зазнає радикальних змін у низці аспектів: у моделях збереження та доступу в масштабах систем, що проектуються, а також у базових інформаційних технологіях. Дослідженню питань, пов’язаних з сучасним поданням інформації, інформаційних ресурсів та інформаційних технологій, присвячено ряд робіт [2–5].

Інформаційний простір структурується крізь інформаційні системи – взаємопов’язані сукупності методів та засобів збору, накопичення та зберігання інформації [6]. Інформаційні ресурси концентруються в межах інформаційних систем (ІС).

Об’єднання ресурсів на основі інформаційно-комунікаційної взаємодії інформаційних систем виводить їх на рівень корпоративних інформаційних ресурсів, які отримали назву Єдиний Інформаційний Простір (ЄІП).

Реалізація ЄІП масштабу регіону, корпорації, підприємства можлива за умови створення та подальшого дотримання стандарту щодо взаємодії між собою як інформаційних систем, так і окремих реалізацій (рис. 1.1).

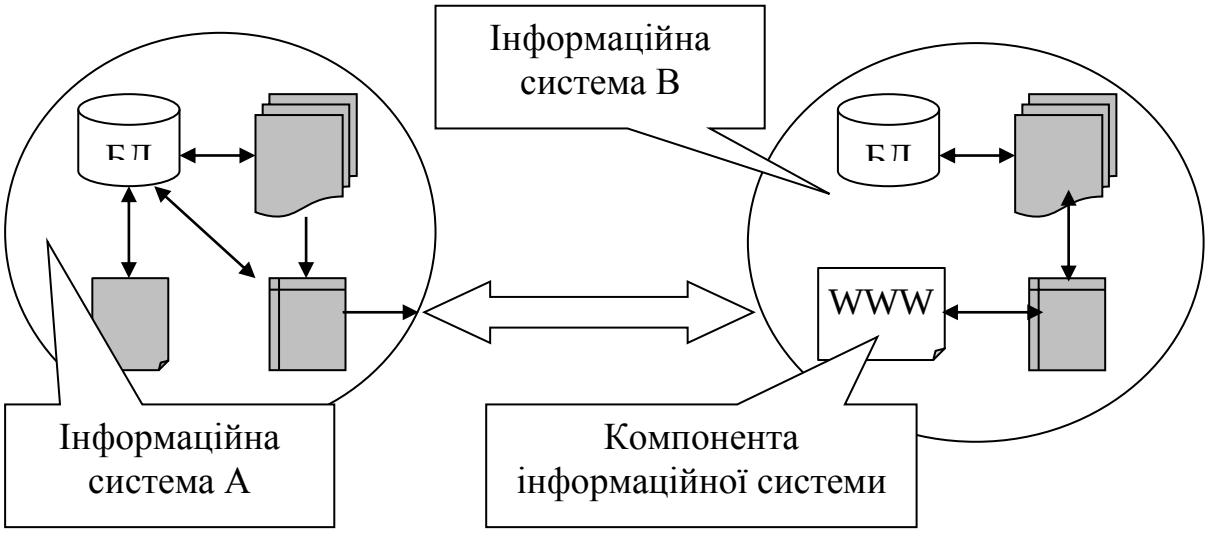


Рис. 1.1. Структура взаємодії інформаційних систем

ЄІП містить поняття Єдиного Простору Даних (ЄПД), яке реалізує технології доступу до віддалених баз даних, при цьому інформаційні системи виступають у ролі клієнта і сервера, взаємодіючи один з одним за сценарієм, поданим на рис. 1.2.

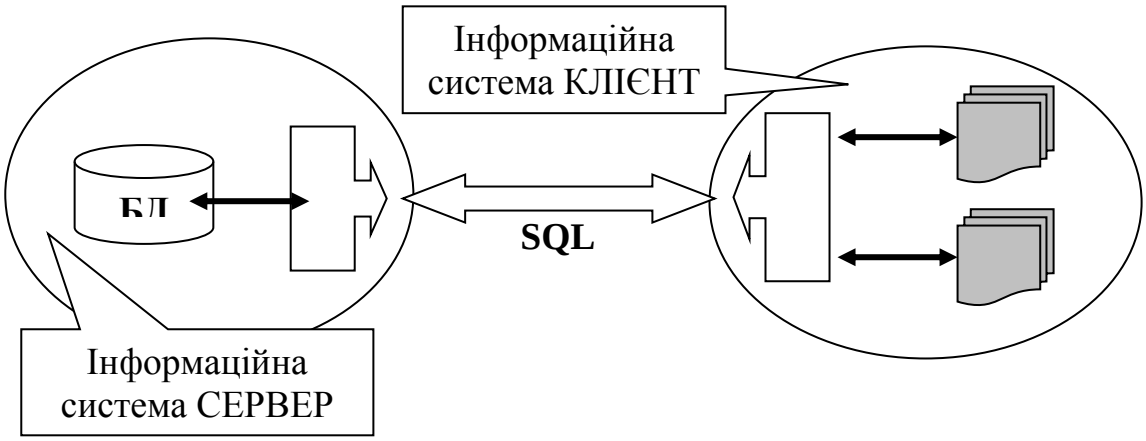


Рис. 1.2. Структура взаємодії інформаційних систем

Інформаційна система-клієнт (ІСК) посилає інформаційній системі-серверу (ІСС) запит, отримуючи в результаті дані, які підлягають подальшій обробці. Як мова запитів найчастіше використовується мова SQL – стандарт спілкування з реляційними системами керування базами даних (СКБД). Доступ до віддалених баз даних (БД) у більшості випадків здійснюється за допомогою продуктів, що підтримують протоколи ODBC (Open Database Connectivity) та JDBC (Java Database Connectivity), або використовуються шлюзи, що постачаються виробниками СКБД або третіми фірмами-розробниками.

Фактично, в процесі побудови єдиного простору даних використовується архітектура доступу до віддалених баз даних, що є аналогом дворівневої архітектури клієнт-сервер [7]. Ця архітектура передбачає реалізацію на стороні клієнта як функцій вводу та відображення даних, так і прикладних функцій програмного забезпечення, тобто методів обробки даних. Клієнт спрямовує запити на сервер, який їх оброблює та повертає клієнту результат, оформлений як блок даних.

Описаному вище сценарію взаємодії систем притаманні всі недоліки, характерні для дворівневої архітектури клієнт-сервер:

– необхідно знати на стороні ІСК особливості використовуваної СКБД і структуру віддаленої БД ІСС, що знижує рівень ефективності та безпеки усієї системи в цілому;

– ускладнений супровід і модифікація тих програм інформаційних систем-клієнтів, які взаємодіють з базами даних інформаційних систем-серверів, оскільки будь-яка зміна схеми віддаленої БД на стороні ІСС спричиняє зміну у ІСК, що ускладнює обслуговування, оновлення та заміну програм, встановлених на десятках-сотнях комп’ютерів;

– значно ускладнюється адміністрування БД ІСС, яке включає в себе керування правами доступу користувачів ІСК.

Значним недоліком розглянутого вище підходу є дублювання прикладних програм ІСС у ІСК, що призводить до неефективного використання ресурсів взаємодіючих інформаційних систем.

Зростання популярності глобальної мережі Internet та технології World Wide Web за останній час викликає підвищений попит до них з боку розробників корпоративних інформаційних систем. Спочатку WWW створювався тільки як засіб, що надає графічний інтерфейс до Internet та спрощує доступ до інформації, розподіленої по множині комп'ютерів усього світу [8, 9]. При цьому основними компонентами були сторінки, вузли, браузери та сервери Web. Не вдаючись у подробиці, відзначимо, що користувачам була надана можливість навігації по Internet з використанням технології гіпертексту, який підтримується протоколом HTTP (Hypertext Transfer Protocol) та стандартом мови HTML (Hypertext Markup Language).

Поява CGI (Common Gateway Interface) вирішила проблему обміну інформацією між сервером Web і такими програмами, як бази даних, які не можуть безпосередньо обмінюватися даними з браузерами Web. Внаслідок з'явилася можливість реалізації інтерактивної взаємодії користувача з програмами Web-серверу, які обробляли інформацію, введену користувачем в браузері, і як результат повертали сформовану HTML-сторінку. Більшість існуючих рішень доступу до БД у середовищі Internet засновані на цьому підході.

Слід зазначити, що поява мови Java надала розробникам інформаційних систем абсолютно нові технологічні рішення побудови програмних засобів у середовищі Internet/Intranet. Але було б невірно розглядати технологію Java тільки як частину технології WWW, оскільки Java дає змогу вирішувати задачі більш широкого класу, ніж технологія, що базується на мові HTML, протоколі HTTP та CGI.

Можливості, що надаються WWW-технологією, безумовно, розширили спектр рішень, якими керуються проектувальники в процесі побудови ІС. Але виникає питання: що являють собою системи взаємодіючих ІС, засновані на технології WWW? Чи мають вони змогу вирішити проблему єдиного інформаційного простору? Із впевненістю можна сказати, що ні.

Настільки сильне твердження пов'язане з тим, що з розглядом взаємодії інформаційних систем ІСК з браузером виступають у ролі компонента представлення, а ІСС з WWW-сервером та прикладними програмами виступають у ролі компонента, який реалізує функціональну логіку та доступ до даних, що, по суті, відповідає дворівневій архітектурі з інтелектуальним сервером (рис. 1.3). Не зважаючи на те, що, так само, як і у підході, який базується на доступі до віддалених даних, WWW-технологія має змогу покращити ситуацію з імпортом/експортом даних між ІСК та ІСС, але все-таки має місце ряд недоліків, властивих дворівневій архітектурі з інтелектуальним сервером.

Так, одним з недоліків, без сумніву, є реальна відсутність можливості реалізації процесу обробки даних, що поставляються WWW-сервером, на боці ІСК. Дійсно, ІСК отримує інформацію від ІСС у вигляді HTML-сторінок, що практично робить неможливою організацію процесу обробки отриманих

даних компонентами ІСК. Як наслідок, це призводить до зниження ефективності використання обчислювальних ресурсів інформаційних систем. З іншого боку, гостро постає проблема підтримання безпеки системи, яка на даний момент не має цілісного рішення у середовищі Internet, що не є допустимим для організацій, які висувають підвищені вимоги до безпеки. І, нарешті, як і в попередньому підході, суттєво ускладнюється адміністрування ресурсів ІСС, що включає керування правами доступу користувачів ІСС.

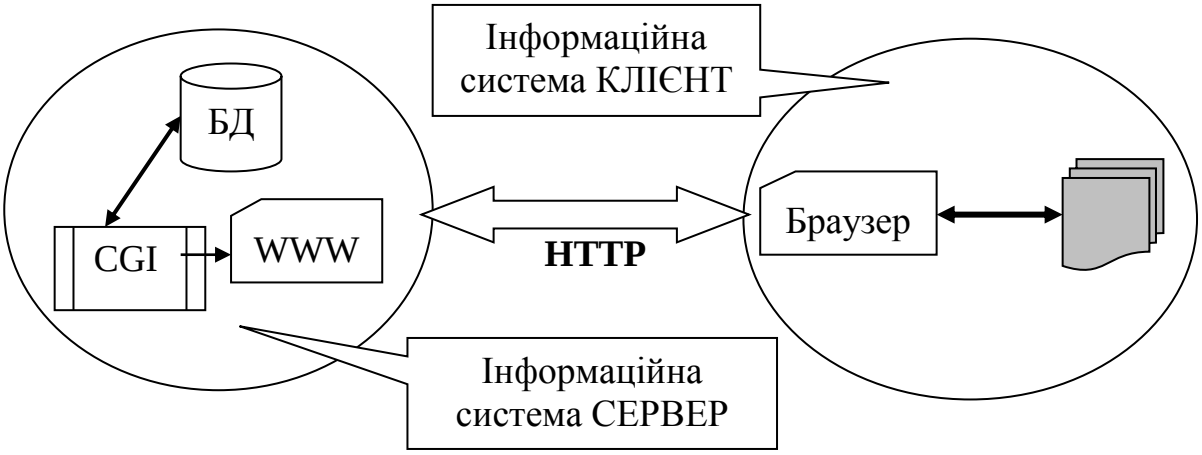


Рис. 1.3. Структура взаємодії інформаційних систем

На відміну від розглянутих підходів, у концепції єдиного інформаційного простору передбачається, що у ролі інформаційних ресурсів (відносно ІС, яка розглядається) виступають не тільки дані, але й різноманітні застосунки інформаційних систем (application software). Тоді в кожній з інформаційних систем частина методів обробки даних реалізується у вигляді програмних застосунків, які є доступними з інших інформаційних систем. Наприклад, в ході взаємодії двох ІС перша використовується сервісами, які надає друга, і як результат отримує вже оброблені дані, що можуть підлягати подальшій обробці компонентами першої ІС.

Даний підхід відповідає розподіленій одноранговій архітектурі взаємодії [9]. Згідно з цією архітектурою, будь-які застосовні програми з різноманітних ІС можуть виступати як у ролі клієнта, так і у ролі сервера стосовно один одного, спільно вирішуючи ті чи інші задачі. Такий підхід мінімізує дублювання програмних модулів. Розподілення інформаційних ресурсів різними інформаційними системами дозволяє досягти оптимального балансу завантаження програм та апаратних засобів, і, отже, приводить до ефективного використання інформаційних ресурсів систем у цілому.

Знання схеми бази даних необхідне тільки тому застосунку, яке оброблює дані з цієї бази даних. Використання ІСС-сервісів, які надаються інформаційною системою-сервером і реалізують методи обробки даних, дозволяє вирішувати проблему зміни схеми віддаленої бази даних. При цьому статичність інтерфейсів компонентів, які надають ІСС набір сервісів, досягається шляхом застосування методологій об’єктно-орієнтованого аналізу та проектування розподілених об’єктних технологій (CORBA, Java,

DCOM) на різних етапах створення інформаційних систем. І, нарешті, оскільки у рамках конкретних інформаційних систем локалізовані не тільки дані, але й методи їх обробки, відбувається істотне зменшення витрат на адміністрування, супровід та модифікацію інформаційних систем, що складають єдиний інформаційний простір.

Більшість як вже існуючих, так і тих інформаційних систем, що розробляються, є програмними реалізаціями у дворівневій архітектурі «клієнт-сервер». При цьому як засоби спілкування клієнта та сервера доволі часто використовуються не повністю стандартизовані механізми тригерів та процедур, які зберігаються. Специфіка їх реалізації (невід'ємність від ядра СКБД) призводить до необхідності наявності додаткових обчислювальних ресурсів на стороні сервера.

Зі збільшенням робіт, що виконує сервер, системи у дворівневій архітектурі «клієнт-сервер» стають все більш схожими на великі ЕОМ (мейнфрейми), а структури даних, які вони оброблюють, та способи їх подання мало доступні для використання спільно з іншими застосунками. Зазвичай взаємодію розглянутих «клієнт-серверних» застосунків організують засобами СКБД, що помітно перевантажує серверну частину. З іншого боку, сучасні технології та програмні засоби дозволяють зв'язувати компоненти інформаційних систем в інтегроване середовище у рамках концепції Єдиного Інформаційного Простору.

1.2 Аналіз методів і технологій керування розподіленою інформацією в обчислювальних системах

Специфіка задач, що вирішуються за допомогою ІС, складність їх створення, модифікації, супроводу, інтеграції з іншими ІС тощо, дозволяють розділити інформаційні системи на такі класи [10]:

- малі інформаційні системи;
- середні інформаційні системи;
- великі інформаційні системи (корпоративні інформаційні системи – системи рівня регіональних організацій).

До класу малих інформаційних систем входять системи рівня невеликого підприємства. До основних ознак таких систем слід віднести:

- нетривалий життєвий цикл;
- орієнтація на масове використання;
- невисока ціна;
- практична відсутність засобів аналітичної обробки даних;
- відсутність можливості незначної модифікації без участі розробників;
- використання середнього класу СКБД, таких як Clarion, FoxPro, Clipper, Paradox, Access;
- однорідність апаратного та системного програмного забезпечення;
- практична відсутність засобів забезпечення безпеки.

На відміну від попереднього класу, ознаками середніх інформаційних систем є:

– довгий життєвий цикл (можливість росту до великих систем);
– наявність аналітичної обробки даних;
– наявність штату співробітників, які здійснюють функції адміністрування апаратних та програмних засобів;
– наявність засобів забезпечення безпеки;
– тісна взаємодія з фірмами – розробниками програмного забезпечення з питань супроводу компонентів ІС.

І, нарешті, до характерних ознак корпоративних інформаційних систем слід віднести:

– довгий життєвий цикл;
– різноманітність програмного забезпечення, що використовується, життєвий цикл якого менший, ніж у системи, що створюється;
– різноманітність програмного забезпечення, що використовується;
– масштабність і складність задач, які вирішуються;
– перехрещення великої кількості різноманітних предметних областей;
– орієнтація на аналітичну обробку даних;
– територіальну розподіленість.

На цей час з'явилась значна кількість робіт, пов'язаних з описом продуктів, технологій і методологій, розрахованих на створення малих та середніх інформаційних систем [11–13]. Технології та методології побудови великих інформаційних систем, що об'єднують у собі велику кількість локальних інформаційних систем, практично не розглядаються та не обговорюються. Доволі часто це призводить до того, що як технології створення великої інформаційної системи розробники вибирають ті, які спочатку на це не розраховані.

У зв'язку з цим на перше місце виходять питання створення концепції інтеграції розподілених інформаційних систем рівня регіональних організацій з використанням розподілених об'єктних технологій. Технологія створення інформаційних систем є сукупністю практичних інженерних знань, що застосовуються у процесі розробки програмного забезпечення впродовж всього його життєвого циклу. Технологія також описує принципи організації та керування процесом розробки, представляючи його у вигляді ряду послідовних і паралельних етапів, а також продуктів, що створюються на цих етапах. З технологію нерозривно пов'язані інструментальні засоби, що використовуються на різноманітних стадіях розробки.

Вибір прийнятної технології створення інформаційної системи прямо залежить від вибору архітектури ІС. Розглядаючи інформаційну систему як сукупність взаємодіючих компонентів, можна розподілити їх за такими рівнями:

– апаратний рівень – комп'ютери, периферійні пристрої, мережне та телекомунікаційне обладнання та ін.;
– системний і системно-залежний рівень – операційні системи, мережні протоколи та ін.;

– рівень прикладного середовища – засоби middleware (CORBA, DCE, Tuxedo та ін.), DBMS, Intranet, OLAP, комунікаційні інтерфейси.

На практиці найбільше застосування отримала дворівнева та трирівнева архітектури розподіленого керування [14].

Дворівнева архітектура припускає, що кількість рівнів дорівнює двом. Один із двох компонентів виступає в ролі сервера, тобто реалізує набір сервісів, доступних іншому компоненту, що виступає в ролі клієнта, тобто в процесі роботи користується сервісами, що надаються сервером. Компоненти можуть розташовуватися як на одному комп'ютері, так і на різних комп'ютерах, об'єднаних у мережу. Розходження в реалізаціях дворівневої архітектури визначаються в основному тим, функції яких груп виконує клієнт, а яких – сервер. Існує кілька варіантів декомпозиції функцій подання, прикладних функцій і функцій зберігання й керування даними в рамках дворівневої архітектури ІС.

Відповідно до зазначених варіантів декомпозиції, можна говорити про такі дворівневі архітектури:

– інтелектуального клієнта (доступ до вилучених даних, архітектура на базі файлового серверу);

– інтелектуального серверу (доступ у режимі терміналу, архітектура інтелектуального серверу баз даних);

– розподіленої функціональної логіки.

Дворівневі архітектури мають низку переваг і недоліків, частково описаних у [9]. Використання дворівневих архітектур під час побудови регіональних інформаційних систем, виходячи із властивих їм недоліків, вважається неефективним [13].

Трирівневі архітектури передбачають не настільки жорсткі зв'язки між клієнтом і сервером та більш гнучкі форми розподіленої обробки [14]. Найпоширенішою вважається архітектура, відповідно до якої виділяються три компоненти ІС (подання, прикладний, доступу до інформаційних ресурсів), що є автономними й спілкуються через засоби міжпроцесної взаємодії за допомогою стандартних інтерфейсів. Окремі компоненти можуть розташовуватися як на одному комп'ютері, так і на різних комп'ютерах, забезпечуючи тим самим розподілену обробку інформації. Компонент подання часто розташовується на персональному комп'ютері, прикладний компонент (що називається сервером застосунків − application server) виконується сервером середнього рівня під керуванням операційної системи Unix або Windows NT, а компонент доступу до даних і самі дані розташовуються або на потужних Unix-Серверах, або на великих ЕОМ. Проте на практиці всі три компоненти можуть із успіхом виконуватися й на одному комп'ютері.

Основним елементом трирівневої архітектури є сервер застосунків. Як правило, у ньому реалізується кілька прикладних функцій, кожна з яких оформлена як сервіс і надає деякі послуги всім компонентам подання, які бажають і можуть ними скористатися. Серверів застосунків може бути декілька, і кожен з них може надавати певний набір сервісів. Деталі реалізації прикладних функцій у серверах застосунків повністю приховані від клієнтів.

Крім того, розробники можуть створювати, змінювати або переносити будь-які компоненти ІС, практично не чіпаючи інших.

За оцінками фахівців в області інформаційних технологій, перспективним напрямком розвитку такого класу систем є розподілені однорангові архітектури керування.

Відповідно до розподіленої однорангової архітектури, клієнт, що взаємодіє із сервером, трактується більш широко, ніж компонент подання. Він може підтримувати інтерфейс із кінцевим користувачем, а може також виконувати прикладні функції і бути сервером застосовних програм. У загальному випадку, у рамках даної архітектури клієнт (сервер) може як надавати, так і робити запити до деяких сервісів. Це дозволяє на етапі проектування інформаційної системи здійснити таку декомпозицію функцій із зазначених вище трьох груп за компонентами ІС, яка була б оптимальною в контексті розв'язуваної задачі.

Для забезпечення взаємодії компонентів інформаційної системи, що підтримує розподілену однорангову архітектуру, необхідно створити проміжний програмний рівень (middleware), за допомогою якого запити приймаються від клієнтів і направляються відповідному серверу. Сьогодні вже існує або анонсована достатня кількість інструментальних засобів, які дозволяють розробникам будувати розподілені ІС, не вдаючись у деталі реалізації взаємодії клієнта й сервера. Більшість з цих програмних продуктів реалізують стандарт CORBA (Common Object Request Broker Architecture), а деякі інструментальні пакети (наприклад, продукт Orbix фірми IONA Technologies) пропонують розширені варіанти цього стандарту.

Із проведеного огляду можна зробити висновок, що розподілені однорівневі архітектури мають більш універсальний характер, ніж дворівневі й трирівневі архітектури. Чітке розмежування логічних компонентів, властиве розподіленим однорівневим архітектурам, і раціональний вибір програмних засобів дозволяють досягти такого рівня гнучкості, відкритості й продуктивності ІС, який поки є недосяжним в процесі використання дворівневих і трирівневих архітектур.

1.3 Технології та програмні засоби інтеграції інформаційних ресурсів розподілених обчислювальних систем

Гетерогенні обчислювальні середовища сьогодні стали реальністю для багатьох організацій. У зв'язку з цим підвищуються вимоги до інтеграції різнорідних програмних застосунків, які автоматизують діяльність підприємств і функціонують у розподілених середовищах із широким діапазоном платформ і мереж. Великі організації, наприклад, банки, податкові служби, мають різні комп'ютерні системи (SUN, RS6000, IBM PC, Alpha та ін.), які встановлені в одному або різних місцях.

Для адміністраторів комп'ютерних систем основною проблемою є забезпечення взаємодії цих систем для їхньої спільної роботи. Перед розробниками прикладних програм, по-перше, постає проблема створення

такого програмного забезпечення, яке могло б працювати на максимально можливій кількості платформ, що використовуються в організації. По-друге, розробка програмних засобів у концепції Єдиного Інформаційного Простору має здійснюватися або на основі власних стандартів (замкнуте рішення), або на основі загальноприйнятих міжнародних стандартів. По-третє, розробникам необхідно передбачити можливість модифікації застосунків ІС так, щоб процес модернізації був мінімальний за часом і витратами. А цього можна досягти, дотримуючись певних принципів виділення та об'єднання компонентів інформаційних систем.

Для оцінки існуючих технологій інтеграції компонентів інформаційних систем визначаються такі категорії технологічних рішень, що характеризують погляд конкретної організації на архітектуру інформаційної системи в цілому [15, 16]:

- окремі рішення;
- різноманітні (змішані) механізми (Miscellaneous Mechanisms);
- віддалені виклики процедур на базі DCE RPC;
- розподілені об'єкти (CORBA, DCOM) (Distributed Object);
- технологія frameworks (відкритої інфраструктури);
- стандартні архітектури (Standard Architectures).

Коротко зупинимося на рішеннях, що відповідають кожній з категорій.

Як згадувалося вище, однією з основних вимог, якій мають задовольняти розподілені інформаційні системи, є використання програмного забезпечення й технологій, що узгоджені із загальновизнаними стандартами, які визначають принципи взаємодії компонент ІС.

Технологічні рішення, що належать до першої категорії, базуються на основі власних (унікальних для даної організації) протоколів та інтерфейсів взаємодії. Такі рішення в більшості випадків породжують непереборні труднощі під час організації спілкування компонентів даної ІС із компонентами інформаційних систем, побудованих на основі інших рішень міжкомпонентної взаємодії [17].

До другої категорії належать технологічні рішення, що припускають побудову інформаційних систем з розрахунку на конкретну задачу й спочатку не розраховані на використання технологій інтеграції систем. Тому, у цьому випадку як засоби інтеграції використовуються такі механізми, як сокети (sockets) протоколу TCP/IP і ONC RPC (Object Network Computing Remote Procedure Call). Програмне забезпечення даної категорії, як правило, розробляється для використання тільки усередині конкретної організації, що приводить до різкого збільшення витрат в ході інтеграції з іншими системами.

У технологічних рішеннях третьої категорії передбачається використання технологій, що базуються на засобах інтеграції компонентів і мають переваги порівняно з рішеннями попередньої категорії. До даної категорії належать технології на базі механізму виклику віддалених процедур (RPC) такі, як OSF DCE. Як відомо, OSF DCE визначає сервіси безпеки, іменування й інші важливі механізми, необхідні для інтеграції систем у розподіленому середовищі. З іншого боку, створення об'єктно-орієнтованої системи на базі

DCE не може бути оптимальним рішенням через неповну об’єктну орієнтованість останньої.

Характерним недоліком OSF DCE вважається складність створення об’єктів як незалежних компонентів розподіленої системи й орієнтованість на процедурний стиль програмування. Тому є тенденція розглядати DCE як застарілу технологію порівняно з новими об’єктно-орінтованими технологіями побудови розподілених систем, такими, як технологія CORBA і DCOM (в міру прийняття стандарту). Використання даної технології може привести до морального зношування системи [15].

У четвертій категорії в ході побудови інформаційних систем використовуються ORB-технології CORBA, зокрема високорівневий механізм RPC. Тут застосовують сервіси й мову опису інтерфейсів (OMG IDL), визначені в специфікації CORBA тільки для забезпечення міжплатформенної взаємодії. У випадку, коли міжплатформенна взаємодія не потрібна, використовуються власні механізми взаємодії компонентів системи. Важливо відзначити, що ці системи ризикують технологічно застаріти, використовуючи специфічні механізми, які не забезпечують реальних переваг програмної архітектури CORBA [16–18].

Технологічні рішення п’ятої категорії припускають розробку frameworks як основи програмної архітектури для використання в декількох проектах. Програмний фреймворк може містити допоміжні програми, бібліотеки коду, скрипти та загалом все, що полегшує створення та поєднання різних компонентів великого програмного забезпечення чи швидке створення готового і не обов’язково великого програмного продукту. Як правило, framework втілює ретельно продумані принципи побудови програмної архітектури. Розробка frameworks – перспективний напрямок, що бурхливо розвивається останнім часом, але це вже тема окремого дослідження [19].

І, нарешті, до шостої категорії належать високоякісні технології, програмні архітектури й сервіси, які розраховані на повторне використання в багатьох різних проектах. Важливо відзначити, що не кожна організація здатна створити технологію світового класу. Організації, що розробляють рішення шостої категорії, – всесвітньо відомі фірми, які є творцями стандартів взаємодії систем у своїй галузі.

1.4 Проблеми й підходи до обробки даних і керування процесами складних технологічних об’єктів за умов невизначеності

Сучасні складні технологічні об’єкти в різних предметних областях значною мірою функціонують за умов невизначеності, що виникає внаслідок нечіткості, неточності, стохастичності процесів керування. Це пов’язано переважно з тим, що зазвичай на процеси автоматизованого керування накладаються жорсткі обмеження на часові, матеріальні, інформаційні ресурси. Варто також враховувати наявність суб’єктивного фактора за рахунок участі людини в контурі керування. Якість керування у виробничих системах значною мірою залежить від ступеня врахування зазначених вище факторів, що висуває

особливі вимоги до складу та змісту інформаційного, математичного й технічного забезпечення автоматизованих інформаційно-керуючих систем складними об’єктами, засобів відображення процесів на моделях. Різні аспекти цих проблем досить глибоко досліджені в роботах [20–24], проблеми моделювання, розробки, дослідження та впровадження засобів автоматизації керування розглянуто також у роботах [25–31]. Ці роботи охоплюють широке коло питань, які вирішуються, є актуальними в досить значному часовому інтервалі й можуть бути поширені на сучасні складні виробничі об’єкти.

Використовуючи підходи до виявлення властивостей і особливостей функціонування складних систем, які досить глибоко досліджені в науковій літературі, зокрема, у роботах [20–24] і деяких інших, розширимо ці поняття також і на складні об’єкти. Наведемо, на наш погляд, деякі істотні фактори, властиві складним об’єктам при автоматизованому керуванні:

– наявність глобальних і локальних критеріїв якості їхнього функціонування й цілей, які вони реалізують;

– територіальна й функціональна розподіленість об’єкта;

– процеси, що відбуваються, (технічні, організаційні, технологічні тощо) мають різну природу і характеризуються складною причинно-наслідковою, часто асинхронною взаємодією, що вимагає різноманіття підходів;

– багатокритеріальність, що вимагає врахування великої кількості критеріїв, які часто є суперечливими;

– процеси, які характеризують об’єкт, з погляду керування носять неточний, нечіткий [32], у ряді випадків стохастичний характер [33], у зв’язку із цим об’єкт керування функціонує в умовах невизначеності;

– значна роль суб’єктивного людського фактора, що вимагає його додаткового дослідження й максимального врахування засобами адаптації автоматизованих систем.

У роботі [22] відзначено, що в практичних реалізаціях доцільно керуватися принципом необхідної та достатньої різноманітності. Приймаючи його за аксіому, робиться висновок про те, що автоматизована система керування, зокрема з прийняття рішень, повинна мати реалізовані функції, що перевищують за своєю складністю об’єкт автоматизації. Це, як випливає з викладеного вище, припускає, а в ряді випадків і передбачає деяку надмірність системи. Очевидно, що частина інформаційних ресурсів може використовуватися нераціонально, а це викликає необхідність додаткових досліджень.

Взявши за основу принципи побудови інтегрованої автоматизованої системи керування організаційного типу, основи яких викладені в роботах [23, 29, 30], розширимо їх на випадок функціонування об’єкта і системи автоматизованого керування за функціонального й територіального розподілу, а також нечіткості процесів дослідження. Основні з них з необхідними обмеженнями й застереженнями викладемо в такій інтерпретації:

– принцип ергодичності заснований на тому, що система керування створюється як людино-машинна система, у якій для автоматизації слабко формалізованих процесів використовуються інтерактивні процедури. Для функціонування системи в реальному часі розвитку процесів складного

об’єкта цей принцип мало застосовується через його функціональну обмеженість і слабку прив’язку до реального часу;

– принцип системності припускає, що система будується як відкрита і така, що розвивається [33], здійснюється взаємозв’язок з іншими системами й зовнішнім середовищем, а процеси розвитку належать до всіх елементів системи, вони є універсальними й охоплюють весь комплекс розв’язуваних задач;

– принцип ієрархічності. Останнім часом, коли значне поширення отримали розподілені системи, цей принцип носить дещо обмежений характер і може застосовуватися разом із принципом розподіленості [34];

– принцип розподіленості передбачає, що об’єкт має яскраво виражені властивості територіальної й функціональної розподіленості, система автоматизованого керування, відповідно до принципу необхідної різноманітності [23], має бути реалізована з виконанням основних положень принципу розподіленості;

– принцип сумісності базується на єдності інформаційного, математичного й технічного забезпечення, його відносній універсальності, можливостях керування детермінованими, стохастичними й нечіткими процесами;

– принцип оптимальності враховує потреби мінімізації часових, інформаційних і матеріальних ресурсів під час випуску й використання продукції необхідної якості;

– принцип моделювання процесів реальних об’єктів і автоматизованого керування є сучасним науковим підходом для створення, випробовування, впровадження й експлуатації автоматизованих систем з метою отримання обґрунтованих рішень з необхідною якістю в очікуваний час.

Важливо визначити множину форм прояву невизначеності. У роботі [35] автором виділені такі форми невизначеності:

– невизначеність, викликана недостатньою кількістю даних і їхньою недостатньою достовірністю внаслідок технологічних, організаційних і деяких інших причин;

– невизначеність, пов’язана з обмеженнями на часові, інформаційні й матеріальні ресурси;

– невизначеність, пов’язана з обмеженнями на фінансові можливості й ресурси;

– невизначеність через неадекватність критеріїв, моделей, що відображають поведінку об’єкта.

У ряді робіт, зокрема в [36, 37], запропоноване подібне, але дещо відмінне трактування понять і складових невизначеностей:

– події або стан середовища, обумовлені випадковістю. Ці процеси можуть бути представлені апаратом теорії стохастичних процесів і теорії ймовірностей;

– явища, які не піддаються аналізу й вимірюванню зі скінченою точністю. Ці процеси можуть бути представлені на основі принципу невизначеності квантової механіки;

– невизначеність, яка викликана нечіткістю й включає такі складові:

а) нечіткість як наслідок суб'єктивності або індивідуальності людини. Ці процеси можуть бути представлені апаратом теорії нечітких множин і їх практичних застосувань;

б) нечіткість або невизначеність у процесах мислення або умовиводу;

– невизначеність або нечіткість, характерна для природних мов:

а) нечіткість опису або подання. Ці процеси можуть бути представлені апаратом теорії нечітких множин, нечіткої логіки, модальної логіки;

б) невизначеність, пов'язана зі складністю й (або) різноманіттям семантик і структур природних мов. Ці процеси можуть бути представлені на основі семантики інформації;

– невизначеність внаслідок структурної складності й (або) різноманіття інформації. Ці процеси можуть бути представлені й досліджені на основі нечіткого структурного моделювання.

Особливістю другого підходу є значний наголос на врахування нечіткості, як фактора виникнення невизначеності, що є істотним у сучасних автоматизованих системах складних об'єктів.

Очевидно, що наведені класифікації не претендують на повноту у відображенні предмета дослідження, а істина може бути отримана, як мінімум, на основі інтеграції запропонованих підходів.

Реалізація й розвиток принципу моделювання процесів на цей час є досить перспективним. Це пов'язано з тим, що ряд процесів реальних об'єктів є недостатньо формалізованими, часто носять нечіткий характер. Тому існуючі підходи на основі подання у вигляді ймовірнісних [33, 38] часто є малоефективними, погіршують отримані результати, а в ряді випадків роблять їх неадекватними очікуваним рішенням. Моделювання процесів, включаючи імітаційне, більше не може керуватися традиційними класичними підходами [20, 38, 39] і вимагає нових рішень.

Визначимо сфери ефективного застосування управлінських рішень [40] в автоматизованих системах, які представлені у вигляді динамічних процесів. До них, передусім, варто віднести такі види робіт [40]:

– стратегічне керування (Strategic Planning) − процеси прийняття рішень, пов'язані з розподілом ресурсів, контролем ефективності організації тощо;

– адміністративне керування (Management Control) − рішення, які належать до придбання й використання ресурсів із залученням управлінського персоналу (людського фактора), поводження потенційних клієнтів і постачальників, проектування, дослідження і виготовлення нових виробів;

– оперативний контроль (Operational Control) − рішення із забезпечення ефективності організаційних дій, моніторингу якості виробів і т. ін.;

– операційне виконання (Operational Performance) − оперативні дії з метою виконання стратегічних і тактичних рішень, а також виконання поточних рішень.

Аналіз показав, що важливими для науки й практики є сфери, які належать до стратегічного (Strategic Planning) та адміністративного (Management Control) керування, що значною мірою визначає напрямок досліджень.

Як показав світовий досвід і тенденції розвитку існуючих рішень в області автоматизованого керування, важливе місце займають системи з використанням мультиагентних технологій [41, 42]. Виконані в роботі [40] дослідження показали, що світовий розвиток має стійку тенденцію вдосконалювання автоматизованих систем керування в цьому напрямку. З розвитком Internet агентні технології набули значного розвитку як в області теорії, так і для розв'язання практичних задач, це підтверджується значним обсягом досліджень, наприклад [41, 42, 43]. У цьому сенсі використання й розробка адекватних ефективних моделей для автоматизованого керування стає все більш актуальною і важливою задачею. Наведемо деякі з найбільш довершених рішень, що є ефективними, насамперед, за рахунок застосування моделей.

Так, компанія Lumina Decision Systems, разом з університетом Carnegie-Mellon, запропонувала для комерційних цілей автоматизовану систему прийняття рішень в умовах невизначеності − Analytica 2.0. Сфера її ефективного використання, згідно [40, 44], є бізнес і фінанси, керування повітряним простором, електронна комерція, екологія, енергетика й деякі інші області. Її переваги над існуючими рішеннями полягають, якщо не брати до уваги «дружній» інтерфейс, у розширених сервісних функціях, використанні ресурсів Internet, наявності засобів керування ризиком і зниженням стохастичної невизначеності на основі використання моделювання процесів методом Монте-Карло. Analytica є важливим інструментом вирішення прикладних задач, але вона функціонально обмежена тому, що не використовує знань експертів, не враховує нечіткості даних і знань. А це істотно знижує її можливості з автоматизованого керування в середовищі, де існують переважно унікальні процеси, для яких відсутні статистика та інформація про закони розподілу.

У роботі [40] наведені деякі особливості побудови систем Ithink, Stella, які запропоновані корпорацією High Performanct Systems, Inc. [45]. Система Ithink орієнтована на створення і реалізацію імітаційних динамічних моделей в області бізнесу. Система Stella використовує також імітаційні стохастичні моделі для задач синтезу систем.

Корпорація Applied Decision Analysis, Inc. [46] запропонувала орієнтовану на використання моделей систему DPL (Decision Programming Language), що дозволяє будувати дерево рішень, діаграми впливу в детермінованому та імовірнісному середовищі.

На відміну від системи Analytica 2.0, корпорація Expert Choice Inc. [47] розробила систему Expert Choice, орієнтовану на багатокритеріальний ієрархічний підхід (Analytic Hierarchy Process) до автоматизованого керування та прийняття рішень. Основою підходу є аналіз процесів, які є стохастичними. Вибір альтернатив здійснюється також на основі методів попарного порівняння, визначення пріоритетів цілей і рейтингових оцінок. Незважаючи на важливі переваги, відзначимо очевидну обмеженість такого підходу.

У роботі [48] розглянуто проблеми автоматизованого керування в умовах невизначеності з використанням марківських моделей. Об'єктом є ймовірнісні

процеси. Наведено новий алгоритм, визначена його складність порівняно з алгоритмом без моделі. Розглянуто практичні застосування моделі для процесів реальних областей: президентські вибори, фондова біржа тощо. Конкретні алгоритми, їхні особливості, складність реалізації розглядаються також у роботах [49–52]. Орієнтація переважно на імовірнісні процеси дещо знижує ефективність отриманих результатів.

Беручи до уваги викладене вище, перспективним є застосування моделей, включаючи імітаційні, як засобів розширення можливостей автоматизованих систем з керування складними об’єктами, що функціонують в умовах невизначеності. Для реалізації задач керування в реальному часі доцільно розглянути також можливість включення моделей у контур керування й прийняття рішень в умовах нечіткості вихідних даних, критеріїв і цілей.

1.5 Математичні моделі для моделювання, аналізу й реалізації задач обробки даних і керування складними об’єктами

Як випливає з викладеного вище, одним з важливих і перспективних для підвищення якості керування складними об’єктами є принцип моделювання процесів реального об’єкта. Для побудови й реалізації моделі необхідно виконати ряд цілеспрямованих дій [23]:

– на основі критеріїв якості й цілей функціонування об’єкта визначити комплекс задач моделювання;

– виконати структуризацію функціонування складного об’єкта;

– виявити фактори невизначеності, їхніх джерел, природи й видів (відсутність даних, неточність, стохастичність, нечіткість);

– розглянути можливості й шляхи зниження рівня невизначеності;

– здійснити вибір та обґрунтування математичного апарата опису процесів моделювання реального об’єкта;

– формалізувати задачі моделювання мовою обраного математичного апарата побудови моделей;

– вирішити комплекс задач моделювання;

– надати рекомендації з модифікації процесів керування на основі моделювання;

– розглянути доцільність включення моделей у контур автоматизованого керування.

Розглянемо деякі важливі й недостатньо, на наш погляд, досліджені етапи побудови моделей та їхньої реалізації на об’єктах. Найбільш істотні результати в задачах структуризації були свого часу отримані вченими Інституту проблем керування [53]. При цьому, як показали дослідження [23], структуризація й синтез системи часто зводиться до розгляду слабко формалізованих задач на основі підходів, моделей і методів математичного програмування або до вирішення задач на основі принципів агрегування [54, 55]. Варто також враховувати вимоги функціонування автоматизованих систем керування в реальному часі або в близьких режимах, специфічні особливості яких вимагають спеціальних підходів [56–58].

Розглянуті процеси в таких задачах і підходах носять детермінований і (або) стохастичний характер. Ці підходи нині малоефективні тому, що зазвичай не враховують невизначеності типу нечіткість. Рішення таких задач можливе з розробкою нових підходів, моделей і методів або з залученням розширень класичних підходів на основі положень теорії нечітких множин, нечіткої логіки, розширень теорії можливостей.

Під час вирішення задачі математичного опису об'єктів і власне побудови моделі передбачається, що об'єкт є структурованим. Як апарат використовують теоретико-множинні підходи, теорію масового обслуговування [59], мови й підходи імітаційного моделювання, наприклад, у роботах [22, 26, 53, 45], інтегрального й диференціального числення, теорію прийняття оптимальних рішень [60] за детермінованих і стохастичних функціоналів та обмежень.

Нині з'явився ряд публікацій, присвячених різним аспектам застосування моделей в автоматизованих системах. Так, у роботі [61] визначені концептуальні особливості застосування моделей в автоматизованих системах і засобах прийняття рішень. Моделі можуть бути корисними на етапах формування цілей, критеріїв, вибору альтернатив і визначення ступеня очікуваної вірогідності й можливого позитивного ефекту. Розглянуто важливі аспекти виникнення тупиків і конфліктів в ході прийняття рішень, їхній вплив на вірогідність рішень. На описовому рівні пропонуються деякі постановки окремих задач.

У роботі [62] розглядаються особливості автоматизації процесів з використанням стандартних підходів та інструментів. Автори висувають на перший план проблеми екологічного моніторингу. Інструментарій дозволяє розширити можливості з підвищення ефективності практичних результатів.

У дослідженні [63] розглянуто ймовірнісні моделі з урахуванням людського (суб'єктивного) фактору, що є актуальним для систем і об'єктів, де є можливості та ресурси набору й обробки статистичних даних.

У [64, 65] пропонується опис моделей процесів автоматизованого керування й прийняття рішень за умов ризику. Описано процеси під час вибору альтернатив для випадку участі людини в контурі керування за наявності ризику. Ризик розглядається як фактор зниження очікуваної корисності. Для врахування ризику вводяться типова функція цінності й функція надбавки. Функція надбавки оперує математичними ймовірностями подій, щоб перетворити їх у ваги рішення. Типова функція цінності та функція надбавки визначені у вигляді набору правил ідентифікації ризику, що редагуються й виконуються, і у формуванні яких бере участь експерт.

У роботах [64, 65] розглядаються ймовірнісні та детерміновані моделі в припущенні, що людина завжди діє раціонально. Це дещо ідеалізує процеси прийняття рішень в автоматизованих системах, що може привести до їхньої недостовірності.

Цілий ряд робіт і досліджень присвячено застосуванню в інформаційних системах марківських моделей, наприклад [47, 49, 50, 51, 52, 66]. Так, у роботі [66] розглянуто питання реалізації процедур перевірки процесів прийняття рішень на основі їхнього подання обчислювальною моделлю, що

використовує марківські процеси й керує стохастичними процесами. Відповідно до твердження авторів, ця модель забезпечує просте пояснення всіх головних отриманих даних, використовуючи менше параметрів, ніж попередні теорії.

Важливою складовою рішення прикладних задач підвищення вірогідності прийнятих рішень в умовах невизначеності є підходи на основі зниження рівня невизначеності [67, 68]. Ці роботи, не розглядаючи особливості використання математичного апарата, зводяться до зниження рівня невизначеності в задачах математичного програмування за рахунок введення додаткової інформації, що хоча й може містити невизначеність, але більш низького порядку. Істотним обмеженням корисності цих досліджень є те, що вони не враховують вартісні показники надбання додаткової інформації й необхідної кількості ітерацій для отримання задовільного рішення в умовах обмеження на часові й матеріальні ресурси. Таким чином, у реальних розробках розглянуто підходи, що часто є мало придатними через їхню функціональну обмеженість або наявність суто постановкових загальних рішень.

Існують роботи, в яких розглядаються процеси керування в нечіткому середовищі, наприклад, [69, 70, 71, 72]. Вони носять переважно постановковий [70] або вузько спеціалізований характер [71, 72], у цих публікаціях зазвичай відсутні відомості про засоби їхньої адаптації до інших класів об’єктів.

На цей час з’явилася низка публікацій, наприклад [73, 74], які спрямовані на дослідження в системах автоматизованого керування, процеси й об’єкти у яких представлені в нечіткому середовищі. Властивості нечіткості об’єкта розглядаються окремо від імовірнісних властивостей об’єкта дослідження. У виданнях [75, 76] і деяких інших регулярно публікуються дослідження з актуальних питань застосування нечіткої логіки для вирішення задач керування. Найбільш корисними й глибокими публікаціями можна вважати роботи [77–79]. У більшості випадків, проте, розглядаються переважно загальні описи, які не дозволяють повною мірою оцінити їхню якість. Очевидно, говорити про те, що такі рішення доведені до практичної реалізації й можуть бути використані для широкого класу задач керування, не доводиться. У ряді випадків їхнє застосування утруднене також через те, що результати носять виключно комерційний характер і недоступні для наукових досліджень, а також для сторонніх користувачів.

1.6 Застосування агентних технологій у сучасних інформаційних системах

Програмні агенти стали одним з найбільш важливих досягнень в області комп’ютерних наук у 90-х роках XX століття. Проблема інтелектуальних агентів і мультиагентних систем (МАС), яка має багаторічну історію, сформувалася в рамках робіт з розподіленого штучного інтелекту, а останніми роками претендує на одну із провідних ролей в інтелектуальних інформаційних технологіях.

Агенти використовуються в застосовних програмах, де людина і комп'ютер тісно пов'язані між собою в керуванні інформаційними процесами [80]. Велика кількість різноманітної інформації щодо програмних агентів вносить розбіжність у визначення того, що є програмним агентом. Тому спочатку необхідно визначитися щодо суті поняття «програмний агент» і виділити основні властивості програмних агентів.

Фахівці, які працюють в області проектування агентних технологій, пропонують багато варіантів визначення поняття «агент». Порівняємо й проаналізуємо деякі із цих визначень:

– The AIMA Agent [81]: «Агент – це все, що може бути описане як об'єкт, що сприймає навколишній простір через сенсори й функціонує в ньому за допомогою виконавчих елементів». AIMA («Artificial intelligence: modern approach») – відомий у США проект, що поєднує близько 200 коледжів і університетів. Очевидно, що визначення AIMA побудоване значною мірою на тому, що розуміти під функціонуванням і сприйняттям. Якщо припустити, що простір – це програмне середовище, функціонування – процес обробки вхідної інформації та отримання результату, то будь-яка програма є агентом.

– The Maes Agent [81]: «Автономний агент − це обчислювальна система, що перебуває в комплексному, динамічному просторі, приймає рішення й діє автономно в цьому просторі й у такий спосіб виконує покладені на неї функції».

– The KidSim Agent [82]: «Ми визначаємо агента як стійкий, постійний елемент програмного забезпечення, призначеного для виконання конкретних цілей». Фахівці з компанії Apple наполягають на такому: «Агенти повинні мати самостійний «почерк» вирішення поставленої перед ними задачі й певний порядок дій».

– The Hayes-Roth Agent [83]: «Інтелектуальні агенти повинні виконувати, як мінімум, три функції: сприймати динаміку навколишнього середовища, робити аналіз стану й генерувати дію або набір дій відповідно до розв'язуваної задачі».

– The IBM Agent [84]: «Інтелектуальні агенти – це об'єкти програмного забезпечення, які виконують ряд операцій для задоволення зацікавленості користувача або іншої програми, мають певний ступінь незалежності й автономності». При цьому вони використовують знання в процесі прийняття того або іншого рішення.

– The Wooldridge/Jennings Agent [85, 86]: «Агенти – це програмно-апаратні елементи комп'ютерних систем, які мають такі властивості:

– автономність поведінки: агенти функціонують без прямого втручання людини й самостійно контролюють свій внутрішній стан і дії;

– соціальна пристосованість: агенти взаємодіють з іншими агентами за допомогою мовних засобів спілкування;

– можливість реагування на зміни навколишнього середовища: агенти спостерігають за навколишнім середовищем або простором і вчасно реагують на його зміни;

– адаптивність поводження: агенти діють не тільки у відповідь на зміни навколишнього середовища, вони приймають рішення на основі власного інтелекту».

– The SodaBot Agent [80]: «Програмний агент – це програма, що веде діалог, обробляє й передає інформацію». Корпорація SodaBot займається розробкою середовища для програмних агентів. Можна помітити деяку відмінність їхнього визначення агента від попередніх, але це скоріше семантичні відмінності, ніж принципові.

– The Bristolini Agent [87]: «Автономний агент – це система, що здатна самостійно й цілеспрямовано функціонувати в реальному світі», тобто агент, у такій постановці повинен «жити» і виконувати покладені на нього функції в «реальному світі». Залишається тільки дати формальне визначення цим двом базовим поняттям.

Аналіз визначень програмних агентів дозволив виявити значну кількість властивостей цієї категорії інформаційних технологій. Внаслідок проведеного огляду можна виділити такі властивості програмних агентів.

1. Здатність реагування – ця властивість у публікаціях, присвячених програмним агентам, трактується по-різному – зокрема, є два основних визначення:

– визначає можливість агентів вчасно реагувати на дії зовнішнього середовища;

– агенти діють без будь-яких зовнішніх або внутрішніх консультацій.

Автор роботи [80] використовує це визначення для опису агентів, які сприймають своє зовнішнє оточення й реагують на зміни в цьому середовищі.

У роботі [88] ця властивість розглядається як можливість агентів проявляти реакцію на зміни за принципом «умови – дії».

2. Автономність – автономність дуже часто згадується як найбільш важлива властивість програмних агентів, яку складно визначити формально. Автори роботи [89] визначають автономного агента в такий спосіб: «Автономний агент – це система, що розміщена в навколишньому просторі, є його частиною, при цьому агент діє відповідно зі своєю програмою (задачею або задачами) так, щоб досягти поставленої перед ним мети».

У роботі [90] наведено таке твердження: «Поведінка агентів може базуватися як на своєму власному досвіді, так і на інтегрованій базі знань співтовариства агентів, що функціонують у загальному просторі».

У роботі [91] автори виділяють 4 види автономності:

– абсолютна автономність – чим менш завбачливий агент, тим він більше автономний;

– соціальна автономність – агент «знає» усе про своїх колег і про свою соціальну спрямованість;

– автономність інтерфейсу – у більшості практичних систем, де неприпустима абсолютна автономність, існує набір внутрішніх обмежень, що визначають область дії;

– автономність виконання – агент має волю в діях доти, поки працює в заданому просторі.

3. Програмність: програмний агент є, насамперед, комп'ютерною програмою. Хоча в агентах можна виділити властивості апаратні і людські, проте, дати їм чітке визначення дуже складно.

4. Раціональність поводження: якщо агента розглядають як об'єкт із елементами інтелектуальної системи такими, як мета наміру, бажання тощо, то мають на увазі інтелектуального агента. Автори роботи [92] розглядають дану властивість як внутрішній стан агента.

5. Розважливість: ця властивість використовується для того, щоб описати здатність агентів до мислення. Проявляється у випадку, коли агент, використовуючи знання й мислення, намагається визначити, що йому варто почати робити (або чи робити взагалі).

6. Комунікабельність: агенти можуть ігнорувати один одного, працювати разом або конкурувати між собою. Найбільш часто має місце кооперація (або координація дій) агентів, якщо агенти об'єднують свої зусилля для досягнення загальної мети. Координація дуже тісно пов'язана з раціональним поводженням агентів, оскільки в ході координації кожному окремому агентові необхідно знати цілі й наміри інших агентів.

Для кооперації агентів мають бути наявними відповідні координаційні стратегії, колективне навчання, ієрархічна організація тощо. Міжагентні зв'язки (комунікація) є дуже важливими для кооперації. У загальному випадку комунікація – це абстрактний рівень, заснований на алгоритмах, мовах і протоколах взаємодії.

7. Тривалість функціонування: визначає те, що агент має функціонувати протягом тривалого періоду часу.

Властивості програмних агентів, що найбільш часто зустрічаються в літературних джерелах, наведені в табл. 1.1.

Таблиця 1.1

Основні властивості програмних агентів

Властивості програмних агентів	Посилання на літературне джерело
Здатність реагування	[80], [88], [93], [94], [91]
Автономність	[91], [90], [89]
Програмність	[80], [92], [90], [93], [91]
Розсудливість	[92], [93], [94], [91]
Раціональність поведінки (мета, критерії, дії)	[92], [89], [88], [91]
Комунікативність	[89], [80], [95], [94], [91]
Тривалість функціонування	[93], [94], [90], [91]
Адаптивність	[88], [89], [91]
Орієнтованість на навколишнє середовище	[91]
Апаратна орієнтація	[80], [93], [92]
Модульність	[89], [90]
Складність	[89]
Індивідуальність	[89],[93,[92]
Цільове призначення	[91]
Робота в середовищі Internet	[89], [91]
Здатність отримувати та передавати інформацію	[89], [91]

Залежно від властивостей програмних агентів вони можуть бути віднесені до різних типів. Один з варіантів такої класифікації наведений у табл. 1.2.

Таблиця 1.2

Класифікація програмних агентів

Характеристики \ Типи агентів	Прості	Винахідливі (smart)	Інтелектуальні (intelligent)	Дійсно (truly) інтелектуальні
Автономне виконання	+		+	+
Взаємодія з іншими агентами	+	+	+	+
Стеження за середовищем	+	+	+	+
Здатність використання абстракцій		+	+	+
Здатність використання предметних знань		+	+	
Можливість адаптивної поведінки			+	+
Навчання з оточення			+	+
Толерантність до помилок			+	
Real-time виконання			+	
Природна мовна взаємодія			+	

Існує кілька підходів щодо класифікації програмних агентів [80]:

1. За ступенем мобільності, тобто за їхньою здатністю пересуватися в рамках деякого простору. За цією ознакою агенти можуть бути класифіковані як *статичні й мобільні*.

2. *Дорадчі* й *реагуючі (реактивні агенти)*. Дорадчі агенти походять від парадигми дорадчого мислення. Агенти мають мову спілкування, модель мислення й поведінки, можуть вести переговори для координації дій з іншими агентами. Роботи з дослідження реактивних агентів були проведені Бруксом [96] і Чепменом [97]. Такі агенти, на відміну від дорадчих, діють шляхом реагування на поточний стан середовища, у якому вони перебувають [98]. Зокрема, у цій роботі приводиться досить суперечливе обґрунтування того, що доцільна поведінка агентів може бути реалізована без чітких моделей і методів, характерних для традиційного штучного інтелекту.

3. Агенти можуть бути класифіковані на основі декількох основних функціональних властивостей. До переліку таких властивостей-атрибутів можуть бути віднесені: *автономність поведінки, можливість навчання, кооперація (співробітництво)*. Говорячи про автономію, варто мати на увазі принцип, відповідно до якого агент може діяти самостійно без потреби втручання людини. Саме тому такі агенти мають індивідуальну внутрішню структуру, мету функціонування, модель поведінки, щоб вирішити поставлену користувачем задачу.

Ключовий елемент автономії – це здатність агентів «взяти ініціативу на себе», а не просто діяти у відповідь на зміни середовища [85]. Взаємодія з іншими агентами є однією з основних властивостей. Для того, щоб кооперуватися, агентам потрібно мати соціальну здатність взаємодіяти з іншими агентами й, можливо, з людьми за допомогою деякої мови спілкування [85]. Говорячи про це, слід відзначити можливість агентів координувати свої дії й без співробітництва [80].

Для того, щоб агентні системи насправді були «інтелектуальними», їм необхідно «навчитися вчитися» внаслідок взаємодії із зовнішнім середовищем. По суті, це означає, що агенти є «елементами розподіленого інтелекту».

На підставі аналізу та огляду літературних джерел може бути запропонована класифікація типів програмних агентів, подана на рис. 1.4.

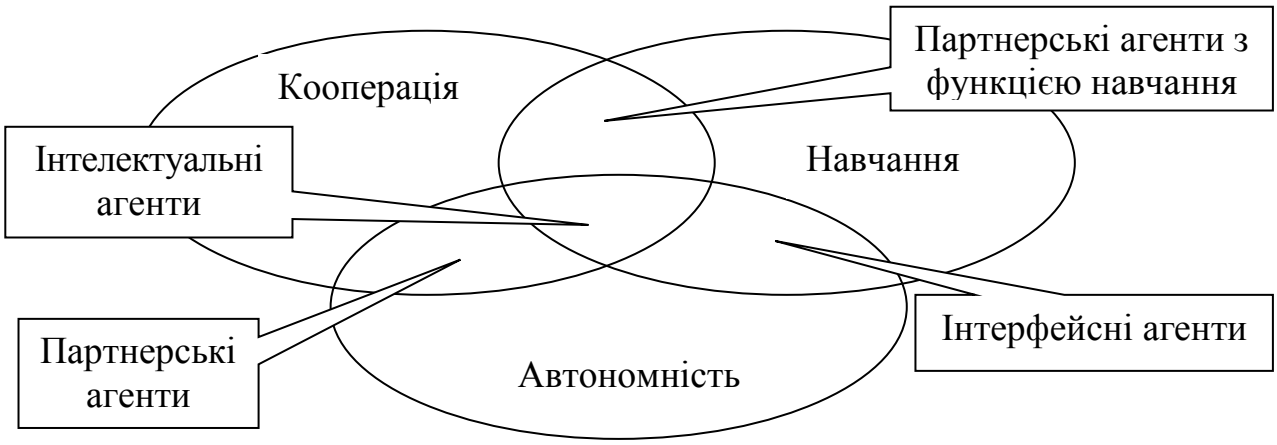


Рис. 1.4. Класифікація типів програмних агентів

Наведена типологія програмних агентів не є остаточною. Так, у випадку партнерських агентів акцентують такі властивості, як співробітництво й автономія, а не на «можливість навчання», проте це не означає, що партнерські агенти ніколи не можуть навчатися. Також, говорячи про інтерфейсні агенти, варто зробити акцент на автономію й «можливість навчання», а не на співробітництво. В ідеалі програмні агенти повинні мати всі базові якості, хоча це є більше побажанням, а не твердою вимогою.

За допомогою об’єднання класифікаційних конструкцій можна було б отримати різні типи програмних агентів: дорадчі партнерські агенти, статичні інтерфейсні агенти, рухливі реагуючі інтерфейсні агенти тощо. Проте ці категорії можуть бути використані для детального дослідження, розвитку й впровадження.

Фахівцями в області агентних технологій найчастіше розглядаються такі типи програмних агентів:

- партнерські агенти;
- інтерфейсні агенти;
- мобільні агенти;
- інформаційні Internet-агенти;
- реактивні агенти;

– гібридні агенти;
– інтелектуальні агенти.

Розглянемо докладно наведену на рис. 1.5. класифікацію програмних агентів у термінах: призначення-ціль, мотивації, ролі, недоліки й переваги.

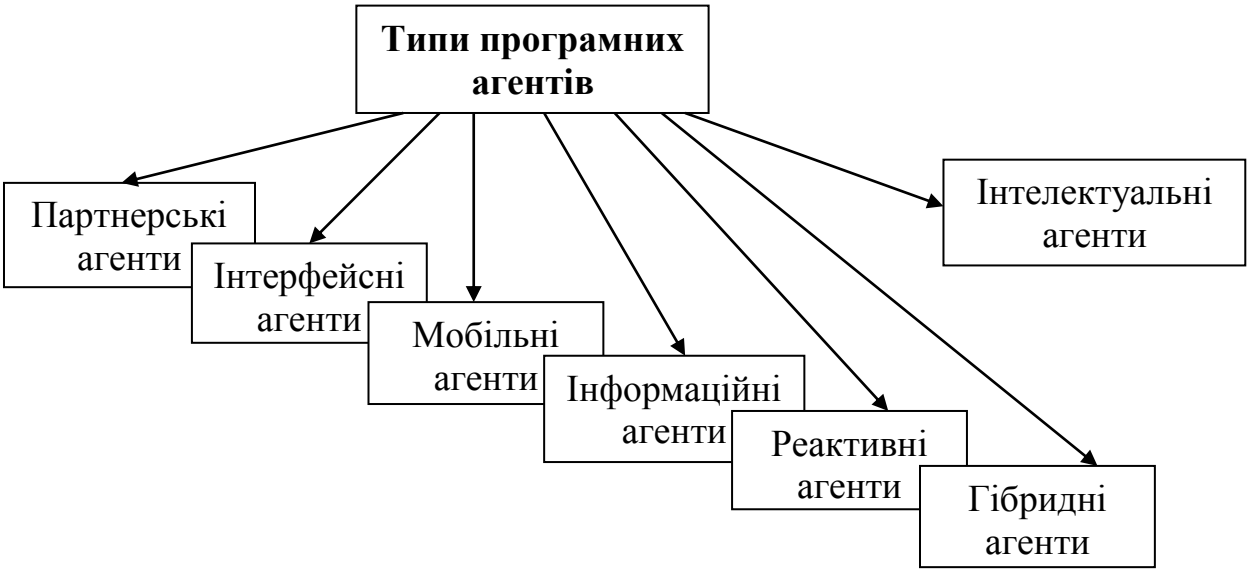


Рис. 1.5. Класифікація програмних агентів

Партнерські агенти. Партнерським агентам властиві, насамперед, автономія й взаємодія з іншими агентами для того, щоб виконати поставлену користувачем задачу.

До числа особливостей цих агентів варто віднести автономність поводження, взаємодію з іншими агентами, можливість оцінювати ситуацію та проактивність. Це означає, що вони можуть діяти раціонально й автономно у відкритих мультиагентних системах з урахуванням обмежень на час дії. Найбільш відомі на сьогодні партнерські агенти не виконують жодного комплексного навчання, проте можуть виконувати обмежене параметричне навчання або навчання з механічним запам’ятовуванням.

Мету функціонування системи партнерських агентів найбільш точно сформульовано в роботі [99]. Перефразовуючи цих авторів, можна констатувати, що мультиагентна система, яка поєднує партнерських агентів, здатна функціонувати сумарно з більшими можливостями, ніж можливості окремо взятих програмних агентів. Формально це можна представити у вигляді:

$$\pi\left(\sum_{i=1}^{n} Agent_i\right) > \pi\left[\max\left(Agent_i\right)\right],\ i=\overline{1,n}, \qquad (1.1)$$

де π – узагальнений показник сумарних дій, n – кількість партнерських агентів. Показник сумарних дій може мати такі значення, як, наприклад, швидкість, надійність, пристосованість, точність або деяка комбінація цих атрибутів.

Мотивацією для застосування мультиагентної системи, що складається з партнерських агентів, може бути один з таких випадків:

– вирішення задач, розмірність і складність яких надто велика для вирішення одним програмним агентом;

– забезпечення взаємодії різнорідних систем з успадкуванням;
– інтеграція розподілених обчислювальних систем;
– пошук рішень у системах, які базуються на розподілених джерелах інформації.

Інтерфейсні агенти. Під час розгляду інтерфейсних агентів акцент варто робити на таких властивостях, як автономність і можливість навчання, що дозволяє їм вирішувати різні задачі користувачів. Патті Маєс, ідеолог цього класу агентів, відзначає, що основною відмінністю інтерфейсних агентів є те, що вони виступають у ролі *особистого асистента користувача* і працюють за його завданням в інформаційному середовищі [81]. При цьому взаємодія з користувачем відрізняється від взаємодії з іншими партнерськими агентами. Взаємодія з користувачем вимагає формальної мови опису задач, а спілкування агентів – спеціального протоколу обміну даними.

Інтерфейсні агенти можуть надавати допомогу користувачеві під час процесу навчання у роботі з прикладними програмами такими, як електронні таблиці, текстові редактори тощо. Агент може контролювати дії, які робить користувач, аналізувати їх і пропонувати кращі варіанти вирішення задачі.

По суті, агент користувача є автономним особистим асистентом, що допомагає здійснювати взаємодію як у роботі з програмними застосуваннями користувача, так і в процесі вирішення допоміжних задач в інформаційному середовищі [81]. Схему взаємодії інтерфейсного агента наведено на рис. 1.6.

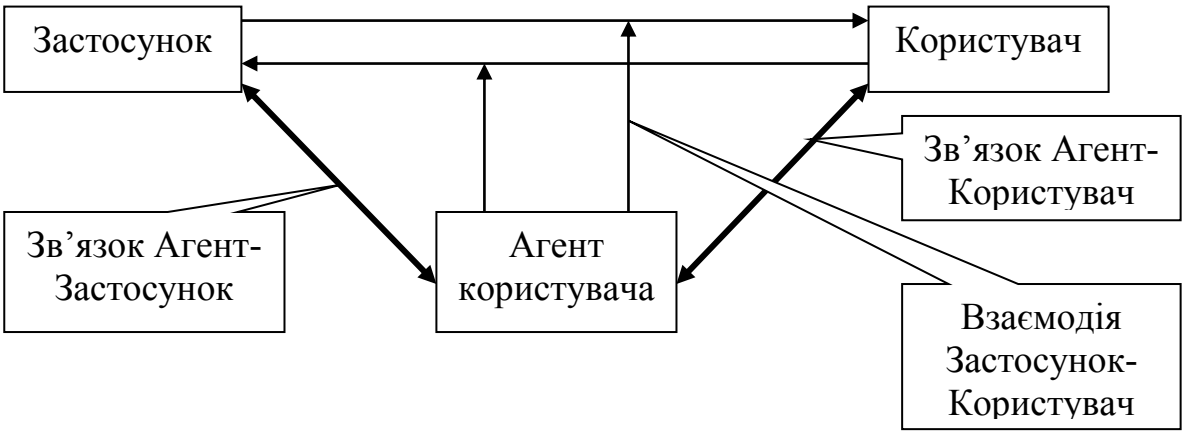


Рис. 1.6. Схема взаємодії агента у системі «Користувач – Застосунок»

Інтерфейсний агент з можливостями навчання – це агент, що на відміну від будь-яких інших видів агентів, використовує методи машинного навчання для того, щоб надати користувачеві «інтелектуальний» інтерфейс для виконання різних задач в інформаційному середовищі [100].

Дослідження в області інтерфейсних агентів, як це показано в [81], проводяться в напрямку розробки інтелектуальних інтерфейсів класу «комп’ютер–людина–комп’ютер». Аргумент на користь такого напрямку досліджень: існуючі інтерфейси «користувач-обчислювальна система» тільки

реагують на пряму маніпуляцію, тобто комп'ютер є пасивною складовою й завжди чекає того, щоб виконати конкретні інструкції користувача.

Це забезпечує лише незначну проактивну допомогу у вирішенні складних задач [81]. Таким чином, замість того, щоб формувати прямі команди до деякого інтерфейсу, користувач міг би бути залучений до кооперативного процесу, у якому людина й програмні агенти спільно вирішують складні задачі в системі «користувач – інформаційна система».

Можна сформулювати найближчу мету в розвитку цього класу агентів: поступовий перехід від концепції прямого маніпулювання до концепції, відповідно до якої користувач делегує частину повноважень програмним інтерфейсним агентам для ефективної адаптації користувачів до взаємодії з інформаційним середовищем. При цьому агенти можуть бути досить надійними виконавцями прикладних задач, які їм доручить виконувати користувач.

У перспективі може бути поставлено задачу більш високого рівня – «за певних умов інтерфейсний агент може програмувати себе» [101].

Підбиваючи підсумки, можна констатувати, що інтерфейсний агент є квазіінтелектуальним компонентом програмного забезпечення, що допомагає користувачеві ефективно взаємодіяти з інформаційним середовищем [81].

Мобільні агенти. Мобільні агенти – це обчислювальні програмні модулі, здатні до пересування «у глобальних комп'ютерних мережах» (таких, як, наприклад, Internet), взаємодії із зовнішніми обчислювальними системами на предмет збору інформації в інтересах користувача й «повернення додому» після виконання поставленого завдання.

Такі обов'язки можуть полягати в широкому діапазоні задач: від бронювання авіаквитків до керування мережею передачі даних. Однак мобільність є необхідною, але не достатньою умовою для існування цього класу агентів. Мобільні агенти мають взаємодіяти з іншими програмними агентами, наприклад, з партнерськими агентами. Мобільні агенти мають кооперуватися або встановлювати зв'язок з іншими агентами, визначати місцезнаходження об'єктів в інформаційному просторі, використовуючи методи, відомі іншим агентам.

Мета функціонування, що зазвичай ставиться перед мобільними агентами:

– попередній аналіз поточної інформації;

– розподілені обчислення: потужність локальних систем обробки даних може бути обмеженою (обробка й організація пошуку): саме в цьому випадку досягається значний ефект за рахунок використання мобільних агентів;

– асинхронна обробка даних: користувач може конфігурувати своїх агентів на виконання різних функціональних задач;

– мобільні агенти дають можливість для радикального переосмислення класичних процесів проектування в цілому. Це може призвести до появи нових продуктів, які базуються на технологіях мобільних агентів.

Інформаційні/Internet агенти. Інформаційні агенти з'явилися у зв'язку з нагальною потребою керування всіма зростаючими потоками інформації як

інструментальними засобами підтримки програмних застосувань користувача. Інформаційні агенти виконують задачі з керування й маніпулювання інформацією від багатьох розподілених джерел даних.

Складно сформулювати чіткі відмінності цього типу програмних агентів від інших, раніше розглянутих (наприклад, партнерських або інтерфейсних). Метою функціонування інформаційних агентів, як показано в роботі [80], є те, що вони можуть ліквідувати проблему інформаційного перевантаження і при цьому вони є загальним елементом інформаційного керування. Інформаційні агенти мають бути забезпечені можливістю отримання знань про те, *де* шукати інформацію, *як* її знайти і з *чим* її порівнювати.

Інформаційні агенти можуть мати змінну функціональну структуру, адаптивно пристосовуючись до типу розв'язуваних задач. Особливим типом агентів є Internet-агенти.

Сучасний інформаційний простір містить величезну кількість інформації, при цьому найпоширенішим і найдоступнішим засобом її добування є машини пошуку. У таких системах використовуються Internet-агенти – спайдери (spiders). Спайдер пересувається від сайту до сайту, іноді від сервера до сервера, використовуючи пріоритети, посилає звіт пошуковій машині про результати дій. Основними характеристиками машин пошуку є: тип мови запитів, вид і подання вхідних і вихідних документів, час індексації й пошуку, обсяг індексу. Відомі пошукові системи: Google, Yahoo, MSN, Microsoft Live Search, AltaVista, Exite, AOL, Lycos. З урахуванням морфології російської мови – Яндекс, Rambler, Mail.ru.

Розглянемо типову організацію машин пошуку, наведеної на рис. 1.7

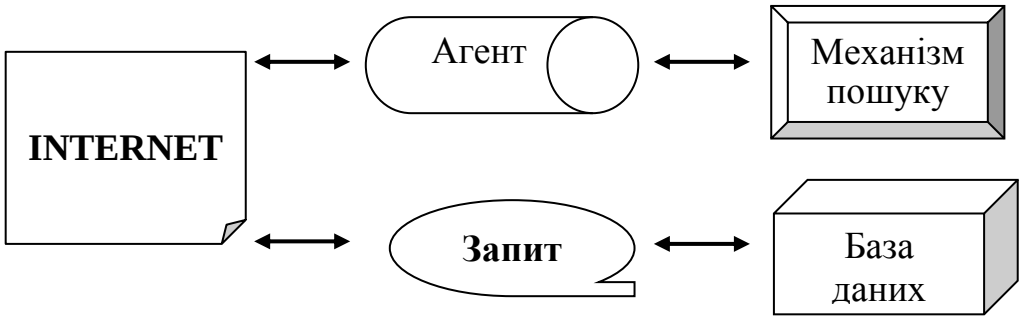


Рис. 1.7. Структура машини пошуку

Пошукова система починає процес пошуку нових сайтів з відомих документів і переходить за посиланнями на інші сторінки. Вона розглядає мережу як орієнтований граф і працює в такому режимі:

– знайти новий документ;
– відзначити його як переглянутий;
– розшифрувати посилання;
– проіндексувати зміст документа.

Пошукова система працює в двох режимах: пошук у реальному часі й індексування документів. Агенти відповідають за пошук і добування інформації. Отримавши задачу на пошук, Internet-агент або повертає зміст

документа, або пояснення, чому даний документ не можна доставити. Агенти запускаються як окремі процеси, одночасно можуть використовуватися декілька агентів. У базі даних зберігаються метадані документів, зв'язку, індекси, службова інформація. Кожен об'єкт – документи, сервери, посилання – запам'ятовується в окремому каталозі. Такий поділ дозволяє швидко визначити невикористовувані або часто використовувані сервери.

Реактивні агенти. Реактивні агенти є спеціальною категорією агентів, що не містять моделей навколишнього простору. Разом з тим, вони діють (реагують) на поточний стан навколишнього середовища, у якому перебувають.

Реактивні агенти вперше розглядалися в роботах Брукса [96], Агре і Чепмена [102]. На цей час створено багато різних моделей і архітектур цього класу агентів. Але всіх їх поєднує одне – ці агенти є відносно простими програмними агентами. Маєс [81, 103] виділяє кілька особливостей, що характеризують реактивних агентів.

По-перше, це – «функціональність». Реактивні агенти розглядаються як набір модулів, що діють автономно і відповідають за специфічні задачі (наприклад, зчитування, керування, обчислення тощо). По-друге, реактивні агенти мають тенденцію реагувати на зображення у вигляді образів, на противагу високорівневим символічним зображенням, що сприймаються іншими типами агентів.

Слід зазначити, що на цей час існує обмежена кількість прикладів проактивного використання реактивних агентів. Частково через це не існує стандартного методу опису їхньої дії. Реактивні агенти використовуються в таких технічних системах, як роботи. Наприклад, дослідники концерну Philips створили цифрові відео і тривимірні маніпулятори на основі реактивних агентів [80].

Гібридні програмні агенти. У попередніх підрозділах було розглянуто п'ять типів агентів: партнерські, інтерфейсні, мобільні, Internet-агенти і реактивні агенти. Питання про те, який із програмних агентів кращий, носить скоріше академічний характер, оскільки кожен тип має свої переваги й недоліки. У зв'язку з цим у колі фахівців з інтелектуальних систем з'являються думки про необхідність розробки і досліджень такого типу програмних агентів, як «гібридні», що поєднують у своїй структурі переваги інших типів.

На рис. 1.8 подано архітектуру гібридного програмного агента, запропоновану Фішером та іншими авторами [104, 93]. Як зовнішні впливи виступають повідомлення і команди від менеджера агентів, а також повідомлення від операційної системи. За допомогою спеціальних функцій і процедур агент може взаємодіяти з пристроями комп'ютера (для звернення до твердого диску і мережних пристроїв), робити звернення до баз даних і файлової системи комп'ютера.

Архітектура гібридного агента містить базу знань і блок керування, що складається з підсистеми моделювання поведінки, локальної підсистеми планування і підсистеми кооперативного планування. Очевидно, що розглянута

архітектура є деяким симбіозом структур дорадчої і реактивної філософії. Реагуюча частина містить набір зразків і правил поведінки (модель) у конкретних ситуаціях. Вона є рівнем «майстерності» агента – здатністю швидко розпізнавати ситуації. Проміжний рівень у вигляді підсистеми локального планування дає можливість агенту кооперуватися з іншими структурами для ефективного виконання поставленої задачі [94, 95].

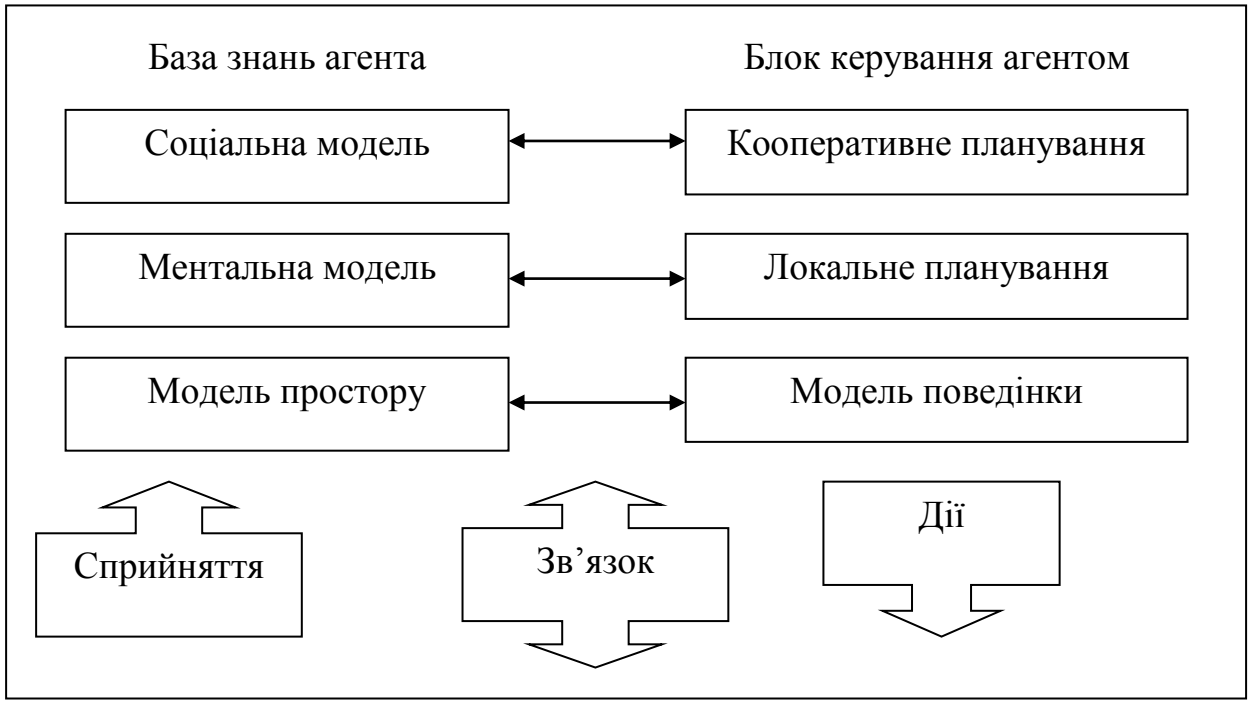


Рис. 1.8. Структурна схема гібридного програмного агента

На основі розглянутої архітектури було розроблено інтелектуальну мультиагентну систему керування вантажно-розвантажувальними і монтажними роботами в доках [80].

1.7 Мультиагентні технології керування інформаційним простором розподілених обчислювальних систем

Останні десять років успішно розвивається новий напрямок в області інтелектуальних систем – розподілені інтелектуальні програмні системи. Такі системи можуть бути реалізовані різними способами, але саме мультиагентні системи (МАС) концентрують усі необхідні для таких технологій властивості з найбільшою виразністю і повнотою. Результати впровадження агентних технологій підтвердили передбачувану перспективність цього напрямку. Технологія і теорії агентів продовжують розвиватися в рамках дослідницьких і комерційних проектів [105, 106].

Особлива увага приділяється інтеграції методів штучного інтелекту, що дотепер знаходили застосування переважно в дослідницьких роботах. Внаслідок практичного застосування технології мультиагентих систем з'явилися реальні комерційні застосування:

– персональні помічники;

– обробники пошти;

– програми для електронної комерції;

– комп’ютерні ігри;

– системи керування і контролю складними процесами в медицині, промисловості;

– системи пошуку й обробки інформації.

Мультиагентні системи на основі інтелектуальних програмних агентів є логічним продовженням розвитку двох напрямків – сучасних мережевих інформаційних технологій та інтелектуальних систем.

Програмні агенти як основна підсистема мультиагентих систем – новий клас систем програмного забезпечення, що діє від імені користувача. Вони є могутньою абстракцією для «візуалізації» і структурування складних процесів проектування, експлуатації і супроводу мультиагентних систем. Якщо процедури, функції, методи і класи – відомі абстракції, які розробники програмного забезпечення використовують щодня, то програмні агенти – принципово нова парадигма, що вимагає додаткових досліджень для впровадження в програмні системи [107].

Розвиток мультиагентних систем був би неможливий без попереднього досвіду і практичного освоєння концепції відкритих систем, що характеризуються такими властивостями:

– масштабованість – можливість зміни набору складових системи;

– мобільність – простота переносу програмної системи на різні апаратно-програмні платформи;

– інтероперабельність – здатність до взаємодії з іншими системами.

Мультиагентна технологія у своїй основі побудована на «клієнт-серверній» технології (client-server) [108, 109]. На цей час виділяються дві моделі «клієнт-серверної» взаємодії:

– «товстий клієнт/тонкий сервер» – варіант, який найчастіше зустрічається, де серверна частина реалізує тільки доступ до ресурсів, а основна частина застосунків знаходиться на клієнті;

– «тонкий клієнт/товстий сервер» – модель, активно використовувана у зв’язку з поширенням Internet-технологій, Web-браузерів, де клієнтські застосунки забезпечують реалізацію інтерфейсу, а сервер поєднує інші частини програмних застосувань.

В процесі створення МАС можуть з успіхом використовуватися обидві моделі. Незалежно від використовуваної моделі засобу виконання розподілених застосунків, якими є МАС, можуть використовувати статичний чи динамічний підхід. За статичного підходу передаються тільки дані, а за динамічного існує можливість передачі коду програми. Тому динамічний підхід передбачає використання парадигми мобільних агентів. Фахівці в області агентних технологій вважають, що мобільні агенти забезпечують прогресивний метод роботи у мережних застосунках [110].

Використання МАС технології на основі мобільних агентів має такі переваги:

– дозволяє виконувати асинхронні обчислення (наприклад, запустивши агента на виконання задачі, користувач може переключитися на виконання

іншої задачі і навіть від'єднуватися від мережі, результат буде доставлений агентом адресату після виконання задачі);

– полегшує координоване виконання взаємозалежних задач;

– дозволяє перебороти обмеження локальних ресурсів обчислювальної мережі;

– зменшує час і вартість передачі даних (за великих обсягів даних замість передачі всієї неопрацьованої інформації з мережі на хост посилається агент, що вибирає тільки необхідну інформацію і передає її користувачеві).

Мобільні агенти є перспективним напрямком розвитку для МАС систем, але на цей час немає єдиних стандартів їхньої розробки і все ще залишається невирішеним ряд проблем, таких як легальний спосіб пересування агентів мережею, захист від переданих мережею програмних вірусів тощо.

На цей час найбільш відомими технологіями реалізації статичних і динамічних розподілених застосунків є програмування сокетів, виклик вилучених процедур – RPC (Remote Procedure Call), DCOM (Microsoft Distributed Component Object Model), Java RMI (Java Remote Method Invocations) і CORBA (Common Object Request Broker Architecture).

Модель Microsoft DCOM є об'єктною моделлю, що підтримується такими операційними системами, як Windows, Sun Solaris, Unix та ін. Основна її цінність у представленні можливостей інтеграції програмних застосунків, реалізованих у різних системах програмування.

Java RMI – розподілена об'єктна модель, в якій комп'ютери виконують ролі клієнта і сервера тільки для конкретного виклику. При цьому на сервері створюються деякі об'єкти, які можна передавати мережею, або їхні методи визначаються як доступні для виклику вилученими об'єктами, а на клієнті реалізуються застосунки, що використовуються вилученими об'єктами. Особливістю RMI є можливість передачі мережею не тільки методів, але й самих об'єктів, що забезпечує, реалізацію технології мобільних агентів.

CORBA є частиною OMA (Object Management Architecture), розробленої для стандартизації архітектури й інтерфейсів взаємодії об'єктно-орієнтованих додатків. Інтерфейси між CORBA-об'єктами визначаються через спеціальну мову IDL (Interface Definition Language), що є мовою опису інтерфейсів. Самі інтерфейси можуть при цьому бути реалізовані на будь-яких інших мовах програмування і приєднані до CORBA-додатків. У рамках стандарту передбачається, що CORBA-об'єкти можуть бути зв'язані з DCOM-об'єктами через спеціальні CORBA-DCOM мости (bridges).

За даними ряду джерел, технології Java RMI і CORBA є на цей час найбільш гнучкими й ефективними засобами реалізації розподілених застосувань. Ці технології дуже близькі за своїми характеристиками. Основною перевагою CORBA є інтерфейс IDL, що уніфікує засоби комунікації та інтероперабільність між застосунками. З іншого боку, Java RMI є більш гнучким і могутнім засобом створення розподілених застосувань на платформі Java, що включає можливість реалізації мобільних агентів. На цей час не є цілком зрозумілим, за якою із платформ майбутнє в боротьбі за мультиагентні системи. Цілком імовірно, що в цей процес може втрутитися і концепція на платформі DCOM.

Внаслідок проведених досліджень можна зробити такий висновок: мультиагентний підхід в ході вирішення специфічних проблем керування інформаційними ресурсами має переваги перед іншими технологіями, тому що в основі його містяться такі основні принципи:

- принцип розподіленого середовища обчислень;
- принцип вилучених обчислень (Remote Evaluation);
- перетворення, кодування після виконання дії (Code On Demand);
- мобільність об'єктів (агентів) обчислювального середовища (Mobile agents).

Перспективною і важливою також є можливість функціонування мультиагентних технологій як у детермінованому, так і в стохастичному і нечіткому просторах станів розподілених інформаційних ресурсів.

1.8 Нечіткі процеси в інформаційних інтелектуальних системах, що функціонують за умов невизначеності

Нечіткі уявлення про явища, процеси та їхню взаємодію в усіх сферах людської діяльності були і є надзвичайно важливими на всіх етапах розвитку суспільства. Для людини завжди були очевидними, зрозумілими і природними поняття: «людина високого зросту», «сильний вітер», «маленьке містечко», «висока продуктивність», «велика погрішність», «невелика кількість ресурсу» тощо. Нечіткий висновок з нечітких умовних тверджень типу: «*Якщо* на вулиці сильний мороз і сильний вітер, *то* одяг має бути теплим», також характерний для людської діяльності. У зв'язку з цим виникає задача представлення, опису й обробки нечітких подій з метою ефективного використання їх у системах обчислювального інтелекту [111] предметних областей.

У точних науках дослідник оперує точними, часто ідеалізованими уявленнями, що в практичних застосуваннях іноді погіршує і навіть знецінює результати класичної математики. Необхідність прийняття рішень в умовах обмежених ресурсів, невизначеності, неточності, нечіткості в ряді практичних застосувань (маркетингові дослідження, передпроектні дослідження, унікальні одиничні виробництва, умови екстремальних і надзвичайних ситуацій) у більшості випадків призводить до труднощів у застосуванні точних класичних підходів.

Стає очевидним твердження професора Л. Заде в передмові до книги A. Kaufmann «Introduction a la theorie des sous-ensembles flous», виданої в 1972 році: «Теорія нечітких множин – це, по суті справи, крок на шляху до зближення точності класичної математики і проникаючої всюди неточності реального світу, до зближення, породженого невпинним людським прагненням до кращого розуміння процесів мислення і пізнання».

У багатьох практичних рішеннях донедавна домінувало уявлення про системи керування різної складності, з різним ступенем участі людини в контурі керування і прийняття рішень (автоматичних, автоматизованих), як про об'єкти, реалізовані на основі детермінованих і (або) стохастичних моделей. Дослідженню таких об'єктів приділялася велика увага, були отримані важливі для теорії і практики результати, наприклад у [112], з актуальних питань сучасної теорії оптимального керування і побудови систем. Навіть в об'єктах

промисловості й енергетики, для яких характерне функціонування в умовах суттєвої невизначеності, нечіткості, донедавна пропонувався детермінований чи, принаймні, ймовірнісний підхід, який часто доповнюється адаптивними контурами керування. Таким чином, у ряді випадків, навіть в умовах суттєво нечітких процесів, використовувалися підходи і математичний апарат, які не повною мірою враховували специфіку процесів і об'єктів, що знижувало ефективність систем і приводило іноді до їхньої принципової непридатності для функціонування в реальних умовах.

На цей час назріла необхідність розробки нових підходів та інтелектуальних обчислювальних технологій обробки інформації. Вперше положення нової теорії, заснованої на нечітких представленнях про процеси і явища, були запропоновані й розроблені американським ученим Л. Заде [113–115], перші основні роботи якого були опубліковані в 1965 році. Ці роботи і ряд наступних створили теоретичну, методологічну і практичну базу ефективного використання і розвитку теорії нечітких множин, систем обчислювального інтелекту та їхніх застосувань.

У [116] визначені деякі обставини, що приводять до необхідності роботи з вивчення і використання моделей і рішень на основі логіко-лінгвістичних уявлень. Тут можна виділити такі причини: не всі цілі керування об'єктом можуть бути представлені у вигляді кількісних співвідношень; між деякими параметрами, які впливають на процес керування, не вдається або дуже складно встановити точні кількісні залежності; процес керування є багатокроковим чи багатоетапним, а зміст кожного кроку й етапу не завжди може бути заздалегідь однозначно визначено; існуючі описи і представлення занадто громіздкі і їхнє практичне використання неможливе; об'єкти і процеси розвиваються в часі, що вимагає змін законів керування, а їхній зміст неясний чи представлений нечітко; цілі функціонування об'єкта не завжди чітко й кількісно виражені. Варто доповнити наведені обставини тим фактом, що часто об'єкт функціонує в умовах обмежень як на часові, так і на матеріальні ресурси. Деякі характеристики об'єктів недоступні для кількісних оцінок і можуть бути представлені тільки лінгвістично. До таких об'єктів слід віднести: об'єкти, що функціонують в екстремальних, надзвичайних умовах; екологічні об'єкти, системи екологічного моніторингу [117]; об'єкти і системи енергетики [118]; об'єкти, засоби подання й інтелектуальні нелінійні системи керування цими об'єктами [119], системи, що розвиваються [120], виробничі системи, що включають рішення на основі положень та ідей штучного інтелекту [121]. Слід підкреслити, що рішення доцільно розглядати на етапах життєвого циклу систем, виробів і технологій [122].

Надалі ми розглядатимемо суттєву нечіткість процесів у контексті нечіткості як наслідку суб'єктивності або індивідуальності людини; нечіткості або незрозумілості в процесах мислення чи умовиводу. Коли ми говоримо про суттєву нечіткість, то маємо на увазі, що властивість нечіткості стосується більшості визначальних характеристик розглянутих процесів. Вважаємо також необхідним доповнити ці два класи уточненням, що нас цікавитиме також нечіткість у сприйнятті процесів, які виникають внаслідок жорстких обмежень на ресурси, включаючи часові.

1.9 Аналіз підходів до обробки нечітких даних і знань

Відповідно до підходу, запропонованого Л. Заде, традиційна теорія множин часто розглядається як окремий випадок теорії нечітких множин (у науковій літературі часто говорять про теорію нечітких підмножин). Для звичайних множин очевидним є поняття характеристичної функції. Нехай E° – множина, A° – підмножина множини E. Факт належності деякого елемента підмножині можна представити характеристичною функцією вигляду

$$\mu_A(x) = \begin{cases} 1, if\ x \in A, \\ 0, if\ x \notin A. \end{cases} \tag{1.2}$$

Припустимо, що характеристична функція (1.2) може приймати деякі значення на інтервалі $[0,1]$, тобто належність буде нечіткою, що і визначає особливість нечіткої множини. Введемо визначення нечіткої підмножини згідно з Л. Заде. Нехай існує множина, скінчена або ні, та x – елемент E. Тоді нечіткою підмножиною $\tilde{A}$ множини E називатимемо множину упорядкованих пар

$$\left\{\left(x \mid \mu_{\tilde{A}}(x)\right)\right\},\ \forall x \in E, \tag{1.3}$$

де $\mu_{\tilde{A}}(x)$ – ступінь належності x до $\tilde{A}$.

Представлення нечітких процесів на основі функцій належності. Важливою проблемою, що визначає ефективність нечітких систем, є побудова функцій належності, забезпечення їхньої коректності в практичних реалізаціях, розробка правил побудови. У реалізаціях деякої нечіткої системи, наприклад, функції належності твердження «величина x має мале значення» може бути представлена у вигляді:

$$\mu(x) = e^{-kx^2},\ k > 0.$$

Твердження «величина x має велике значення» може бути подана як [123]:

$$\mu(x) = 1 - e^{-kx^2},\ k > 0.$$

Від коректності завдання функцій багато в чому залежить ефективність практичних рішень у нечіткому середовищі динамічних взаємодіючих процесів.

Відомі роботи [124–127], у яких узагальнюються й обґрунтовуються можливі підходи до побудови функцій належності й визначення ступеня належності. В історичному аспекті близькою до проблеми коректної побудови функцій належності можна вважати роботу [128]. Відповідно до положень цієї роботи, проблема групового вибору – це проблема зведення декількох індивідуальних думок про порядок переваги об’єктів у єдину «групову» перевагу в задачах прийняття рішень. Під час формулювання групової переваги вибір середнього з попереднім відкиданням мінімальних і максимальних значень, як це прийнято, наприклад, у деяких видах спорту, прийнятно за наявності чітко виробленої системи і критеріїв оцінок, що не завжди має місце на практиці. Застосування правила більшості може привести до парадокса

такого типу: нехай існує три експерти, що на множинах $\{a, b, c\}$ мають такі порядки переваг $(a, b, c), (b, c, a), (c, a, b)$. За правилом більшості отримуємо, що a краще b, b краще c, c краще a, але не гірше, як очікувалося [128]. Очевидно, що парадокс може бути усунений простішим чином на основі введення поняття про відстань [123] і використання його в задачах групового вибору. Питання створення «розумних» принципів погодженості в експертних оцінках хвилювали відомих математиків, серед яких К.Дж. Ерроу, Дж. фон Нейман та ін. Їхні роботи й отримані результати багато в чому визначають сучасні підходи до погодженого формування функцій належності. Важливі і досить повні результати зі створення принципів формування функцій належності отримані авторами [123–125], наукові інтереси яких значною мірою стосуються дослідження і створення нечітких систем.

У роботі [125] запропоновано два методи побудови функцій належності: метод прямого оцінювання (метод оцінки величини); метод зворотного оцінювання, що звичайно використовується для перевірки інформації, отриманої від декількох експертів. На основі експериментальних досліджень визначено, що індивідууми можуть сприймати нечіткість по-різному. Це у свою чергу вимагає розробки більш загальних моделей і процедур оцінювання результатів висновку на основі інформації, отриманої в декількох випробуваннях.

У [124] пропонуються підходи, засновані на такій класифікації:

1) область визначення нечіткої множини: числова дискретна – a; числова неперервна – b; не числова – c;

2) застосовуваний спосіб експертного оцінювання: індивідуальний – d_1; груповий – d_2;

3) тип використовуваної експертної інформації: порядкова – e_1; кардинальна – e_2;

4) інтерпретація даних експертного оцінювання: імовірнісна – D; детермінована – N;

5) застосування стандартних наборів таблиць, графіків та аналітичних залежностей функцій належності.

Підхід 5) є ефективним у більшості практичних реалізацій.

У цьому випадку розробник сам визначає найбільш прийнятну залежність, визначає її параметри й у разі потреби виконує корекцію як параметрів, так і самої функції.

Пропоновані підходи є досить ефективними, засновані на гнучкій структурі оцінок, орієнтовані на різні способи експертних оцінок. Проте слід зазначити, що в роботі [124] уявлення про функції належності часто розглядаються в імовірнісному аспекті, а це в ряді випадків викликає певні труднощі, пов’язані з принциповими розходженнями положень теорії нечітких множин і теорії ймовірностей.

У [123] запропоновано огляд деяких найпростіших функцій належності для таких універсальних множин: R^+ – невід’ємних дійсних чисел; N – натуральних чисел; R – дійсних чисел; Z – цілих чисел. Для універсальних

множин R^+, N запропоновано функції належності твердження «величина x має мале значення», «величина x має велике значення». Для універсальних множин R, Z запропоновано функції належності твердження «величина $|x|$ має мале значення», «величина $|x|$ має велике значення».

Очевидно, що на практиці як незалежна змінна може виступати також лінгвістична змінна типу «багато», «дуже багато», «середня кількість», «мало», що важливо в реалізаціях.

У практичних застосуваннях можна вважати, що проблема побудови коректних підходів до створення функцій належності в нечіткій логіці загалом вирішена і доведена до практичних реалізацій. Це надає дослідникам право використовувати результати зазначених робіт на практиці.

Особливості побудови нечітких відношень представлення динамічних взаємодіючих нечітких процесів. В ході вирішення теоретичних і практичних задач, пов'язаних із процесами, що характеризуються істотною нечіткістю, доцільно визначити нечітке відношення на множинах нечітких процесів (1.3). Нехай задані дві множини $\tilde{E}_1, \tilde{E}_2$, причому $\tilde{x} \in \tilde{E}_1, \tilde{y} \in \tilde{E}_2$. Тоді нечітке відношення $\tilde{R} = \tilde{E}_1 \times \tilde{E}_2$ запишемо, як

$$\forall (\tilde{x}, \tilde{y}) \in \tilde{E}_1 \times \tilde{E}_2 \mid \mu_{\tilde{X} \times \tilde{Y}}(\tilde{x}, \tilde{y}) \in \{\mu_i\}. \qquad (1.4)$$

Представивши нечіткі множини, як (1.3) і нечіткі відношення, як (1.4), ми можемо оперувати з лінгвістичними представленнями нечітких процесів і їхньою взаємодією. Відзначимо, що декількома дослідниками 123, [129–131] досить глибоко пророблено і визначено правила виконання і властивості операцій над нечіткими множинами і нечіткими відношеннями, деякі з яких зручно використовувати в наших подальших дослідженнях.

Розглянемо особливості побудови і використання нечітких лінгвістичних представлень. Згідно з [131, 132], нечіткі лінгвістичні представлення – це формальне представлення систем, реалізованих за допомогою умов ***ЯКЩО, ТО*** $(if, then)$. Хоча нечіткі лінгвістичні представлення зазвичай формулюються природною мовою [132], проте вони мають точні математичні основи, що включають нечіткі множини і нечіткі відношення. Кодування даних і знань здійснюється на основі інструкцій у формі: *if* – набір умов виконаний, *then* – набір наслідків може бути виведений. Наприклад, в умовах виробництва бажане поводження системи [131] може бути представлене групою правил, об'єднаних за допомогою зв'язки *ELSE*:

error is ZERO and Δ*error is ZERO then* Δu *is ZERO ELSE*

if error is PS and Δ*error is ZERO then* Δu *is NS ELSE* (1.5)

if error is SMALL and Δ*error is NS then* Δu *is BIG,*

де *error*, Δ*error* (помилка і зміна помилки) – лінгвістичні змінні, що описують вхідні змінні системи, Δu – лінгвістична змінна, що описує зміни даних на виході.

Група правил (1.5) формує нечіткий алгоритм, реалізація якого дозволяє досягти системою поставленої мети. У зв'язку з цим виникає проблема використання співвідношення (1.5). Нехай задана деяка чітка функціональна залежність $y = f(x)$. Зазначена процедура зазвичай розглядається як виведення значення y за відомого значення x. Близьким за змістом є нечітке логічне виведення, що оцінює нечіткий лінгвістичний опис. Існує, принаймні, дві проблеми в реалізації процедур нечіткого логічного виведення. Нехай відоме значення нечіткого логічного входу до системи $\tilde{A}'$ і необхідно отримати нечітке значення виходу системи $\tilde{B}'$. Друга проблема пов'язана з тим, що відомо значення нечіткого логічного виходу із системи $\tilde{B}'$, а необхідно отримати значення нечіткого логічного входу до системи $\tilde{A}'$. Перша проблема визначає реалізацію прямої процедури нечіткого логічного виведення, засновану на узагальненому способі modus ponens (GMP). Друга проблема визначає реалізацію зворотної процедури нечіткого логічного виведення, засновану на узагальненому способі modus tollens (GMT).

Розглянемо таку просту процедуру:

$$\begin{array}{l} \textit{if } x \textit{ is } \tilde{A} \textit{ then } y \textit{ is } \tilde{B} \\ \quad x \textit{ is } \tilde{A}' \\ \hline \quad y \textit{ is } \tilde{B}', \end{array} \tag{1.6}$$

де відомий антецедент $\tilde{A}'$, а результат (консеквент) $\tilde{B}'$ – не відомий.

Процедура (1.6) може бути реалізована на основі узагальненого способу GMP [131] у такий спосіб:

$$\tilde{B}' = \tilde{A}' \circ \tilde{R}(x, y), \tag{1.7}$$

де $\tilde{R}(x, y)$ – відношення, отримане з *if / then* правила (1.6).

У ряді практичних реалізацій важливо реалізувати зворотну процедуру логічного виведення.

Розглянемо просту процедуру:

$$\begin{array}{l} \textit{if } x \textit{ is } \tilde{A} \textit{ then } y \textit{ is } \tilde{B} \\ \quad y \textit{ is } \tilde{B}' \\ \hline \quad x \textit{ is } \tilde{A}', \end{array} \tag{1.8}$$

де відомий консеквент $\tilde{B}'$, а антецедент $\tilde{A}'$ – не відомий.

Процедура (1.8) може бути реалізована на основі узагальненого способу GMT [131] у такий спосіб:

$$\tilde{A}' = \tilde{R}(x, y) \circ \tilde{B}', \tag{1.9}$$

де $\tilde{R}(x, y)$ – відношення, отримане з *if / then* правила (1.8).

Особливістю процедур (1.7), (1.9) є те, що на цей час їхня реалізація може здійснюватися як на звичайних «чітких» обчислювальних засобах, так і з використанням нечітких спеціалізованих мікропроцесорів [133, 134].

Розглянемо можливі шляхи реалізації процедур (1.7), (1.9) і визначимо особливості обчислення функцій належності $\mu(x,y)$ для відношення $R(x,y)$. Функцію належності $\mu(x,y)$ представимо в такий спосіб:

$$\mu(x,y)=\psi\left(\mu_{\tilde{A}}(x),\mu_{\tilde{B}}(y)\right). \tag{1.10}$$

Нині відомо кілька підходів до визначення (1.10). В історичному аспекті, можливо, першим ефективним підходом до визначення функції є оператор Заде (Zadeh max-min) у вигляді [135]

$$\psi\left(\mu_{\tilde{A}}(x),\mu_{\tilde{B}}(y)\right)=\left(\mu_A(x)\wedge\mu_B(y)\right)\vee\left(1-\mu_{\tilde{A}}(x)\right). \tag{1.11}$$

Тоді, з урахуванням (1.11) відповідне значення (1.10) з використанням оператора Заде має вигляд

$$\mu(x,y)=\left(\mu_{\tilde{A}}(x)\wedge\mu_{\tilde{B}}(y)\right)\vee\left(1-\mu_{\tilde{A}}(x)\right). \tag{1.12}$$

Вираз (1.12) орієнтовано на ослаблення впливу функції належності антецедента $\mu_{\tilde{A}}(x)$ за її відносно великих значень на користь консеквента і посилення її впливу на керування за малих значень.

В ході вирішення практичних задач нечіткого керування в [136] запропонований оператор Мамдані (Mamdani min)

$$\psi\left(\mu_{\tilde{A}}(x),\mu_{\tilde{B}}(y)\right)=\mu_{\tilde{A}}(x)\wedge\mu_{\tilde{B}}(y). \tag{1.13}$$

Тоді, з урахуванням (1.13), відповідне значення (1.10) з використанням оператора Мамдані має вигляд

$$\mu(x,y)=\mu_{\tilde{A}}(x)\wedge\mu_{\tilde{B}}(y). \tag{1.14}$$

Функція (1.14) багато в чому орієнтована на песимістичні сценарії розвитку процесів керування тому, що підсилюється вплив меншого зі значень функцій належності антецедента і консеквента.

Модифікацією (1.11), (1.12) є арифметичний оператор (Arithmetic) [137]:

$$\psi\left(\mu_{\tilde{A}}(x),\mu_{\tilde{B}}(y)\right)=1\wedge\left(1-\mu_{\tilde{A}}(x)+\mu_{\tilde{B}}(y)\right), \tag{1.15}$$

функція належності якого з урахуванням (1.15) має вигляд:

$$\mu(x,y)=1\wedge\left(1-\mu_{\tilde{A}}(x)+\mu_{\tilde{B}}(y)\right). \tag{1.16}$$

У функції (1.16) є істотним вплив функції антецедента. Дійсно, якщо $\mu_{\tilde{A}}(x)<\mu_{\tilde{B}}(y)$, то значення функції (1.16) дорівнює одиниці. Якщо ж $\mu_{\tilde{A}}(x)>\mu_{\tilde{B}}(y)$, то значення функції (1.16) визначається значенням антецедента. Очевидно, що випадок $\mu_{\tilde{A}}(x)=\mu_{\tilde{B}}(y)$ є тривіальним, що приводить до значення (1.16), рівного одиниці.

У роботі [138] Ларсен запропонував арифметичний оператор (Larsen Product), що по суті є алгебраїчним добутком відповідних функцій і належить до основних операцій над нечіткими відношеннями [131]

$$\psi\left(\mu_{\tilde{A}}(x),\mu_{\tilde{B}}(y)\right)=\mu_{\tilde{A}}(x)\cdot\mu_{\tilde{B}}(y). \tag{1.17}$$

Тоді, з урахуванням (1.17) відповідне значення (1.9) з використанням оператора Ларсена має вигляд

$$\mu(x,y)=\mu_{\tilde{A}}(x)\cdot\mu_{\tilde{B}}(y). \tag{1.18}$$

Значення (1.18) припускає ослаблення функції стосовно значень функцій антецедента і консеквента, що може привести до істотного збільшення витрат у процедурах досягнення цілей керування.

У роботі [131] запропоновано кілька альтернативних підходів до знаходження (1.10), проте, на нашу думку, застосування цих рішень носить обмежений характер.

Булевий оператор (Boolean) заснований на класичній логіці і може використовуватися в застосуваннях, пов'язаних із прийняттям рішень, нечітким контролем і керуванням:

$$\psi\left(\mu_{\tilde{A}}(x),\mu_{\tilde{B}}(y)\right)=\left(1-\mu_{\tilde{A}}(x)\right)\vee\mu_{\tilde{B}}(y), \tag{1.19}$$

функція належності в даному випадку з урахуванням (1.19) має вигляд:

$$\mu(x,y)=\left(1-\mu_{\tilde{A}}(x)\right)\vee\mu_{\tilde{B}}(y). \tag{1.20}$$

З (1.20) випливає, що в нечіткому керуванні домінує антецедент за його досить малих значень і досить невеликих значень функції консеквента. В іншому випадку є істотним вплив функції консеквента.

Обмежений оператор (Bounded Product) може бути застосований у задачах керування і визначається в такий спосіб:

$$\psi\left(\mu_{\tilde{A}}(x),\mu_{\tilde{B}}(y)\right)=0\vee\left(\mu_{\tilde{A}}(x)+\mu_{\tilde{B}}(y)-1\right), \tag{1.21}$$

функція належності з урахуванням (1.20) має вигляд:

$$\mu(x,y)=0\vee\left(\mu_{\tilde{A}}(x)+\mu_{\tilde{B}}(y)-1\right). \tag{1.22}$$

З (1.22) випливає, що за малих значень функцій антецедента і консеквенту $\mu_{\tilde{A}}(x)+\mu_{\tilde{B}}(y)<1$ функція (1.22) дорівнює нулю. Зі збільшенням значень функцій і $\mu_{\tilde{A}}(x)+\mu_{\tilde{B}}(y)>1$ нечітке керування залежить від складових, які є рівноправними.

Так званий стандартний оператор (Standard Sequence) характеризується тією особливістю, що приймає одне з двох значень $\{0,1\}$ і визначається у такий спосіб:

$$\psi\left(\mu_{\tilde{A}}(x),\mu_{\tilde{B}}(y)\right)=\begin{cases}1, if\ \ \mu_{\tilde{A}}(x)\le\mu_{\tilde{B}}(y),\\ 0, if\ \ \mu_{\tilde{A}}(x)\ge\mu_{\tilde{B}}(y),\end{cases} \tag{1.23}$$

функція належності з урахуванням (1.23) має вигляд:

$$\mu(x,y)=\begin{cases}1, if\ \ \mu_{\tilde{A}}(x)\le\mu_{\tilde{B}}(y),\\ 0, if\ \ \mu_{\tilde{A}}(x)\ge\mu_{\tilde{B}}(y).\end{cases} \tag{1.24}$$

У цьому випадку в керуванні перевагу отримує консеквент тому, що при $\mu_{\tilde{A}}(x)\ge\mu_{\tilde{B}}(y)$ значення (1.24) дорівнює нулю.

Чіткий (упевнений) оператор (Drastic Product) характеризується тією особливістю, що він дає конкретне значення при граничних значеннях функцій і визначається в такий спосіб:

$$\psi\left(\mu_{\tilde{A}}(x), \mu_{\tilde{B}}(y)\right) = \begin{cases} \mu_{\tilde{A}}(x), if\ \mu_{\tilde{B}}(y) = 1, \\ \mu_{\tilde{B}}(y), if\ \mu_{\tilde{A}}(x) = 1, \\ 0, if\ \mu_{\tilde{A}}(x) < 1, \mu_{\tilde{B}}(y) < 1, \end{cases} \tag{1.25}$$

функція належності з урахуванням (1.25) має вигляд:

$$\mu(x,y) = \begin{cases} \mu_{\tilde{A}}(x), if\ \mu_{\tilde{B}}(y) = 1, \\ \mu_{\tilde{B}}(y), if\ \mu_{\tilde{A}}(x) = 1, \\ 0, if\ \mu_{\tilde{A}}(x) < 1, \mu_{\tilde{B}}(y) < 1. \end{cases} \tag{1.26}$$

У (1.26) враховується той факт, що якщо $\mu_{\tilde{A}}(x) < 1, \mu_{\tilde{B}}(y) < 1$, то функція приймає значення, рівне нулю, а це в практичних застосуваннях часто є неприпустимим за суттєво нечіткого простору станів об'єкта дослідження.

У застосуваннях розглядається також можливість використання оператора Ґугена (Gougen) і оператора Ґоделіана (Godelian) [131]. Їхня особливість полягає в тому, що значення функції $\psi\left(\mu_{\tilde{A}}(x), \mu_{\tilde{B}}(y)\right)$ залежить від співвідношення функції належності антецедента і консеквента. Оператор Ґугена визначений у вигляді:

$$\psi\left(\mu_{\tilde{A}}(x), \mu_{\tilde{B}}(y)\right) = \begin{cases} 1, if\ \mu_{\tilde{A}}(x) \le \mu_{\tilde{B}}(y), \\ \dfrac{\mu_{\tilde{B}(y)}}{\mu_{\tilde{A}}(x)}, if\ \mu_{\tilde{A}}(x) > \mu_{\tilde{B}}(y), \end{cases} \tag{1.27}$$

функція належності з урахуванням (1.26) має вигляд:

$$\mu(x,y) = \begin{cases} 1, if\ \mu_{\tilde{A}}(x) \le \mu_{\tilde{B}}(y), \\ \dfrac{\mu_{\tilde{B}(y)}}{\mu_{\tilde{A}}(x)}, if\ \mu_{\tilde{A}}(x) > \mu_{\tilde{B}}(y). \end{cases} \tag{1.28}$$

Аналіз (1.28) дає підставу стверджувати, що за відносно малих значень функції належності антецедента $\mu_{\tilde{A}}(x)$, нечітке відношення підсилюється і досягає одиниці. В іншому випадку нечітке відношення послаблюється й суттєво залежить від співвідношення стану антецедента і консеквента.

Близьким за фізичним змістом є оператор Ґоделіана

$$\psi\left(\mu_{\tilde{A}}(x), \mu_{\tilde{B}}(y)\right) = \begin{cases} 1, if\ \mu_{\tilde{A}}(x) \le \mu_{\tilde{B}}(y), \\ \mu_{\tilde{B}}(y), if\ \mu_{\tilde{A}}(x) > \mu_{\tilde{B}}(y), \end{cases} \tag{1.29}$$

функція належності з урахуванням (1.29) має вигляд:

$$\mu(x,y) = \begin{cases} 1, if\ \mu_{\tilde{A}}(x) \le \mu_{\tilde{B}}(y), \\ \mu_{\tilde{B}}(y), if\ \mu_{\tilde{A}}(x) > \mu_{\tilde{B}}(y). \end{cases} \tag{1.30}$$

Відмінність дії оператора (1.30) від оператора (1.28) полягає в тому, що зі збільшенням функції $\mu_{\tilde{A}}(x)$ домінуючого значення набуває функція консеквента $\mu_{\tilde{B}}(y)$.

На певному етапі досліджень у дослідників виникла також проблема відповідності операторів і функцій (1.12), (1.14), (1.15), (1.18), (1.20), (1.22), (1.24), (1.26), (1.28), (1.30) класичним способам логічних висновків ponens, tollens. У роботах [139–141] розглянуто окремі аспекти зазначеної проблеми, показано принципову можливість використання операторів у процедурах нечіткого логічного виведення.

Розглянемо далі механізми реалізації процедур прямого і зворотного нечітких логічних виведень.

Логічне виведення на основі нечітких множин і нечітких відношень у просторі станів динамічних взаємодіючих процесів. Проблемам дослідження процедур і нових підходів до розробки ефективних механізмів логічного виведення приділяється в науковій літературі достатня увага [142–145].

Як відзначено вище, у роботі [131] визначено основні процедури нечіткого логічного виведення – це нечітке GMP (1.7) і нечітке GMT (1.9) виведення. У термінах функцій належності, рівняння (1.8) можна подати у вигляді [131]:

$$\mu_{\tilde{B}'}(y) = \vee_{x}\left(\mu_{\tilde{A}'}(x) \wedge \mu(x,y)\right), \tag{1.31}$$

а рівняння (1.9) [131] –

$$\mu_{\tilde{A}'}(x) = \vee_{y}\left(\mu_{\cdot}(x,y) \wedge \mu_{\tilde{B}'}(y)\right). \tag{1.32}$$

Застосувавши деякий оператор з (1.12), (1.14), (1.16), (1.18), (1.20), (1.22), (1.24), (1.26), (1.28), (1.30), що реалізує функцію належності відношення $\mu(x,y)$, використовуючи (1.31), (1.32), ми можемо отримати шукане рішення з (1.7), (1.9) у практичних розробках.

Корисною реалізованою процедурою можна вважати знаходження нечіткого розв'язку на основі двох і більше правил [131]. Нехай ми маємо такі правила:

$$\begin{aligned} &\textit{if } x \textit{ is } \tilde{A} \textit{ then } y \textit{ is } \tilde{B} \\ &\textit{if } y \textit{ is } \tilde{B} \textit{ then } z \textit{ is } \tilde{C}. \end{aligned} \tag{1.33}$$

Тоді, відповідно до правил силогізму, з (1.33) ми можемо вивести умову

$$\textit{if } x \textit{ is } \tilde{A} \textit{ then } z \textit{ is } \tilde{C}. \tag{1.34}$$

Рішення (1.34) реалізується шляхом знаходження

$$\tilde{R}(x,z) = \tilde{R}_1(x,y) \circ \tilde{R}_2(y,z)$$

і використання описаної вище процедури.

Залежно від реалізованої процедури і використовуваних операторів побудови відношень $\tilde{R}(x,y)$ у структурі (1.5) доцільно розглянути інтерпретацію зв'язки *ELSE* [21]. Нехай відношення $\tilde{R}(x,y)$, визначене з використанням

деякої функції ψ , тоді інтерпретація зв'язки $ELSE$ залежно від використовуваного оператора може бути представлена в такий спосіб: Zadeh max-min – and; Mamdani min – or; Arithmetic – and; Larsen Product – or; Boolean – and; Bounded Product – or; Drastic Product – or; Standard Sequence – and; Gougen – and; Godelian – and.

Слід зазначити, що зв'язка $ELSE$ може інтерпретуватися також як арифметична сума. Застосування пропонованих інтерпретацій зв'язки $ELSE$ дає можливість досліднику вирішувати складні задачі в процедурах нечіткого логічного виведення.

Підходи до дефаззіфікації в нечіткому логічному виведенні. Зазвичай як рішення в системі логічного виведення важливо знайти конкретне рішення з урахуванням ступеня виконання правила (DOF). Проблема в цьому випадку полягає в перетворенні нечіткої підмножини (рішення) у скаляр [146].

Питанням побудови уточнюючих методів дефаззіфікації приділяється належна увага й у ряді інших робіт [131, 147]. Найбільш розповсюдженими на практиці є такі уточнюючі методи: метод пошуку центра області (COA); метод пошуку центра сум (COS); метод пошуку центра максимумів (MOM). Наведений порівняльний аналіз методів дає підставу стверджувати, що метод COA є досить розповсюдженим, але він тяжіє до центральних областей функції, а це часто приводить до збільшення часу виведення. Метод COS досить ефективний, дозволяє отримувати швидкі результати, але через те, що в ньому не враховується перекриття компонентів, які утворюють результуючу функцію, допускає в ряді випадків значні погрішності. Метод MOM є простим і ефективним засобом дефаззіфікації більш швидким, ніж COA і заснованим на пошуку центра абсолютного максимуму. Його недолік полягає в тому, що він тяжіє до компонентів функції з найбільшим максимумом і не враховує інші складові функції. Часто в модифікаціях зазначених методів додатково вводиться мінімально припустимий рівень значення функції (DOF^*), що усуває елементи з незначними ступенями належності $DOF < DOF^*$ і тим самим прискорює отримання шуканих результатів.

1.10 Аналіз нечітких моделей динамічних взаємодіючих процесів

Особливості нечітких моделей і нечіткого моделювання. Визначимо деякі вимоги до нечітких моделей. Унікальною особливістю таких моделей є те, що вони мають забезпечувати гнучку стратегію обробки різнорідних динамічних взаємодіючих процесів, які представляють дані і знання в досить нечіткому просторі станів об'єктів аналізу. Динамічні взаємодіючі процеси описуються як числовими, так і лінгвістичними змінними.

У зв'язку з цим можна стверджувати, що нечіткі моделі орієнтовані на моделювання конструкцій, для яких характерно [148]:

– функціонування на рівні лінгвістичних термів (нечітких множин);

– характеристики системи можуть бути зображені в такому самому лінгвістичному форматі;

– представлення й обробка даних в умовах невизначеності.

Згідно з [148] деяка система може бути представлена набором нечітких моделей залежно від ступеня деталізації нечітких множин, що використовуються для представлення спеціалізованої моделі із середовищем моделювання. Ступінь деталізації лінгвістичних представлень (міток) визначає об'єкт досліджень, а логіка лінгвістичного представлення процесів описується рівнем логічно-орієнтованих відношень.

Логічність запропонованої структури очевидна, проте вона не реалізує окремі моделі представлення середовища на різних ієрархічних рівнях з єдиною методологічною базою і з єдиним математичним апаратом, що не завжди розумно й істотно ускладнює обчислювальні процедури та інтерпретацію отриманих результатів.

Розглянемо деяку узагальнену структуру нечітких моделей. Нечітка модель може бути представлена на основі цілеспрямованої взаємодії трьох основних модулів: нечіткого кодера, модуля обробки, нечіткого декодера. Інструментом формування інтерфейсу між середовищем моделювання і власне обчислювальним модулем моделі є нечіткі множини [149, 150].

Модуль обробки може істотно змінюватися залежно від специфічної проблеми й особливостей предметної області. Зокрема, з погляду автора роботи [151], перспективним є відображення правил у формі нечітких нейронних мереж. У цій роботі також обґрунтовується структура моделювання предметних областей, що ґрунтується на нечітких даних і знаннях. Справедливість пропонованої структури очевидна, проте багато в чому не враховує вимог перевірки забезпечення адекватності даних і знань предметної області.

Розглянемо деякі особливості нечіткого кодування і декодування. Істотна роль нечіткого кодера і нечіткого декодера полягає в тому, що вони мають кодувати/декодувати інформацію, що виходить з середовища або спрямовану до середовища, в якому відбувається моделювання. Інформація може бути різнорідною за характером, включаючи точні числові дані, інтервали чітких і нечітких даних і знань, а також лінгвістичні змінні. Перетворення цієї зовнішньої форми інформації у внутрішній формат, що використовується в нечіткій моделі, реалізованій через різні процедури відповідності, є важливою задачею. Часто ці процедури залежать від можливості надбання даних і знань, а також потреб користувача. Такі перетворення зазвичай здійснюються за допомогою функцій належності і процедур дефаззіфікації [131]. У літературі [124, 116, 152, 153, 154, 148] пропонується опис і деякий аналіз існуючих нечітких моделей. Розглянемо деякі основні класи нечітких моделей, серед яких слід виділити моделі на основі табличних представлень, нечітких граматик, нечітких реляційних рівнянь, правил продукцій, локальних моделей регресії, штучних нейронних мереж і нечітких нейронних мереж, нечітких мереж Петрі.

Табличне представлення нечіткої моделі. Цей клас моделей найменш структурований, оскільки фіксує базисні зв'язки між лінгвістичними змінними системи. Наприклад, дискретно-часова динамічна нечітка модель з однією змінною керування (u) і фазовою змінною (x) може бути просто подана в табличній формі (рис. 1.9).

На рис. 1.9 – A1, A2, A3 і B1, B2, B3, є деякими лінгвістичними представленнями, пов'язаними з відповідними змінними. Ця таблиця може бути перетворена на ряд правил (умовних тверджень) вигляду

$$if\ u(k)\ is\ \tilde{A}_i\ and\ x(k)\ is\ \tilde{B}_j\ then\ x(k+1)\ is\ \tilde{B}_l,\quad i,\ j,\ l=1,2,3.$$

На відміну від систем, заснованих на правилах, у даних реалізаціях не розглядаються схеми логічного виведення. Фактично, таблична форма нечіткої моделі найбільш прозора серед усіх типів нечітких моделей, проте її операційні можливості дуже обмежені. Очевидно, що таблиці можуть також включати лінгвістичні змінні.

u(k) \ x(k)	B1	B2	B3
A1	B3	B1	B3
A2	B2	B2	B1
A3	B2	B1	B1
			x(k+1)

Рис. 1.9. Нечітка модель з однією змінною

Нечіткі реляційні рівняння. Реляційні рівняння на основі нечіткої логіки були запропоновані на початку 80-х років минулого сторіччя і досить досліджені як на теоретичному, так і на прикладному рівні [155, 156].

На рис. 1.10 запропоновано класифікацію нечітких реляційних рівнянь. Нечіткі реляційні моделі виражають залежності між змінними системи в термінах нечітких відношень і є більш ефективними порівняно з функціонально-орієнтованими моделями.

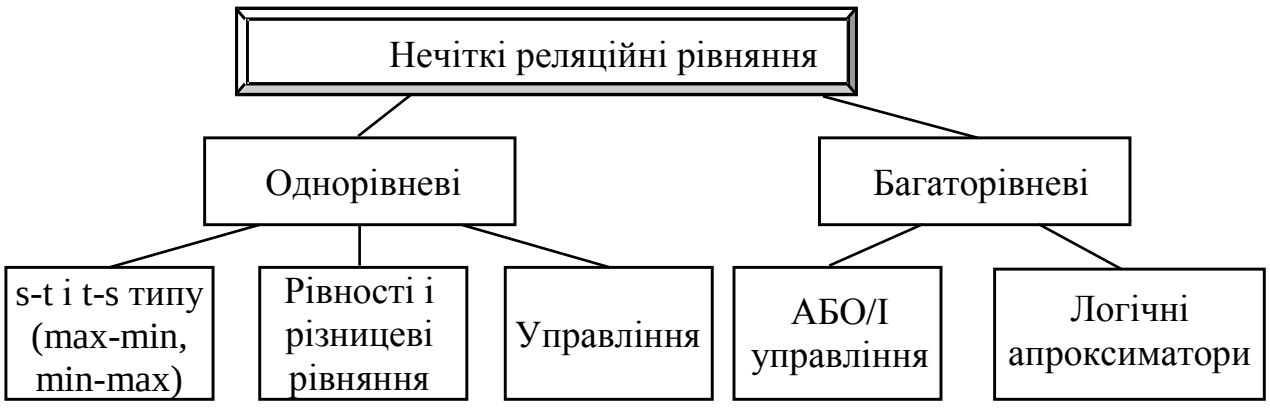


Рис. 1.10. Класифікація нечітких реляційних рівнянь

Наприклад, нечітке реляційне рівняння, яке включає стандартні max–min оператори, може бути представлене виразом

$$y_j = \bigvee_{i=1}^{n} \left(x_i \wedge r_{ij} \right), \tag{1.35}$$

де $j = 1,2,...,m,\quad \tilde{R}(x,y) = \left[r_{ij} \right]$ – відношення, визначене на декартовому добутку $[0,1]^n \times [0,1]^m$.

Взагалі (1.35) – це реляційна модель системи з одним входом і одним виходом. Очевидно, що ця модель не дає змоги здійснювати аналіз адекватності й оптимізацію представлення даних і знань в процесі їхньої взаємодії.

Нечіткі граматики. Нечіткі граматики і нечіткі мови [157] – суть нечіткі символьно-орієнтовані формалізми, що можуть досить просто використовуватися в описі різних систем. Особливий інтерес вони представляють у задачах аналізу часових рядів і розробки класифікаторів сигналу. У формальному представленні нечітка граматика заснована на класі породжуючих формальних граматик, які засновані на загальноприйнятій теорії Хомського [154], і визначена четвіркою

$$G = (V_N, V_T, P, \sigma), \tag{1.36}$$

де V_T – множина термінальних символів (алфавіт);

V_N – множина нетермінальних символів, причому $V_N \cap V_T = \varnothing$;

P – список продукційних правил;

σ – початковий символ.

Елементи P у (1.36) мають форму $a \xrightarrow{\beta} b$, де a, b – два рядки, що належать до $V_T \cup V_N$, а β визначає важливість правила.

У практичних реалізаціях нечіткі формальні граматики є громіздкими, їхнє застосування для цілей аналізу властивостей нечітких систем великої розмірності ускладнене.

Нечіткі моделі, що базуються на правилах, засновані на правилах обчислення з нечіткими множинами є ефективним засобом представлення взаємодіючих динамічних процесів, що відображають дані і знання (1.5) у нечіткому просторі станів.

У загальному вигляді продукція – це вирази такого вигляду: $(i); Q; P; A \to B; N$, де i – ім'я продукції. Як ім'я може виступати також порядковий номер. Елемент Q характеризує сферу застосування продукцій. Важливою складовою продукцій є їхнє ядро – $A \to B$, яке в загальному випадку містить $if/then$ структури типу: $if\ A\ then\ B\ ELSE$. У реальних конструкціях ядра складова A характеризується складною структурою, що включає також деякі предикати, логічні оператори типу and, or, not та їхні похідні. Компонента P визначає умови застосовності ядра продукції, що часто

представлено деяким предикатом. Компонента N визначає постумови продукції і зазвичай актуалізується під час виконання правила.

У процедурах керування логічним виведенням часто використовують такі стратегії:

1. Принцип «купки книг» заснований на тому, що найбільш часто використовувані продукції є найбільш важливими і корисними. Продукції утворюють «купку», на самому верху якої знаходиться найбільш часто використовувана продукція. Цей принцип застосовують, якщо правила незалежні одне від одного або застосована така організація їхньої обробки, що вона не залежить від їхнього перебування. Підхід набув практичного застосування в деяких виробничих системах на основі робототехнічних комплексів.

2. Принцип «найбільш довгої умови». У цьому випадку з готових правил вибираємо те, у якого найбільш довга умова (антецедент) виконання ядра. Це обумовлено гіпотезою, що правила, які визначають рішення у вузьких класах ситуацій, є найбільш важливими, а отже, і невідкладними. Вони, у свою чергу, враховують більшу кількість інформації стосовно більш загальних ситуацій. Очевидно, що підхід застосовують у випадку, якщо правила добре структуровані і прив’язані до типових ситуацій, що на практиці викликає певні труднощі.

3. Принцип «шкільної дошки», заснований на ідеях так званих спускових функцій. У цьому випадку в системі виділяється область пам’яті, що є деяким аналогом шкільної дошки, на якій можуть писати і за необхідності усувати деякі дані. У цій області паралельно виконувані процеси знаходять інформацію, що визначає умови їхнього запуску і застосовності ядра правил. На «дошці» можуть фіксуватися окремі фрагменти бази знань. З принципом «шкільної дошки» може бути також задіяна і навіть сполучена процедура метаправил для перевірки деяких умов.

4. Принцип «метапродукцій», заснований на ідеї введення в систему спеціальних метаправил, задачею яких є спеціальна організація і керування продукціями в умовах їхнього неоднозначного вибору і запуску на виконання. Природно, що база знань у цьому випадку також повинна бути добре структурованою.

5. Принцип «пріоритетного вибору». Цей принцип пов’язаний із процедурами організації статичних і динамічних пріоритетів. Статичні пріоритети формуються в процесі створення бази знань на основі даних і знань про відносну важливість виконання окремих правил або групи правил. Динамічні пріоритети формуються в процесі їхнього виконання за деякими заздалегідь визначеними критеріями. Найпростішим з них може бути максимально припустимий час очікування і запуску окремих правил на виконання.

6. Принцип «керування по іменах», заснований на процедурах, близьких до визначення динамічних пріоритетів. У цьому випадку для імен продукцій пропонуються деякі граматики або процедури, що визначають підмножину правил, а найчастіше одне з правил, яке запускається на виконання. Очевидно, що даний підхід громіздкий і вимагає гарної структурованості правил, а це викликає труднощі в ході розвитку бази знань на основі введення деякої підмножини нових правил.

Сьогодні існує велика кількість літератури з різних концептуальних і прикладних аспектів систем, заснованих на правилах, їх функціонування та можливої оптимізації. Більшість з них має справу з процесами набуття й обробки знань. У зв'язку з цим нас можуть цікавити такі фундаментальні аспекти:

– набуття знань;

– процедури аналізу властивостей адекватності набутих знань;

– розробка ефективних механізмів логічного виведення, які забезпечують обчислення і керування з нечіткими правилами.

Проблеми набуття знань і розробки ефективних механізмів логічного виведення загалом вже розроблено і вирішено [131, 158–162], а їхні результати можуть бути використані в теоретичних і практичних розробках. Питання ж розробки ефективних формалізованих процедур аналізу властивостей адекватності набутих знань, представлених на основі нечітких правил продукції, як правило, не досліджені і очікують на подальше дослідження.

Локальні моделі регресії. Моделі цього класу, представлені в [163–165] і застосовувані в багатьох програмних застосуваннях, зазвичай спеціалізуються на представленні правил продукції з тією відмінністю, що висновок правила є деякою функцією

$$y = f_i(x; a_i)$$

типу

$$if\ x_1\ is\ \tilde{A}_1\ and\ x_2\ is\ \tilde{A}_2\ and ...and\ x_n\ is\ \tilde{A}_n\ then\ y = f_i(x; a_i), \qquad (1.37)$$

де $\tilde{A}_i$ – нечіткі множини у просторах вводу, а $f_i : R_n \to R$ – n-змінні функції з векторами параметрів a_i, $i = 1, 2, ..., c$.

Проблема організації зв'язку між локальними моделями (1.37) здійснюється відповідно до схеми на рис. 1.11.

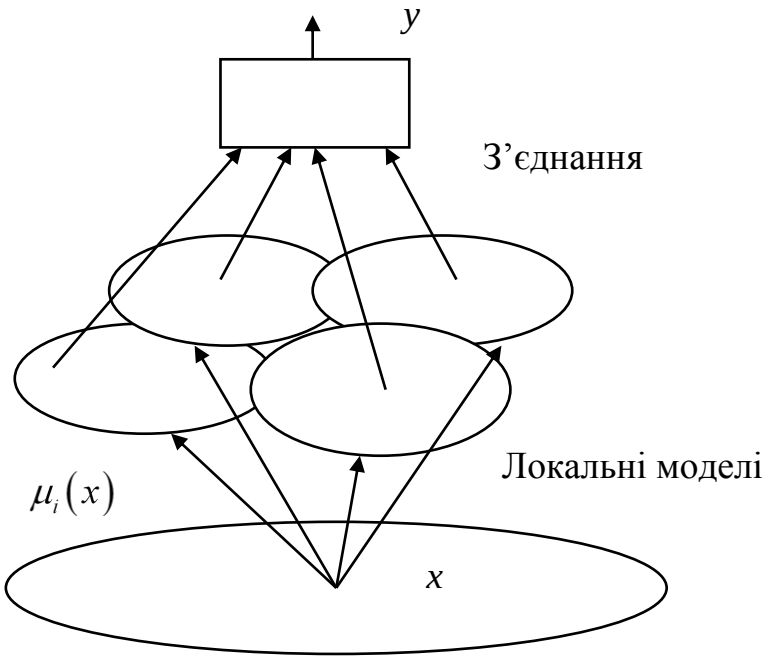


Рис. 1.11. Організація зв'язків між локальними моделями

Локальні моделі формують відмінні від нуля антецеденти правил більш високого рівня.

Локальні моделі регресії або моделі Takagi – Sugeno – Kang (TSK), які у літературі часто визначені як моделі Сугено, значною мірою застосовні в задачах керування динамічними об'єктами [166]. У задачах дослідження адекватності представлення даних і знань застосовувати їх досить важко.

Штучні нейронні мережі і нечіткі нейронні мережі. Нині штучні нейронні мережі (ШНМ) і нейро-фаззі мережі (НФМ) набули великого поширення як апарат моделювання й аналізу складних процесів, а також їхньої взаємодії і динаміки розвитку. Слід зазначити, що зараз успішно розвивається як теорія, так і її застосування, включаючи нейроконтролери і нейрокомп'ютери [167–172].

ШНМ, що є універсальними апроксиматорами, мають унікальні можливості в процесі дослідження складних нелінійних процесів у системах великої розмірності. Проте, вимога щодо здатності навчатися в реальному часі і складність реалізації часто утруднює ефективне їхнє самостійне використання в системах малої розмірності, особливо якщо процеси та їхні взаємодії можуть представлятися на множинах відношень «умова–дія». Нижче ми зупинимося на деяких особливостях нейромережних структур і їхньому розвитку.

Мережі Петрі та нечіткі мережі Петрі. У сучасних дослідженнях нечітких взаємодіючих динамічних процесів, представлених на множинах відношень «умова–дія» і тих, що характеризуються складною паралельно-послідовною взаємодією, важливе місце займають мережі Петрі. У моделюванні й аналізі процесів, представлених у нечіткому просторі станів предметної області, важливе місце займає апарат нечітких мереж Петрі [173–186]. У зв'язку з цим у наукових виданнях приділяється досить серйозна увага розробкам і дослідженню апарата нечітких мереж Петрі та його застосувань. Щорічно у світі проходить ряд наукових конференцій, симпозіумів, семінарів, на яких розглядаються нові досягнення в області нечітких мереж Петрі та їхніх застосувань. До них передусім слід віднести роботи щорічних Міжнародних конференцій з Прикладних програм і теорії мереж Петрі (International Conference on Applications and Theory of Petri Nets).

У ряді країн проводяться дослідження нечітких мереж Петрі. Зокрема, в університеті Мельбурна в рамках відділу електричної та електронної техніки проводиться семінар з досліджень на тему «Синтез нечітких мереж Петрі» [187]. У нашій країні останнім часом також приділяється істотна увага дослідженням в області мереж Петрі та їхніх застосувань на основі сучасних інформаційних технологій.

Нечіткі мережі Петрі зручним способом описують як структуру взаємодіючих процесів, так і динаміку їхнього розвитку. Важливою перевагою існуючого апарата можна вважати його добру здатність до формалізації, інтерпретованість, можливість модифікації моделей і процесів, детального

дослідження простору станів реальних об'єктів з використанням структури і простору станів моделі. Разом з тим слід зазначити, що існуючі модифікації апарата нечітких мереж Петрі часто не дають можливості враховувати множину параметрів предметної області (час, надійність, складність, вартість, точність і т.п.), що можуть бути задані аналітичними і (або) логічними залежностями, предикатами. Важливим є також те, що зі збільшенням розмірності розв'язуваних задач на множинах процесів «умова–дія», а також простору станів моделі її ефективність зменшується. Це вимагає додаткових досліджень зі створення ефективних нечітких моделей. Часто ця задача розв'язується на основі інтеграції нечітких мереж Петрі і ШНМ, НФМ у складі гібридних моделей.

Деякі особливості адекватного представлення процесів нечіткими моделями. Розглянемо деякі підходи до перевірки ефективності моделі і підтвердження їхньої коректності. Внаслідок нечислового характеру інформації, оброблюваної системою, можна запропонувати такі механізми перевірки і підтвердження правильності моделі:

– використання навчальних даних: ефективність моделі визначається кількісно, розглядаються ті самі дані, що використовуються в реальній системі;

– використання перевіркових даних: якість моделі оцінюється шляхом використання даних, що відрізняються від використовуваних спочатку в розробці моделі.

Специфічною для нечітких моделей є перевірка, що визначається рівнем, на якому дії перевірки виконані.

Зовнішня стосовно моделі перевірка оцінює ефективність усієї моделі. Внутрішня стосовно моделі перевірка більшою мірою оцінює ефективність модуля обробки та меншою мерою модуля дефаззіфікації (нечіткого декодера).

Перевірка, що виконується за схемою внутрішньої перевірки, часто є оптимістичною. В ідеалі $Q_1 = Q_2$, де Q_1 – результат зовнішньої перевірки, Q_2 – результат внутрішньої перевірки.

Важливим є також забезпечення оптимізації моделі за заданими критеріями на множинах обмежень, що може бути реалізоване шляхом цілеспрямованого впливу на модуль обробки.

Як критерії можна прийняти такі: логічна адекватність моделі; повноцінність моделі з вирішенням комплексу задач (проблем), на які вона орієнтована; здатність моделі обробляти інформацію з різним ступенем деталізації [148].

У роботі [188] критерії ефективності й адекватності моделей пропонується доповнити вимогами коректного відображення та інтерпретації нечітких процесів, статичної й динамічної стабільності і чутливості. Ефективність досліджень автори підтверджують моделюванням реальної електромеханічної системи.

Таким чином, наведений короткий огляд по суті питання дає підставу стверджувати, що прийнятними нечіткими моделями, що можуть бути платформою комплексного вирішення питань аналізу, локалізації й

усунення логічної неадекватності у взаємодіючих динамічних процесах, представлених на множинах відношень «умова-дія», подаються нечіткі мережі Петрі [173, 185], нейро-фаззі мережні структури [131] і їхня інтеграція у складі нечітких моделей.

1.11 Проблеми подання й аналізу динамічних взаємодіючих процесів, що функціонують за умов невизначеності простору станів об'єктів керування й обробки даних і знань

Широкий клас динамічних взаємодіючих процесів $\left\{\tilde{\Pi}_i\right\}, i \in I$ у системах обчислювального інтелекту складних технологічних комплексів, що функціонують в умовах невизначеності простору станів об'єкта досліджень і характеризуються складною паралельно-послідовною взаємодією функціонально і територіально розподілених об'єктів, може бути представлений на множині і відношень «умова-дія» [189].

У зв'язку з цим виникає проблема забезпечення коректної адекватної взаємодії процесів. Зокрема у роботі [124] висвітлено деякі проблеми, успішне вирішення яких може істотно впливати на ефективність розробок і функціонування автоматизованих систем керування. До них передусім слід віднести: узгодженість чи несуперечність даних, повноту даних, відсутність надмірності. При цьому розглядаються як зовнішні, так і внутрішні компоненти проблем. Згідно з [124], на цей час відсутні теоретично точно обґрунтовані методи вирішення зазначених задач навіть у чіткому середовищі взаємодії процесів. Для вирішення питань аналізу несуперечності, повноти, відсутності надмірності в [124] розвивається і поглиблюється представлення про зазначені сторони неадекватного і некоректного представлення процесів, але не пропонуються формальні ефективні підходи до вирішення задач аналізу.

Враховуючи також, що [124] інтерпретує нечіткі процеси, принаймні, на рівні функцій належності ймовірнісними характеристиками, пропоновані узагальнення можуть використовуватися або розвиватися зі значними обмеженнями. Застосування і розвиток цих рішень важкі через відсутність у них формалізованих підходів, оскільки вони орієнтовані на вузькоспеціальні задачі. Це викликає необхідність заходження принципово нових формальних підходів і відповідного інструментарію.

Аналіз таких предметних областей, як об'єкти й інтелектуальні системи виробничого та організаційно-технічного призначення, системи та об'єкти екологічного моніторингу, системи та об'єкти, що функціонують в екстремальних умовах, показав, що доцільно, за критерієм складності, виділити задачі [189], якість рішень яких визначає ефективність функціонування інтелектуальних систем і технологічних комплексів у цілому. Слід також врахувати, що об'єкти аналізу зазвичай включають як традиційні, так і інтелектуальні компоненти з використанням великих баз знань і машин нечіткого логічного виведення.

До **першої групи** за критерієм складності насамперед слід віднести задачі, пов'язані з моделюванням і спільним аналізом структури і простору станів процесів прийняття рішень і керування:

– аналіз і виявлення властивостей досяжності цілей прийнятих рішень $\{Ds_j\}, j \in J$ при взаємодії процесів у нечіткому середовищі функціонування об'єктів аналізу;

– аналіз, виявлення і локалізація конфліктних ситуацій $\{Conf_k\}, k \in K$ в ході взаємодії процесів у нечіткому середовищі функціонування об'єктів аналізу;

– пошук та оптимізація альтернативних рішень і шляхів розвитку процесів $\{B_r\}, r \in R$ за критеріями чіткості, надійності, часовими і вартісними параметрами на заданих обмеженнях;

– аналіз, виявлення, локалізація надмірності процесів $\{Is_l\}, l \in L$ у нечіткому середовищі функціонування об'єктів аналізу;

– аналіз, виявлення, локалізація нераціональних зациклень процесів $\{Z_m\}, m \in M$ у нечіткому середовищі функціонування об'єктів аналізу.

До **другої** за критерієм складності групи можуть бути віднесені підзадачі, пов'язані з моделюванням і аналізом процесів прийняття рішень і керування у просторі станів в умовах невизначеності:

– аналіз повноти $\{Pl_n\}, n \in N$ вихідних даних, реалізованих процесів і цілей прийнятих рішень на їхній основі;

– аналіз суперечливості $\{Npt_s\}, s \in S$ вихідних даних, реалізованих процесів і цілей прийнятих рішень на їхній основі.

До **третьої групи** за критерієм складності слід насамперед віднести комплексне вирішення задач, орієнтованих на:

– моделювання і спільний аналіз структури і простору станів процесів прийняття рішень і керування;

– керування інформаційними ресурсами в умовах невизначеності;

– прогнозування розвитку процесів в умовах невизначеності.

Таким чином, стає очевидним, що зазначені проблеми можуть бути задовільно вирішені на основі цілеспрямованого застосування гібридних нейро-фаззі моделей, мультиагентних підходів і технологій.

Вирішенню і детальному вивченню деяких із роглянутих задач та їх застосуванню в мультиагентних технологіях систем штучного інтелекту, інформаційних та геоінформаційних системах і присвячено дане видання.

2 ФОРМУВАННЯ ТЕОРЕТИЧНИХ ОСНОВ ІНТЕЛЕКТУАЛЬНОГО АНАЛІЗУ АЕРОДИНАМІЧНИХ ПОКАЗНИКІВ ВЕРХНІХ ДИХАЛЬНИХ ШЛЯХІВ

2.1 Аналіз основних методів функціональної діагностики носового дихання

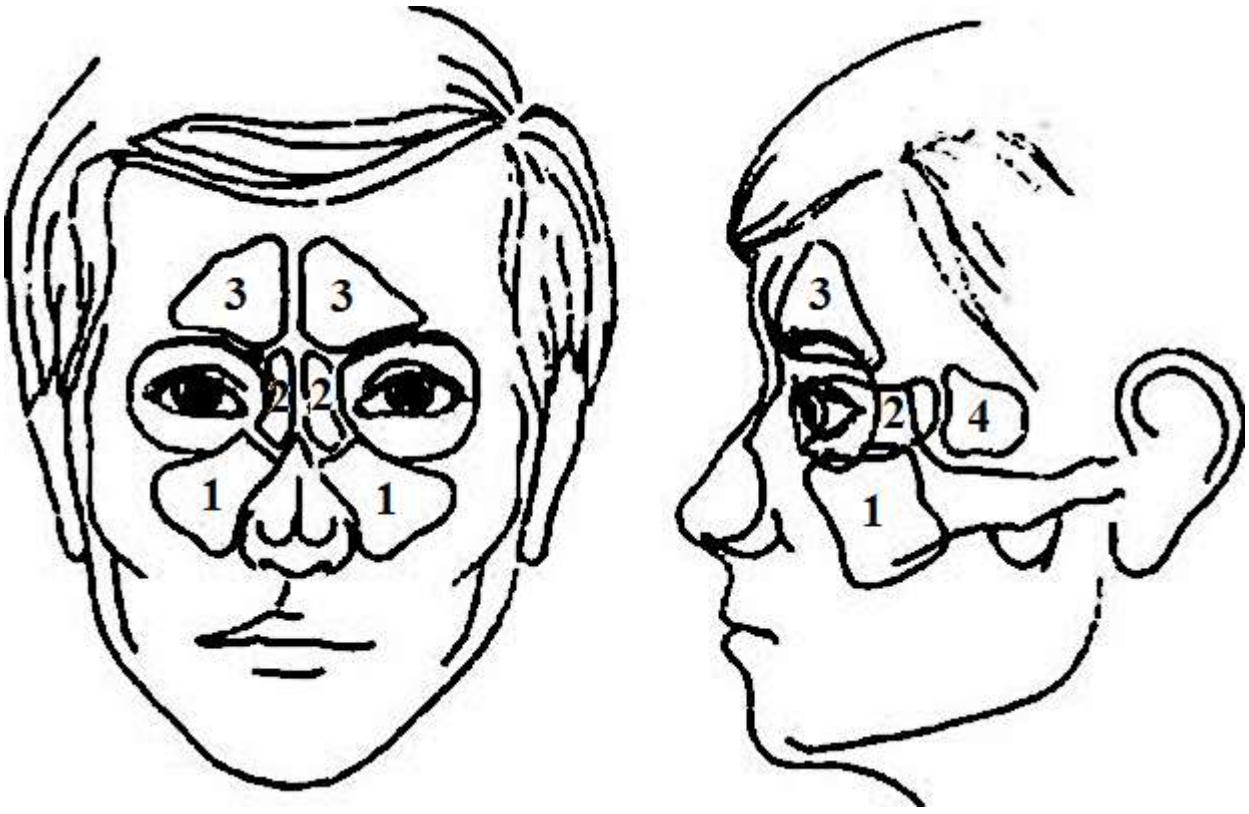
Верхні дихальні шляхи розглядаються як система, що складається з порожнини носа, носоглотки і ротоглотки, та частково ротової порожнини, яка може бути використана для дихання [190–193]. Далі в роботі, як прийнято в ринології [190], під верхніми дихальними шляхами розуміють порожнину носа, якщо це не обумовлено додатково.

Порожнина носа є повітряним каналом змінного перетину, що оточений кістковими структурами лицьового і мозкового відділів черепа і сполучається спереду через носові отвори (ніздрі) із зовнішнім середовищем і ззаду (через хоани) з носоглоткою. Порожнина носа (рис. 2.1) поділяється носовою перегородкою на дві (в загальному випадку, нерівні за розмірами і конфігурацією) частини, звані носовими лівим і правим проходами, відповідно (викривлення носової перегородки різного ступеня спостерігається більш ніж у 95% випадків [190]). На латеральній стінці кожного носового проходу розташовані кісткові утворення – носові раковини: нижня, середня та верхня. Наявність раковин збільшує площу поверхні носової порожнини, що сприяє зігріванню вдихуваного повітря. У поглибленнях під відповідними носовими раковинами знаходяться умовно виділені (неізольовані) нижній, середній і верхній носові ходи, а також загальний носовий хід, розташований між перегородкою носа і медіальною поверхнею носових раковин. У безпосередній близькості від входу розташований носовий клапан – найвужче місце носової порожнини, який у нормі має бути рухливим і обмежувати потік повітря, що надходить [190, 190–193]. У порожнину носа відкриваються вивідні отвори (співустя) придаткових пазух носа: верхньощелепних 1 (гайморових); ґратчастої кістки (2), лобних (3), клиновидних (4).

Порожнина носа поділяється на дихальну область, розташовану на рівні нижнього і середнього носових ходів та нюхову область – на рівні верхнього носового ходу, в якій розташовані закінчення нюхового нерва. Слизова оболонка дихальної області порожнини носа вкрита миготливим (війчастим) епітелієм, який виконує транспортну функцію видалення частинок, що осідають. Також у слизовій оболонці містяться залози, що виробляють секрет, який сприяє зволоженню повітря, осіданню і переміщенню частинок пилу. Велика кількість венозних судин, що утворюють на нижній і частково середній носових раковинах густі сплетіння, подібні за функцією із запалими тілами, сприяє зігріванню повітря, що надходить шляхом регулювання розмірів носових проходів.

Нормальним, з точки зору фізіології, зовнішнім диханням є носове дихання [190–193]. При цьому вводиться поняття аеродинамічного носового опору – опору внутрішньоносових структур повітряному струменю під час

проходження його через порожнину носа. Вважається, що людиною в нормі носовий опір не відчувається. Проте з його підвищенням відчувається недолік кисню і відбувається перехід на дихання ротом, яке, за сучасними уявленнями, є нефізіологічним і може призводити до кисневого голодування головного мозку через зниження інтенсивності газообміну в слизовій оболонці носової порожнини.



а) б)

Рис. 2.1. Схематична ілюстрація придаткових пазух носа:
а – фронтальний вигляд; б – сагітальний вигляд

Повітряний потік, що проходить через обидва носових проходи – асиметричний. При цьому у більшості здорових людей відзначається циклічна зміна носового опору повітряному потоку, що проходить через ліву і праву половини носа, за постійного сумарного носового опору. Ця періодична диференціальна зміна носового опору (з періодом порядку декількох годин) називається носовим циклом, який може бути викликаний необхідністю відновлення слизової оболонки носової порожнини від функціональних перевантажень і мікротравм.

Також надзвичайно важливим є повітрообмін між порожниною носа і прилеглими пазухами, здійснюваний через співустя – з'єднальні отвори. Блокування співустій внаслідок різних причин веде до припинення повітрообміну, порушення транспортування слизу (мукоциліарного кліренсу) і запалення слизової оболонки придаткових пазух.

В ході оцінювання дихальної функції носової порожнини сьогодні в більшості випадків присутній описовий підхід.

Наведений вище опис є коротким якісним уявленням про анатомію і фізіологію носової порожнини і заснований на описовому підході, що застосовується сьогодні в ринології, який в окремих роботах іноді доповнюється численними значеннями деяких фізичних показників [194–207], досить часто некорельованих між собою через застосовувані різні нестандартизовані методи і засоби вимірювань. Тому, актуальними є завдання, що розглядаються нижче, спрямовані на розробку уявлення фізичних процесів у верхніх дихальних шляхах на основі математичних моделей, які в сукупності можуть бути теоретичною основою для розробки об'єктивних методів і засобів функціональної діагностики в ринології.

Носове дихання є основним фізіологічним типом дихання, порушення якого, насамперед, призводить до зниження якості життя, а в подальшому може призводити до тяжких наслідків [190]. Тому, в ринології традиційно намагалися оцінити аеродинамічний носовий опір як величину, що залежить від втрат тиску та витрати повітря, або зворотну величину – повітряну провідність носа. До найпростіших способів визначення прохідності носа належать: проба за Воячеком, заснована на дослідженні відхилення потоком повітря пушинки, що піднесена до ніздрів пацієнта під час видиху, тест Коля [190], який дозволяє за величиною поліпшення носового дихання під час проведення відповідних маніпуляцій у порожнині носа визначити функціональні порушення в передньому відділі носового клапана. Аналогічними найпростішими методами оцінки порушень носового дихання є тести, засновані на аналізі запотівання поверхні дзеркала, піднесеного до ніздрів пацієнта під час видиху, наприклад, метод Глятцеля, в якому проводиться оцінка діаметра плями конденсату на градуйованому дзеркалі [190]. Проте ці методи є якісними, мають велику похибку і не відповідають стандартам сучасної доказової медицини, які базуються на інтелектуальному аналізі вимірюваних діагностичних даних з використанням об'єктивних кількісних критеріїв.

Найбільш сучасним кількісним методом оцінки функції носового дихання є риноманометрія – метод, за яким проводяться вимірювання перепаду тиску в носовій порожнині і відповідної об'ємної витрати повітря через ніс [190, 193] з подальшим розрахунком похідних показників і формуванням діагностичних висновків. Загальноприйнятий останніми роками метод комп'ютерної риноманометрії [190, 207–215] дозволяє характеризувати ступінь порушення носового дихання шляхом визначення показника аеродинамічного носового опору у вигляді відношення перепаду тиску в носовій порожнині до величини витрат повітря в різних фазах одного дихального циклу. Усереднена за часом (за кількістю дихальних циклів) величина співвідношення пікових значень перепаду тисків до витрати повітря Q є значущим діагностичним показником аеродинамічного носового опору і вимірюється як відношення тиску в кПа, поділених на літр за секунду – [кПа/(л/с)], або, відповідно, для отримання однакових числових значень – в Паскалях, поділених на кубічний сантиметр за секунду [Па/(см^3/с)]. До складу сучасних риноманометрів входять мініатюрні

перетворювачі тиску та об’ємної витрати (або швидкості) повітряного потоку, що дозволяє за рахунок спеціалізованих програмних засобів відображати на твердому носії (або екрані монітора) графічні залежності показників назального повітряного потоку під час дихання. Останнім часом популярності набуває метод, що має назву ринорезистометрія, в якому аеродинамічний носовий опір обчислюється не за запатентованим алгоритмом і модельним уявленням, а безпосередньо в кожній точці дихального циклу. Принципи проектування таких пристроїв викладено в літературі [216–220].

2.2 Уточнення основних положень механіки дихання

Зовнішнє дихання – сукупність фізіологічних процесів, що забезпечують надходження в організм атмосферного кисню, а також видалення з організму вуглекислого газу, що утворюється в процесі метаболізму [190–193, 221]. Далі в роботі розглядатиметься тільки процес зовнішнього (легеневого) дихання. Вхідним трактом для надходження повітря в легені є верхні дихальні шляхи (порожнина носа і придаткові пазухи), які виконують важливі фізіологічні функції – дихальну, зволожувальну, зігрівальну, транспортну та фільтрувальну. Найбільш життєво важливою з них є дихальна, на відновлення якої, насамперед, спрямована функціональна ринохірургія.

Основними параметрами механіки дихання є:

– витрата повітря, що транспортується людиною під час дихання. За даними фундаментальної роботи з фізіології [221], витрата споживаного людиною повітря знаходиться в межах 80 л/хв у нетренованого, 120...150 л/хв у тренованого і 150... 00 л/хв у короткочасному режимі (або 1,33; 2...2,5 і 2,5...3,33 л/с, відповідно). Водночас згідно з відомою графічною залежністю «витрата – об’єм» [222] значення витрати істотно вище, досягаючи понад 8 л/с під час вдиху і 11 л/с під час видиху, а спірометри, що випускаються за кордоном, розраховані на максимальну витрату до 14...16 л/с (840...960 л/хв) [223, 224]. Проведені власні дослідження також показали, що навіть середні за своїми фізичними можливостями люди при інтенсивному диханні взмозі під час вдиху споживати повітря з миттєвим значенням витрати близько 6 л/с (360 л/хв);

– тиск, що розвивається м’язами легень. При інтенсивному диханні людини зростання тиску в одиницю часу досягає = 25,7 кПа/с (наприклад, при інтенсивному диханні носом підвищення тиску до 9 кПа відбувається протягом 0,35 с, що відповідає частоті приблизно в 43 вдихів за хвилину). Відзначимо, що в технічних системах, наприклад, у насосах високого тиску, цей параметр досягає 350х103 кПа/с – на чотири порядки вище;

– статичний тиск, що розвивається м’язами легень, вимірюється за допомогою моновакуумметра і за законом Паскаля [225] поширюється однаково по всій довжині повітряного тракту людини: порожнина носа – носоглотка – гортань – трахея – бронхи – легені. Проведені власні експериментальні дослідження показали (див. підрозділ 6.2), що максимальне значення надлишкового тиску становить 0,15 кгс/см або ~ 15 кПа, а

максимальне значення розрядження 0,4 кгс/см або ~ 40 кПа (ці дані отримані у людей із середніми фізичними можливостями);

– оцінка механіки дихання людини за потужністю за формулою

$$P_{\text{пн}} = \Delta p \cdot Q = \left[\frac{\text{Па} \cdot \text{м}^3}{\text{с}} = \text{Вт} = \frac{\text{кПа} \cdot 10^3 \cdot \text{л} \cdot 10^{-3}}{\text{с}} = \frac{\text{кПа} \cdot \text{л}}{\text{с}} \right], \text{Вт}, \qquad (2.1)$$

де Δp – перепад тиску повітря на вимірюваному каналі (визначається в ротовій або носовій порожнині), кПа,

Q – витрата повітря, л/с.

Причому, під час дихання носом з витратою 0,67 л/с тиск розрядження в порожнині рота (передається по трубці до перетворювача тиску) становить 0,13 кПа, а потужність – порядку 0,09 Вт;

– під час дихання ротом з витратою 0,67 л/с тиск розрядження на витратомірі (на основі сопла Вентурі) становить 0,19 кПа і потужність порядку 0,12 Вт.

Максимальну потужність повітря, що транспортується людиною при диханні ротом, згідно з вищенаведеними даними можна розрахувати, як

$$P_{\text{видих}} = p_{\text{видих}} \cdot Q_{\text{видих}} = 15 \text{ кПа} \cdot 16 \text{ л/с} = 240 \text{ Вт},$$

$$P_{\text{вдих}} = p_{\text{вдих}} \cdot Q_{\text{вдих}} = 40 \text{ кПа} \cdot 8 \text{ л/с} = 320 \text{ Вт}$$

де $p_{\text{видих}}$ = 15 кПа – максимальний тиск під час видиху;

$p_{\text{вдих}}$ = 40 кПа – максимальний тиск під час вдиху;

$Q_{\text{видих}}$ = 16 л/с – максимальна витрата під час видиху;

$Q_{\text{вдих}}$ = 8 л/с – максимальна витрата під час вдиху.

Отримані значення максимальної потужності слід віднести до так званої «установчої» потужності, яка практично ніколи не досягається, але визначається як добуток максимальних значень її складових (наприклад, у технічній документації зарубіжних виробників об'ємних гідравлічних машин зустрічається термін «corner power» – кутова потужність). Як порівняння можна показати, що максимальна механічна потужність, яка розвивається людиною, наприклад, штангістів світового класу, досягає істотно більшого значення. Так, наприклад, з підйомом штанги масою 200 кг від рівня підлоги на висоту 1 м за час 0,5 с потужність, що розвивається людиною, складе

$$P = F \cdot v = m \cdot g \cdot \frac{l}{t} = 200 \cdot 9{,}8 \frac{1}{0{,}5} = 3920 \text{ Вт} \approx 4 \text{ кВт},$$

де $F = m \cdot g$ – зусилля, що розвиваються, Н;

m – маса штанги, кг;

g – прискорення вільного падіння, м/с^2;

v – лінійна швидкість, м/с;

l = 1 м – висота штанги;

t = 0,5 с – середній час підйому штанги.

Якщо зіставити потужності, що розвиваються м'язами легень для всмоктування повітря і руками для підняття зазначеного вантажу, то це

співвідношення може скласти до 10%. Водночас, згідно з [222], під час максимальної фізичної роботи дихальні м'язи можуть споживати до 20% від загального обсягу поглиненого кисню.

Вплив опору носових проходів на витрату повітря, що пропускається, буде наочно проілюстровано експериментами, в яких показано, що практично за однаковаї потужності дихання в 30 Вт значення витрат відрізняються вдвічі (3,36 і 1,62 л/с) при обернено пропорційному співвідношенні тисків у 2,2 рази (19,7 і 8,95 кПа).

Для побудови математичних моделей аеродинаміки верхніх дихальних шляхів необхідно оцінити вплив стисливості повітря у носовій порожнині в процесі дихання. При цьому як спрощена модель носового ходу приймається цілий трубопровід з відповідним середнім перетином.

Визначення ступеня стисливості повітря проводиться на підставі рівняння нерозривності суцільного середовища [225], поданого у вигляді

$$\frac{d\rho}{dt} + \rho div\vec{u} = 0 \ , \qquad (2.2)$$

де ρ – щільність середовища (для повітря $\rho = 1{,}3$ кг/м^3);

$\vec{u}$ – вектор швидкості повітря.

Рівняння (2.2) можна подати в координатній формі (у циліндричній системі координат)

$$\frac{1}{\rho}\cdot\frac{\partial\rho}{\partial t} + \frac{1}{r}\frac{\partial\left(ru_r\right)}{\partial r} + \frac{1}{r}\frac{\partial u_\alpha}{\partial\alpha} + \frac{\partial u_z}{\partial z} = 0 \ , \qquad (2.3)$$

де r – радіус носового ходу;

u_z, u_r, u_α – аксіальна (спрямована уздовж осі носового ходу), радіальна і кутова компоненти швидкості повітряного потоку, відповідно.

Для аксіальносиметричного потоку $\partial u_\alpha/\partial\alpha = 0$ і тоді після очевидних перетворень отримуємо з формули (2.3)

$$\frac{1}{\rho}\cdot\frac{\partial\rho}{\partial t} + \frac{u_r}{r} + \frac{\partial u_r}{\partial r} + \frac{\partial u_z}{\partial z} = 0 \ . \qquad (2.4)$$

Зробимо чисельну оцінку величин $\partial u_z/\partial z$ і $\partial\rho/\partial t$. Середнє значення дивергенції аксіальної швидкості згідно з даними [226]

$$\frac{\partial u_z}{\partial z} \approx \frac{\Delta u_z}{\Delta z} \approx \frac{-1{,}5}{0{,}07} \approx -21\text{с}^{-1} \ .$$

Виконаємо оцінку похідної від щільності за часом, вважаючи повітря газом, близьким до ідеального, а процес його поширення в носовій порожнині близьким до ізотермічного, тоді з рівняння Клапейрона – Менделєєва знаходимо

$$\frac{d\rho}{dt} = \frac{1}{R_\mu \text{T}}\cdot\frac{dp}{dt} \ , \qquad (2.5)$$

де $R_\mu = R/\mu = 287\,м^2/\left(c^2\cdot K\right)$ – молярна газова стала для повітря.

Замінюючи величину похідної від тиску середнім значенням зміни тиску в процесі вдиху або видиху до його тривалості, таким чином, $\frac{dp}{dt} \approx \frac{\Delta p}{\Delta t} \approx \frac{300}{1,5}$ (враховуючи, що за даними [208 – 210] $\Delta p = 300$ Па і $\Delta t = 1,5$ с – середня тривалість вдиху), і підставляючи у формулу (2.5) відповідні значення чисельних величин при температурі 300 К, знаходимо значення похідної від щільності за часом

$$\frac{dp}{dt} = \frac{1}{287 \cdot 300} \cdot \frac{300}{1,5} = 2,3 \cdot 10^{-3} \frac{кг}{м^3 \cdot с}.$$

Після підстановки знайдених середніх значень похідних у рівняння (2.4), отримаємо

$$\frac{u_r}{r} + \frac{du_r}{dr} = \pm \frac{2,3 \cdot 10^{-3}}{1,3} + 21 \ .$$

Оскільки перший член у правій частині цього рівняння (середнє значення похідної $\partial\rho/\partial t$) мізерно малий порівнянно з іншим, який становить дивергенцію аксіальної швидкості повітря, можна вважати, що

$$div\vec{u} \approx 0,$$

і з формули (2.3) отримаємо

$$\frac{d\rho}{dt} = 0.$$

Таким чином, приймаємо, що за даних умов повітря є нестисливим газом.

Наведену вище теоретичну оцінку нестисливості повітря під час дихання можна підтвердити, виходячи з емпіричного критерію стисливості повітря, який засновано на визначенні значення числа Маха М, що дорівнює відношенню величини швидкості U повітряного потоку в носовій порожнині до швидкості поширення звуку в повітряному середовищі $V_{зв}$

$$M = \frac{U}{V_{зв}} \ .$$

При значеннях $M \le 0,3$ повітря можна вважати нестисливим середовищем [225]. З огляду на те, що діючі значення швидкості W повітряного потоку у верхніх дихальних шляхах не перевищують 50 м/с (≈0,15 зі швидкістю звуку в повітрі, прийнятою рівною близько 320 м/с), що дозволяє розглядати повітря як нестисливе середовище.

2.3 Розробка моделі одновимірної течії повітря у носовій порожнині під час дихання

Для проведення комп'ютерного планування у функціональній ринохірургії необхідно знати аеродинамічні характеристики повітряного потоку, що проходить через верхні дихальні шляхи. Базовим параметром при цьому є безрозмірне число Рейнольдса, за значенням якого визначається режим течії повітря, а також вибір відповідних моделей і припущень.

Число Рейнольдса [225] визначається як

$$\mathrm{Re} = 10^3 \frac{V \cdot d_г}{\nu_{в'язк}}\ , \tag{2.6}$$

де V – швидкість течії повітря в каналі, м/с;

$\nu_{в'язк}$ – коефіцієнт кінематичної в'язкості повітря, $мм^2/с$;

$d_г$ – гідравлічний діаметр, який визначається за формулою

$$d_г = \frac{4S}{П}\text{, мм,} \tag{2.7}$$

де S – площа перетину каналу;

$П$ – змочений периметр носового каналу, мм.

Залежно від значення числа Рейнольдса визначають режим течії повітря (ламінарний або турбулентний), знання якого необхідно для розрахунку аеродинамічних опорів у носовій порожнині і планування оперативного втручання щодо усунення критичних ділянок.

Для визначення режиму течії повітря у порожнині носа необхідно виконати побудову перетинів носового каналу, перпендикулярних руху повітря, визначити їх площі і змочені периметри, провести розрахунок числа Рейнольдса і порівняти його з критичним значенням. Розрахунок геометричних характеристик перетинів носового каналу виконувався шляхом побудови МПР СКТ-даних у фронтальній площині перпендикулярно течії повітря в носовій порожнині.

Модуль побудови МПР забезпечує відображення томографічних зрізів, довільно орієнтованих щодо площини сканування. Для носових ходів, що мають складну розгалужену структуру, цей режим візуалізації є одним з основних, оскільки дозволяє наочно відобразити їхню орієнтацію і анатомічну конфігурацію живих перетинів – перпендикулярних до напрямку течії повітря. Алгоритм побудови мультипланарної реконструкції (рис. 2.2, а) базується на завданні площині Р реконструкції за трьома точками $A\left(A_x, A_y, A_z\right)$, $B\left(B_x, B_y, B_z\right)$ і $C\left(C_x, C_y, C_z\right)$. Дві точки, як правило, задаються у площині одного томографічного зрізу, а третя визначає орієнтацію площини реконструкції у вертикальній площині. Також вказуються межі реконструкції у вигляді номерів верхнього і нижнього томографічних зрізів.

У разі побудови реконструкцій, перпендикулярних площині сканування, третя точка обирається автоматично і має x, y-координати, які збігаються з координатами однієї з двох раніше обраних точок, а z-координата визначається максимальним/мінімальним номером зрізу S_k, що використовується в реконструкції. Таким чином, площина реконструкції визначається параметрично з виразу

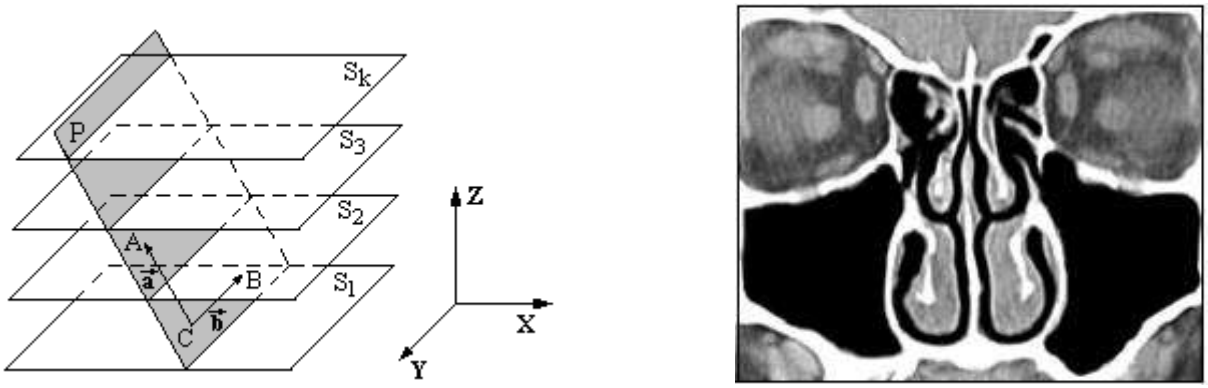
$$P(t,s) = C + \vec{a}t + \vec{b}s = C + \overrightarrow{(A-C)}t + \overrightarrow{(B-C)}s\,, \tag{2.8}$$

де t і s – параметри, причому $t; s \in [0,1]$.

Відповідно в координатній формі, безпосередньо використовуваної для розрахунків, рівняння (2.8) набуде вигляду

$$P(t,s)=\left(C_x+a_xt+b_xs, C_y+a_yt+b_ys, C_z+a_zt+b_zs\right). \qquad (2.9)$$

Для усунення ефекту ступінчастості, пов'язаного з меншим просторовим дозволом у площині Z, до реконструйованого зображення на стадії остаточної обробки застосовувалася процедура усереднюючої фільтрації. На рис. 2.2, б наведено приклад МПР у площині, перпендикулярній носовим ходам.



а) б)

Рис. 2.2. Схематичні зображення площин вихідних аксіальних СКТ-зрізів S_k (а) і МПР у фронтальній площині (б)

Далі для проведення аеродинамічних розрахунків використовувалася бінарна воксельна модель $F(i,j,k,1)$ повітроносних порожнин носових ходів, отримана після багатозначної сегментації вихідних томографічних даних (побудова моделі розглядається далі у розд. 4), причому кількість перетинів визначалася як

$$N = l/h_{XY} \ ,$$

де l – довжина носового ходу (становить у середньому близько 70 мм);
h_{XY} – крок реконструкції (задавався 2 мм, і, відповідно, N = 35).

На отриманих внаслідок сегментації зображеннях відбувалося інтерактивне видалення областей, що належать придатковим пазухам носа і їхім сполученням (оцінка ролі придаткових пазух носа в аеродинаміці верхніх дихальних шляхів буде розглянута у підрозділі 2.5).

Далі проводилося обчислення площ і змочених периметрів перетинів носових каналів для кожної реконструкції

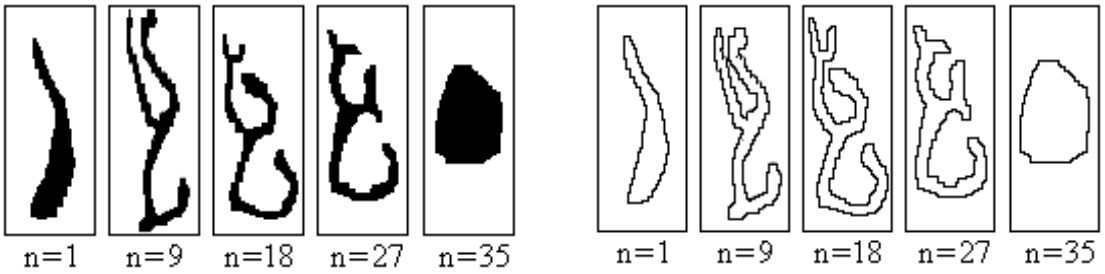
$$S = M_S \cdot h_{XY}^2 \ ; \quad \Pi = M_\Pi \cdot h_{XY} \ , \qquad (2.10)$$

де M_S – кількість елементів зображення, що належать перерізу носового каналу;
M_Π – кількість елементів зображення, що належать змоченому периметру перетину носового каналу.

Розрахунки проводилися за формулами (2.10) з урахуванням масштабного коефіцієнта, компенсуючого відмінність просторового дозволу за відповідними координатами в ході побудови мультипланарних реконструкцій. У ході визначення кількості елементів зображення, що належать змоченому периметру M_Π, в

кожному перетині носового каналу проводилася локальна обробка сегментованих зображень оператором просторового диференціювання з подальшим інвертуванням отриманих даних. Результати сегментації характерних перетинів носового ходу (для лівого носового проходу) та їх змочених периметрів наведені на рис. 2.3 (n – номери перетинів C_n). При цьому конфігурація перетинів носового каналу на вході має щелеподібну еліптичну форму, далі умовно поділяється на нижній, середній і верхній носові ходи, які на виході в носоглотку об'єднуються в широкий овальний отвір – хоану.

Проведемо аналіз отриманих геометричних характеристик перетинів носового ходу. Як видно з рис. 2.4, а, розподіли площі S перетинів носового каналу по 35 зрізах з кроком 2 мм, площа перетинів практично монотонно зростає від значення 70 мм^2 на вході в носовий канал до максимальної – площі хоани, що досягає більше 200 мм^2.



а) б)

Рис. 2.3. Сегментація характерних перетинів (а) і змочених периметрів (б) носових ходів

Згідно з конфігурацією перетинів носового каналу, змочений периметр (рис. 2.4, б) має максимум приблизно в середині довжини каналу і знижується з наближенням до хоани. Максимальне значення змоченого периметра складає 118 мм. Гідравлічний діаметр $d_г$ на вході й виході з носового каналу (рис. 2.4, в) досягає максимального значення, а його мінімум знаходиться в центральних перетинах. Глобальний максимум $d_г$ складає близько 20 мм на рівні хоани.

Порівняємо значення отриманих гідравлічних діаметрів з аналогічними параметрами для характерних об'єктів, наприклад, плоскої щілини

$$h_с << l_с ,$$

де $h_с$ і $l_с$ – висота і довжина щілини, відповідно;

$d_{гс}$ – гідравлічний діаметр (2.7) щілини дорівнює її подвоєній висоті

$$d_{гс} = \frac{4h_с \cdot l_с}{2 \cdot h_с + 2l_с} \approx 2h_с , \qquad (2.11)$$

і трубопроводу круглого перетину, для якого гідравлічний діаметр $d_{гтр}$ (2.7) дорівнює діаметру $d_{тр}$ цього трубопроводу

$$d_{гт} = \frac{4\pi \cdot d_т^2}{4\pi \cdot d_{тр}} = d_{тр} . \qquad (2.12)$$

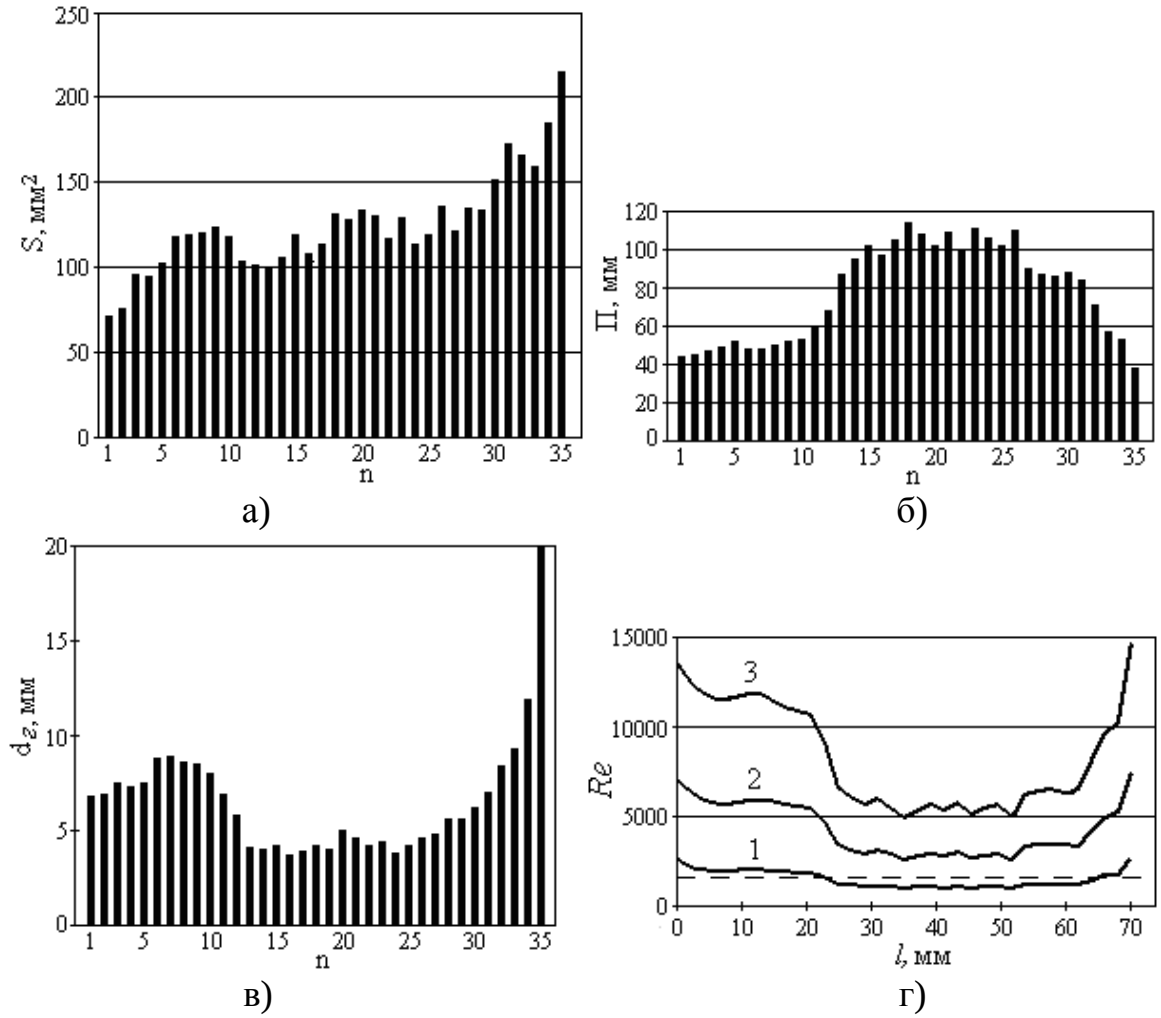


Рис. 2.4. Зміна параметрів за довжиною носової порожнини:
а – площі S перетинів; б – змоченого периметра П;
в – гідравлічного діаметра $d_г$; г – числа Рейнольдса
(при значеннях витрати: 1 – 0,3 л/с; 2 – 1 л/с; 3 – 2 л/с)

Таким чином, вхідний перетин носового каналу (n = 1) можна вважати пласкою щілиною: $l_с$ = 34мм; $h_с$ = 4,8 мм; $l_с / h_с \approx 7$ і $d_{гс} \approx$ 9,6 мм, що можна порівняти з розрахунковим значенням $d_г$ = 7 мм (рис. 2.4, в). Для круглого перетину на виході носового ходу (n = 35) діаметр і, відповідно, гідравлічний діаметр складають 15 мм, що також можна порівняти з розрахунковим значенням $d_г$ = 19,5 мм. Отримані результати дозволяють зробити висновок про можливість використання для визначення аеродинамічних опорів носових ходів стандартної методики розрахунку трубопроводів [227].

Для розрахунку числа Рейнольдса (2.6) значення коефіцієнта кінематичної в'язкості повітря визначається за формулою

$$\nu_{вязк} = 10^6 \frac{\mu}{\rho} = 10^6 \frac{1{,}81 \cdot 10^{-5}}{1{,}205} = 15{,}02 \text{ мм}^2/\text{с (сантіСтокс)}, \qquad (2.13)$$

де μ і ρ – коефіцієнт динамічної в'язкості і щільність повітря, значення яких за нормальних атмосферних умов (температури 20°С і тиску 760 мм рт.ст.) складають μ = 1,81·10–5 Н·с/м^2 і ρ = 1,205 кг/м^3 [302], відповідно.

Швидкість течії повітря носовим каналом визначаємо за формулою

$$V = 10^3 \frac{Q}{S}\ , \text{м / с}, \qquad (2.14)$$

де Q – витрата повітря через носовий канал, л/с;

S – площа поперечного перерізу каналу, мм^2.

З урахуванням співвідношень (2.13) і (2.14) формула (2.6) набуває вигляду

$$\mathrm{Re} = 10^3 \frac{V \cdot d_г}{\nu_{вязк}} = \frac{10^6 Q \cdot 4S}{\nu_{вязк} \cdot S \cdot \Pi} = \frac{4 \cdot 10^6 Q}{\nu_{вязк} \cdot \Pi}\ . \qquad (2.15)$$

Після підстановки значення в'язкості (2.13) отримуємо остаточний вираз для обчислення числа Рейнольдса

$$\mathrm{Re} = 2,7 \cdot 10^5 \frac{Q}{\Pi}, \qquad (2.16)$$

з якого випливає, що за постійної в'язкості число Рейнольдса залежить тільки від витрати повітря і змоченого периметра носового проходу. Задаючи витрату повітря в носовому проході в діапазоні реальних значень від 0,3 до 2 л/с і визначаючи змочені периметри перетинів каналів (рис. 2.5, в), проводимо розрахунок за формулою (2.16).

На рис. 2.4, в наведено залежності зміни чисел Рейнольдса за довжиною l носового каналу за різних значень витрати повітря. Таким чином, за витрат 0,3...2 л/с діапазон значень чисел Рейнольдса становить від 650 до 13500. Потік повітря є ламінарним при $\mathrm{Re} < \mathrm{Re}_{кр}$ і турбулентним у протилежному випадку. Для труб круглого перетину і гумових рукавів $\mathrm{Re}_{кр} = 1600 \ldots 2300$ [227], для інших гідравлічних опорів може знижуватися до $\mathrm{Re}_{кр} = 260 \ldots 1100$ залежно від конфігурації входу в опір і ступеня турбулізації потоку перед входом.

Таким чином, максимуми значень гідравлічного діаметра (рис. 2.4, в) мають місце при щелеподібній формі перетину на вході в носовий канал і при майже круглій на виході в носоглотку, що пояснюється малими змоченими периметрами даних перетинів порівняно зі серединними, які умовно поділяються на носові ходи. В ході аналізу зміни числа Рейнольдса по перетинах можна зробити висновок про те, що найбільша турбулізація потоку виникає при вході і виході з носового каналу. Під час спокійного дихання режим течії повітря у верхніх дихальних шляхах можна вважати ламінарним, при форсованому – турбулентним.

Отримані результати повністю кореспондуються з даними, наведеними в роботі [228], де показано, що тільки із витратою повітря до 0,25 л/с в носових проходах може мати місце ламінарний режим, а подальше зростання витрати повітря (наведені дані до 0,4 л/с) призводить до турбулізації течії повітря.

Проте, слід також відзначити, що в носових проходах витрата і швидкість повітря носять нестаціонарний характер, який обумовлений механікою роботи дихальних м'язів легень за типом гармонійних коливань, наприклад, як у поршневого насоса, тому ламінарний режим течії повітря практично не можливо реалізувати.

Таким чином, отримаємо значення швидкості повітря в одному носовому каналі із витратою 3,36 л/с

$$V = 10^3 \frac{Q}{2S} = 10^3 \frac{3{,}36}{2 \cdot 70} = 24\,\text{м/с}, \qquad (2.17)$$

де 2 – коефіцієнт у знаменнику, що враховує допущення про рівномірний розподіл витрат через кожний носовий прохід.

Розрахунок діючих значень швидкостей повітря має важливе практичне значення для визначення можливої травматизації слизової оболонки носової порожнини [190].

2.4 Динамічна модель течії повітря у носовій порожнині

Розглянемо процес дихання як аеродинамічний процес проходження повітря через носову порожнину і встановимо особливості, що виникають із рухом повітряного потоку через вхідні та вихідні отвори носових ходів.

Для дослідження основних аеродинамічних закономірностей дихального процесу в носовій порожнині розглянемо спрощену модель носового проходу, яка становить круглу циліндричну трубу радіусом a, в якій під впливом перепаду тиску, що періодично змінюється, переміщається повітря. При цьому як аналог приймаємо пульсуючий ламінарний рух в'язкої нестисливої рідини по круглій циліндричній трубі.

Рівняння нестаціонарного ламінарного усталеного (котрий залежить від вісьової координати) руху в'язкої нестисливої рідини в циліндричній трубі круглого перетину при зовнішньому тиску $p = \Delta p \cos \omega t$, що гармонійно змінюється, має вигляд [225]

$$\frac{\partial w}{\partial t} - \nu \left(\frac{\partial^2 w}{\partial r^2} + \frac{1}{r} \frac{\partial w}{\partial r} \right) = \frac{\Delta p}{\rho L} \cos \omega t \ , \qquad (2.18)$$

де w – швидкість руху повітря уздовж осі z;

r – радіальна координата уздовж радіуса труби;

t –час;

Δp – перепад тисків уздовж труби довжиною L;

ρ – щільність повітря;

ν – коефіцієнт кінематичної в'язкості повітря;

ω – частота гармонійних коливань зовнішнього тиску.

Це рівняння слід підпорядкувати очевидній граничній умові рівності нулю швидкості повітря на стінці труби ($w = 0$ при $r = a$). Початкова умова визначається максимальним значенням перепаду Δp зовнішнього тиску, що змінюється за гармонійним законом (при $t = 0$, $p = \Delta p$). Існує аналітичне розв'язання цього рівняння [225], яке виражається через модифіковані

циліндричні функції Кельвіна $ber(x)$ і $bei(x)$, пов'язані з функцією Бесселя нульового порядку від комплексного аргументу $J_0\left(x\sqrt{i}\right)$ співвідношенням

$$J_0\left(x\sqrt{i}\right)=ber\left(x\right)-i\,bei(x), \tag{2.19}$$

і має вигляд

$$w(r,t)=\frac{\Delta p}{\omega\rho L}\left[\begin{array}{l}\left(1-\dfrac{bei\,x_a bei\,x+ber\,x_a ber\,x}{ber^2 x_a+bei^2 x_a}\right)\sin\omega t+\\ +\dfrac{bei\,x_a ber\,x-ber\,x_a bei\,x}{ber^2 x_a+bei^2 x_a}\cos\omega t\end{array}\right], \tag{2.20}$$

де $x=r\sqrt{\frac{\omega}{\nu}}$, $x_a=a\sqrt{\frac{\omega}{\nu}}$ і $\omega=\frac{2\pi}{T}$ з періодом коливань T.

Вважаючи середній радіус носового ходу $r=a\approx 6$ мм, щільність повітря і його коефіцієнт кінематичної в'язкості за нормальних умов ρ=1,205 кг/м^3 і $\nu=15{,}02$ мм^2/с, відповідно, з періодом коливань, що задається $T\approx 5$ с отримаємо значення x_a 0,8. Функції Кельвіна цього аргументу мають значення $ber\left(0{,}8\right)=0{,}99$ і $bei\left(0{,}8\right)=0{,}16$.

У режимі спокійного дихання, коли характер руху повітря в носовій порожнині близький до ламінарного, розглянута модель відображає загальні закономірності дихального процесу. Це видно з наведених на рис. 2.5 графіків залежності від часу (фази коливання) відносної швидкості потоку, розрахованої за формулою (2.20) і за експериментальними даними (відносна швидкість є відношенням поточного значення швидкості до максимального значення). Ідентичність залежностей відносних величин швидкостей від часу дозволяє перенести особливості даної моделі на натурний зразок.

Введемо позначення

$$C_1\left(\omega,r\right)=1-\frac{bei\,x_a bei\,x+ber\,x_a ber\,x}{ber^2 x_a+bei^2 x_a}\ , \tag{2.21}$$

$$C_2\left(\omega,r\right)=\frac{bei\,x_a ber\,x-ber\,x_a bei\,x}{ber^2 x_a+bei^2 x_a}\ . \tag{2.22}$$

Перетворимо вираз (2.20) з урахуванням співвідношень (2.21) і (2.22)

$$w(r,t)=\frac{\Delta p}{\omega\rho L}\cdot\left[C_1\left(\omega,r\right)\sin\left(\omega t\right)+C_2\left(\omega,r\right)\cos\left(\omega t\right)\right]=\frac{\Delta p\cdot\sqrt{C_1^2\left(\omega,r\right)+C_1^2\left(\omega,r\right)}}{\omega\rho L}\times$$

$$\times\left[\frac{C_1\left(\omega,r\right)}{\sqrt{C_1^2\left(\omega,r\right)+C_1^2\left(\omega,r\right)}}\sin\left(\omega,t\right)+\frac{C_2\left(\omega,r\right)}{\sqrt{C_1^2\left(\omega,r\right)+C_1^2\left(\omega,r\right)}}\cos\left(\omega,t\right)\right]=$$

$$=\frac{\Delta p\cdot\sqrt{C_1^2\left(\omega,r\right)+C_1^2\left(\omega,r\right)}}{\omega\rho L}\left[\sin\left(\delta\right)\sin\left(\omega t\right)+\cos\left(\delta\right)\cos\left(\omega t\right)\right].$$

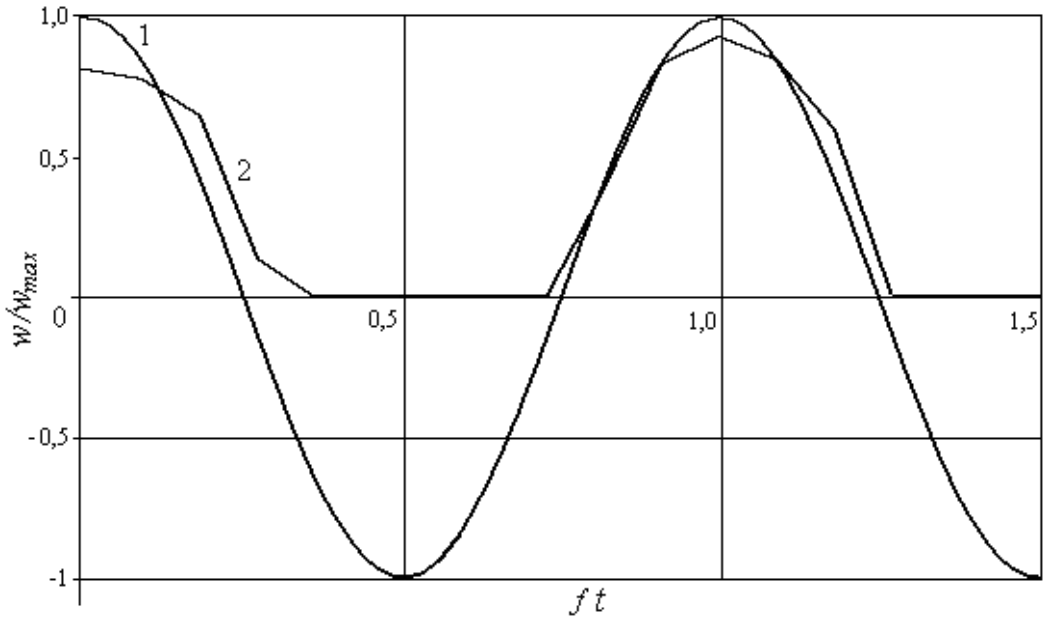


Рис. 2.5. Залежність відносної швидкості повітря в носовій порожнині від фази коливань: 1 – теоретичне значення; 2 – експериментальна залежність

Тоді, з урахуванням тригонометричних перетворень, рівняння (2.20) набуває вигляду

$$w(r,t)=\frac{\Delta p}{\omega \rho L}\sqrt{C_1^2(\omega,r)+C_2^2(\omega,r)}\cdot\cos(\omega t\text{-}\delta)\ , \qquad (2.23)$$

де $C_1(\omega,r)$ і $C_2\ (\omega,r)$ – множники при тригонометричних функціях, визначені за формулами (2.21) і (2.22);

δ – різниця фаз [град] між швидкістю і тиском, причому відношення зазначених множників $C_1(\omega,r)$ і $C_2\ (\omega,r)$ дорівнює тангенсу кута різниці фаз

$$\frac{C_1(\omega,r)}{C_2(\omega,r)}=tg\delta\ . \qquad (2.24)$$

Оскільки зовнішній тиск задано у вигляді $p=\Delta p\cos\omega t$, то між швидкістю і тиском існує різниця фаз δ, яка є функцією координат і частоти (в'язкість повітря в даних умовах є постійною). Тональна діаграма залежності різниці фаз δ від безрозмірного радіусу r/a і частоти f наведено на рис. 2.6, з якої випливає, що різниця фаз зростає із зростанням безрозмірного радіусу за малих частот і має мінімум по радіусу в області високих частот (низькі та високі частоти визначаються наведеними на діаграмі значеннями цієї величини). Залежність різниці фаз від частоти більш складна – за малих значень різниця фаз збільшується зі зростанням частоти, а за великих має максимум і мінімум.

Зазначимо, що відношення $\Delta p/L$ є градієнтом тиску, і рівняння (2.23) може бути подано у вигляді

$$w(r,t)=\frac{\sqrt{C_1^2(\omega,r)+C_2^2(\omega,r)}}{\omega\rho}\cos(\omega t-\delta)\cdot \operatorname{grad} p\ . \qquad (2.25)$$

З огляду на те, що швидкість є функцією витрати, рівняння (2.25) формально становить кінетичне рівняння переносу, в якому $\operatorname{grad} p$ можна розглядати як термодинамічну силу, яка обумовлює дану витрату, а коефіцієнт, що знаходиться перед $\operatorname{grad} p$ – як відповідний коефіцієнт перенесення. Середнє за періодом значення скалярного добутку швидкості потоку на термодинамічну силу (процедура усереднення аналогічна усередненню потужності змінного струму) визначає величину щільності дисипативної функції потужності D

$$D(r) = \frac{\sqrt{C_1^2(\omega, r) + C_2^2(\omega, r)}}{\omega \rho} \operatorname{grad}^2 p \cdot \cos\delta \ . \qquad (2.26)$$

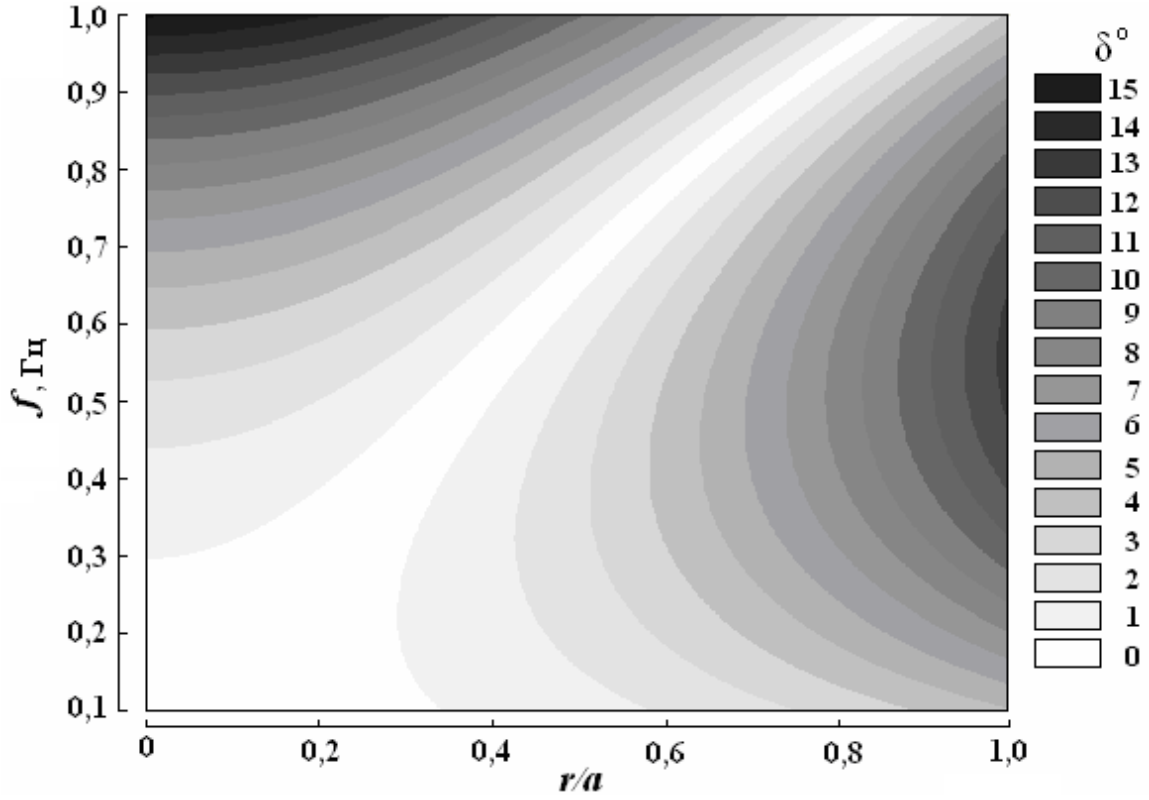


Рис. 2.6. Тональна залежність різниці фаз
від безрозмірного радіусу r/a і частоти f з перепадом тиску 1 кПа

Світла область на тональній діаграмі (рис. 2.6) відповідає найбільшим значенням дисипативної функції. За малої частоти дихання область максимальної дисипації потужності дихання знаходиться на осі носового ходу, а з підвищенням частоти зміщуватиметься в пристінкову область. Створена модель не враховує взаємодію повітряного потоку зі стінкою носа, але дозволяє виявити області максимальної дисипації енергії в порожнині носа за рахунок внутрішнього тертя.

2.5 Розробка математичних моделей аеродинамічних і дифузійних процесів у придаткових пазухах носа та їх експериментальна перевірка

Розглянемо повітряний потік на ділянці носового ходу, що включає сполучення повітряної порожнини. На рис. 2.7 показаний аксіальний (відносно осі співустя пазухи) перетин повітряної порожнини з характерними для верхньощелепної пазухи розмірами, яка вважається в плані ізометричної із середнім розміром близько 25 мм. Можна виділити всередині пазухи область, обмежену внутрішньою поверхнею пазухи і бічною поверхнею циліндра, діаметр якого збігається з діаметром D співустя, а висота – з глибиною h пазухи і розрахувати потік вектора швидкості повітря через дану поверхню. Оскільки внутрішня поверхня пазухи непроникна для повітря, потік може бути відмінним від нуля тільки на бічній поверхні циліндра.

Відповідно до теореми Гауса, величина цього потоку дорівнює швидкості генерування об'єму V повітря в порожнині

$$\oiint v_n dS = \frac{dV}{dt} \ , \tag{2.27}$$

де v_n – нормальна відносно бічної поверхні циліндра швидкість.

Оскільки всередині виділеної поверхні відсутні джерела і стоки повітря, то

$$\frac{dV}{dt} = 0 , \tag{2.28}$$

звідки випливає, що формула (2.27) перетвориться до вигляду

$$\oiint v_n dS = 0 .$$

При цьому можливі два випадки:

– у всіх точках бічної поверхні циліндра нормальна швидкість дорівнює нулю $(v_n = 0)$;

– на одних ділянках поверхні швидкість $v_n > 0$, а на інших $v_n < 0$, що також може привести до обвернення в нуль повного потоку вектора швидкості.

Проте, якщо розбити виділену область компланарними перетинами, відстані між якими утворюють нескінченно тонкі шари величиною dh, у межах яких швидкість може характеризуватися тільки цілком певним значенням, то з умови (2.28) випливає, що це значення має бути рівним нулю, тобто реалізується перший випадок (нормальна швидкість дорівнює нулю).

При цьому можливе існування тангенціальної швидкості v_r, спрямованої, як видно з рисунка, вздовж r.

Для оцінки величини цієї швидкості запишемо рівняння нерозривності для точки A, яке за умови нестисненості середовища і відсутності джерел має вигляд

$$\frac{\partial v_z}{\partial z} + \frac{\partial v_r}{\partial r} = 0 \ . \tag{2.29}$$

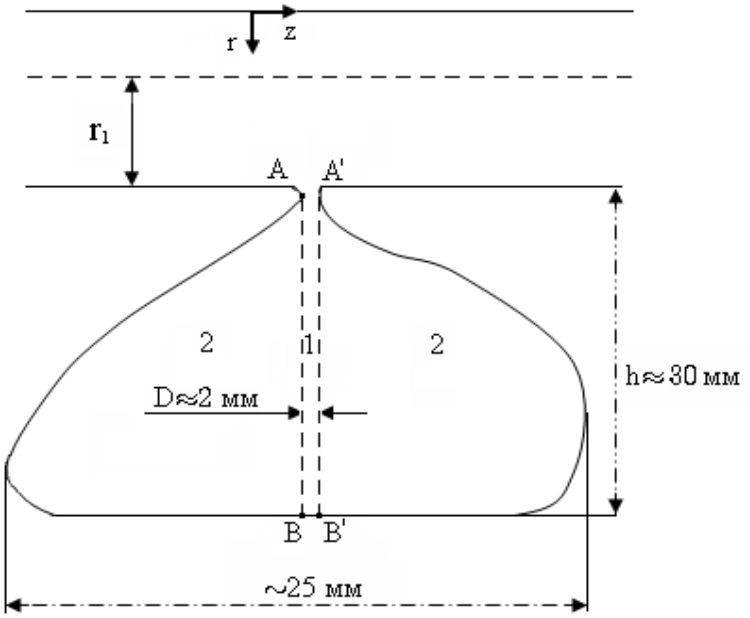


Рис. 2.7. Аксіальний (відносно осі співустя) перетин верхньощелепної пазухи

Переходячи в цьому рівнянні від часткових диференціалів до кінцевих приростів і вважаючи, що точка A знаходиться безпосередньо на поверхні тканини на умовній межі між носовим ходом і співустям, а також, вважаючи, що зміна швидкості відбувається стрибком на товщині граничного шару δ $(\Delta z \approx \Delta r \approx \delta)$, отримаємо

$$\Delta v_z = -\Delta v_r \ . \tag{2.30}$$

Для знаходження цих величин запишемо рівність потоків до входу в співустя $(z < A)$ і після входу в нього $(A < z < A')$

$$\pi r_1^2 v_{z(1)} \approx \left[\pi r_1^2 + \pi \left(\frac{h}{2} \right)^2 \right] v_{z(2)} \ , \tag{2.31}$$

де $v_{z(1)}$, $v_{z(2)}$ – швидкості повітряного потоку перед співустям $(z < A)$ та після входу до співустя $(A < z < A')$, відповідно.

Вважаючи на підставі даних [1190, 196, 197], що $v_{z(1)} \approx 2{,}75$ м/с і середній радіус носового ходу $r_1 \approx 3$мм, і враховуючи відповідно до рисунка $h \approx 30$мм, визначаємо за формулою (2.31) $v_{z(2)} \approx 0{,}04$ м/с.

Згідно з отриманими даними, на підставі рівняння (2.30) можна визначити величину стрибка радіальної швидкості при вході в співустя

$$\Delta v_r = -\Delta v_z \approx 2{,}7 \text{ м/с},$$

і оскільки перед входом у співустя радіальна швидкість $v_{r(1)} \approx 0$, то всередині співустя її значення складе $v_{r(2)} \approx 2{,}7$ м/с.

У симетричній точці A', якщо не враховувати незначний у цьому випадку вплив в'язкості, повторюючи пророблені перетворення, можна отримати значення швидкості $v'_{r(2)} \approx -2{,}7$ м/с.

Таким чином, радіальна швидкість змінює свій напрямок, перетворюючись у нуль на осі співустя пазухи. Зазначимо, що радіальна швидкість потоку обумовлює явище ежекції – залучення у потік прилеглих до струменя областей рідини або газу, проте, згідно з експериментальними даними ежекція відбувається у чим вужчій області, чим менше в'язкість рідини, і для повітря можна вважати це явище несуттєвим.

Згідно з розглянутою моделлю, вентиляція в пазусі відбувається лише в об'ємі 1 (див. рис. 2.7), що безпосередньо примикає до області співустя, що дорівнює

$$V = \frac{\pi D^2}{4} \cdot h \ ,$$

має значення близько 1 см^3, у той час як загальний об'єм великих придаткових пазух (верхньощелепних, клиновидних і лобових) становить 10...20 см^3 [190]. Припускаючи, що радіальна швидкість змінюється в межах співустя уздовж осі z за лінійним законом, зменшуючись до нуля від стінки співустя до його осі і зростаючи за модулем з наближенням до протилежної стінки, можна вважати, що середня швидкість на кожній половині співустя дорівнює половині пристінкової радіальної швидкості.

При цьому витрата повітря в зоні вентиляції через кожну з половин співустя дорівнює

$$Q \approx \frac{1}{2} \frac{v_{r(2)}}{2} \frac{\pi D^2}{4} \ ,$$

що становить близько 2 см^3/с, а середній час вентиляції

$$t_v \approx 4 \frac{h}{v_{r(2)}} \qquad (2.32)$$

складає $t_v \sim 0{,}05$ с.

Таким чином, вентильована частина повітряної порожнини складає близько 5...10% від її загального об'єму, а інша частина є зоною застою.

Якісно оцінюючи роль співустій придаткових пазух носа в аеродинаміці верхніх дихальних шляхів, можна відзначити, що в нормі при спокійному диханні ламінарний потік повітря, який поширюється через ніздрю, частково відгалужується до співустя пазухи, що призводить до створення місцевого опору, що сприяє турбулізації потоку [225]. У разі блокування співустя придаткової пазухи ділянка носового проходу стає однорідною, що сприяє більшій стійкості ламінарного режиму. Висновок про турбулентний характер потоку за відсутності патології носових каналів підтверджується результатами, наведеними в [226].

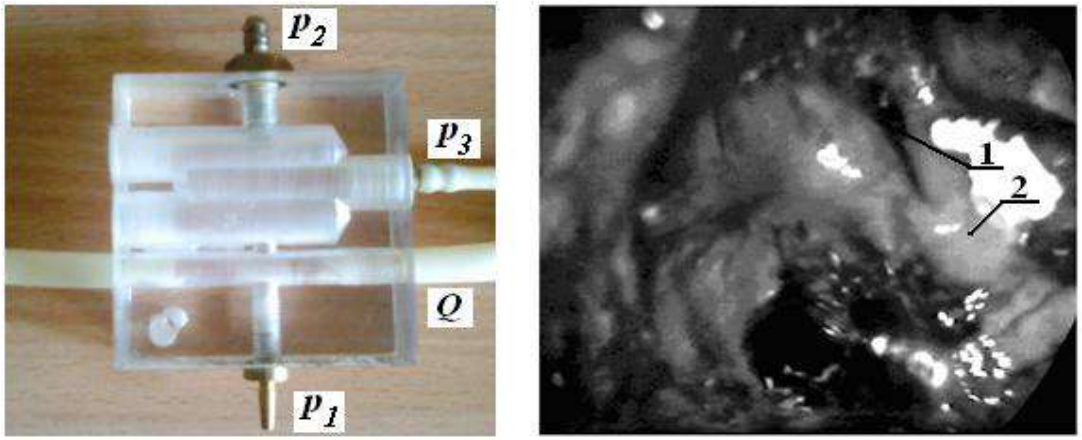
Для підтвердження теоретичної моделі доцільно провести експериментальні дослідження, спрямовані на уточнення повітрообміну в придаткових пазухах носа.

Перший експеримент для з'ясування можливості руху повітря за рахунок перепаду тиску між придатковою пазухою і носовим ходом виконувався на спеціально виготовленій натурній моделі носового ходу і придатковій пазусі, виконаної з прозорого скла (рис. 2.8, а). Носовий прохід моделювався трубою

круглого перетину діаметром 8 мм, а порожнину придаткової пазухи трьома сполученими циліндрами діаметром 12 мм і довжиною 50 мм. Сполучення пазухи виконувалося у вигляді циліндричного отвору діаметром 3 мм у місці з'єднання з носовим ходом. Точками вимірювання тиску були: $p1$ – у носовому проході навпроти співустя; $p2$ – у латеральній стінці пазухи навпроти співустя; $p3$ – у дистальній стінці пазухи (для вимірювання перепаду тиску в периферійній області пазухи).

Методика експерименту полягала у вимірюванні показників датчиків, встановлених у точках $p1, p2$ і $p3$ при значеннях витрати повітря через ніздрю від 0,3 до 8 л/с (0,3; 0,5; 1,0; 2,0; 3,0; 4 , 0; 6,0 і 8,0 л/с). При цьому показники датчиків у сталих режимах за різних витрат не змінювалися $p1 = p2 = p3$, а зі зміною витрати значення перепаду тисків $p1, p2$ і $p3$ змінювалися синхронно, що свідчить про відсутність значущих аеродинамічних ефектів ежекції (всмоктування/висмоктування) повітря з придаткових пазух носа під час дихання, що також підтверджується клінічним експериментом, розглянутим нижче.

Вихідними даними клінічного експерименту слугували результати досліджень, проведених під час гайморотомії – втручань для лікування хронічного гайморита. Дослідження проводилися в оториноларингологічному відділенні Харківської обласної клінічної лікарні.



а б

Рис. 2.8. Дослідження повітропостачання придаткових пазух носа:
а – натурна модель для оцінки перепаду тисків між областями додаткової пазухи і носовим ходом (стрілкою показано напрямок потоку повітря Q через ніздрю); б – ілюстрація проходження цигаркового диму (2) з носового ходу через співустя (1) верхньощелепної (гайморової) пазухи під час дихання за даними ендоскопічного відеоспостереження під час гайморотомії

При ендоскопічному відеоспостереженні в носову порожнину (в природному циклі вдиху) вводився цигарковий дим, який через співвустя 1 (рис. 2.8, б) під час дихання проникав у порожнину гайморової пазухи кільцеподібними

клубками 2 у вигляді спіралі, спостереження характеру яких дозволило судити про турбулентний дифузійний процес поширення повітря (у даному випадку – аерозольної суміші) із співустя у придаткові пазухи.

Для теоретичної оцінки швидкості поширення повітря у придаткових пазухах носа розглянемо модель дифузії аерозолю в придаткових пазухах носа. Згідно з законом Фіка, для стаціонарної дифузії щільність $\overrightarrow{j_\partial}$ дифузійного потоку дорівнює

$$\overrightarrow{j_\partial} = -D \cdot grad\ n,$$

де D – коефіцієнт дифузії;

n – об'ємна концентрація частинок (число частинок в одиницю об'єму).

З огляду на те, що щільність дифузійного потоку $\overrightarrow{j_\partial}$ пов'язана з концентрацією n і швидкістю $\vec{V}$ співвідношенням

$$\overrightarrow{j_\partial} = n \cdot \vec{V},$$

то для довільного напрямку $\vec{r}$ отримуємо

$$n \cdot V = -D\frac{dn}{dr}. \qquad (2.33)$$

Після розділення змінних маємо

$$Vdr = -D\frac{dn}{n}; \quad V \cdot r\,|_0^{r'} = -D\ln\ n\,|_{n_{\max}}^{n'}; \quad V = \frac{-D \cdot \ln\frac{n'}{n_{\max}}}{r'} = \frac{D}{r'}\ln\frac{n_{\max}}{n'},$$

де n' – концентрація в точці r'.

Якісно швидкість дифузії можна визначити, вважаючи, що в початковий момент часу $\Delta n = n - 0$ (вважаючи, що на стінці $n = 0$, а $r = r_{\text{П}}$, де $r_{\text{П}}$ – глибина пазухи). Тоді рівняння (2.33) можна подати у вигляді

$$n \cdot V \approx D\frac{n}{r_{\text{П}}},$$

звідки

$$V \approx \frac{D}{r_{\text{П}}}. \qquad (2.34)$$

З огляду на те, що для повітря у ламінарному режимі $D \approx 10^{-5}\ \text{м}^2/c$ і характерний розмір пазухи близько $r \approx 2 \cdot 10^{-2}$ м, швидкість дифузії становить

$$V \approx \frac{10^{-5}}{2 \cdot 10^{-2}} = 0{,}5 \cdot 10^{-3}\ \text{м/ с}$$

і, відповідно, час дифузного повітрообміну

$$\tau = \frac{r_{\text{П}}}{V} = \frac{2 \cdot 10^{-2}}{0{,}5 \cdot 10^{-3}} = 40\ \text{с}.$$

Для турбулентного режиму течії повітря коефіцієнт дифузії D^* збільшується приблизно на два порядки і відповідно

$$D^* \approx 100 \cdot D,\ \ V^* \approx 100 \cdot V,\ \ \tau^* \approx \frac{\tau}{100}.$$

Швидкість V^* дифузії, згідно з формулою (2.34), становить близько 0,05 м/с, а час τ^* дифузійного повітрообміну порядку 0,4 с. Таким чином, швидкість оновлення повітря у придаткових пазухах носа більш ніж на два порядки менше швидкості повітря в носовій порожнині в процесі дихання, що візуально підтверджується експериментами. Час аерації застійної області на порядок більше, ніж області, що примикає до співустя. Витрата повітря визначатиметься площею співустя пазухи. При цьому, відповідно до експерименту, повністю відсутні ефекти, пов'язані зі значною ежекцію повітря з придаткових пазух через співустя, що пояснюється порівняно невисокими швидкостями повітря у верхніх дихальних шляхах і малою в'язкістю повітря.

Таким чином, у результаті аналізу енергетичних параметрів дихання людини визначено, що механічна потужність дихальних м'язів за форсованого дихання складає близько 10% від максимальної короткочасної потужності людини під час виконання фізичної роботи.

Відповідно до теоретичної та емпіричної оцінки ступеня стисливості повітря у верхніх дихальних шляхах людини під час дихання встановлено, що при швидкостях повітряного потоку в носових проходах менше 50 м/с можна вважати повітря нестисненим середовищем за критерієм $M \approx 0{,}15 < 0{,}3$ та застосувати відповідні базові аеродинамічні моделі і розрахункові співвідношення.

За результатами аналізу розробленої моделі одновимірної течії повітря в носовій порожнині введено поняття гідравлічного діаметра носових проходів, що характеризує відношення площі до змоченого периметра в кожному перетині повітряного каналу. Розподіл значень гідравлічного діаметра по довжині носового ходу має максимуми при щелевидній формі перетину на вході в носовий канал і при майже круглій біля виходу в носоглотку, що пояснюється малими змоченими периметрами даних перетинів порівняно зі серединними.

Аналіз зміни числа Рейнольдса по перетинах носового проходу дозволяє зробити висновок про те, що найбільша турбулізація потоку виникає при вході і на виході з носового каналу, причому при спокійному диханні (витрата повітря не більше 0,3 л/с) режим течії повітря у верхніх дихальних шляхах можна вважати ламінарним, а при форсованому – турбулентним.

На основі розробленої динамічної моделі течії повітря в носовій порожнині побудовано просторовий розподіл величини дисипативної функції потужності повітряного потоку від частоти дихання, що дозволяє визначити ділянки носових ходів, у яких відбуваються найбільші втрати потужності за рахунок внутрішнього тертя. При цьому область максимальної дисипації енергії дихання, що знаходиться на осі носового ходу, а з підвищенням частоти дихання зміщується в пристінкову область. Проведене теоретичне обґрунтування виникнення різниці фаз між сигналами витрати повітря і перепаду тиску на носовій порожнині дозволяє визначати ступінь дисипації енергії внаслідок тертя повітря об стінки носової порожнини заи малої частоти дихання.

На підставі теоретичних і експериментальних досліджень аеродинаміки і оцінки дифузійного потоку у верхньощелепній пазусі встановлено, що повітрообмін у придаткових пазухах носа здійснюється зі швидкістю близько 5 см/с (0,05 м/с) і носить дифузний турбулентний характер, що не залежить від аеродинамічних процесів у носовій порожнині, за яких швидкість повітря складає близько 10...20 м/с (час дифузійного повітрообміну становить близько 0,4 с). При цьому витрата повітря визначатиметься тільки площею співустя пазухи (згідно з проведеним математичним і натурним моделюванням повністю відсутні ефекти, пов'язані з висмоктуванням повітря з придаткових пазух через співустя, що пояснюється порівняно невисокими швидкостями повітря у верхніх дихальних шляхах, малою в'язкістю повітря і відсутністю наскрізної вентиляції придаткових пазух). Дане фундаментальне положення дозволяє провести переоцінку фізіологічної функції співусть придаткових пазух носа з точки зору ступеня їх вентиляції, що може стати теоретичним обґрунтуванням для перегляду або оптимізації лікувальної тактики під час проведення терапії захворювань верхніх дихальних шляхів.

3 РОЗРОБКА МОДЕЛЕЙ І МЕТОДІВ ІНТЕЛЕКТУАЛЬНОГО АНАЛІЗУ ДАНИХ ПІД ЧАС ТЕСТУВАННЯ НОСОВОГО ДИХАННЯ

3.1 Основні принципи тестування носового дихання

Риноманометрія – метод кількісної оцінки функції носового дихання [190 –219], заснований на вимірюванні перепаду тиску між входом і виходом з носової порожнини та витрати повітря, що пропускається при цьому. Традиційно основним показником риноманометричної діагностики є коефіцієнт опору носового дихання (аеродинамічного опору носових проходів), який визначається як відношення перепаду тисків Δp на носовій порожнині до відповідного значенням витрати повітря Q

$$A = \frac{\Delta p}{Q}\left[\frac{кПа}{л/с}\right]. \qquad (3.1)$$

Усереднена за часом (за кількістю дихальних циклів) величина відношення пікових значень перепаду тисків до витрати повітря Q є значущим діагностичним показником носової провідності [194 – 207]. В основі створення сучасних діагностичних пристроїв, що забезпечують проведення передньої активної (ПАРМ) або задньої активної риноманометрії (ЗАРМ) носових проходів, лежать закони і рівняння гідравліки (пневматики) – закон Паскаля і рівняння нерозривності [225].

На рис. 3.1 наведена принципова пневматична схема системи дихання людини, де для опорів носових проходів і носового клапана введено такі позначення:

– дросель $ДР_{л.кп}$ – аналог опору клапана (крил носа) на вході в лівий носовий прохід, який автоматично закривається під час інтенсивного дихання через ніс;

– дросель $ДР_{л.нп}$ – аналог комбінації опорів за довжиною і місцевих уздовж лівого носового проходу;

– дроселі $ДР_{п.кп}$ і $ДР_{п.нп}$ – опори в правому носовому проході, аналогічні зазначеним для лівого проходу;

– дросель $ДР_{Р}$ включає місцеві опори у вигляді раптового звуження і розширення потоку під час проходження повітря через губи і втрати за довжиною вздовж носоглотки. У зв’язку з істотно більшою площею порожнини рота порівняно з носовими проходами, втратами за довжиною в носоглотці можна знехтувати.

Можливість вимірювання перепаду тисків на носових проходах реалізується за допомогою закону Паскаля – властивості текучого середовища (рідини або повітря) передавати зовнішній тиск всім розташованим всередині неї частинкам без зміни [225]

$$p_{л.н} = p_{л.нг} = p_{п.н} = p_{п.нг} = p_{р.нг} = const \text{ , КПа,} \qquad (3.2)$$

де $p_{л.нг}$ і $p_{п.нг}$ – тиски на виході з лівого і правого носових проходів (хоан) в носоглотку відповідно;

$p_{\text{л.н}}$ *і* $p_{\text{п.н}}$ – тиски, що вимірюються датчиками, встановленими герметично на вході лівого або правого носового проходів відповідно;

$p_{\text{р.нг}}$ – тиск, що вимірюється в носоглотці за допомогою датчика на кінці трубки, дистальний кінець якої встановлюється герметично в роті між губами пацієнта.

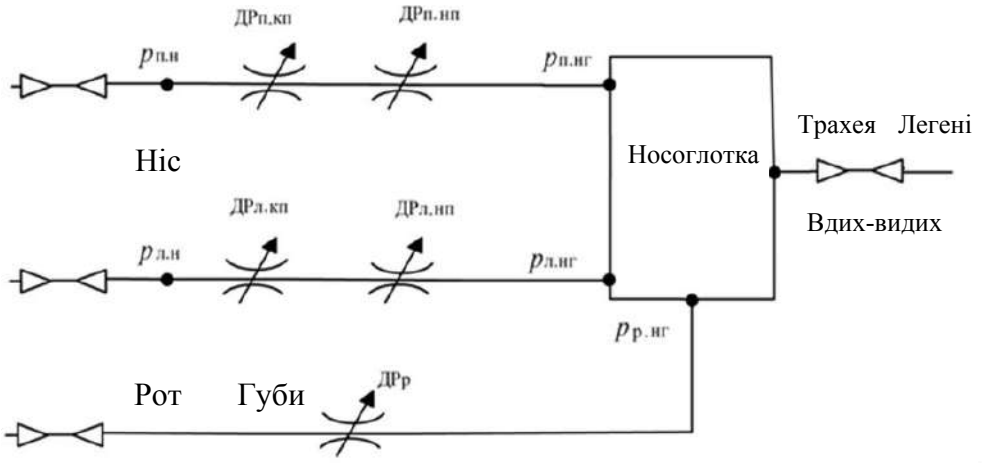


Рис. 3.1. Принципова пневматична схема системи дихання людини із зазначенням точок вимірювання тиску

При цьому для методу передньої риноманометрії з розміщенням датчика тиску по черзі і герметично в кожному носовому проході, закон Паскаля дозволяє отримати такі рівності згідно з формулою (3.2)

– при тестуванні лівого носового проходу

$$p_{\text{л.нг}} = p_{\text{пр.н}}; \tag{3.3}$$

– при тестуванні правого носового проходу

$$p_{\text{п.нг}} = p_{\text{л.н}}. \tag{3.4}$$

Для методу задньої риноманометрії рівність тисків відповідно до формули (3.2) може бути подана у вигляді

$$p_{\text{л.нг}} = p_{\text{п.нг}} = p_{\text{р.нг}}. \tag{3.5}$$

На підставі наведених рівностей складена табл. 3.1 із зазначенням точок вимірювання тиску залежно від методу риноманометрії.

Таблиця 3.1

Точки вимірювання тиску залежно від методу риноманометрії

Вимірювання тиску	Передня риноманометрія			Задня риноманометрія		
	Л.П	Пр.П	Л.П.+ Пр.П	Л.П	Пр.П	Л.П.+ Пр.П
$p_{\text{л.н}}$		+	-			
$p_{\text{п.н}}$	+		-			
$p_{\text{р.нг}}$				+	+	+

Примітки: 1. Л.П. і Пр.П – тестування і дихання через лівий і правий носовий проходи, відповідно; 2. Л.П + Пр.П – тестування і дихання через обидва носових проходи.

З рівняння нерозривності або суцільності потоку нестисливого текучого середовища згідно з [225] випливає, що витрата середовища через кожен (*i*-й) перетин каналу є постійною величиною

$$Q = v_1 \cdot S_1 = v_2 \cdot S_2 = v_i \cdot S_i = const \text{, м/с,} \qquad (3.6)$$

де v – швидкість течії повітря, м/с;

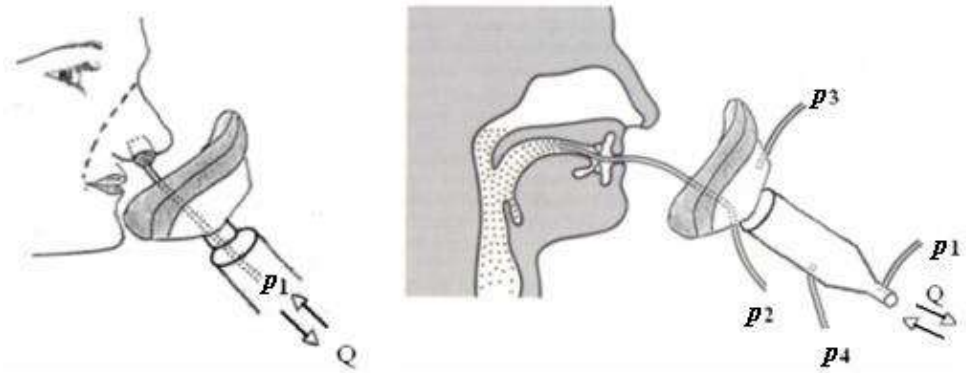
S – площа перерізу (часто зустрічається термін площа «живого» перетину як частина поперечного перерізу каналу, заповненого повітрям), м^2.

Розміщення перетворювачів (датчиків) тиску і витрати повітря ілюструється на схемах пристроїв для проведення передньої та задньої активної риноманометрії, наведених на рис. 3.2, а і б відповідно.

Під час проведення ПАРМ вимірюється витрата повітря Q через один з носових проходів і перепад тисків Δp між атмосферним і в носоглотці (вимірювання тиску фізично проводяться відповідно до виразів (3.3) і (3.4) на вході одного з носових ходів, герметично закритого) за допомогою диференціального датчика p_1, причому дихання здійснюється через інший носовий хід. Достовірність діагностики при цьому істотно зменшується через розширення одного носового ходу при обтуруванні іншого і, як наслідок, неможливість коректного алгебраїчного додавання послідовно вимірюваних витрат повітря $Q_{Л_{ПАРМ}}$ і $Q_{П_{ПАРМ}}$ через лівий і правий носові ходи відповідно, для отримання сумарної витрати повітря Q під час фізіологічного дихання для подальшого розрахунку за формулою (3.1)

$$Q \approx Q_{Л_{ПАРМ}} + Q_{П_{ПАРМ}} .$$

Метод ЗАРМ передбачає вимірювання сумарної витрати повітря Q під час дихання носом через обидва носові ходи і перепаду тисків Δp між атмосферним і в носоглотці (дистальний кінець вимірювальної трубки датчика тиску $p2$ вводиться через ротову порожнину) згідно з виразом (3.19).



а б

Рис. 3.2. Схеми розміщення датчиків тиску *p* і витрати повітря при передній (а) і задній (б) активній риноманометрії

Таким чином, згідно зі схемою, наведеною на рис. 3.2, б, визначення величини витрати повітря Q можна здійснити за допомогою розташованого в дифузорі (згідно з принципом роботи сопла Вентурі) датчика тиску p_1, а перепад тиску на носових проходах визначається як різниця значень диференціальних датчиків тиску p_2 і p_3

$$\Delta p = p_2 - p_3 \,, \tag{3.7}$$

виконують вимірювання у ротовій порожнині і на вході в носові ходи (всередині маски), причому $p_2 = p_{\text{Р.НГ}}$ і $p_3 = p_{\text{Л.Н}} = p_{\text{П.Н}}$ згідно з виразами (3,2) і (3.5), і схемою, зображеною на рис. 3.1. Датчики p_1, p_2 і p_3 вимірюють розрідження відносно атмосферного тиску (в циклі вдиху), а датчик p_4 − надлишковий тиск у циклі видиху для фіксації дихальних фаз. З огляду на те, що датчики тиску є диференціальними, то далі рівнозначними є значення $\Delta p_* = p_*$, де індекс $(_*)$ позначає будь-який з використовуваних датчиків.

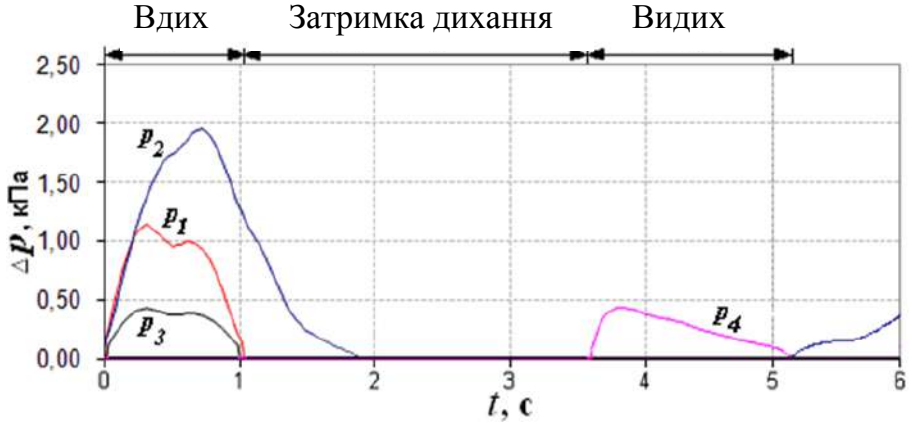
3.2 Розробка методу динамічної задньої активної риноманометрії

Запропонований метод динамічної ЗАРМ передбачає вивчення стандартних показників ЗАРМ у процесі дихання (в динаміці). На рис. 3.3, а наведено діаграму одного дихального циклу, отриманого за допомогою розробленого комп'ютерного риноманометра КРМ типу ТНДА-ПРХ [229]. У циклі вдиху, який фіксується ненульовим значенням датчика тиску p_1, встановленого у витратомірі на основі сопла Вентурі, сигнали тиску датчиків p_1, p_2 і p_3, фіксують розрядження, досягають максимального значення, а із затримкою дихання сигнали всіх датчиків дорівнюють нулю. Цикл видиху фіксується за ненульовими показаниками датчика тиску p_1. Показання датчика p_2, що вимірює тиск у носоглотці (на виході з хоан), дистальний край вимірювальної трубки якого розташований у ротовій порожнині, можуть бути відмінними від нуля при герметичному відділенні порожнини рота від носоглотки структурами м'якого піднебіння (див. рис. 3.3, б і 3.4) під час затримки дихання і складати близько 100 Па. Цей показник може мати діагностичну значущість у ході вивчення ступеня рухливості м'якого піднебіння, наприклад, під час лікування хропіння і синдрому обструктивного апное сну [195].

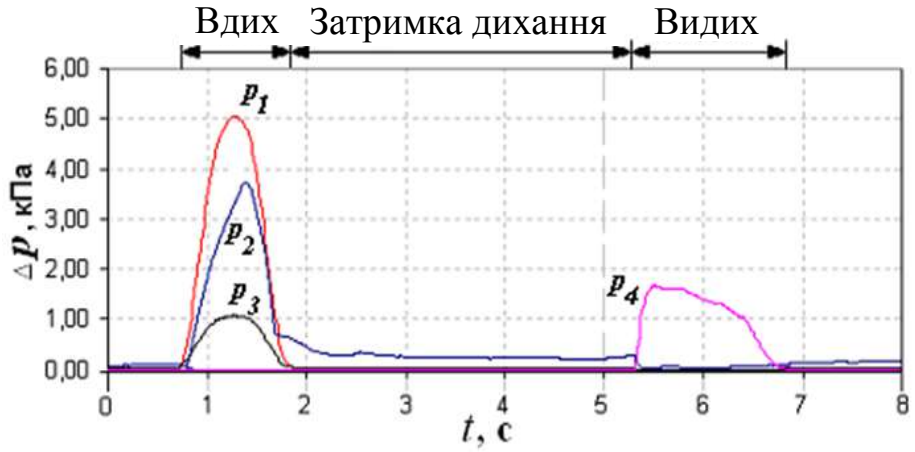
Діагностичним показником може слугувати також часове зрушення Δt (рис. 3.5) між максимумами сигналів перепаду тиску на носових проходах і витрати повітря на вдиху (визначається за показниками датчика), що має фізичний сенс різниці фаз згідно з формулою (2.25), і за значенням якого можна судити про дисипації енергії під час проходження повітря через верхні дихальні шляхи.

Як видно з графіка на рисунку 3.5, часове зрушення Δt між максимумами сигналів перепаду тиску на носових проходах і перепаду тиску у витратомірі дорівнює 0,05 с, що відповідає фазовому зрушенню між сигналами $\delta = 9^\circ$. Проте визначення статистичної значущості наведених вище показників під час

діагностики захворювань верхніх дихальних шляхів вимагає подальшого вивчення і медичного обґрунтування. Діаграми декількох дихальних циклів за даними динамічної ЗАРМ наведено на рис. 3.6.



а



б

Рис. 3.3. Діаграми дихальних циклів за даними динамічної ЗАРМ при сполученні (а) і герметичному відділенні (б) порожнини рота від носоглотки структурами м’якого піднебіння

Рис. 3.4. Статичне зображення кадру високошвидкісної динамічної рентгенографії носоглотки в сагітальній проекції; позначення:
1 – дихальні шляхи носоглотки; 2 – задня стінка глотки; 3 – м’яке піднебіння

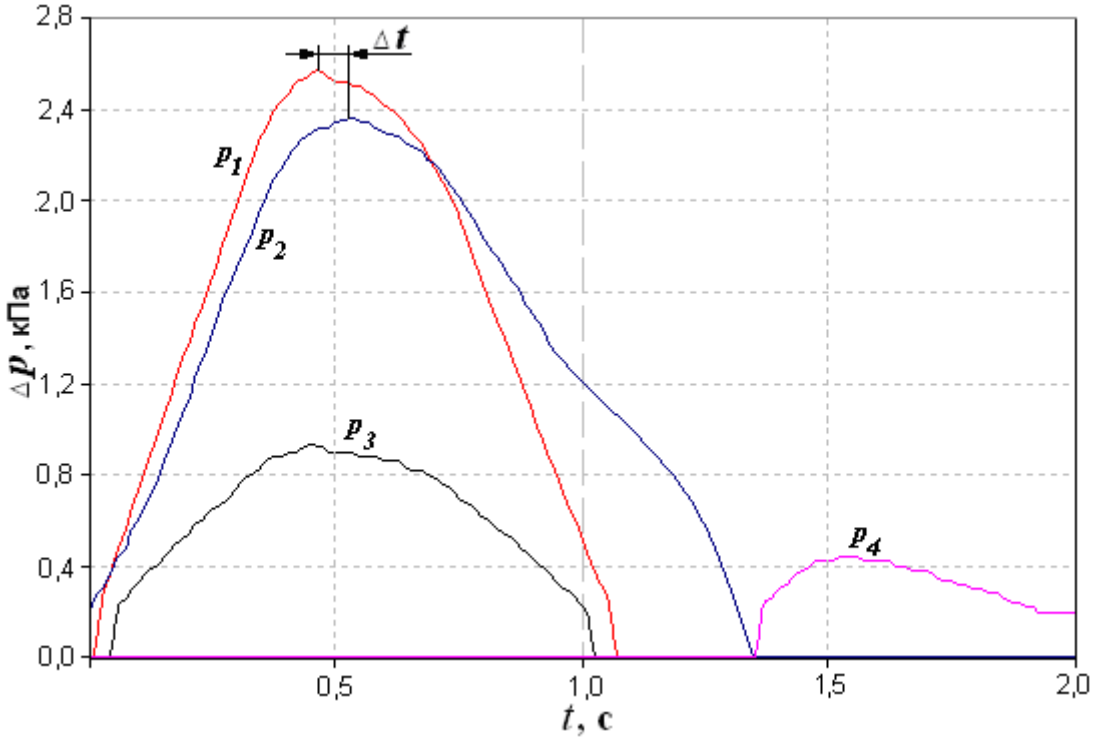


Рис. 3.5. Діаграма дихального циклу, що показує часове зрушення Δt між амплітудами сигналів датчиків тиску p_1 і p_2 за даними динамічної ЗАРМ

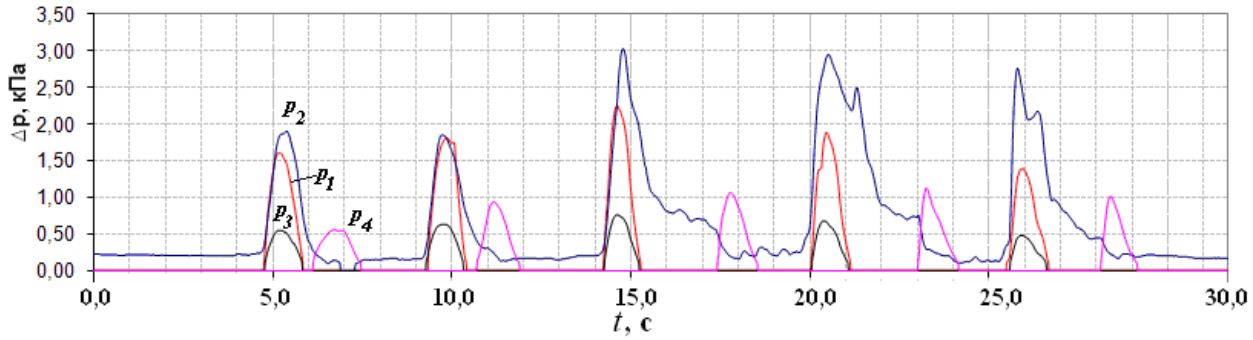


Рис. 3.6. Діаграма дихальних циклів за даними ЗАРМ

Результуючим показником активної риноманометрії, як задньої, так і передньої, є величина аеродинамічного опору носових проходів, що визначається за формулою (3.1) як відношення амплітудних значень перепаду тиску на носових проходах до витрати повітря, яке пропускається, і усереднена (в загальному випадку) за кількістю n дихальних циклів

$$\overline{A} = \sum_{i=1}^{n} \frac{\Delta p_{\max}^{(i)}}{Q_{\max}^{(i)}} \ , \qquad (3.8)$$

де $\Delta p_{\max}^{(i)}$ і $Q_{\max}^{(i)}$ – максимальні значення перепаду тиску і витрати повітря через носові проходи під час і-го дихального циклу.

Далі будується графік, по осі абсцис якого відкладаються значення витрати повітря Q, а по осі ординат значення перепаду тисків Δp на носових проходах (рис. 3.7). При цьому величина аеродинамічного опору носових

проходів набуває сенсу тангенса кута нахилу прямої, що з'єднує точку з координатами $(Q, \Delta p)$ з початком координат (0,0). За значенням цього показника (кута нахилу прямої) і здійснюється стандартний діагностичний висновок про величину опору носового дихання. На величину аеродинамічного опору носових проходів істотно впливає фаза носового циклу [190–195], що призводить до труднощів досягнення повторюваності результатів під час обстеження одного й того самого пацієнта з інтервалом в декілька десятків хвилин, і знижує діагностичну достовірність під час дослідження носової провідності, а також фізичний стан і вік пацієнта. Тому актуальною є задача забезпечення повторюваності результатів риноманометричної діагностики. Крім того, в більшості випадків, суб'єктивне відчуття порушення носової провідності виникає в ході виконання фізичного навантаження і, як наслідок, прискореного глибокого, тобто форсованого дихання.

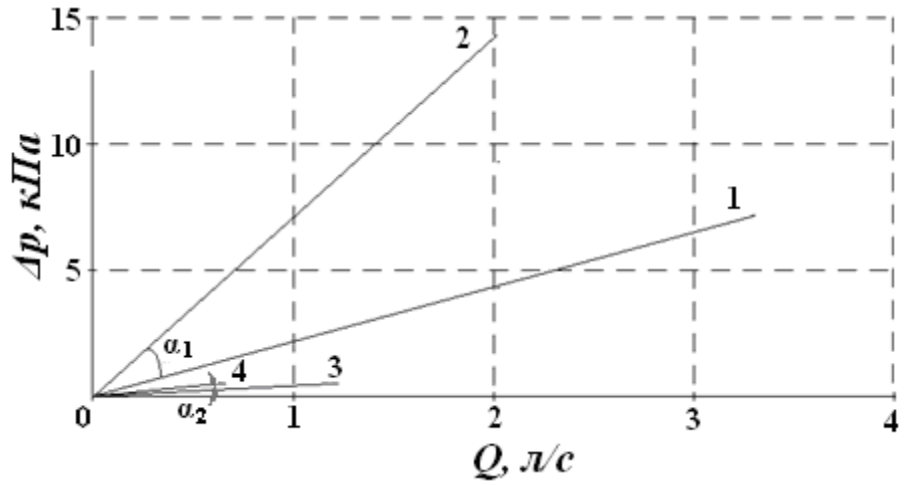


Рис. 3.7. Залежність перепаду тиску на носових проходах від витрати повітря, що пропускається за даними форсованої ЗАРМ (1 – відповідає умовній нормі, 2 – відхилення від норми) і традиційної ПАРМ (3 – відповідає умовній нормі, 4 – відхилення від норми)

Тому пропонується спосіб динамічної ПАРМ під час форсованого дихання, який дозволяє максимально враховувати компенсаторні можливості організму людини, пов'язані з тим, що необхідну витрату повітря можна отримати шляхом короткочасного створення більшого перепаду тиску на носових проходах за рахунок напруги дихальних м'язів діафрагми або досягнення однакових показників аеродинамічного опору носових проходів (механічної потужності дихання) за різних значень перепаду тиску і витрати повітря.

Стандартний метод ПАРМ [190, 230] передбачає вимір носового опору під час спокійного дихання і за фіксованого значення перепаду тиску в 300 Па.

Порівняльний аналіз коефіцієнтів аеродинамічного опору носових проходів для двох пацієнтів з нормальним носовим диханням і порушенням, викликаним викривленням носової перегородки (рис. 3.7) в ході вимірювання методом ЗАРМ під час форсованого дихання і традиційним методом ПАРМ

показує, що відмінність величин коефіцієнтів аеродинамічного опору носових проходів становить для методу форсованої ЗАРМ 2,1 (7 кПа/(л/с) / 3,3 кПа/(л/с), прямі 2 і 1), і 1,6 (0,5 кПа/(л/с) / 0,3 кПа/(л/с), прямі 4 і 3) для методу ПАРМ, що наочно видно за різницею кутів α1 і α2 між відрізками прямих 1, 2 і 3, 4, відповідно. Прямі на рис. 3.7 з'єднують початок координат і пікові значення досягнутих у процесі діагностики значень перепаду і витрати повітря. Дискримінантні можливості [230 – 234] методів форсованої ЗАРМ і форсованої динамічної ЗАРМ порівняно зі стандартним методом ПАРМ будуть розглянуті в розділі 5.

3.3 Розробка методу оцінки функціонування носового клапана

Під носовим клапаном розуміють простір між каудальним краєм верхнього латерального хряща і перегородкою носа [190–193]. Основною функцією носового клапана є його здатність обмежувати витрату повітря, що пропускається шляхом динамічного дроселювання. Цей ефект досягається за рахунок рухливості його зовнішніх анатомічних структур. Під час проведення комп'ютерного планування функціональних ринохірургічних втручань необхідно визначити кількісні критерії оцінки функціонування носового клапана з регулювання повітряного потоку.

Інструментально визначити ступінь рухливості носового клапана можна за даними електроміографії. Методика дослідження полягає в закріпленні електродів на крилах носа обстежуваного, реєстрації електроміографічного сигналу в циклах форсованого дихання (коли проявляється рухливість анатомічних структур клапана) і подальшому аналізі отриманих електроміограм. Виміри проводилися за допомогою нашкірних електродів стандартного електроміографа типу Нейро-МВП-4 за двома відведеннями.

Аналіз електроміографічних сигналів виконувався в частотній області шляхом побудови періодограм, які становлять оцінки спектральної щільності потужності, отримані за N відліків однієї реалізації випадкового процесу згідно з

$$\hat{W}(\omega) = \frac{1}{N f_{\text{Д}}} \left| \sum_{k=0}^{N-1} x(k) e^{-j\omega kT} \right|^2 , \qquad (3.9)$$

де N – кількість відліків електроміографічного сигналу;

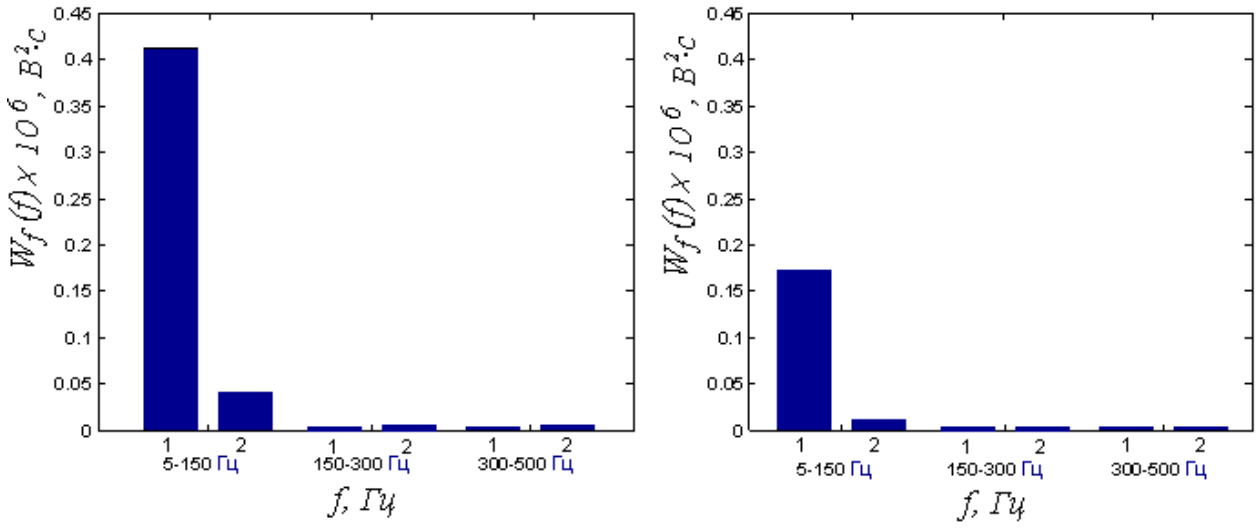
$f_{\text{Д}}$ – частота дискретизації;

$x(k)$ – відліки електроміографічного сигналу;

T – інтервал дискретизації.

Переходячи від циклічної частоти до частоти сигналу у формулі (3.9), на розрахованих періодограмах виділялися три частотні діапазони: низькі частоти (5...150 Гц), середні частоти (150...300 Гц) і високі частоти (300...500 Гц). Далі розраховувався сумарний внесок періодограми в кожен з частотних діапазонів, і проводилася побудова діаграм співвідношень спектральних потужностей за частотними діапазонами.

На рис. 3.8 наведено типові діаграми співвідношень спектральних потужностей за частотними діапазонами для лівої (рис. 3.8, а) і правої (рис. 3.8, б) груп м’язів носового клапана в нормі (1) і з порушенням рухливості клапана в зв’язку з вираженою деформацією зовнішнього носа (2). При цьому очевидно, що найбільша відмінність проявляється в нижньому частотному діапазоні (5 ... 150 Гц).



а б

Рис. 3.8. Діаграми співвідношень спектральних потужностей за частотними діапазонами для лівої (а) і правої (б) груп м’язів носового клапана в нормі (1) і з порушенням рухливості носового клапана (2)

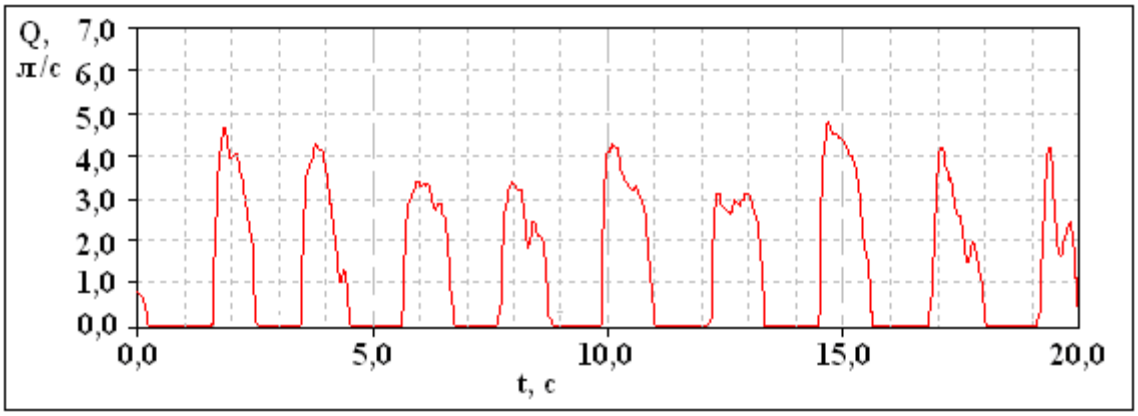
Для обстежуваного з вираженою деформацією зовнішнього носа відзначено значне зниження (на порядок) амплітуди стовпців діаграми порівняно зі значенням обстежуваних без порушення рухливості клапана. Також виявлено у більшості обстежуваних як у нормі, так і з порушенням функції носового клапана, асиметричність у рухливості правого і лівого клапанів, яка виражається в зміні амплітуди стовпців діаграми для правого і лівого клапанів (приблизно вдвічі), що обумовлено анатомічною варіабельністю. Результати були отримані за даними обстеження 25 пацієнтів з порушенням функціонування носового клапана (20 пацієнтів становили контрольну групу).

Проте розглянутий метод важко використовувати в практичній ринології через складність отримання електроміографічного сигналу без артефактів від руху.

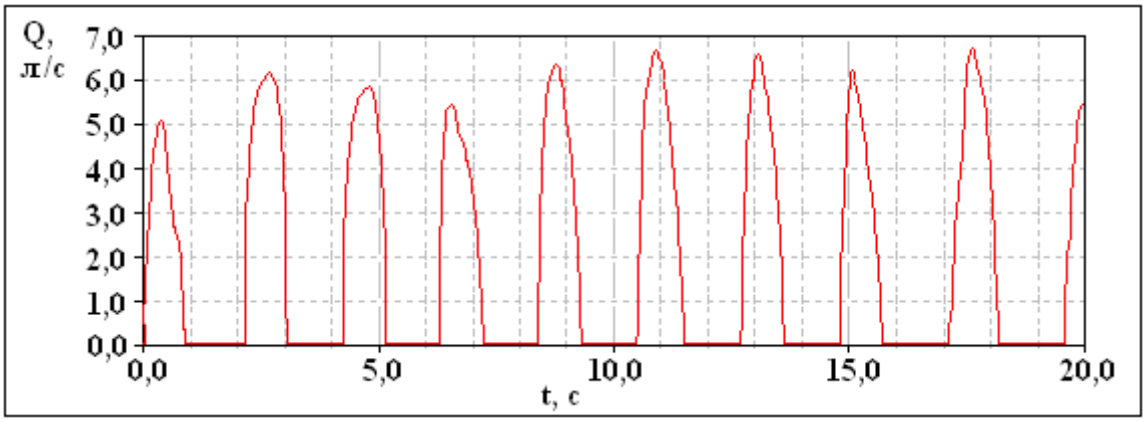
Крім того, оцінка рухливості носового клапана тільки опосередковано дозволяє судити про його функції регулювання повітряного потоку при вході в носову порожнину. Тому доцільно запропонувати метод, заснований на безпосередньому динамічному вимірі значень витрати повітря за форсованого дихання й аналізу отриманих даних.

Запропонований метод заснований на оцінці форми сигналу витрати повітря у дихальних циклах за даними форсованої динамічної ЗАРМ. У нормі сигнал витрати повітря у дихальних циклах має періодичну форму з явно

вираженими максимумами. Причому під час форсованого дихання під дією розрядження всередині носової порожнини рухливі крила носа (зовнішні анатомічні структури носового клапана, що утворюють латеральні стінки носової порожнини при вході в носові проходи) зсуваються в медіальному напрямку і, за рахунок зменшення площі живого перетину носових проходів, створюють додатковий аеродинамічний опір (у граничному випадку до повної обструкції носового ходу), перешкоджаючи збільшенню витрати повітря 230]. Відповідна форма сигналу витрати повітря в циклах вдиху наведена на рис. 3.9, а. При цьому добре видно дещо сплющену форму сигналів витрати повітря з вираженими локальними максимумами, що свідчать про обмеження витрати, що пропускається. Відсутність рухливості структур носового клапана призводить до діаграми сигналу витрати повітря в циклах вдиху, наведеної на рис. 3.9, б. При цьому форма сигналу витрати повітря близька до ідеальної синусоїди. При штучному знерухомленні структур носового клапана спостерігається також зростання (до 30%) пікових значень величини витрати повітря, що сприяє створенню підвищених швидкостей повітряного потоку в носовій порожнині і пов'язаної з цим хронічної травматизації слизової оболонки верхніх дихальних шляхів [190].



а



б

Рис. 3.9. Діаграми дихальних циклів: а – при нормальному функціонуванні носового клапана; б – при порушенні рухливості носового клапана (вимірювання виконуються тільки в циклі вдиху)

Автоматизація методу полягає в програмній фіксації сплощеної форми і локальних екстремумів сигналу витрати повітря шляхом чисельного диференціювання сигналу витрати повітря за часом і аналізом кількості нульових значень похідної під час фази вдиху дихального циклу (за позитивного значення сигналу витрати повітря). Чисельне диференціювання вимірюваного сигналу витрати повітря здійснюється згідно з формулою

$$Q'(i)=\frac{\Delta Q}{\Delta t}=\frac{Q(i)-Q(i-1)}{t(i)-t(i-1)}\ , \qquad (3.10)$$

де $Q(i)$ – дискретно задані *i*-ті значення сигналу витрати повітря в моменти часу $t(i)$, $i\in[1;n]$;

n – кількість відліків сигналу витрати повітря;

Δt – часовий інтервал між відліками сигналу.

Приклад сигналу витрати повітря у двох дихальних циклах з відповідними фазами вдиху за даними форсованої динамічної ЗАРМ наведено на рис. 3.10, а (відповідний графік похідної від витрати повітря за часом наведено на рис. 3.10, б).

Чисельним показником наявності обмеження витрати повітря є $F(k)$ бінарна характеристична функція сплощеної вершини сигналу витрати повітря в k-му циклі дихання

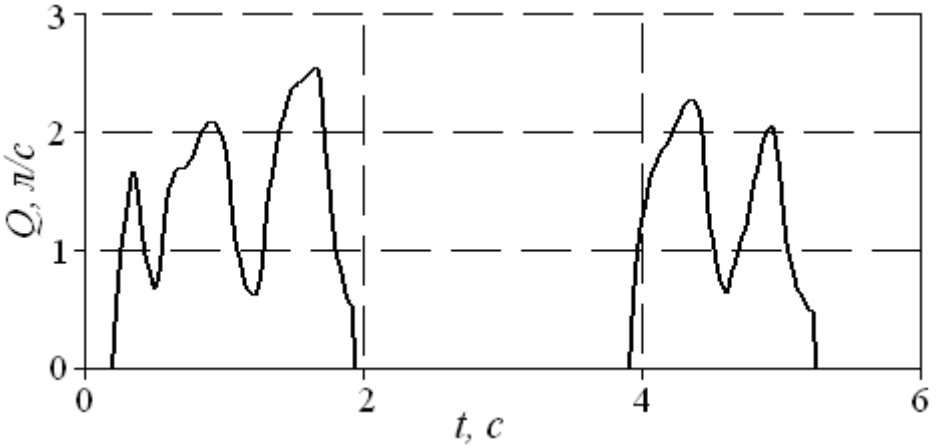
$$F(k)=\begin{cases}1;\left(Q\left(i_{\tau_k}\right)>0{,}1\ л/с\right)\&\left(m\left(Q'\left(i_{\tau_k}\right)=0\right)\right)>2)\&\left(\tau_k<2\ с\right);\\ 0;\text{ інакше,}\end{cases}$$

де $\tau_k=t_{E_k}-\mathrm{t}_{S_k}$ – часовий інтервал циклу вдиху, який визначається як різниця кінцевого t_{E_k} і t_{S_k} початкового моментів часу фази вдиху, що визначаються після завершення і початку зростання значень сигналу витрати повітря відносно нульового рівня;

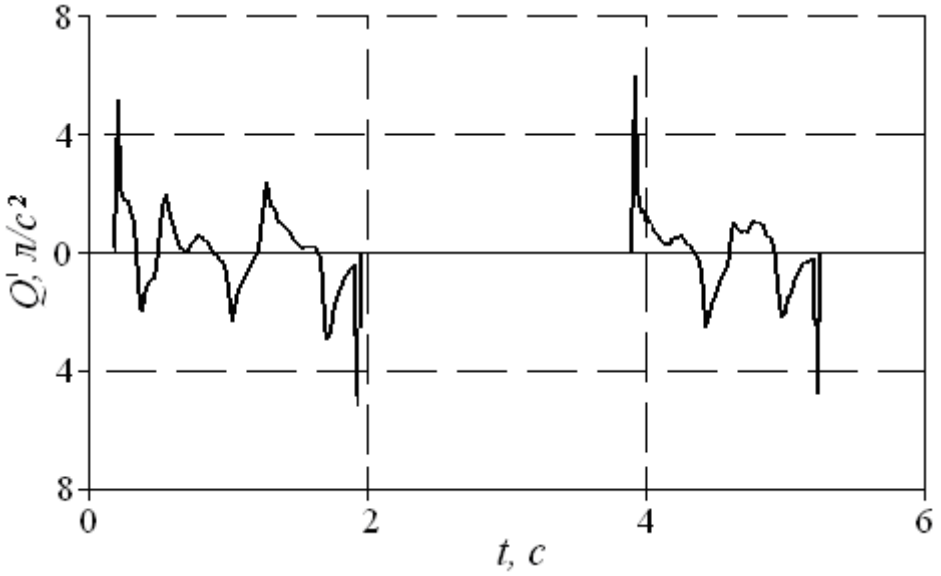
$m(Q'(i))$ – функція-лічильник кількості нульових значень похідної сигналу витрати повітря за часом $Q'(i)$, що визначається як

$$m(Q'(i))=\begin{cases}m(Q'(i))+1;\ Q'(i)=0;\\ m(Q'(i));\ Q'(i)\neq 0.\end{cases} \qquad (3.12)$$

Таким чином, згідно з формулою (3.11), ознакою сплощення вершин сигналу витрати повітря є наявність більше двох локальних екстремумів протягом циклу вдиху тривалістю не більше двох секунд і величиною витрати більше 0,1 л/с. Наявність таких ознак у більш ніж трьох з десяти циклів форсованого дихання за даними проведеної клінічної апробації свідчить про нормальне функціонування носового клапана [190]. Однак, зважаючи на природу процесу дихання, доцільно було б провести інтелектуальний аналіз риноманометричних сигналів у динаміці для визначення діагностично-вагомих показників.



а



б

Рис. 3.10. Графіки сигналу витрати повітря в двох дихальних циклах при вдиху за даними форсованої динамічної ЗАРМ (а) і похідною від витрати повітря за часом (б)

Розроблений метод задньої риноманометрії дозволяє за даними перетворювача перепаду тиску між носоглоткою і підмасковим простором і сопла Вентурі, що вимірює відповідну витрату повітря, визначати величину аеродинамічного носового опору під час фізіологічного дихання як у стані спокою, так і у форсованому режимах.

Проведено оцінку різниці фаз між сигналами витрати повітря і перепаду тиску на носовій порожнині, яка становить близько 9°, що дозволяє за даними динамічної риноманометрії побічно враховувати стан слизової оболонки верхніх дихальних шляхів.

Розроблено метод оцінки функціонального стану носового клапана, який дозволяє за рахунок аналізу форми сигналу витрати повітря (наявності високочастотних складових і сплощеної вершини) визначати ступінь рухливості структур носового клапана і його функціональну роль в обмеженні потоку повітря через верхні дихальні шляхи під час форсованого дихання.

4 ШТУЧНІ НЕЙРОННІ МЕРЕЖІ В ЗАДАЧАХ ДІАГНОСТИКИ ТА ПРИЙНЯТТЯ РІШЕНЬ

З формальної точки зору завдання медичної діагностики може розглядатися як проблема прийняття рішень в умовах невизначеності. Особливостями медичної діагностики є те, що якість рішень, що приймаються, значною мірою, визначаються досвідом лікаря-діагноста, який накопичується навчанням та самонавчанням протягом тривалого періоду. Формальним математичним аналогом біологічного мозку є штучні нейронні мережі, теорія яких (нейроматематика) і, передусім, апарат глибинних нейронних мереж інтенсивно розвивається у цей час [235–238].

З позицій нейроматематики процес навчання розглядається як адаптація параметрів, а можливо й архітектури мережі для вирішення поставленої задачі шляхом оптимізації прийнятого критерію якості. Таке формулювання є загальноприйнятим і неявно припускає, що в основі нейроматематики лежать методи оптимізації й ідентифікації.

В цілому прийнято, що процес навчання має перманентний характер і з часом мережа поліпшує свої характеристики, поступово «наближаючись» до оптимального вирішення поставленої задачі.

Тип і характер навчання визначаються насамперед обсягом апріорної і поточної інформації про середовище, в яке «занурена» мережа, а також критерієм якості (цільовою функцією), що характеризує ступінь відповідності нейромережі розв'язуваній нею задачі. Інформація про зовнішнє середовище задана, як правило, у вигляді навчальної вибірки образів або прикладів, обробляючи яку мережа накопичує дані, необхідні для отримання шуканого вирішення. Саме характер і обсяг цієї інформації визначають як тип навчання, так і конкретний алгоритм.

4.1 Основні парадигми, побудова, правила навчання штучних нейронних мереж

Найбільш популярною й очевидною на цей часу є парадигма навчання «із вчителем», що схематично подана на рис. 4.1.

У цій схемі «вчителю» відома інформація про зовнішнє середовище, задане у вигляді послідовності або пакета вхідних векторів, а також «правильна реакція» на ці сигнали, подана у вигляді навчального сигналу d. Звичайно, що реакція ненавченої мережі y відрізняється від «правильної» реакції вчителя, внаслідок чого виникає похибка $e = d - y$. У процесі навчання необхідно так налаштувати параметри ШНМ, щоб деяка скалярна функція від помилки $E(e)$ (критерій якості) досягла б свого мінімального значення. Навченою вважається мережа, яка у деякому статистичному сенсі повторює реакцію вчителя. Оскільки інформація про зовнішнє середовище зазвичай має нестаціонарний

характер, процес навчання йде беззупинно, для чого використовуються ті чи інші рекурентні процедури.

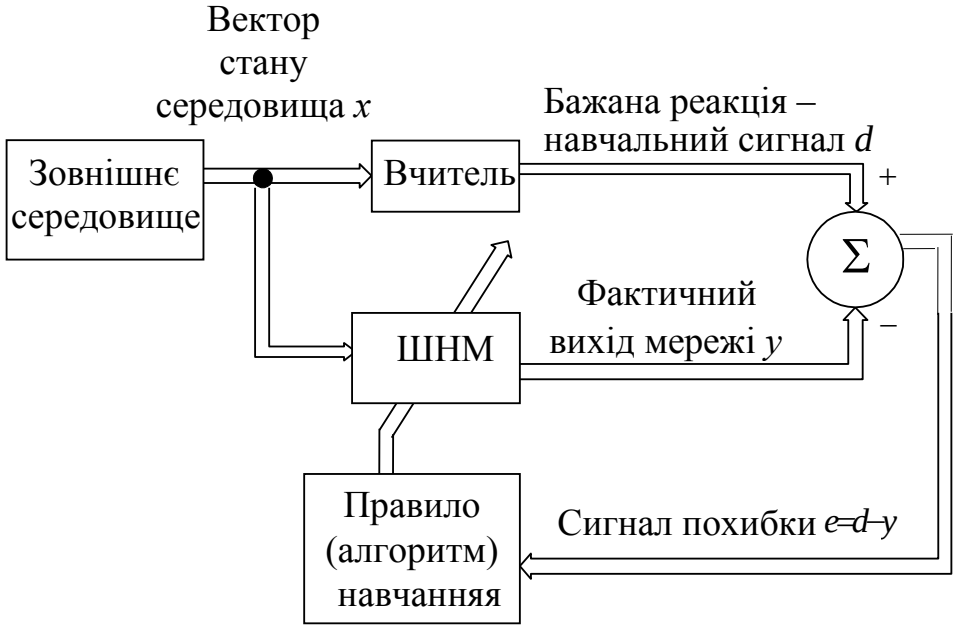


Рис. 4.1. Схема навчання з вчителем

Альтернативою цій парадигмі є навчання «без вчителя», або самонавчання, коли правильна реакція на сигнали зовнішнього середовища невідома. Процес самонавчання схематично подано на рис. 4.2.

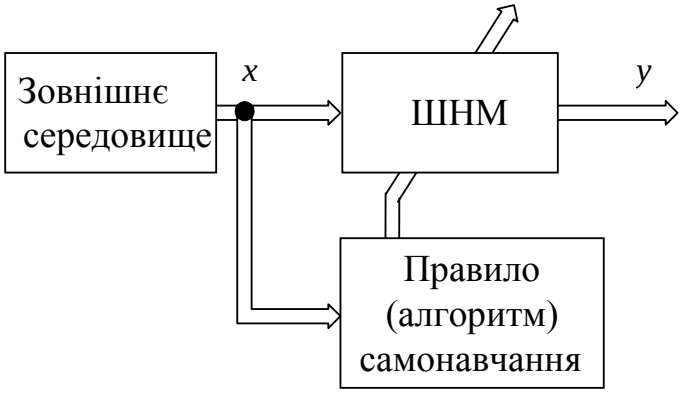


Рис. 4.2. Схема самонавчання

Мережі, що реалізують парадигму самонавчання, призначені, як правило, для аналізу внутрішньої латентної структури вхідної інформації і вирішують задачі автоматичної класифікації, кластеризації, факторного аналізу, компресії даних.

Своєрідним компромісом між двома цими парадигмами є навчання з підкріпленням [235] (не плутати з навчанням із заохоченням [239]), при якому доступна лише непряма інформація про правильну реакцію на вхідний сигнал x. На рис. 4.3 наведено схему процесу навчання з підкріпленням.

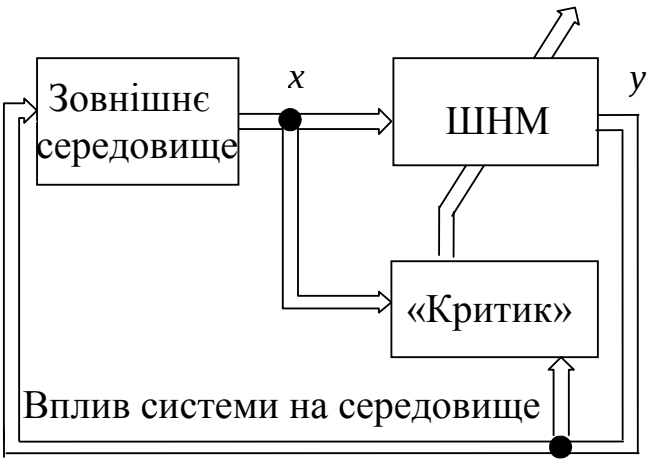


Рис. 4.3. Схема навчання з підкріпленням

Нейронна мережа робить відображення вхідної інформації x у вихідний вектор y у вигляді $y = F(x)$, проте, оскільки навчальний сигнал d у явному вигляді не заданий, неможливо отримати похибку $e = d - y$, на підставі якої відбувається навчання. Передбачається, що є деякі апріорні знання, які дозволяють зв'язати евристичний сигнал підкріплення $\tilde{d}$ з неспостереженим бажаним виходом d за допомогою деякої функції $\tilde{F}$, що відображує d в $\tilde{d}$. Зазвичай ця функція враховує зв'язок вихідних сигналів мережі y з подіями, що спостерігаються у зовнішньому середовищі, для чого в схему навчання вводиться додатковий блок-«критик» [240], що відображує поведінку мережі в сигнал $\tilde{y} = \tilde{F}\left(F(x)\right)$. Далі обчислюється евристична похибка $\tilde{e} = \tilde{d} - \tilde{y}$, на основі якої і реалізується процес навчання.

Процес навчання з підкріпленням розбивається на два відносно незалежних етапи: навчання тому, як вихідний сигнал мережі y впливає на змінні середовища x, тобто відновлення відображення $\tilde{F}$, і власне навчання мережі на основі мінімізації прийнятого критерію $E(\tilde{e})$.

Ця парадигма тісно пов'язана з ідеями динамічного програмування [241] і в теорії штучних нейронних мереж відома також як нейродинамічне програмування [240].

Досить широкого поширення набула також парадигма змішаного навчання, коли частина параметрів мережі настроюється за допомогою навчання із вчителем, а інша частина – за допомогою самонавчання. Цей підхід отримав найбільшого поширення в ході навчання радіально-базисних ШНМ.

З введеними парадигмами тісно пов'язані правила навчання, які лежать в основі конкретних алгоритмів. С. Хайкін [240] відзначає п'ять основних правил: навчання на основі корекції за помилками, навчання за Больцманом, навчання за Геббом, навчання пам'яті і конкурентне навчання.

Правило корекції за помилками – типовий випадок навчання із вчителем, при цьому за допомогою тих або інших процедур оптимізації й адаптивної

ідентифікації мінімізується апріорі задана скалярна цільова функція $E(e)$. З цим правилом пов'язане найбільше число відомих алгоритмів навчання, що до цього часу перевищує сотню.

В основу навчання за Больцманом покладені принципи теоретичної термодинаміки, при цьому настроювання синаптичних ваг стохастичної мережі забезпечує необхідний (бажаний) розподіл імовірностей станів окремих нейронів. Певною мірою навчання за Больцманом може розглядатися як поширення ідей навчання з учителем на стохастичний випадок.

Із самонавчанням тісно пов'язане правило Гебба і навчання пам'яті, в основу яких покладено нейрофізіологічний постулат, який говорить про те, що, якщо нейрони по обидва боки синапса знаходяться в збудженому стані, то сила зв'язку між ними зростає (збільшується синаптична вага) і, навпаки, якщо сусідні нейрони знаходяться в різних станах, то зв'язок між ними слабшає.

У конкурентному навчанні можуть бути реалізовані всі описані парадигми, при цьому його відмінною рисою є процес «змагання» нейронів вихідного шару за принципом «Winner Takes All», тобто збуджується тільки один вихідний нейрон – «переможець». Найбільш яскравими прикладами мереж, що використовують це правило, є мережі адаптивного резонансу (ART) і самоорганізувальні мапи (SOM).

4.2 Задачі навчання

У цьому підрозділі розглянуто деякі прикладні інженерні задачі, розв'язувані нейромережами, навченими відповідним чином. Природно, що цей список далеко не повний і включає тільки проблеми, які традиційно цікавлять фахівців в області комп'ютерних наук, інженерії і керування.

Розпізнавання образів. Поряд з навчанням, розпізнавання є однією з основних функцій біологічного мозку. Отримуючи дані з навколишнього світу за допомогою біологічних сенсорів, мозок досить просто розпізнає джерело даних і виділяє з нього необхідну інформацію. Так людина без особливих проблем розпізнає знайоме обличчя, хоча бачила його давно і воно встигло змінитися, голос, перекручений телефонними перешкодами, місто, в якому не була багато років. Це є результат навчання, причому в ідеальному випадку нейромережа має розпізнавати пропоновані їй образи не гірше, ніж це робить живий організм.

Формально розпізнавання образів визначається як процес, внаслідок якого отримуваний образ (сигнал) належить до одного з апріорі призначених класів (категорій) [242–246]. У процесі навчання нейромережі пред'являються різні образи з відомою класифікацією (навчальна вибірка), а внаслідок мережа має розпізнати об'єкт, що раніше не пред'являвся, але який належить до тієї самої сукупності, що і навчальна вибірка. Задача розпізнавання є статистичною за своєю природою, при цьому образи представляються випадковими векторами в багатовимірному просторі ознак, а результат навчання полягає в

побудові вирішальних гіперповерхонь, що розділяють «у середньому» простір ознак на відповідні класи.

Як правило, нейромережні системи, що розпізнають, складаються з двох частин. Перша – це самонавчальна мережа, що вирішує задачу селекції й вибору ознак, а друга – це мережа, яка настроюється за допомогою зовнішнього навчального сигналу, що містить інформацію про приналежність образів навчальної вибірки визначеним класам. Загалом така послідовність вирішення задачі є характерною для більшості систем, що розпізнають: спочатку зниження розмірності вектора ознак за допомогою, наприклад, традиційного перетворення Карунена-Лоева, а потім власне побудова гіперповерхонь, що розділяють. Перевага нейромережного підходу перед іншими методами розпізнавання образів полягає в тому, що нейромережі здатні відновлювати гіперповерхні, які розділяють, будь-якої складної форми, не спираючись на гіпотези про компактність або лінійну подільність класів.

Асоціація і кластеризація. Для біологічних систем поряд з навчанням і розпізнаванням характерною є також здатність до асоціацій, тобто відновлення (спогад) раніше пред'явлених образів за деякими непрямими стимулами. Будь-якій людині знайомі випадки, коли певний випадковий звук чи запах викликав в уяві складні зорові образи.

У нейромережах асоціації реалізуються в двох формах: автоасоціація і гетероасоціація. У випадку автоасоціації мережа обробляє множину послідовно пропонованих їй образів, причому ці образи можуть бути зашумлені або перекручені. Розглядаючи і запам'ятовуючи основні ознаки пропонованих образів, мережа здобуває здатність відновлювати (згадувати) раніше показані їй приклади. Гетероасоціація відрізняється тим, що довільна множина вхідних образів зв'язується (асоціюється) з довільною множиною вихідних прикладів. Основна відмінність між цими формами полягає в тому, що автоасоціація реалізується на основі парадигми самонавчання, а гетероасоціація – навчання із вчителем. Нехай $x(k)$ – вхідний образ-вектор (стимул), у загальному випадку довільно взятий з навчальної вибірки і пред'явлений мережі асоціативної пам'яті, а $y(k)$ – запам'ятований (вихідний) образ-вектор. Асоціація образів, виконувана мережею, описується відношенням $x(k) \to y(k), k = 1,2,\ldots,N$, де N – кількість образів, запам'ятована ШНМ. Вхідний образ $x(k)$ діє як стимул, що викликає відгук $y(k)$, а відтак є ключем до відновлення.

В автоасоціативній пам'яті $y(k) = x(k)$, тобто вхідний і вихідний простір мережі збігаються. У гетероасоціативній пам'яті $y(k) \neq x(k)$, при цьому розмірності просторів, як правило, також не збігаються.

У роботі асоціативних нейромереж виділяють дві фази: накопичення, що відповідає періоду навчання, і відновлення, що викликає спогад запам'ятованого образа після пред'явлення зашумленого чи перекрученого стимула.

Кількість N образів, накопичених в асоціативній пам'яті, є мірою ємності мережі. В процесі проектування таких мереж основною проблемою є вибір і

забезпечення максимальної ємності, вираженої як відношення кількості прикладів, що запам'ятовуються, N до загальної кількості нейронів мережі за мінімальної кількості некоректно відновлюваних образів.

До проблеми автоасоціації тісно примикає задача кластеризації (автоматичної класифікації), коли мережа, аналізуючи навчальну вибірку $x(k)$, розміщує «схожі» образи за групами-кластерами. Пропонований зашумлений образ, раніше не показаний мережі, за асоціацією з вже запам'ятованими має бути віднесений до «рідного кластеру». Мережі, що реалізують кластеризацію образів, використовуються зазвичай для стиснення даних та вилучення з них знань.

Апроксимація функцій. З проблемою навчання тісно пов'язана досить часто виникаюча на практиці задача апроксимації функцій, заданих на деякій множині точок.

Розглянемо нелінійне відображення «вхід-вихід», що описується функціональним співвідношенням

$$d = f(x), \tag{4.1}$$

де d та x – $(m\times 1)$ і $(n\times 1)$ – вектори виходів і входів відповідно, $f(\bullet)$ – невідома вектор-функція, яку необхідно оцінити за допомогою заданої навчальної вибірки $\{x(k), d(k)\}, k = 1,2,\ldots,N$.

Задача навчання апроксимуючої нейромережі полягає в знаходженні функції $F(x)$ у деякому сенсі досить близької до $f(x)$ так, що

$$\|F(x) - f(x)\| \le \varepsilon \quad \textit{для усіх } x(k), k = 1,2,\ldots,N, \tag{4.2}$$

де $F(x)$ – відображення, реалізоване мережею, ε – мале позитивне число.

Якщо обсяг вибірки N досить великий, а мережа має достатню кількість синаптичних ваг, помилка апроксимації ε може бути зроблена як завгодно малою, хоча тут є небезпека перетворення мережі з апроксимуючої в інтерполюючу.

Нескладно бачити, що проблема апроксимації в даному контексті цілком збігається з задачею навчання із вчителем, де послідовність $x(k)$ відіграє роль вхідного сигналу ШНМ, а $d(k)$ – навчального сигналу.

Здатність нейромереж апроксимувати невідомі відображення «вхід-вихід» знаходить два найважливіших застосування в задачах інтелектуального керування [247, 248, 249]. Перше з них – ідентифікація об'єктів керування [250, 251, 241], чи емуляція – у термінах нейрокерування [252].

Схему системи ідентифікації (емуляції) наведено на рис. 4.4, при цьому передбачається, що багатовимірний статичний об'єкт описується співвідношенням (4.1), а нейронна мережа, підключена паралельно до об'єкта, навчається в реальному часі, «підганяючи» свої вихідні сигнали до виходів реального об'єкта.

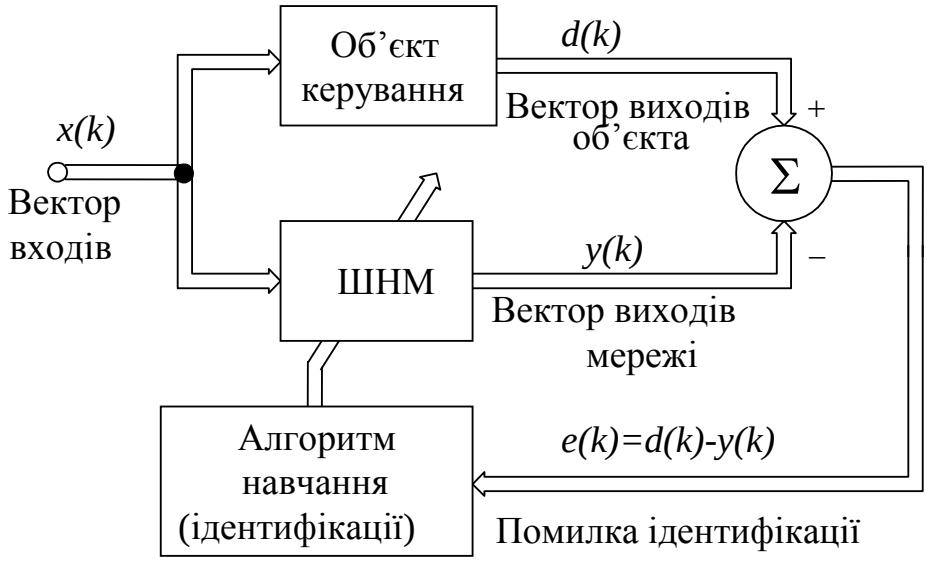


Рис. 4.4. Схема системи ідентифікації

Друге застосування – це зворотне моделювання, використовуване у деяких адаптивних системах керування [253–256] і суть його полягає в тому, що для об'єкта керування (4.1) потрібно побудувати «зворотну систему», що генерує вектор $x(k)$ як відгук на вхідний сигнал $d(k)$. У загальному вигляді зворотна система має форму

$$x = f^{-1}(d), \tag{4.3}$$

проте, оскільки функція f або невідома, або занадто складна, розумним виходом є використання ШНМ як зворотної моделі так, як це показано на рис. 4.5.

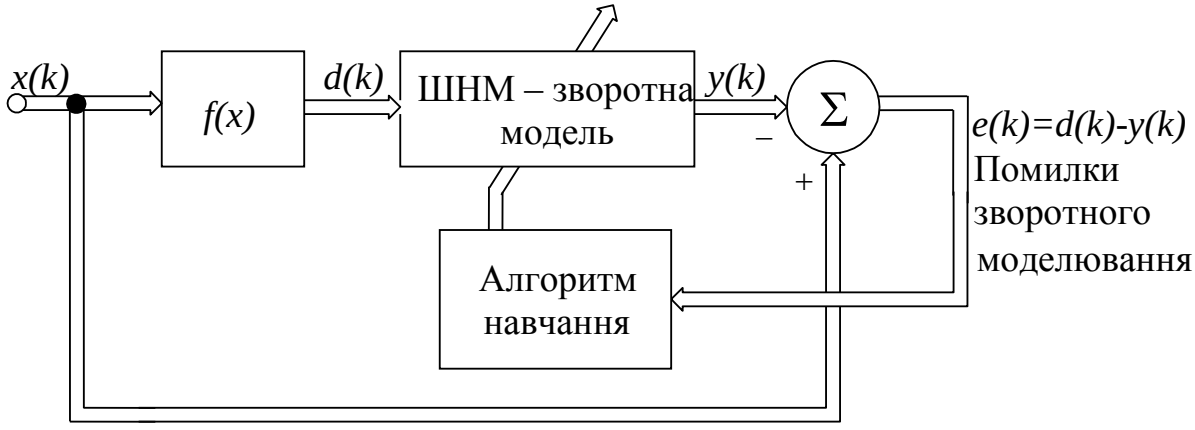


Рис. 4.5. Схема зворотного моделювання

У цій схемі ролі сигналів $x(k)$ і $d(k)$ помінялися: вектор $d(k)$ використовується як вхід мережі, а $x(k)$ – як бажаний відгук (навчальний сигнал). Подібно системі ідентифікації сигнал помилки $e(k) = x(k) - y(k)$ використовується для навчання ШНМ.

Керування й оптимізація. Керування об'єктами в умовах структурної і параметричної невизначеності – ще одна задача, пов'язана з навчанням нейромереж. На рис. 4.6 наведено схему керування зі зворотним зв'язком, при цьому передбачається, що в розпорядженні проектувальника системи керування немає інформації ні про структуру нелінійного об'єкта, ні, тим більше, про його параметри.

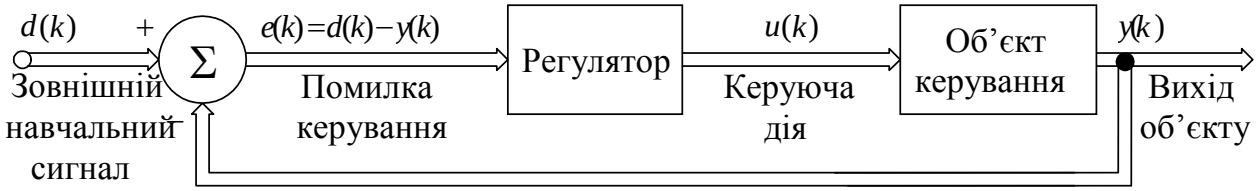


Рис. 4.6. Система керування зі зворотним зв'язком

Метою керування є вироблення керуючих сигналів $u(k)$, що забезпечують стійке слідкування виходом об'єкта $y(k)$ зовнішнього сигналу, що задає бажану траєкторію руху $d(k)$.

Оскільки про об'єкт керування нічого не відомо, як регулятор можна використовувати нейромережу, входом якої є вектор помилок керування $e(k)=d(k)-y(k)$, а виходом – сигнал керування $u(k)$, що подається на об'єкт.

Синтез оптимального керування пов'язаний з оцінкою якобіана об'єкта $J=\{\partial y_j/\partial u\}_i$ [257], для визначення якого знов-таки можуть бути використані апроксимуючі властивості ШНМ.

У теорії адаптивного керування сформувалося два основних напрямки. Перший – непрямий чи ідентифікаційний підхід, при якому в схему вводиться модель, що навчається в темпі з процесом керування, параметри якої в лінійному випадку є оцінками елементів матриці-якобіана. У нашому випадку в систему керування додатково вводиться нейромережа-емулятор, що оцінює частинні похідні $\partial y_j/\partial u_i$, які далі використовуються нейромережею-регулятором.

Альтернативою ідентифікаційному є прямий підхід до синтезу регулятора, при якому передбачається, що проектувальнику доступна інформація про знаки частинних похідних $\partial y_j/\partial u_i$. У прямій системі керування присутня одна нейромережа-регулятор, що навчається за допомогою алгоритмів, які використовують тільки знаки оброблюваних сигналів.

Безпосередньо до задачі керування примикає задача оптимізації, коли потрібно визначити екстремум багатовимірної неявно заданої функції за наявності обмежень. Хоча для вирішення проблеми оптимізації спроектовані спеціальні архітектури нейромереж [258], досить великий клас задач може бути вирішений у рамках систем нейрокерування [252], коли як цільова функція використовується лагранжіан, що враховує всі обмеження, які накладаються на змінні об'єкта.

Фільтрація, згладжування, прогнозування. Системи обробки «зашумлених» сигналів в умовах невизначеності в цей час знаходять широке застосування в найрізноманітніших галузях [259–262]. Власне поняття «обробка сигналів» традиційно містить три задачі: фільтрація, згладжування і прогнозування. Якщо в розпорядженні дослідника є вибірка «забруднених» спостережень $x(1), x(2), \ldots x(k), \ldots, x(N)$, то задача фільтрації зводиться до знаходження найкращої оцінки процесу $\hat{x}(N \mid N)$ у момент часу N за інформацією про N спостережень, згладжування – оцінки $\hat{x}(k \mid N)$ при $k < N$ і прогнозування – $\hat{x}(N + l \mid N)$ при $N + l > N$, де l – інтервал упередження.

Останніми роками увага дослідників спрямована ще до однієї нетрадиційної задачі обробки – «сліпої» сепарації та ідентифікації сигналів [263]. Передбачається, що є множина невідомих джерел сигналів $\{u_i(k)\}_{i=1}^{n}$, які не залежать один від одного. Сенсори сприймають ці сигнали не покомпонентно, а в суміші, що представляє собою невідому лінійну комбінацію $x(k) = Au(k)$ так, як це показано на рис. 4.7.

Задача зводиться до відновлення вектора $y(k) \approx u((k)$ за даними спостережень вектора $x(k)$ за невідомої $(n \times n)$-матриці A.

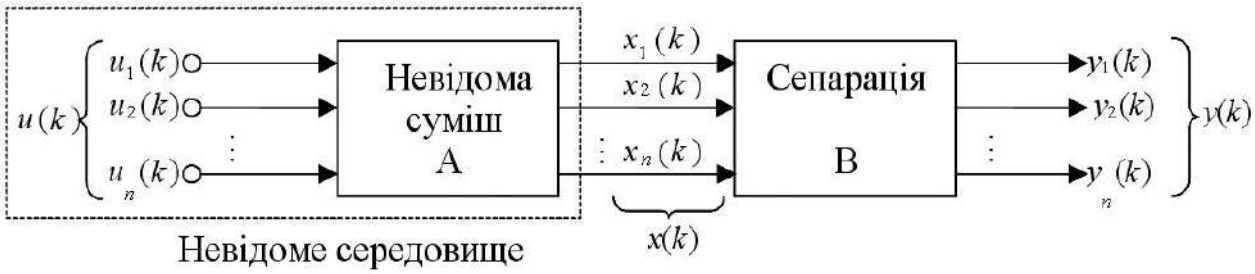


Рис. 4.7. Схема сліпої сепарації

Нескладно побачити, що перші три задачі дуже близькі до проблеми ідентифікації, а задача сліпої сепарації практично збігається з задачею зворотного моделювання і зводиться до знаходження оператора сепарації $B = A^{-1}$. Природно, що застосування ШНМ для вирішення цих задач принципових ускладнень не викликає.

Зупинимося коротко на задачі поточного прогнозування стохастичної послідовності $x(k)$ за даними про її передісторію $x(k-1), x(k-2), \ldots$. Проблема зводиться до побудови оцінки $\hat{x}(k) = F((x(k-1), x(k-2), \ldots, x(k-p))$ у реальному часі в темпі з надходженням даних. У лінійному випадку ця задача добре досліджена й успішно може бути вирішена за допомогою адаптивних прогнозуючих авторегресійних моделей [264]. Для побудови ж нелінійних прогнозів найбільш доцільним є застосування ШНМ, наприклад так, як це показано на рис. 4.8.

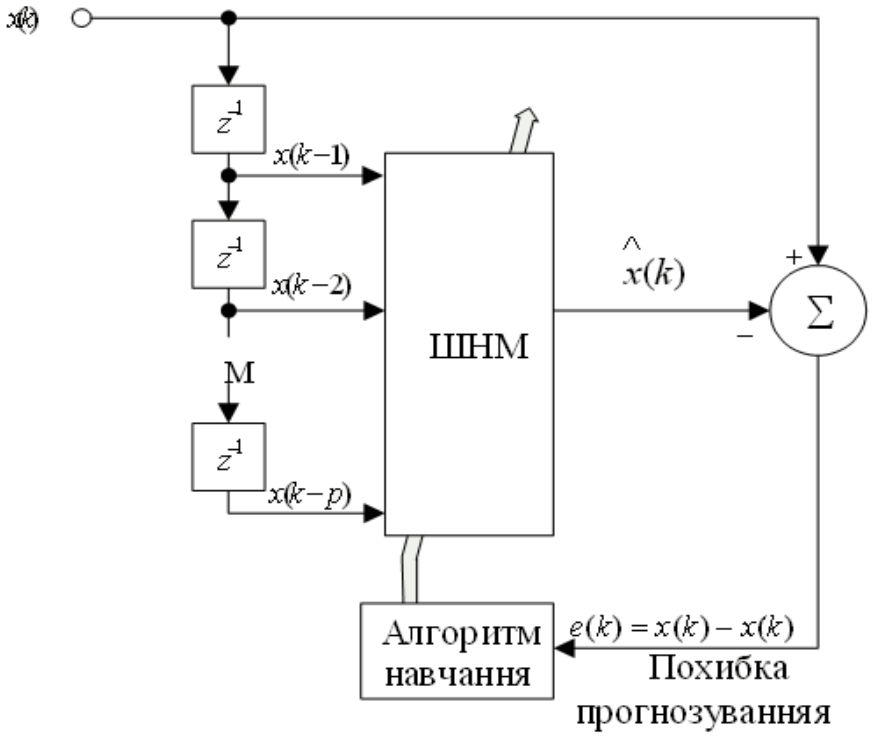


Рис. 4.8. Схема нейромережного прогнозування

Зараз нейромережні прогнозуючі моделі успішно використовуються для вирішення широкого кола задач науки, техніки, економіки [265, 266, 267, 268].

4.3 Лінійні алгоритми навчання

У цьому підрозділі розглянуто алгоритми, що засновані на парадигмі навчання із вчителем і реалізують правило корекції за помилкою, при цьому припускається, що сама помилка є лінійною функцією синаптичних ваг.

З математичної точки зору процес навчання в цьому випадку зводиться до мінімізації критерію якості навчання (цільової функції) за синаптичними вагами w_{ji} $(i = 0,1,\ldots,n)$ і може протікати як у неперервному, так і у дискретному $k = 0,1,2,\ldots$ часі.

Як цільова функція найбільш часто приймається квадрат поточного значення помилки навчання, тобто

$$E_j(t) = \frac{1}{2}e_j^2(t) = \frac{1}{2}\left(d_j(t) - u_j(t)\right)^2 \tag{4.4}$$

або, що в принципі те саме,

$$E_j(k) = \frac{1}{2}e_j^2(k) = \frac{1}{2}\left(d_j(k) - u_j(k)\right)^2. \tag{4.5}$$

Градієнтна оптимізація (4.4) у неперервному часі [269] призводить до системи диференціальних рівнянь

$$\frac{dw_{ji}}{dt} = -\eta\frac{\partial E_j(t)}{\partial w_{ji}} = -\eta\frac{\partial E_j(t)}{\partial u_j}\cdot\frac{\partial u_j}{\partial w_{ji}} \tag{4.6}$$

або з урахуванням того, що

$$u_j = \sum_{i=0}^{n} w_{ji} x_i, \tag{4.7}$$

$$\frac{dw_{ji}}{dt} = \eta e_j(t) x_i(t) = \eta \left(d_j(t) - \sum_{l=0}^{n} w_{jl} x_l \right) x_i, \quad i = 0,1,\ldots,n, \tag{4.8}$$

де $\eta > 0$ – скалярний параметр, що визначає швидкість навчання.

На практиці найбільшого поширення набули дискретні алгоритми навчання типу

$$w_{ji}(k+1) = w_{ji}(k) + \eta(k) e_j(k) x_i(k), \tag{4.9}$$

чи у векторній формі

$$w_j(k+1) = w_j(k) - \eta(k) \nabla_{w_j} E_j(k) = w_j(k) + \eta(k) e_j(k) x(k), \tag{4.10}$$

де $\nabla_{w_j} E_j(k) = -e_j(k) x(k)$ – вектор-градієнт цільової функції за синаптичними вагами.

Швидкість процесу навчання за допомогою алгоритму (4.9), (4.10) цілком визначається вибором параметра $\eta(k)$, що визначає крок зсуву в просторі параметрів, що настроюються. Природно вибрати цей параметр так, щоб швидкість збіжності поточних значень $w_j(k)$ до оптимальних гіпотетичних ваг w_j була максимальною. Вводячи до розгляду вектор відхилень поточних значень $w_j(k)$ від оптимальних w_j у вигляді

$$\tilde{w}_j(k) = w_j - w_j(k) \tag{4.11}$$

і вирішуючи диференціальне рівняння

$$\frac{\partial \left\| \tilde{w}_j(k) \right\|^2}{\partial \eta} = 0, \tag{4.12}$$

нескладно отримати оптимальне значення параметра кроку у вигляді

$$\eta(k) = \left\| x(k) \right\|^{-2}, \tag{4.13}$$

що приводить до алгоритму навчання

$$w_j(k+1) = w_j(k) + \frac{e_j(k) x(k)}{\left\| x(k) \right\|^2}, \tag{4.14}$$

відомому в теорії штучних нейронних мереж як алгоритм Уідроу–Гоффа.

Не можна не відзначити, що вперше цей алгоритм був запропонований С. Качмажем набагато раніше [270, 271] і задовго до появи нейроматематики використовувався для вирішення задач адаптивної ідентифікації об’єктів керування [302] у так званій мультиплікативній формі

$$w_j(k+1) = w_j(k) + \frac{\eta e_j(k) x(k)}{\left\| x(k) \right\|^2}, \quad 0 < \eta < 2, \tag{4.15}$$

що забезпечує при відповідному виборі параметра η завадостійкість процесу навчання.

Процес настроювання одиничного нейрона-адаліни нескладно розповсюдити на деякі види нейромереж у цілому. Записавши перетворення, здійснюване мережею з лінійними синаптичними вагами, наприклад, радіально-базисною мережею

$$y = F(x) = w_0 + \sum_{i=1}^{h} w_i \phi_i(x) = w^T \phi(x), \tag{4.16}$$

де $w = (w_0, w_1, w_2, \ldots, w_h)^T$, $\phi(x) = (1, \phi_1(x), \ldots, \phi_h(x))^T$, приходимо до градієнтного алгоритму навчання радіально-базисної мережі

$$w(k+1) = w(k) + \eta \frac{d(k) - w^T(k)\phi(x(k))}{\|\phi(x(k))\|^2} \phi(x(k)), \tag{4.17}$$

що забезпечує збіжність синаптичних ваг до своїх оптимальних значень для будь-якої послідовності лінійно-незалежних векторів $\phi(x(1)), \phi(x(2)), \ldots, \phi(x(k)), \ldots$ Для мережі з множиною вихідних сигналів $y_j(k)$ отримуємо

$$\begin{cases} w_j(k+1) = w_j(k) + \eta \dfrac{d_j(k) - y_j(k)}{\|\phi(x(k))\|^2} \phi(x(k)), \\ y_j(k) = w_j^T(k)\phi(x(k)); \quad j = 1, 2, \ldots m. \end{cases} \tag{4.18}$$

Поряд з квадратичними критеріями якості навчання (4.4), (4.5) набули поширення й інші форми цільових функцій, вибір яких значною мірою визначається апріорною інформацією про характер розподілу вхідних сигналів і діючих перешкод [250]. Так найбільшу завадостійкість забезпечує використання модульного критерію

$$E_j(k) = |d_j(k) - y_j(k)| = |e_j(k)|, \tag{4.19}$$

що приводить до алгоритму

$$w_j(k+1) = w_j(k) - \eta(k) sign\left(y_j(k) - w_j^T(k)\phi(x(k))\phi(x(k))\right). \tag{4.20}$$

Якщо замість звичайної сигнум-функції використовується релейна функція з зоною нечутливості 2Δ, то процедура (4.20) набуває вигляду [272]

$$\begin{aligned} w_j(k+1) = w_j(k) - \frac{\eta(k)}{2}\left(sign\left(y_j(k) - w_j^T(k)\phi(x(k))\right) + \Delta\right) + \\ + sign\left(y_j(k) - w_j^T(k)\phi(x(k)) - \Delta\right)\phi(x(k)). \end{aligned} \tag{4.21}$$

У [273] було запропоновано методику побудови алгоритмів навчання, суть якої в тому, що за відомої оцінки $w_j(k)$ наступне значення $w_j(k+1)$ знаходиться з умови мінімуму норми $\left|\sum_{i=0}^{h} |w_{ji}(k+1) - w_{ji}(k)|^q\right|^{\frac{1}{q}}$ з обмеженням

$$y_j(k) - w_j^T(k+1)\phi(x(k)) = 0, \tag{4.22}$$

тобто уточнений вектор синаптичних ваг $w_j(k+1)$ перетворює в нуль апостеріорну помилку навчання. З цих міркувань випливає, що алгоритм

Качмажа–Уідроу–Гоффа мінімізує евклідову норму $(q=2)$. Мінімум кубічній нормі доставляє алгоритм Нагумо–Ноди [274]

$$w_j(k+1)=w_j(k)+\frac{y_j(k)-w_j^T(k)\phi(x(k))}{\phi^T(x(k))\,sign\phi(x(k))}sign\phi(x(k)), \tag{4.23}$$

а алгоритм Некрасова [275]

$$\begin{cases} w_{ji}(k+1)=w_{ji}(k)+\dfrac{y_j(k)-w_j^T(k)\phi(x(k))}{\max\limits_{i=0,1,\ldots,h}|\phi_i(x(k))|}, \\ w_{jp}(k+1)=w_{jp}(k) \end{cases} \tag{4.24}$$

мінімізує октаедричну норму.

Заслуговує на увагу алгоритм навчання, що є розширенням (4.23)

$$w_j(k+1)=w_j(k)+\frac{y_j(k)-w_j^T(k)\phi(x(k))}{\psi(\phi^T(x((k))))\phi(x(k))}\psi(\phi(x(k))), \tag{4.25}$$

(тут $\psi(\phi(x(k)))=(\psi_0(\phi(x(k))),\psi_1(\phi(x(k))),\ldots,\psi_h(\phi(x(k))))^T$ і тісно пов’язаний з методом інструментальних змінних [251].

Ціла низка алгоритмів може бути отримана в ході використання цільової функції у вигляді нерівності

$$E_j(k)=e_j(k)\Delta e_j(k)\le 0, \tag{4.26}$$

де $\Delta e_j(k)=e_j(k+1)-e_j(k)$ – перша різниця послідовності помилок навчання.

Нескладно побачити, що всі розглянуті вище алгоритми задовольняють цій нерівності. Крім того, легко можуть бути отримані алгоритми, які використовують нелінійності типу сигнум-функції, наприклад,

$$w_j(k+1)=w_j(k)+\eta(k)(y_j(k)-w_j^T(k)\phi(x(k)))sign\phi(x(k)), \tag{4.27}$$

$$w_j(k+1)=w_j(k)+\eta(k)\frac{sign(y_j(k)-w_j^T(k)\phi(x(k)))}{\|\phi(x(k))\|^2}\phi(x(k)) \tag{4.28}$$

і багато інших.

Розглянуті процедури належать до так званих однокрокових алгоритмів навчання, оскільки за кожного уточнення синаптичних ваг використовується тільки одне останнє значення помилки $e_j(k)$. Застосовуючи алгоритми більш складної структури, що враховують інформацію про передісторію процесу навчання, можна домогтися істотного скорочення часу настроювання і забезпечити можливість сталої роботи як в умовах перешкод, так і нестаціонарності зовнішнього середовища.

На практиці найбільше поширення набули алгоритми, пов’язані з критерієм мінімуму суми квадратів помилок навчання

$$E_j^k=\sum_{p=0}^{k}\alpha(p)e_j^2(p)=\sum_{p=0}^{k}\alpha(p)E_j(p) \tag{4.29}$$

і їхні модифікації, що обумовлені обраною системою вагових коефіцієнтів $\alpha(p)$, $p=0,1,2,\ldots,k$. При цьому вкрай важливим є той факт, що всі процедури мають уніфіковану форму [251, 241]

$$w_j(k+1)=w_j(k)+\eta(k)\big(d_j(k)-w_j^T(k)\phi(x(k))\big)\phi(x(k)), \qquad (4.30)$$

а відмінність між ними визначається лише коефіцієнтом $\eta(k)$, який може бути не тільки скаляром, але і матрицею. Наприклад, алгоритму Качмажа відповідає $\eta(k)=\|\phi(x(k))\|^{-2}$, стохастичній апроксимації [276] $-\eta(k)=\left(\sum_{p=0}^{k}\phi^T(x(k))\phi(x(k))\right)^{-1}$, методу найменших квадратів – матриця $\eta(k)=\left(\sum_{p=0}^{k}\phi(x(k))\phi^T(x(k))\right)^{-1}$ тощо.

Нижче ми розглянемо групу багатокрокових алгоритмів навчання, що породжуються мінімізацією критерію

$$E_j^k=\sum_{p=0}^{k}e_j^2(k-p)g(k-p), \qquad (4.31)$$

де $g(k-p)$ – функція вірогідності p-го спостереження щодо поточного моменту часу k. Виходячи зі зручності реалізації обчислювальних процедур і фізичного тлумачення процесу обробки «нової» і «застарілої» інформації, функцію $g(k-p)$ зазвичай задають у двох варіантах:

– у вигляді «ковзного вікна»

$$g(k-p)=\begin{cases}1, & \text{якщо } 0\le p<\chi,\\ 0, & \text{якщо } \chi\le p\le k,\end{cases} \qquad (4.32)$$

(тут χ – величина ковзного вікна, чи пам'ять алгоритму),

– у вигляді «експонеційного убування цінності інформації»

$$g(k-p)=\alpha^p, \quad 0\le\alpha\le 1, \qquad (4.33)$$

де α – фактор забування [251].

Функція вірогідності типу «ковзного вікна» породжує багатокроковий алгоритм [277, 278]

$$w_j(k+1)=w_j(k)+\eta(k)\big(d_j(k)-w_j^T(k)\phi(x(k))\big)\phi(x(k)), \qquad (4.34)$$

де

$$\tilde{r}(k-1)=P_\phi(k-1)\phi(x(k-\chi)), \qquad (4.35)$$

$$\tilde{P}_\phi(k-1)=\begin{cases}P_\phi(k-1)-\dfrac{\tilde{r}(k-1)\tilde{r}^T(k-1)P_\phi(k-1)+P_\phi(k-1)\tilde{r}(k-1)\tilde{r}^T(k-1)}{\|\tilde{r}(k-1)\|^4}+\\ +\dfrac{\tilde{r}^T(k-1)p_\phi(k-1)\tilde{r}(k-1)\tilde{r}(k-1)\tilde{r}^T(k-1)}{\|\tilde{r}(k-1)\|^2},\\ \text{якщо}\quad \Phi^T(k-1)\tilde{r}(k-1)=g\pm\varepsilon_1(k),\\ P_\phi(k-1)+\dfrac{\tilde{r}(k-1)\tilde{r}^T(k-1)}{1-\phi^T(x(k-\chi))\tilde{r}(k-1)} \text{ в іншому випадку,}\end{cases} \qquad (4.36)$$

$$\tilde{A}(k-1)=\begin{cases} A(k-1)+\dfrac{\tilde{r}(k-1)\tilde{r}^T(k-1)}{\left\|\tilde{r}(k-1)\right\|^2}, & \text{якщо } \Phi^T(k-1)\tilde{r}(k-1)=g\pm\varepsilon_1(k), \\ A(k-1) & \text{в іншому випадку,} \end{cases} \tag{4.37}$$

$$\begin{cases} r(k)=\tilde{P}_\phi(k-1)\phi(x(k)), \\ q(k)=\tilde{A}(k-1)\phi(x(k)), \end{cases} \tag{4.38}$$

$$P_\phi(k)=\begin{cases} \tilde{P}_\phi(k-1)-\dfrac{r(k)q^T(k)+q(k)r^T(k)}{\phi^T(x(k))q(k)}+ \\ +\dfrac{1+\phi^T(x(k))r(k)}{\left(\phi^T(x(k))q(k)\right)^2}q(k)q^T(k), \text{ якщо } \phi^T(x(k))q(k)\ge\varepsilon_2(k), \\ \tilde{P}_\phi(k-1)-\dfrac{r(k)r^T(k)}{1+\phi^T(x(k))r(k)} \text{ в іншому випадку,} \end{cases} \tag{4.39}$$

$$A(k)=\begin{cases} \tilde{A}(k-1)-\dfrac{q(k)q^T(k)}{\phi^T(x(k))q(k)}, & \text{якщо } \phi^T(x(k))q(k)\ge\varepsilon_2(k), \\ \tilde{A}(k-1) & \text{в іншому випадку,} \end{cases} \tag{4.40}$$

$$\eta(k)=\begin{cases} \dfrac{A(k-1)}{\phi^T(x(k))A(k-1)\phi(x(k))}, \\ \text{якщо } \phi^T(x(k))A(k-1)\phi(x(k))\ge\varepsilon_2(k), \\ \dfrac{P_\phi(k-1)}{1+\phi^T(x(k))P_\phi(k-1)\phi(x(k))} \text{ в іншому випадку,} \end{cases} \tag{4.41}$$

$g=(1,0,\ldots,0)^T-(\chi\times1)$ – вектор, $\varepsilon_1(k)$ і $\varepsilon_2(k)$ – деякі граничні величини, що залежать від ступеня мультиколінеарності векторів $\phi(x(p))$, що задають відповідний спосіб їхньої обробки, $\Phi(k)=\left(\phi(x(k-\chi+1)),\ldots,\phi(x(k-1)),\phi(x(k))\right)$.

З алгоритму (4.34) – (4.41) випливає цілий ряд відомих процедур. Так $\chi=1$ відповідає алгоритму Качмажа–Уідроу–Гоффа, при $1<\chi<h+1$ приходимо до модифікованого алгоритму Качмажа [279], при $h+1\le\chi<k$ отримуємо алгоритм поточного регресійного аналізу [280]

$$w_j(k+1)=w_j(k)+\frac{P_\phi(k-1)\left(d_j(k)-w_j^T(k)\phi(x(k))\right)}{1+\phi^T(x(k))P_\phi(k-1)\phi(x(k))}\phi(x(k)), \tag{4.42}$$

$$\begin{cases} \tilde{P}_\phi(k-1)=P_\phi(k-1)+\dfrac{P_\phi(k-1)\phi(x(k-\chi))\phi^T(x(k-\chi))P_\phi(k-1)}{1-\phi^T(x(k-\chi))P_\phi(k-1)\phi(x(k-\chi))}, \\ P_\phi(k)=\tilde{P}_\phi(k-1)-\dfrac{\tilde{P}_\phi(k-1)\phi(x(k))\phi^T(x(k))\tilde{P}_\phi(k-1)}{1+\phi^T(x(k))\tilde{P}_\phi(k-1)\phi(x(k))}, \end{cases} \tag{4.43}$$

і, нарешті, при $\chi = k$ отримуємо стандартний рекурентний метод найменших квадратів, що набув широкого поширення як в адаптивній ідентифікації [251, 241], так і навчанні нейронних мереж [235, 238]:

$$\begin{cases} w_j(k+1) = w_j(k) + \dfrac{P_\phi(k-1)\left(d_j(k) - w_j^T(k)\phi\left(x(k)\right)\right)}{1+\phi^T\left(x(k)\right)P_\phi(k-1)\phi\left(x(k)\right)}\phi\left(x(k)\right), \\ P_\phi(k) = P_\phi(k-1) - \dfrac{P_\phi(k-1)\phi\left(x(k)\right)\phi^T\left(x(k)\right)P_\phi(k-1)}{1+\phi^T\left(x(k)\right)P_\phi(k-1)\phi\left(x(k)\right)}. \end{cases} \tag{4.44}$$

Принциповим питанням використання розглянутих алгоритмів є обґрунтований вибір величини «вікна», що залежить як від характеру нестаціонарності зовнішнього середовища, так і від рівня діючих завад. Оскільки слідкуючі та фільтруючі властивості алгоритму вступають у протиріччя, тому необхідно передбачити можливість підключення додаткової процедури керування пам'яттю, що реалізує компроміс між цими властивостями [239, 281, 282]. Поліпшення фільтруючих властивостей пов'язане з необхідністю збільшення пам'яті алгоритму, а, отже, з накопиченням великих обсягів даних. У цьому випадку більш доцільне використання функції $g(k-p)$ у формі «експонеційного убування цінності інформації».

Мінімізація критерію якості навчання

$$E_j^k = \sum_{p=0}^{k} \alpha^{k-p} e_j^2(p) \tag{4.45}$$

приводить до широко розповсюдженого експоненційно зваженого рекурентного методу найменших квадратів:

$$\begin{cases} w_j(k+1) = w_j(k) + \dfrac{P_\phi(k-1)\left(d_j(k) - w_j^T(k)\phi\left(x(k)\right)\right)}{\alpha+\phi^T\left(x(k)\right)P_\phi(k-1)\phi\left(x(k)\right)}\phi\left(x(k)\right), \\ P_\phi(k) = \dfrac{1}{\alpha}\left(P_\phi(k-1) - \dfrac{P_\phi(k-1)\phi\left(x(k)\right)\phi^T\left(x(k)\right)P_\phi(k-1)}{\alpha+\phi^T\left(x(k)\right)P_\phi(k-1)\phi\left(x(k)\right)} \right), \end{cases} \tag{4.46}$$

де $0 < \alpha \le 1$.

Проблема практичного використання алгоритму (4.46) ускладнюється тим, що в процесі навчання може виникнути так званий «вибух параметрів» коваріційної матриці $P_\phi(k)$, тобто експонеційне зростання її елементів. Попередження цього небажаного явища пов'язане з правильним вибором фактора забування α, який зазвичай обирається в діапазоні $0.95 \le \alpha \le 0.99$, що відповідає $20 \le \chi \le 100$ в алгоритмі з ковзним вікном. У загальному випадку відомо [283], що алгоритми (4.42) і (4.46) приводять до аналогічних результатів при

$$\alpha = \frac{\chi - 1}{\chi + 1}, \tag{4.47}$$

однак зменшення α приводить до швидкого виродження матриці $P_\phi^{-1}(k) = \sum_{p=0}^{k} \alpha^{k-p}\phi(p)\phi^T(p)$ і, як наслідок, до «вибуху параметрів».

Застосування в алгоритмі (4.46) замість обернення матриці операції псевдообернення [284] приводить до процедури [285, 286]:

$$w_j(k+1) = w_j(k) + \eta(k)\left(d_j(k) - w_j^T(k)\phi\left(x(k)\right)\right)\phi\left(x(k)\right), \tag{4.48}$$

де

$$\eta(k) = \begin{cases} \dfrac{\tilde{A}(k-1)}{\phi^T\left(x(k)\right)\tilde{A}(k-1)\phi\left(x(k)\right)}, \text{ якщо } \phi^T\left(x(k)\right)\tilde{A}(k-1)\phi\left(x(k)\right) \geq \varepsilon_2(k), \\ \dfrac{P_\phi(k-1)}{1+\phi^T\left(x(k)\right)P_\phi(k-1)\phi\left(x(k)\right)} \text{ в іншому випадку,} \end{cases} \tag{4.49}$$

$$\tilde{A}(k) = \begin{cases} \tilde{A}(k-1) - \dfrac{\tilde{A}(k-1)\phi\left(x(k)\right)\phi^T\left(x(k)\right)\tilde{A}(k-1)}{\phi^T\left(x(k)\right)\tilde{A}(k-1)\phi\left(x(k)\right)}, \\ \text{якщо } \phi^T\left(x(k)\right)\tilde{A}(k-1)\phi\left(x(k)\right) \geq \varepsilon_2(k), \\ \tilde{A}(k-1) \text{ в іншому випадку,} \end{cases} \tag{4.50}$$

$$P_\phi(k) = \begin{cases} \dfrac{1}{\alpha}\left(P_\phi(k-1) - \dfrac{\left(P_\phi(k-1)\phi\left(x(k)\right)\right)\left(\tilde{A}(k-1)\phi\left(x(k)\right)\right)^T}{\phi^T\left(x(k)\right)\tilde{A}(k-1)\phi\left(x(k)\right)} + \right. \\ + \dfrac{\left(\tilde{A}(k-1)\phi\left(x(k)\right)\right)\left(P_\phi(k-1)\phi\left(x(k)\right)\right)^T}{\phi^T\left(x(k)\right)\tilde{A}(k-1)\phi\left(x(k)\right)} + \\ \left. + \dfrac{\alpha + \phi^T\left(x(k)\right)P_\phi(k-1)\phi\left(x(k)\right)}{\left(\phi^T\left(x(k)\right)\tilde{A}(k-1)\phi\left(x(k)\right)\right)^2}\left(\tilde{A}(k-1)\phi\left(x(k)\right)\left(P_\phi(k-1)\phi\left(x(k)\right)\right)^T\right)\right), \\ \text{якщо } \phi^T\left(x(k)\right)\tilde{A}(k-1)\phi\left(x(k)\right) \geq \varepsilon_2(k), \\ \dfrac{1}{\alpha}\left(P_\phi(k-1) - \dfrac{P_\phi(k-1)\phi\left(x(k)\right)\phi^T\left(x(k)\right)P_\phi(k-1)}{\alpha + \phi^T\left(x(k)\right)P_\phi(k-1)\phi\left(x(k)\right)}\right) \text{ в іншому випадку,} \end{cases} \tag{4.51}$$

$$\begin{cases} \tilde{A}(k) = \mathrm{I} - \left(\Phi^T(k)\right)^+\Phi^T(k), \ \tilde{A}(0) = \mathrm{I}, \\ P_\phi(k) = \left(\Phi(k)A^{-2}(k)\Phi^T(k)\right)^+, \ P_\phi(0) = 0, \\ \Phi(k) = \left(\phi\left(x(0)\right), \phi\left(x(1)\right), \ldots, \phi\left(x(k)\right)\right), \\ A(k) = diag\left(\alpha^{\frac{1-k}{2}}, \alpha^{\frac{2-k}{2}}, \ldots, \alpha^{-\frac{1}{2}}, 1\right), \end{cases} \tag{4.52}$$

$\mathrm{I} - (h+1)\times(h+1)$ – одинична матриця, $(\bullet)^+$ – символ псевдообернення за Муру–Пенроузу [284].

Хоча алгоритм (4.48) – (4.52) є працездатним при будь-яких значеннях фактора забування α, його громіздкість змушує шукати альтернативні підходи до синтезу багатокрокових алгоритмів навчання.

У теорії і практиці адаптивних систем, що навчаються, поряд з рекурентним методом найменших квадратів і його модифікацій широкого поширення набули алгоритми, засновані на стохастичній апроксимації [239, 276, 287].

Прикладом є модифікований алгоритм типу стохастичної апроксимації [288]

$$\begin{cases} w_j(k+1) = w_j(k) + ar^{-1}(k)\big(d_j(k) - w_j^T(k)\phi(x(k))\big)\phi\big(x(k)\big), \\ r(k) = \alpha r(k-1) + \big\|\phi\big(x(k)\big)\big\|^2, \end{cases} \tag{4.53}$$

(тут $r(0) = 1,\ 0 \le \alpha \le 1,\ 0 < a < 2$) співпадаючий при $\alpha = 0$ з однокроковим алгоритмом Качмажа–Уідроу–Гоффа, а при $\alpha = 1$ – з адаптивним алгоритмом стохастичної апроксимації Гудвіна–Ремеджа–Кейнеса [289, 290]. У [291, 272] досліджено збіжність цієї процедури, що відрізняється від алгоритму, введеного в [289, 290], наявністю фактора забування α, що дозволяє забезпечити процесу навчання слідкуючі властивості і водночас виключає можливість «вибуху параметрів».

Аналогічно попередньому можна записати алгоритм типу (4.53) з «ковзним вікном», при цьому

$$\begin{cases} w_j(k+1) = w_j(k) + ar^{-1}(k)\big(d_j(k) - w_j^T(k)\phi\big(x(k)\big)\big)\phi\big(x(k)\big), \\ r(k) = r(k-1) + \big\|\phi\big(x(k)\big)\big\|^2 - \big\|\phi\big(x(k-\chi)\big)\big\|^2. \end{cases} \tag{4.54}$$

Порівняльний аналіз алгоритмів (4.46) і (4.53) показує, що процедура (4.46), маючи високу швидкість збіжності, працює в дуже вузькому діапазоні зміни фактора забування, а алгоритм (4.53), як і всі процедури стохастичної апроксимації, характеризується низькою швидкістю. Цих недоліків значною мірою позбавлений градієнтний (зі скалярним коефіцієнтом $\eta(k)$) експоненційно зважений оптимальний за швидкістю алгоритм [292, 293]

$$w_j(k+1) = w_j(k) + \frac{\overline{e}_j^2(k)\big(r_j(k) - R(k)w_j((k)\big)}{\big\|r_j(k) - R(k)w_j(k)\big\|^2}, \tag{4.55}$$

де

$$\begin{cases} \overline{e}_j^2(k) = e_j^2(k) + \alpha\,\overline{e}_j^2(k-1), \\ r_j(k) = d_j(k)\phi\big(x(k)\big) + \alpha\, r_j(k-1), \\ R(k) = \phi\big(x(k)\big)\phi^T\big(x(k)\big) + \alpha\, R(k-1), \\ 0 \le \alpha \le 1. \end{cases} \tag{4.56}$$

У [291] досліджено збіжність цієї процедури в нестаціонарних стохастичних умовах та доведено, що за фільтруючими властивостями, вона перевершує розглянуті градієнтні алгоритми навчання.

У випадку, якщо функція вірогідності $g(k-p)$ має вигляд «ковзного вікна», співвідношення (4.56) набувають форми

$$\begin{cases} \overline{e}_j^2(k) = e_j^2(k) + \overline{e}_j^2(k-1) - e_j^2(k-\chi), \\ r_j(k) = d_j(k)\phi(x(k)) + r_j(k-1) - d_j(k-\chi)\phi(x(k-\chi)), \\ R(k) = \phi(x(k))\phi^T(x(k)) + R(k-1) - \phi(x(k-\chi))\phi^T(x(k-\chi)). \end{cases} \quad (4.57)$$

Задача навчання штучних нейронних мереж може значно ускладнюватися у випадку, якщо сигнали $\phi(x(k))$ характеризуються високим рівнем кореляції. За цих умов методи, що засновані на традиційних квадратичних критеріях, виявляються ненадійними, а отримувані за їх допомогою оцінки синаптичних ваг не забезпечують необхідної точності.

Ефективним засобом підвищення якості навчання може слугувати використання методів зміщеного оцінювання [294, 295], що дозволяє у більшості випадків отримувати значення параметрів більш близькі до оптимальних, ніж оцінки, отримувані за допомогою методу найменших квадратів.

У теорії і практиці зміщеного оцінювання як найбільш універсальні можна виділити так звані двопараметричні оцінки, що у загальному випадку мають вигляд

$$w_j(k) = (\left(\Phi(k) + l\left(\Phi(k)\Phi^T(k)\right)^q\right)^{-1}\Phi(k)D_j(k), \quad (4.58)$$

де $\Phi(k) = \left(\phi(x(0)), \phi(x(1)), \ldots, \phi(x(k))\right) - (h+1)\times(h+1)$ – матриця вхідних сигналів, $D_j(k) = \left(d_j(0), d_j(1), \ldots, d_j(k)\right)^T - (k+1)\times 1$ – вектор навчальних сигналів, l і q – деякі скалярні параметри, що визначають властивості отримуваних оцінок, $l\left(\Phi(k)\Phi^T(k)\right)^q = L(k)$ – добавка, що забезпечує стійкість процедури оцінювання. Слід підкреслити, що регуляризація інформаційної матриці $\Phi(k)\Phi^T(k)$ за допомогою добавки $L(k)$ має на меті не тільки більш стійке її обернення, але й поліпшення статистичних властивостей оцінок, хоча спочатку ідея регуляризації носила суто обчислювальний характер.

Використання оцінок (4.58) для навчання в реальному часі дещо утруднене, оскільки вимагає збереження всієї навчальної вибірки $\Phi(k), D_j(k)$. Вводячи у розгляд матриці $R(k) = P_\phi^{-1}(k) = \Phi(k)\Phi^T(k)$, $r_j = \Phi(k)D_j(k)$, оцінки (4.58) можна переписати в рекурентній формі [296, 297]

$$\begin{cases} w_j(k+1) = \left(R(k) + lR^q(k)\right)^{-1} r(k), \\ r_j(k) = d_j(k)\phi(x(k)) + r_j(k-1), \\ R(k) = \phi(x(k))\phi^T(x(k)) + R(k-1), \\ P_\phi(k) = P_\phi(k-1) - \dfrac{P_\phi(k-1)\phi(x(k))\phi^T(x(k))P_\phi(k-1)}{1+\phi^T(x(k))P_\phi(k-1)\phi(x(k))}. \end{cases} \quad (4.59)$$

З (4.59) як окремі випадки випливають:

– узагальнені гребеневі оцінки Єрмакова–Панкрат'єва [298] (при $q=-1$);

– звичайні гребеневі (ридж) оцінки (при $q=0$);

– стиснуті оцінки з параметром стиску $(1+l)^{-1}$ (при $q=1$);

– звичайні оцінки найменших квадратів (при $l=0$).

Скориставшись формулою Шермана–Моррісона–Вудбері, запишемо очевидні перетворення

$$\begin{aligned}&\left(R(k)+l\,R^{q}(k)\right)^{-1}=\left(P_{\phi}^{-1}(k)+l\,P_{\phi}^{-q}(k)\right)^{-1}=\\&=\left(\mathrm{I}-l\,P_{\phi}^{\frac{2-q}{2}}(k)\left(\mathrm{I}+l\,P_{\phi}^{1-q}(k)\right)^{-1}P_{\phi}^{-\frac{q}{2}}(k)\right)P_{\phi}(k)=\\&=\left(\mathrm{I}+l\,P_{\phi}^{1-q}(k)\right)^{-1}P_{\phi}(k),\end{aligned}\tag{4.60}$$

з яких з урахуванням (4.59) і (4.60) випливає

$$\begin{aligned}&w_j(k+1)=\left(\mathrm{I}-l\,P_{\phi}^{\frac{2-q}{2}}(k)\left(\mathrm{I}+l\,P_{\phi}^{1-q}(k)\right)^{-1}P_{\phi}^{-\frac{q}{2}}(k)\right)w_j^{*}(k)=\\&=\left(\mathrm{I}+l\,P_{\phi}^{1-q}(k)\right)^{-1}w_j^{*}(k),\end{aligned}\tag{4.61}$$

де $w_j^{*}(k)$ – звичайна оцінка методу найменших квадратів (4.44).

Відповідні рекурентні співвідношення для обчислення синаптичних ваг у загальному випадку набувають вигляду [299]

$$\begin{cases}w_j(k+1)=U(k)(w_j^{*}(k)+P_{\phi}(k)\phi(x(k))\left(d_j(k)-\phi^{T}(x(k))w_j^{*}(k)\right),\\P_{\phi}(k)=P_{\phi}(k-1)-\dfrac{P_{\phi}(k-1)\phi(x(k))\phi^{T}(x(k))P_{\phi}(k-1)}{1+\phi^{T}(x(k))P_{\phi}(k-1)\phi(x(k))},\\U(k)=\left(\mathrm{I}-l\,P_{\phi}^{\frac{2-q}{2}}(k)\left(\mathrm{I}+l\,P_{\phi}^{1-q}(k)\right)^{-1}P_{\phi}^{-\frac{q}{2}}(k)\right)=\left(\mathrm{I}+l\,P_{\phi}^{1-q}(k)\right)^{-1},\end{cases}\tag{4.62}$$

де для оцінок Єрмакова–Панкрат'єва –

$$U(k)=\left(\mathrm{I}-l\,P_{\phi}^{\frac{3}{2}}(k)\left(\mathrm{I}+l\,P_{\phi}^{2}(k)\right)^{-1}P_{\phi}^{\frac{1}{2}}(k)\right)=\left(\mathrm{I}+l\,P_{\phi}^{2}(k)\right)^{-1},\tag{4.63}$$

для звичайних ридж-оцінок –

$$U(k)=\mathrm{I}-l\,P_{\phi}(k)\left(\mathrm{I}+l\,P_{\phi}(k)\right)^{-1}=\left(\mathrm{I}+l\,P_{\phi}(k)\right)^{-1},\tag{4.64}$$

для стиснутих оцінок –

$$U(k)=(1+l)^{-1}\mathrm{I},\tag{4.65}$$

для оцінок методу найменших квадратів –

$$U(k)=\mathrm{I}.\tag{4.66}$$

Таким чином, вибір конкретного типу зміщених оцінок залежно від умов навчання зводиться до вибору оператора $U(k)$, діючого на звичайну оцінку найменших квадратів.

Разом з тим, введені оцінки непридатні для навчання за нестаціонарних умов, оскільки враховують всю ретроспективну інформацію з однаковою вагою. Кількість робіт із синтезу рекурентних алгоритмів навчання з кінцевою пам'яттю, що використовує ідеї регуляризації, досить незначна [151,152,153]. Тут, зокрема, можна відзначити роботу [299], де запропоновано алгоритм

$$\begin{cases} w_j(k+1) = w_j(k) + \dfrac{P_\phi(k-1)\phi(x(k))e_j(k)}{1+\phi^T(x(k))P_\phi(k-1)\phi(x(k))}, \\ P_\phi(k) = P_\phi(k-1) - \dfrac{P_\phi(k-1)\phi(x(k))\phi^T(x(k))P_\phi(k-1)}{1+\phi^T(x(k))P_\phi(k-1)\phi(x(k))}, \\ w_j(0) = 0,\ P_\phi(0) = l^{-1}\mathrm{I} \end{cases} \tag{4.67}$$

і [300] –

$$\begin{cases} w_j(k+1) = w_j(k) + \dfrac{P_\phi(k-1)\phi(x(k))e_j(k)}{1+\phi^T(x(k))P_\phi(k-1)\phi(x(k))}, \\ P_\phi(k) = P_\phi(k-1) - \dfrac{P_\phi(k-1)\phi(x(k))\phi^T(x(k))P_\phi(k-1)}{1+\phi^T(x(k))P_\phi(k-1)\phi(x(k))}, \\ w_j(0) = 0,\ P_\phi(0) = L^{-1}, \end{cases} \tag{4.68}$$

при цьому алгоритм (4.67) забезпечує отримання звичайної ридж-оцінки, а (4.68) – узагальненої. Основним недоліком цих алгоритмів є те, що зі зростанням обсягу вибірки частка регуляризуючої добавки в інформаційній матриці постійно падає, що веде до втрати властивостей оцінок.

Адаптивний алгоритм із ковзним вікном може бути отриманий з (4.68) шляхом додавання співвідношення для «скидання» застарілої інформації [301]

$$\begin{cases} w_j(k+1) = w_j(k) + \dfrac{P_\phi(k-1)\left(d_j(k) - w_j^T(k)\phi(x(k))\right)}{1+\phi^T(x(k))P_\phi(k-1)\phi(x(k))}\phi(x(k)), \\ \tilde{P}_\phi(k-1) = P_\phi(k-1) + \dfrac{P_\phi(k-1)\phi(x(k-\chi))\phi^T(x(k-\chi))P_\phi(k-1)}{1-\phi^T(x(k-\chi))P_\phi(k-1)\phi(x(k-\chi))}, \\ P_\phi(k) = \tilde{P}_\phi(k-1) - \dfrac{\tilde{P}_\phi(k-1)\phi(x(k))\phi^T(x(k))\tilde{P}_\phi(k-1)}{1+\phi^T(x(k))\tilde{P}_\phi(k-1)\phi(x(k))}, \\ w_j(0) = 0,\ P_\phi(0) = L^{-1}. \end{cases} \tag{4.69}$$

При $\chi = 1$ приходимо до однокрокового алгоритму

$$w_j(k+1) = w_j(k) - L^{-1}\frac{d_j(k) - w_j^T(k)\phi(x(k))}{1+\phi^T(x(k))L^{-1}\phi(x(k))}\phi(x(k)), \qquad (4.70)$$

який для звичайної ридж-оцінки здобуває адитивну форму алгоритму Качмажа [302]

$$w_j(k+1) = w_j(k) - \frac{d_j(k) - w_j^T(k)\phi(x(k))}{l + \left\|\phi(x(k))\right\|^2}\phi(x(k)). \qquad (4.71)$$

Додаткову гнучкість розглянутим процедурам можна забезпечити, передбачивши можливість варіювання параметра регуляризації l в процесі навчання. Питання вибору цього параметра досить докладно висвітлені в [295], зазначимо лише, що робота в реальному часі обмежує клас результатів операційними оцінками, у яких параметр l розраховується на підставі отримуваних вибіркових значень дисперсії діючих збурень. При цьому алгоритм (4.69) слід доповнити співвідношеннями для скидання застарілого значення $l(k-1)$ і введення нового $l(k)$:

$$\begin{cases} w_j(k+1) = w_j(k) + \dfrac{P_\phi(k-1)\left(d_j(k) - w_j^T(k)\phi(x(k))\right)}{1+\phi^T(x(k))P_\phi(k-1)\phi(x(k))}\phi(x(k)), \\ \tilde{P}_\phi(k-1) = \tilde{D}(k-1) + \dfrac{\tilde{D}(k-1)\phi(x(k-\chi))\phi^T(x(k-\chi))\tilde{D}(k-1)}{1-\phi^T(x(k-\chi))\tilde{D}(k-1)\phi(x(k-\chi))}, \\ \tilde{D}(k) = D(k-1) + l(k-1)D(k-1)\left(\mathrm{I} - l(k-1)D(k-1)\right)^{-1}D(k-1), \\ P_\phi(k) = \tilde{P}_\phi(k-1) - \dfrac{\tilde{P}_\phi(k-1)\phi(x(k))\phi^T(x(k))\tilde{P}_\phi(k-1)}{1+\phi^T(x(k))\tilde{P}_\phi(k-1)\phi(x(k))}, \\ D(k) = P_\phi(k) + l(k)P_\phi(k)\left(\mathrm{I} + l(k)P_\phi(k)\right)^{-1}P_\phi(k), \\ w_j(0) = 0, \quad P_\phi(0) = \tilde{P}_\phi(0) = l^{-1}(0)\mathrm{I}. \end{cases} \qquad (4.72)$$

Особливістю алгоритму (4.72) є те, що при $\chi \le h$ він переходить у форму регуляризованого багатокрокового алгоритму, а отримувані за його допомогою оцінки є незміщеними. При $\chi > h$ (4.72) є стійкою модифікацією поточного методу найменших квадратів. Варіюючи регуляризуючою добавкою по ходу навчання, можна отримувати різні форми рекурентних процедур настроювання синаптичних ваг нейронних мереж.

Більшість розглянутих вище алгоритмів настроювання тим чи іншим чином пов’язані з квадратичними критеріями. Оскільки досить часто навчання відбувається в умовах інтенсивних завад, доцільно більш докладно зупинитися на багатокрокових процедурах. Відомо [250], що квадратичний критерій (4.31) дозволяє отримати оптимальну якість навчання у випадку, коли завади

$\xi(k)$, $k=0,1,2,\ldots$ розподілені за нормальним законом чи за розподілом (в більш загальному випадку) з обмеженою дисперсією, тобто

$$\int_{-\infty}^{\infty}\xi^2 p(\xi)d\xi=\sigma_\xi^2<\infty, \tag{4.73}$$

де $p(\xi)$ – щільність розподілу завад, що, як правило, невідома. Існує досить багато розподілів, які не входять у цей клас, наприклад, так звані розподіли з «важкими хвостами». Збурення, що мають такий розподіл, характеризуються можливістю виникнення викидів, що можуть внести перекручування в процес навчання.

Найбільш велика група розподілів може бути описана за допомогою «класу невироджених розподілів», до якого входять усі розподіли з

$$p(0)=\frac{1}{2a}>0, \tag{4.74}$$

і «класу приблизно нормальних розподілів», при цьому елементи цього класу мають щільності

$$p(\xi)=(1-\varepsilon)\,p_N(\xi)+\varepsilon q(\xi), \tag{4.75}$$

де $p_N(\xi)$ – щільність нормального закону розподілу $N(0,\sigma_\xi^2)$, $q(\xi)$ – довільна щільність, $0\le\varepsilon<1$ – параметр ступеня «забруднення» основного розподілу $p_N(\xi)$.

Для кожного класу існують найгірші (у сенсі фішерівської інформації $I(p^*): p^*=\min I(p)$) розподілу:

– для класу вироджених розподілів – розподіл Лапласа

$$p^*(\xi)=\frac{1}{2a}\exp\left\{-\frac{|\xi|}{a}\right\}, \tag{4.76}$$

– для класу розподілів з обмеженою дисперсією – нормальний розподіл

$$p^*(\xi)=\frac{1}{\sqrt{2\pi\sigma_\xi^2}}\exp\left\{-\frac{\xi^2}{2\sigma_\xi^2}\right\}, \tag{4.77}$$

– для класу приблизно нормальних розподілів – комбінація нормального і лапласівського розподілів

$$p^*(\xi)=\begin{cases}\dfrac{1-\varepsilon}{\sqrt{2\pi\sigma_\xi^2}}\exp\left\{-\dfrac{\xi^2}{2\sigma_\xi^2}\right\} & \text{при} \quad |\xi|\le\varepsilon_1,\\[2ex] \dfrac{1-\varepsilon}{\sqrt{2\pi\sigma_\xi^2}}\exp\left\{-\dfrac{\xi_1^2}{2\sigma_\xi^2}\right\}\exp\left\{-\dfrac{\varepsilon_1|\xi|}{\sigma_\xi^2}\right\} & \text{у іншому випадку,}\end{cases} \tag{4.78}$$

де значення ε_1 знаходиться за допомогою рівняння

$$\frac{1}{1-\varepsilon}=\int_{-\varepsilon_1}^{\varepsilon_1}p_N(x)dx+2p_N(\varepsilon_1)\frac{\sigma_\xi^2}{\varepsilon_1}. \tag{4.79}$$

Відповідно до конкретного $p^*(\xi)$ отримуємо такі критерії якості навчання:

– для класу вироджених розподілів

$$E_j^k = \sum_{p=0}^{k} \left| e_j(p) \right|, \tag{4.80}$$

– для класу розподілів з обмеженою дисперсією

$$E_j^k = \sum_{p=0}^{k} e_j^2(p), \tag{4.81}$$

– для класу приблизно нормальних розподілів

$$E_j^k = \sum_{p=0}^{k} f\left(e_j(p)\right), \tag{4.82}$$

де

$$f\left(e_j(p)\right) = \begin{cases} \dfrac{1}{2\sigma_\xi^2} e_j^2(p) \quad \text{при} \quad \left|e_j(p)\right| \le \varepsilon_1, \\ -\dfrac{\varepsilon_1^2}{2\sigma_\xi^2} + \dfrac{\left|e_j(p)\right|\varepsilon_1}{\sigma_\xi^2} \quad \text{в іншому випадку.} \end{cases} \tag{4.83}$$

Відповідні критеріям (4.80) – (4.82) алгоритми з експоненційним зважуванням інформації можуть бути записані у формі [267]:

$$\begin{cases} w_j(k+1) = w_j(k) + \eta(k) P_\phi(k) e_j(k) \phi\left(x(k)\right), \\ P_\phi(k) = \dfrac{1}{\alpha}\left(P_\phi(k-1) - \dfrac{P_\phi(k-1)\phi\left(x(k)\right)\phi^T\left(x(k)\right)P_\phi(k-1)}{\alpha\eta^{-1}(k) + \phi^T\left(x(k)\right)P_\phi(k-1)\phi\left(x(k)\right)} \right), \end{cases} \tag{4.84}$$

де $\eta(k)$ залежить від прийнятого критерію і має вигляд:

– для класу вироджених розподілів

$$\eta(k) = \begin{cases} \varepsilon_2^{-1} \quad \text{при} \quad \left|e_j(k)\right| \le \varepsilon_2, \\ \left|e_j(k)\right|^{-1} \quad \text{в іншому випадку,} \end{cases} \tag{4.85}$$

(тут ε_2 – мала ненегативна величина),

– для класу розподілів з обмеженою дисперсією

$$\eta(k) = 1, \tag{4.86}$$

– для класу приблизно нормальних розподілів

$$\eta(k) = \begin{cases} 1 \quad \text{при} \quad \left|e_j(k)\right| \le \varepsilon_1, \\ \varepsilon_1\left|e_j(k)\right|^{-1} \quad \text{в іншому випадку.} \end{cases} \tag{4.87}$$

Вирази (4.84), (4.85) відповідають рекурентному алгоритму найменших модулів, (4.84), (4.86) – експоненційно зваженому методу найменших квадратів, (4.84), (4.87) – адаптивному робастному алгоритму, що увібрав у себе високий рівень стійкості методу найменших модулів і високу швидкість збіжності методу найменших квадратів.

У ряді практичних ситуацій про збурення немає взагалі ніякої інформації, крім їхньої приналежності деякому обмеженому інтервалу

$$\left|\xi(k)\right| \le r(k), \quad k = 0,1,2,\ldots. \tag{4.88}$$

Більш того, ці збурення можуть мати регулярний детермінований характер чи штучну природу типу навмисних перешкод. Зрозуміло, що навіть оптимальні значення синаптичних ваг w_j у цьому випадку не дозволяють отримати на виході нейромережі точне значення $y_j(k) = d_j(k)$, а можуть лише задати деякий інтервал [303–312]

$$d_j(k) - r(k) \le w_j^T \phi\left(x(k)\right) \le d_j(k) + r(k). \tag{4.89}$$

Нескладно помітити, що нерівність (4.89) визначає в просторі синаптичних ваг пари гіперповерхонь, між якими і лежать параметри, що настроюються, $w_j(k)$. Послідовність навчальних сигналів $d_j(0), d_j(1),\ldots, d_j(N)$ породжує $N+1$ пар гіперплощин, що висікають у цьому просторі деяку область (політоп) D_N. Це і є область параметрів, що уточнюються, при цьому всі точки, які належать цій області, рівноправні в тому сенсі, що серед них неможливо виділити найкращий вектор ваг, хоча для зручності можна використовувати деякий центр області D_N. Очевидно, що результатом навчання буде не традиційна точкова оцінка, а інтервальна, яка в ряді випадків буває дуже зручно.

Перший і очевидний шлях вирішення задачі полягає в знаходженні рішення системи $N+1$ лінійних нерівностей (4.89), проте оскільки кількість вершин політопу D_N зростає значно швидше, ніж $k = 0,1,\ldots,N,\ldots$, з обчислювальної точки зору цей підхід є малоефективним.

Альтернативний підхід полягає в апроксимації політопа D_k, отриманого в k-й момент часу, еліпсоїдом

$$E_k : \left(w_j - w_j(k)\right)^T P^{-1}(k)\left(w_j - w_j(k)\right) \le 1, \tag{4.90}$$

чий центр $w_j(k)$ і симетрична додатно визначена матриця $P(k)$ настроюються так, щоб E_k був якомога «ближче» до D_k. Оскільки $w_j(k)$ і $P(k)$ містять $(h+1) + \dfrac{(h+2)(h+1)}{2}$ параметрів, що настроюються, ідея використання еліпсоїдів порівняно з політопами є кращою.

Підхід, запропонований Ф. Швеппе [303], полягає в тому, що еліпсоїд E_k має містити всі можливі значення параметрів, що належать перетину E_{k-1} (еліпсоїд, побудований у $(k-1)$-й момент часу) з областю G_k, що лежить між двома гіперплощинами останньої k-ї нерівності (4.89) так, як це показано на рис. 4.9.

Оскільки перетин E_{k-1} і G_k не є еліпсоїдом, необхідно так побудувати $w_j(k)$ і $P(k)$, щоб E_k максимально точно його апроксимував. Об'єднавши (4.89) і (4.90), нескладно бачити, що шукані параметри описуються системою нерівностей

$$\begin{cases}\left(w_j - w_j(k-1)\right)^T P^{-1}(k-1)\left(w_j - w_j(k-1)\right) \le 1, \\ r^{-2}(k)\left(d_j(k) - w_j^T \phi\left(x(k)\right)\right)^2 \le 1,\end{cases} \tag{4.91}$$

або для деякого ненегативного $\rho(k)$ –

$$\begin{aligned}&\left(w_j - w_j(k-1)\right)^T P^{-1}(k-1)\left(w_j - w_j(k-1)\right) + \\ &+\rho(k) r^{-2}(k)\left(d_j(k) - w_j^T \phi\left(x(k)\right)\right)^2 \le 1 + \rho(k).\end{aligned} \tag{4.92}$$

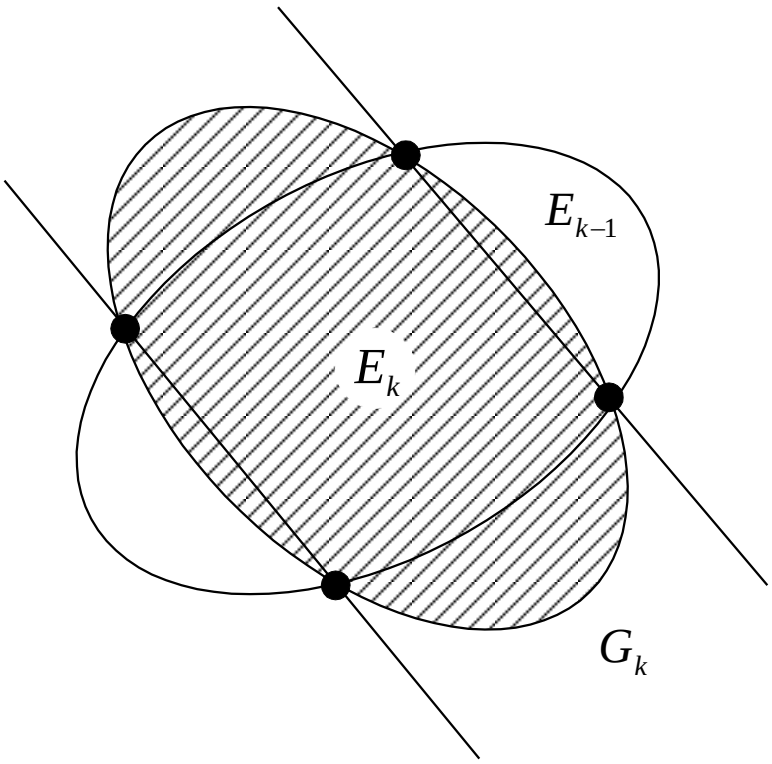


Рис. 4.9. Апроксимація перетину еліпсоїда парою гіперплощин

Уводячи вектор відхилень $\tilde{w}_j(k)$ (4.11), після нескладних, але громіздких перетворень квадратичної форми в лівій частині (4.92), приходимо до алгоритму оцінювання Фогеля–Хуанга [306]

$$\begin{cases} w_j(k+1) = w_j(k) + \rho(k) r^{-2}(k) \tilde{P}(k)\left(d_j(k) - w_j^T(k)\phi\left(x(k)\right)\right)\phi\left(x(k)\right), \\ \tilde{P}(k) = P(k-1) - \dfrac{\rho(k) r^{-2}(k) P(k-1)\phi\left(x(k)\right)\phi^T\left(x(k)\right)P(k-1)}{1 + \rho(k) r^{-2}(k)\phi^T\left(x(k)\right)P(k-1)\phi\left(x(k)\right)}, \\ P(k) = \tilde{P}(k-1)\left(1 + \rho(k) - \dfrac{\rho(k)\left(d_j(k) - w_j^T(k)\phi\left(x(k)\right)\right)^2}{r^2(k) + \rho(k)\phi^T\left(x(k)\right)P(k-1)\phi\left(x(k)\right)}\right), \end{cases} \tag{4.93}$$

що є різновидом зваженого методу найменших квадратів.

Процедура (4.93) містить невизначений параметр $\rho(k)$, який вибирається так, щоб об'єм еліпсоїда E_k на перетинанні E_{k-1} і G_k був мінімальним. Ця задача пов'язана з пошуком у кожен момент k мінімуму функції

$$\det P(k) = \left(1+\rho(k)-\frac{\rho(k)e_j^2(k)}{r^2(k)+\rho(k)\phi^T\left(x(k)\right)P(k-1)\phi\left(x(k)\right)}\right)^{h+1} \times$$
$$\times\left(1-\frac{\rho(k)\phi^T\left(x(k)\right)P(k-1)\phi\left(x(k)\right)}{r^2(k)+\rho(k)\phi^T\left(x(k)\right)P(k-1)\phi\left(x(k)\right)}\right)\det P(k-1) \tag{4.94}$$

або, що те саме, з рішенням диференціального рівняння

$$\frac{\partial \det P(k)}{\partial \rho} = 0. \tag{4.95}$$

Оскільки (4.95) не має аналітичного рішення, необхідно скористатися процедурою одновимірного пошуку глобального мінімуму (4.94) або чисельною процедурою пошуку дійсних ненегативних коренів (4.95).

Введенням змінних

$$\begin{cases} \alpha(k) = \rho^{-1}(k)r^2(k), \\ \beta(k) = 1+\dfrac{r^2(k)}{\alpha(k)}-\dfrac{e_j^2(k)}{\alpha(k)+\phi^T\left(x(k)\right)P(k-1)\phi\left(x(k)\right)} \end{cases} \tag{4.96}$$

алгоритм (4.93) може бути перетворений до форми [313]

$$\begin{cases} w_j(k+1) = w_j(k)+\dfrac{P(k-1)\left(d_j(k)-w_j^T(k)\phi\left(x(k)\right)\right)}{\alpha(k)+\phi^T\left(x(k)\right)P(k-1)\phi\left(x(k)\right)}\phi\left(x(k)\right), \\ P(k) = \beta(k)\left(P(k-1)-\dfrac{P(k-1)\phi\left(x(k)\right)\phi^T\left(x(k)\right)P(k-1)}{\alpha(k)+\phi^T\left(x(k)\right)P(k-1)\phi\left(x(k)\right)}\right), \end{cases} \tag{4.97}$$

структурно співпадаючої з експоненційно зваженим рекурентним методом найменших квадратів, але істотно відрізняється своїми властивостями, а крім того, вимагає на кожному такті k вирішення задачі мінімізації по $\alpha(k)$ функції

$$\det P(k) = \left(1+\frac{r^2(k)}{\alpha(k)}-\frac{e_j^2(k)}{\alpha(k)+\phi^T\left(x(k)\right)P(k-1)\phi\left(x(k)\right)}\right)^{h+1} \times$$
$$\times\left(1-\frac{\phi^T\left(x(k)\right)P(k-1)\phi\left(x(k)\right)}{\alpha(k)+\phi^T\left(x(k)\right)P(k-1)\phi\left(x(k)\right)}\right). \tag{4.98}$$

Необхідність мінімізації цієї функції істотно ускладнює процес навчання.

Для спрощення алгоритму введемо в розгляд скалярну змінну $\gamma(k)$ таку, що

$$\begin{cases} D^{-1}(k) = \gamma(k)P^{-1}(k), \\ D(k) = \gamma^{-1}(k)P(k), \\ \gamma(k) > 0, \end{cases} \tag{4.99}$$

після чого перепишемо (4.91), (4.92) у вигляді

$$\begin{cases} \left(w_j - w_j(k-1)\right)^T D^{-1}(k-1)\left(w_j - w_j(k-1)\right) \le \gamma(k-1), \\ r^{-2}(k)\left(d_j(k) - w_j^T \phi\left(x(k)\right)\right)^2 \le 1, \\ \left(w_j - w_j(k-1)\right)^T D^{-1}(k-1)\left(w_j - w_j(k-1)\right) + \\ + \rho(k) r^{-2}(k)\left(d_j(k) - w_j^T \phi\left(x(k)\right)\right)^2 \le \gamma(k-1) + \rho(k). \end{cases} \tag{4.100}$$

Після перетворень (4.100), можна отримати процедуру вигляду

$$\begin{cases} w_j(k+1) = w_j(k) + \delta(k) e_j(k) D(k) \phi\left(x(k)\right), \\ D(k) = D(k-1) - \delta(k) \dfrac{D(k-1)\phi\left(x(k)\right)\phi^T\left(x(k)\right)D(k-1)}{1 + \delta(k)\phi^T\left(x(k)\right)D(k-1)\phi\left(x(k)\right)} \end{cases} \tag{4.101}$$

(тут $\delta(k) = \rho(k) r^{-2}(k) = \alpha^{-1}(k)\big)$, структурно співпадаючу з алгоритмом Хегглунда [314], що мінімізує цільову функцію

$$E_j^k = \sum_{p=0}^{k} \delta(p) e_j^2(p). \tag{4.102}$$

Ця процедура відрізняється від алгоритму Хегглунда наявністю двох параметрів $\gamma(k)$ і $\delta(k)$, що цілком визначають її властивості. Можна показати [239, 315], що алгоритм

$$\begin{cases} w_j(k+1) = w_j(k) + \gamma(k)\delta(k) e_j(k) P(k) \phi\left(x(k)\right), \\ P(k) = \dfrac{\gamma(k)}{\gamma(k-1)}\left(P(k-1) - \delta(k) \dfrac{P(k-1)\phi\left(x(k)\right)\phi^T\left(x(k)\right)P(k-1)}{\gamma(k-1) + \delta(k)\phi^T\left(x(k)\right)P(k-1)\phi\left(x(k)\right)} \right), \end{cases} \tag{4.103}$$

у якого ці параметри задовольняють співвідношенням

$$\begin{gathered} \gamma(k) = \gamma(k-1) + \delta(k) r^2(k) - \frac{\gamma(k-1)\delta(k) e_j^2(k)}{\gamma(k-1) + \delta(k)\phi^T\left(x(k)\right)P(k-1)\phi\left(x(k)\right)}, \\ 0 < \delta(k) \le \gamma(k-1) \frac{\dfrac{e_j^2(k)}{r^2(k)} - 1}{\phi^T\left(x(k)\right)P(k-1)\phi\left(x(k)\right)}, \end{gathered} \tag{4.104}$$

забезпечує збіжність ваг, що настроюються, до еліпсоїдів мінімального об’єму, що містить оптимальні параметри, не вимагаючи при цьому вирішення допоміжних задач чи оптимізації пошуку коренів. Алгоритм досить простий в обчислювальному відношенні і, в міру накопичення інформації в процесі настроювання, поступово приймає форму зваженого рекурентного методу найменших квадратів, досить популярного в задачах навчання ШНМ [238].

4.4 Нелінійні алгоритми навчання

У цьому підрозділі розглянуто алгоритми, також засновані на парадигмі навчання із вчителем, що реалізують правило корекції за помилкою, однак сама помилка навчання

$$e_j(k) = d_j(k) - y_j(k) = d_j(k) - \psi\left(\left(w_j^T(k)x(k)\right)\right) \tag{4.105}$$

у цьому випадку є нелінійною функцією синаптичних ваг і визначається прийнятою активаційною функцією $\psi(\bullet)$.

Типовим прикладом такого алгоритму може слугувати процедура навчання персептрона Розенблатта із сигнум-функцією активації. Як цільова функція використовується вираз [316]

$$\begin{aligned} E_j(k) &= d_j(k)u_j(k) - \left|u_j(k)\right| = \\ &= e_j(k)u_j(k) = \left(d_j(k) - sign w_j^T x(k)\right)w_j^T x(k), \end{aligned} \tag{4.106}$$

а алгоритму настроювання – співвідношення (4.10), що з урахуванням того, що

$$\nabla_{w_j} E_j(k) = -e_j(k)x(k), \tag{4.107}$$

набуває простої форми

$$\begin{aligned} w_j(k+1) &= w_j(k) + \eta e_j(k)x(k) = \\ &= w_j(k) + \eta\left(d_j(k) - sign w_j^T(k)x(k)\right)x(k), \end{aligned} \tag{4.108}$$

де навчальний $d_j(k)$ і вихідний $y_j(k) = sign u_j(k)$ сигнали нейрона можуть приймати тільки два значення $+1$ і -1.

Для того, щоб виключити вплив на процес збіжності амплітуди вхідного сигналу, може бути використано модифікацію (4.108), яка має вигляд [239]

$$w_j(k+1) = w_j(k) + \eta\frac{d_j(k) - sign\, w_j^T(k)x(k)}{\left\|x(k)\right\|^2}x(k), \tag{4.109}$$

однак, відрізняється за властивостями як від алгоритму Качмажа–Уідроу–Гоффа (4.18), так і від алгоритму навчання (4.28).

Розглянемо далі навчання квадратичного нейрона, описаного в [265] і здійснюючого перетворення

$$y_j(k) = \theta_j(k) + \sum_{i=1}^{n} w_{ji}(k)x_i(k) + \sum_{p=1}^{n}\sum_{l=1}^{n} w_{jpl}(k)x_p(k)x_l(k), \tag{4.110}$$

яке з урахуванням позначень $w_{j0}(k) = \theta_j(k)$, $b_j(k) = \left(w_{j1}(k), w_{j2}(k), \ldots, w_{jn}(k)\right)^T$ – $(n\times 1)$-вектор, $C_j(k) = \left\{w_{jpl}(k)\right\}$ – $(n\times n)$-матриця, $x_{\_}(k) = \left(x_1(k), x_2(k), \ldots, x_n(k)\right)^T$ – $(n\times 1)$-вектор, $x(k) = \left(1, x_{\_}^T(k)\right)^T$, можна переписати у вигляді

$$y_j(k) = w_{j0}(k) + b_j^T(k)x_{\_}(k) + x_{\_}^T(k)C_j(k)x_{\_}(k) \tag{4.111}$$

або в ще більш компактній формі

$$y_j(k) = x^T(k)W_j(k)x(k), \tag{4.112}$$

де

$$W_j(k)=\left(\begin{array}{c|c} w_{j0}(k) & 0.5b_j^T(k) \\ \hline 0.5b_j(k) & C_j(k) \end{array}\right) \tag{4.113}$$

– блокова $(n+1)\times(n+1)$-матриця.

Настроювання матриці синаптичних ваг W_j здійснюватиме шляхом мінімізації критерію

$$E_j(k)=\frac{1}{2}e_j^2(k)=\frac{1}{2}\left(d_j(k)-x^T(k)W_jx(k)\right)^2 \tag{4.114}$$

за допомогою градієнтної процедури

$$W_j(k+1)=W_j(k)+\eta(k)e_j(k)x(k)x^T(k), \tag{4.115}$$

де

$$e_j(k)=d_j(k)-x^T(k)W_j(k)x(k). \tag{4.116}$$

Для пошуку параметра $\eta(k)$, що забезпечує алгоритму (4.115) оптимальні властивості, введемо матрицю відхилень поточних значень $W_j(k)$ від оптимальних

$$\tilde{W}_j(k)=W_j-w_j(k) \tag{4.117}$$

після чого, вирішуючи диференціальне рівняння

$$\frac{\partial Tr\left(\tilde{W}_j(k)\tilde{W}_j^T(k)\right)}{\partial\eta}=0, \tag{4.118}$$

(тут $Tr(\bullet)$ – символ сліду матриці), можна отримати оптимальне значення параметра кроку у вигляді [317]

$$\eta(k)=\left\|x(k)\right\|^{-4}. \tag{4.119}$$

Підстановка (4.119) у (4.115) приводить до алгоритму навчання

$$W_j(k+1)=W_j(k)+\frac{d_j(k)-x^T(k)W_j(k)x(k)}{\left\|x(k)\right\|^4}x(k)x^T(k), \tag{4.120}$$

що є розширенням алгоритму Качмажа–Уідроу–Гоффа на квадратичний нейрон.

У [317] вивчено збіжність цього алгоритму і запропоновані різні його модифікації, включаючи багатокрокові процедури.

Насьогодні у нейронних мережах найбільшого поширення отримали сигмоїдальні активаційні функції типу уніполярної сигмоїди

$$\psi\left(\gamma_j u_j\right)=\sigma\left(\gamma_j u_j\right)=\frac{1}{1+e^{-\gamma_j u_j}} \tag{4.121}$$

і біполярного гіперболічного тангенсу

$$\psi\left(\gamma_j u_j\right)=\tanh\left(\gamma_j u_j\right)=\frac{1-e^{-2\gamma_j u_j}}{1+e^{-2\gamma_j u_j}}, \tag{4.122}$$

які пов’язані між собою. Зазначимо також, що за великих значень параметра крутості γ_j вони практично збігаються з релейною і сигнум-функцією відповідно.

Процес навчання у неперервному часі (або, що те саме, мінімізація цільової функції (4.4)) може бути реалізований за допомогою градієнтного спуску, описуваного системою диференціальних рівнянь

$$\frac{dw_{ji}}{dt} = -\eta \frac{\partial E_j(t)}{\partial w_{ji}} = -\eta \frac{\partial E_j(t)}{\partial e_j} \frac{\partial e_j}{\partial w_{ji}}, \tag{4.123}$$

або з урахуванням того, що

$$e_j = d_j - y_j = d_j - \psi\left(u_j\right) = d_j - \psi\left(\sum_{i=0}^{n} w_{ji} x_i\right), \tag{4.124}$$

системою

$$\begin{aligned} &\frac{dw_{ji}}{dt} = -\eta e_j \frac{\partial e_j}{\partial w_{ji}} = -\eta e_j \frac{\partial e_j}{\partial u_j} \frac{\partial u_j}{\partial w_{ji}} = \\ &= \eta e_j \frac{\partial \psi\left(u_j\right)}{\partial u_j} x_i = \eta e_j \psi'\left(u_j\right) x_i = \eta \delta_j x_i, \end{aligned} \tag{4.125}$$

де δ_j – так звана локальна помилка, яка виражається у вигляді

$$\delta_j = e_j \psi'\left(u_j\right) = -\frac{\partial E_j(t)}{\partial u_j}. \tag{4.126}$$

Якщо як активаційна функціа використовується сигмоїда (4.121), то

$$\psi'\left(u_j\right) = \frac{\partial \psi\left(u_j\right)}{\partial u_j} = \gamma_j y_j \left(1 - y_j\right), \tag{4.127}$$

а рівняння (4.125) набувають вигляду

$$\frac{dw_{ji}}{dt} = \eta \gamma_j e_j y_j \left(1 - y_j\right) x_i, \tag{4.128}$$

при цьому похідна $\partial y_j / \partial u_j$ досягає максимуму при $y_j = 0.5$ і мінімуму – коли y_j знаходяться в околі 0 чи 1.

У випадку гіперболічного тангенса

$$\psi'\left(u_j\right) = \frac{\partial \psi\left(u_j\right)}{\partial u_j} = \gamma_j \left(1 - \left(\tanh \gamma_j u_j\right)^2\right) = \gamma_j \left(1 - y_j^2\right) \tag{4.129}$$

навчання відбувається відповідно до диференціальних рівнянь

$$\frac{dw_{ji}}{dt} = \eta \gamma_j e_j \left(1 - y_j^2\right) x_i, \tag{4.130}$$

при цьому настроювання ваг практично зупиняється, якщо y_j прямує до -1 або $+1$, оскільки похідна $\partial y_j / \partial u_j$, що дорівнює $\gamma_j \left(1 - y_j^2\right)$, досягає свого максимуму при $y_j = 0$ і мінімуму – при ± 1.

У дискретному випадку навчання відбувається шляхом мінімізації критерію

$$E_j(k)=\frac{1}{2}e_j^2(k)=\frac{1}{2}\big(d_j(k)-y_j(k)\big)^2=\frac{1}{2}\big(d_j(k)-\psi\big(u_j(k)\big)\big)^2=$$
$$=\frac{1}{2}\left(d_j(k)-\psi\left(\sum_{i=0}^{n}w_{ji}x_i(k)\right)\right)^2 \qquad (4.131)$$

за допомогою рекурентної процедури

$$w_{ji}(k+1)=w_{ji}(k)-\eta(k)\frac{\partial E_j(k)}{\partial e_j(k)}\frac{\partial e_j(k)}{\partial w_{ji}}=$$
$$=w_{ji}(k)-\eta(k)e_j(k)\frac{\partial e_j(k)}{\partial w_{ji}}=w_{ji}(k)-\eta(k)e_j(k)\frac{\partial e_j(k)}{\partial u_j(k)}\frac{\partial u_j(k)}{\partial w_{ji}}= \qquad (4.132)$$
$$=w_{ji}(k)+\eta(k)e_j(k)\psi'\big(u_j(k)\big)x_i(k)=w_{ji}(k)+\eta(k)\delta_j(k)x_i(k),$$

де

$$\delta_j(k)=e_j(k)\psi'\big(u_j(k)\big)=-\frac{\partial E_j(k)}{\partial u_j} \qquad (4.133)$$

– локальна похибка.

У векторній формі алгоритм (4.132) має вигляд

$$w_j(k+1)=w_j(k)+\eta(k)\delta_j(k)x(k), \qquad (4.134)$$

що набув у теорії і практиці штучних нейронних мереж широкого поширення під ім’ям дельта-правила навчання.

Для сигмоїди (4.121) цей алгоритм має форму

$$w_j(k+1)=w_j(k)+\eta(k)\gamma_j e_j(k)y_j(k)\big(1-y_j(k)\big)x(k), \qquad (4.135)$$

а для активаційної функції (4.122) –

$$w_j(k+1)=w_j(k)+\eta(k)\gamma_j e_j(k)\big(1-y_j^2(k)\big)x(k). \qquad (4.136)$$

На рис 4.10 наведено схему навчання нейрона за допомогою дельта-правила (4.134).

Реалізація цієї схеми вимагає досить точного обчислення активаційних функцій і їхніх похідних, форма яких істотно залежить від параметра γ_j, який у загальному випадку також може бути таким, що підлягає настроюванню.

У деяких застосуваннях обчислення похідних ускладнене, у зв’язку з чим було запропоновано альтернативний алгоритм навчання [258], що не використовує операцію диференціювання.

У цьому випадку генерується малий зондувальний сигнал збурювання $\Delta u_j(k)$, який накладається на сигнал $u_j(k)$ для того, щоб оцінити миттєве значення градієнта функції помилки. Ефект малого збурювання на помилку $e_j(k)=d_j(k)-y_j(k)$ запам’ятовується.

При цьому, оскільки

$$\frac{\partial E_j(k)}{\partial w_{ji}}=\frac{1}{2}\frac{\partial e_j^2(k)}{\partial w_{ji}}=\frac{1}{2}\frac{\partial e_j^2(k)}{\partial u_j(k)}\frac{\partial u_j(k)}{\partial w_{ji}}=-e_j(k)\frac{\partial e_j(k)}{\partial u_j(k)}x_i(k), \qquad (4.137)$$

то для малого змінення $\Delta u_j(k)$ можна записати

$$\frac{\partial e_j^2(k)}{\partial u_j(k)} \approx \frac{\left(\Delta e_j(k)\right)^2}{\Delta u_j(k)} \tag{4.138}$$

або

$$\frac{\partial e_j^2(k)}{\partial u_j(k)} = 2e_j(k)\frac{\partial e_j(k)}{\partial u_j(k)} \approx 2e_j(k)\frac{\Delta e_j(k)}{\Delta u_j(k)}. \tag{4.139}$$

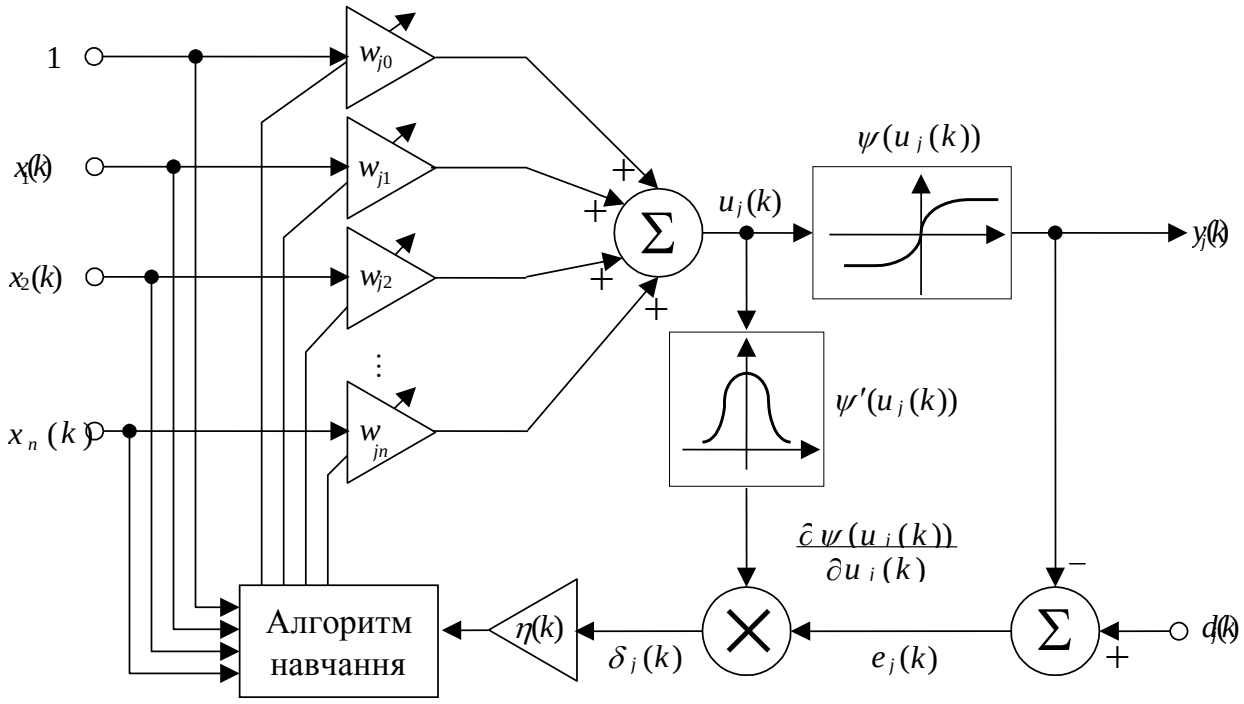


Рис. 4.10. Дельта-правило навчання

Використовуючи це співвідношення, можна ввести такі алгоритми навчання:

$$w_{ji}(k+1) = w_{ji}(k) - \frac{1}{2}\eta(k)\frac{\left(\Delta e_j(k)\right)^2}{\Delta u_j(k)}x_i(k) \tag{4.140}$$

і

$$w_{ji}(k+1) = w_{ji}(k) - \eta(k)e_j(k)\left(\frac{\Delta e_j(k)}{\Delta u_j(k)}\right)x_i(k) \tag{4.141}$$

практично ідентичні при малих $\Delta u_j(k)$.

Схему навчання, що реалізує алгоритми (4.140), (4.141) наведено на рис. 4.11.

Дельта-правило навчання нескладно поширити на критерії, відмінні від квадратичного, які у загальному випадку мають вигляд для неперервного і дискретного випадків відповідно

$$E_j(t) = f\left(e_j(t)\right) = f\left(d_j(t) - y_j(t)\right) \tag{4.142}$$

і

$$E_j(k) = f\left(e_j(k)\right) = f\left(d_j(k) - y_j(k)\right), \tag{4.143}$$

де $f\left(e_j\right)$ – деяка опукла диференційована функція втрат.

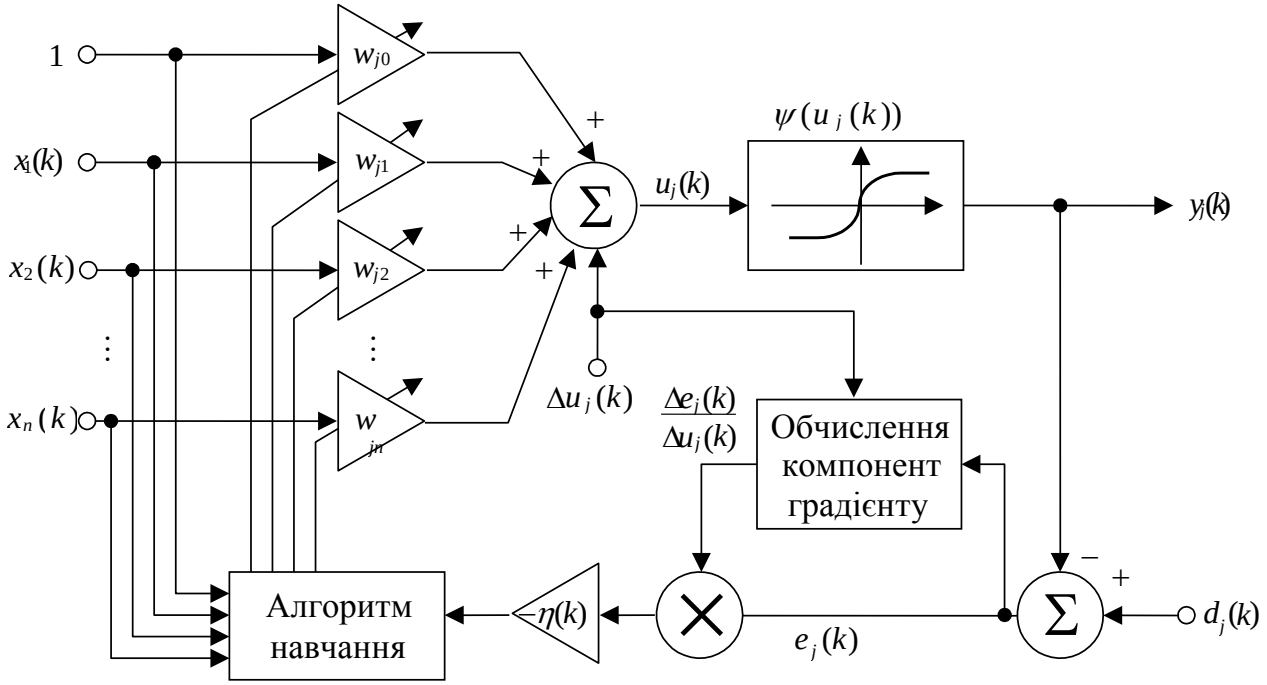


Рис. 4.11. Схема навчання за допомогою зондувальних сигналів

За аналогією з (4.123) і (4.125) можна записати

$$\frac{dw_{ji}}{dt} = -\eta \frac{\partial E_j(t)}{\partial w_{ji}} = -\eta \frac{\partial f}{\partial e_j}\frac{\partial e_j}{\partial y_j}\frac{\partial y_j}{\partial u_j}\frac{\partial u_j}{\partial w_{ji}} = \eta f'\left(e_j\right)\psi'\left(u_j\right)x_i = \eta\delta_j x_i, \tag{4.144}$$

де

$$\delta_j = \frac{\partial f\left(e_j\right)}{\partial e_j}\frac{\partial \psi\left(u_j\right)}{\partial u_j} = f'\left(e_j\right)\psi'\left(u_j\right). \tag{4.145}$$

У дискретній формі (4.144) має вигляд

$$\begin{aligned} w_j(k+1) &= w_j(k) + \eta(k)\, f'\left(e_j(k)\right)\psi'\left(u_j(k)\right)x(k) = \\ &= w_j(k) + \eta(k)\delta_j(k)x(k) \end{aligned} \tag{4.146}$$

і відрізняється від (4.134) тільки конструкцією локальної помилки $\delta_j(k)$.

Як функцію $f\left(e_j\right)$ у [258] розглянуто міру, пов'язану з ентропією, що веде до критерію

$$E_j(t) = \frac{1}{2}\left(1 + d_j(t)\right)\ln\frac{1 + d_j(t)}{1 + y_j(t)} + \frac{1}{2}\left(1 - d_j(t)\right)\ln\frac{1 - d_j(t)}{1 - y_j(t)}. \tag{4.147}$$

Цей критерій завжди додатний, крім випадку $y_j(t) = d_j(t)$ (ідеальне навчання). Приймаючи як активаційну функцію гіперболічного тангенса, можна отримати правило навчання

$$\frac{dw_{ji}}{dt} = \eta \delta_j x_i, \tag{4.148}$$

де локальна помилка має вигляд

$$\delta_j(t) = d_j(t) - y_j(t) = e_j(t). \tag{4.149}$$

Така сама проста форма у дискретному випадку має вигляд

$$w_j(k+1) = w_j(k) + \eta(k) e_j(k) x(k), \tag{4.150}$$

що пояснюється тим, що компонента $\psi'(u_j)$ після перетворень зникає, а це дозволяє використовувати для навчання всі алгоритми, розглянуті в підрозділі 4.3.

Узагальненням критеріїв (4.142) і (4.143) є конструкції типу [258]

$$E_j(t) = \frac{\alpha}{2}\|w_j(t)\|^2 + f(e_j(t)) \tag{4.151}$$

і

$$E_j(k) = \frac{\alpha}{2}\|w_j(k)\|^2 + f(e_j(k)), \tag{4.152}$$

де $\alpha \geq 0$, а як $f(\bullet)$ крім (4.147) може використовуватися, наприклад,

$$f(e_j) = |e_j| \tag{4.153}$$

або

$$f(e_j) = \beta^2 \cosh \frac{e_j}{\beta}, \tag{4.154}$$

(тут $\beta > 0$ – скалярний параметр).

Процес мінімізації (4.151) має вигляд

$$\frac{dw_{ji}}{dt} = \eta(-\alpha w_{ji} + \delta_j x_i), \tag{4.155}$$

де $\delta_j = \dfrac{\partial f(e_j)}{\partial e_j}$ – локальна помилка.

У дискретній формі (4.155) можна записати у вигляді рекурентної процедури

$$w_j(k+1) = w_j(k) + \eta(k)(-\alpha w_j(k) + \delta_j(k) x(k)). \tag{4.156}$$

Поліпшити апроксимуючі властивості нейронних мереж можна, вводячи додатково навчання параметра крутості γ_j, хоча в практичних випадках він покладається постійним. Для цього може бути використаний, наприклад, алгоритм Крушке–Мовеллана [318]

$$\gamma_j(k+1) = \gamma_j(k) - \eta_\gamma(k)\frac{\partial E_j(k)}{\partial \gamma_j} = \gamma_j(k) + \eta_\gamma(k) e_j(k) \frac{\partial \psi(\gamma_j(k) u_j(k))}{\partial \gamma_j}. \tag{4.157}$$

Об’єднання процедур навчання (4.134) і (4.157) дозволяє настроювати всі параметри мережі, хоча при цьому можуть виникнути деякі проблеми чисельної реалізації, пов’язані, насамперед, з необхідністю диференціювання активаційних функцій досить довільного вигляду.

Ці труднощі просто переборюються, якщо використовувати узагальнений формальний нейрон як базовий блок ШНМ. При цьому алгоритм навчання синаптичних ваг і параметра крутості може бути записаний у досить простому вигляді [319]

$$\begin{cases} w_{ji}(k+1) = w_{ji}(k) + \eta(k) e_j(k) \left(\sum_{l=0}^{L} (2l+1) \left(\gamma_j(k) u_j(k) \right)^{2l} \phi_l \gamma_j(k) \right) x_j(k), \\ \gamma_j(k+1) = \gamma_j(k) + \eta_\gamma(k) e_j(k) \left(\sum_{l=0}^{L} (2l+1) \left(\gamma_j(k) u_j(k) \right)^{2l} \phi_l u_j(k) \right). \end{cases} \tag{4.158}$$

Схему навчання на основі (4.158) наведено на рис 4.12.

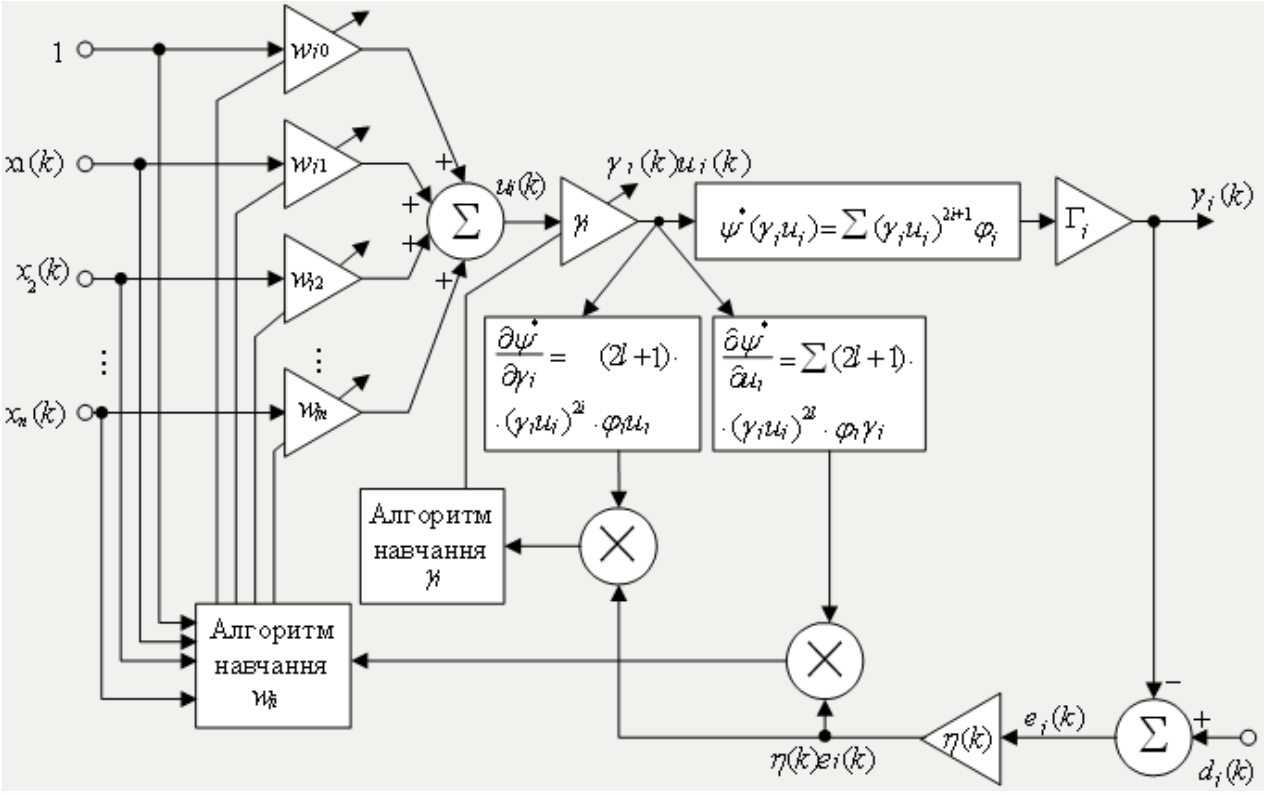


Рис. 4.12. Схема навчання узагальненого формального нейрона

Однією з проблем, що виникають у процесі нелінійного навчання, є проблема вибору параметра кроку $\eta(k)$, яка виявляється дуже гостро в ситуаціях, коли навчання відбувається або в обстановці перешкод, або в нестаціонарних умовах. Досить часто коефіцієнт $\eta(k)$ покладається фіксованим і однаковим для всіх нейронів мережі. Зазвичай його значення не перевищує одиниці для того, щоб уникнути небажаних коливань. Проте мале постійне значення параметра кроку зменшує швидкість збіжності, а, отже, збільшує загальний час навчання. Протиріччя між вимогами стійкості і високої

швидкості привело до виникнення цілого сімейства алгоритмів навчання з адаптивним вибором параметра кроку [258], що забезпечують високу швидкість збіжності за збереженням стійкості процесу настроювання мережі в цілому. При цьому необхідно відзначити, що більшість з відомих алгоритмів має евристичний характер.

У [320] розглянуто так звану «Search-Then-Converge» стратегію, відповідно до якої параметр кроку в процесі навчання поступово зменшується. На початковій стадії, званій фазою пошуку, швидкість навчання практично незмінна. На наступній стадії – фазі збіжності параметр кроку експоненційно прямує до нуля і розраховується відповідно до виразу

$$\eta(k)=\eta_0\frac{1}{1+k/k_0} \tag{4.159}$$

або

$$\eta(k)=\eta_0\frac{1+\dfrac{c}{\eta_0}\dfrac{k}{k_0}}{1+\dfrac{c}{\eta_0}\dfrac{k}{k_0}+k_0\left(\dfrac{k}{k_0}\right)^2}, \tag{4.160}$$

де $\eta_0>0,\quad c>0,\quad k_0>>1$ (зазвичао $100\le k_0\le 500$). Нескладно побачити, що при $k<<k_{0,}$, $\eta(k)\approx\eta_0$, а при $k>>k_0$ – зменшується пропорційно $1/k$, тобто задовольняє умовам Дворецького [276]. Таким чином, «Search-Then-Converge» – стратегія є процедурою стохастичної апроксимації, що забезпечує збіжність в обстановці інтенсивних завад.

Ще один підхід до навчання, заснований на стохастичній апроксимації, розвивається Б.Т. Поляком [321, 322]. Запропоновані ним рекурентні співвідношення для уточнення синаптичних ваг мають вигляд

$$\begin{cases} w_{ji}(k+1)=w_{ji}(k)-\eta(k)\dfrac{\partial E_j^k}{\partial w_{ji}}, \\ \bar{w}_{ji}(k+1)=\bar{w}_{ji}(k)+\bar{\eta}(k)\left(w_{ji}(k+1)-\bar{w}_{ji}(k)\right), \end{cases} \tag{4.161}$$

де

$$\frac{\partial E_j^k}{\partial w_{ji}}=\sum_k\frac{\partial E_j(k)}{\partial w_{ji}},\quad \eta(k)=\eta_0/k^{\gamma},\quad 0.5<\gamma\le 1,\quad \bar{\eta}(k)=1/(1+k).$$

Цей підхід поєднує в собі дві процедури. Перша – це рекурентна процедура стохастичної апроксимації з коефіцієнтом кроку $\eta(k)=\eta_0/k^{\gamma}$. Друга – це процес усереднення з коефіцієнтом $\bar{\eta}(k)=1/(1+k)$. На відміну від звичайного алгоритму навчання тут обчислюються дві послідовності ваг $w_{ji}(k)$ і $\bar{w}_{ji}(k)$, де $\bar{w}_{ji}(k)$ – це усереднене значення $w_{ji}(k)$. Алгоритм використовує два крокових коефіцієнти: $\eta(k)=\eta_0k^{-\gamma}$ і $\bar{\eta}(k)=(k+1)^{-1}$, причому коефіцієнт $\eta(k)$ убуває більш повільно, ніж $\bar{\eta}(k)$. На практиці процес усереднення має

починатися не при $k=0$, а з моменту $k \geq k_0$, для якого $w_{ji}(k)$ уже знаходиться в околі оптимальних значень, що веде до значень коефіцієнтів кроку

$$\eta(k) = \frac{\eta_0}{k_0} \frac{1}{(1+k/k_0)^{\gamma}} \tag{4.162}$$

і

$$\bar{\eta}(k) = \frac{1}{k-k_0} \tag{4.163}$$

для $k > k_0$. Як вказує автор методу, процес усереднення дозволяє підвищити швидкість збіжності в обстановці перешкод.

Один з найпростіших прийомів збільшення швидкості навчання полягає в тому, що кроковий коефіцієнт $\eta(k)$ збільшується, якщо глобальна цільова функція $E_j^k = \sum_k E_j(k)$ зменшується, і різко зменшується, якщо відбувається зростання критерію. В останньому випадку синаптичні ваги взагалі не уточнюються, тобто $w_{ji}(k+1) = w_{ji}(k)$. Таким чином, стратегію керування параметром кроку можна записати у вигляді [258]

$$\eta(k) = \begin{cases} a\eta(k-1), & \text{якщо } E_j^k < E_j^{k-1}, \\ b\eta(k-1), & \text{якщо } E_j^k \geq KE_j^{k-1}, \\ \eta(k-1), & \text{в інших випадках}, \end{cases} \tag{4.164}$$

де $a = 1.05, b = 0.7, K = 1.04$.

Шмидхубер запропонував [323] ще більш простий спосіб визначення параметра кроку в реальному часі

$$\eta(k) = \min\left\{ \frac{E_j(k) - E_j^*}{\left\| \nabla_{w_j} E_j(k) \right\|^2}, \eta_{\max} \right\}, \tag{4.165}$$

де $\eta_{\max}$ – максимально можливе значення параметра (зазвичай $\eta_{\max} = 20$), E_j^* – бажане значення цільової функції (звичайно $0.01 \leq E_j^* \leq 0.1$).

Чан і Фоллсайд розробили алгоритм навчання з регуляризуючим членом [324]

$$w_{ji}(k+1) = w_{ji}(k) + \eta(k)\left(-\frac{\partial E_j(k)}{\partial w_{ji}} + \beta(k)\Delta w_{ji}(k-1) \right), \tag{4.166}$$

де

$$\beta(k) = \beta_0 \frac{\Delta E_j(k-1)}{\left\| \Delta w_j(k-1) \right\|}; \tag{4.167}$$

$$\eta(k) = \eta(k-1)\left(1 + a\cos\Theta(k)\right); \tag{4.168}$$

$$\cos\Theta(k) = -\frac{\nabla_{w_j} E_j^T(k)\Delta w_j(k-1)}{\left\|\nabla_{w_j} E_j(k)\right\| \times \left\|\Delta w_j(k-1)\right\|}; \tag{4.169}$$

$$\Delta w_j(k-1) = w_j(k) - w_j(k-1); \quad \Delta E_j(k-1) = E_j(k) - E_j(k-1); \quad 0.1 \le a \le 0.5 \tag{4.170}$$

У [325, 326] було запропоновано як основу алгоритмів навчання використовувати метод сполучених градієнтів, що у загальному випадку має вигляд [322]

$$\begin{cases} w_j(k+1) = w_j(k) + \eta(k) s_j(k), \\ s(o) = -\nabla_{w_j} E_j((0), \\ s_j(k) = -\nabla_{w_j} E_j(k) + \beta(k) s_j(k-1) \end{cases} \tag{4.171}$$

і приводить до алгоритму Флетчера–Ривза при

$$\beta(k) = \frac{\nabla_{w_j} E_j^T(k) \nabla_{w_j} E_j(k)}{\nabla_{w_j} E_j^T(k-1) \nabla_{w_j} E_j(k-1)}, \tag{4.172}$$

алгоритму Полака–Риб'єра при

$$\beta(k) = \frac{\left(\nabla_{w_j} E_j(k) - \nabla_{w_j} E_j(k-1)\right)^T \nabla_{w_j} E_j(k)}{\nabla_{w_j} E_j^T(k-1) \nabla_{w_j} E_j(k-1)} \tag{4.173}$$

і алгоритму Гестенса–Штифеля при

$$\beta(k) = \frac{\left(\nabla_{w_j} E_j(k) - \nabla_{w_j} E_j(k-1)\right) \nabla_{w_j} E_j(k)}{s_j^T(k-1)\left(\nabla_{w_j} E_j(k) - \nabla_{w_j} E_j(k-1)\right)}. \tag{4.174}$$

У [327] для поліпшення збіжності процесу навчання було запропоновано використовувати такі евристики:

– кожна вага w_{ji} має власний параметр кроку η_{ji};

– параметр кроку адаптується по ходу процесу навчання на основі інформації про поточні значення похідних $\partial E_j(k)/\partial w_{ji}$;

– у випадку, якщо похідні $\partial E_j(k)/\partial w_{ji}$ кілька кроків підряд не змінюють знак, параметр кроку збільшується;

– у випадку, якщо похідні $\partial E_j(k)/\partial w_{ji}$ змінюють знак, параметр кроку експоненційно зменшується.

На базі цих евристик був запропонований алгоритм, відомий під ім'ям «Delta-Bar-Delta», що має вигляд

$$w_{ji}(k+1) = w_j(k) - \eta_{ji}(k)\frac{\partial E_j(k)}{\partial w_{ji}}, \tag{4.175}$$

де

$$\eta_{ji}(k)=\begin{cases}\eta_{ji}(k-1)+a, & \text{якщо } \delta_{ji}(k-1)\delta_{ji}(k)>0,\\ b\eta_{ji}(k-1), & \text{якщо } \delta_{ji}(k-1)\delta_{ji}(k)<0,\\ \eta_{ji}(k-1) & \text{в інших випадках;}\end{cases} \tag{4.176}$$

a – параметр адитивного збільшення (звичайно $10^{-4}\le a\le 0.1$); b – параметр мультиплікативного зменшення (звичайно $0.5\le b\le 0.9$); $\delta_{ji}(k)=\partial E_j(k)/\partial w_{ji}$;

$$\bar{\delta}_{ji}(k)=(1-\alpha)\delta_{ji}(k)+\alpha\bar{\delta}_{ji}(k-1),\quad 0\le\alpha<1. \tag{4.177}$$

Слід також зазначити, що, маючи підвищену швидкість збіжності «Delta-Bar-Delta» – алгоритм не допускає коливань у процесі навчання.

Своєрідною комбінацією алгоритму Чана–Фоллсайда і «Delta-Bar-Delta» є процедура Сильви–Алмейди, що має вигляд [328]

$$w_{ji}(k+1)=w_{ji}(k)-\eta_{ji}(k)\left(\frac{\partial E_j(k)}{\partial w_{ji}}+\beta\Delta w_{ji}(k-1)\right), \tag{4.178}$$

де

$$\eta_{ji}(k)=\begin{cases}a\eta_{ji}(k), & \text{якщо } \delta_{ji}(k)\delta_{ji}(k-1)\ge 0,\\ b\eta_{ji}(k) & \text{в інших випадках;}\end{cases} \tag{4.179}$$

$$1.1\le a\le 1.3;\quad 0.75\le b\le 0.9;\quad a\approx b^{-1};\quad \eta_{ji}(0)=10^{-3};\quad \beta=0.1.$$

Відповідно до цього алгоритму, якщо компоненти градієнта $\partial E_j/\partial w_{ji}$ мають один знак на сусідніх кроках, параметр кроку експоненційно зростає, а якщо відбувається зміна знака похідних – цей параметр зменшується.

У [329] розглянуто так званий «Super SAB»-алгоритм, у якому з незмінним знакуом похідних $\partial E_j/\partial w_{ji}$ на двох сусідніх кроках відбувається збільшення параметра кроку до досягнення ним максимального значення (зазвичай $\eta_{\max}=20$); у протилежному випадку уточнення ваг не відбувається. Ця процедура може бути записана в такий спосіб:

$$w_{ji}(k+1)=\begin{cases}w_{ji}(k)-\eta_{ji}(k)\dfrac{\partial E_j(k)}{\partial w_{ji}}+\beta\Delta w_{ji}(k-1), & \text{якщо } \delta_{ji}(k)\delta_{ji}(k-1)\ge 0,\\ w_{ji}(k) & \text{в інших випадках,}\end{cases} \tag{4.180}$$

де

$$\eta_{ji}(k)=\begin{cases}a\eta_{ji}(k-1), & \text{якщо } \dfrac{1}{2}\delta_{ji}(k)\delta_{ji}(k-1)>0 \text{ і } \eta_{ji}(k-1)\le\eta_{\max},\\ b\eta_{ji}(k-1) & \text{в інших випадках;}\end{cases} \tag{4.181}$$

$a\approx 1.05;\ 0.5\le b\le 0.7$.

Радміллером і Брауном був досліджений знаковий алгоритм навчання [330], відомий під ім'ям «RPROP», що має вигляд

$$w_{ji}(k+1)=w_{ji}(k)-\eta_{ji}(k)sign\frac{\partial E_j(k)}{\partial w_{ji}} \tag{4.182}$$

з параметром кроку

$$\eta_{ji}(k)=\begin{cases}\min\left\{a\eta_{ji}(k-1),\eta_{\max}\right\}, & \text{якщо } \delta_{ji}(k)\delta_{ji}(k-1)>0,\\ \max\left\{b\eta_{ji}(k-1),\eta_{\min}\right\}, & \text{якщо } \delta_{ji}(k)\delta_{ji}(k-1)<0,\\ \eta_{ji}(k) & \text{в інших випадках;}\end{cases} \tag{4.183}$$

і коефіцієнтами $a=1.2,\ b=0.5,\ \eta_{\min}=10^{-6},\ \eta_{\max}=50$.

Фальманом для прискорення процесу навчання було запропоновано модифікувати активаційні функції стандартних нейронів. Так замість звичайної сигмоїди було введено конструкцію [331]

$$\psi\left(u_j\right)=\left(1+e^{-\gamma u_j}\right)^{-1}+0.1u_j, \tag{4.184}$$

а замість гіперболічного тангенса –

$$\psi\left(u_j\right)=\tanh\left(\gamma u_j\right)+0.1u_j. \tag{4.185}$$

Алгоритм навчання, що отримав назву «Quickprop», має форму

$$\begin{cases}w_{ji}\left(k+1\right)=w_{ji}\left(k\right)-\eta\left(k\right)s_{ji}\left(k\right)+\beta_{ji}\left(k\right)\Delta w_{ji}\left(k-1\right),\\ s_{ji}\left(k\right)=\dfrac{\partial E_j\left(k\right)}{\partial w_{ji}}+\gamma w_{ji}\left(k\right),\end{cases} \tag{4.186}$$

де

$$\eta\left(k\right)=\begin{cases}\eta_0, \text{ якщо } \Delta w_{ji}\left(k-1\right)=0 \text{ або } s_{ji}\left(k\right)\Delta w_{ji}\left(k-1\right)>0,\\ 0 \text{ в інших випадках;}\end{cases} \tag{4.187}$$

$$\beta_{ji}\left(k\right)=\begin{cases}\beta_{\max}, \text{ якщо } \beta_{ji}\left(k\right)>\beta_{\max} \text{ або } s_{ji}\left(k\right)\Delta w_{ji}\left(k-1\right)\tilde{\beta}_{ji}\left(k\right)<0,\\ \tilde{\beta}_{ji}\left(k\right)=\dfrac{s_{ji}\left(k\right)}{s_{ji}\left(k-1\right)-s_{ji}\left(k\right)} \text{ в інших випадках.}\end{cases} \tag{4.188}$$

Значення вільних параметрів при цьому приймаються $0.01\le\eta_0\le0.6$, $\beta_{\max}=1.75$.

Досить часто «Quickprop»-алгоритм використовується у спрощеній формі [258]:

$$\Delta w_{ji}\left(k\right)=\begin{cases}\beta_{ji}\left(k\right)\Delta w_{ji}\left(k-1\right), \text{ якщо } \Delta w_{ji}\left(k-1\right)\neq0,\\ \eta_0\dfrac{\partial E_j\left(k\right)}{\partial w_{ji}}, \text{ якщо } \Delta w_{ji}\left(k-1\right)=0,\end{cases} \tag{4.189}$$

де

$$\beta_{ji}\left(k\right)=\min\left\{\frac{\dfrac{\partial E_j\left(k\right)}{\partial w_{ji}}}{\dfrac{\partial E_j\left(k-1\right)}{\partial w_{ji}}-\dfrac{\partial E_j\left(k\right)}{\partial w_{ji}}},\beta_{\max}\right\}. \tag{4.190}$$

Ще один метод прискорення процесу навчання, що отримав назву алгоритму динамічної адаптації, розглянуто у [330]. Ідея цього методу полягає в тому, що в напрямку антиградієнта $-\nabla_{w_j} E_j(k)$ обчислюються два набори синаптичних ваг замість одного

$$\begin{cases} {}_1w_j(k+1) = w_j(k) - \eta(k-1)\nabla_{w_j} E_j(k)\beta, \\ {}_2w_j(k+1) = w_j(k) - \eta(k-1)\nabla_{w_j} E_j(k)/\beta, \\ \beta > 1, \end{cases} \tag{4.191}$$

розраховується параметр кроку

$$\eta(k) = \begin{cases} \eta(k-1)\beta, \text{ якщо } E_j\left({}_1w_j(k+1)\right) \le E_j\left({}_2w_j(k+1)\right) \\ \eta(k-1)/\beta \text{ в інших випадках.} \end{cases} \tag{4.192}$$

і, нарешті, перераховуються синаптичні ваги

$$w_j(k+1) = \begin{cases} {}_1w_j(k+1), \text{ якщо } E_j\left({}_1w_j(k+1)\right) \le E_j\left({}_2w_j(k+1)\right), \\ {}_2w_j(k+1), \text{ в інших випадках.} \end{cases} \tag{4.193}$$

Перелік можливих підходів до прискорення процесу навчання можна було б продовжувати, проте оскільки всі розглянуті алгоритми будуються на тих чи інших евристиках, досить важко сформулювати загальні рекомендації з їхнього використання. У кожній конкретній задачі найкращим може виявитися кожен з описаних тут підходів.

Усі розглянуті вище алгоритми навчання з погляду теорії оптимізації [322] належать до градієнтних процедур, чи процедур оптимізації першого порядку тобто таких, під час побудови яких використовуються тільки перші похідні цільових функцій.

Домогтися істотного підвищення якості процесів навчання можна, переходячи до так званих ньютонівських алгоритмів, або процедур другого порядку, із синтезом яких крім перших використовуються похідні другого порядку [235, 238].

Для глобальної цільової функції

$$E_j^k = \frac{1}{2}\sum_{p=0}^{k} e_j^2(p) = \frac{1}{2}\sum_{k} e_j^2(k), \tag{4.194}$$

заданої на всій навчальній вибірці, алгоритм настроювання синаптичних ваг може бути записаний у вигляді

$$w_j(k+1) = w_j(k) - \left(\nabla^2_{w_j} E_j^k\right)^{-1}\left(\nabla_{w_j} E_j^k\right), \tag{4.195}$$

де

$$\nabla^2_{w_j} E_j^k = \left\{ \frac{\partial^2 E_j^k}{\partial w_{ji} \partial w_{jq}} \right\} = \begin{pmatrix} \frac{\partial^2 E_j^k}{\partial w_{j0}^2} & \frac{\partial^2 E_j^k}{\partial w_{j0} \partial w_{j1}} & \cdots & \frac{\partial^2 E_j^k}{\partial w_{j0} \partial w_{jn}} \\ \frac{\partial^2 E_j^k}{\partial w_{j1} \partial w_{j0}} & \frac{\partial^2 E_j^k}{\partial w_{j1}^2} & \cdots & \frac{\partial^2 E_j^k}{\partial w_{j1} \partial w_{jn}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 E_j^k}{\partial w_{jn} \partial w_{j0}} & \frac{\partial^2 E_j^k}{\partial w_{jn} \partial w_{j1}} & \cdots & \frac{\partial^2 E_j^k}{\partial w_{jn}^2} \end{pmatrix} \tag{4.196}$$

– $(n+1)\times(n+1)$ – матриця-гессіан, утворена другими похідними;

$\nabla_w E_j^k = \left(\frac{\partial E_j^k}{\partial w_{j0}}, \frac{\partial E_j^k}{\partial w_{j1}}, \ldots, \frac{\partial E_j^k}{\partial w_{jn}} \right)^T$ – $(n+1)\times 1$ – вектор-градієнт.

Хоча з теоретичної точки зору ньютонівські методи істотно перевершують градієнтні, на практиці виникають проблеми як з обчисленням самого гессіану, так і з його обертанням, особливо під час роботи в реальному часі.

Тому в задачах навчання штучних нейронних мереж широкого поширення отримали псевдоньютонівські методи, які використовують ті чи інші спрощені представлення гессіану.

У найпростішому з таких алгоритмів [330]

$$w_{ji}(k+1) = w_{ji}(k) - \left(\frac{\partial^2 E_j^k}{\partial w_{ji}^2} \right)^{-1} \cdot \frac{\partial E_j^k}{\partial w_{ji}} \tag{4.197}$$

зневажають позадіагональними елементами гессіану. На жаль, незважаючи на простоту чисельної реалізації, цей алгоритм може приводити до значних коливань у процесі настроювання ваг.

Більш точні й ефективні результати можуть бути отримані на основі методів нелінійної ідентифікації систем [332–337].

Перепишемо цільову функцію (4.194) у вигляді

$$\begin{aligned} E_j^k &= \frac{1}{2} \sum_{p=0}^{k} e_j^2(p) = \frac{1}{2} \sum_k e_j^2(k) = \\ &= \frac{1}{2} \sum_k \left(d_j(k) - y_j(k) \right)^2 = \frac{1}{2} \sum_k \left(d_j(k) - \psi \left(\sum_{i=0}^{n} w_{ji} x_i(k) \right) \right)^2 = \\ &= \frac{1}{2} \sum_k \left(d_j(k) - \psi \left(w_j^T x(k) \right) \right)^2 = \frac{1}{2} \left\| D_j(k) - \psi \left(X(k) w_j \right) \right\|^2 \end{aligned} \tag{4.198}$$

(тут $D_j(k) = \left(d_j(0), d_j(1), \ldots, d_j(k) \right)^T - (k+1)\times 1$ – вектор навчальних сигналів; $X(k) = \left(x(0), x(1), \ldots x(k) \right)^T - (k+1)\times(n+1)$ – матриця вхідних сигналів на часовому інтервалі від 0 до k) і введемо до розгляду $(n+1)\times(k+1)$ – матрицю J_j^k,

з елементами $J_{pi}^{k} = \partial\psi\left(w_j^T x(p)\right)\big/\partial w_{ji}$, і $(n+1)\times(n+1)$ – матрицю H_j^k, елементи якої мають вигляд

$$H_{iq}^{k} = \frac{\partial^2 E_j^k}{\partial w_{ji}\partial w_{jq}} = \sum_k \frac{\partial\psi\left(w_j^T x(k)\right)}{\partial w_{ji}}\frac{\partial\psi\left(w_j^T x(k)\right)}{\partial w_{jq}} - \\ -\sum_k \left(d_j(k) - \psi\left(w_j^T x(k)\right)\right)\frac{\partial^2\psi\left(w_j^T x(k)\right)}{\partial w_{ji}\partial w_{jq}} = H_{iq}^{k,1} - H_{iq}^{k,2}. \tag{4.199}$$

За деяких дуже необтяжливих припущень щодо характеру активаційних функцій $\psi(\bullet)$ [238], членами $H_{iq}^{k,2}$ можна зневажити, тобто покласти

$$H_j^k \approx J_j^k J_j^{kT}. \tag{4.200}$$

Тоді ітераційний релаксаційний процес мінімізації критерію (4.198) може бути записаний у формі процедури Гаусса–Ньютона [332]

$$w_j^{k^*+1}(k) = w_j^{k^*}(k) + \left(J_j^k J_j^{kT}\right)^{-1} J_j^k\left(D_j(k) - \psi\left(X(k)w_j^{k^*}(k)\right)\right), \tag{4.201}$$

де верхній індекс k^* позначає номер ітерації прискореного машинного часу, в якому обробляється повний набір сигналів $D_j(k), X(k)$.

На практиці найбільшого поширення набули алгоритми Гартлі [338]

$$w_j^{k^*+1}(k) = w_j^{k^*}(k) + \eta^{k^*}(J_j^k J_j^{kT})^{-1} J_j^k(D_j(k) - \psi(X(k)w_j^{k^*}(k))), \tag{4.202}$$

де η^{k^*} – деякий позитивний демпфуючий параметр, і Марквардта [339]

$$w_j^{k^*+1}(k) = w_j^{k^*}(k) + \left(J_j^k J_j^{kT} + \beta^{k^*} L^{k^*}\right)^{-1} J_j^k\left(D_j(k) - \psi\left(X(k)w_j^{k^*}(k)\right)\right), \tag{4.203}$$

де β^{k^*} – скалярний параметр; L^{k^*} – ненегативно визначена регуляризуюча матриця, що забезпечує стійкість процесу навчання.

Поєднуючи (4.202) і (4.203), можна записати такий узагальнений алгоритм обчислення синаптичних ваг:

$$w_j^{k^*+1}(k) = w_j^{k^*}(k) + \eta^{k^*}\left(J_j^k J_j^{kT} + \beta^{k^*} L^{k^*}\right)^{-1} J_j^k\left(D_j(k) - \psi\left(X(k)w_j^{k^*}(k)\right)\right). \tag{4.204}$$

Процес уточнення оцінок у машинному часі припиняється або з надходженням нового спостереження $d_j(k+1),\ x(k+1)$, або з виконанням умови

$$\left\| w_j^{k^*+1}(k) - w_j^{k^*}(k)\right\| \le \varepsilon. \tag{4.205}$$

Алгоритми (4.201) – (4.204) належать до так званих пакетних алгоритмів навчання, коли вся наявна вибірка спостережень обробляється одночасно за «епохами», а уточнення синаптичних ваг відбувається у прискореному часі. Природно, що використання подібних процедур у реальному часі досить проблематичне, оскільки між двома тактами реального часу має встигнути відбутися кілька епох навчання.

Використовувати псевдоньютонівські алгоритми навчання в реальному часі можна, обмежуючи обсяг оброблюваної вибірки, тобто переходячи до однокрокових і багатокрокових алгоритмів з кінцевою пам’яттю.

Розглядаючи однокроковий варіант алгоритму (4.204) у вигляді

$$w_j(k+1)=w_j(k)+\eta\left(J_j(k)J_j^T(k)+\beta\mathrm{I}\right)^{-1}J_j(k)\left(d_j(k)-\psi\left(w_j^T(k)x(k)\right)\right), \quad (4.206)$$

з урахуванням співвідношень для псевдообернених матриць [284]

$$\lim_{\beta\to 0} J\left(_j(k)J_j^T(k)+\beta\mathrm{I}\right)^{-1}=\left(J_j(k)J_j^T(k)\right)^{+}, \quad (4.207)$$

$$\left(J_j(k)J_j^T(k)\right)^{+}J_j(k)=\left(J_j^T(k)\right)^{+}=J_j(k)\left\|J_j(k)\right\|^{-2}, \quad (4.208)$$

(тут $J_j(k)=\nabla_{w_j}\psi\left(w_j^T x(k)\right)$), приходимо до процедури [178]

$$w_j(k+1)=w_j(k)+\eta\frac{d_j(k)-\left(w_j^T(k)x(k)\right)}{\left\|J_j(k)\right\|^2}J_j(k), \quad (4.209)$$

що є узагальненням алгоритму Качмажа (4.15) у нелінійному випадку.

На основі процедури (4.209) нескладно отримати оптимальні модифікації дельта-правила навчання [340, 341]. Так, наприклад, оптимальна форма алгоритму (4.135) має вигляд

$$w_j(k+1)=w_j(k)+\frac{e_j(k)x(k)}{\gamma_j y_j(k)\left(1-y_j(k)\right)\left\|x(k)\right\|^2}, \quad (4.210)$$

а алгоритму (4.136) відповідає процедура

$$w_j(k+1)=w_j(k)+\frac{e_j(k)x(k)}{\gamma_j\left(1-y_j^2(k)\right)\left\|x(k)\right\|^2}. \quad (4.211)$$

Аналогічно попередньому для довільної регуляризуючої добавки можна записати однокроковий алгоритм навчання

$$w_j(k+1)=w_j(k)+\eta\left(J_jJ_j^T+\beta L\right)^{-1}J_j(k)\left(d_j(k)-\psi\left(w_j^T(k)x(k)\right)\right), \quad (4.212)$$

який за допомогою формули Шермана–Моррисона перетворюється до простого вигляду [342]

$$w_j(k+1)=w_j(k)+\eta L^{-1}\frac{d_j(k)-\psi\left(w_j^T(k)x(k)\right)}{\beta+J_j^T(k)L^{-1}J_j(k)}J_j(k), \quad (4.213)$$

який є узагальненням (4.71).

Для навчання штучних нейронних мереж значного поширення отримав псевдон’ютонівський алгоритм Левенберга–Марквардта [238, 242], що має вигляд

$$w_j(k+1)=w_j(k)+\left(J_j(k)J_j^T(k)+\beta\mathrm{I}\right)^{-1}J_j(k)\left(d_j(k)-\psi\left(w_j^T(k)x(k)\right)\right). \quad (4.214)$$

Нескладно побачити, що він є окремим випадком процедури (4.213) і шляхом нескладних перетворень може бути приведений до форми [343]

$$w_j(k+1)=w_j(k)+\frac{d_j(k)-\psi\left(w_j^T(k)x(k)\right)}{\beta+\left\|J_j(k)\right\|^2}J_j(k), \quad (4.215)$$

що є нелінійним узагальненням адитивного алгоритму Качмажа. У такий спосіб з використанням алгоритму Левенберга–Марквардта цілком відпадає необхідність в операції обертання матриць.

Вводячи експонеційне зважування застарілої інформації, на основі (4.209) можна побудувати модифікований алгоритм типу стохастичної апроксимації

$$\begin{cases} w_j(k+1) = w_j(k) + \eta r^{-1}(k)\left(d_j(k) - \psi\left(w_j^T(k)x(k)\right)\right)J_j(k), \\ r(k) = \alpha r(k-1) + \left\|J_j(k)\right\|^2, \quad 0 \le \alpha \le 1, \end{cases} \tag{4.216}$$

окремим випадком якого є (4.53).

Беручи за основу (4.213), отримуємо алгоритм

$$\begin{cases} w_j(k+1) = w_j(k) + \eta r^{-1}L^{-1}(k)\left(d_j(k) - \psi\left(w_j^T(k)x(k)\right)\right)J_j(k), \\ r(k) = \alpha r(k-1) + \beta + J_j^T(k)L^{-1}J_j(k), \quad 0 \le \alpha \le 1, \end{cases} \tag{4.217}$$

що також має згладжуючі та слідкуючі властивості.

Поряд з експоненційним зважуванням інформації за аналогією з (4.34) – (4.41) і (4.72) можна розглянути алгоритми з кінцевою пам'яттю типу «ковзне вікно». Введемо такі позначення:

$1 \le \chi \le k$ – пам'ять алгоритму;

$J_j^{\chi}(k) - (n+1)\times\chi$ – матриця, утворена градієнтами $J_j(k-\chi+1)$, $J_j(k-\chi+2),\ldots,J_j(k)$;

$D_j^{\chi}(k) - \chi\times 1$ – вектор навчальних сигналів на інтервалі $k-\chi+1, k-\chi+2,\ldots,k$,

$\psi\left(X^{\chi}(k)w_j^{k^*}(k)\right)$ – вектор виходів нейрона на інтервалі $k-\chi+1,\ldots,k$, отриманий внаслідок k^*-ї машинної ітерації;

$$P^{k^*}(k,\chi) = \left(J_j^{\chi}(k)J_j^{\chi T}(k) + \beta^{k^*}L^{k^*}\right)^{-1}, \quad P^0(0,\chi) = \left(\beta^0 L^0\right)^{-1}; \tag{4.218}$$

$$\mathrm{M}^{k^*}(k,\chi) = J_j^{\chi}(k)\left(D_j^{\chi}(k) - \psi\left(X^{\chi}(k)w_j^{k^*}(k)\right)\right) \tag{4.219}$$

– $(n+1)\times 1$ – вектор.

Ще раз зазначимо, що машинні ітерації відбуваються з частотою, яка перевищує частоту надходження даних. Тоді багатокроковий нелінійний алгоритм може бути записаний у вигляді [272]

$$w_j^{k^*+1}(k) = w_j^{k^*}(k) + \eta^{k^*}P^{k^*}(k,\chi)\mathrm{M}^{k^*}(k,\chi), \tag{4.220}$$

– введення k-го спостереження:

$$\tilde{P}^{k^*-1}(k,\chi) = P^{k^*-1}(k-1,\chi) - \frac{P^{k^*-1}(k-1,\chi)J_j(k)J_j^T(k)P^{k^*-1}(k-1,\chi)}{1+J_j^T(k)P^{k^*-1}(k-1,\chi)J_j(k)}, \tag{4.221}$$

– скидання $(k-\chi)$-го спостереження:

$$P^{k^*-1}(k,\chi) = \tilde{P}^{k^*-1}(k,\chi) + \frac{\tilde{P}^{k^*-1}(k,\chi)J_j(k-\chi)J_j^T(k-\chi)\tilde{P}^{k^*-1}(k,\chi)}{1-J_j^T(k-\chi)\tilde{P}^{k^*-1}(k,\chi)J_j(k-\chi)}, \tag{4.222}$$

– введення нового регуляризатора $\beta^{k^*} L^{k^*}$ на k^*-й машинній ітерації:

$$\tilde{P}^{k^*}(k,\chi) = P^{k^*-1}(k,\chi) - \beta^{k^*}\sqrt{L^{k^*}}\left(\mathrm{I} + \beta^{k^*}\sqrt{L^{k^*}}P^{k^*-1}(k,\chi)\sqrt{L^{k^*}}\right)^{-1} \times$$
$$\times\sqrt{L^{k^*}}P^{k^*-1}(k,\chi), \tag{4.223}$$

– скидання старого регуляризатора:

$$P^{k^*}(k,\chi) = \tilde{P}^{k^*}(k,\chi) + \beta^{k^*-1}\sqrt{L^{k^*-1}}\left(\mathrm{I} + \beta^{k^*-1}\sqrt{L^{k^*-1}}\tilde{P}^{k^*}(k,\chi)\sqrt{L^{k^*-1}}\right)^{-1} \times$$
$$\times\sqrt{L^{k^*-1}}\tilde{P}^{k^*}(k,\chi), \tag{4.224}$$

$$\mathrm{M}^{k^*}(k,\chi) = \mathrm{M}^{k^*-1}(k,\chi) + J_j(k)\left(d_j(k) - \psi\left(w_j^{k^*T}(k)x(k)\right)\right) -$$
$$-J_j(k-\chi)\left(d_j(k-\chi) - \psi\left(w_j^{k^*T}(k)x(k-\chi)\right)\right). \tag{4.225}$$

Розглянуті алгоритми (4.206)–(4.225) є адаптивними модифікаціями ньютонівських процедур навчання і дозволяють забезпечити настроювання нейронних мереж у стохастичних і нестаціонарних умовах.

Говорячи про настроювання нейронних мереж, у загальному випадку варто мати на увазі не тільки синаптичні ваги, але й інші вільні параметри. Так у [240] за допомогою алгоритмів навчання пропонується крім синаптичних ваг настроювати центри і коваріаційні матриці рецепторних полів радіально-базисних мереж. Так, вибравши як радіально-базисну функцію форму

$$\phi_i(x) = \Phi\left(\left\|x - c_i\right\|^2_{\Sigma_i^{-1}}\right), \tag{4.226}$$

вводячи відображення, здійснюване мережею у вигляді

$$y_j(k) = \sum_{i=0}^{h} w_{ji}\Phi\left(\left\|x - c_i\right\|^2_{\Sigma_i^{-1}}\right), \tag{4.227}$$

і локальну цільову функцію

$$E_j(k) = \frac{1}{2}e_j^2(k) = \frac{1}{2}\left(d_j(k) - \sum_{i=0}^{h} w_{ji}\Phi\left(\left\|x - c_i\right\|^2_{\Sigma_i^{-1}}\right)\right), \tag{4.228}$$

нескладно розрахувати похідні за вільними параметрами

$$\begin{cases} \dfrac{\partial E_j(k)}{\partial w_{ji}} = -e_j(k)\Phi\left(\left\|x - c_i\right\|^2_{\Sigma_i^{-1}}\right), \\ \nabla_{c_i} E_j(k) = 2e_j(k)w_{ji}\Phi'\left(\left\|x - c_i\right\|^2_{\Sigma_i^{-1}}\right)\Sigma_i^{-1}\left(x(k) - c_i\right), \\ \left\{\dfrac{\partial E_j(k)}{\partial \Sigma_i^{-1}}\right\} = -e_j(k)w_{ji}\Phi'\left(\left\|x - c_i\right\|^2_{\Sigma_i^{-1}}\right)\left(x(k) - c_i\right)\left(x(k) - c_i\right)^T. \end{cases} \tag{4.229}$$

Алгоритм навчання радіально-базисної нейронної мережі, що містить лінійну і нелінійну процедури, у загальному випадку має вигляд

$$\begin{cases} w_{ji}(k+1) = w_{ji}(k) + \eta_w(k) e_j(k) \Phi\left(\left\| x(k) - c_i(k) \right\|^2_{\Sigma_i^{-1}(k)} \right), \\ c_i(k+1) = \\ = c_i(k) - \eta_c(k) e_j(k) w_{ji}(k) \Phi'\left(\left\| x(k) - c_i(k) \right\|^2_{\Sigma_i^{-1}(k)} \right) \Sigma_i^{-1}(k) \left(x(k) - c_i(k) \right), \\ \Sigma_i^{-1}(k+1) = \Sigma_i^{-1}(k) + \eta_\Sigma(k) e_j(k) w_{ji}(k) \Phi'\left(\left\| x(k) - c_i(k) \right\|^2_{\Sigma_i^{-1}(k)} \right) \times \\ \times \left(x(k) - c_i(k) \right) \left(x(k) - c_i(k) \right)^T, \end{cases} \quad (4.230)$$

де $\eta_w(k), \eta_c(k), \eta_\Sigma(k)$ – параметри кроку за відповідними настроюваними змінними.

4.5 Еволюційні алгоритми навчання

Розглянуті вище алгоритми навчання, засновані на градієнтних і ньютонівських процедурах оптимізації, реалізують так званий регулярний підхід, у рамках якого обчислення на кожному кроці синаптичних ваг здійснюється на основі досить чітко формалізованих правил. Проте, як відзначалося в [248, 344], регулярні процедури є ефективними, але підходять не для будь-якої цільової функції і не для будь-якої архітектури мережі. У ситуаціях, коли цільова функція або вихідний сигнал мережі є недиференційованими, аба багатоекстремальними, чи просто обчислення похідних з певних причин небажане, на перший план виходять еволюційні алгоритми навчання, що розвиваються в трьох незалежних напрямках: випадковий пошук [345–349], еволюційне планування [350–353] і генетичне програмування [354–357].

У загальному випадку навчання на основі еволюційного підходу ґрунтується на елементарному методі спроб і помилок, коли рішення шукається випадково і при удачі приймається, а при невдачі відкидається для того, щоб негайно знову звернутися до випадкового вибору як джерела можливостей. Така випадкова основа пошуку рішень спирається на впевненість, що саме випадковість містить у собі всі можливості, у тому числі й найкраще рішення [269]. У загальному випадку метод спроб і помилок є універсальним підходом (хоча і не завжди досить «швидким») до вирішення задач в умовах дефіциту апріорної і поточної інформації, до яких повною мірою можуть бути віднесені задачі навчання нейронних мереж.

Одним із проявів ефективної реалізації спроб і помилок у природі є закони еволюції, які є ітеративним процесом, що включає відтворення в умовах мутацій, природний відбір, випадкові рекомбінації, індивідуальне навчання [358]. Найбільш яскравим елементим випадковості у спробах і

помилках є мутації, що мають стохастичний характер. Ці мутації можуть або поліпшити, або погіршити пристосувальні властивості біологічної особи. У першому випадку ця особа має більше шансів на виживання і, отже, на закріплення отриманої мутації в потомстві.

Використання ідей біологічної еволюції в техніці і породило те, що сьогодні називається еволюційними алгоритмами, що поряд з нейронними мережами і фаззі-системами сформували новий науковий напрямок – обчислювальний інтелект.

Алгоритми випадкового пошуку. Загальна форма алгоритмів випадкового пошуку в задачах навчання нейронних мереж має вигляд

$$w_j(k+1) = w_j(k) + \Delta w_j(k), \tag{4.231}$$

де $\Delta w_j(k)$ – випадкова коригувальна добавка, що визначає напрямок зсуву вектора синаптичних ваг на кожній ітерації їхнього уточнення.

Найпростішою з таких процедур є метод випадкової оптимізації (ROM) [359], суть якого полягає в такому – до поточного вектора синаптичних ваг $w_j(k)$ додається випадковий вектор $\zeta(k)$. Далі обчислюється значення цільової функції $E_j\left(w_j(k)+\zeta((k))\right)$ і у випадку

$$E_j\left(w_j(k)+\zeta(k)\right) < E_j\left(w_j(k)\right) \tag{4.232}$$

вважається, що

$$w_j(k+1) = w_j(k) + \zeta(k). \tag{4.233}$$

У протилежному ж випадку $w_j(k+1) = w_j(k)$, тобто уточнення не відбувається, а генерується нова випадкова проба $\zeta(k)$.

До цього алгоритму близький гомеостатичний пошук [346] у формі

$$w_j(k+1) = \begin{cases} w_j(k) + \eta\zeta(k), \\ \text{якщо } E_j(k+1) = E_j\left(w_j(k)+\zeta(k)\right) < E_j\left(w_j(k)\right) = E_j(k), \\ w_j(k) \quad \text{в іншому випадку,} \end{cases} \tag{4.234}$$

де $\zeta(k) = \left(\zeta_0(k), \zeta_1(k), \ldots, \zeta_n(k)\right)^T$ – випадковий рівномірно розподілений вектор такий, що $-1 \le \zeta_i(k) \le 1$; η – крок пошуку.

Незважаючи на крайню простоту цих двох алгоритмів, їхнє використання для вирішення реальних задач не виправдане, оскільки пов’язане з великими часовими витратами. Як відзначав Л.А. Растригін, причини криються в тому, що подібні алгоритми вирішують задачі, гарантуючи при цьому лише те, що час відшукання цього рішення є кінцевим, але може бути занадто довгим.

Прискорити процес навчання на основі випадкового пошуку можна, скориставшись додатковою інформацією про характер цільової функції $E_j(k)$ або E_j^k, якщо звичайно такі данні є.

Так, наприклад, інформація про гладкість гіперповерхні $E_j(k)$ дозволить підвищити швидкодію за допомогою використання наближеного співвідношення

$$\nabla_{w_j} E_j(k) \approx \nabla_{w_j} E_j(k+1), \tag{4.235}$$

справедливого при малих кроках η. Це співвідношення приводить до інтуїтивного висновку, що, зробивши один вдалий крок, слід продовжувати рухатися в цьому ж напрямку з високою імовірністю успіху. На цьому висновку будуються алгоритми випадкового пошуку з лінійною тактикою [269], найпростішим з яких є алгоритм випадкового спуску

$$w_j(k+1) = w_j(k) + \Delta w_j(k), \tag{4.236}$$

$$w_j(k) = \begin{cases} \eta\zeta(k), & \textit{якщо } R^-(k), \\ \Delta w_j(k-1), & \textit{якщо } R^+(k), \end{cases} \tag{4.237}$$

де символами $R^-(k)$ і $R^+(k)$ позначені так звані реакції процесу навчання. Негативна реакція $R^-(k)$ пов'язана зі збільшенням критерію якості

$$\Delta E_j(k) = E_j\left(w_j(k+1)\right) - E_j\left(w_j(k)\right) \geq 0 \to R^-(k), \tag{4.238}$$

а позитивна – з його зменшенням

$$\Delta E_j(k) < 0 \to R^+(k). \tag{4.239}$$

Зміст алгоритму (4.236), (4.237) полягає в тому, що зі стану мережі $w_j(k)$ робляться випадкові кроки в просторі синаптичних ваг, поки не буде знайдений напрямок $\zeta(k)$, який веде до зменшення цільової функції. Позитивна реакція алгоритму полягає в повторенні такого кроку доти, поки критерій якості не почне збільшуватися, що викликає негативну реакцію – випадкові проби нових напрямків і т.д.

Цей алгоритм побудований на принципі «покарання» випадковістю, відповідно до якого оператор випадкового кроку $\zeta(k)$ вводиться як негативна реакція на невдалий крок навчання. У випадку ж удачі пошук діє тим же чином, що привів до позитивної реакції. Така форма поведінки доцільна для цільових функцій близьких до лінійних, властивості яких з переходом з одного стану в інший змінюються незначно. Тому цей алгоритм іноді називається автоматом з лінійною тактикою [347].

Процес навчання за допомогою алгоритмів з лінійною тактикою чітко поділяється на два етапи, які відповідають двом різним типам поведінки. Перший етап зводиться до визначення напрямку спуска s_j, другий – це власне настроювання ваг в обраному напрямку доти, поки цільова функція не почне збільшуватися.

Розглянемо деякі з можливих способів визначення напрямку спуска.

Найбільш природною є суто випадкова оцінка напрямку спуска. Зміст її зводиться до спроби вводити випадково обранний напрямок:

$$s_j = \zeta, \tag{4.240}$$

де $\zeta = (\zeta_0, \zeta_1, \ldots, \zeta_n)^T$ – одиничний випадковий вектор, рівномірно розподілений в усіх напрямках у просторі синаптичних ваг.

Далі оцінка напрямку спуска за найкращою з декількох випадкових проб. З вихідної точки $w_j(k)$ на відстань пробного кроку η_ζ робиться кілька випадкових проб показника якості $E_j\left(w_j(k)+\eta_\zeta\zeta^q\right)$ у випадкових напрямках ζ^q, $q=1,2,\ldots,Q$. За напрямок спуска s_j обирається той, що забезпечує найменше значення показника якості

$$s_j=\zeta^*, \tag{4.241}$$

де ζ^* задовольняє очевидній умові

$$E_j\left(w_j(k)+\eta_\zeta\zeta^*\right)=\min_{q=1,2,\ldots,Q}\left\{E_j\left(w_j(k)+\eta_\zeta\zeta^q\right)\right\}. \tag{4.242}$$

Нескладно бачити, що при $Q=1$ ця оцінка вироджується в попередню.

Використовується також оцінка напрямку спуска методом статистичного градієнта. У цьому випадку за напрямок руху приймається середньозважене з Q випадкових напрямків, кожен з яких береться з вагою, що відповідає збільшенню критерію якості уздовж цього напрямку

$$s_j=-dir\sum_{q=1}^{Q}\zeta^q\left(E_j\left(w_j(k)+\eta_\zeta\zeta^q\right)-E_j\left(w_j(k)\right)\right), \tag{4.243}$$

де dir – це одиничний вектор, що визначає напрямок, який нас цікавить

$$dir\,x=\frac{x}{\|x\|}. \tag{4.244}$$

Нарешті, слід зазначити ортогоналізований метод статистичного градієнта. Цей метод визначення напрямку спуска s_j відрізняється від попереднього тим, що випадкові напрямки ζ^q, $q=1,2,\ldots,Q$, $Q\le n+1$ ортогональні, тобто

$$\left(\zeta^q\right)^T\zeta^p=\begin{cases}1, & \text{якщо } q=p,\\ 0, & \text{якщо } q\ne p.\end{cases} \tag{4.245}$$

Можна побачити, що при $Q=n+1$, тобто з кількістю ортогональних спроб, що дорівнюють числу ваг, що настроюються, цей метод вироджується в алгоритм оптимізації Гаусса-Зайделя [269, 360].

Далі можна перейти до другого етапу навчання – власне спуску. Сам по собі процес спуску полягає у визначенні мінімуму цільової функції уздовж обраного напрямку s_j. Найчастіше це крокова процедура, причому після кожного кроку приймається рішення: чи рухатися далі, чи припинити спуск і звернутися до першого етапу.

Як найбільш відомі алгоритми спуску можна виділити, наприклад, спуск з парними пробами, коли уздовж напрямку спуску беруться парні проби, що дають можливість прийняти рішення: спускатися далі чи припинити спуск і повернутися до першого етапу. Алгоритм має вигляд

$$w_j(k+1)=w_j(k)+\begin{cases}\eta s_j, & \text{якщо } E_j\left(w_j(k)+\eta_\zeta s_j\right)-E_j\left(w_j(k)-\eta_\zeta s_j\right)<\varepsilon,\\ 0, & \text{якщо } E_j\left(w_j(k)+\eta_\zeta s_j\right)-E_j\left(w_j(k)-\eta_\zeta s_j\right)\ge\varepsilon\end{cases} \tag{4.246}$$

і для нього характерне чітке розділення пробних і робочих кроків. Саме тому його іноді називають алгоритмом з розділеними пробними і робочими кроками [346].

У задачах навчання ШНМ, особливо в реальному часі, найкращими є алгоритми зі сполученими пробними і робочими кроками, такі, як сполучений спуск

$$w_j(k+1) = w_j(k) + \begin{cases} \eta s_j, & \text{якщо } E_j\left(w_j(k)\right) - E_j\left(w_j(k-1)\right) < \varepsilon, \\ 0, & \text{якщо } E_j\left(w_j(k)\right) - E_j\left(w_j(k-1)\right) \geq \varepsilon \end{cases} \tag{4.247}$$

і реверсний пошук, що є розширенням базового ROM [359]

$$w_j(k+1) = w_j(k) + \begin{cases} \eta s_j, & \text{якщо } E_j\left(w_j(k)\right) - E_j\left(w_j(k-1)\right) < \varepsilon, \\ -\eta s_j, & \text{якщо } E_j\left(w_j(k)\right) - E_j\left(w_j(k-1)\right) \geq \varepsilon. \end{cases} \tag{4.248}$$

Алгоритм (4.248) будується на інтуїтивних припущеннях про те, що якщо напрямок s_j веде до зростання, то $-s_j$ – до спадання цільової функції.

Розглянуті алгоритми випадкового пошуку з лінійною тактикою мають істотний недолік [348]: у процесі спуску обраний напрямок все менше і менш відповідає антиградієнтному і тому лінійний спуск досить швидко втрачає сенс. Це змушує звернутися до корекції напрямку спуску в процесі самого спуску. Це може бути здійснено різними способами. Одним з найпростіших таких способів є введення незначного «нишпорення» (зондування) у процесі навчання, при якому оцінюється ефективність нових напрямків для того, щоб прийняти чи не прийняти їх. Нишпорення в процесі спуску при цьому може бути як регулярним, так і суто випадковим. Алгоритм такої корекції може мати, наприклад, такий вигляд:

$$s_j(k+1) = dir\left(s_j(k) + \Delta s_j(k)\right), \tag{4.249}$$

де

$$\Delta s_j(k) = -a\zeta(k)\Delta E_j(k-1); \tag{4.250}$$

a – коефіцієнт нишпорення.

Видно, що при $a = 0$ – це звичайний лінійний спуск. При $a \neq 0$ вектор $s_j(k+1)$ у процесі спуску розвертатиметься в тому напрямку, де збільшення $\Delta E_j(k-1)$ мінімальне.

Вншим, більш радикальним способом поліпшення характеристик спуску є використання так званих локальних алгоритмів випадкового пошуку [269]. Локальність алгоритму пошуку визначається його незалежністю від передісторії. Так алгоритми з лінійною тактикою є нелокальними, причому ця нелокальність викликана характером самого спуску, в якому робочий крок повторюється.

В істотно нелінійній обстановці часто недоцільно повторювати вдалі кроки, оскільки характер цільової функції істотно змінюється на кожному кроці. У цьому випадку краще обрати локальну нелінійну тактику, тобто

починати послідовно незалежні спроби зі зменшенням критерію якості і виправляти помилки, якщо вони виникають.

Якщо при цьому скористатися випадковими кроками-пробами, то отримаємо алгоритми з заохоченням випадковістю, в яких елемент випадковості $\zeta(k)$ вводиться як позитивна реакція $R^{+}(k)$, а негативною реакцією $R^{-}(k)$ є заходи з усунення наслідків невдалого випадкового кроку. Цей алгоритм можна записати у вигляді

$$w_j(k+1) = w_j(k) + \begin{cases} \eta\zeta(k), & \text{якщо } R^{+}(k), \\ f\left(\Delta E_j(k-1)\right), & \text{якщо } R^{-}(k). \end{cases} \qquad (4.251)$$

Як видно, оператор випадкового кроку $\zeta(k)$ вводиться як заохочення на вдалий крок $R^{+}(k)$ $\left(\Delta E_j(k-1) < 0\right)$. Негативну реакцію $R^{-}(k)$ викликає дія, наприклад

$$f\left(\Delta E_j(k-1)\right) = -\Delta w_j(k-1) = w_j(k-1) - w_j(k), \qquad (4.252)$$

спрямована на подолання отриманого негативного ефекту $R^{-}(k)\left(\Delta E_j(k-1) \geq 0\right)$, після чого знову виникає випадковий крок $\zeta(k+1)$. Таким чином, алгоритм (4.251) виправляє помилки, допущені в процесі випадкового пошуку.

Різні алгоритми локального випадкового пошуку відрізняються один від одного способами визначення напрямку, в якому робиться спроба робочого крок пошуку.

Тут варто виділити алгоритм із парними пробами [346], що припускає чіткий поділ між пошуковими (спробними) і робочими кроками. У випадковому напрямку, обумовленому вектором ζ, по обидва боки від вихідного стану $w_j(k)$ робляться проби. Значення цільової функції в точках $w_j(k) \pm \eta_\zeta \zeta(k)$ визначають напрямок робочого кроку, що робиться в бік найменшого значення критерію якості

$$w_j(k+1) = w_j(k) - \eta\zeta(k)\, sign\left(E_j\left(w_j(k) + \eta_\zeta\zeta(k)\right) - E_j\left(w_j(k) - \eta_\zeta\zeta(k)\right)\right). \qquad (4.253)$$

Характерною рисою цього алгоритму є тенденція до «блукання» навіть у тому випадку, якщо оптимум критерію якості знайдений. Дійсно, знайшовши екстремум, алгоритм відразу вводить убік оптимальний набір параметрів, що настроюються, що взагалі ж у нестаціонарних ситуаціях не так погано.

Також можна відзначити алгоритм із поверненням при невдалому кроці, сенс якого полягає в такому. У просторі параметрів, що настроюються, з вихідного стану $w_j(k)$ робиться крок у випадковому напрямку $\zeta(k)$. Якщо значення цільової функції в новому стані дорівнює значенню функції у вихідній точці $E_j(k)$, тобто випадкова проба виявилася невдалою, то алгоритм повертається у початковий стан $w_j(k)$, після чого знову робиться крок у новому випадковому напрямку. Якщо ж цільова функція зменшилася, то крок

вважається робочим і наступний випадковий крок робиться вже з нового стану $w_j(k+1)$. Цей алгоритм можна записати у вигляді

$$w_j(k+1)=w_j(k)+\begin{cases}\eta\zeta(k), & \text{якщо } E_j(k+1)<E_j(k),\\ 0, & \text{якщо } E_j(k+1)\geq E_j(k).\end{cases} \tag{4.254}$$

Далі можна розглянути алгоритм із перерахуванням при невдалому кроці, що є модифікацією попереднього. У цій процедурі поворотний крок при невдалій пробі не відбивається, за рахунок чого алгоритм має підвищену швидкодію. Алгоритм після невдалого кроку знову робить випадкову пробу, відлічену з попереднього стану, тобто повернення начебто перераховується разом з наступним випадковим кроком.

Рекурентна формула для зсуву в просторі параметрів, що настроюються, за цим алгоритмом має вигляд

$$w_j(k+1)=w_j(k)+\begin{cases}\eta\zeta(k), & \text{якщо } E_j(k+1)<E_j^*(k),\\ -\eta\zeta(k-1)+\eta\zeta(k), & \text{якщо } E_j(k+1)\geq E_j^*(k),\end{cases} \tag{4.255}$$

де

$$E_j^*(k)=\min_{p=0,1,\ldots,k}\{E_j(p)\} \tag{4.256}$$

– найменше значення цільової функції за k попередніх кроків пошуку.

Цей алгоритм має підвищену швидкодію, але низький рівень завадостійкості.

Нарешті слід зазначити алгоритм найкращої проби. Цей алгоритм випадкового пошуку спирається на багаторазову випадкову вибірку. З вихідної точки $w_j(k)$ робиться Q випадкових проб $\eta_\zeta\zeta^1(k),\eta_\zeta\zeta^2(k),\ldots,\eta_\zeta\zeta^Q(k)$ у просторі ваг і запам’ятовується той крок, що привів до найбільшого зниження цільової функції. Робочий крок робиться саме в цьому напрямку

$$w_j(k+1)=w_j(k)+\eta\zeta^*(k), \tag{4.257}$$

де $\zeta^*(k)$ – напрямок найкращої проби, що задовольняє співвідношенню

$$E_j\left(w_j(k)+\eta_\zeta\zeta^*(k)\right)=\min_{q=1,2,\ldots Q}\left\{E_j\left(w_j(k)+\eta_\zeta\zeta^q(k)\right)\right\}. \tag{4.258}$$

Очевидно, що зі збільшенням кількості проб обраний напрямок усе більш наближається до найкращого, тобто антиградієнтного, і в асимптоті при $Q\to\infty$ збігається з ним.

Перелік можливих алгоритмів можна було б продовжити, проте набагато більш важливим є їхня порівняльна оцінка за яким-небудь досить універсальним критерієм. Як така природно прийняти конструкцію, іменовану втратами на пошук [269]:

$$K=\frac{\textit{втрати}}{\textit{позитивний ефект}}, \tag{4.259}$$

де втрати визначаються кількістю проб на один цикл пошуку, а ефект виміряються величиною зсуву до точки оптимуму.

На рис 4.13 показано границю зміни втрат на пошук між алгоритмами випадкового пошуку і регулярними градієнтними.

Втрати на випадковий пошук мають параболічний характер, тобто змінюються пропорційно $\sqrt{n}$. Таким чином, випадковий пошук має переваги перед градієнтним за швидкістю при значній кількості параметрів n, що настроюються, причому ця перевага тим більше, чим більше n.

На рис. 4.13 ρ – це відстань до екстремуму цільової функції, що вимірюється в робочих кроках пошуку η. Заштрихована область – де швидкість випадкового пошуку перевищує швидкість градієнтного за критерієм втрат на пошук (4.259).

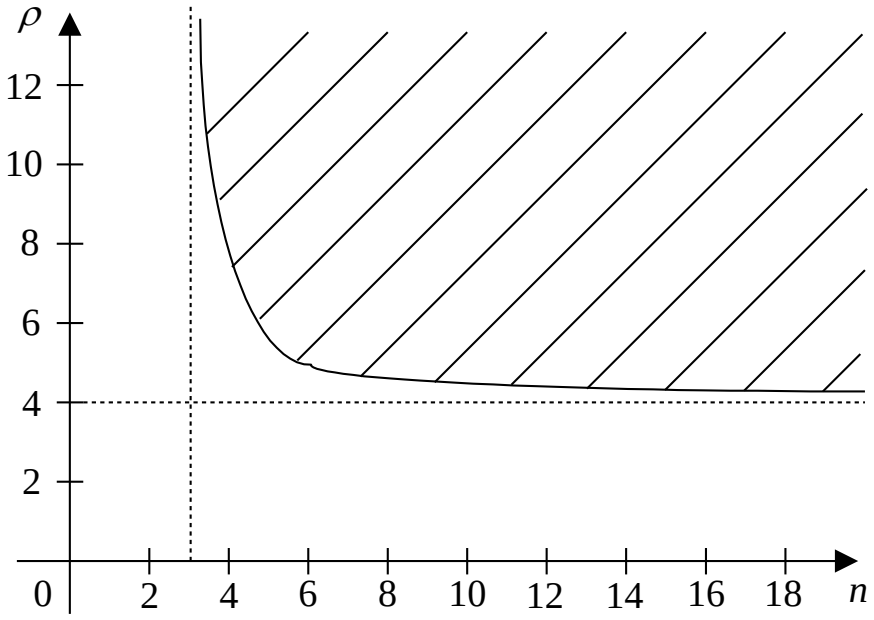


Рис. 4.13. Втрати на пошук

Таким чином, випадковий пошук є більш ефективним в процесі оптимізації об'єктів з великою кількістю параметрів (а ШНМ є саме такими об'єктами) віддалених від екстремуму. У ситуаціях малої розмірності $(n \le 3)$ і в околі екстремуму $(\rho \le 4)$ доцільніше використовувати градієнтні методи.

Властивості випадкового пошуку можна істотно поліпшити, забезпечивши конкретним алгоритмам здатність до самонавчання і самовдосконалення в процесі роботи. Таким чином, виникає самонавчання алгоритмів навчання. Це приводить до того, що в кожному новому стані система навчання або починає пошук спочатку, тобто цілком ігнорує попередній досвід, як, наприклад, за локальних методів пошуку, або «довіряє» попередньому результату і намагається просунутися до оптимуму за напрямком, що у попередньому стані виявився вдалим (пошук з лінійною тактикою).

Оптимальне поводження є проміжним між цими двома крайностями: потрібно запам'ятовувати попередній вдалий напрямок, але довіряти йому не цілком, а лише з визначеною ймовірністю і змінювати цю ймовірність в міру

набуття нового досвіду роботи, тобто між двома сусідніми станами має існувати ймовірнісний взаємозв'язок.

Самонавчання вводиться в алгоритми випадкового пошуку у формі перестроювання його ймовірнісних характеристик, тобто в цілеспрямованому впливі на вектор ζ, який перестає бути рівноймовірним і здобуває деякі переважні напрямки, наприклад, убік попереднього найкращого кроку.

Навчання алгоритму навчання, як правило, починається в обстановці рівноймовірного пошуку. У процесі накопичення досвіду роботи система навчання має здобувати «представлення» про розв'язувану задачу у вигляді деяких імовірних гіпотез про найкращий напрямок руху в просторі параметрів, що настроюються.

Таким чином, процес самонавчання прагне «задетермінувати» пошук у найкращому напрямку, тобто в такий спосіб скоригувати його імовірнісні характеристики, щоб процес настроювання ШНМ став невипадковим. З іншого боку, алгоритм самонавчання повинен мати здатність перенавчатися, якщо ситуація яким-небудь чином змінилася.

І, нарешті, введення самонавчання не повинне виключати застосування описаних вище алгоритмів випадкового пошуку в «чистому вигляді», тобто воно має торкатися тільки ймовірнісних властивостей вибору напрямку робочого кроку, але не змінювати структуру алгоритму. Це дає можливість комбінувати самонавчання з різними алгоритмами, розглянутими вище.

Як підходи до самонавчання випадкового пошуку можна відзначити такі [348].

Найбільш простим є самонавчання методом виключення. Цей підхід спирається на таке очевидне положення. Якщо кількість можливих напрямків руху кінцева, то виключення з розгляду невдалих напрямків, що збільшують критерій якості, підвищує ймовірність відшукання вдалих напрямків, рухаючись уздовж яких, алгоритм навчання зменшує значення цільової функції.

Характерною рисою і недоліком цього підходу є необхідність великої пам'яті для збереження невдалих варіантів випадкових зсувів, що виключаються з подальшого пошуку.

Більш ефективним є алгоритм покоординатного самонавчання, багато в чому схожий з алгоритмом Гаусса–Зайделя. Цей алгоритм використовує ймовірність вибору напрямку руху уздовж i-ї змінної w_{ji} у формі функції від деякої величини, яка називається параметром чи просто «пам'яттю» по i-й координаті на k-му кроці пошуку

$$P_i(k) = P(\alpha_i(k)). \tag{4.260}$$

Як найбільш розповсюджені функції $P(\alpha_i)$ використовуються [269]:

– експоненційна

$$P(\alpha_i) = \begin{cases} \frac{1}{2} e^{-\gamma\alpha_i}, & \text{якщо } \alpha_i \le 0, \\ 1 - \frac{1}{2} e^{-\gamma\alpha_i}, & \text{якщо } \alpha_i > 0, \end{cases} \tag{4.261}$$

– лінійна

$$P(\alpha_i)=\begin{cases}0, & \text{якщо } \alpha_i<-1,\\ \dfrac{1}{2}(1-\alpha_i), & \text{якщо } -1\le\alpha_i<1,\\ 1, & \text{якщо } \alpha_i>1,\end{cases} \tag{4.262}$$

– гауссова

$$P(\alpha_i)=\frac{1}{2}\left(1+\Phi(\alpha_i)\right), \tag{4.263}$$

– синусна

$$P(\alpha_i)=\begin{cases}0, & \text{якщо } \alpha_i<-1,\\ \dfrac{1}{2}+\dfrac{1}{\pi}\arcsin\alpha_i, & \text{якщо } -1\le\alpha_i\le1,\\ 1, & \text{якщо } \alpha_i\ge1\end{cases} \tag{4.264}$$

та інші. Видно, що при $\alpha_i=0$, пошук має рівноймовірний характер, тобто $P(0)=0.5$. Цікаво також те, що функції (4.261) – (4.264) є по суті активаційними сигмоїдальними функціями стандартних нейронів.

Процедура самонавчання реалізується шляхом відповідної зміни параметра пам’яті, наприклад, за допомогою рекурентної формули

$$\alpha_i(k+1)=\alpha_i(k)+\eta_\alpha sign\left(\Delta w_{ji}(k-1)\Delta E_j(k-1)\right). \tag{4.265}$$

Зміст виразу (4.265) полягає в такому: якщо зроблений крок настроювання призвів до збільшення цільової функції, тобто був зроблений у несприятливому напрямку, то ймовірність вибору цього напрямку за наступного кроку зменшується. І навпаки, у випадку зменшення критерію якості ймовірність вибору цього напрямку збільшується.

Як видно, алгоритм самонавчання працює в двох режимах: режимі заохочення при $\Delta E_j<0$ і в режимі покарання при $\Delta E_j\ge0$. Якщо в першому випадку реалізується позитивний зворотний зв’язок, коли зроблений крок призводить до збільшення ймовірності такого самого кроку, то в другому – зворотний зв’язок має негативний характер, коли зроблений невдалий крок призводить до зменшення ймовірності цього кроку за рахунок збільшення ймовірності протилежного кроку. Таким чином, пошук починається як суто випадковий, а згодом здобуває усе більш детермінованих рис.

У розглянутому алгоритмі параметр пам’яті за будь-якою координатою (параметр, що настроюється) внаслідок одного кроку пошуку змінюється на постійну величину, рівну кроку по пам’яті η_α. Проте необхідність навчання конкретної синаптичної ваги залежить насамперед від отриманого результату ΔE_j і ступеня участі цієї ваги у цьому результаті. Тому в ряді випадків доцільніше використовувати так званий пропорційний алгоритм самонавчання у формі

$$\alpha_i(k+1)=\alpha_i(k)-\eta_\alpha\Delta w_{ji}(k-1)\Delta E_j(k-1), \tag{4.266}$$

реагуючий як на результат кроку пошуку, так і на ступінь участі конкретного параметра, що настроюється, у цьому результаті.

Алгоритми (4.265), (4.266) «пам'ятають» усі попередні етапи пошуку, тому в нестаціонарних ситуаціях доцільне введення в процедури самонавчання фактора забування у формі

$$\alpha_i(k+1)=\alpha\alpha_i(k)-\eta_\alpha\Delta w_{ji}(k-1)\Delta E_j(k-1),\ 0\le\alpha\le 1, \qquad (4.267)$$

при цьому $\alpha=1$ забезпечує запам'ятовування системою навчання всього пройденого шляху, а $\alpha=0$ – тільки результату останнього кроку.

Недолік цих алгоритмів, як і всіх процедур покоординатної оптимізації типу Гаусса–Зайделя, пов'язаний з їхньою організацією, що вимагає в кожен момент руху тільки по одній змінній. Відсутність проміжних напрямків знижує швидкість збіжності. Більш ефективним є підхід, заснований на беззупинному самонавчанні.

Нехай $A=(\alpha_0,\alpha_1,\ldots,\alpha_n)^T$ – вектор, що характеризує імовірнісні властивості пошуку за всіма параметрами, що настроюються, причому при $\alpha_i=0\ (i=0,1,\ldots,n)$ пошук передбачається рівноймовірним. Тоді напрямок випадкового кроку в просторі синаптичних ваг зручно представити у вигляді векторної функції

$$\Delta w_j=\eta F(\zeta,A), \qquad (4.268)$$

де F – деяка неперервна за нормою і напрямком векторна функція двох векторних змінних ζ і A.

Функція $F(\zeta,A)$ має задовільняти таким природним вимогам:

$$F(\zeta,0)=\zeta, \qquad (4.269)$$

тобто за нульового значення пам'яті пошук є рівноймовірним;

$$\mathrm{M}\{F(\zeta,A)\}=dirA, \qquad (4.270)$$

тобто математичне сподівання напрямку випадкового кроку повинне збігатися з напрямком вектора A;

дисперсія випадкового кроку $\sigma^2_{\Delta w}$ оберненопропорційна нормі вектора A.

З виконанням цих умов функція $F(\zeta,A)$ забезпечує просторовий розподіл випадкового кроку, що цілеспрямовано змінюється в міру накопичення досвіду навчання.

Як такі функції використовуються [269]

$$F(\zeta,A)=\frac{\zeta+A}{\|\zeta+A\|} \qquad (4.271)$$

та

$$F=\zeta \quad \text{при} \quad \left\|\frac{A}{\|A\|}-\zeta\right\|\le f(\|A\|), \qquad (4.272)$$

де $f(\bullet)$ – монотонно спадна скалярна функція $0\le f(\bullet)\le 2$, причому $f(0)=0$.

Зазначимо, що функція (4.272) описує влучення в $(n+1)$ – вимірний гіперконус з кутом $4\arcsin\dfrac{f(\|A\|)}{2}$ у вершині.

Алгоритм неперервного самонавчання можна записати у вигляді рекурентного співвідношення, що зв'язує два наступних один за одним вектори пам'яті

$$A(k+1)=\alpha A(k)-\eta_{\alpha}\left(\Delta E_j(k-1)+b\right)\Delta w_j(k-1), \tag{4.273}$$

де $b\geq 0$ – коефіцієнт «скептицизму».

Неважко помітити, що з використанням цього алгоритму і з виконанням умов, які накладаються на функцію $F(\bullet)$, вектор A прагне перестроїтися в напрямку, зворотному градієнту цільової функції. Це означає, що кроки пошуку будуть у середньому спрямовані убік найшвидшого зменшення цільової функції.

Ще один підхід до введення самонавчання в алгоритми навчання пов'язаний із введенням у процедуру настроювання ваг ковзного середнього [359]. Це призводить до того, що середнє значення випадкового вектора $\zeta(k)$ стає ненульовим, тобто пошук знов-таки задетермінується в сприятливому напрямку. Алгоритм випадкового пошуку з ковзним середнім може бути записаний у вигляді [349]:

$$\begin{cases} w_j(k+1)=w_j(k)+\eta\bar{\zeta}(k),\ \bar{\zeta}(k+1)=a\zeta(k)+b\bar{\zeta}(k), \\ \text{якщо}\quad E_j\left(w_j(k)+\eta\bar{\zeta}(k)\right)<E_j\left(w_j(k)\right), \\ w_j(k+1)=w_j(k)-\eta\bar{\zeta}(k),\ \bar{\zeta}(k+1)=\bar{\zeta}(k)-c\zeta(k) \\ \text{в іншому випадку,} \\ w_j(k+1)=w_j(k),\ \bar{\zeta}(k+1)=d\bar{\zeta}(k), \\ \text{якщо}\quad E_j\left(w_j(k)-\eta\bar{\zeta}(k)\right)\geq E_j\left(w_j(k)\right). \end{cases} \tag{4.274}$$

Сенс пошуку за допомогою процедури (4.274) полягає в такому. Зі стану $w_j(k)$ робиться випадковий крок $\eta\bar{\zeta}(k)$, і якщо він виявився вдалим, то середнє $\bar{\zeta}(k+1)$ уточнюється за допомогою першого співвідношення алгоритму. У випадку невдалого кроку відбувається реверс у напрямку $-\bar{\zeta}(k)$. Якщо ж і реверсний крок виявився невдалим, синаптичні ваги на цьому такті не уточнюються, а середнє коректується на коефіцієнт d. Автори цієї процедури рекомендують такі значення параметрів: $a=0.4; b=0.2; c=0.4$ и $d=0.5$ і відзначають, що алгоритм «не застрягає» у незначних локальних екстремумах цільової функції.

Келлі й Уілінг [360] запропонували алгоритм так званого повторюваного випадкового пошуку, що має вигляд

$$\begin{cases} w_j(k+1) = w_j(k) + \eta(k)\left(\beta \dfrac{A(k)}{\|A(k)\|} + (1-\beta)\zeta(k) \right), \\ A(k+1) = \alpha A(k) + (1-\alpha)\left(w_j(k+1) - w_j(k) \right), \end{cases} \tag{4.275}$$

де $\eta(k)$ – змінний крок пошуку, що збільшується після успішного кроку і зменшується після невдалого; $A(k)$ – вектор пам'яті, що вказує середній напрямок пошуку на попередніх кроках; α і β – скалярні вагові множники.

На k-му кроці пошуку випадковий вектор $\zeta(k)$ і вектор пам'яті $A(k)$ формують зважену суму, що визначає напрямок руху в просторі параметрів, що настроюються. Значення, що уточнюється $w_j(k+1)$ буде прийнято чи відкинуто залежно від виконання нерівності $E_j\left(w_j(k+1)\right) < E_j\left(w_j(k)\right)$. Після того, як новий вектор $w_j(k+1)$ прийнято чи відкинуто, $\eta(k)$ збільшують або зменшують. Ця процедура в процесі самонавчання також намагається сформувати середній напрямок руху, близький до антиградієнтного.

Загалом, розглянуті алгоритми навчання на основі випадкового пошуку вирішують ту саму задачу, що й алгоритми, в основі яких лежать процедури градієнтної і ньютонівської оптимізації. Разом з тим існує широкий клас задач, де детерміновані методи не працюють взагалі – це задачі глобальної оптимізації, коли цільова функція має множину екстремумів. Задача відшукання глобального екстремуму функції багатьох змінних є проблемою значно більш складною і трудомісткою, ніж визначення локального екстремуму. Справа в тому, що багатоекстремальна функція майже не дає можливості судити про поводження критерію якості за декількома спостереженнями, що можливо з унімодальністю. Природно, що кількість спостережень в процесі пошуку глобального екстремуму має бути дуже великою.

Найпростішою процедурою пошуку глобального екстремуму є так званий блукаючий глобальний пошук [346]. У загальному випадку ця процедура є статистичним розширенням регулярного градієнтного методу (стандартного дельта-правила навчання). З метою надання пошуку глобального характеру на градієнтний рух алгоритму навчання накладається випадкове збурювання $\zeta(k)$, яке створює режим стохастичного блукання.

У неперервному випадку градієнтний метод мінімізації цільової функції $E_j(t)$ зводиться до руху вектора $w_j(t)$ у $(n+1)$-вимірному просторі параметрів, що настроюються, під дією «сили», спрямованої убік антиградієнта.

Траєкторія руху по антиградієнту $w_j(t)$ приводить процес навчання до деякої особливої точки. Якщо вихідна точка $w_j(0)$ знаходилася в області притягання глобального екстремуму, то відповідна траєкторія приведе до глобального мінімуму функції $E_j(t)$. Якщо ж точка $w_j(t)$ не належала до області притягання глобального екстремуму, то рух у напрямку антиградієнта приведе в локальний мінімум, з якого неможливо вибратися під впливом сил,

спрямованих по антиградієнту. Саме в таких випадках виявляється корисним включення в дельта-правило навчання деякого випадкового механізму. Випадкові поштовхи можуть допомогти точці $w_j(t)$ подолати бар'єр, що відокремлює локальний мінімум, у який потрапив процес навчання, від області, у якій цільова функція $E_j(t)$ може ще спадати. Такий рух під впливом детермінованого зносу убік антиградієнта і випадкових поштовхів визначається диференціальним рівнянням

$$\frac{dw_j(t)}{dt} = -\eta \nabla_{w_j} E_j(t) + \zeta(t), \tag{4.276}$$

де $\zeta(t)$ – $(n+1)$-вимірний нормальний випадковий процес з нульовим математичним сподіванням, дельтоподібною автокореляційною функцією і дисперсією складових $\sigma_0^2, \sigma_1^2, \ldots, \sigma_n^2$.

У [269] доведено, що в загальному випадку цей алгоритм забезпечує відшукання глобального екстремуму.

Розумно підбираючи дисперсії σ_i^2 у процесі пошуку, тобто адаптуючи до конкретної форми цільової функції, можна значно прискорити відшукання глобального екстремуму. При цьому адаптацію в процес такого пошуку можна ввести двояким чином.

По-перше, вводячи інерційність у процес навчання, отримаємо пошук, аналогічний руху за методом «важкої кульки» [322]. Такий рух описується диференціальним рівнянням:

$$\frac{d^2 w_j(t)}{dt^2} + b\frac{dw_j(t)}{dt} = -\eta \nabla_{w_j} E_j(t) + \zeta(t), \tag{4.277}$$

де b – коефіцієнт демпфування (чим більше b, тим менше позначається введена інерційність).

У дискретному часі виразу (4.277) відповідає алгоритм навчання, описуваний різницевим рівнянням другого порядку

$$w_j(k+1) = w_j(k) + \beta w_j(k-1) - \eta \nabla_{w_j} E_j(k) + \zeta(k), \tag{4.278}$$

співпадаючий при $\beta = 0$ з блукаючим випадковим пошуком.

По-друге, адаптація в процесі глобального пошуку може бути введена шляхом відповідного керування випадковим процесом $\zeta(t)$, наприклад, у такий спосіб:

$$\frac{d\zeta(t)}{dt} = -\gamma\zeta(t) - \eta_\zeta \frac{dE_j(t)}{dt} + \sigma_\zeta^2 H(t), \tag{4.279}$$

де $\gamma > 0$ – параметр автокореляції випадкового процесу $\zeta(t)$; σ_ζ^2 – величина, що визначає дисперсію $\zeta(t)$; $H(t)$ – векторний білий шум.

У дискретному випадку рівнянню (4.279) відповідає рекурентний алгоритм настроювання

$$\zeta(k+1) = (1-\gamma)\zeta(k) - \eta_\zeta \Delta E_j(k-1) + \sigma_\zeta^2 H(k). \tag{4.280}$$

Як видно з (4.279), (4.280), оптимізація процесу пошуку може проводитися за рахунок відповідного вибору параметрів γ, η_ζ і σ_ζ^2, кожен з яких впливає на визначену характеристику процесу пошуку. Так, варіюючи величиною параметра автокореляції γ, який визначає швидкість згасання процесу $\zeta(t)$ і, отже, ступінь його зв'язку з минулим, можна впливати на характер випадкового пошуку, тобто за необхідності зробити його більш-менш залежним від передісторії.

Цікавою є взаємодія параметрів γ и η_ζ. Якщо крок пошуку η_ζ визначає інтенсивність процесу накопичення досвіду навчання, то γ характеризує рівень забування цього досвіду під час пошуку. У цьому сенсі дані параметри є антагоністичними. Якщо $\gamma = 0$, то забування немає взагалі і вектор $\zeta(t)$ зростає в напрямку антиградієнта.

Дисперсія процесу $\zeta(t)$ визначається величиною σ_ζ^2 та інтенсивністю білого шуму $H(t)$. За великого значення σ_ζ^2 процес пошуку може стати хитливим (пошук «розносить»), за малого – погіршуються глобальні властивості.

Вводячи в процес пошуку режим самонавчання, заснований на аналізі реакцій (4.237), можна істотно поліпшити його глобальні характеристики. При цьому необхідно пам'ятати, що в режимі глобального пошуку ці реакції повинні мати двоякий характер: з одного боку – це негайна реакція як у локальному пошуку, спрямована на усунення результатів невдалого кроку, а з іншого боку – за допомогою механізму самонавчання мають перебудовуватися статистичні характеристики процесу $\zeta(t)$.

Таким чином, частина результату навчання приходиться на один, а інша частина отриманого ефекту – на інший тип реакції. Виключення однієї з цих реакцій не позбавляє алгоритм пошуку здатності до оптимізації. На рис. 4.14 для порівняння наведено три схеми навчання за різних комбінацій негайної реакції і самонавчання.

На схемі а) показано випадковий пошук без самонавчання, що працює тільки з урахуванням негайної реакції. У цьому випадку ймовірнісні характеристики випадкового кроку ξ незмінні.

Наступна схема б) відбиває застосування самонавчання разом з алгоритмом пошуку. Тут імовірнісні характеристики випадкових кроків переналаштовуються відповідним чином по каналу зворотного зв'язку за одночасної роботи алгоритму пошуку.

Остання схема в) відповідає навчанню мережі тільки на основі алгоритму самонавчання, коли негайна реакція виключена. У цьому випадку настроювання мережі здійснюється тільки за рахунок перебудови ймовірнісних характеристик пошуку. Як неважко помітити, подібного роду процедура навчання перш ніж перестроїтися на новий напрямок, може зробити кілька кроків у старому напрямку, незалежно від отриманих результатів. У такий спосіб алгоритм навчання може певний час «підніматися по схилу», долаючи тим самим «хребти» цільової функції, забезпечуючи пошуку глобальний характер.

Розглянемо саме такий глобальний пошук, отриманий за рахунок виключення негайної реакції на невдалий крок.

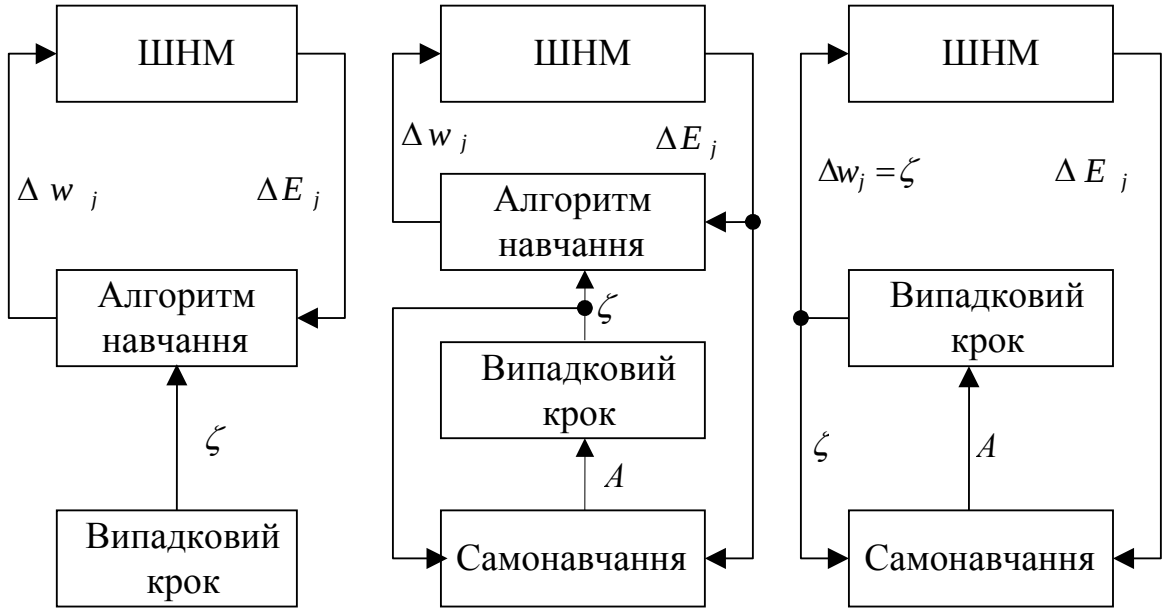


Рис. 4.14. Схеми навчання на основі випадкового пошуку:
а) без самонавчання, б) із самонавчанням і алгоритмом пошуку,
в) із самонавчанням без алгоритму пошуку

Нехай напрямок випадкових кроків пошуку в просторі параметрів, що настроюються, визначається заданим багатовимірним розподілом $p(\zeta, s)$, що залежить від деякого $(n+1)$ – вимірного одиничного вектора $s = (s_0, s_1, \ldots, s_n)^T$ як від параметра. Розподіл $p(\zeta, s)$ повинен мати таку властивість: напрямок математичного сподівання випадкового вектора ζ по всіх можливих реалізаціях має збігатися з напрямком вектора s, тобто

$$dir \int \zeta \, p(\zeta, s) d\zeta = s. \qquad (4.281)$$

Отже, s визначає середній напрямок пошуку. З іншого боку, цей напрямок має залежати від передісторії процесу пошуку, тобто бути найкращим з погляду попередньої роботи. Тому природно називати напрямок s вектором попереднього досвіду [269]. Можна помітити, що цей вектор схожий на вектор пам'яті A у процесі неперервного самонавчання. Різниця полягає в тому, що в цьому випадку вектор s вказує лише напрямок, а його норма не несе жодної інформації на противагу вектору пам'яті, норма якого визначає дисперсні властивості випадкового вибору.

Процес навчання на основі глобального пошуку розбивається на низку етапів.

Під час першого етапу (аналізу) з вихідної точки $w_j(k)$, що визначає стан ШНМ у поточний момент часу, робиться Q незалежних проб $\eta_\zeta \zeta^q$, $q=1,2,\ldots,Q$ відповідно до розподілу $p(\zeta, s(k))$. При цьому щораз визначається значення цільової функції $E_j\left(w_j(k)+\eta_\zeta\zeta^q\right)$.

На другому етапі (вирішенні) визначається напрямок робочого кроку, що залежить від результатів аналізу, зробленого на першому етапі і вирішального правила $D(\bullet)$, що об'єднує попередній досвід і отриману інформацію

$$w_j(k+1)= \\ = w_j(k)+\eta D\left(\zeta^1,\zeta^2,\ldots,\zeta^Q, E_j\left(w_j(k)+\eta_\zeta\zeta^1\right)\ldots, E_j\left(w_j(k)+\eta_\zeta\zeta^Q\right),\ s(k)\right), \quad (4.282)$$

де D – векторна одинична функція, що визначає найкращий у деякому сенсі напрямок робочого кроку у світлі тільки що отриманої інформації.

Тому подальший пошук варто скерувати саме в цьому напрямку, а оскільки напрямок пошуку визначається вектором досвіду, то на третьому етапі (навчання) природно змінити напрямок s відповідно до отриманих результатів

$$s(k+1)=D_s\left(s(k),\Delta w_j(k)\right), \quad (4.283)$$

наприклад

$$s(k+1)=\eta^{-1}\Delta w_j(k), \quad (4.284)$$

тобто у напрямку попереднього робочого кроку.

У випадку, якщо навчання відбувається в обстановці перешкод і немає впевненості, що напрямок s дійсно є найкращим, необхідне введення накопичення, що, наприклад, реалізується у вигляді

$$s(k+1)=dir\left(s(k)+\eta_s\Delta w_j(k)\right), \quad (4.285)$$

де η_s – параметр швидкості накопичення досвіду.

Співвідношення (4.285) встановлює наступність між новим і старим напрямками вектора досвіду, що за малих значень параметра η_s велика, а за великих – мала. В останньому випадку вираз (4.285) наближається до (4.284).

Неважко побачити, що цей алгоритм має глобальний характер. Дійсно, спробні кроки $\zeta^1,\zeta^2,\ldots,\zeta^Q$ тут відбуваються не в будь-якому, а лише у визначеному кращому секторі напрямків, обумовленому вектором s. Цей вектор, а точніше розподіл $p(\zeta,s)$ немовби встановлює своєрідні «шори», що обмежують напрямки випадкових проб лише у визначеному секторі простору параметрів, що настроюються. Напрямок робочого кроку при цьому визначається за правилом D, виходячи з отриманої в такий спосіб інформації. Внаслідок того, що напрямок пошуку s не може значно змінитися за один крок, такий пошук набуває визначену інерційність. Наявність розподілу $p(\zeta,s)$, що визначає напрямок кроків пошуку, забезпечує «плавність» траєкторії навчання, що подібна траєкторії руху важкої точки. За наявності яру в цільовій функції пошук відбуватиметься уздовж цього яру незалежно від того, піднімається він

чи опускається. Це дозволяє долати «хребти» за «перевалами» критерію якості та відшукувати нові райони його локальних низин.

Розглянутий підхід не знаходить глобальний оптимум цільової функції, а лише виділяє ті області простору параметрів, де він може знаходитися.

На практиці для вирішення конкретних задач доцільним є використання більш простих алгоритмів.

Насамперед можна відзначити алгоритм глобального пошуку, у якого $p(\zeta, s)$ є дискретним розподілом. Випадковий вектор пробного кроку ζ, обраний відповідно до цього розподілу, має координати $\zeta_0, \zeta_1, \ldots, \zeta_n$, які визначаються відповідно до правила

$$\zeta_i = \begin{cases} \dfrac{1}{\sqrt{n+1}} & \text{з імовірністю } p_i, \\ -\dfrac{1}{\sqrt{n+1}} & \text{з імовірністю } 1 - p_i, \end{cases} \tag{4.286}$$

де ймовірність p_i обчислюється за формулою

$$p_i = \frac{1 + (1-2c)\, s_i \sqrt{n+1}}{2}, \quad i = 1, 2, \ldots, n, \tag{4.287}$$

s_i - i -а компонента вектора $s = (s_0, s_1, \ldots, s_n)^T$, $0 < c < 1$ – деяка константа. Напрямок робочого кроку для цього алгоритму природно визначити як

$$D = \zeta^*, \tag{4.288}$$

де ζ^* – напрямок найкращої проби, що задовольняє умові

$$E_j\left(w_j(k) + \eta_\zeta \zeta^*\right) = \min_{q=1,2,\ldots Q} E_j\left(w_j(k) + \eta_\zeta \zeta^q\right). \tag{4.289}$$

Досвід, накопичений за один цикл аналізу, запам’ятовується в цьому випадку у вигляді вектора s, який збігається з найкращою пробою

$$s = \zeta^*. \tag{4.290}$$

Таким чином, цілеспрямованість пошуку досягається за рахунок того, що ймовірності (4.287) приймають одне з двох значень: c або $1-c$ залежно від передісторії.

Досить ефективним є також так званий алгоритм із направляючою сферою [346], що відрізняється від попереднього тим, що замість дискретного він використовує неперервне самонавчання.

Нехай випадкові проби ζ^q визначаються точками на поверхні $(n+1)$-вимірної гіперсфери, а сама ця гіперсфера дещо висунута в напрямку вектора s. Тоді утворені в такий спосіб випадкові напрямки мають тенденцію убік вектора досвіду, причому ця тенденція тим сильніше виражена, чим на більшу величину висунута гіперсфера уздовж вектора s. Напрямок випадкового кроку в цьому алгоритмі визначається виразом

$$\zeta = dir\left(s + r\zeta^0\right), \tag{4.291}$$

де ζ^0 – випадковий одиничний вектор, рівноймовірно розподілений в усіх напрямках простору параметрів; r – радіус гіперсфери.

На рис. 4.15 показано взаємодію векторів ζ^0 і s у процесі утворення ζ (пунктиром позначено гіперсферу можливих реалізацій). Як видно, при $r>1$ усі спробні кроки відбуваються усередині гіперконуса з віссю s і кутом напіврозкриття $\arcsin\frac{r}{\|s\|}$.

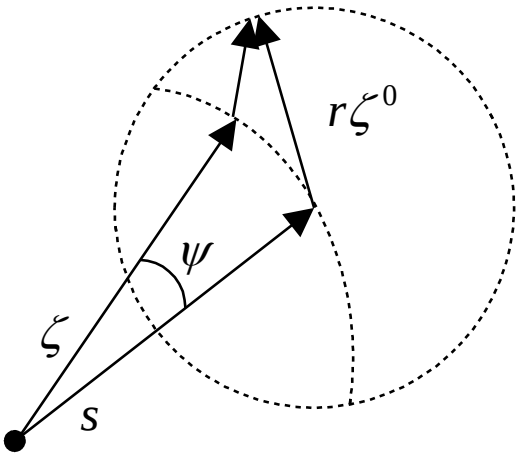


Рис. 4.15. Визначення напрямку в алгоритмі з направляючою сферою

Нескладно також помітити, що чим менше r, тим вужчий конус і тим ближче одне до одного випадкові проби.

Модифікацією алгоритму з направляючою сферою є алгоритм із направляючим конусом. Нехай у просторі параметрів, що настроюються, визначений гіперконус з вершиною в точці w_j, вісь якого збігається з напрямком вектора s, а кут при вершині дорівнює 2ψ. Навколо вершини конуса, як щодо центра, проводиться гіперсфера радіуса η_ζ. Конус відтинає від цієї сфери частину поверхні, на якій випадково вибирається Q спробних точок $\zeta^1,\zeta^2,\ldots,\zeta^Q$. За значеннями цільової функції в цих точках $E_j\left(w_j+\zeta^q\right)$ визначається найкраща точка, що відповідає мінімальному значенню критерію якості (4.289). У цьому напрямку і відбувається робочий крок. Напрямок пошуку в такий спосіб цілком визначається зазначеним конусом, тобто випадкові проби обираються усередині нього. Напрямок вектора досвіду s при цьому визначається найкращою пробою попереднього етапу (4.290).

На рис. 4.16 показано кілька кроків пошуку для $\eta_\zeta<1$, $Q=n+1=2$ зі стану $w_j(0)$ з довільним початковим напрямком вектора $s(0)$, який у процесі пошуку коректується за найкращою пробою.

На рис. 4.16 видно, що в міру накопичення інформації про поводження цільової функції вектор s прагне розгорнутися в напрямку, зворотному градієнтному.

Очевидно, що зі зменшенням кута розкриття конуса, унаслідок інерційності такого роду пошуку, можливості повороту вектора s

зменшуються. Це означає, що за різкої зміни напрямку градієнта алгоритм певний час рухатиметься в старому напрямку, а потім вектор s поступово перестроюється на новий правильний напрямок.

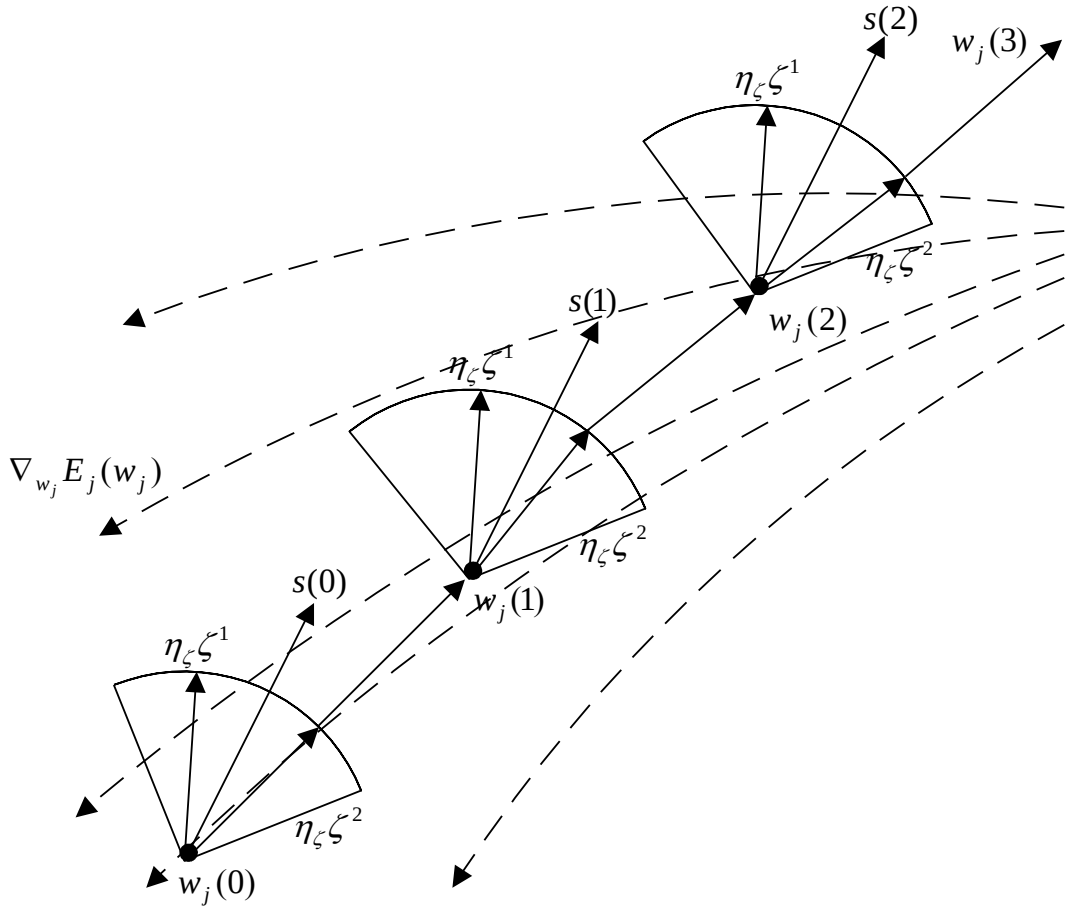


Рис. 4.16. Навчання в просторі параметрів w_j
на основі глобального випадкового пошуку з направляючим конусом

Втрати на пошук для такого алгоритму при правильному розташуванні вектора s (у напрямку яру або антиградієнтному) зменшуються зі зменшенням кута розкриття конуса. Збільшення кута ψ приводить до більшої мобільності і «верткості» пошуку, проте при цьому зростають і втрати на пошук.

Очевидно, що оптимальні значення параметрів пошуку цілком залежать від вигляду цільової функції $E_j\left(w_j\right)$ і її особливостей. У задачах, де ця функція через велику складність має багато екстремумів, очевидно, найбільш доцільним є використання глобального випадкового пошуку із самонавчанням [361].

Алгоритми еволюційного планування. Наприкінці 50-х років минулого сторіччя відомим англійським статистиком Дж. Боксом було запропоновано підхід до оптимізації технологічних процесів, що отримав назву еволюційного планування (EVOP) [350] та породив множину алгоритмів оптимізації, не потребуючи обчислення похідних цільової функції й ефективно працюючих в умовах значної «спотвореності» спостережень. І хоча в «чистому вигляді»

EVOP сьогодні практично не застосовується (тим більше для навчання нейронних мереж), з методологічної точки зору корисно розглянути ідеї, покладені в його основу.

Основна ідея полягає в тому, що достатньо довільним чином обирається деякий вектор параметрів ${}_0w_j(k)$, іменований базовою точкою, і оцінюється значення цільової функції $E_j\left({}_pw_j(k)\right)$ у точках, що оточують базову. Цей набір точок, включаючи базову, називається зразком. У двовимірному випадку – це квадратний зразок, показаний на рис. 4.17.

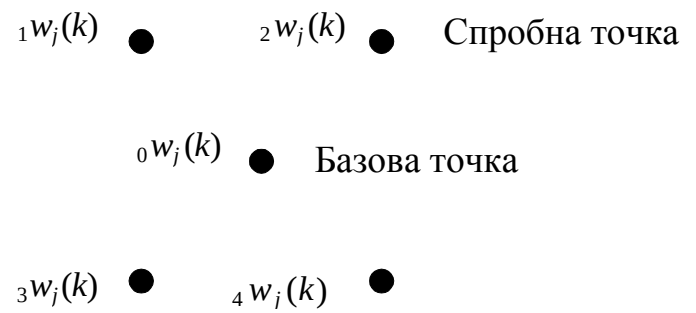


Рис. 4.17. Квадратний зразок

Потім найкраща з п’яти досліджуваних точок ${}_0w_j(k), {}_1w_j(k),\ldots,{}_4w_j(k)$, що відповідає мінімальному значенню цільової функції $E_j\left({}_pw_j(k)\right)$, $p=0,1,\ldots,4$, що визначається як ${}_lw_j(k)$, обирається як базова для наступного кроку оптимізації:

$${}_0w_j(k+1)={}_l\,w_j(k) \tag{4.292}$$

і навколо неї будується аналогічний зразок.

Якщо жодна зі спробних кутових точок не має переваг перед базовою, розміри зразка зменшуються, після чого пошук оптимуму продовжується.

У задачах більшої розмірності обчислення значень цільової функції відбувається у всіх вершинах, а також у центрі гіперкуба, тобто в точках так званого гіперкубічного зразка, а кількість обчислень цільової функції складає при цьому

$$Q=2^n+1, \tag{4.293}$$

де n – розмірність простору, де відбувається оптимізація.

Тому, незважаючи на логічну простоту пошуку за кубічним зразком, виникає необхідність використання більш ефективних методів.

Подальшим розвитком еволюційного планування є обертальне еволюційне планування (ROVOP) [353], у якому зразок розвертається навколо базової точки і може як збільшувати, так і зменшувати свої розміри. Незважаючи на те, що пошук на основі ROVOP здобуває глобальні властивості, значна кількість спостережень і слабка формалізація обмежують його застосування.

Набагато більш ефективним алгоритмом еволюційного планування є послідовний симплексний пошук, запропонований У. Спендлі, Дж. Хекстом і

Ф. Гімсвортом [351]. Слід зазначити, що цей алгоритм не має жодного відношення до симплекс-методу лінійного програмування, а подібність назв має випадковий характер. Процедура симплексного пошуку базується на тому, що експериментальним зразком, що містить найменшу кількість точок, є регулярний симплекс. Симплекс у n-вимірному просторі (далі для збереження позначень, традиційно прийнятих у геометрії симплексів, ми вважатимемо, що кількість параметрів, що настроюються, дорівнює n) являє собою багатогранник, утворений $n+1$ рівновіддаленими один від одного точками-вершинами. Аналітично – це множина точок вигляду

$$ {}_j w = {}_1 w_j \lambda_1 + {}_2 w_j \lambda_2 + \cdots + {}_{n+1} w_j \lambda_{n+1}, \qquad (4.294) $$

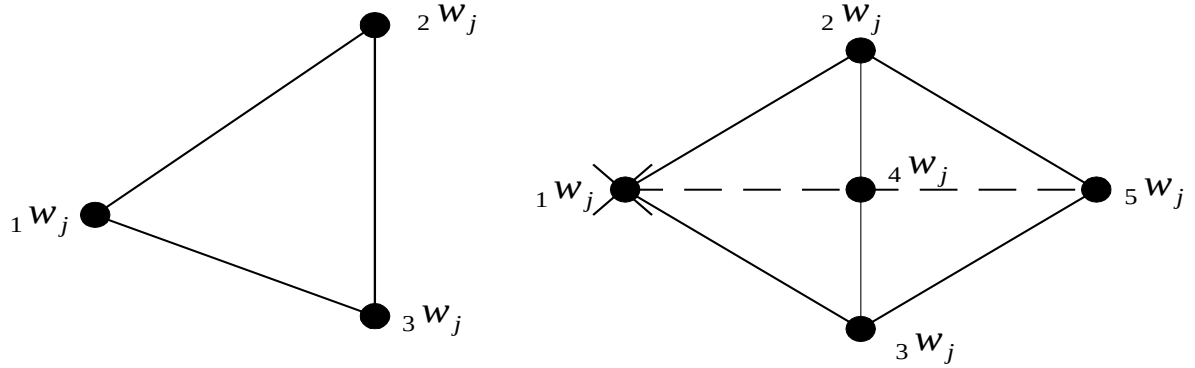
де λ_p – коефіцієнти, що задовольняють обмеженням

$$ \sum_{p=1}^{n+1} \lambda_p = 1,\ \lambda_p \geq 0,\ p = 1,2,\ldots,n+1, \qquad (4.295) $$

а ${}_p w_j = \left({}_p w_{j1}, \ldots, {}_p w_{ji}, \ldots, {}_p w_{jn} \right)^T$ – координати p-ї вершини симплекса в n-вимірному просторі.

Нескладно побачити, що у випадку двох параметрів-змінних симплексом є трикутник, у тривимірному просторі симплекс – це тетраедр тощо.

В алгоритмі симплексного пошуку використовується важлива властивість симплексів, відповідно до якої новий симплекс можна побудувати на будь-якій грані початкового симплекса шляхом перенесення обраної вершини на належну відстань уздовж прямої, проведеної через центр ваги інших вершин вихідного симплексу. Отримана в такий спосіб точка є вершиною нового симплекса, а обрана в ході побудови вершина початкового симплекса виключається. Таким чином, з переходом до нового симплекса потрібно тільки одне обчислення значення цільової функції. Рис. 4.18 ілюструє процедуру побудови симплекса на площині $(n=2)$.



а) б)

Рис. 4.18. Побудова нового симплекса:
а) початковий симплекс ${}_1 w_j$, ${}_2 w_j$, ${}_3 w_j$, б) новий симплекс ${}_2 w_j$, ${}_3 w_j$, ${}_5 w_j$

Робота алгоритму симплексного пошуку починається з побудови регулярного симплекса в просторі параметрів, що настроюються, і оцінювання значень цільової функції в кожній з вершин. При цьому визначається вершина, якій відповідає найбільше значення цільової функції. Потім ця «найгірша» вершина проектується через центр ваги інших вершин симплекса в нову точку, що використовується як вершина нового симплекса. Якщо цільова функція спадає досить плавно, ітерації продовжуються доти, поки симплексом не буде «накрито» точку мінімуму, про що свідчить або циклічний рух навколо однієї вершини, або повторювані відбиття-коливання через те саме ребро багатогранника.

В умовах перешкод симплекс-пошук знаходить область мінімуму з точністю до своїх розмірів і зациклюється навколо мінімуму. Якщо ж точка мінімуму дрейфуватиме, то і симплекс піде за нею, описуючи спіраль навколо траєкторії оптимуму, що зміщається в часі. Ця властивість покладена в основу принципу адаптаційної оптимізації [353], на основі якого крім власне симплекс-методу може бути побудовано досить багато алгоритмів.

Формально симплекс-пошук можна описати в такий спосіб. З аналітичної геометрії відомо, що координати вершин регулярного симплексу визначаються матрицею W_j, у якій стовпці є вершинами, пронумерованими від 1 до $n+1$, а рядки – координатами від 1 до n [360]:

$$W_j = \begin{pmatrix} {}_1w_{j1} & {}_2w_{j1} & \cdots & {}_{n+1}w_{j1} \\ {}_1w_{j2} & {}_2w_{j2} & \cdots & {}_{n+1}w_{j2} \\ \vdots & \vdots & \ddots & \vdots \\ {}_1w_{jn} & {}_2w_{jn} & \cdots & {}_{n+1}w_{jn} \end{pmatrix} = \begin{pmatrix} 0 & \rho_1 & \rho_2 & \cdots & \rho_2 \\ 0 & \rho_2 & \rho_1 & \cdots & \rho_2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \rho_2 & \rho_2 & \cdots & \rho_1 \end{pmatrix} = \tag{4.296}$$

$$= \left({}_1w_j, \ldots, {}_pw_j, \ldots, {}_{n+1}w_j \right),$$

де

$$\begin{cases} \rho_1 = \dfrac{\eta_s}{n\sqrt{2}}\left(\sqrt{n+1}+n-1\right), \\ \rho_2 = \dfrac{\eta_s}{n\sqrt{2}}\left(\sqrt{n+1}-1\right); \end{cases} \tag{4.297}$$

η_s – параметр, що задає відстань між вершинами.

Нехай ${}_pw_j(k) = \left({}_pw_{j1}(k), {}_pw_{j2}(k), \ldots, {}_pw_{jn}(k) \right)^T$ є p-ю вершиною симплекса на k-му кроці пошуку і нехай $E_j\left({}_pw_j(k)\right)$ – значення цільової функції в цій вершині. Крім того, відзначимо ті вершини, у яких на k-й ітерації цільова функція приймає максимальне і мінімальне значення:

$$\begin{cases} E_j\left({}_hw_j(k)\right) = \max\limits_{p=1,2,\ldots,n+1}\left\{E_j\left({}_1w_j(k)\right), \ldots, E_j\left({}_pw_j(k)\right), \ldots, E_j\left({}_{n+1}w_j(k)\right)\right\}, \\ E_j\left({}_lw_j(k)\right) = \min\limits_{p=1,2,\ldots,n+1}\left\{E_j\left({}_1w_j(k)\right), \ldots, E_j\left({}_pw_j(k)\right), \ldots, E_j\left({}_{n+1}w_j(k)\right)\right\}. \end{cases} \tag{4.298}$$

Визначивши центр ваги усіх вершин симплекса за винятком «найгіршої» ${}_h w_j(k)$ у вигляді

$$ {}_{n+2} w_j(k) = \frac{1}{n}\left(\sum_{p=1}^{n+1} {}_p w_j(k) - {}_h w_j(k) \right), \tag{4.299} $$

можна записати алгоритм відбиття-руху симплекса в просторі параметрів, що настроюються:

$$ {}_{n+3} w_j(k) = 2_{n+2} w_j(k) - {}_h w_j(k) = {}_p w_j(k+1). \tag{4.300} $$

Відбита точка ${}_{n+3} w_j(k)$ і буде новою вершиною симплекса на $(k+1)$-й ітерації пошуку.

Алгоритм симплексного пошуку характеризується не тільки обчислювальною простотою, але й високою ефективністю. Якщо як міру ефективності розглядати кількість вимірів цільової функції на кожному кроці і помилку визначення напрямку градієнта, то якість методу можна оцінити за допомогою теореми Брукса [345, 362], що свідчить про те, що максимум критерію

$$ E_s = \frac{1}{Q} \mathrm{M}\{\cos\Theta\}, \tag{4.301} $$

де Q – кількість вимірів на кожному кроці; Θ – помилка визначення градієнта, досягається при $Q = n+1$.

Виходячи з цього критерію, можна судити про високу ефективність симплексного пошуку, що перевершує випадковий пошук у середньому в 1.4 рази [363].

Прискорення пошуку можна домогтися, відмовившись від регулярності симплекса і керуючи його розмірами так, як це робиться в алгоритмі деформованого багатогранника Нелдера–Міда [352, 364].

У цьому алгоритмі замість відбиття, що задається формулою (4.300), використовуються такі операції:

– відбиття – проектування ${}_h w_j(k)$ через центр ваги відповідно до виразу

$$ {}_{n+3} w_j(k) = {}_{n+2} w_j(k) + \alpha\left({}_{n+2} w_j(k) - {}_h w_j(k) \right); \tag{4.302} $$

– розтягання, що відбувається у випадку, якщо $E_j\left({}_{n+3} w_j(k) \right) \le E_j\left({}_l w_j(k) \right)$, і полягає в тому, що вектор ${}_{n+3} w_j(k) - {}_{n+2} w_j(k)$ «розтягується» відповідно до виразу

$$ {}_{n+4} w_j(k) = {}_{n+2} w_j(k) + \gamma\left({}_{n+3} w_j(k) - {}_{n+2} w_j(k) \right), \tag{4.303} $$

при цьому якщо $E_j\left({}_{n+4} w_j(k) \right) < E_j\left({}_l w_j(k) \right)$, то ${}_h w_j(k)$ замінюється на ${}_{n+4} w_j(k)$ і здійснюється перехід на відбиття-проектування при $k+1$. У протилежному випадку ${}_h w_j(k)$ замінюється на ${}_{n+3} w_j(k)$ і також відбувається відбиття на $(k+1)$-му кроці;

– стиск, що відбувається у випадку $E_j\left({}_{n+3}w_j(k)\right) > E_j\left({}_{l}w_j(k)\right)$ і полягає в тому, що вектор ${}_{h}w_j(k) - {}_{n+2}w_j(k)$ «стискується» відповідно до виразу

$${}_{n+5}w_j(k) = {}_{n+2}w_j(k) + \beta\left({}_{n+3}w_j(k) - {}_{n+2}w_j(k)\right). \qquad (4.304)$$

Потім ${}_{h}w_j(k)$ замінюється на ${}_{n+5}w_j(k)$ і відбувається повернення до операції відбиття-проектування для продовження пошуку на $(k+1)$-му кроці;

– редукція, що відбувається у випадку $E_j\left({}_{n+3}w_j(k)\right) > E_j\left({}_{h}w_j(k)\right)$ і полягає в тому, що усі вектори ${}_{p}w_j(k) - {}_{l}w_j(k), \quad p = 1,2,\ldots,n+1$, зменшуються вдвічі з відліком від ${}_{l}w_j(k)$ відповідно до виразу

$${}_{p}w_j(k) = {}_{l}w_j(k) + 0.5\left({}_{p}w_j(k) - {}_{l}w_j(k)\right). \qquad (4.305)$$

Потім відбувається повернення до операції відбиття-проектування для продовження пошуку на $(k+1)$-й ітерації.

Деформований багатогранник на противагу правильному симплексу адаптується до топології цільової функції, витягаючись уздовж довгих похилих поверхонь, змінюючи напрямок у вигнутих ярах і стискуючись в околиці мінімуму.

Коефіцієнт відображення α використовується для проектування вершини з найбільшим значенням $E_j\left({}_{h}w_j(k)\right)$ через центр ваги багатогранника, що деформований. Коефіцієнт γ уводиться для розтягання вектора пошуку у випадку, якщо відбиття дає вершину зі значенням $E_j\left({}_{n+3}w_j(k)\right)$ меншим, ніж найменше значення цільової функції, отримане до відбиття.

Коефіцієнт стиску β використовується для зменшення вектора пошуку, якщо операція відбиття не призвела до поліпшення результату порівняно з $E_j\left({}_{l}w_j(k)\right)$. Таким чином, параметри α, β і γ забезпечують адаптацію деформованого багатогранника до топології цільової функції, а їхній обґрунтований вибір впливає на результат вирішення задачі.

Після того, як деформований вихідний багатогранник промасштабовано належним чином, наприклад за допомогою співвідношень (4.296), (4.297), його розміри в процесі навчання мають залишатися постійними, поки характер цільової функції не зажадає симплекса іншої форми. Це можна реалізувати тільки при $\alpha = 1$. Крім того, Дж. Нелдер і Р. Мід показали [352], що з використанням $\alpha < 1$ кількість ітерацій методу зростає, а при $\alpha > 1$ деформований багатогранник гірше адаптується до змін цільової функції, особливо за наявності вигнутих ярів.

Щоб з'ясувати, як впливає на процедуру пошуку вибір β і γ, у [360, 364, 362] було проведене вирішення тестових задач з різними комбінаціями коефіцієнтів симплекса. Як прийнятні значення рекомендується прийняти $\alpha = 1 \quad \beta = 0.5, \gamma = 2$. Розміри та орієнтація вихідного симплекса

деякою мірою впливають на кількість ітерацій, але значення α, β, γ впливають значно більше. Водночас було встановлено, що чітко вирішити питання щодо вибору β і γ неможливо і що вплив вибору β на ефективність пошуку більш помітний, ніж вплив вибору γ. Ці параметри слід вибирати з діапазонів $0.4 \le \beta \le 0.6;\ 2.8 \le \gamma \le 3.0$.

Ідеологія адаптаційної оптимізації породила множину алгоритмів, в основі яких лежить симплексний рух.

Для того, щоб наблизити рух симплекса до антиградієнтного, як центр відображення можна взяти зважений центр ваги (алгоритм Умеди–Ічикави [353]), в алгоритмах Горського–Адлера пропонується зміщувати центр ваги симплекса в антиградієнтому напрямку, інформація про який міститься у вершинах симплекса, у [365] розглянутий симплекс-пошук, що має властивості стохастичної апроксимації. Так чи інакше, в основі всіх цих алгоритмів крім відбиття лежать операції розтягання, стиску і редукції, відмовлення від яких у ряді випадків дозволяє не тільки спростити алгоритм пошуку, але й підвищити його швидкість.

Запишемо процес руху деформованого багатогранника у вигляді

$$\alpha\left({}_{n+2}w_j(k) - {}_{h}w_j(k)\right) = {}_{n+3}w_j(k) - {}_{n+2}w_j(k) \tag{4.306}$$

і зажадаємо, щоб на кожному кроці пошуку центр ваги симплекса зміщувався в антиградієнтому напрямку:

$$\bar{w}_j(k+1) = \bar{w}_j(k) - \eta \nabla_{w_j} E_j(k), \tag{4.307}$$

де

$$\bar{w}_j(k) = \frac{1}{n}\sum_{\substack{p=1 \\ p \ne h}}^{n+1} {}_{p}w_j(k) \tag{4.308}$$

– центр ваги усіх вершин відбитого симплекса.

Використовуючи замість вектора-градієнта $\nabla_{w_j} E_j(k)$ його оцінку $\nabla_j(k)$, перепишемо (4.307) у формі

$$\frac{\sum_{\substack{p=1 \\ p \ne h}}^{n+1} {}_{p}w_j(k)}{n+1} + \frac{{}_{n+3}w_j(k)}{n+1} = \frac{\sum_{\substack{p=1 \\ p \ne h}}^{n+1} {}_{p}w_j(k)}{n+1} + \frac{{}_{h}w_j(k)}{n+1} - \eta \nabla_j(k), \tag{4.309}$$

яка з урахуванням (4.300) істотно спрощується і набуває вигляду [366]

$${}_{p}w_j(k+1) = {}_{h}w_j(k) - \eta(n+1)\nabla_j(k). \tag{4.310}$$

Оцінка градієнта $\nabla_j(k)$ будується, виходячи з можливості апроксимації цільової функції $E_j(k)$ в околі відбитого симплекса n-вимірною гіперплощиною. Використовуючи ту властивість симплексів, що на кожній ітерації відкидається одна вершина симплекса і додається одна нова вершина, для розрахунку параметрів вектора ∇_j можна використовувати алгоритм поточного регресійного аналізу (4.42), (4.43) у формі

$$\begin{cases} \nabla_j(k+1)=\nabla_j(k)+\dfrac{P(k-1)\left(E_j\left({}_p w_j(k+1)\right)-\nabla_j^T(k)\,{}_p w_j(k+1)\right)}{1+{}_p w_j^T(k+1)P(k-1)\,{}_p w_j(k+1)}\,{}_p w_j(k+1), \\ \tilde{P}(k-1)=P(k-1)+\dfrac{P(k-1)\,{}_h w_j(k)\,{}_h w_j^T(k)P(k-1)}{1-{}_h w_j^T(k)P(k-1)\,{}_h w_j(k)}, \\ P(k)=\tilde{P}(k-1)+\dfrac{\tilde{P}(k-1)\,{}_p w_j(k+1)\,{}_p w_j^T(k+1)\tilde{P}(k-1)}{1+{}_p w_j^T(k+1)\tilde{P}(k-1)\,{}_p w_j(k+1)}. \end{cases} \quad (4.311)$$

Таким чином, рух симплекса можна записати тільки в координатах відбиваної і відбитої вершин і наблизити його до антиградієнтного напрямку.

Використовуючи співвідношення (4.294), (4.296), (4.302), всі алгоритми симплексного пошуку можна записати в узагальненій формі

$$\begin{aligned} &{}_p w_j(k+1)={}_{n+3} w_j(k)= \\ &={}_h w_j(k)\lambda_h+{}_1 w_j(k)\lambda_1+\cdots+{}_{\substack{p\\ p\neq h}} w_j(k)\lambda_p+\cdots+{}_n w_j(k)\lambda_n=w_j^T\lambda, \end{aligned} \quad (4.312)$$

де $\lambda=(\lambda_h,\lambda_1,\ldots,\lambda_n)^T-(n+1)\times 1$ – вектор параметрів, що визначають конкретний алгоритм.

Так

$$\lambda=\left(-1,\frac{2}{n},\ldots,\frac{2}{n}\right)^T \quad (4.313)$$

відповідає регулярному симплексу,

$$\lambda=\left(-\alpha,\frac{1+\alpha}{n},\ldots,\frac{1+\alpha}{n}\right)^T \quad (4.314)$$

– алгоритму Нелдера–Міда і т.д.

Симплексний пошук породив множину модифікацій, серед яких насамперед можна відзначити комплекс-метод [353, 360], що містить у собі методологію симплекс-методу і випадкового пошуку. У цьому методі замість $(n+1)$-вершинного симплекса використовується сукупність («хмара») точок, обраних випадковим чином. Їхнє число Q має бути не менше, ніж $n+1$. У «хмарі» виділяється «погана» точка ${}_h w_j(k)$, відповідна найбільшому значенню цільової функції $E_j\left({}_h w_j(k)\right)$. Замість протилежної грані використовується центр ваги «хмари» з виключеною точкою ${}_h w_j(k)$. Якщо позначити цей центр як ${}_{Q+1} w_j(k)$, то нова точка комплексу може бути визначена за допомогою співвідношення

$${}_p w_j(k+1)={}_{Q+2} w_j(k)={}_{Q+1} w_j(k)+\alpha\left({}_{Q+1} w_j(k)-{}_h w_j(k)\right). \quad (4.315)$$

При $\alpha>1$ справедлива рівність [367]

$$\left\|{}_p w_j(k+1)-{}_{Q+1} w_j(k)\right\|=\left\|{}_{Q+1} w_j(k)-{}_h w_j(k)\right\|. \quad (4.316)$$

Вибір $\alpha > 1$ призводить до розтягання комплексу, $\alpha < 1$ – до стиску, $Q = n + 1$ перетворює комплекс-пошук (4.315) в алгоритм Нелдера–Міда (4.302).

Як параметри алгоритму комплекс-методу в [202] рекомендовано використовувати $\alpha = 1.3;\ Q = 2n$; у [360] розглянуто процедуру з $Q = n + 2$; у [364] описано комплекс Мітчелла–Каплана, в якому точки «хмари» вибираються не випадково, а деяким регулярним чином.

Нескладно побачити також, що комплекс-метод «вписується» у рамки конструкції (4.312) з

$$\lambda = \left(\lambda_h, \lambda_1, \ldots, \lambda_{Q-1}\right)^T = \left(-\alpha, \frac{1+\alpha}{Q-1}, \ldots, \frac{1+\alpha}{Q-1}\right)^T, \qquad (4.317)$$

при цьому за великих Q він фактично збігається з методом статистичного градієнта (4.243), а рух центра ваги «хмари» наближається до антиградієнтного.

Говорячи про еволюційні алгоритми, не можна не згадати випадкове еволюційне планування (REVOP) [353], що є протилежністю комплекс-методу. Відповідно до процедури REVOP рух здійснюється за правилом

$$w_j(k+1) = w_j(k) + \Delta w_j(k), \qquad (4.318)$$

де $\Delta w_j(k)$ – випадковим чином обраний напрямок руху.

У випадку, якщо $E_j\left(w_j(k+1)\right) < E_j\left(w_j(k)\right)$, то подальший рух продовжується в напрямку $\Delta w_j(k)$, у протилежному випадку вибирається новий випадковий напрямок $\Delta w_j(k+1)$. Нескладно побачити, що випадкове еволюційне планування збігається з ROM-алгоритмом випадкового пошуку (4.231), (4.233).

Введення в REVOP випадкового блукання в поєднанні із самонавчанням дозволяє задетермінувати рух у сприятливих напрямках і захиститися від «зупинок» у локальних екстремумах. Для цього можна використовувати адаптивне випадкове еволюційне планування у формі [368, 369]

$$w_j(k+1) = w_j(k) - \eta \nabla_j(k) + \zeta(\Delta E_j(k-1)), \qquad (4.319)$$

де $\nabla_j(k)$ – оцінка градієнта; $\zeta\left(\Delta E_j(k-1)\right)$ – випадкова добавка, у якої дисперсія компонентів $\sigma_w^2(k)$ визначається збільшенням цільової функції $\Delta E_j(k-1) = E_j\left(w_j(k)\right) - E_j\left(w_j(k-1)\right)$.

Оцінку градієнта в цьому випадку на відміну від багатокрокової процедури (4.311) зручніше будувати за допомогою однокрокового адитивного алгоритму Качмажа (4.71), що набуває у цьому випадку вигляд

$$\nabla_j(k+1) = \nabla_j(k) + \frac{E_j\left(w_j(k+1)\right) - \nabla_j^T(k) w_j(k+1)}{l + \left\| w_j(k+1) \right\|^2} w_j(k+1), \qquad (4.320)$$

а для керування дисперсією можна використовувати співвідношення

$$\sigma_w^2(k) = \sigma_\zeta^2 e^{\Delta E_j(k-1)}. \qquad (4.321)$$

Під час руху в сприятливому напрямку $\left(\Delta E_j\left(k-1\right)<0\right)$ – випадковий компонент придушується і рух наближається до антиградієнтного, застрягаючі в локальному мінімумі випадковий компонент має дисперсію σ_ζ^2, у випадку, якщо алгоритм робить невдалий крок $\left(\Delta E_j\left(k-1\right)>0\right)$, випадкова добавка зростаючи за амплітудою, «збиває» рух з невірного напрямку.

У такий спосіб алгоритм (4.319)–(4.321) набуває глобальних властивостей.

Генетичні алгоритми. Генетичні алгоритми сьогодні є найбільш популярними представниками еволюційних алгоритмів і є по суті моделлю розмноження біологічних організмів, що призначена для пошуку глобального оптимуму багатоекстремальних функцій. В основі генетичних алгоритмів лежать механізми натуральної селекції і генетики, що реалізують «виживання найсильніших» серед розглянутих структур у процесі їхньої еволюції. Основна відмінність процесу оптимізації за допомогою генетичних алгоритмів полягає в тому, що вони працюють, як правило, не з параметрами (синаптичними вагами), а з закодованою множиною параметрів, при цьому пошук відбувається з популяції вихідних точок, і для оцінки якості використовують не збільшення цільової функції, а безпосередньо її миттєве значення, застосовуючи при цьому визначені ймовірнісні правила.

Генетичні алгоритми були введені Дж. Холландом [354, 370] і з формальної точки зору є послідовністю операцій, що моделює еволюційні процеси на основі аналогів механізмів генетичного спадкування і природного, а іноді й штучного добору. У загальному вигляді генетичний алгоритм може бути представлений у вигляді схеми, наведеної на рис. 4.19.

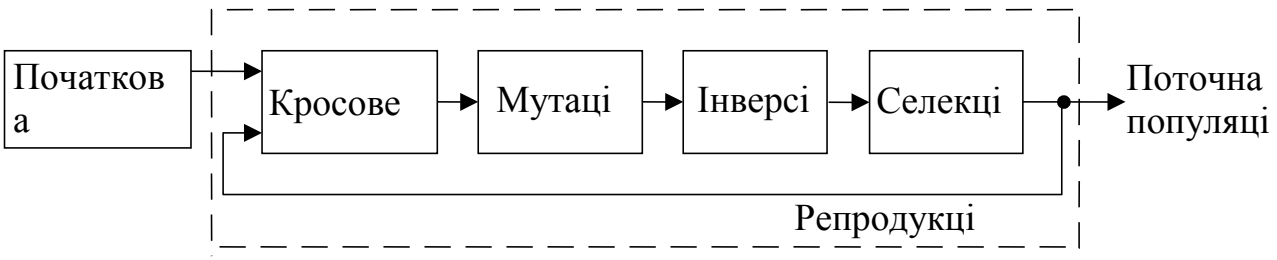


Рис. 4.19. Схема генетичного алгоритму

Для опису генетичних алгоритмів використовується біологічна термінологія, де ключовим поняттям є хромосома (стринг), що є вектором (послідовність, ланцюжок), утворений нулями й одиницями, кожна позиція (біт) якого називається геном. Саме у вигляді хромосом подається вектор синаптичних ваг $w_j\left(k\right)=\left(w_{j1}\left(k\right),w_{j2}\left(k\right),\ldots,w_{jn}\left(k\right)\right)^T$, кодованих або в двійковому, або у форматі з плаваючою точкою. Якщо для кодування кожного параметра, що настроюється $w_{ji}\left(k\right)$ використовується N бітів, то хромосома, яка відповідає вектору синаптичних ваг $w_j\left(k\right)$, має nN генів.

Алгоритм починає свою роботу з генерації (випадковим чином) початкової популяції хромосом $W_j(0)=\left({}_1w_j^T(0),\ldots,{}_pw_j^T(0),\ldots,{}_Qw_j^T(0)\right)$ (тут ${}_pw_j^T(0)=\left({}_pw_{j1}(0),{}_pw_{j2}(0),\ldots,{}_pw_{jn}(0)\right)-p$-а хромосома популяції), розмір якої часто вважається постійним. Для кожної зі згенерированих хромосом можна оцінити її пристосованість (fitness), обумовлену значенням цільової функції $E_j\left({}_pw_j(0)\right)$ для p-го вектора синаптичних ваг. Очевидно, що чим менше значення $E_j\left({}_pw_j(0)\right)$, тим більше шансів «вижити» у популяції, що еволюціонує, у p-ї хромосоми.

Далі починається процес репродукції популяції, що формується генетичними операторами кросовера, мутації, інверсії й операцією селекції. Найважливішим генетичним оператором є кросовер, що формує хромосоми-нащадки шляхом обміну генетичного матеріалу між хромосомами-батьками так, як це показано на рис 4.20.

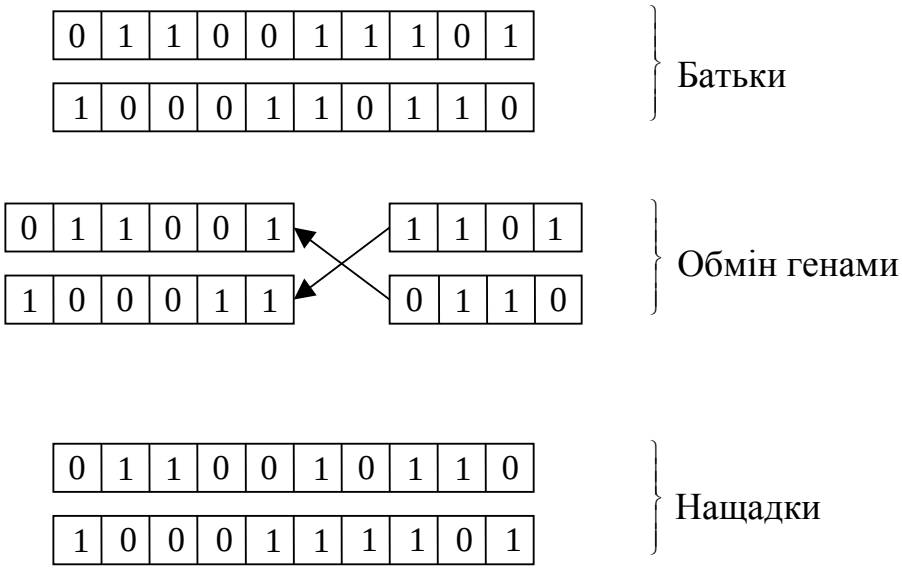


Рис. 4.20. Кросовер

Мутація пов’язана з випадковою зміною одного чи декількох генів у хромосомі так, як це показано на рис. 4.21

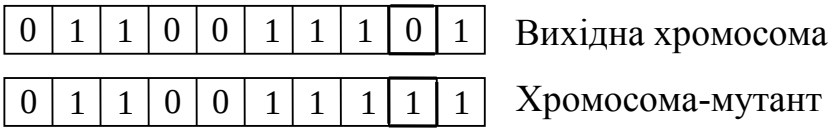


Рис. 4.21. Мутація

Як видно, тут випадково обраний біт змінює свій стан на протилежний. Оператор мутації не дозволяє процесу навчання «застрягти» у локальних екстремумах цільової функції.

Оператор інверсії змінює порядок генів у хромосомі шляхом їхньої циклічної перестановки так, як це показано на рис. 4.22.

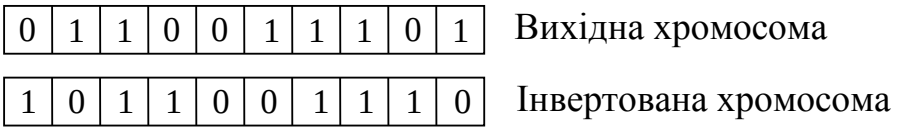


Рис. 4.22. Інверсія

І хоча в задачах оптимізації інверсія застосовується не часто, у навчанні ШНМ цей оператор дозволяє змінювати всі синаптичні ваги стринга ${}_p w_j(k)$.

Внаслідок схрещувань, мутацій та інверсій, що відбулися, формується розширена популяція хромосом, що містить як вихідну множину хромосом-батьків, так і множину нащадків. Кожен стринг розширеної популяції оцінюється з погляду його пристосованості за критерієм $E_j\left({}_p w_j(0)\right)$, $p=1,2,\ldots,Q,\ldots$, після чого формується нова популяція $W_j(1)$, що містить $Q(1)$ хромосом з найменшими значеннями $E_j\left({}_p w_j\right)$. У цьому полягає суть операції селекції.

Далі на кожній ітерації k процес репродукції циклічно повторюється. Таким чином, генетичний алгоритм накопичує вдалі рішення, «стягаючи» популяцію до глобального екстремуму цільової функції.

Схема, наведена на рис. 4.19, ілюструє деяку «універсальну» генетичну процедуру, у рамках якої може бути сформовано множину алгоритмів, що відрізняються один від одного параметрами генетичних операторів і способами селекції. Як найважливіші характеристики, що визначають властивості конкретного генетичного алгоритма, можна відзначити такі [359]:

– спосіб формування вихідної популяції $W_j(0)$;

– кількість особин у вихідній популяції $Q(0)$, яка має бути досить великою, щоб покрити всю область можливих рішень;

– частота кросовера, обумовлена кількістю хромосом у поточній популяції, що піддаються схрещуванню;

– ймовірність кросовера для кожної з хромосом поточної популяції;

– частота мутацій, обумовлена кількістю хромосом у поточній популяції, які змінюються;

– частота інверсій, обумовлена кількістю хромосом у поточній популяції, що піддаються циклічній перестановці генів;

– параметр зміни поколінь $G(k)$, що визначає частину поточної популяції $P(k)$, яка замінюється на кожній ітерації, при цьому $G(k)=1$ відповідає заміні всієї популяції в кожному поколінні;

– кількість особин у поточній популяції $Q(k)$;

– стратегія селекції.

Найбільш розповсюджені модифікації генетичних алгоритмів ґрунтуються, як правило, на керованому кросовері і спрямованій селекції, що імітують штучний добір.

Так в адаптивних генетичних алгоритмах [359, 371] ймовірність кросовера для кожної хромосоми пропорційна її пристосованості, при цьому схрещуванню піддаються тільки найкращі стринги. Таким чином, випадковий процес навчання поступово перетворюється в детермінований, а захистом від «застрягання» у локальних екстремумах слугують нечасті мутації та інверсії менш пристосованих особин.

До адаптивних генетичних алгоритмів досить близькі процедури із селекцією на основі елітизма [372], коли до розмноження допускаються тільки кращі особини в популяції. Відомі й інші форми репродукції, наприклад, що імітують еволюцію на ізольованих островах (Island models) [371] з рідким обміном генетичним матеріалом, що здійснюють розбивку простору параметрів і незалежний пошук у сформованих підпросторах [373] тощо.

І хоча, як відзначається в [374, 372], генетичні алгоритми перевершують за швидкістю випадковий пошук, навчання ШНМ у реальному часі на основі генетичних процедур наштовхується на істотні труднощі, обумовлені наявністю в кожен момент часу множини векторів-стрингів синаптичних ваг. У зв'язку з цим у [375] описано генетичний алгоритм, призначений для роботи в реальному часі, що має підвищену швидкість збіжності. У цьому алгоритмі використовується тільки один генетичний оператор – кросовер, причому всі особи популяції можуть схрещуватися тільки з однією хромосомою-«королевою» з мінімальним значенням $E_j\left({}_l w_j(k)\right)$. Природно, що на кожній ітерації k «королева» може змінюватися і саме її генетичний код використовується як поточний вектор синаптичних ваг.

Генетичні алгоритми в загальному випадку можуть оперувати не тільки з векторами-хромосомами, але і з більш складними об'єктами типу таблиць, списків і графів [359, 372]. Ця здатність дозволила успішно застосувати генетичні методи не тільки для навчання параметрів ШНМ, але й архітектур у цілому [376]. Дж. Коза запропонував підхід, що одержав назву генетичне програмування, в якому популяція утворюється не векторами-параметрами, а архітектурами нейронних мереж, представленими у формі потокових графів. У процесі мутації графів зі збереженням вхідних і вихідних розмірностей формується оптимальна архітектура мережі, щонайкраще пристосована для вирішення конкретної задачі. На жаль, обчислювальна громіздкість процедур генетичного програмування обмежує їхні можливості в процесі вирішення задач великої розмірності в реальному часі.

4.6 Алгоритми навчання на основі зворотного поширення помилок

Розглянуті вище алгоритми навчання призначені або для настроювання синаптичних ваг одиничного нейрона, або одношарової нейронной мережі. У задачі навчання багатошарових мереж виникають труднощі з настроюванням

ваг прихованих шарів, що можуть бути переборені за допомогою спеціальної процедури, яка отримала назву алгоритму зворотного поширення помилок, чи узагальненого дельта-правила.

Алгоритм зворотного поширення вперше було запропоновано П. Вербосом [377], але він залишався практично невідомим до його перевідкриття Д. Румельхартом, Дж. Гінтоном і Р. Вілліамсом [247]. Необхідно відзначити, що саме відсутність придатної процедури навчання не дозволила багатошаровим мережам набути широкого поширення і загальмувала розвиток цього напрямку на багато років.

Тут ми розглянемо використання концепції зворотного поширення щодо багатошарового персептрону в задачах, пов'язаних з нелінійним відображенням простору входів таких, як класифікація, діагностика, розпізнавання образів, адаптивне керування, ідентифікація тощо. При цьому вихідна інформація має бути задана у вигляді послідовності пар образів «вхід/вихід», що утворюють навчальну вибірку. Навчання полягає в адаптації параметрів усіх шарів таким чином, щоб розбіжність між вихідним сигналом мережі і зовнішнім навчальним сигналом у середньому була б мінімальною. З цього випливає, що алгоритм навчання є по суті процедурою пошуку екстремума спеціально сконструйованої цільової функції помилок.

Без втрати спільності розглянемо процес навчання тришарового персептрона з n_0 входами, n_1 нейронами в першому прихованому шарі, n_2 нейронами – у другому і n_3 нейронами у вихідному шарі. Кожен вхідний образ є $(n_0 \times 1)$-вектором $x=\left(x_1,\ldots,x_i,\ldots,x_{n_0}\right)^T$, вихідний образ – $(n_3 \times 1)$-вектор $y=\left(y_1,\ldots,y_i,\ldots,y_{n_3}\right)^T$ і навчальний образ – $(n_3 \times 1)$-вектор $d=\left(d_1,\ldots,d_j,\ldots,d_{n_3}\right)^T$. Необхідно в процесі навчання забезпечити мінімальну неузгодженість між поточними значеннями вихідних $y_j(k)$ і бажаних $d_j(k)$ сигналів для всіх $j=1,2,\ldots,n_3$ і k. Традиційно як функція помилок використовується або локальний критерій

$$E(k)=\frac{1}{2}\sum_{j=1}^{n_3}\left(d_j(k)-y_j(k)\right)^2=\frac{1}{2}\sum_{j=1}^{n_3}e_j^2(k)=\sum_{j=1}^{n_3}E_j(k), \qquad (4.322)$$

або глобальна цільова функція

$$E^k=\sum_k E(k)=\frac{1}{2}\sum_k\sum_j\left(d_j(k)-y_j(k)\right)^2=\frac{1}{2}\sum_k\sum_j e_j^2(k). \qquad (4.323)$$

Нині існує два основних підходи до пошуку мінімуму прийнятої цільової функції. Перший підхід пов'язаний в основному з цільовою функцією (4.322) і полягає в послідовному настроюванні ваг в міру надходження вхідних образів (часто у випадковому порядку) один за одним у реальному часі. При цьому для кожної пари образів x,d ваги $w_{ji}^{[s]}\ (s=1,2,3)$ змінюються на величину $\Delta w_{ji}^{[s]}$, пропорційну антиградієнту цільової функції $E(k)$, тобто

$$w_{ji}^{[s]}(k+1)-w_{ji}^{[s]}(k)=\Delta w_{ji}^{[s]}(k)=-\eta(k)\frac{\partial E(k)}{\partial w_{ji}^{[s]}}. \qquad (4.324)$$

Слід зазначити, що якщо кроковий коефіцієнт $\eta(k)$ досить малий, наприклад, задовольняє умовам Дворецького [276], то ця процедура мінімізує і глобальну цільову функцію (4.323).

Зазначимо також, що рекурентній процедурі настроювання (4.324) відповідає в неперервному часі диференціальне рівняння типу (4.123)

$$\frac{dw_{ji}^{[s]}}{dt} = -\eta \frac{\partial E(t)}{\partial w_{ji}^{[s]}}, \quad \eta > 0. \tag{4.325}$$

У другому підході, іменованому «пакетним навчанням», глобальна функція E^k мінімізується відразу по всій навчальній вибірці, яка заздалегідь має бути заданою.

Розглянемо спочатку алгоритм навчання реального часу, пов'язаний з мінімізацією на кожному кроці локальної функції $E(k)$. Очевидно, що для синаптичних ваг вихідного шару $w_{ji}^{[3]}$ справедливе співвідношення (4.132)

$$\Delta w_{ji}^{[3]}(k) = -\eta(k)\frac{\partial E(k)}{\partial w_{ji}^{[3]}(k)} = -\eta(k)\frac{\partial E(k)}{\partial u_j^{[3]}(k)} \cdot \frac{\partial u_j^{[3]}(k)}{\partial w_{ji}^{[3]}}. \tag{4.326}$$

Вводячи локальну помилку

$$\delta_j^{[3]}(k) = -\frac{\partial E(k)}{\partial u_j^{[3]}} = -\frac{\partial E(k)\partial e_j(k)}{\partial e_j^{[3]}(k)\partial u_j^{[3]}} = e_j(k)\frac{\partial \psi_j^{[3]}}{\partial u_j^{[3]}}, \tag{4.327}$$

з урахуванням того, що

$$u_j^{[3]}(k) = \sum_{i=1}^{n_2} w_{ji}^{[3]}(k) x_i^{[3]}(k) = \sum_{i=1}^{n_2} w_{ji}^{[3]}(k) o_i^{[3]}(k) \tag{4.328}$$

(тут $x_i^{[3]}(k) = o_i^{[3]}(k)$ означає, що входом третього шару є вихід другого), нескладно записати загальну формулу настроювання ваг вихідного шару у вигляді

$$\Delta w_{ji}^{[3]}(k) = \eta(k)\delta_j^{[3]}(k)x_i^{[3]}(k) = \eta(k)\delta_j^{[3]}(k)o_i^{[3]}(k), \tag{4.329}$$

де

$$\delta_j^{[3]}(k) = e_j(k)\left(\psi_j^{[3]}\left(u_j^{[3]}(k)\right)\right)' = \left(d_j(k) - y_j(k)\right)\frac{\partial \psi_j^{[3]}}{\partial u_j^{[3]}}. \tag{4.330}$$

Настроювання синаптичних ваг прихованих шарів набагато складніше. Для другого прихованого шару запишемо

$$\begin{aligned}\Delta w_{ji}^{[2]}(k) &= -\eta(k)\frac{\partial E(k)}{\partial w_{ji}^{[2]}} = -\eta(k)\frac{\partial E(k)}{\partial u_j^{[2]}(k)} \times \frac{\partial u_j^{[2]}(k)}{\partial w_{ji}^{[2]}} = \\ &= \eta(k)\delta_j^{[2]}(k)x_j^{[2]}(k) = \eta(k)\delta_j^{[2]}(k)o_i^{[1]}(k),\end{aligned} \tag{4.331}$$

де локальна помилка другого прихованого шару визначається виразом

$$\delta_j^{[2]}(k) = -\frac{\partial E(k)}{\partial u_j^{[2]}} \quad j = 1,2,\ldots,n_2. \tag{4.332}$$

Проблема полягає в тому, що цю помилку неможливо визначити безпосередньо за типом (4.330), у зв'язку з чим необхідно спробувати її виразити або через спостережувані сигнали, або через змінні, які можна оцінити.

Переписавши (4.332) у вигляді

$$\delta_j^{[2]}(k) = -\frac{\partial E(k)}{\partial u_j^{[2]}} = -\frac{\partial E(k)}{\partial o_j^{[2]}(k)} \cdot \frac{\partial o_j^{[2]}(k)}{\partial u_j^{[2]}}, \tag{4.333}$$

з урахуванням того, що

$$o_j^{[2]}(k) = \psi_j^{[2]}\left(u_j^{[2]}(k)\right), \tag{4.334}$$

отримуємо

$$\delta_j^{[2]}(k) = -\frac{\partial E(k)}{\partial o_j^{[2]}(k)} \cdot \frac{\partial \psi_j^{[2]}}{\partial u_j^{[2]}}. \tag{4.335}$$

Представивши $-\dfrac{\partial E(k)}{\partial o_j^{[2]}(k)}$ у вигляді

$$\begin{aligned} -\frac{\partial E(k)}{\partial o_j^{[2]}(k)} &= -\sum_{i=1}^{n_3} \frac{\partial E(k)}{\partial u_i^{[3]}(k)} \cdot \frac{\partial u_i^{[3]}(k)}{\partial o_j^{[2]}} = \\ &= \sum_{i=1}^{n_3} \left(-\frac{\partial E(k)}{\partial u_i^{[3]}(k)} \right) \cdot \frac{\partial}{\partial o_j^{[2]}} \left(\sum_{p=1}^{n_2} w_{ip}^{[3]}(k) x_p^{[3]}(k) \right) = \\ &= \sum_{i=1}^{n_3} \delta_i^{[3]}(k) \cdot \frac{\partial}{\partial o_j^{[2]}} \left(\sum_{p=1}^{n_2} w_{ip}^{[3]}(k) o_p^{[2]}(k) \right) = \sum_{i=1}^{n_3} \delta_i^{[3]}(k) w_{ij}^{[3]}(k), \end{aligned} \tag{4.336}$$

локальну помилку другого прихованого шару можна обчислити за допомогою виразу

$$\delta_j^{[2]}(k) = \frac{\partial \psi_j^{[2]}}{\partial u_j^{[2]}} \sum_{i=1}^{n_3} \delta_i^{[3]}(k) w_{ij}^{[3]}(k), \tag{4.337}$$

звідки випливає

$$\Delta w_{ji}^{[2]}(k) = \eta(k) o_i^{[1]}(k) \frac{\partial \psi_j^{[2]}}{\partial u_j^{[2]}} \sum_{i=1}^{n_3} \delta_i^{[3]}(k) w_{ij}^{[3]}(k). \tag{4.338}$$

Аналогічно можна записати формулу настроювання ваг першого шару

$$\Delta w_{ji}^{[1]}(k) = \eta(k) \delta_j^{[1]}(k) x_i^{[1]}(k) = \eta(k) \delta_j^{[1]}(k) o_i^{[0]}(k) = \eta(k) \delta_j^{[1]}(k) x_i(k), \tag{4.339}$$

де локальна помилка першого шару має вигляд

$$\delta_j^{[1]}(k) = \frac{\partial \psi_j^{[1]}}{\partial u_j^{[1]}} \sum_{i=1}^{n_2} \delta_i^{[2]}(k) w_{ij}^{[2]}(k). \tag{4.340}$$

Нескладно побачити, що локальна помилка внутрішнього (прихованого) шару визначається на основі помилок наступного шару. Починаючи з вихідного шару, за допомогою виразу (4.330) обчислюється локальна помилка $\delta_j^{[3]}(k)$, а потім шляхом її поширення від виходу до входу мережі обчислюються помилки $\delta_j^{[2]}(k)$ й $\delta_j^{[1]}(k)$. На рис 4.23 наведено схему навчання тришарового персептрона, за допомогою алгоритму зворотного поширення помилок.

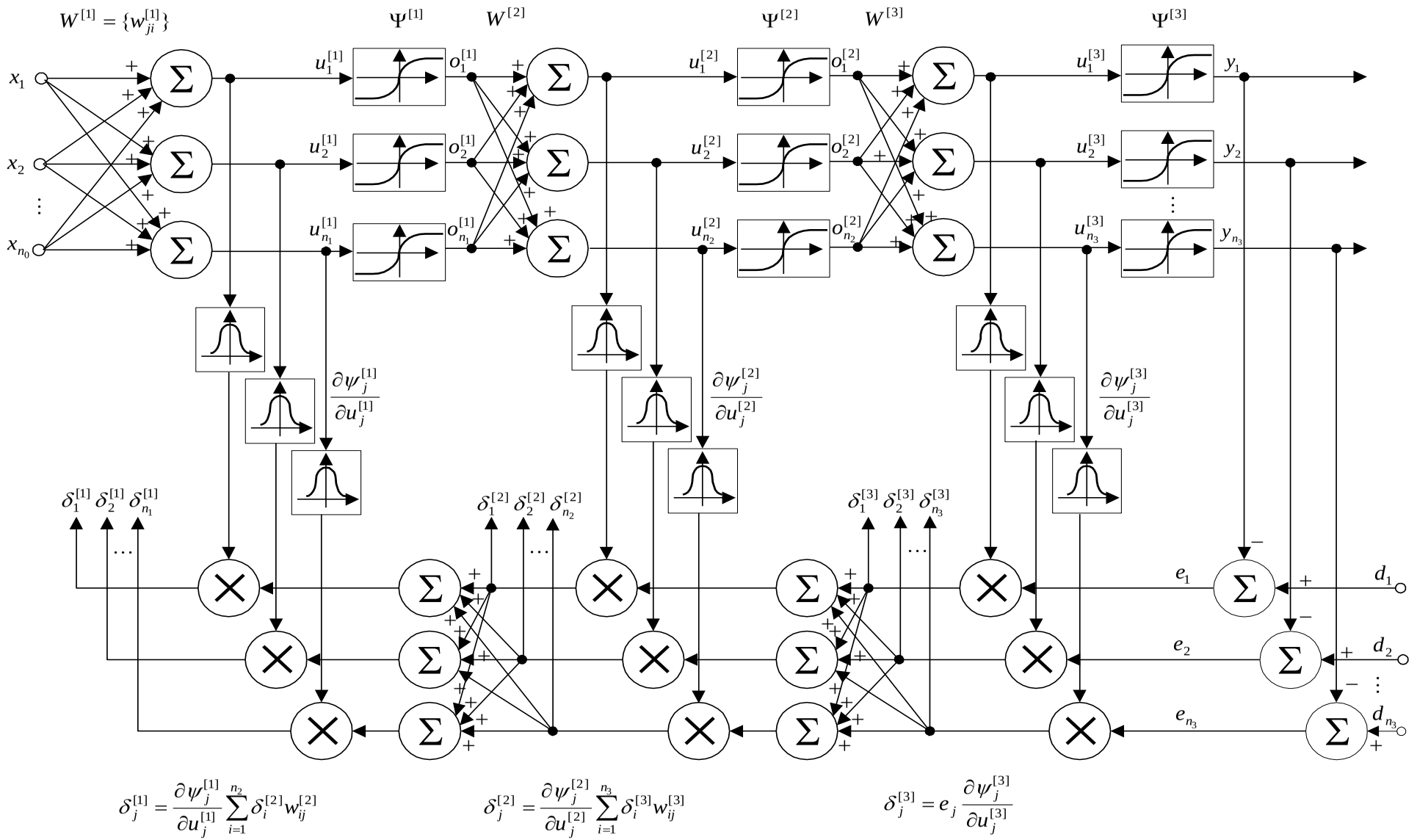


Рис. 4.23. Тришаровий персептрон, що навчається за допомогою зворотного поширення помилок

Головна відмінність розглянутого алгоритму навчання від процедур, описаних у підрозділі 4.4, полягає в розрахунку локальних помилок $\delta_j^{[s]}(k)$ $(s=1,2)$ прихованих шарів. Якщо у вихідному шарі локальна помилка – це функція бажаного й фактичного виходів мережі та похідної активаційної функції, то для прихованих шарів локальні помилки визначаються на основі локальних помилок наступних шарів.

Роботу алгоритму зворотного поширення помилок зручно описати у вигляді послідовності таких кроків:

– формування початкових умов для всіх синаптичних ваг мережі у вигляді досить малих випадкових чисел (зазвичай – $-0.5/n_{s-1} < w_{ji}^{[s]} < 0.5/n_{s-1}$) для того, щоб активаційні функції нейронів не увійшли в режим насичення на початкових стадіях навчання (захист від «паралічу» мережі);

– подача на вхід мережі образа x та обчислення виходів усіх нейронів за заданих значень $w_{ji}^{[s]}$;

– за заданим навчальним вектором d та обчисленими проміжними виходами $o_j^{[s]}$ розрахунок локальних помилок $\delta_j^{[s]}$ для всіх шарів;

– уточнення всіх синаптичних ваг за формулою

$$\Delta w_{ji}^{[s]} = \eta \delta_j^{[s]} x_i^{[s]}, \quad s=1,2,3; \tag{4.341}$$

– подача на вхід мережі наступного образа x тощо.

Процес навчання триває доти, поки помилка на виході ШНМ не стане досить малою, а ваги стабілізуються на деякому рівні. Після навчання нейронна мережа здобуває здатності до узагальнення, тобто починає правильно класифікувати образи, не представлені в навчальній вибірці. Це головна риса багатошарових персептронів, які можуть здійснювати після навчання довільне нелінійне відображення простору входів у простір виходів на основі апроксимації складних багатовимірних нелінійних функцій.

Властивості розглянутої вище процедури навчання істотно залежать від вибору параметра кроку $\eta(k)$. З одного боку, він має бути досить малим, щоб забезпечити оптимізацію глобальної цільової функції E^k. З іншого боку, малий кроковий коефіцієнт різко знижує швидкість збіжності, а, отже, збільшує час навчання.

Велике значення $\eta(k)$ збільшує швидкість збіжності, але може спровокувати нестійкість. Слід також пам’ятати, що якщо цільова функція має локальні мінімуми, процес навчання може в них «застрягти».

Поліпшити характеристики процесу навчання можна, відповідним чином модифікувавши процедури, описані в 4.4. Так, наприклад, вводячи регуляризуючий член, можна записати процедуру зворотного поширення, що є розвитком алгоритму Чана-Фоллсайда (4.166) [258]

$$\Delta w_{ji}^{[s]}(k) = \eta \delta_j^{[s]}(k) o_i^{[s-1]}(k) + \beta \Delta w_{ji}^{[s]}(k-1), \tag{4.342}$$

де $\eta > 0; \quad 0 \le \beta \le 1; \quad s=1,2,3.$

При прямуванні через область плато цільової функції, коли компоненти градієнта малі та практично не змінюються від кроку до кроку, (4.342) можна переписати у вигляді

$$\Delta w_{ji}^{[s]}(k) = -\eta \frac{\partial E(k)}{\partial w_{ji}^{[s]}} + \beta \Delta w_{ji}^{[s]}(k-1) \approx -\frac{\eta}{1-\beta} \frac{\partial E(k)}{\partial w_{ji}^{[s]}}, \qquad (4.343)$$

з якого випливає, що варіюванням параметра регуляризації β можна домогтися збільшення швидкості навчання.

На відміну від послідовної процедури навчання, коли синаптичні ваги мережі настроюються на кожному такті часу k, пакетне навчання спочатку накопичує всю навчальну вибірку, а потім водночас оброблює весь пакет наявних пар образів. При цьому для кожного k-го образа розраховується коригуюча добавка $\Delta w_{ji}^{[s]}(k)$ за формулами (4.329), (4.338) і (4.339) і тільки після пред'явлення останньої пари з навчальної вибірки відбувається корекція ваг згідно з формулою

$$\Delta \bar{w}_{ji}^{[s]} = \sum_k \Delta w_{ji}^{[s]}(k) = -\eta \frac{\partial E^k}{\partial w_{ji}^{[s]}} = \eta \sum_k \delta_j^{[s]}(k) o_i^{[s-1]}(k). \qquad (4.344)$$

Ця процедура може повторюватися кілька разів, а кожен такий «прохід» вибіркою називається епохою навчання. На практиці в режимі пакетного навчання використовуються більше складні процедури, ніж (4.344), а як один з тих алгоритмів, що добре зарекомендували себе, можна відзначити конструкцію типу [258]

$$\Delta \bar{w}_{ji}^{[s]}(p) = \frac{\eta}{n_{s-1}(1-\beta)} \sum_k \delta_j^{[s]}(k) o_i^{[s-1]}(k) + \beta \Delta \bar{w}_{ji}^{[s]}(p-1) - \alpha \bar{w}_{ji}^{[s]}(p), \qquad (4.345)$$

де $p = 1, 2, \ldots$ – номер епохи; $0 < \eta < 1$; β – параметр регуляризації; α (зазвичай $10^{-3} - 10^{-5}$) – параметр, що захищає процес навчання від виникнення неприпустимо великих значень ваг $w_{ji}^{[s]}$.

З метою захисту від застрягання в локальних мінімумах глобальної цільової функції в алгоритм навчання може бути додане випадкове збурення (за типом (4.269)), що відіграє роль зондувального сигналу.

Вибір послідовного або пакетного алгоритмів навчання визначається конкретною задачею, що розв'язується за допомогою багатошарової мережі. Послідовний підхід застосовується в тих випадках, коли навчальна вибірка не є доступною до початку процесу навчання й образи надходять послідовно в часі; у випадку більших об'ємів навчальної вибірки та високої розмірності образів-векторів пакетна обробка вимагає резервування більших об'ємів пам'яті, що робить кращим знов-таки послідовний підхід; крім того, послідовні процедури часто просто «швидші». На користь пакетної обробки говорить те, що вона має додаткові фільтруючі властивості й забезпечує більш високу точність оцінювання.

Як випливає з викладеного, навчання багатошарової ШНМ на основі зворотного поширення помилок – досить громіздка процедура, швидкість

збіжності якої істотно залежить від кількості нейронів у мережі. Кількість нейронів у шарах заздалегідь оцінити неможливо, тому на практиці зазвичай користуються емпіричними правилами типу: $n_1 = n_0, n_2 = 2n_0 + 1, n_3 = m$, хоча говорити про оптимальність у цьому випадку, природно, не доводиться. Більше конструктивним є підхід, за якого створюється вихідна ШНМ з надлишковою кількістю нейронів, що скорочується в процесі навчання від епохи до епохи. При цьому всі нейрони, які не вносять у вирішення задачі жодного внеску або не передають інформацію на наступний шар, вилучаються. Щоб з'ясувати, які приховані нейрони можуть бути вилучені, контролюються їхні вихідні сигнали після кожної епохи навчання при цьому, якщо сигнал близький до нуля або дорівнює виходу будь-якого іншого нейрона цього шару, цей нейрон може бути ліквідований. Після кожної епохи ця процедура може повторюватися.

Існує й інший підхід, коли кількість нейронів від епохи до епохи збільшується до досягнення необхідної точності. Крім того, збільшенням кількості нейронів мережу можна вивести з «паралічу» або локального мінімуму цільової функції. Якщо ця функція не зменшується або зменшується дуже повільно, у мережу додається нейрон з вагами, обраними випадковим чином. Якщо в процесі навчання після декількох ітерацій поліпшення не відбулося, додається ще один вузол тощо. Тут, щоправда, існує небезпека «перенавчання» (overfitting), за якого мережа втрачає свої узагальнюючі властивості й починає «відпрацьовувати» випадкові флуктуації.

Стандартний алгоритм зворотного поширення мінімізує цільову функцію (4.322), засновану на квадратах помилок навчання $e_j(k)$. У багатьох практичних застосуваннях з успіхом можуть бути використані інші конструкції типу (4.143), що приводять до критерію якості

$$E(k) = \sum_{j=1}^{n_3} f\left(e_j(k)\right), \tag{4.346}$$

де $f(\bullet)$ – деяка опукла функція.

Випадок $f\left(e_j(k)\right) = \frac{1}{2} e_j^2(k)$ дає стандартний квадратичний критерій, $f\left(e_j(k)\right) = \left|e_j(k)\right|$ – веде до критерію найменших модулів тощо. І хоча квадратичний критерій дозволяє отримати оптимальні оцінки у випадку гауссівського розподілу сигналів і збурювань, для розподілів з так званими «важкими хвостами» (Лапласа, Коші й ін.) оцінки, засновані на (4.322), (4.323), можуть виявитися незадовільними.

У цьому випадку ефективними можуть виявитися робастні методи оцінювання [258, 250, 378–381], засновані на цільових функціях, відмінних від квадратичного критерію таких, як

– логістична функція Велша

$$f_W\left(e_j(k)\right) = \varepsilon_1^2 \ln\left(\cosh\frac{e_j(k)}{\varepsilon_1}\right), \tag{4.347}$$

– функція Хубера

$$f_H\left(e_j(k)\right)=\begin{cases}\dfrac{e_j^2(k)}{2}, & \text{якщо } \left|e_j(k)\right|\le\varepsilon_1,\\ \varepsilon_1\left|e_j(k)\right|-\dfrac{\varepsilon_1^2}{2} & \text{в іншому випадку,}\end{cases} \tag{4.348}$$

– функція Талвара

$$f_T\left(e_j(k)\right)=\begin{cases}\dfrac{e_j^2(k)}{2}, & \text{якщо } \left|e_j(k)\right|\le\varepsilon_1,\\ \dfrac{\varepsilon_1^2}{2} & \text{в іншому випадку,}\end{cases} \tag{4.349}$$

– функція Гемпела

$$f_{Ha}\left(e_j(k)\right)=\begin{cases}\dfrac{\varepsilon_1^2}{\pi}\left(1-\cos\dfrac{\pi e_j(k)}{\varepsilon_1}\right), & \text{якщо } \left|e_j(k)\right|\le\varepsilon_1,\\ 2\dfrac{\varepsilon_1^2}{\pi} & \text{в іншому випадку}\end{cases} \tag{4.350}$$

та інші, наприклад, (4.80), (4.82). Тут $\varepsilon_1>0$ – керуючий параметр, що зазвичай обирається з емпіричних міркувань.

З метою підвищення швидкості навчання й поліпшення узагальнюючих властивостей мережі в [382] пропонується використовувати комбінації цих функцій, наприклад,

$$E(k)=(1-\alpha)\sum_{j=1}^{n_3} f_1\left(e_j(k)\right)+\alpha\sum_{j=1}^{n_3} f_2\left(e_j(k)\right), \tag{4.351}$$

де α – ваговий параметр, що змінюється в процесі навчання за правилом

$$\alpha=\alpha\left(E^k\right)=\exp\left(-\frac{\alpha_0}{\left(E^k\right)^2}\right),\ \alpha_0>0. \tag{4.352}$$

Найкращі результати були отримані в ході вибору

$$f_1\left(e_j(k)\right)=\varepsilon_1^2\ln\left(\cosh\frac{e_j(k)}{\varepsilon_1}\right) \tag{4.353}$$

з $0<\varepsilon_1<<1$ та

$$f_2\left(e_j(k)\right)=\frac{1}{2}e_j^2(k). \tag{4.354}$$

Зазначимо, що на початкових етапах навчання домінує функція $f_1\left(e_j(k)\right)$, яка за малих значень параметра ε_1 за властивостями наближається до критерію найменших модулів, тобто має виражені робастні властивості, будучи при цьому двічі диференційованою.

Для мінімізації локальної цільової функції (4.347) можна використовувати стандартну градієнтну техніку оптимізації. При цьому

$$\Delta w_{ji}^{[s]}(k) = -\eta(k)\frac{\partial E(k)}{\partial w_{ji}^{[s]}}, \tag{4.355}$$

або

$$\Delta w_{ji}^{[s]}(k) = -\eta(k)\frac{\partial E(k)}{\partial o_j^{[s]}(k)} \times \frac{\partial o_j^{[s]}(k)}{\partial u_j^{[s]}(k)} \times \frac{\partial u_j^{[s]}(k)}{\partial w_{ji}^{[s]}}, \tag{4.356}$$

звідки видно, що останні два співмножники визначаються тільки характеристиками нейронів і не залежать від виду прийнятого критерію якості $E(k)$. Це означає, що основна структура алгоритму зворотного поширення помилок зберігається, змінюючись лише в частині, пов'язаній з похідною цільової функції.

Записавши для вихідного шара очевидне співвідношення

$$\frac{\partial E(k)}{\partial o_j^{[3]}} = \frac{\partial E(k)}{\partial y_j} = \frac{\partial f\left(e_j(k)\right)}{\partial y_j}, \quad j = 1,2,\ldots,n_3 \tag{4.357}$$

і прийнявши як цільову функцію Велша (4.347) з

$$\frac{\partial E(k)}{\partial y_j} = -\varepsilon_1^2 \tanh\frac{e_j(k)}{\varepsilon_1}, \tag{4.358}$$

отримуємо

$$\delta_j^{[3]}(k) = -\frac{\partial E(k)}{\partial o_j^{[3]}(k)}\frac{\partial o_j^{[3]}(k)}{\partial u_j^{[3]}} = \varepsilon_1^2 \tanh\left(\frac{e_j(k)}{\varepsilon_1}\right) \times \frac{\partial \psi_j^{[3]}}{\partial u_j^{[3]}} \tag{4.359}$$

та

$$\Delta w_{ji}^{[3]}(k) = \eta(k)\delta_j^{[3]}(k)o_i^{[2]}(k). \tag{4.360}$$

Аналогічно без додаткових коментарів можна записати

$$\frac{\partial E(k)}{\partial o_j^{[2]}} = -\sum_{i=1}^{n_3}\delta_i^{[3]}(k)w_{ij}^{[3]}(k), \quad j = 1,2,\ldots,n_2, \tag{4.361}$$

$$\delta_j^{[2]}(k) = \frac{\partial \psi_j^{[2]}}{\partial u_j^{[2]}}\sum_{i=1}^{n_3}\delta_i^{[3]}(k)w_{ij}^{[3]}(k), \tag{4.362}$$

$$\Delta w_{ji}^{[2]}(k) = \eta(k)\delta_j^{[2]}(k)x_i^{[2]}(k) = \eta(k)\delta_j^{[2]}(k)o_i^{[1]}(k) \tag{4.363}$$

та

$$\frac{\partial E(k)}{\partial o_j^{[1]}} = -\sum_{i=1}^{n_2}\delta_i^{[2]}(k)w_{ij}^{[2]}(k), \quad j = 1,2,\ldots,n_1, \tag{4.364}$$

$$\delta_j^{[1]}(k) = \frac{\partial \psi_j^{[1]}}{\partial u_j^{[1]}}\sum_{i=1}^{n_2}\delta_i^{[2]}(k)w_{ij}^{[2]}(k), \tag{4.365}$$

$$\Delta w_{ji}^{[1]}(k) = \eta(k)\delta_j^{[1]}(k)x_i^{[1]}(k) = \eta(k)\delta_j^{[1]}(k)x_i(k). \tag{4.366}$$

Нескладно бачити, що вигляд цільової функції впливає тільки на локальну помилку вихідного шару $\delta_j^{[3]}(k)$, не змінюючи структури процедур настроювання прихованих шарів.

Для поліпшення апроксимуючих властивостей мережі в [258] пропонується поряд із синаптичними вагами налаштовувати й параметри крутості активаційних функцій за допомогою модифікованого алгоритму Крушке–Мовеллана [318]. При цьому для тришарового персептрона очевидні співвідношення:

$$o_j^{[s]}(k) = \psi_j^{[s]}\left(\gamma_j^{[s]}(k) u_j^{[s]}\right) = \tanh\left(\gamma_j^{[s]}(k) u_j^{[s]}(k)\right), \tag{4.367}$$

$$\Delta w_{ji}^{[s]}(k) = -\eta(k)\frac{\partial E(k)}{\partial w_{ji}^{[s]}} = \eta(k)\delta_j^{[s]}(k) o_i^{[s-1]}(k), \tag{4.368}$$

$$\Delta \gamma_j^{[s]}(k) = -\eta_\gamma(k)\frac{\partial E(k)}{\partial \gamma_j^{[s]}} = \frac{\eta_\gamma(k)\delta_j^{[s]}(k) u_j^{[s]}(k)}{\gamma_j^{[s]}(k)}, \tag{4.369}$$

$$\delta_j^{[s]}(k) = -\frac{\partial E(k)}{\partial u_j^{[s]}} = \frac{\partial \psi_j^{[s]}}{\partial u_j^{[s]}} \sum_{p=1}^{n_{s+1}} \delta_p^{[s+1]}(k) w_{pj}^{[s+1]}(k), \quad s = 1, 2, \tag{4.370}$$

$$\delta_j^{[3]}(k) = -\frac{\partial E(k)}{\partial u_j^{[3]}} = \frac{\partial \psi_j^{[3]}}{\partial u_j^{[3]}} e_j(k). \tag{4.371}$$

Наприкінці цього підрозділу слід зазначити, що для навчання багатошарових мереж за допомогою зворотного поширення, після відповідної модифікації, пов’язаної з обчисленням локальних помилок прихованих шарів, з успіхом можуть бути використані практично всі процедури, описані в підрозділах 4.4 і 4.5.

4.7 Алгоритми самонавчання

Даний підрозділ, на відміну від попередніх, присвячений процедурам навчання без вчителя (самонавчання), що є, за визначенням Б. Уїдроу [383], алгоритмами адаптації синаптичних ваг у розімкненому контурі. Найбільш широке поширення самонавчання отримало в задачах кластеризації, квантування неперервного простору входів, зниження розмірності простору сигналів (нелінійного факторного аналізу), виділення інформативних ознак під час розпізнавання образів тощо [248, 249]. Самонавчання полягає в основі таких широко розповсюджених ШНМ, як мапи Когонена, що самоорганізуються, мережі Гроссберга, мережі, засновані на теорії адаптивного резонансу, конкурентні мережі тощо. Тут ми розглянемо правила навчання окремих нейронів, а також області їхнього доцільного використання.

Правило навчання Гебба. Навчання за Д. Геббом є найбільш відомим, а його історія нараховує більше п’ятдесятьох років [240, 384, 235, 258, 385]. Суть його полягає в тому, що, якщо два сусідніх нейрони, пов’язаних через синаптичний зв’язок, активуються водночас, то сила цього зв’язку збільшується; якщо ж ці нейрони активуються асинхронно, то ця сила або зменшується, або виключається взагалі.

Формально для j-го нейрона мережі із $(n+1)$ входами правило Гебба виглядає в такий спосіб:

$$w_{ji}(k+1)=w_{ji}(k)+\eta(k)\,f_j\left(y_j(k)\right)\psi_i\left(x_i(k)\right), \tag{4.372}$$

де $f_j(\bullet)$ й $\psi_i(\bullet)$ – деякі функції, що обираються, як правило, з емпіричних міркувань.

У випадку, якщо як нейрон використовується лінійний асоціатор

$$y_j(k)=\sum_{i=0}^{n} w_{ji}x_i(k)=w_j^T x(k), \tag{4.373}$$

правило навчання набуває вигляду

$$w_{ji}(k+1)=w_{ji}(k)+\eta(k)\,y_j(k)x_i(k) \tag{4.374}$$

або

$$w_j(k+1)=w_j(k)+\eta(k)\,y_j(k)x(k). \tag{4.375}$$

І, зрештою, для мережі, утвореної m паралельно підключеними до входу нейронами (4.373), правило Гебба записується у формі

$$W(k+1)=W(k)+\eta(k)\,y(k)x^T(k), \tag{4.376}$$

де $W(k)=\left(w_1(k),\ldots,w_j(k),\ldots,w_m(k)\right)^T - m\times(n+1)$ – матриця синаптичних ваг. Нескладно бачити, що за нульової матриці $W(0)$

$$W(k+1)=\sum_{p=1}^{k+1}\eta(p)\,y(p)x^T(p), \tag{4.377}$$

звідки випливає пропорційність синаптичних ваг коефіцієнтам кореляції між вхідними й вихідними змінними.

Практичне використання алгоритмів (4.374) – (4.377) ускладнюється тією обставиною, що зі зростанням навчальної вибірки синаптичні ваги можуть збільшуватися необмежено, що в свою чергу, призводить до істотних обчислювальних труднощів. Обмежити значення коефіцієнтів можна, використовуючи алгоритм

$$W(k+1)=W(k)+\eta y(k)x^T(k)-\alpha W(k)=(1-a)W(k)+\eta y(k)x^T(k), \tag{4.378}$$

де $0<\alpha\le 1$ – фактор забування.

Можна показати [384], що граничне значення синаптичних ваг у цьому випадку визначається співвідношенням

$$w_{ji}^{\max}=\eta/\alpha. \tag{4.379}$$

Різновидами алгоритму (4.376) є [235] автоасоціативне правило навчання Гебба

$$W(k+1)=W(k)+\eta(k)\,x(k)x^T(k), \tag{4.380}$$

пов’язане з автокореляційними властивостями вхідних сигналів, і автоасоціативне правило Уїдроу–Гоффа

$$W(k+1)=W(k)+\eta(k)\left(x(k)-W(k)x(k)\right)x^T(k), \tag{4.381}$$

що мінімізує цільову функцію

$$E(k)=\frac{1}{2}\left\|x(k)-Wx(k)\right\|^2. \tag{4.382}$$

Нескладно бачити, що (4.381) збігається з алгоритмом навчання із вчителем Качмажа–Уїдроу–Гоффа, в якому, проте, замість зовнішнього навчального сигналу $d(k)$ використовується вхідний вектор $x(k)$.

Геббівське правило навчання може бути отримане й із суто формальних міркувань шляхом мінімізації критерію якості, званого в цьому випадку енергетичною функцією

$$E_j(t) = \frac{\alpha}{2}\|w_j\|^2 - \psi\left(w_j^T x(t)\right), \tag{4.383}$$

(тут $\alpha \geq 0$), що відрізняється від (4.151) відсутністю зовнішнього навчального сигналу $d_j(t)$. Процес мінімізації (4.383) має вигляд, близький до (4.155)

$$dw_{ji}/dt = \eta\left(-\alpha w_{ji} + \delta_j x_i\right) \tag{4.384}$$

і відрізняється від нього структурою локальної помилки δ_j , що описується в цьому випадку елементарним співвідношенням

$$\delta_j = y_j = \frac{d\psi\left(u_j\right)}{du_j}. \tag{4.385}$$

Тоді (4.384) з урахуванням (4.385) можна переписати у вигляді

$$dw_{ji}/dt = \eta\left(y_j x_i - \alpha w_{ji}\right) \tag{4.386}$$

для неперервного часу або

$$w_{ji}(k+1) = w_{ji}(k) + \eta(k)\left(y_j(k)x_i(k) - \alpha w_{ji}(k)\right), \tag{4.387}$$

$$w_j(k+1) = w_j(k) + \eta(k)\left(y_j(k)x(k) - \alpha w_j(k)\right), \tag{4.388}$$

$$W(k+1) = W(k) + \eta(k)\left(y(k)x^T(k) - \alpha W(k)\right) \tag{4.389}$$

для дискретного.

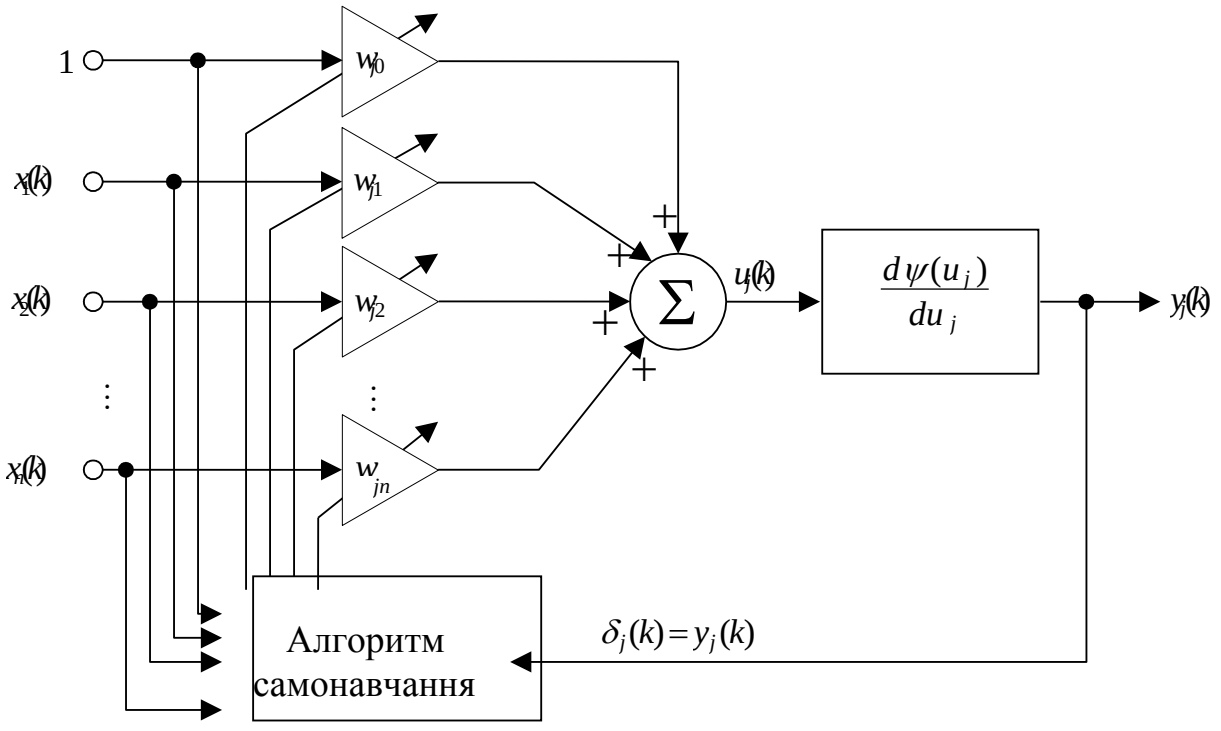


Рис. 4.24. Правило навчання Гебба

Вхідна зірка Ґроссберга. Вхідна зірка (Instar) С. Ґроссберга є нейроном за структурою аналогічним адаліні, призначеним для вирішення найпростіших задач розпізнавання образів і здійснюючим перетворення

$$y_j = \psi\left(w_j^T x + \theta_j\right), \tag{4.390}$$

де

$$\psi\left(u_j\right) = \begin{cases} 1, & \text{якщо } u_j \geq 0, \\ 0 & \text{в іншому випадку.} \end{cases} \tag{4.391}$$

Схему вхідної зірки наведено на рис. 4.25.

Нескладно бачити, що цей нейрон активується (на виході з'являється 1) у випадку, якщо вектор вхідних сигналів $x(k) \in$ у деякому сенсі близький до поточного вектора синаптичних ваг $w_j(k)$, тобто з виконанням умови

$$w_j^T(k)x(k) = \left\|w_j(k)\right\|\left\|x(k)\right\|\cos\theta \geq \theta_j, \tag{4.392}$$

де θ – кут між векторами $w_j(k)$ й $x(k)$; θ_j – сигнал зсуву, який задає поріг «близькості» векторів, що визначає спрацьовування вхідної зірки.

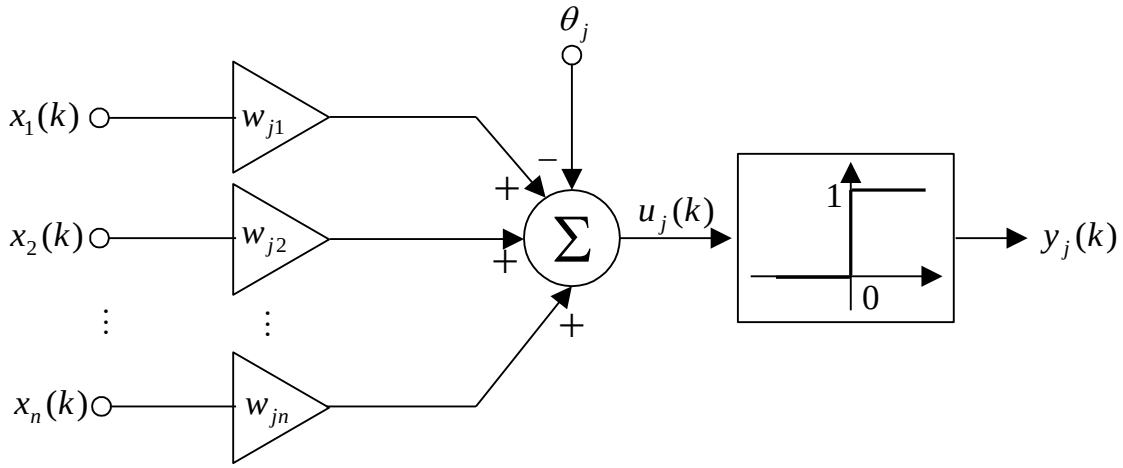


Рис. 4.25. Вхідна зірка

Якщо прийняти

$$\theta_j = \left\|w_j\right\|\left\|x\right\|, \tag{4.393}$$

то зірка активізується тільки у випадку, якщо вхідний сигнал збігається з вектором синаптичних ваг, тобто розпізнається тільки один образ. Чим менше значення θ_j, тим більше можливих образів можуть активізувати нейрон, що стає при цьому усе менш «розбірливим».

Навчання вхідної зірки відбувається за допомогою модифікованого алгоритма (4.378), що приймає в цьому випадку вид

$$w_j(k+1) = w_j(k) + \eta y_j(k)x(k) - \alpha y_j(k)w_j(k). \tag{4.394}$$

Необхідність модифікації пов'язана з тим, що у випадку подачі на вхід нейрона послідовності $x(k)$, що не активізує зірку $\left(y_j(k) = 0\right)$ відбувається поступове забування всієї накопиченої інформації. Справді, в цьому випадку алгоритм (4.378) набуває вигляд

$$w_j(k+1) = (1-\alpha)w_j(k). \tag{4.395}$$

Відмінною ж особливістю правила (4.394) є те, що самонавчання відбувається тільки в активованому стані, коли $y_j(k)=1$.

Покладаючи для простоти $\alpha=\eta$, отримуємо так зване стандартне правило самонавчання вхідної зірки

$$w_j(k+1)=w_j(k)+\eta y_j(k)\big(x(k)-w_j(k)\big), \qquad (4.396)$$

яке можна проілюструвати за допомогою рис. 4.26.

При $y_j(k)=0$ згідно з (4.396) навчання не відбувається, тобто $w_j(k+1)=w_j(k)$. При $y_j(k)=1$ алгоритм набуває вигляду

$$w_j(k+1)=w_j(k)+\eta\big(x(k)-w_j(k)\big)=(1-\eta)w_j(k)+\eta x(k), \qquad (4.397)$$

тобто вектор синоптичних ваг «підтягується» до вхідного образа на відстань, пропорційну параметру кроку η. Чим більше η, тим ближче $w_j(k+1)$ до $x(k)$ і при $\eta=1$ збігається з ним. Зазвичай в реальних задачах використовується змінне значення $\eta(k)$, що відповідає умовам Дворецького [276]. Можна також відзначити, що з метою обчислювальних зручностей, замість вектора $x(k)$ частіше використовують його нормований аналог

$$\tilde{x}(k)=\frac{x(k)}{\|x(k)\|}. \qquad (4.398)$$

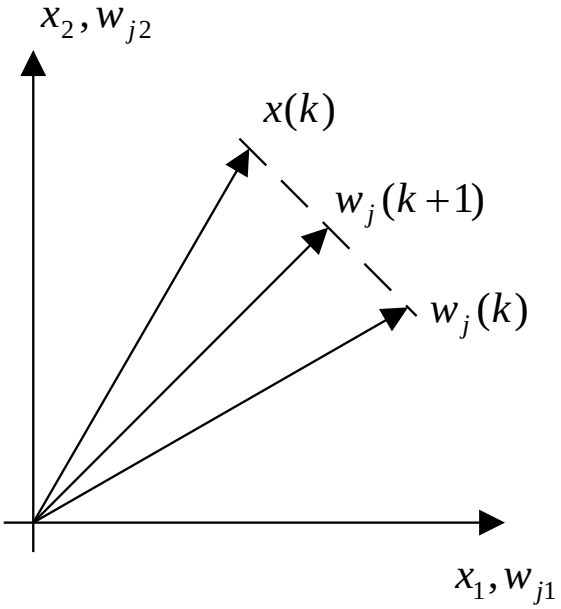


Рис. 4.26. Навчання вхідної зірки

Вихідна зірка. Своєрідним антиподом вхідної зірки є вихідна зірка (Outstar), призначена для вирішення задач відновлення образів, схему якої наведено на рис. 4.27.

Цей нейрон має скалярний вхід і векторний вихід і здійснює перетворення

$$y_j=\psi\big(w_{j1}x\big), \quad j=1,2,\ldots,m \qquad (4.399)$$

з функцією активації

$$\psi(u_j)=\begin{cases} u_j, & \text{якщо } -1\le u_j\le 1,\\ 1, & \text{якщо } 1<u_j,\\ -1, & \text{якщо } u_j<-1. \end{cases} \tag{4.400}$$

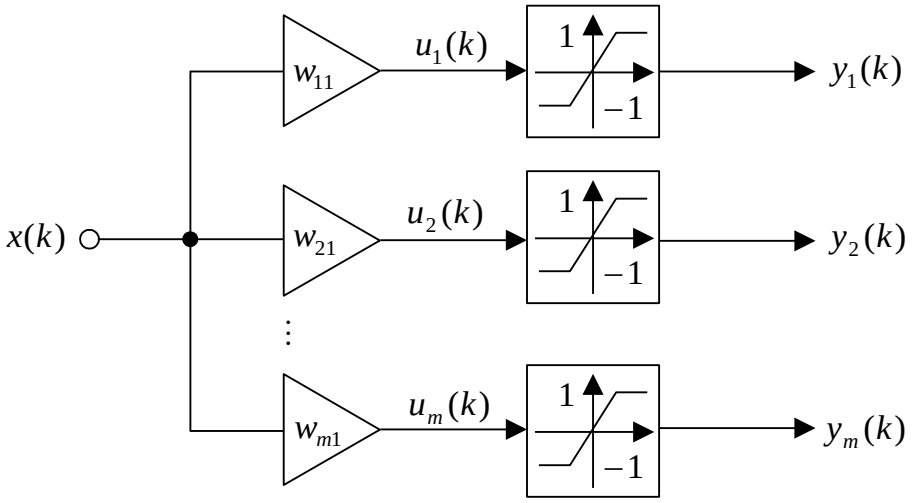


Рис. 4.27. Вихідна зірка

Правило самонавчання вихідної зірки має (4.401),

$$w_{ji}(k+1)=w_{ji}(k)+\eta\, y_j(k)x_i(k)-\alpha x_i(k)w_{ji}(k), \tag{4.401}$$

а при $\eta=\alpha$ –

$$w_{ji}(k+1)=w_{ji}(k)+\eta x_i(k)\big(y_j(k)-w_{ji}(k)\big), \tag{4.402}$$

таким чином настроювання синаптичних ваг відбувається тільки у випадку $x_i(k)\neq 0$. Тут, як видно, у процесі самонавчання синаптичні ваги «підтягуються» до вихідного вектора $y(k)$.

У векторній формі правило самонавчання має вигляд

$$w_i(k+1)=w_i(k)+\eta x_i(k)\big(y(k)-w_i(k)\big), \tag{4.403}$$

де $w_i(k)$ - i-й стовпчик матриці коефіцієнтів $W(k+1)$ (4.377).

Потенційне правило навчання Амарі. Потенційне правило самонавчання С. Амарі [386] за структурою досить близьке до геббівського навчання й пов’язане з мінімізацією енергетичної функції виду (4.383), заснованої на так званому внутрішньому потенціалі, що визначає рівень активності нейрона.

Для енергетичної функції

$$E_j(k)=\frac{\alpha}{2}\|w_j\|^2-\psi\left(w_j^T x(t)\right) \tag{4.404}$$

як рівень активності використовується конструкція

$$\psi(u_j)=\frac{1}{2}u_j^2(t), \tag{4.405}$$

що веде до алгоритму мінімізації в неперервному часі

$$\frac{dw_{ji}}{dt} = \eta\left(-\alpha w_{ji} + \delta_j x_i\right), \tag{4.406}$$

де

$$\delta_j = \frac{d\psi\left(u_j\right)}{du_j} = u_j\,. \tag{4.407}$$

Тоді нескладно записати правило навчання

$$\frac{dw_{ji}}{dt} = \eta\left(u_j x_i - \alpha w_{ji}\right) \tag{4.408}$$

у неперервному часі й

$$w_{ji}\left(k+1\right) = w_{ji}\left(k\right) + \eta\left(k\right)\left(u_j\left(k\right)x_i\left(k\right) - \alpha w_{ji}\left(k\right)\right), \tag{4.409}$$

$$w_j\left(k+1\right) = w_j\left(k\right) + \eta\left(k\right)\left(u_j\left(k\right)x\left(k\right) - \alpha w_j\left(k\right)\right) \tag{4.410}$$

у дискретному.

На рис. 4.28 наведено схему самонавчання за допомогою правила Амарі.

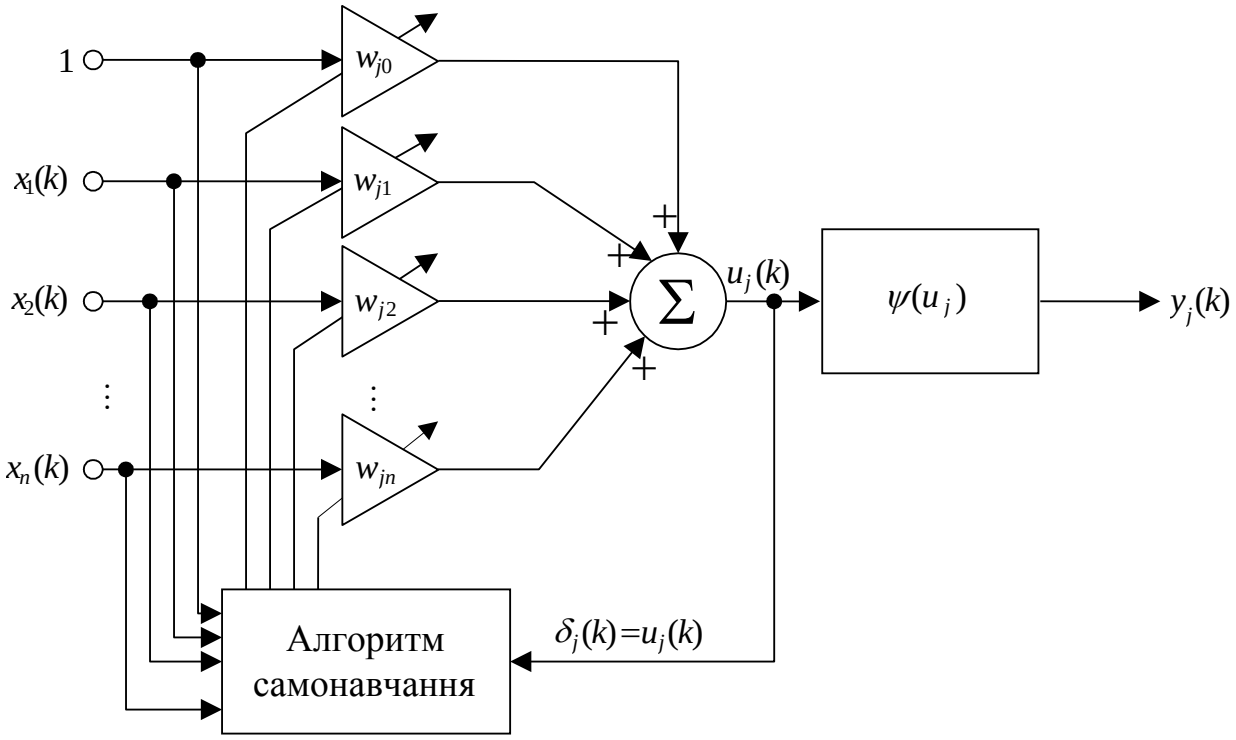


Рис. 4.28. Правило навчання Амарі

Правило навчання Оя. Правило навчання Е. Оя [387, 388] тісно пов’язане з алгоритмами Гебба й Амарі та породжується мінімізацією енергетичної функції

$$E_j\left(t\right) = \frac{1}{2}\left\|e_j\left(t\right)\right\|^2, \tag{4.411}$$

де

$$e_j\left(t\right) = x\left(t\right) - \hat{x}_j\left(t\right) \tag{4.412}$$

- помилка між вхідним сигналом $x\left(t\right)$ і його оцінкою $\hat{x}_j\left(t\right)$.

Це правило призначене для навчання лінійного асоціатора

$$y_j = u_j = w_j^T x, \tag{4.413}$$

при цьому шукана оцінка $\hat{x}_j$ визначається виразом

$$\hat{x}_j = w_j y_j. \tag{4.414}$$

Перетворюючи (4.411) з урахуванням (4.414), отримуємо

$$E_j(t) = \frac{1}{2}\left\| x - w_j y_j \right\|^2 = \frac{1}{2}\left(x^T x - 2 w_j^T x y_j + w_j^T w_j y_j^2 \right), \tag{4.415}$$

звідки випливає

$$\nabla_{w_j} E_j(t) = -x y_j + w_j y_j^2 \tag{4.416}$$

та

$$\frac{dw_{ji}}{dt} = \eta\left(y_j x_i - w_{ji} y_j^2 \right) = \eta y_j \left(x_i - w_{ji} y_j \right). \tag{4.417}$$

У дискретному часі алгоритм Оя має вигляд

$$w_{ji}(k+1) = w_{ji}(k) + \eta(k) y_j(k)\left(x_i(k) - w_{ji}(k) y_j(k) \right), \tag{4.418}$$

$$w_j(k+1) = w_j(k) + \eta(k) y_j(k)\left(x(k) - w_j(k) y_j(k) \right). \tag{4.419}$$

Найбільше поширення це правило отримало для вирішення задач факторного аналізу [389, 390], коли з масивів емпіричних даних у реальному часі потрібно виділяти головні компоненти.

Для знаходження першого головного компонента Оя запропонував структуру, наведену на рис. 4.29, в основі якої лежить адаптивний лінійний асоціатор і алгоритм самонавчання (4.419).

Для попередньо центрованих даних

$$\tilde{x}_i(k) = x_i(k) - \overline{x}_i(k), \quad \overline{x}_i(k) = \frac{1}{k}\sum_{p=1}^{k} x(p), \quad x_0(k) \equiv 0 \tag{4.420}$$

алгоритм виділення першого компонента має вигляд

$$\begin{cases} w_1(k+1) = w_1(k) + \eta(k) y_1(k)\left(\tilde{x}(k) - w_1(k) y_1(k) \right), \\ y_1(k) = w_1^T(k)\tilde{x}(k), \quad w_1(0) \neq 0, \end{cases} \tag{4.421}$$

та забезпечує мінімум критерію

$$E_1^k = \frac{1}{k}\sum_{p=1}^{k}\left(w_1^T \tilde{x}(p) \right)^2. \tag{4.422}$$

В [391] було доведено збіжність алгоритму (4.421) у припущенні, що крок пошуку $\eta(k)$ обирається згідно з умовами Дворецького. Зокрема, доцільно обирати цей параметр відповідно до виразу [392, 393]

$$\eta(k) = r^{-1}(k), \quad r(k) = \alpha r(k-1) + \left\| \tilde{x}(k) \right\|^2, \quad 0 \le \alpha \le 1, \tag{4.423}$$

де α – параметр забування, що забезпечує компроміс між слідкуючими та фільтруючими властивостями алгоритму (4.421).

Можна показати також, що правило Оя забезпечує нормування вектора синаптичних ваг $\left\| w_1(k) \right\| = 1$, вектор $w_1(k)$ є власним вектором кореляційної

матриці входів, а вихідний сигнал нейрона $y_1(k)$ характеризується максимально можливою дисперсією, тобто містить максимум інформації про вхідний сигнал $\tilde{x}(k)$.

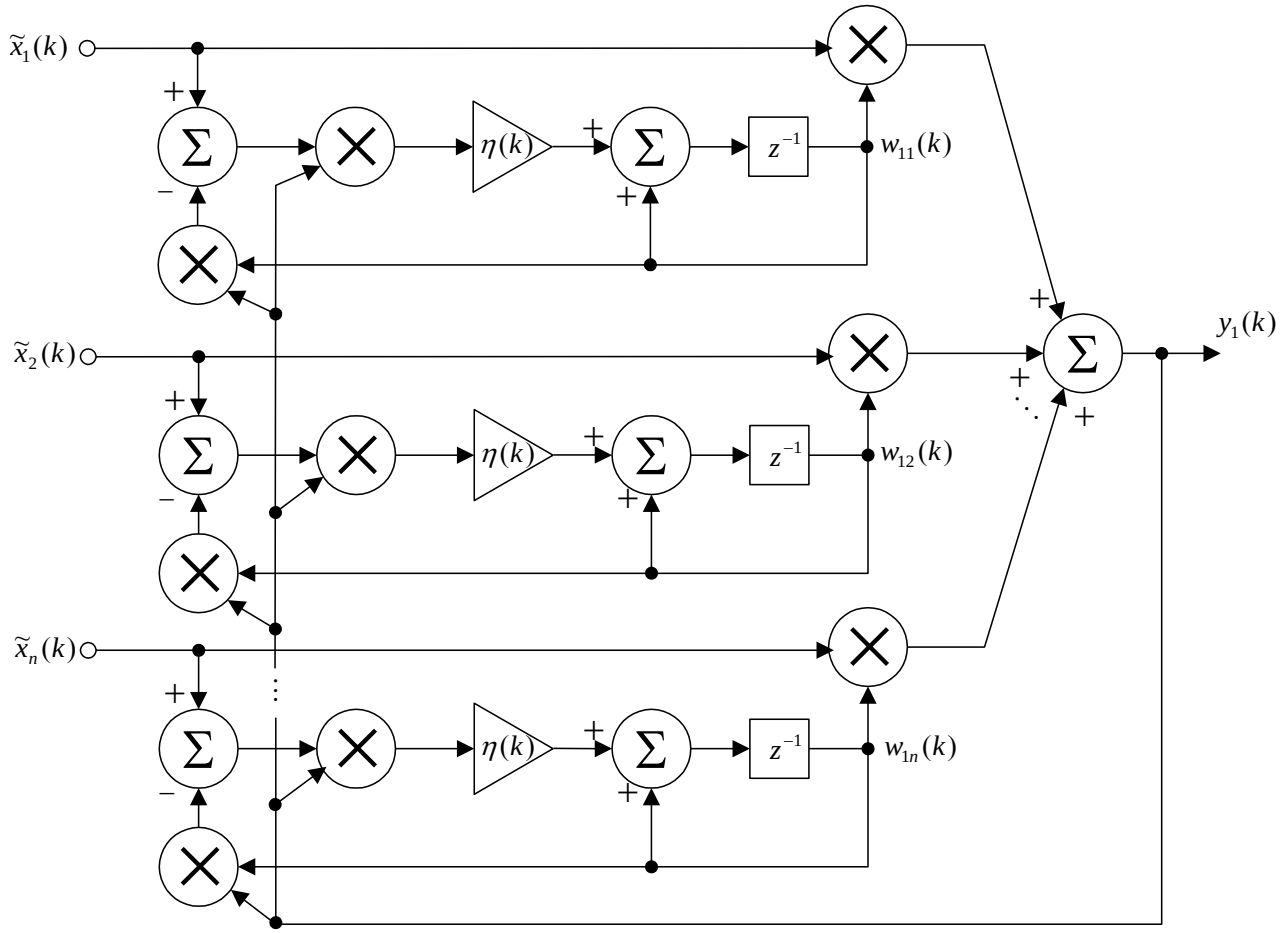


Рис. 4.29. Нейрон Оя

Робота алгоритму (4.421) ілюструється рис. 4.30.

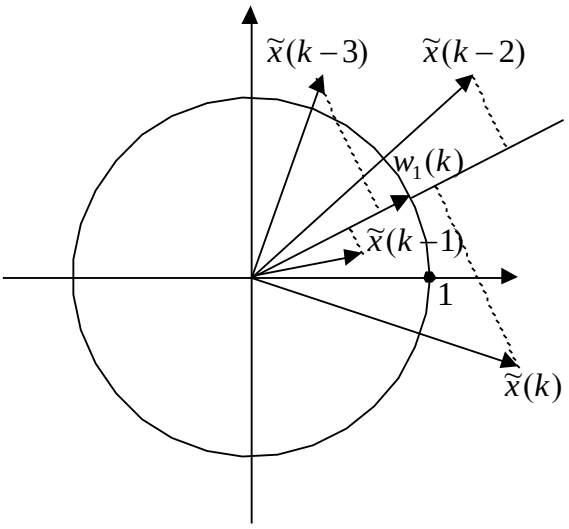


Рис. 4.30. Множина векторів і перший головний компонент

Нейрони для попередньої обробки інформації. У задачах обробки експериментальних даних досить часто доводиться обчислювати основні вибіркові статистичні характеристики такі, як середнє, дисперсію, екстремальні значення у вибірках. Для вирішення цих задач доцільно ввести елементарні нейроноподібні структури, що дозволяють обчислювати шукані характеристики в реальному часі без накопичення даних, що надходять послідовно.

Так обчислення середнього

$$\bar{x}\left(k+1\right)=\frac{1}{k+1}\sum_{p=1}^{k+1}x\left(p\right) \tag{4.424}$$

можна здійснювати за допомогою рекурентного співвідношення

$$\bar{x}(k+1)=\bar{x}(k)+\frac{1}{k+1}(x(k+1)-\bar{x}(k)), \tag{4.425}$$

схемну реалізацію якого наведено на рис. 4.31.

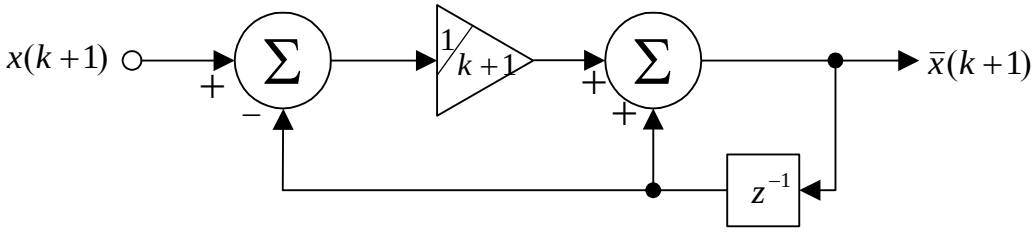


Рис. 4.31. Нейрон для обчислення середнього

Для розрахунку вибіркової дисперсії

$$\sigma_x^2\left(k+1\right)=\frac{1}{k+1}\sum_{p=1}^{k+1}\left(x\left(p\right)-\bar{x}\left(k+1\right)\right)^2 \tag{4.426}$$

можна скористатися формулою

$$\sigma_x^2\left(k+1\right)=\sigma_x^2\left(k\right)+\frac{1}{k+1}\left(\left(x\left(k+1\right)-\bar{x}\left(k+1\right)\right)^2-\sigma_x^2\left(k\right)\right), \tag{4.427}$$

реалізованою схемою, наведеною на рис. 4.32.

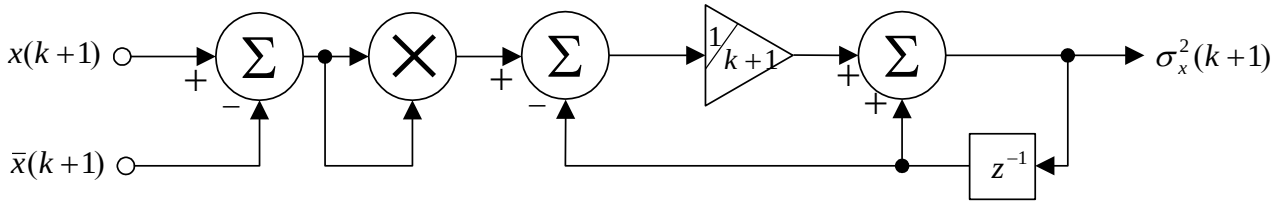


Рис. 4.32. Нейрон для обчислення дисперсії

Порівняння двох чисел x_1 і x_2 можна проводити, використовуючи вираз

$$\begin{cases}\max\left\{x_1,x_2\right\}=x_1-0.5\left(1-sign\left(x_1-x_2\right)\right)\left(x_1-x\right)_2,\\ \min\left\{x_1,x_2\right\}=x_1-0.5\left(1+sign\left(x_1-x_2\right)\right)\left(x_1-x_2\right)\end{cases} \tag{4.428}$$

і нейроноподібну структуру, показану на рис. 4.33.

В ході послідовної обробки часових рядів досить часто доводиться визначати їхні екстремальні значення, для чого можна скористатися схемою, наведеною на рис. 4.34, що є модифікацією (4.428) та має вигляд

$$\begin{cases} x^*(k+1) = \max\{x(k+1), x^*(k)\} = \\ = x(k+1) - 0.5\big(1 - sign\big(x(k+1) - x^*(k)\big)\big)\big(x(k+1) - x^*(k)\big), \\ x_*(k+1) = \min\{x(k+1), x_*(k)\} = \\ = x(k+1) - 0.5\big(1 + sign\big(x(k+1) - x_*(k)\big)\big)\big(x(k+1) - x_*(k)\big) \end{cases} \quad (4.429)$$

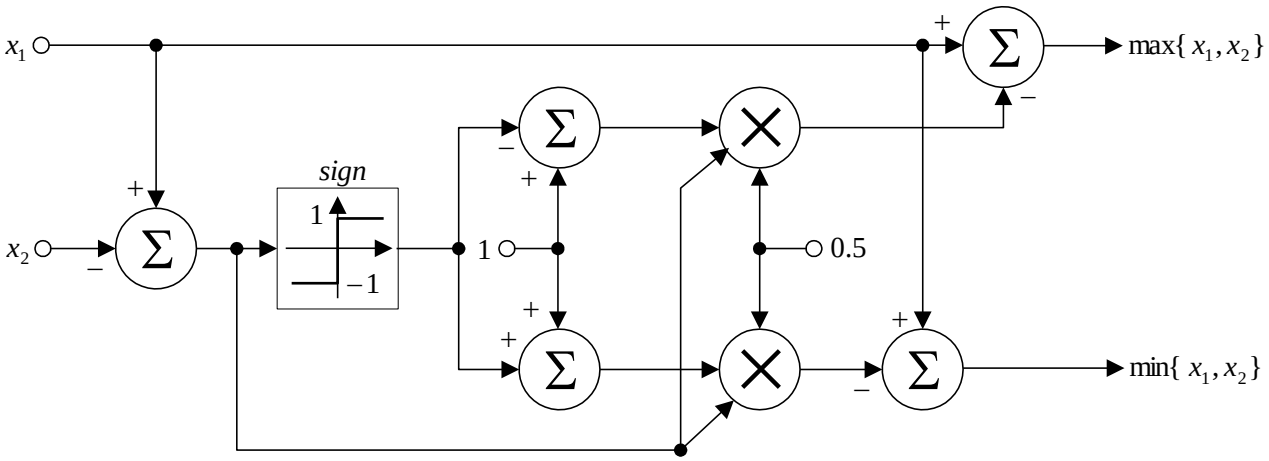


Рис. 4.33. Нейрон для порівняння двох чисел

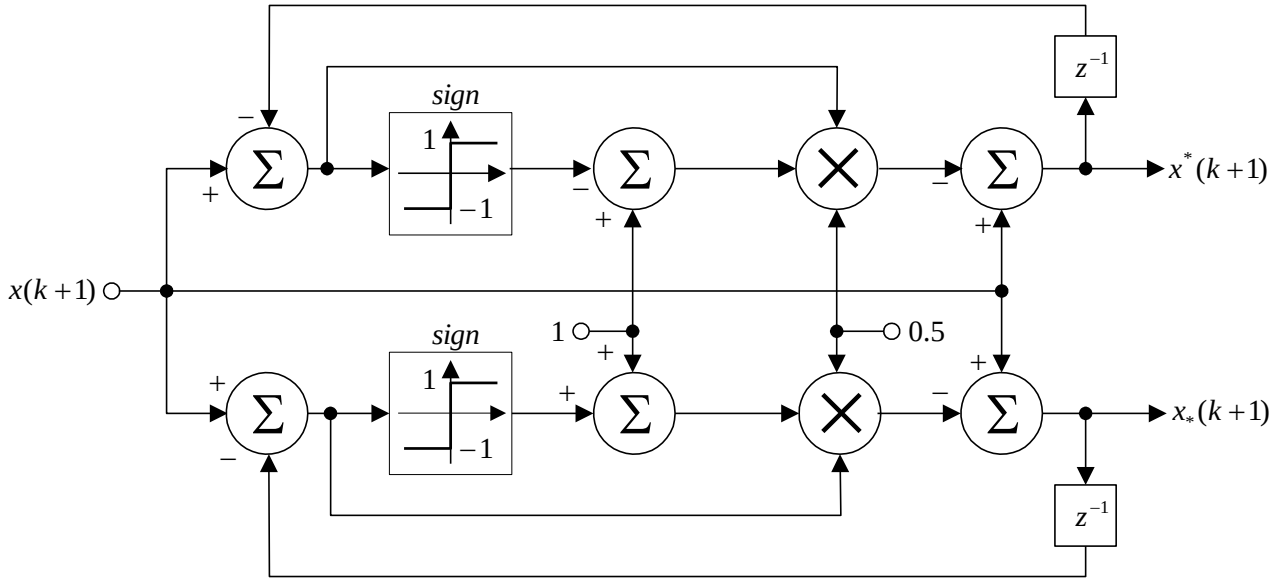


Рис. 4.34. Нейрон для виділення екстремальних значень часового ряду

Самоорганізація центрів радіально-базисних нейронних мереж. У підрозділі 4.4 було розглянуто алгоритм навчання із вчителем (4.230), що дозволяє настроювати всі параметри радіально-базисних ШНМ. На практиці зазвичай використовуються лише алгоритми настроювання синаптичних ваг,

що лінійно входять в оператор перетворення, здійснюваного мережею, що дозволяє застосовувати оптимальні за швидкодією рекурентні алгоритми адаптивної ідентифікації.

Для настроювання параметрів центрів може використовуватися процедура їхньої самоорганізації, в основі якої лежать алгоритми кластеризації, що розбивають всю множину вхідних даних на групи, однорідні в деякому сенсі. Найбільш широке поширення отримала процедура кластеризації, відома як «алгоритм k-середніх» [240, 258], ідея якої полягає в тому, щоб розміщувати центри в місцях, де вхідні дані сконцентровані найбільш щільно, тобто утворюють свого роду кластери. Алгоритм знаходить множину центрів цих кластерів, зв'язуючи кожен центр із одним з h вузлів мережі. В процесі самоорганізації дані розбиваються таким чином, щоб всі точки навчальної вибірки належали кластеру з найближчим поточним центром.

Початкові положення центрів $c_i(0), i=1,2,\ldots,h$ задаються випадковим чином, а їхнє перенастроювання-самоорганізація відбувається в процесі надходження навчальної вибірки. Робота алгоритму k-середніх при цьому відбувається в такий спосіб:

– нехай у k-й момент часу є h центрів $c_i(k)$ і вхідний вектор $x(k)$;

– обчислюються відстані між кожним центром і вхідним вектором

$$D_i(k)=\|x(k)-c_i(k)\|, \quad i=1,2,\ldots,h; \tag{4.430}$$

– відшукується центр $c_l(k)$ найближчого до поточного вектора $x(k)$ такий, що

$$D_l(k)=\min_{i=1,2,\ldots,h}\left\{\|x(k)-c_1(k)\|,\ldots,\|x(k)-c_h(k)\|\right\}, \tag{4.431}$$

– відбувається настроювання центрів відповідно до правила

$$\begin{cases} c_i(k+1)=c_i(k), \quad 1\le i\le h, \quad i\ne l, \\ c_l(k+1)=c_l(k)+\eta_c(k)\left(x(k)-c_l(k)\right), \end{cases} \tag{4.432}$$

– відбувається настроювання параметра кроку, наприклад, за допомогою співвідношення [394]

$$\eta_c(k+1)=\frac{\eta_c(k)}{\left(1+\operatorname{int}\left(\dfrac{k+1}{h}\right)\right)^{\frac{1}{2}}}, \tag{4.433}$$

де $\operatorname{int}(\bullet)$ – ціла частина числа.

Далі відбувається настроювання синаптичних ваг усіх вузлів мережі за допомогою будь-якого з алгоритмів, розглянутих у підрозділі 4.3. Саме сполучення самоорганізації центрів і навчання із вчителем синаптичниих ваг забезпечує високу ефективність радіально-базисних нейронних мереж [240, 258, 394, 385].

Конкурентне навчання. Особливим видом самонавчання є так зване конкурентне навчання, коли всі нейрони мережі «змагаються» між собою за

право бути активним, реалізуючи принцип «переможець отримує все» (winner takes all), що веде до того, що в мережі може активізуватися тільки один нейрон. Саме ця особливість конкурентного навчання забезпечила йому широке використання в задачах класифікації й кластеризації.

На рис 4.35 наведено найпростішу нейронну мережу, що використовує конкурентне навчання і має лише один шар нейронів.

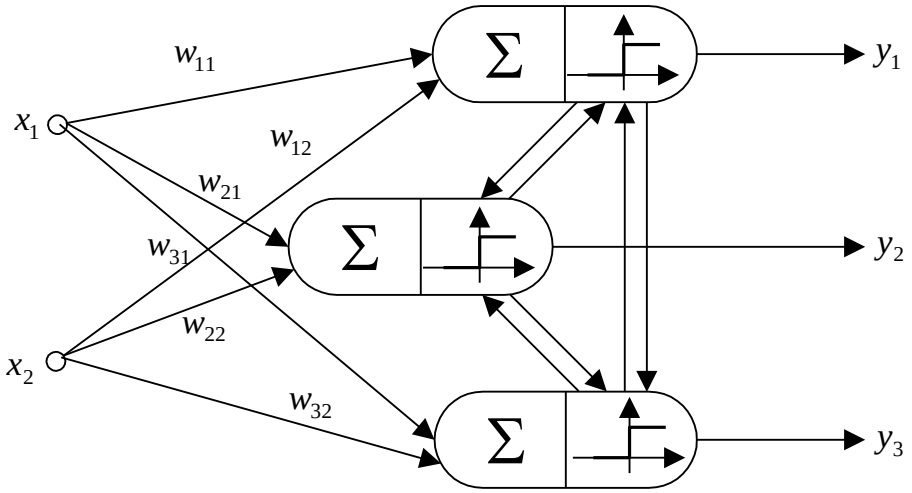


Рис. 4.35. Найпростіша конкурентна мережа

Особливістю цієї мережі є те, що крім прямих зв'язків, які передають інформацію із входу на вихід через синаптичні ваги w_{ji}, у ній присутні поперечні (латеральні) зв'язки, якими і відбувається змагання, внаслідок якого «перемагає» тільки один нейрон.

Формально процес конкуренції може бути представлений у формі

$$y_j(k)=\begin{cases}1, & \text{якщо } w_j^T(k)\tilde{x}(k) > w_p^T(k)\tilde{x}(k) \text{ для всіх } p\neq j,\\ 0 & \text{в іншому випадку,}\end{cases} \tag{4.434}$$

де

$$\tilde{x}(k)=\frac{x(k)}{\|x(k)\|}, \quad \|w_j(k)\|=1, \tag{4.435}$$

при цьому в кожен момент часу k настроюється тільки нейрон-переможець за допомогою алгоритма, відомого як правило навчання Т. Когонена [395]:

$$w_j(k+1)=\begin{cases}w_j(k)+\eta(k)\left(\tilde{x}(k)-w_j(k)\right), & \text{якщо } j-\text{й нейрон переміг,}\\ w_j(k) & \text{у іншому випадку.}\end{cases} \tag{4.436}$$

Алгоритм (4.436) досить близький до правила навчання вхідної зірки й у процесі настроювання також «підтягує» вектор синаптичних ваг нейрона-переможця $w_j(k)$ до поточного вхідного образа $x(k)$.

Рис. 4.36 ілюструє вирішення задачі кластеризації за допомогою найпростішої конкурентної ШНМ, наведеної на рис. 4.35, що настроюється за допомогою алгоритма (4.436).

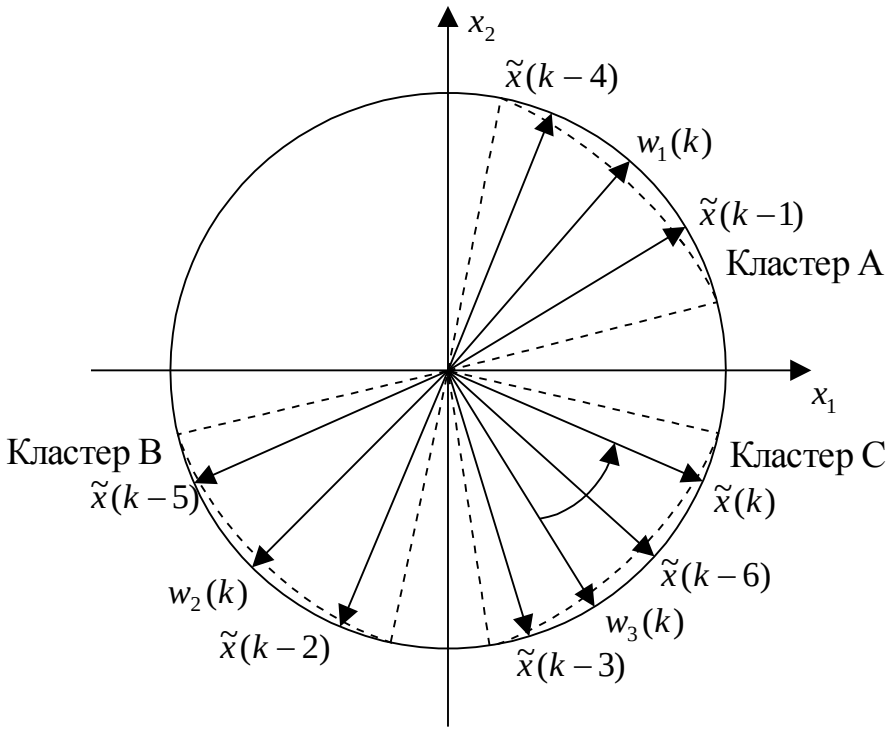


Рис. 4.36. Кластеризація на основі конкурентного навчання

Видно, що поточний нормований вектор $\tilde{x}(k)$ ближче всього до вектора синаптичних ваг $w_3(k)$, внаслідок чого «перемагає» третій нейрон

$$\begin{cases} w_3^T(k)x(k) > w_1^T(k)x(k), \\ w_3^T(k)x(k) > w_2^T(k)x(k), \end{cases} \tag{4.437}$$

який і настроює свої параметри за допомогою правила Когонена, «підтягуючи» $w_3(k)$ до $x(k)$ на відстань, пропорційну параметру кроку $\eta(k)$.

Конкурентне навчання полягає в основі низки нейромереж, які отримали досить широке поширення в задачах обробки інформації та інтелектуального аналізу даних.

5 ІНТЕЛЕКТУАЛЬНИЙ АНАЛІЗ ДАНИХ У ХОДІ ТЕСТУВАННЯ НОСОВОГО ДИХАННЯ

5.1 Основні особливості інтелектуального аналізу риноманометричних даних

Підвищення точності технічних засобів вимірювального контролю і функціональної діагностики практично не знижує невизначеність результатів вимірювань за непереборної невизначеності властивостей складних, дифузних об'єктів контролю, якими є біологічні та медичні процеси. У таких об'єктах подібна невизначеність обумовлена найчастіше динамічними властивостями. Підвищення кількості очікуваної вимірювальної інформації прямо пов'язане зі зменшенням невизначеності результатів вимірювань [230–235]. Найбільш ефективними при цьому є структурно-алгоритмічні методи [231]. Їхнє застосування в поєднанні з інформаційно-вимірювальними технологіями перетворення первинної кількісно інформації в логічні рішення дозволяє зменшити невизначеність останніх.

Основною метою розробки методу є отримання додаткової інформації про динамічні властивості об'єкта контролю, обумовлених зміною його станів на основі застосування регресійних моделей дисперсійного аналізу нестаціонарних вимірювальних сигналів динамічної ЗАРМ. Для цього доцільно використовувати математичну модель групового регресійного перетворення.

Розглянемо послідовність результатів вимірювань фізичної величини X із зазначенням моментів часу її вимірювання. Така послідовність становить впорядковану за часом множину двовимірних спостережень

$$X(t)=\left\{(x_1,t_1),(x_2,t_2),...,(x_N,t_N)\right\}. \qquad (5.1)$$

Нехай Θ_0 і Θ_1 – позначення функціональних станів об'єкта контролю (в «нормі» і, відповідно, не в «нормі»). Інформацію про зміну виду стану об'єкта контролю несуть інформативні параметри вимірювального сигналу $X(t)$. Виділення таких параметрів пов'язано із завданням синтезу математичної моделі сигналу $X(t)$, в якій зміна виду стану об'єкта призводить до зміни, наприклад, середніх значень інформативних параметрів (коефіцієнтів моделі). У загальному вигляді модель вимірювального сигналу містить m умовних параметрів $\overline{a_1}(\Theta_r),...,\overline{a_m}(\Theta_r)$, $r=\overline{0,1}$, середнє значення яких змінюється зі зміною стану об'єкта контролю

$$a_l=\begin{cases}\overline{a}_l^{(0)}, & \text{якщо } r=0;\\ \overline{a}_l^{(1)}, & \text{якщо } r=1,\end{cases}$$

де $l=\overline{1,m}$.

На відміну від статичних вимірюваних величин, динамічні сигнали відкривають додаткову можливість отримання інформаційної надмірності за рахунок обліку кореляційних зв'язків цих сигналів з часом їх спостереження. Кореляція може проявлятися в наявності трендів (першого порядку і вище). Додатковими інформативними параметрами в цьому випадку будуть коефіцієнти, які входять у математичні моделі подібних трендів. Такі тренди є регресіями величини X на час t, а залишкова дисперсія такої регресії може бути використана для оцінки отримуваної під час контролю інформації (інформативність тим вище, чим менше залишкова дисперсія). Розглянемо апроксимацію сигналу $X(t)$ послідовністю K часткових лінійних регресій з випадковими коефіцієнтами

$$x_{j,i} = A_j + B_j t_{j,i}, \;\; j = \overline{1,k}; \;\; i = \overline{1,n_j},$$

де k – число груп результатів вимірювань, для яких побудовані часткові регресії; n_j – число результатів вимірювань в j-й групі.

Загальна кількість вимірювань дорівнює

$$N = \sum_{j=1}^{k} n_j \quad .$$

Нехай

$$\hat{X}_{j,i} = A + Bt_{j,i}$$

є спільною регресією X на t, коефіцієнти якої визначені за всією множиною (5.1) двовимірних результатів спостережень. Коефіцієнти ж $\left\{A_j, B_j\right\}_1^k$ часткових регресій визначаються за результатами відповідних групових вимірів.

Відомо з [233], що суму S квадратів відхилень результатів спостережень від загального середнього $\overline{x}$

$$S = \sum_{j=1}^{k} \sum_{i=1}^{n_j} \left(x_{i,j} - \overline{x} \right)^2,$$

можна розкласти на п'ять доданків

$$S = S_0 + S_{WG} + S_G + S_W + S_R, \qquad (5.2)$$

де

$$S_0 = w_0 B_0^2,$$

$$S_{WG} = \frac{w_c w_m}{w_0} \left(B_c - B_m \right)^2,$$

$$S_G = \sum_{j=1}^{k} n_j \left[\overline{x}_j - \overline{x} - B_m \left(\overline{t}_j - \overline{t} \right) \right]^2,$$

$$S_W = \sum_{j=1}^{k} w_j \left(B_j - B_c \right)^2,$$

$$S_R = \sum_{j=1}^{k} \sum_{i=1}^{n_j} \left[x_{j,i} - \overline{x}_j - B_j \left(t_{j,i} - \overline{t}_j \right) \right]^2.$$

У свою чергу

$$w_m = \sum_{j=1}^{k} n_j \left(\overline{t}_j - \overline{t} \right)^2,$$

$$w_j = \sum_{i=1}^{n_i} \left(t_{j,i} - \overline{t}_j \right)^2,$$

$$w_c = \sum_{j=1}^{k} w_j,$$

$$w_0 = w_m + w_c,$$

де $\overline{x}, \overline{t}$ – загальні середні за множинми $\left\{ x_S \right\}_1^N$ і $\left\{ t_S \right\}_1^N$;

$\overline{x_j}, \overline{t_j}$ – групові середні за множинми $\left\{ x_{j,i} \right\}_{j=1}^k$ і $\left\{ t_{j,i} \right\}_{j=1}^k$.

Під час вибору інформативних параметрів враховуємо, що сума S_R дозволяє оцінити залишкову дисперсію $\overline{S_R}$ даної регресійної моделі результату вимірювання

$$\overline{S_R} = \frac{S_R}{N - 2k}, \tag{5.3}$$

і виберемо як інформативні параметри моделі статистики

$$\begin{cases} F_0 = S_0 / \overline{S_R}; \\ F_{WG} = S_{WG} / \overline{S_R}; \\ F_G = S_G / \left[\overline{S_R} (k-2) \right]; \\ F_W = S_w / \left[\overline{S_R} (k-1) \right], \end{cases} \tag{5.4}$$

де $\overline{S_R}$ – залишкова дисперсія регресійної моделі, яка визначається за формулою (5.3).

Дані статистики є відносинами середніх квадратів сум S_0, S_{WG}, S_G і S_W до середнього квадрата залишкової суми S_R, тобто є випадковими величинами з F-розподілом Фішера–Снедекора.

Дисперсійне розкладання (5.2) дозволяє розраховувати F-статистики відповідно до формул (5.4) за реалізаціями сигналу $X(t)$. Умовами такого розкладу є:

1. Нормальність розподілу випадкового залишку

$\varepsilon_{j,i} = x_{j,i} - \overline{x_j} - B_j \left(t_{j,i} - \overline{t_j} \right)$, $\varepsilon_{j,i} \approx NORM(0, \sigma_\varepsilon^2)$;

2. $M\left[\varepsilon_{j,i} \right] = 0$;

3. $M\left[\varepsilon_{j,i}^2 \right] = \sigma_\varepsilon^2$;

4. Некорельованість залишків

$M\left[\varepsilon_{j,i} \times \varepsilon_{j,z} \right] = 0$ для всіх $i \neq z$.

Інформативність будь-якої з F-статистик згідно з формулою (3.16) визначається кількістю інформації, яку можна отримати за цією статистикою про вид стану Θ_r об'єкта контролю. Перевага F-статистик – незалежність один від одного через незалежність членів дисперсійного розкладання (5.2) [223]. Це означає, що статистики (5.4) можна розглядати як складові вектора

$$\vec{F} = \left(F_0, F_{WG}, F_G, F_W\right), \tag{5.5}$$

що є багатовимірним інформативним параметром. Повна інформація визначатиметься сумою

$$I = I_0 + I_{WG} + I_G + I_W \quad , \tag{5.6}$$

де складові правої частини можуть бути розраховані незалежно один від одного [305].

Кількість інформації (5.6) характеризує параметри, які визначаються як складові повної дисперсії вимірювального сигналу $X(t)$ на інтервалі спостереження $(0, t_N)$. Ця дисперсія є лінійною функцією залишкової дисперсії (5.3).

При нормальному законі розподілу вимірюваної величини X її лінійне, щодо часу t перетворення, характеризуватиметься незалежністю між середніми значеннями і дисперсією [214] (якщо не змінюється ширина інтервалу спостереження). Тому інформація про зміну стану об'єкта Θ, отримана за F-статистиками дисперсійного розкладання (5.2), може доповнювати інформацію, знайдену за вимірюванням середнього значення величини X.

5.2 Розробка математичної моделі і методи обробки риноманометричних даних у динаміці

Виконаємо оцінку кількості додаткової інформації в отриманих даних. Нехай $F_{\cdot}^{(0)}$ і $F_{\cdot}^{(1)}$ – статистики (5.3) дисперсійного розкладання (5.2), де точка замінює один з індексів «0», «WG», «G» або «W». Дані статистики, як випадкові величини, змінюють, в загальному випадку, нецентральний F-розподіл з V_1 і V_2 ступенями свободи з параметром нецентральних $\lambda_{\cdot}^{(r)}$ (де $r = \overline{0,1}$)

$$F_{\cdot}^{(r)} \approx F_{V_1;V_2} \cdot \lambda_{\cdot}^{(r)} \quad . \tag{5.7}$$

Середнє і дисперсія статистики $F_{\cdot}^{(r)}$, відповідно, дорівнюють [234]

$$\chi_1^{(r)} = \frac{V_2}{(V_2 - 2)}\left(1 + \frac{\lambda_{\cdot}^{(r)}}{V_1}\right), \tag{5.8}$$

$$\chi_2^{(r)} = \frac{2V_2^{\,2}}{(V_2 - 2)(V_2 - 4)}\left(\frac{2\lambda_{\cdot}^{(r)}}{V_1} + \frac{\left(1 + \lambda_{\cdot}^{(r)}\right)^2}{(V_2 - 2)}\right). \tag{5.9}$$

Кількість інформації, отримуваної за статистикою $F.^{(r)}$ (5.10), визначається з виразу [233]

$$I = \log_2 \sqrt{1 + \left(\frac{\sigma_F}{\sigma_{\Delta F}} \right)^2} \ , \qquad (5.11)$$

де σ_F^2 – дисперсія F-статистики до виміру (контролю),

$\sigma_{\Delta F}^2$ – дисперсія F-статистики після вимірювання.

З урахуванням виразів (5.8) і (5.9) дисперсії σ_F^2 і $\sigma_{\Delta F}^2$ можна визначити як

$$\sigma_F^2 \geq \left(\chi_1^{(0)} - \chi_1^{(1)} \right)^2 \Big/ 12 \ ,$$

$$\sigma_{\Delta F}^2 \geq 4 \left(\max \chi_2^{(r)} \right) \ .$$

У табл. 5.1 наведено результати дисперсійного аналізу виміряних значень сигналу $X(t)$ для станів Θ_0 і Θ_1 біологічного об'єкта контролю $\left(N = 9, \ K = 3, \ n_j = n \text{ для всіх } j = \overline{1,3} \right)$, яким є процес динамічної форсованої ЗАРМ.

Таблиця 5.1

Результати дисперсійного аналізу

Середній квадрат відхилень	Число ступенів свободи	Середній квадрат	F-статистика	I (біт)
$S_0^{(1)} = 0,736502$ $S_0^{(1)} = 1,857845$	$V_0 = 1$	$\overline{S_0}^{(0)} = S_0^{(0)}$	$F_0^{(0)} = 92,9$ $F_0^{(1)} = 368,1$	0,92
$S_{WG}^{(1)} = 0,01579$ $S_{WG}^{(1)} = 0,0119$	$V_{WG} = 1$		$F_{WG}^{(0)} = 1,9$ $F_{WG}^{(1)} = 2,4$	0,00078
$S_G^{(0)} = 0,0088707$ $S_G^{(1)} = 0,11395$	$V_G = 1$		$F_G^{(0)} = 1,119$ $F_G^{(1)} = 22,6$	0,17
$S_\varepsilon^{(0)} = 0,03964$ $S_\varepsilon^{(1)} = 0,02523$	$V_\varepsilon = 1$	$\overline{S_\varepsilon}^{(0)} = 0,007928$ $\overline{S_\varepsilon}^{(1)} = 0,005045$	–	–

Примітка. Для забезпечення умови $V_2 > 4$ суми S_W і S_R об'єднані, відповідно зросла і кількість ступенів свободи $\left(V_2 = V_\varepsilon = 5 \right)$, сумарна кількість інформації $I_F = 0,92 + 0,00078 + 0,17 = 1,09078$ (біт).

У цій самій таблиці дані значення кількості інформації (в бітах), при $\chi_1^{(0)} = F.^{(0)}$; $\chi_1^{(1)} = F.^{(1)}$, а дисперсія $\chi_2^{(r)}$ подана як функція середнього $\chi_1^{(r)}$ [233]

$$\chi_2^{(r)} = \frac{4V_2}{(V_2 - 4)} \chi_1^{(r)} + \frac{2V_1^2}{(V_2 - 4)} \chi_1^{(r)2} - \frac{4V_2^2}{(V_2 - 2)(V_2 - 4)}.$$

Якщо врахувати, що оцінка [233] кількості інформації (5.6) за вимірюванням середнього значення сигналу $X(t)$, тобто оцінка за $\overline{X}^{(0)}$ і $\overline{X}^{(1)}$ дала величину $I_{\overline{X}} = 2,69$ (біт), то додатковий приріст $I_{\overline{X}} = 1,09078$ (біт) становить не менше 40%, що вказує на ефективність запропонованого методу дисперсійного перетворення вимірювального сигналу. На рис. 5.1 наведено тимчасові ряди виміряних значень сигналу $X(t)$ для станів $\Theta = \Theta_0$ і $\Theta = \Theta_1$ з послідовною регресійною апроксимацією рядів.

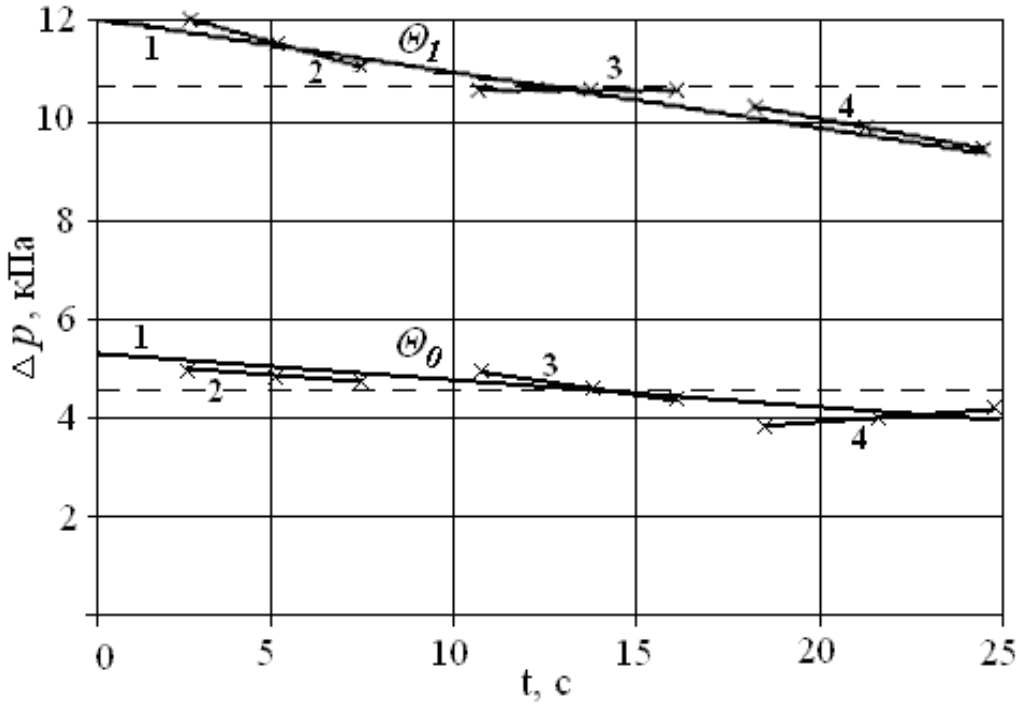


Рис. 5.1. Тимчасові ряди виміряних значень сигналу тиску $\Delta p(t)$ з послідовною регресійною апроксимацією рядів у нормі (стан Θ_0) і при патології (стан Θ_1):
1 – прямі загальних регресій; 2, 3 і 4 – прямі часткових регресій
(пунктиром показано середні значення)

На рис. 5.1 наочно видно зміну кутових коефіцієнтів часткових регресій (2; 3 і 4) порівняно з кутовим коефіцієнтом прямої загальної регресії (1) зі зміною стану об’єкта контролю (як в умовній нормі, так і при патології).

Для аналізу риноманометричних даних необхідно вибрати модель за критерієм найменшої похибки визначення діагностичних параметрів і отримання найбільшої кількості інформації.

Основними моделями при цьому є:

– усереднюючі

$$x_i = \overline{x} + \varepsilon_i^{(1)} \quad , \tag{5.12}$$

з поданням результатів вимірювань x_i середнім значенням $\overline{x}$ із залишками $\varepsilon_i^{(1)}$;

– на основі загальної лінійної регресії з поданням результатів вимірювань x_i рівнянням загальної лінійної регресії з кутовим коефіцієнтом B і залишками $\varepsilon_i^{(2)}$

$$x_i = \bar{x} + B\left(t_i - \bar{t}\right) + \varepsilon_i^{(2)}, \tag{5.13}$$

де $\bar{x}$ – середнє значення сигналу,

$\bar{t}$ і t_i – середнє та i-е значення часового інтервалу, відповідно;

– з урахуванням часткових лінійних регресій з поданням результатів вимірювань сумою значень загальної лінійної регресії з кутовим коефіцієнтом і часткових лінійних регресій з кутовими коефіцієнтами b_j із залишками $\varepsilon_{ji}^{(3)}$, причому часовий ряд виміряних значень ділиться на j інтервалів

$$x_{ji} = \bar{x} + B\left(t_{ji} - \bar{t}\right) + b_j\left(t_{ji} - \bar{t}_j\right) + \varepsilon_{ji}^{(3)}, \tag{5.14}$$

де $\bar{t}_j$ – середнє значення часу на j-му інтервалі;

t_{ji} – i-е значення часу на j-му інтервалі.

Згідно з [233], величини залишків моделей (5.13), (5.13) і (5.14) мають співвідноситися як

$$\varepsilon_{ji}^{(3)} < \varepsilon_i^{(2)} < \varepsilon_i^{(1)}, \tag{5.15}$$

де верхній індекс у дужках вказує на номер моделі.

Тестування моделей проводилося окремо для сигналів перепаду тиску і витрати повітря з кількістю вимірювань і розбиття часового ряду на 3 інтервали $(n = 3)$ по 3 значення в кожному інтервалі $j \in [1, J]; J = 3$.

Для тестування аналогічно формулі (5.11) обчислювалися значення кількості інформації [233]

$$I = \log_2 \sqrt{1 + \left(\frac{\sigma_x}{\sigma_{\varepsilon_x}}\right)^2}, \tag{5.16}$$

де σ_x і σ_{ε_x} – середньоквадратичне відхилення сигналу x і залишків ε, відповідно, які обчислюють за формулами

$$\bar{x} = \frac{1}{m}\sum_{i=1}^{m} x_i; \quad \sigma_x = \sqrt{\frac{1}{m}\sum_{i=1}^{m}\left(x_i - \bar{x}\right)^2}$$

$$\bar{\varepsilon}^* = \frac{1}{I}\sum_{i=1}^{I} \varepsilon_i^*; \quad \sigma_{\varepsilon_x}^* = \sqrt{\frac{1}{m}\sum_{i=1}^{m}\left(\varepsilon_{x_i}^* - \bar{\varepsilon}_x^*\right)^2},$$

де * позначає довільний вибір виду моделі.

У табл. 5.2 і 5.3 наведено результати розрахунку кількості інформації, отримуваної в ході використання трьох видів моделей для умовної норми і з порушенням носового дихання при викривленні носової перегородки, відповідно. Очевидно, що як при нормі, так і з порушенням носового дихання, модель, заснована на використанні часткових регресій, володіє значно більшою інформаційною ємністю. Це пояснюється урахуванням тенденції до втоми

пацієнта під час виконання послідовності форсованих дихальних маневрів через витратомір з великим вхідним опором за допомогою прямої загальної регресії та обліку локальних трендів, які враховують індивідуальні особливості дихання пацієнта під час тестування.

Поняття кількості інформації є в даному випадку досить абстрактним, але дозволяє оцінити ступінь зменшення невизначеності в ході використання різних методів обробки вимірюваних даних. Справедливість нерівності (3.26) про зменшення величин залишків моделей апроксимації доведена так само експериментально, що видно за даними табл. 5.2 і 5.3. Для сигналів витрати повітря кількість інформації I відповідно дорівнює 0,22; 0,36 і 0,51 в нормі і 0,24; 0,33 і 0,56 з порушення носового дихання. Проте зменшення невизначеності під час аналізу значень витрати повітря істотно менше (майже вдвічі), що свідчить про меншу інформативність сигналу витрати повітря через носові ходи порівняно із значенням перепаду тиску, що його викликав. Усереднені величини за 30 пацієнтами з порушенням носового дихання при викривленні носової перегородки і такій же чисельності контрольної групи (умовної норми) мають відповідати вказаним у табл. 5.2 і 5.3.

З огляду на вимоги мінімального дискомфорту пацієнта при риноманометріичному обстеженні доцільно організувати процедуру тестування носового дихання на обробку даних вимірювань так:

– пацієнтові виконати 9 повних дихальних маневрів за максимально можливого форсування фази вдиху $(m=9)$;

– отримати максимальні значення вимірюваних величин перепаду тиску і витрати повітря у кожному циклі вдиху і сформувати тимчасовий ряд (x_i, t_i), $i \in [1, m]; m = 9$;

– розбити отриманий тимчасовий ряд на 3 інтервали $(n=3)$ по 3 значення в кожному інтервалі $j \in [1, J]; J = 3$;

– виконати апроксимацію даних вимірювань згідно з (5.14).

Таблиця 5.2

Результати аналізу кількості інформації
під час обробки риноманометричних даних
за допомогою трьох видів моделей в нормі

Вид моделі	M_x, кПа	σ_x, кПа	M_ε, кПа	σ_{ε_x}, кПа	I (біт)
$x_i = \overline{x} + \varepsilon_i^{(1)}$	5,3	0,26	0,20	0,33	0,35
$x_i = \overline{x} + B(t_i - \overline{t}) + \varepsilon_i^{(2)}$			0,14	0,2	0,70
$x_{ji} = \overline{x} + B(t_{ji} - \overline{t}) + b_j(t_{ji} - \overline{t_j}) + \varepsilon_{ji}^{(3)}$			0,10	0,15	1,02

Таблиця 5.3

Результати аналізу кількості інформації
під час обробки риноманометричних даних
за допомогою трьох видів моделей з порушенням носового дихання

Вид моделі	M_x, кПа	σ_x, кПа	M_ε, кПа	σ_{ε_x}, кПа	I (біт)
$x_i = \bar{x} + \varepsilon_i^{(1)}$	11,28	0,60	0,46	0,75	0,35
$x_i = \bar{x} + B\left(t_i - \bar{t}\right) + \varepsilon_i^{(2)}$			0,30	0,45	0,72
$x_{ji} = \bar{x} + B\left(t_{ji} - \bar{t}\right) + b_j\left(t_{ji} - \bar{t_j}\right) + \varepsilon_{ji}^{(3)}$			0,24	0,39	0,86

Таким чином, використання методу кусково-лінійної регресійної апроксимації вимірювальних сигналів перепаду тиску і витрати повітря дозволило отримати додаткову інформацію щодо змін випадкових коефіцієнтів часткових лінійних регресій.

Доведено, що додаткову інформацію, крім коефіцієнтів часткових лінійних регресій, несуть чотири члени дисперсійного розкладання, що вказує на можливість отримання додаткової інформації.

Так само показано на практичному прикладі риноманометричної діагностики, що додаткове збільшення очікуваної вимірювальної інформації може досягати 40% від початкової. Остання була отримана з аналізу змін середніх значень виміряних сигналів.

Запропонований метод дисперсійного аналізу часткових лінійних регресій дозволяє отримати додаткову інформацію за складовими дисперсійного розкладання сигналу. Така процедура еквівалентна процедурі спектрального аналізу за відсутності інформації про енергетичний спектр нестаціонарного за середнім вимірювального сигналу, оскільки досліджувані послідовності результатів вимірювань є тимчасовими рядами.

Розроблений метод можна розглядати в рамках вдосконалення інформаційно-вимірювальних технологій контролю та технічної діагностики при обмеженнях за часом спостереження (або числом вимірювань) і при апріорній невизначеності властивостей об’єкта контролю і діагностики. Метод дозволяє планувати багаторазові групувати вимірювання, отримані на базі нестаціонарних вимірювальних сигналів з апріорі невідомими частотними властивостями, до яких належать результати функціональних обстежень верхніх дихальних шляхів людини, зокрема, риноманомеричної діагностики.

5.3 Порівняльна оцінка достовірності методів риноманомеричних вимірювань

Анатомічні особливості верхніх дихальних шляхів, складність фізіологічного процесу дихання і відсутність фактичного еталона призводять до того, що оцінка носового опору, що характеризує ступінь порушення дихання,

істотно залежить від методу вимірювання і володіє значною варіабельністю. Тому актуальним завданням є розширення діагностичних можливостей методів дослідження й обґрунтування доцільності їх застосування з діагностикою конкретних патологій.

Під час розробки нових діагностичних методів і засобів заключним етапом є порівняння дискримінантних характеристик запропонованого методу з існуючими. При цьому важливим завданням є вибір інформативних параметрів діагностики та контролю, а також критерію, за яким порівнюватимуться дискримінантні можливості методів.

Ефективність вирішення завдань контролю станів об'єктів з випадковими властивостями, як правило, залежить від правильного вибору максимально інформативної системи параметрів (ознак), чутливих до змін характеристик об'єкта. Будь-який контроль формально реалізує процедуру тестування, ефективність результату якого визначається достовірністю, тобто ймовірністю прийняття правильного рішення [234]. З невизначеністю властивостей об'єкта завдання відбору інформативних параметрів стає проблемним. Особливо, якщо ускладнено метрологічне забезпечення інформаційних перетворень у структурі системи контролю, що часто має місце під час проведення медичної діагностики.

Вибір оптимальної, за критерієм максимуму достовірності, системи інформаційних ознак є класичною задачею статистичного синтезу в умовах апріорної невизначеності [230, 234]. Ранжування ознак за інформативністю здійснюють за величиною показника достовірності контролю [231] або ймовірністю помилок [233].

У даному підрозділі проводиться оцінка можливості використання критеріїв і моделей параметричного розпізнавання (дискримінації) в ході порівняння діагностичних можливостей розглянутих вище риноманометричних методів дослідження.

Як було показано вище, загальноприйнятим є метод передньої активної риноманометрії, що проводиться при спокійному диханні і заснований на аналізі даних витрати повітря за фіксованих значень перепаду тиску (300 Па). Запропонований метод задньої риноманометрії під час форсованого дихання дозволяє оцінити функцію носового клапана та отримати інформацію про граничні величини перепаду тиску і витрати повітря, що особливо важливо для спортивної медицини. Порівняння методів діагностики проводилося на базі оториноларингологічного відділення Харківського центру екстреної медичної допомоги та медицини катастроф (Харківської обласної клінічної лікарні) за допомогою розробленого комп'ютерного риноманометра КРМ типу ТНДА-ПРХ (свідоцтво про державну метрологічну атестацію пристрої ПРХ, № 05-0102 від 01.04.2010 р.), конструкція якого розглядається в монографії [230].

З огляду на те, що коефіцієнт оцінки носового опору (3.15) в ході визначення дискримінантних властивостей методів діагностики додаткової інформації не несе, оскільки є тільки відношенням виміряних величин, то аналізу підлягатимуть розподіли безпосередньо вимірюваних параметрів – перепаду тиску і витрати повітря.

Розглянемо модель лінійної дискримінації. Інформативний параметр X, який використовується для отримання інформації про апріорно невизначені властивості об'єкта контролю, може розглядатися як випадкова величина. Остання, в разі двох станів об'єкта (Θ_0 – норма, Θ_1 – відхилення від норми) характеризується умовними щільностями розподілу ймовірностей

$$X \approx f\left(X/\Theta_0\right), \text{ якщо } \Theta \in \Theta_0,$$

$$X \approx f\left(X/\Theta_1\right), \text{ якщо } \Theta \in \Theta_1.$$

Якщо $m^{(0)}, m^{(1)}, \sigma^{(0)^2}, \sigma^{(1)^2}$ – середні і дисперсії величини X для умов $\Theta \in \Theta_0$, і $\Theta \in \Theta_1$ відповідно, то за гаусових розподілів, $f\left(X/\Theta_0\right)$, $f\left(X/\Theta_1\right)$ ймовірність помилки прийняття рішень у вигляді станів об'єкта визначається при $\sigma^{(0)^2} = \sigma^{(1)^2}$ через інтеграл ймовірності $\Phi(\cdot)$ [305]

$$P_{ош} = 1 - \Phi\left(\delta/2\right), \tag{517}$$

де

$$\delta = \left| \frac{m^{(0)} - m^{(1)}}{\sigma} \right|. \tag{5.18}$$

Якщо $\sigma^{(0)^2} \neq \sigma^{(1)^2}$, то межа для $P_{ош}$ може оцінюватися нерівністю

$$P_{ош} \leq 1 - \Phi\left(\delta/2\right). \tag{5.19}$$

За багатопараметричному контролі, коли число інформативних параметрів $X_1, ..., X_n$ більше одного $\left(n \geq 2\right)$ змінна δ у виразах (5.17) або (5.19) описується рівнянням

$$\delta = \sqrt{\sum_{i=1}^{n} \left(\frac{m_i^{(0)} - m_i^{(1)}}{\sigma_i} \right)^2}. \tag{5.20}$$

Квадрат цієї змінної

$$\delta^2 = \sum_{i=1}^{n} \left(\frac{m_i^{(0)} - m_i^{(1)}}{\sigma_i} \right)^2$$

називають квадратичною відстанню махаланобіса між контрольованими станами (між векторами середніх за станами Θ_0 і Θ_1) [234].

Об'єкт контролю в цьому випадку становить вектор-стовпчик виміряних значень

$$\overline{x} = \begin{vmatrix} x_1 \\ x_2 \\ \cdot \\ \cdot \\ \cdot \\ x_n \end{vmatrix},$$

з умовною n-вимірною нормальною щільністю розподілу

$$f\left(\bar{x}/\Theta_k\right)=\left(2\pi\right)^{-n/2}\left|\Sigma\right|^{-1/2}\exp\left[-\frac{1}{2}\left(\bar{x}-\bar{m}^{(k)}\right)\Sigma^{-1}\left(\bar{x}-\bar{m}^{(k)}\right)\right] \quad . \quad (5.21)$$

У рівнянні (3.32) вектор середніх $\bar{m}^{(k)}$ і дисперсійна матриця Σ мають такий вигляд (k – номер стану об'єкта, $k=\overline{0,1}$)

$$\bar{m}^{(k)}=\begin{vmatrix} m_1^{(k)} \\ m_2^{(k)} \\ \cdot \\ \cdot \\ \cdot \\ m_n^{(k)} \end{vmatrix}, \quad \Sigma=\begin{vmatrix} \sigma_1^{\,2} & & & & \\ & \sigma_2^{\,2} & & & \\ & & \cdot & & \\ & & & \cdot & \\ & & & & \cdot \\ & & & & \sigma_n^{\,2} \end{vmatrix}.$$

Вираз (5.20) передбачає взаємну незалежність складових вектора за лінійній моделі дискримінації [233].

Імовірність помилки тим менше, чим більше δ, тобто чим більше нормований за дисперсією квадрат відстані між векторами середніх.

Таким чином, змінні δ (або δ^2) за виразом (5.20) дозволяють кількісно порівнювати за дискримінуючою здатністю (фактично, за інформативністю) не лише поодинокі інформативні сигнали, але й підмножини (системи) сигналів.

Для практичного застосування моделі дискримінації розглянемо задачу оцінки ефективності двох методів риноманометрії, позначених як метод $M_{\text{а}}$ з вимірами під час спокійного дихання, і метод $M_{\text{б}}$ з вимірами під час форсованого дихання, які забезпечують отримання вимірювальної інформації про стан діагностичного об'єкта, поданого:

а) статичною моделлю (метод $M_{\text{а}}$ риноманометрії під час спокійного дихання);

б) динамічною моделлю (метод $M_{\text{б}}$ риноманометрії під час форсованого дихання).

У методі $M_{\text{а}}$ під час спокійного дихання вимірювані фізичні величини (X_1 – перепад тиску ΔP і X_2 – витрата повітря Q, кількість вимірюваних параметрів $n=2$) на відміну від методу $M_{\text{б}}$ не корельовані з тривалістю інтервалу спостереження. Стан умовної норми і порушення носового дихання позначаються відповідно Θ_0 і Θ_1. Загалом було обстежено 60 пацієнтів, розділених на дві групи по 30 чоловік у нормі і з ускладненням носового дихання. Вимірювання для кожного пацієнта проводилися двома методами – під час спокійного $M_{\text{а}}$ і форсованого $M_{\text{б}}$ вдиху за десятьма циклами дихання. При цьому за загальним алгоритмом для кожного методу обчислювалися максимальні значення перепаду тиску ΔP і витрати повітря Q у верхніх дихальних шляхах пацієнта в кожному циклі вдиху, і проводилося їх

усереднення за десятьма циклами дихання. Потім для кожної групи пацієнтів перебували статистичні показники – середні значення $m_1^{(0)} = \overline{\Delta P}$, $m_2^{(0)} = \overline{Q}$, $m_1^{(1)} = \overline{\Delta P}$, $m_2^{(1)} = \overline{Q}$ в нормі та з порушеннями носового дихання, відповідно, а також середньоквадратичні відхилення відповідних показників, причому для розрахунків вибиралися максимальні значення середньоквадратичних відхилень $\sigma_1 = \max\left(\sigma_{\Delta P}^{(0)}, \sigma_{\Delta P}^{(1)}\right)$ і $\sigma_2 = \max\left(\sigma_Q^{(0)}, \sigma_Q^{(1)}\right)$, відповідно. Далі згідно з введеним позначенням виконувалися розрахунки відстані махаланобіса за формулою (5.20) і ймовірності помилки прийняття рішення за формулою (5.19) для кожного методу. Результати розрахунків, наведені в табл. 5.4, показують очевидність того, що запропонований у роботі метод риноманометричних вимірювань під час форсованого дихання володіє великими (в 1,7 рази) дискримінантними властивостями порівняно з традиційним і дозволяє знизити ймовірність помилки з прийняттям діагностичного рішення з 0,36 до 0,21. Це дозволяє використовувати даний метод для функціональної діагностики верхніх дихальних шляхів [230].

Проведемо оцінку впливу динамічних властивостей процесу риноманометричної діагностики на дискримінантні характеристики методу. За наведеною вище методикою форсованої динамічної риноманометрії в кожному циклі вдиху обчислювалися амплітудні значення перепаду тиску Δp і витрати повітря Q у верхніх дихальних шляхах пацієнта ті проводилося їх усереднення за дев'ятьма циклами дихання, а також знаходилися по чотири F-статистики F_0, F_{WG}, F_G, F_W (див. табл. 5.4) для кожного вимірюваного сигналу (Δp і Q).

Таблиця 5.4

Результати дискримінантного аналізу
для статичних методів риноманометричної діагностики

Параметр, розмірність	Тип методу:			
	традиційний, $M_а$		запропонований, $M_б$	
	Θ_0	Θ_1	Θ_0	Θ_1
$\overline{\Delta P}$, кПа	0,30	0,30	8,7	16,5
$\sigma_{\Delta P}$, кПа	0,07	0,07	2,26	3,80
$\overline{Q}$, л/с	0,40	0,20	3,10	0,80
σ_Q, л/с	0,11	0,08	1,60	0,43
δ	1,80		2,5	
$P_{ОШ}$	$\leq 0{,}36$		$\leq 0{,}21$	

Потім, для кожної групи пацієнтів розраховувались статистичні показники: середні значення $m_i^{(0)}$ і $m_i^{(1)}$ та середньоквадратичні відхилення

відповідних показників, причому для розрахунку за формулою (5.20) вибиралося максимальне середньоквадратичне відхилення

$$\sigma_i = \max\left(\sigma_i^{(0)}, \sigma_i^{(1)}\right).$$

Таким чином, у розрахунках брали участь десять інформативних параметрів X_i $\left(i = \overline{1,10}\right)$ – по п'ять для кожного вимірюваного сигналу. Перші п'ять $\left\{X_i\right\}_1^5$ належать до характеристик перепаду тиску Δp, а другі п'ять $\left\{X_i\right\}_6^{10}$ – до характеристик витрати повітря.

Далі, згідно з введеним позначенням, виконувалися розрахунки відстані махаланобіса за формулою (5.20) і ймовірності помилки прийняття рішення за формулою (3.30) для кожного параметра. Результати розрахунків наводяться в табл. 5.5 і 5.6.

Таблиця 5.5

Дискримінантні характеристики
параметрів сигналу перепаду тиску Δp

Параметр		Стан об'єкта контролю		Відстань Махалонобіса δ	Імовірність помилки $P_{пом}$
		Θ_0	Θ_1		
X_1	$\overline{\Delta p}$, кПа	8,70	16,5	2,1	$\le 0{,}3$
	$\sigma_{\Delta p}$, кПа	2,26	3,80		
X_2	$\overline{F_{0p}}$	111,50	65,20	0,98	$\le 0{,}62$
	$\sigma_{F0\Delta p}$	47,10	22,14		
X_3	$\overline{F_{WG\,\Delta p}}$	6,41	18,70	0,87	$\le 0{,}68$
	σ_{FWGp}	3,47	14,20		
X_4	$\overline{F_{G\,\Delta p}}$	48,40	38,20	0,50	$\le 0{,}81$
	$\sigma_{FW_{\Delta p}}$	19,6	12,95		
X_5	$\overline{F_{W_{\Delta p}}}$	18,25	35,30	0,92	$\le 0{,}65$
	$\sigma_{FW\,\Delta p}$	4,77	11,32		
$\left\{X_i\right\}_1^5$	$\delta_{\Delta p}$			2,7	$\le 0{,}18$

Як видно з наведених таблиць, за середнім значенням сигналів, а також з урахуванням їх F-статистик, метод аналізу динамічного випадкового сигналу перепаду тисків володіє великими дискримінантними характеристиками порівняно з витратою повітря (в 1,5 рази більшу відстань Махалонобіса як під час аналізу середніх значень сигналів, так і з урахуванням F-статистик).

Це пояснюється фізичними можливостями пацієнта за ускладненого носового дихання короткочасно забезпечити витрату, близьку до нормальної за рахунок підвищення перепаду тисків на носових проходах.

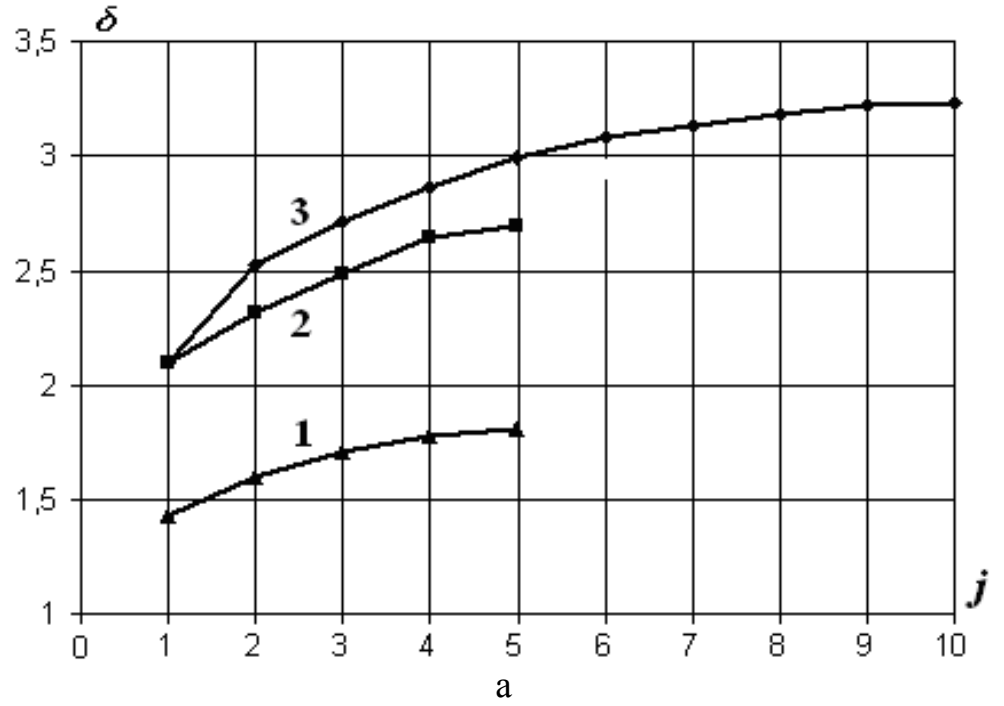
При цьому, використовуючи тільки середні значення сигналів перепаду тисків і витрати повітря, можна отримати ймовірність помилки менше 0,21. Облік усіх інформативних параметрів випадкових вимірювальних сигналів дозволяє отримати відстань Махалонобіса $\delta_{\{X_i\}_1^{10}} = 3,25$ і, відповідно, знизити ймовірність помилки до величини менше 0,1. При ранжируванні ознак у порядку спадання відстані Махалонобіса можна провести аналіз впливу кількості параметрів на дискримінантні характеристики методу діагностики.

Таблиця 5.6

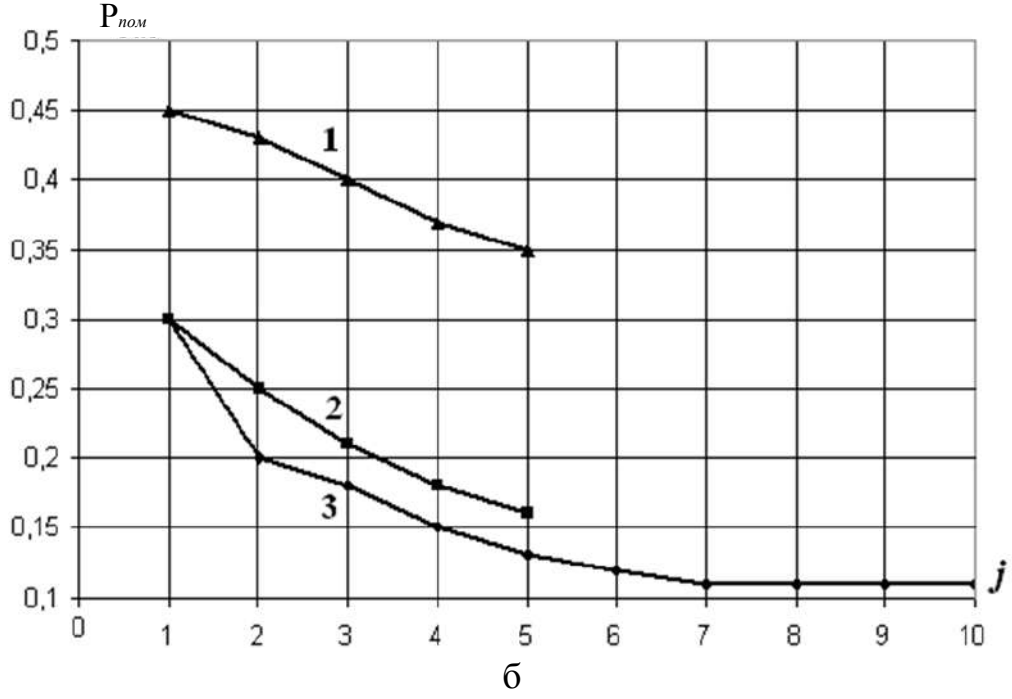
Дискримінантні характеристики параметрів сигналу витрати повітря Q

Параметр		Стан об'єкта контролю		Відстань Махалонобіса δ	Імовірність помилки $P_{пом}$
		Θ_0	Θ_1		
X_6	$\overline{Q}$, л/с	3,10	0,80	1,43	$\le 0,48$
	σ_Q, л/с	1,60	0,43		
X_7	$\overline{F}_{0_Q}$	77,95	47,62	0,74	$\le 0,72$
	σ_{F0Q}	40,50	12,37		
X_8	$\overline{F_{WG\,Q}}$	4,50	3,80	0,33	$\le 0,88$
	σ_{FWG_Q}	2,15	1,76		
X_9	$\overline{F_{G\,Q}}$	6,10	7,94	0,50	$\le 0,81$
	$\sigma_{FW_{\Delta P}}$	3,12	3,63		
X_{10}	$\overline{F_{W_Q}}$	9,70	13,1	0,60	$\le 0,76$
	σ_{FW_Q}	3,77	5,65		
$\{X_i\}_6^{10}$	δ_Q			1,82	$\le 0,37$

На рис. 5.2, а і б наведено кумулятивні характеристики збільшення відстані Махалонобіса і зменшення ймовірності помилки діагностики як функції розмірності простору та інформативних параметрів тиску і витрати повітря. При цьому очевидним є те, що три найменш значущих параметри на ймовірність прийняття діагностичного рішення практично не впливають і можуть бути виключені з розрахунків [230].



а



б

Рисю 5.2. Результати дискримінантного аналізу даних динамічної ЗАРМ:
а – залежність збільшення відстані Махалонобіса в міру додавання ознак у модель дискримінації $\delta = F(j)$; б – залежність зниження ймовірності помилки прийняття рішення по мірі додавання ознак у модель дискримінації $P_{пом} = f(j)$:
(j – розмірність простору інформативних параметрів:
1 – для сигналу витрати повітря, 2 – для сигналу перепаду тиску,
3 – з урахуванням ознак всіх сигналів

Запропонований оригінальний метод тестування носового дихання пацієнта під час форсованого вдиху дозволив виявити таку закономірність –

найбільшими дискримінантними властивостями володіють параметри сигналу перепаду тисків, оскільки пацієнт під час ускладненого носового дихання короткочасно може забезпечити витрату повітря через носові проходи, близьку до нормальної за рахунок підвищення перепаду тисків.

Використовуючи для аналізу дихання тільки середні з максимальних значень сигналів перепаду тисків і витрати повітря, можна отримати ймовірність помилки діагностичного рішення менше 0,21 $\left(P_{пом} \le 0,21\right)$. Урахування всіх показників діагностичних сигналів дозволяє знизити ймовірність помилки діагностики ускладненого носового дихання до величини, менше 0,1 $\left(P_{пом} \le 0,1\right)$. Таким чином, додавання до параметрів апаратно-методологічних засобів вимірювань F-статистик забезпечує істотне (з 0,21 до 0,1) зниження ймовірності помилки прийняття діагностичного рішення.

Об'єм вибірки $N_{\text{П}}$ (кількість пацієнтів) для проведення клінічної апробації форсованої динамічної ЗАРМ визначається з [230]

$$N_{\text{П}} = \frac{t^2 \cdot \sigma_x^2}{\Delta_x} , \qquad (5.22)$$

де p коефіцієнт довіри, який визначається з інтеграла ймовірності Лапласа як

$$\Phi\left(t\right) = p ,$$

де p – рівень значущості (для проведених досліджень $p = 0,95$, $t = 2$);

σ_x – дисперсія оцінюваної величини, приймається як максимальна з двох дисперсій для вимірюваних сигналів перепаду тиску і витрати повітря (згідно з таблицею $\sigma_x = \sigma_{x_{\max}} = \sigma_{\Delta p} = 3,8$);

Δ_x – гранична помилка вибірки, яка визначається, виходячи з вимог точності згідно з $\Delta_x = 0,4 \cdot \sigma_x$.

Остаточно за формулою (3.33) отримаємо

$$N_{\text{П}} = \frac{2^2 \cdot 3,8_x^2}{1,52^2} \approx 25 .$$

Таким чином, обсяг вибірки не має бути менше 25 пацієнтів за заданих вихідних значень рівня значущості, граничної помилки вибірки та дисперсії оцінюваної величини. У випробуваннях для однакової апріорної ймовірності об'єми вибірок становили по 30 пацієнтів з викривленням носової перегородки і в контрольній групі, відповідно.

Таким чином, запропоновано метод дисперсійного перетворення вимірювальних риноманометричних сигналів, який дозволяє за рахунок обліку динамічних властивостей процесу дихання приблизно втричі підвищити інформативність моделі вимірювань показників носового дихання.

Застосування методу кусково-лінійної регресійної апроксимації вимірювальних сигналів дозволило отримати додаткову інформацію щодо змін

випадкових коефіцієнтів часткових лінійних регресій (інформативність підвищується до 1,5 рази). Така процедура еквівалентна процедурі спектрального аналізу за відсутності інформації про енергетичний спектр нестаціонарного (за середнім значенням) вимірювального сигналу, оскільки досліджувані послідовності результатів вимірювань є тимчасовими рядами. Доведено, що додаткову інформацію, крім коефіцієнтів часткових регресій, несуть чотири члени дисперсійного розкладання сигналів витрати повітря і перепаду тиску, при цьому додаткове збільшення очікуваної вимірювальної інформації може досягати 40% від початкової (за середнім значенням вимірюваних сигналів).

Зменшення невизначеності в ході використання динамічної моделі обробки вимірюваних риноманометричних даних під час аналізу вимірюваних значень витрати повітря істотно менше (майже вдвічі) порівняно з перепадом тиску на носовій порожнині, що свідчить про меншу інформативність сигналу витрати повітря через носові ходи порівняно із значенням перепаду тиску, що його викликав.

Оцінка дискримінантної здатності риноманометричних методів діагностики (традиційного – під час спокійного дихання і розробленого під час форсованого дихання) дозволяє зробити висновок про те, що запропонований у роботі метод риноманометричних вимірювань під час форсованого дихання володіє великими (в 1,7 рази) дискримінантними властивостями порівняно з традиційним і дозволяє знизити ймовірність помилки в процесі прийняття діагностичного рішення з 0,36 до 0,21, що дозволяє використовувати даний метод для функціональної діагностики верхніх дихальних шляхів. Додавання до параметрів апаратно-методологічних засобів F-статистик вимірюваних сигналів забезпечує істотне (з 0,21 до 0,1) зниження ймовірності помилки прийняття діагностичного рішення.

ВИСНОВКИ

У монографії розглянуті питання щодо створення сучасних інтелектуальних технологій під час аналізу медичних діагностичних даних на прикладі оцінки результатів функціональної діагностики порушень носового дихання.

Вибір моделі обробки даних тестування носового дихання, які фактично є часовими рядами, потрібно виконувати за критерієм найменшої похибки визначення діагностичних параметрів та отримання найбільшої кількості інформації.

Запропоновано метод задньої активної риноманометрії за форсованого дихання дозволяє знизити ймовірність помилки під час прийняття діагностичного рішення в 1,5 рази порівняно з традиційним методом передньої активної риноманометрії. Додавання до параметрів апаратно-методологічних засобів динамічних характеристик вимірюваних сигналів забезпечує додаткове зниження ймовірності помилки прийняття діагностичного рішення. При цьому більшу, ніж у 1,5 рази діагностичну значущість мають дані перепаду тиску порівняно з величиною витрати повітря.

Також був виконаний аналіз дискримінантних властивостей розробленого методу задньої активної риноманометрії за форсованогому диханні порівняно з традиційним методом передньої активної риноманометрії (при діагностиці порушень носового дихання з викривленням носової перегородки), шляхом визначення Махаланобісової відстані з урахуванням динамічних властивостей досліджуваних сигналів розробленого методу за форсованого дихання, що визначалися за значеннями *F*-статистик дисперсійного розкладу під час розподілі часових рядів вимірювань на три групи по три виміри в кожній.

Проведений аналіз дозволяє зробити висновок, що запропонований у роботі метод риноманометричних вимірювань, який заснований на принципі задньої активної риноманометрії за форсованого дихання, має в 1,7 рази більші дискримінантні властивості порівняно з традиційним, і дозволяє знизити ймовірність помилки (що пов’язана з Махаланобісовою відстанню через інтеграл імовірності Лапласа) під час прийняття діагностичного рішення з 0,36 до 0,21 та використовувати даний метод для функціональної діагностики верхніх дихальних шляхів. Урахування динамічних властивостей (F -статистик дисперсійного перетворювання) часових рядів вимірювальних сигналів, які дозволяють оцінити індивідуальні особливості дихання, забезпечує додаткове істотне (з 0,21 до 0,1) зниження ймовірності помилки прийняття діагностичного рішення. При цьому видно, що значення перепаду тиску має більш важливе (приблизно в 1,5 рази) діагностичне значення порівняно з показниками витрати повітря, а також деякі динамічні властивості вимірювальних сигналів (які не зменшують імовірність похибки) може бути виключено з аналізу в процесі прийняття діагностичного рішення.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Шевчук О.Б. Національна інфраструктура інформатизації // Матеріали конгресу «Інформаційне суспільство – стан, проблеми, перспективи» (25–27 вересня 2000). – Київ, 2000. – С.10-17.

2. Буренин Н.И., Давыдов К.М. Пути развития региональных телекоммуникационных сетей // Проблемы информатизации, 1996.– Вып. 1. – С. 24 – 29.

3. Саймон А.Р. Стратегические технологии баз данных: менеджмент на 2000 год. / Под ред. и с предисл. М.Р. Когаловского. – М.: Финансы и статистика, 1999. – 479 с.

4. Крупнов А.Е. На пути интеграции Российской национальной телекоммуникационно-информационной инфраструктуры в глобальную // Электросвязь. Тр. Межд. Академии связи. – 1998. – № 2(6). – С. 81–84.

5. Концепция формирования информационного общества в России // Информационное общество. – 1999. - № 3. – С. 3–11.

6. Кульба В.В., Ковалевский С.С., Косяченко С.А., Сиротюк В.О. Теоретические основы проектирования оптимальных структур распределенных баз данных. Серия «Информатизация России на пороге XXI века». – М.: СИНТЕГ, 1999.– 660 с.

7. Лазарев В.Г., Пийль Е.И. Интеллектуализация телекоммуникационных сетей // Технологии и средства связи. – 1998. – № 2. – С. 28 – 33.

8. Марти Дж. Планирование развития автоматизированных систем. – М.: Финансы и статистика, 1984. – 376 с.

9. Орлик С.В. Многоуровневые модели в архитектуре клиент–сервер. СУБД. – 1997. – № 1. – С. 74 – 77.

10. Яковлев С.А., Арсеньев Б.П., Ильин В.П. Интеграция распределенных баз данных на основе сетевых технологий. – СПб.: Изд.-полигр.центр СПбГЭТУ (ЛЭТИ), 1998. – 68 с.

11. Бондаренко М.Ф. Основні напрямки створення перспективних інформаційних технологій // Матеріали конгресу «Інформаційне суспільство – стан, проблеми, перспективи» (25–27 вересня 2000). – Київ, 2000. – С. 68–80.

12. Рассел С., Норвиг П. Искусственный интеллект: современный подход. – СПб: Изд– во «Вильямс», 2006. – 1424 с.

13. Гусак П.В. Проектирование ресурсов Windows–приложений. – Киев: Диалектика, 1993. – 276 с.

14. Нейман В.И. Структуры систем распределения информации. – М.: Связь, 1985. – 264 с.

15. Слама Д., Гарбс Г., Перри Р. Корпоративные системы на основе CORBA. – Київ: Диалектика, 2000. – 368 с.

16. Мориссо–Леруа Н., Соломон М., Басу Д. Oracle 8i: Java–компонентное программирование при помощи EJB, CORBA и JSP. – М.: Лори, 2002. – 484 с.

17. Орфали Р., Харки Д. Java и CORBA в приложениях «клиент–сервер». – М.: Лори, 2000. – 734 с.

18. Сигел Д. CORBA–3. – М.: Малин, 2002. – 412 с.

19. Фейбл В. Энциклопедия современных информационных технологий.– К.: Комиздат, 1998. – 688 с.

20. Мамиконов А.Г. Основы построения АСУ. – М.: Высшая школа, 1981. – 248с.

21. Гуд Г.Х., Макол Р.Э. Системотехника. Введение в проектирование больших систем. – М.: Советское радио, 1972. – 383 с.

22. Бусленко Н. П. Моделирование сложных систем. – М.: Наука, 1978. – 399 с.

23. Татанов И.В., Авраменко В.П., Тимофеев В.А., Панасенко А.А. Моделирование организационно–технологических систем. – Рязань: Рус. слово, 1996. – 224 с.

24. Ларіонов Ю.І., Левикін В.М., Хажмурадов М.А. Дослідження операцій в інформаційних системах. – Харків: СМІТ, 2005. – 364 с.

25. Watson H.J., Houdeshel G., Rainer R.K. Bulding exelutive information systems and other decision support applications – New York: John Wiley & Sons Inc., 1997. – 479 p.

26. Ситник В.Ф., Дубровіна А.В. Проблеми моделювання рішень у групових СППР // Моделювання та інформаційні системи в економіці. – 2002. – Вип. 68. – С. 9 – 14.

27. Kock N.F. The effects of asynchronous groupware on business process improvement // PhD thesis University of Waikato. – Hamilton, New Zealand, 1997. – 415 p.

28 Архитектуры, модели и технологии программного обеспечения информационно-управляющих систем / Ткачук Н.В., Шеховцов В.А., Кукленко Д.В. и др. // Под ред. М.Д. Годлевского. – Харьков: НТУ «ХПИ», 2005. – 546 с.

29. Уланов Г.М., Алиев Р.А., Кривошеев В.П. Методы разработки интегрированных АСУ промышленными предприятиями. – М.: Энергоатомиздат, 1983. – 320 с.

30. Системное проектирование интегрированных производственных комплексов / А.Н. Домарацкий, А.А., Лескин, В.М. Пономарев и др. // Под общ. ред. д.т.н., проф. В.М. Пономарева. – Л.: Машиностроение. Ленингр. Отд-ние, 1986. – 319 с.

31. Математическое моделирование // Под ред. Дж. Эндрюс, Р. Мак-Лоун. – М.: Мир, 1979. – 277 с.

32. Zadeh L. A. A computational approach to fuzzy quantifiers in natural languages // Computer and Mathematics. – 1983. – 9. – P. 149 – 184.

33. Шаталов А.С. Отображение процессов управления в пространствах состояний. – М.: Энергоатомиздат, 1986. – 256 с.

34. Вейцман К. Распределенные системы мини- и микро-ЭВМ // Под ред. Васильева Г.П. – М.: Финансы и статистика, 1983. – 362 с.

35. Авраменко В.П. Управление производством в условиях неопределенности. – Киев: УМК ВО, 1992. – 48 с.

36. Снитюк В.Е. Композиционное преодоление неопределенности в задачах нелинейной многофакторной оптимизации // Штучний інтелект. – 2004. – №4. – С. 207 – 210.

37. Танака К. Итоги рассмотрения факторов неопределенности и неясности в инженерном искусстве // Нечеткие множества и теория возможностей. Последние достижения / Под ред. Р.Р. Ягера. – М.: Радио и связь, 1986. – С. 37 – 50.

38. Саридис Дж. Самоорганизующиеся стохастические системы управления. – М.: Наука, 1980. – 400 с.

39. Трухаев Р.И. Модели принятия решений в условиях неопределенности. – М.: Наука, 1981. – 258 с.

40. Ситник В.Ф. Системи підтримки прийняття рішень. – К.: КНЕУ, 2004. – 614 с.

41. Invenko M., Luck M. Understanding Agent Systems. – Berlin: Springer, 2003.– 191 p.

42. Пономаренко Л.А., Филатов В.А. Електронна комерція / За ред. А.А. Мазаракі.– К.: Київ.нац.торг.-екон. ун-т, 2002. − 443 с.

43. Subrahmanian V.S. Heterogeneos Agent Systems. – Cambridge, Massachusetts. – London: The MIT Press, 2000. – 580 p.

44. http://www.lumina.com.

45. http://www.hps–inc.com.

46. http://www.adainc.com/sw/index/htnl.

47. http://www.expertchoice.com.

48. Bovik H.Q., Goldsmith J.Q., Klapper A.Q., Littman M.Q. Markov indecision processes: a formal model of decision–making under extreme confusion // Journal of Machine Learning Gossip. – 2003. – 1. – P. 1– 9.

49. Bovik H.Q. Frenetic algorithms: Students as random program generators // Journal of Artificial Research. – 1990. – P. 31 – 90.

50. Goldsmith J., Sloan R.H. The complexity of model aggregation // Proceedings of the Fifth International Conference on Artifcial Intelligence Planning Systems (AIPS). – 2000. – P. 122 – 129.

51. Kohda T., Tsuneda A. Statistics of chaotic sequences // IEEE Trans. on Information Theory. – 1997. – 43 (1). – P. 104 –112.

52. Littman M.L., Szepesvari C. A. generalized reinforcement-learning model: Convergence and applications // Proceedings of the Thirteenth International Conference on Machine Learning. – 1996. – P. 310 – 318.

53. Цвиркун А.Д., Акинфиев В.К., Филиппов В.А. Имитационное моделирование в задачах синтеза структуры сложных систем. – М.: Наука, 1985. – 174 с.

54. Наумов Б.Н., Кескер Э.Я., Левин Н.А. Алгоритмы оптимизации и автоматизации проектирования АСУ. – М.: Энергоатомиздат, 1983. – 160 с.

55. Автоматизация проектирования АСУ с использованием пакетов прикладных программ // Ю.М. Черкасов и др. – М.: Энергоатомиздат, 1986. – 328 с.

56. Соммервил И. Инженерия програмного обеспечения. – М.: Изд. дом «Вильямс», 2002. – 624 с.

57. Hruschka P., Rupp C. Agile Softwareentwicklung fuer Embedded Real-Time Systems mit der UML. – Muenchen: Hanser Verlag, 2002. – 192 S.

58. Foegen M., Batterfeld J. Die Rolle der Architektur in der Anwendungsentwicklung // Informatik-Spektrum. – Springer, 2001. – № 5 (24). – S. 290–301.

59. Дмитриев А.К., Мальцев П.А. Основы теории построения и контроля сложных систем. – Л.: Энергоатомиздат. Ленингр. отд-ние, 1988. – 192 с.

60. Сарычев А.П. Классификация объектов наблюдений, описываемых системами регрессионных уравнений с детерминированными коэффициентами // Штучний інтелект. – 2005. – № 3. – С. 43 – 56.

61. Scholl R.W. Decision Making Models // University of Rhode Island Revised. – 1999. – www.cba.uri.edu.

62. Todd P.M., Gigerenzer G. Putting naturalistik decision making into the adaptive toolbox // Journal of Behavioral Decision Making. – 2001. – 14. – P. 353–384.

63. Krantz D.H. Probability models and human decision–making. – Columbia, 1999. – www.earthinstitute.columbia.edu.

64. Lipshitz R., Klein G., Orasanu J., Salas E. Theories of risk and decision making. Taking stock of naturalistic decision making: MIT Laboratory for Financial Engineering. // Journal of Behavioral Decision Making. – 2001. – 14 (5). – P. 331–352.

65. Sitkin S.B., Pablo A.L. Reconceptualizing the determinants of risk behavior // Academy of Management Review. – 1992. – 17(1). – P. 9 – 38.

66. Busemeyer J. Last modified: Decision processes brown bag. – 2004. – www.facilities.upenn.edu .

67. Иващенко П.А. Адаптация в экономике. – Харьков: Вища школа, 1986. – 144 с.

68. Тихонов А.Н., Арсенин В.Я. Методы решения некорректных задач. – М.: Наука, 1986. – 288 с.

69. Алтунин А.Е., Семухин М.В. Модели и алгоритмы принятия решений в нечетких условиях. – Тюмень: ТГУ, 2000. – 352 с.

70. Мамедова М.Г., Джабраилова З.Г. Нечеткая логика в прогнозировании демографических аспектов рынка труда // Искусственный интеллект. – 2005. – № 3. – С. 450 – 460.

71. Боженюк А.В., Гинис Л.А. О нахождении нечетких путей и компонент сильной связности между слоями иерархических познавательных карт // Искусственный интеллект. – 2005. – № 3. – С.336 – 347.

72. Алтунин А.Е., Митюшкин Ю.И., Мокин Б.И., Ротштейн А.П. Soft Computing: идентификация закономерностей нечеткими базами знаний. – Вінниця: Універсум–Вінниця, 2002. – 145 с.

73. Fuzzy Optimization and Decision Making. – www.springerlink.com/openurl.

74. Sousa J.M., Kaymak U. Fuzzy decision making in modeling and control // World Scientific Series in Robotics and Intelligent Systems. – http://www.worldscibooks.com/comps]

75. International Journal of Approximate Reasoning. – www.informatik.uni-trier.de/ ~ley/db/journals/ijar/ijar2.html].

76. International Journal of Uncertainty, Fuzziness and Knowledge–Based Systems. – http://www.informatik.uni-trier.de/~ley/db/journals/ijufks/ijufks9.html.

77. Yu M.-M., Yao J.-S. Modification of concordance analysis based on statistical data and ranking of level 1 – a fuzzy numbers // International Journal of Uncertainty. – 2001. –9. – N 3. – P. 313 –340.

78. Kacprzyk J., Zadrozny S. Computing with words in decision making through individual and collective linguistic choice rules // International Journal of Uncertainty. – 2001. – 9. – Supplement. – P. 89 – 102.

79. Chen T. C. Applying linguistic decision–making method to deal with service quality evaluation problems//International Journal of Uncertainty. – 2001. – 9. – Supplement. – P. 103–114.

80. Nwana H.S. Software agents: an overview // Knowledge Engineering Review / Cambridge University Press. – 1996. – 11. – No 3. – P. 1– 40.

81. Maes P. Artificial life meets entertainment: life like autonomous agents // Communications of the ACM. – 1995. – 11. – P. 108–114.

82. Smith D. C., Cypher A., Spohrer J. KidSim: programming agents without a programming language // Communications of the ACM. – 1994. – 37. – P. 55 – 67.

83. Hayes–Roth B. An architecture for adaptive intelligent systems, artificial intelligence: Special issue on agents and interactivity. – 1995. – 72.– P. 329–365.

84. http://activist.gpl.ibm.com:81/WhitePaper/ptc2.htm

85. Wooldridge M., Jennings N.R. Agent theories, architectures, and languages: a survey /in r. Wooldridge and L.E. Jennings (Eds.), Intelligent Agents: − Berlin: Springer-Verlag, 1995.–P. 1 – 22.

86. Labrou Y., Finin T. A semantics approach for KQML – a general purpose communication language for software agents // Third International Conference on Information and Knowledge Management (CIKM'94), November 1994. – 1994. – P. 354 – 378.

87. Brustoloni J.C. Autonomous agents: characterization and requirements //Carnegie Mellon Technical Report CMU–CS–91–204. – Pittsburgh: Carnegie Mellon University, 1999. –P. 122–221.

88. Nwana H. S., Lee L., Jennings N. R. Coordination in software agent systems // British Telecommunications Technology Journal. – 1996. – 14 (4), – P. 21–42.

89. Jennings N.R., Sycara K., Wooldridge M. A Roadmap of agent research and development // Autonomous Agents and Multi–Agent Systems Journal. – Boston: Kluwer Academic Publishers, 1998. – 1, Is. 1. – P. 7–38.

90. Shoham Y. An overview of agent–oriented programming, software agents/ Bradshaw J.M. (ed.). – AAAI Press, Menio Park, CA, USA, 1997. –310 p.

91. Franklin S., Graesser A. Is it an agent, or just a program?: A taxonomy for autonomous agents, In Intelligent Agents III – Agent Theories, Architectures, and Langues / M. Muller (Ed.). – Berlin: Springer, 1996. – P. 11–46.

92. Dean T., Alien J., Aloiffionos Y. Artificial intelligence – theory and practice. – Benjamin: Cummings, 1995. – 273 p.

93. Krogh C. The rights of agents, in intelligent agents II – agent theories, architectures, and langues / M. Muller (Ed.) – Berlin: Springer, 1995. – 279 p.

94. Huhns M.N., Singh M.P. Readings in agents. – Morgan Kaufmann Pub., 1997. – 311 p.

95. Genesereth M. R., Ketchpel S. P. Software agents, communications of the ACM37 (7). – 1994. – P. 48–53.

96. Brooks R. A. A robust layered control system for a mobile robot // IEEE Journal of Robotics and Automation 2(1). – 1986. – P. 14–23.

97. Agre P. E., Chapman D. Pengi: an implementation of a theory of activity // Proceedings of the 6th National Conference on Artificial Intelligence, San Mateo, CA: Morgan Kaufmann. – 1987. – P. 268–272.

98. Ferber J. Simulating with reactive agents / Hillebrand E., Stender J. (Eds.), Many Agent Simulation and Artificial Life: Amsterdam: IOS Press. – 1994. – P. 8–28.

99. Huhns M. N., Singh M. P. Distributed artificial intelligence for information systems. – 1994. – P. 97–104.

100. Foner L. What's an agent, anyway? A sociological case study / Agents Memo 93–01, MIT Media Lab, Cambridge, MA. – 1993. – P. 27–74.

101. Bradshaw J.M., Outfield S., Benoit P., Woolley J.D. KAoS; toward an industrial–strength open agent architecture // Software Agents. – 1997. – P. 375–418.

102. Уотермен Д. Руководство по экспертным системам. – М.: Мир, 1989. – 388 с.

103. Maes P. Designing autonomous agents: theory and practice from biology to engineering and back. – London: The MIT Press. – 1991.– 221 p.

104. Fisher K., Muller J. P., Pischel M. Unifying control in a layered agent architecture/ Technical report TM-94-05, German Research Center for AI – (DFKI GmbH). – 1996.–377 p.

105. Huhns M.N., Singh M.P. Agents and multi–agent systems: themes, approaches, and challenges / Readings in Agents, Huhns M.N., Singh, M.P. (Eds.). – San Francisco, Calif.: Morgan Kaufmann Publishers, 1998. – P. 1 – 23.

106. Mayfield J., Labrou Y., Finin T. Evaluation of KQML as an agent communication language, intelligent agents / Volume II – Proceedings of the 1995 Workshop on Agent Theories, Architectures, and Languages // Lecture Notes in Artificial Intelligence, M. Wooldridge, J. P. Mutter, M. Tambe (Eds.). – Berlin, Springer–Verlag, 1996. – P. 347–360.

107. Peng Y., Finin T., Labrou Y., Chu B., Long J., Tolone W.J., Boughannam A. A multi–agent system for enterprise integration / Proceedings of the Third International Conference and Exhibition on the Practical Application of Intelligent Agents and Multi-Agent Technology, H.S. Nwana, D.T. Ndumu (Eds.). – London, UK, March, 1998. – P. 155–169.

108. Shen W., Barthes J.P. An experimental environment for exchanging engineering design knowledge by cognitive agents / Mantyla M., Finger S., Tomiyama T., (Eds.). – Chapman and Hall: Knowledge Intensive CAD–2, 1997. – P. 19–38.

109. Singh M.P. An ontology for commitments in multiagent systems // Artificial Intelligence and Law. – 1999. – 7. – P. 97–113.

110. Weib G. Adaptation and learning in multi–agent systems / G. Weip, S. Sen (Eds). Adaptation and Learning in Multi-Agent Systems // Lecture Notes in Artificial Intelligence. – Berlin: Springer-Verlag, 1996. – P. 1–21.

111. Танака К. Итоги рассмотрения факторов неопределенности и неясности в инженерном искусстве // Нечеткие множества и теория возможностей. Последние достижения. / Под ред. Р.Р. Ягера. – М.: Радио и связь, 1986. – С. 37 – 50.

112. Сейдж Э.П., Уайт Ч.С., III. Оптимальное управление системами. – М.: Радио и связь, 1982. – 392 с.

113. Zadeh L.A. Fuzzy sets and systems // Proc. Symp. Syst. Theory Polytech. Inst. – Brooklyn, 1965. – P. 29 – 37.

114. Bellman R.E., Zadeh L.A. Decision making in a fuzzy environment // Management Sci. – 1970. – 17. – P. 141 – 164.

115. Заде Л.А. Понятие лингвистической переменной и его применение к принятию приближенных решений. – М.: Мир, 1976. – 165 с.

116. Поспелов Д.А. Логико-лингвистические модели в системах управления. – М.: Энергоатомиздат, 1981. – 232 с.

117. Охрана и оптимизация окружающей среды / Под ред. Лаптева А.А. – К.: Либідь, 1990. – 256 с.

118. Гельман Г.А. Автоматизированные системы управления энергоснабжением промышленных предприятий. – М.: Энергоатомиздат, 1984. – 256 с.

119. Hirasawa K., Ohbayashi M., Sakai S., Hu J. Learning Petri network and its application to nonlinear system control // IEEE Trans. on Systems, Man, and Cybernetics – Part B. – Cybernetics. – 1998. – 28. – № 6. – P. 781 – 789.

120. Моделирование развивающихся систем / В.М. Глушков, В.М. Иванов, В.М. Яненко. – М.: Наука, 1987. – 350 с.

121. Алиев Р.А., Абинеев Н.М., Шахназаров М.М. Производственные системы с искусственным интеллектом. – М.: Радио и связь, 1990. – 265 с.

122. Международные стандарты. Управление качеством продукции. ИСО 9000-ИСО 9004, ИСО 8402. – М.: Изд-во стандартов, 1988. – 95 с.

123. Кофман А. Введение в теорию нечетких множеств. – М.: Радио и связь, 1982. – 432 с.

124. Обработка нечеткой информации в системах принятия решений. / А.Н. Борисов, А. В. Алексеев, Г. В. Меркульева и др. – М.: Радио и связь, 1989. – 304 с.

125. Норвич А.М., Туркcен И.Б. Построение функций принадлежности // Нечеткие множества и теория возможностей. Последние достижения / Под ред. Р.Р. Ягера. – М.: Радио и связь, 1986. – С. 64 – 71.

126. Ягер Р.Р. Множества уровня для оценки принадлежности нечетких подмножеств // Нечеткие множества и теория возможностей. Последние достижения / Под ред. Р.Р. Ягера.– М.: Радио и связь, 1986. – С. 71 – 78.

127. Норвич А.М., Турксен И.Б. Фундаментальное измерение нечеткости // Нечеткие множества и теория возможностей. Последние достижения / Под ред. Р.Р. Ягера. – М.: Радио и связь, 1986. – С. 54 – 64.

128. Миркин Б.Г. Проблема группового выбора. – М.: Наука. Гл. ред. физ.-мат. лит., 1974.– 256 с.

129. Looney C. Fuzzy Petri nets for rule-based decision making // IEEE Trans.on Systems, Man, and Cybernetics. – 1988. –18. – № 1. – P. 178 – 183.

130. Zadeh L. Similarity relations and fuzzy orderings // Information Sciences. – Amsterdam: Elsevier, 1971. – 3. – P. 177 – 200.

131. Tsoukalas L.H., Uhrig R.E. Fuzzy and Neural Approaches in Engineering. – New York: John Wiley&Sons.Inc, 1997. – 587 p.

132. Zadeh L.A. The concept of a linguistic variable and its applications to approximate reasoning – Part III // Information Sciences. – 1976. – 9. – P. 43 – 80.

133. Isik C. Inference hardware for fuzzy rule-based systems // Fuzzy Computing Theory, Hardware and Applications: M.M. Gupta, T. Yamakama (Eds.) – North-Holland, Amsterdam: Elsevier, 1988. – P. 185 – 194.

134. Yamakama T. Intrinsic fuzzy electronic circuits for sixth generation computers // Fuzzy Computing Theory, Hardware and Applications: M.M. Gupta, T. Yamakama (Eds.) – North - Holland, Amsterdam: Elsevier, 1988. – P. 157 – 173.

135. Zadeh L.A. Outline of a new approach to the analysis of complex systems and decision processes // IEEE Transactions on Systems, Man and Cybernetics. – 1973. – 1. – P. 28 – 44.

136. Mamdani E.H. Applications of fuzzy set theory to control system: A survey fuzzy automata and decision processes / M.M. Gupta, G.H. Saridis and B.R. Gaines (Eds.) – New York: North – Holland, 1977. – P. 1 – 13.

137. Zadeh L.A. The concept of a linguistic variable and its application to approximate reasoning // Information Sciences. – 1975. – 8.– P. 199 – 249.

138. Larsen P.M. Industrial applications of fuzzy logic control // International Journal of Man-Machine Studies. – 1980. – 12. – 1. – P. 3 – 10.

139. Mizumoto M. Fuzzy controls under various reasoning methods // Information Sciences. – 1988. – 45. – P. 129 – 141.

140. Lee C.C. Fuzzy logic in control systems: fuzzy logic controller – Part 1 // IEEE Trans. on Systems, Man, and Cybernetics. – 1990. – 20. – 2. – P. 404 –418.

141. Lee C.C. Fuzzy logic in control systems: fuzzy logic controller – Part 2 // IEEE Trans. on Systems, Man, and Cybernetics. – 1990. – 20. – 2 – P. 419 –435.

142. Dubois D., Prade H., Ughetto L. Checking the coherence and redundancy of fuzzy knowledge bases // IEEE Trans. on Fuzzy Systems. – 1997. – 5. – № 3. – P. 398 – 417.

143. Сетлак Г. Интеллектуальная система поддержки решений в нечеткой среде // Искусственный интеллект. – 2002. – № 3. – С. 428 – 438.

144. Сироджа И.Б. Квантовые модели и методы инженерии знаний в задачах искусственного интеллекта // Искусственный интеллект. – 2002. – № 3. – С. 161 – 171.

145. Варламов О.О. Разработка адаптивного механизма логического вывода на эволюционной интерактивной сети гиперправил с

мультианализаторами, управляемой потоком данных // Искусственный интеллект. – 2002. – № 3. – С. 363 – 370.

146. Ринкс Д.Б. Эвристический подход к обобщенному календарному планированию производства с использованием лингвистических переменных: методология и применение // Нечеткие множества и теория возможностей. Последние достижения. / Под ред. Р.Р. Ягера. – М.: Радио и связь, 1986. – С. 349 – 370.

147. Yager R.R., Filev D.P. A simple adaptive defuzzification method // IEEE Trans. on Fuzzy Systems. –1993. – 1. – №. 1. – P. 69 – 78.

148. Pedrycz W. Fuzzy models: methodology, design, applications and challenges // Fuzzy Modelling Paradigms and Practice / Ed. by W. Pedrycz. – Boston - Dortrecht - London, 1996. – P. 3 – 22.

149. Pedrycz W. Selected issues of frame of knowledge representation realized dy means of linguistic labels // Int. J. of Intelligent Systems. – 1992. – 7. – P. 155 – 170.

150. Zadeh L. A. Fuzzy sets and information granularity // Advances in Fuzzy Set theory and Applications / M.M. Gupta, R.K. Ragade, R.R. Yager (Eds.) – Amsterdam: North Holland, 1979. – P. 3 – 18.

151. Pedrycz W. Fuzzy neural networks and neurocomputations // Fuzzy Sets and Systems. – 1993. – 56. – P.1 – 28.

152. Искусственный интеллект: – В 3–х кн. Кн. 2. Модели и методы. Справочник / Под ред. Д.А. Поспелова. – М.: Радио и связь, 1990. – 304 с.

153. Боженюк А.В., Гинис Л.А. Об использовании нечетких баз и антибаз при анализе нечетких когнитивных карт // Штучний інтелект. – 2004. – № 4. – С. 276 – 285.

154. Робототехника и гибкие автоматизированные производства. В 9 кн. Кн.6. Техническая имитация интеллекта / В.М. Назаретов, Д.П. Ким; Под ред. И.М. Макарова. – М.: Высш. шк., 1986. – 144 с.

155. Pedrycz W. Processing in relational structures: fuzzy relational equations // Fuzzy Sets and Systems. – 1990. – 40. – P. 77 – 106.

156. Pedrycz W. Neurocomputations in relational systems // IEEE Trans. on Pattern Analysis and Machine Intelligence. – 1991. – 13. – P. 155 – 170.

157. Santos E.S. Context– free fuzzy languages // Information and Control. – 1974. – 26. – P. 1 – 11.

158. Эрнст К.Дж. Один подход к экспертным системам управления с использованием нечеткой логики // Нечеткие множества и теория возможностей. Последние достижения / Под ред. Р.Р. Ягера. – М.: Радио и связь, 1986. – С. 133 – 143.

159. Мукаидоно М. Нечеткий вывод резолюционного типа // Нечеткие множества и теория возможностей. Последние достижения / Под ред. Р.Р. Ягера. – М.: Радио и связь, 1986. – С. 153 – 161.

160. Мидзумото М. Нечеткое рассуждение с нечетким условным высказыванием вида «если…то…иначе» // Нечеткие множества и теория возможностей. Последние достижения / Под ред. Р.Р. Ягера.– М.: Радио и связь, 1986. – С. 143 – 153.

161. Трильяс Э., Альсина К., Вальверде А. Нужны ли в теории нечетких множеств операции // Нечеткие множества и теория возможностей. Последние достижения / Под ред. Р.Р. Ягера. – М.: Радио и связь, 1986. – С. 199 – 228.

162. Yager P.P., Larsen H.L. Retrieving information by fuzzification of queries // Journal of Intelligent Information Systems. – 1993.– 2.– № 4.– P. 421 – 441.

163. Takagi T., Sugeno M. Fuzzy identification of systems and its application to modeling and control // IEEE Trans. on Systems, Man, and Cybernetics. – 1985. – 15. – P. 116 – 132.

164. Sugeno M., Yasukawa T. A fuzzy–logic–based approarch to qualitative modeling // IEEE Trans. on Fuzzy Systems. – 1993. – 1. – P. 7 – 31.

165. Sugeno M. An introductory survey of fuzzy control // Information Sciences. – 1985. – 36. – P.59 – 83.

166. Sugeno M., Park G.-K. An approach to linguistic instruction based learning // Intern. J. of Uncertainty, Fussiness and Knowledge-Based Systems. – 1993. – 1. – № 1. – P. 19 – 56.

167. Уоссермен Ф. Нейрокомпьютерная техника. – М.: Мир, 1992. – 210с.

168. Бодянський Є.В., Кучеренко Є.І. Нейро-фаззі моделі в системах штучного інтелекту. – Харків: ХНУРЕ, 2006. – 177 с.

169. Chen C.– L., Chen W. – C. Fuzzy controller design by using neural network techniques // IEEE Trans. on Fuzzy Systems. – 1994. – 2. – № 3. – P. 235–244.

170. Hiraga I., Furuhashi T., Uchikawa Y., Nakayama S. An acquisition of operator's rules for collision avoidance using fuzzy neural networks // IEEE Trans. on Fuzzy Systems. – 1995. – 3. – № 3. – P. 280 – 287.

171. Jang J.S.R. ANFIS: Adaptive–Network–Based Fuzzy Inference System // IEEE Trans. on Systems, Man and Cybernetics. – 1993. – 23. – № 3. – P. 665 – 685.

172. Bikbulatov A., Batyrshin I. Tuning of operations in fuzzy models by neural nets // Proceedings of 7th Zittau Fuzzy Colloquium. – Zittau, Germany,1999. – P. 142 – 147.

173. Lipp H. – P. Einsatz von zeitbewerteten Fuzzy – Petri-Netzen in Expertensystemen zur operativen Fuehrung komplexer Productionsysteme / Hommel, G. (Hrsg.): Prozesrechungssysteme '91. – Berlin – Heidelberg – New York: Springer – Verlag, 1991. – S. 103 – 112.

174. Lipp H. – P. Ein Fuzzy – Petri - Netz – Koncept fuer komplexe Entscheidungsprozesse in Produktionsteuerungen / Sheschonk, G. (Hrsg.): Petri–Netze in Einsatz fuer Entwurf und Entwicklung von Informationssystemen. – Berlin – Heidelberg– New York: Springer – Verlag, 1993. – S. 232 – 245.

175. Murata T. Temporal uncertainty and fuzzy–timing high–level Petri nets // Proc. 17th Int. Conf. of Application and Theory of PNs, Osaka, Japan, June 26/ Los Alamitos, CA: IEEE Computer Society Press. – 1996.– P. 11–28.

176. Murata T., Susuki T, Shatz S.M. Fuzzy–timing high–level Petri nets model of a real–time network protocol // Proc. ITC–CSCC 96, Seoul, Korea, July 15 – 17/ Los Alamitos, CA: Computer Society Press. – 1996. – P. 1170 –1173.

177. Murata T., Susuki T, Shatz S.M. Fuzzy–timing high–level Petri nets (FTHNs) for time critical systems / J. Caroso, H. Camango (editors) «Fuzziness in

Peti nets» in the series «Studies in Fuzziness and Soft Computing». – New York: Springer Verlag, 1999. – 22. – P. 88 – 114.

178. Zhou Y., Murata T., DeFanti T. Modeling and analysis of a collaborative virtual envitonment by using extended fuzzy– timig Petri – nets // Proc. of Workshop on Applications of Petri Nets to Intelligent System Development, PN 1999, June 22. – Williamsburg, 1999. – P. 26 – 37.

179. Zhou Y., Murata T. Petri net model with fuzzy–timing and fuzzy – metric temporal logic // International Journal of Intelligent Systems. The special issue on fuzzy Petri nets; concepts and intelligent system modeling. – 1999. –14. – № 8. – P. 719 – 746.

180. Watanuki K., Murata T. Fuzzy – Timing Petri net model of temperature control for car air conditioning system // Proc. of 1999 IEEE International Conference on Systems, Man, and Cybernetics, October 12 – 15. – Tokyo, 1999. – 4.– P. 618–622.

181. Zhou Y., Murata T. Fuzzy–timing Petri net model for distributed multimedia synchronization // Proc. of the 1998 IEEE Conference on Systems, Man and Cybernetics, October 11 – 14. – Lolla, California, 1998. – P. 244 – 249.

182. Ashon S.I. Petri net models of fuzzy neural network // IEEE Trans. on Syst., Man, and Cybernetics. – 1995. – 25. – P. 926 – 932.

183. Garg L., Ashon S.I., Gupta P.V. A fuzzy Petri net for knowledge representation and reasoning // Inform. Proc. Lett. – 1991. – 39. – P. 165 – 171.

184. Chen S. – M., Ke J. – S., Chang J. – F. Knowledge representation using fuzzy Petri nets // IEEE Trans. on Knowledge Data Ingeneering. – 1990. – 2. – P. 311 – 319.

185. Koriem S.M. A fuzzy Petri net tool for modeling and verification of knowledge – based systems // The Computer Journal. –2000. – 43. – № 3. – P. 206 – 223.

186. 20th International Conference on Applications and Theory of Petri Nets. – Williamsburg, VA, USA. – http:/ www.cs.wm.edu./ps99/registration.html.

187. Computer Engineering Group Department of Electrical and Electronic Engineering the University for Melbourne Research Seminar: «Synthesis of Fuzzy Petri Nets». – 1996.– http://www.ee.mu.oz.au./seminars/Cheng.html.

188. Гожальчаны М.Б., Кишка Е.Б., Стахович М.С. Некоторые проблемы изучения адекватности нечетких моделей // Нечеткие множества и теория возможностей. Последние достижения / Под ред. Р.Р. Ягера. – М.: Радио и связь, 1986. – С. 21 – 37.

189. Кучеренко Е.И. Проблемы моделирования и анализа нечетких процессов управления // Радиоэлектроника и информатика. – 2001. – № 2. – С. 118 – 121.

190. Руководство по ринологии / Под ред. Г.З. Пискунова. – М.: Литтерра, 2011. – 960 с.

191. Бабияк В. И. Клиническая оториноларингология. Руководство для врачей / В.И. Бабияк, Я. А. Накатис. – СПб. : Гиппократ. – 2005. – 800 с.

192. Гапанович В.Я. Оториноларингологический атлас / В.Я. Гапанович, В.М. Александров. – Минск: Выш. шк., 1989. – 239 с.

193. Оториноларингологія / Д.І. Заболотний, Ю.В. Мітін, С.Б. Безшапочний, Ю.В. Деєва Ю.В. / Підручник для студентів вищих медичних навчальних закладів IV рівня акредитації. – Київ: Медицина. – 2010. – 472 с.

194. Применение общей плетизмографии для оценки проходимости носа у больных вазомоторным ринитом / М.С. Плужников, П.П. Горбенко, А.Н. Александров, Н.А. Зильбер // ЖУНГБ. – 1987. – №3. – С.10–13.

195. Блоцкий А. А. Феномен храпа и синдром обструктивного сонного апноэ / А. А. Блоцкий, М. С. Плужников. – СПб.: СпецЛит. – 2002. – 176 с.

196. Correlation between subjective assessment and objective measurement of nasal obstruction/ G. Zhang, R. Fenton, R. Rival et al. // Zhonghua. – 2008. – №43(7). – P. 484–489.

197. Study and application of a mathematical model for the provisional assessment of areas and nasal resistance, obtained using acoustic rhinometry and active anterior rhinomanometry/ G. Zambetti, M. Moresi, R. Romeo, F. Filiaci // Clin. Otolaryngol. Allied Sci. – 2001. – №26 (4). – 286–293.

198. Szucs E. Acoustic rhinometry and rhinomanometry in the evaluation of nasal patency of patients with nasal septal deviation / E. Szucs, P. Clement // Am. J. Rhinol. – 1998. – №12 (5). – P. 345–352.

199. Ульянов Ю.П. Септопластика под контролем аэродинамики носа / Ю.П. Ульянов // Врач. – 2000. – № 6. – С. 28–31.

200. Юнусов А. С. Передняя активная риноманометрия при деформации перегородки носа у детей старшей возрастной группы / А С. Юнусов, О.И. Попова // Российская ринология. – 2009. –№ 2. – С. 118.

201. Садыхов Ф.А. Компьютерная риноманометрия в выборе оптимального метода лечения хронических ринитов / Ф.А. Садыхов // Материалы итоговой конференции военно-научного общества слушателей и ординаторов I факультета. – СПб.: ВМедА. – 2006. – С. 170.

202. Говорун М.И. Компьютерная риноманометрия как инструмент системы качества медицинской помощи в ринологии / М.И Говорун, Ф.А. Садыхов // Журнал вушних, носових і горловых хвороб. – 2006. – № 3с. – С. 198–199.

203. Говорун М.И. Основные принципы оценки качества хирургического вмешательства в полости носа / М.И. Говорун, К.В. Герасимов, Ф.А. Садыхов // Журнал вушних, носових і горловых хвороб. – 2006. – № 5. – С. 97.

204. Говорун М.И. Показатели компьютерной риноманометрии как основа определения объема оперативного лечения патологии полости носа / М.И. Говорун, Ф.А. Садыхов// Актуальные проблемы современной оториноларингологии: Материалы Всеармейской научно-практической конференции посвященной 130-летию со дня рождения В.И. Воячека. – СПб.: ВМедА. – 2006. – С. 88–89.

205. Державина Л.Л. Оценка функциональных результатов микроэндоскопических эндоназальных операций методами акустической ринометрии и риноманометрии / Л.Л. Державина, А.А. Шиленков// Рос. ринология. – 1998. – №2. – С. 66.

206. Grymer L. F. Reduction rhinoplasty and nasal patency: change in the cross–sectional area of the nose evaluated by acoustic rhinometry / L. F. Grymer // Laryngoscope. – 1995. – № 105. – P. 429–431.

207. Acoustic rhinometry: Evaluation of nasal cavity geometry by acoustic reflection/ O. Hilberg, A.C. Jackson, D.L. Swift, O.F. Pedersen // J. Appl. Physiol. – 1989. – Vol. 66. – P. 295–303.

208. Cole P. Contemporary rhinomanometry / P. Cole, R. Fenton // J Otolaryngol. – 2006. – № 35(2). – P. – 83–87.

209. Cole P. Rhinomanometry 1988: practice and trends / P. Cole // Laryngoscope. – 1989. – № 99 (3). – P. 311–315.

210. Computer averaged nasal resistance / K. Naito, P. Cole, R. Chaban, D. Humphrey // Rhinology. – 1989. – № 27(1). – P. 45–52.

211. A fundamental study of rhinomanometry and its clinical application to objective evaluation / K. Naito, S. Iwata, M. Kondo et al. // Auris Nasus Larynx. – 1989. – № 16(2). – P. 99–108.

212. Cole P. Anterior and posterior rhinomanometry / P. Cole, A. Ayiomanimitis, M. Ohki et al. // Rhinology. – 1989. – № 27(4). – P. 257–62.

213. Naito K. Unilateral and bilateral nasal resistances: a supplement / K. Naito, P. Cole, D. Humphrey // Rhinology. – 1990. – № 28(2). – P. 91–95.

214. Avrunin, O.G., Sakalo, S.N., Semenetc, V.V. Development of up-to-date laboratory base for microprocessor systems investigation // O.G. Avrunin, S.N. Sakalo, V.V.Semenetc // International Crimean Conference Microwave and Telecommunication Technology, Conference Proceedings KpbiMuKo. – 2009. – P. 301–302.

215. Технология межсоединений электронной аппаратуры: учеб. для вузов / В.В Семенец, Д. Кратц, И.Ш. Невлюдов, В.А. Палагин, Харків: ООО «Компания СМИТ». – 2005. – 432 с.

216. Семенець В. Впровадження технологій дистанційного навчання у навчальний процес / В.Семенець, В.Каук, О.Аврунін // Вища школа. – № 5. – 2009. – С.40 – 57.

217. About One Method of Mathematical Modelling of Human Vision Functions // V. Semenets, Yu. Natalukha, O. Taranukha, V. Tokarev // ECONTECHMOD. An international quarterly journal. – 2014. – Vol. 3. – No. 3.– P. 51–59.

218. Semenets V.V. Coordinate method for estimation of radial velocity in systems of acoustic sounding of the atmosphere/ V.V. Semenets, V.I. Leonidov// Telecommunications and Radio Engineering (English translation of Elektrosvyaz and Radiotekhnika). – 2017. – № 76(3). – P. 245–251

219. Semenets V.V. Analysis of electromagnetic environment and modeling of spurious radiation sources/V.V. Semenets, T.E. Stytcenko // Telecommunications and Radio Engineering (English translation of Elektrosvyaz and Radiotekhnika). –2016. – № 75(15).– P. 1385–1396

220. Аврунин О.Г. Сравнение дискриминантных характеристик риноманометрических методов диагностики / О.Г. Аврунин, В.В. Семенец, П.Ф. Щапов // Радіотехніка. – 2011. – 164. – С. 102–107.

221. Покровский В.М. Физиология человека / (В.М. Покровский, Г.Ф. Коротько); под. ред. В. М. Покровского и Г.Ф. Коротько // Серия: Учебная литература для студентов медицинских вузов. – М. : Медицина, 2007. – 656 с.

222. Гриппи М. Патофизиология легких / М. Гриппи; пер. с англ. – М. : БИНОМ, 1997. – 327 с.

223. Спирометрия. Ее техническое обеспечение. Проблемы и перспективы / Е.И. Сокол, А.В. Кипенский, Р.С. Томашевский и др. // «Технічна електродинаміка». Тематичний випуск. Проблеми сучасної електротехніки. Част. 3. – Київ : Інститут електродинаміки НАНУ, 2008. – С. 119–124.

224. Лопата В. А. Медико-технические требования к флоуспирометрам: стандарты, перспективы и возможности выполнения / В.А. Лопата // Український пульмонологічний журнал. – 2005. – № 3 (додаток). – С. 46–49.

225. Лойцянский Л. Г. Механика жидкости и газа / Л.Г. Лойцянский. – М.: Наука, 1970. – 904 с.

226. Aerodynamics of Nasal Airways with Application to Obstruction Chometon F., Gillieron P., Laurent J. et al. – [Электронный ресурс] / Режим доступа: http://www.nasalspray.com/pdf/nasal_airway_aerodynamics.pdf. – Загол. с экрана.

227. Справочник по гидравлическим расчетам / (П.Г. Киселев, А.Д. Альтшуль, Н.В. Данильченко и др.); под ред. П.Г. Киселева.– М.: Энергия, 1974.– 312 с.

228. Bachmann W. Obstructed nasal breathing. Basis investigation: history, inspection, rhinomanomatry, allergy [Електронный ресурс] / W. Bachmann. – 2001. – 31 с. – Режим доступа: http://www.atmosmed.de. – Загл. с экрана.

229. Аврунин О.Г. Обоснование основных медико–технических требований для проектирования многофункционального риноманометра / О.Г. Аврунин, А.И. Бых, В.В. Семенец // Сборник научных трудов 3-й международной научной конференции «Функциональная компонентная база микро-опто- и нано-электроники». – Харьков, ХНУРЕ. – 2010. – С. 280–281.

230. Аврунин О.Г. Методы и средства функциональной диагностики внешнего дыхания: монография/ О.Г. Аврунин, Р.С. Томашевский, Х.И. Фарук – Харьков, ХНАДУ. – 2015. –208 с.

231. Метрологічне забезпечення вимірювань і контролю / Є.Т. Володарський, В.В. Кухарчук, В.О. Поджаренко, Г.Б. Сердюк. – Вінниця: Велес, 2001.– 219 с.

232. Інформаційно-вимірювальні технології неруйнівного контролю / В.П. Малайчук, О.В. Мозговой, О.М. Петренко. – Дніпропетровськ: РВВ ДНУ, 2001.– 240 с.

233. Джонсон Н. Статистика и планирование эксперимента в технике и науке: Методы планирования эксперимента / Н. Джонсон, Ф. Лион.: Пер. с англ. под ред. Э.К. Лецкого. – М. : Мир, 1981. – 520 с.

234. Дуда Р. Распознавание образов и анализ сцен / Р. Дуда, П. Харт. Пер. с англ. под ред. В. Л. Стефанюк. – М. : Мир, 1976. – 512 с.

235. Golden R. M. Mathematical Methods for Neural Network Analysis and Design. – Cambridge, Massachusetts: The MIT Press, 1996. – 420 p.

236. Braun H. Neuronale Netze. Optimierung durch Lernen und Evolution. – Berlin: Springer – Verlag, 1997. – 279 S.

237. Dracopoulos D. C. Evolutionary Learning Algorithms for Neural Adaptive Control. – Berlin: Springer – Verlag, 1997. – 211 p.

238. Shepherd A. J. Second–Order Methods for Neural Networks. – London: Springer–Verlag, 1997. – 145 p.

239. Цыпкин Я. З. Основы теории обучающихся систем. – М.: Наука, 1970. – 252 с.

240. Haykin S. Neural Networks. A Comprehensive Foundation. – Upper Saddle River, N.J.: Prentice Hall, Inc., 1999. – 842 p.

241. Изерман Р. Цифровые системы управления. – М.: Мир, 1984. – 541 с.

242. Bishop C. M. Neural Networks for Pattern Recognition. – Oxford: Clarendon Press, 1995. – 482 p.

243. Фу К. Последовательные методы в распознавании образов и обучении машин. – М.: Наука, 1971. – 256 с.

244. Фукунага К. Введение в статистическую теорию распознавания образов. – М.: Наука,1979. – 368 с.

245. Патрик Э. А. Основы теории распознавания образов. – М.: Сов. радио, 1980. – 408 с.

246. Растригин Л. А., Эренштейн Р. Х. Метод коллективного распознавания. – М.: Энергоиздат, 1981. – 80 с.

247. Handbook of Intelligent Control: Neural, Fuzzy and Adaptive Approaches/ Ed. by D. A. White, D. A. Sofge. –N.Y.: Van Nostrand Reinhold, 1992. – 568 p.

248. Harris C. J., Moore C. G., Brown M. Intelligent Control. Aspects of Fuzzy Logic and Neural Nets. – Singapore: World Scientific. – 1993. – 380 p.

249. Advances in Intelligent Control / Ed. by C. J. Harris. – London: Taylor and Francis, 1994. – 373 p.

250. Цыпкин Я. З. Основы информационной теории идентификации. – М.: Наука, 1984. – 320 с.

251. Льюнг Л. Идентификация систем. Теория для пользователя. – М.:Наука,1991. – 432 с.

252. Omatu S., Khalid M., Yusof R., Neuro–Control and its Applications. – London: Springer–Verlag, 1995. – 255 p.

253. Фомин В. Н., Фрадков А. Л., Якубович В. А. Адаптивное управление динамическими объектами. – М.: Наука, 1981. – 448 с.

254. Деревицкий Д. П., Фрадков А. Л. Прикладная теория дискретных адаптивных систем управления. – М.: Наука, 1981. – 216 с.

255. Real–Time Computer Control / Ed. by S. Bennet, D. A. Linkens. – London: Peter Peregrinus, 1984. – 254 p.

256. Industrial Digital Control Systems / Ed. by K. Warwick, D. Rees. – London: Peter Peregrinus, 1986. – 439 p.

257. Первозванский А. А. Курс теории автоматического управления. – М.: Наука, 1986. – 616 с.

258. Cichocki A., Unbehauen R. Neural Networks for Optimization and Signal Processing. – Stuttgart: Teubner, 1993. – 526 p.

259. Адаптивные фильтры / Под ред. К. Ф. Н. Коуэна, П. М. Гранта. – М.: Мир, 1988. – 329 с.

260. Бабак В. П., Хандецький В. С., Шрюфер Е. Обробка сигналів. – К.: «Либідь», 1999. – 496 с.

261. Haykin S. Adaptive Filter Theory. – Upper Saddle River, N.J.: Prentice Hall, 1996. – 987 p.

262. Уидроу Б., Стирнз С. Адаптивная обработка сигналов. – М.: Радио и связь, 1989. – 440 с.

263. Amari S., Cardoso J.–F. Blind source separation semiparametric statistical approach // IEEE Trans. on Signal Processing. – 1997. – 45. – P. 2692–2700.

264. Montgomery D. C., Johnson L. A., Gardiner J. S. Forecasting and Time Series Analysis. – N.Y.: Mc Graw-Hill, 1990. – 394 p.

265. Pham D. T., Liu X. Neural Networks for Identification, Prediction and Control. – London: Springer–Verlag, 1995. – 238 p.

266. Masters T. Neural, Novel & Hybrid Algorithms for Time Series Prediction. – N.Y.: John Wiley & Sons, Inc., 1995. – 514 p.

267. Zirilli J. S. Financial Prediction Using Neural Networks. – London: Int. Thomson Computer Press, 1997. – 135 p.

268. Kingdom J. Intelligent Systems and Financial Forecasting. – Berlin: Springer–Verlag, 1997. – 227 p.

269. Растригин Л. А. Системы экстремального управления. – М.: Наука, 1974. – 632 с.

270. Kaczmarz S. Angenaeherte Ausloesung von Systemen linearer Gleichungen// Bull. Int. Acad. Polon. Sci. – 1937. – Let.A. – S. 355–357.

271. Kaczmarz S. Approximate solution of systems of linear equations // Int. J. Control. – 1993 – 53. – P. 1269–1271.

272. Зайцев И.Д., Бодянский Е.В., Руденко О.Г. Применение адаптивных алгоритмов идентификации при разработке математического обеспечения АСУТП химических производств. – Черкассы: ОНИИТЭХИМ, 1989. – 89 с.

273. Avedjan A. D. Bestimmung der Parameter linearer Modellen stationaerer und nichtstationaerer Stoerungen // Messen–Steuern–Regeln. – 1971. – № 9. – S. 348–350.

274. Nagumo J., Noda A. A learning method for system identification // IEEE Trans. on Autom. Control. – 1967. – 12. – P. 282–237.

275. Фаддеев Д. Е., Фаддеева В. Н. Вычислительные методы линейной алгебры. – М., Л.: ГИФМЛ, 1963. – 623 с.

276. Вазан М. Стохастическая аппроксимация. – М.: Мир, 1972. – 289 с.

277. Бодянский Е.В., Буряк Ю.О., Содин М.Л. Адаптивный конечношаговый алгоритм идентификации // АСУ и приборы автоматики. – Харьков: Выща шк. – 1987. – Вып. 84. – С. 53–57.

278. Бодянський Є.В., Борячок М.Д. Оптимальне керування стохастичними об’єктами в умовах невизначеності. – К.: ІСДО, 1993. – 164 с.

279. Аведьян Э.Д. Модифицированные алгоритмы Качмажа для оценки параметров линейных объектов // Автоматика и телемеханика. – 1975. – № 5. – С. 64–69.

280. Перельман И.И. Оперативная идентификация объектов управления. – М.: Энергоатомиздат, 1982. – 272 с.

281. Шильман С.В. Итеративное линейное оценивание с регулируемым объемом предыстории // Автоматика и телемеханика. – 1983. – № 5. – С. 93–98.

282. Бодянский Е. В., Чайников С. И., Ачкасов А. Е., Вороновский Г. К. Адаптивные алгоритмы управления в АСУ ТП и оценка их эффективности на ранних стадиях проектирования. – Харьков: ХОУС, 1995. – 134 с.

283. Чуев Ю. В., Михайлов Ю. Б., Кузьмин И. В. Прогнозирование количественных характеристик процессов. – М.: Сов. радио,1975. – 400 с.

284. Алберт А. Регрессия, псевдоинверсия и рекуррентное оценивание. – М.: Наука, 1977. – 224 с.

285. Бодянский Е. В., Буряк Ю. О., Содин М. Л. Адаптивные алгоритмы идентификации с конечной памятью. – Харьков, 1984. – 47 с. Рук. деп. в УкрНИИНТИ 05.02.1985, № 218Ук. – 85 Деп.

286. Бодянский Е.В., Котляревский С.В., Ачкасов А.Е., Вороновский Г.К. Адаптивные регуляторы пониженного порядка. – Харьков: ХГАХ, 1996. – 114 с.

287. Цыпкин Я. З. Адаптация и обучение в автоматических системах. – М.: Наука, 1968. – 400 с.

288. Бодянский Е.В., Плисс И.П., Соловьева Т.В. Многошаговые оптимальные упредители многомерных нестационарных стохастических процессов // Доклады АН УССР. – 1986. – Сер. А. – № 12. – С. 47–49.

289. Goodwin G.C., Ramadge P.J., Caines P.E. Discrete time stochastic adaptive control // SIAM J. Control and Optimization. – 1981. – 19. – P. 829–853.

290. Goodwin G.C., Ramadge P.J., Caines P.E. A globally convergent adaptive predictor // Automatica. – 1981. – 17. – P. 135–140.

291. Бодянский Е.В., Руденко О.Г. Адаптивные модели в системах управления техническими объектами. – К.: УМК ВО, 1988. – 212 с.

292. Бодянский Е.В., Плисс И.П. Об одном модифицированном алгоритме одновременного действия для идентификации объектов управления.– Харьков, 1981.–13 с. Рук. деп. в ВИНИТИ 16.09.1981, № 4474–81 Деп.

293. Бодянский Е.В., Плисс И.П. Об одном многошаговом адаптивном алгоритме идентификации нестационарных объектов. – Харьков, 1984. – 8 с. Рук. деп. в УкрНИИНТИ 03.02.1984, № 183 Ук – Д 84.

294. Демиденко Е.З. Линейная и нелинейная регрессия. – М: Финансы и статистика, 1981. – 302 с.

295. Копысицкий Т.И., Сергеенкова Е.В. Применение смещенных оценок при разработке моделей для АСУ ТП нефтепереработки и нефтехимии. – М.: ЦНИИТЭнефиихим, 1982. – 58 с.

296. Бодянский Е.В., Плисс И.П. Об одном адаптивном алгоритме смещенного оценивания. // Тез. докл. Всесоюзн. семинара «Эволюционное

моделирование и обработка данных радиофизического эксперимента». – М., 1984 – С. 50–51.

297. Бодянский Е.В., Плисс И.П. О рекуррентных процедурах двухпараметрического смещенного оценивания // АСУ и приборы автоматики. – Харьков: Выща шк. – 1987. – Вып. 83. – С. 109–110.

298. Ермаков С.М., Панкратьев Ю.Д. Смещенные оценки и метод регуляризации // Вестник ЛГУ. – 1970. – 2. – № 7. – С. 27–30.

299. Жуковский Е.Л., Липцер Р.Ш. О рекуррентном способе вычисления нормальных решений линейных алгебраических уравнений // Журнал выч. мат. и мат. физики. – 1972. – 12. – № 4. – С. 343–357.

300. Руденко О.Г. Об одном рекуррентном регуляризованном алгоритме оценивания параметров линейного объекта // Доповіді НАН України. – 2000. – № 9. – С. 112–114.

301. Бодянский Е.В., Плисс И.П. Адаптивное ридж–оценивание. – Харьков, 1984. – 10 с. Рук. деп. в УкрНИИНТИ 02.10.1984, № 1558 Ук – 84 Деп.

302. Райбман Н.С., Чадеев В.М. Построение моделей процессов производства. – М.: Энергия, 1975. – 376 с.

303. Schweppe F.C. Uncertain Dynamic Systems. – Englewood Cliffs., N. J.: Prentice Hall, Inc., 1973. – 533 p.

304. Norton J. P. An Introduction to Identification. – London: Academic Press Inc., 1986. – 310 p.

305. Черноусько Ф. Л. Оценивание состояния динамических систем. Метод эллипсоидов. – М.: Наука, 1988. – 320 с.

306. Fogel E., Huang Y.F. On the value of information in system identification – bounded noise case // Automatica. – 1982. – 18. – P. 229–238.

307. Norton J. P. Identification and application of bounded parameter models // Automatica. – 1987. – 23. – P. 497–507.

308. Halwass M. «Least-Squares»-Modifikationen in Vegenwart begrenzter Stoerungen // Messen-Stoerung-Regeln. – 1990. – 33. – №. 8. – S. 351–355.

309. Norton J. P., Mo S. H. Parameter bounding identification algorithms for time varying systems // Math. and Comp. in Simul. – 1990. – 32. – P. 537–544.

310. Canudas de Wit C., Carrillo J. A modified EW–RLS algorithm for systems with bounded disturbances // Automatica. – 1990. – 26. – P. 599–606.

311. Arruda L. V. R., Favier Y. A review and comparison of robust estimation methods // Preprints 9–th IFAC/IFORS Symp «Identification and System Parameter Estimation». – Vol.2 – Budapest, 1991. – P. 1027–1032.

312. Maksarov D. Y., Norton J. P. State bounding with ellipsoidal set description of the uncertainty // Int J. Control. – 1996. – 65. – P. 847–866.

313. Арчакова А. В., Бодянский Е. В., Сухарев С. А. Об одном алгоритме рекуррентного оценивания с использованием метода эллипсоидов. // Радиоэлектроника и информатика.–1997. – № 1. – С. 77–79.

314. Haegglund T. Recursive estimation of slowly time–varying parameters // Proc. IFAC/IFORS Symp. «Identification and System Parameter Estimation» – York, UK, 1985. – P. 1137–1142.

315. Арчакова А. В., Бодянский Е. В., Сухарев С. А. Об одном алгоритме технической диагностики на основе множественного идентификационного подхода // Радиоэлектроника и информатика. – 1998. – № 2 (03). – С. 37–40.

316. Shynk J. J. Performance surfaces of a single–layer perceptron // IEEE Trans. on Neural Networks. – 1990. – 1. – P. 268–274.

317. Бодянский Е. В., Удовенко С. Г., Ачкасов А. Е., Вороновский Г. К. Субоптимальное управление стохастическими процессами. – Харьков: Основа, 1997. – 140 с.

318. Kruschke J. K., Movellan J. R. Benefits of gain: speeded learning and minimal layers backpropagation networks // IEEE Trans. on Syst., Man and Cybern. – 1991. – 21. – P. 273–280.

319. Бодянский Е. В., Кулишова Н. Е., Руденко О. Г. Об одной модели формального нейрона // Доповіді НАН України. – 2001. – № 4. – С. 69–73.

320. Darken C., Moody J. Towards faster stochastic gradient research // Advances Neural Information Processing Systems. – San Mateo, 1991. – 1. – P. 1009–1016.

321. Polyak B. T. New method of stochastic approximation type // Automation and Remote Control. – 1990. – 51. – P. 937–946.

322. Поляк Б. Т. Введение в оптимизацию. – М: Мир, 1984. – 541 с.

323. Schmidhuber J. Accelerated learning in back–propagation nets // Connectionism in Perspective. – Amsterdam: Elsevier, 1989. – P. 439–445.

324. Chan L.W., Fallside F. An adaptive learning algorithm for backpropagation networks // Computer Speech and Language. – 1987. – 2. – P. 205–218.

325. Johansson E. M., Dowla F. V., Goodman D. M. Backpropagation learning for multiplayer feed–forward networks using the conjugate gradient method // Int. J. Neural Systems. – 1992. – 2. – P. 291–301.

326. Charalambous C. Conjugate gradient algorithm for efficient training of artificial neural networks // IEE Proc. – Part G. – 1992. – 139. – P. 301–310.

327. Jacobs R. A. Increased rates of convergence through learning rate adaptation // Neural Networks. – 1988. – 1. – P. 295–307.

328. Silva F. M., Almeida L. B. Speeding up backpropagation / Ed. by R. Eckmiller «Advances of Neural Computers.» – North–Holland: Elsevier Science Publishers. – B.V., 1990. – P. 151–158.

329. Tollenaere T. Super SAB: fast adaptive backpropagation with good scaling properties // Neural Networks. – 1990. – 3. – P. 561–573.

330. Rojas R. Neural Networks. A Systematic Introduction. – Berlin: Springer–Verlag, 1996. – 502 p.

331. Fahlman S. E., Lebiere C. The cascade–correlation learning architecture / Ed. by D. S. Touretzky «Advances in Neural Information Processing Systems.» – San Mateo, CA: Morgan Kaufman, 1990. – P. 524–532.

332. Бард Й. Нелинейное оценивание параметров. – М.: Статистика, 1979. – 349 с.

333. Evans P. J., Betz R. E. New results and applications of adaptive control to classes of nonlinear systems // Ricerche di Automatica. – 1982. – 13. – P. 277–297.

334. Allen D. M. Parameter estimation for nonlinear models with emphasis on compartmental models // Biometrics. – 1983. – 39. – P. 629–637.

335. Billings S. A., Voon W. S. F. Least squares parameter-estimation algorithms for nonlinear systems // Int. J. Syst. Sci. – 1984. – 15. – P. 601–615.

336. Lindstroem P., Wedin P.-A. A new linesearch algorithm for nonlinear squares problem // Math. Progr. – 1984. – 29. – P. 268–296.

337. Lachmann K. H. Selbsteinstellende nichtlineare Regelalgorithmen fuer eine bestimmte Klasse nichtlinearer Prozesse // Automatisierungstechnik. – 1985. – 33. – № 7. – S. 210–218.

338. Hartley H. The modified Gauss–Newton method for the fitting of nonlinear regression function of least squares // Technometrics. – 1961. – №3. – P. 269–280.

339. Marquardt D. An algorithm for least squares estimation of nonlinear parameters // SIAM J. Appl. Math. – 1963. – № 11. – P. 431–441.

340. Бодянский Е. В., Воробьев С. А. Рекуррентная нейронная сеть для обнаружения изменений свойств нелинейных стохастических последовательностей // Автоматика и телемеханика. – 2000. – № 7. – С. 55–67.

341. Бодянский Е. В., Попов С. В. Прогнозирующая нейронная сеть и алгоритмы ее обучения // Радіоелектроніка. Інформатика. Управління. – 2000. – № 1. – С. 60–64.

342. Бодянский Е. В. Адаптивные алгоритмы идентификации нелинейных объектов управления // АСУ и приборы автоматики. – Харьков: Выща шк., 1987. – Вып. 81. –С. 43–46.

343. Bodyanskiy Ye., Kolodyazhniy V., Stephan A. An adaptive learning algorithm for a neuro–fuzzy network / Ed. by B. Reusch «Computational Intelligence. Theory and Applications.» – Berlin – Heidelberg – New York: Springer, 2001. – P. 68–75.

344. Wang H., Liu G. P., Harris C.J., Brown M. Advanced Adaptive Control. – Oxford: Pergamon, 1995. – 262 p.

345. Brooks S. H. A discussion of random methods for seeking maxima // Oper. Research – 1985. – 6. – P. 244–253.

346. Растригин Л. А. Статистические методы поиска. – М.: Наука, 1968. – 376 с.

347. Растригин Л. А., Рипа К. К. Автоматическая теория случайного поиска. – Рига: Зинатне, 1973. – 343 с.

348. Растригин Л. А. Случайный поиск в процессах адаптации. – Рига: Зинатне, 1973. – 132 с.

349. Solis F. J., Wets J. B. Minimization by random search techniques // Math. of Operation Research. – 1981. – 9. – P. 19–30.

350. Box G. E. P. Evolution operation: A method for increasing industrial productivity // Applied Statistics. – 1957. – 6. – P. 81–101.

351. Spendley W., Hext G. R., Himsworth F. R. Sequential application of simplex design in optimization and evolutionary operation // Technometrics. – 1962. – 4. – P. 441–461.

352. Nelder J. A., Mead R. A simplex method for function minimization // Computer J. – 1965. – 7. – P. 308–313.

353. Горский В. Г., Адлер Ю. П. Планирование промышленных экспериментов. – М.: Металлургия, 1974. – 264 с.

354. Holland J. H. Adaptation in Natural and Artificial Systems. An Introductory Analisis with Application to Biology, Control and Artificial Intelligence. – London: Bradford Book Edition, 1994. – 211 p.

355. Батищев Д. И. Генетические алгоритмы решения экстремальных задач – Воронеж: Воронеж. гос. техн. ун–т., 1995. – 69 с.

356. Вороновский Г. К., Махотило К. В., Петрашев С. Н., Сергеев С. А. Генетические алгоритмы, искусственные нейронные сети и проблемы виртуальной реальности. – Харьков: Основа, 1997. – 112 с.

357. Курейчик В. М. Генетические алгоритмы. Состояние. Проблемы. Перспективы // Известия РАН. Теория и системы управления. – 1999. – № 1. – С. 144–160.

358. Фогель Л., Оуэнс А., Уолш М. Искусственный интеллект и эволюционное моделирование. – М.: Мир, 1969. – 230 с.

359. Schalkoff R.J. Artificial Neural Networks. – N.Y.: The McGraw–Hill Comp., Inc., 1997. – 422 p.

360. Реклейтис Г., Рейвиндран А., Рэгсдел К. Оптимизация в технике: Кн. 1. – М.: Мир, 1986. – 349 с.

361. Бодянский Е. В., Муравьев С. В. Синтез комбинированного алгоритма поиска глобального экстремума функции многих переменных // АСУ и приборы автоматики. – Харьков: Выща шк., 1979. – Вып. 50. – С. 66–74.

362. Налимов В. В., Чернова Н. А. Статистические методы планирования экстремальных экспериментов. – М.: Наука, 1965. – 340 с.

363. Задачи статистической оптимизации. – Рига: Зинатне, 1971. – 232 с.

364. Химмельблау Д. М. Прикладное нелинейное программирование. – М.: Мир, 1975. – 534 с.

365. Бодянский Е.В., Илюнин О.К., Муравьев С.В. О задаче экспериментальной оптимизации технологических объектов в условиях помех // АСУ и приборы автоматики. – Харьков: Выща шк., 1979. – Вып. 52. – С. 97–99.

366. Бодянский Е. В. Об одном комбинированном алгоритме адаптационной оптимизации // АСУ и приборы автоматики. – Харьков: Выща шк., 1979. – Вып. 51. – С. 36–40.

367. Box M. J. A new method of constrained optimization and a comparison with other methods // Computer J. – 1965. – 8. – P. 42–52.

368. Салыга В.И., Удовенко С.Г., Бодянский Е.В., Шамша Б.В. О применении метода эволюционного планирования для оптимизации технологических процессов / В кн. «Повышение эффективности, совершенствование процессов и аппаратов химических производств» – Харьков, 1976. – С. 25–27.

369. Бодянский Е. В., Илюнин О. К., Муравьев С. В. О применении адаптивных методов в планировании эксперимента // АСУ и приборы автоматики. – Харьков: Выща шк., 1978. – Вып. 45. –С. 62–65.

370. Holland J. H. Genetic algorithms and the optimal allocations of trails // SIAM J. of Computing. – 1973. – 2. – P. 88–105.

371. Whitley D. A Genetic Algorithm Tutorial // Technical Report CS – 93 – 103. – Colorado State Univesity, 1993. – 40 p.

372. Дюк В., Самойленко А. Data mining. – СПб: Питер, 2001. – 368 с.

373. Dorado J., Santos A., Rabunal J. R., Pedreira N., Pazos A. Hybrid two-population genetic algorithm / Ed. by B. Reusch «Computational Intelligence. Theory and Application». – Berlin – Heidelberg – New-York: Springer, 2001. – P. 464–470.

374. Круглов В. В., Борисов В. В. Искусственные нейронные сети. Теория и практика. – М.: Горячая линия – Телеком, 2001. – 382 с.

375. Seifipour N., Menhaj M. B. A GA–based algorithm with a very fast rate of convergence / Ed. by B. Reusch «Computational Intelligence. Theory and Application». – Berlin – Heidelberg – New-York: Springer, 2001. – P. 185–193.

376. Koza J. R., Rice J. P. Genetic generation of both the weights and architecture for neural network // Proc. Int. Joint Conf. Neural Networks «IJCNN'91». – 1991. – Part II. – P. 397–404.

377. Werbos P. Backpropagation through time. What it does and how to do it // Proc. IEEE. – 1990. – 78. – P. 1550–1560.

378. Рубан А. И., Ялатдинов Р. Р. Исследование адаптивных робастных алгоритмов идентификации параметров нестационарных объектов. – Томск, 1985. – 20 с. Рук. деп. в ВИНИТИ 11.07.1985., № 5637 – 85 Деп.

379. Hogg R. V. Adaptive robust procedures: a partial review and some suggestions for future applications and theory // JASA. – 1969. – № 348. – P. 909–927.

380. Rey J. W. W. Robust Statistical Methods. – Berlin – Heidelberg – New York: Springer– Verlag, 1978. – 128 p.

381. Hogg R. V. Statistical robustness: one view of its use in applications today // The American Statistician. – 1979. – 33. – P. 108–115.

382. Karaiannis N. B., Venetsanopoulos A. N. Fast learning algorithm for neural networks // IEEE Trans. on Circuits and Systems. – 1992. – 39. – P. 453–474.

383. Widrow B., Lee M. 30 years of adaptive neural networks: perceptron, adaline and backpropagation // Proc. IEEE. – 1990. – 78. – P. 1415–1442.

384. Hagan M. T., Demuth H. B., Beale M. Neural Network Design. – Boston: PWS Publishing Company, 1996. – 729 p.

385. Ham F. M., Kostanic I. Principles of Neurocomputing for Science & Engineering. – N.Y.: Mc Graw–Hill, Inc., 2001. – 642 p.

386. Amary S. Mathematical theory of neural learning // New Generation Computing. – 1991. – 8. – P. 281–294.

387. Oja E. A simplified neuron model as a principal component analyzer// J. of Math. Biology. – 1982. – 15. – P. 267–273.

388. Oja E. Neural networks, principal components, and subspaces// Int. J. of Neural Systems. – 1989. – 1. – P. 61–68.

389. Лоули Д., Максвелл А. Факторный анализ как статистический метод.– М.: Мир, 1967. – 144 с.

390. Иберла К. Факторный анализ. – М.: Статистика, 1980. – 398 с.

391. Chen T., Hua Y., Yan W.–Y. Global convergence of Oja's subspace algorithm for principal component extraction // IEEE Trans. on Neural Networks. – 1988. – 9 – P. 58–67.

392. Бодянский Е. В., Запорожец О. В., Путятина Т. В., Рагулина О. Е. Нейросетевая модель факторного анализа // Проблемы бионики. – Харьков, 1999. – Вып.51. – С. 84–90.

393. Бодянський Є. В., Михальов О. І., Плісс І. П. Адаптивне виявлення розладнань в об'єктах керування за допомогою штучних нейронних мереж. – Дніпропетровськ: Системні технології, 2000. – 140 с.

394. Chen S., Billings S. A. Neural networks for nonlinear dynamic system modelling and identification / Ed. by C. J. Harris «Advances in Intelligent Control». – London: Taylor and Francis, 1994. – P. 85–112.

Наукове видання

Аврунін Олег Григорович,
Бодянський Євгеній Володимирович,
Калашник Михайло Васильович,
Семенець Валерій Васильович,
Філатов Валентин Олександрович

СУЧАСНІ ІНТЕЛЕКТУАЛЬНІ ТЕХНОЛОГІЇ ФУНКЦІОНАЛЬНОЇ МЕДИЧНОЇ ДІАГНОСТИКИ

Монографія

Рецензенти:
А.О. Каргін, д-р. техн.. наук, професор, завідувач кафедри інформаційних технологій Українського державного університету залізничного транспорту
А.Д. Черенков, д-р. техн.. наук, професор, професор кафедри біомедичної інженерії та теоретичної електротехніки Харківського національного технічного університету сільського господарства ім. П. Василенка

Відповідальний випусковий Т.В. Носова
Редактор О.Г. Троценко
Комп'ютерна верстка Л.Ю. Свєтайло

Підп. до друку 30.01.18. Формат 60x84 1/16. Спосіб друку – ризографія.
Умов.друк.арк. 13,7. Облік. вид.арк. 12,3. Тираж 300 прим.
Ціна договірна Зам № 2-387

ХНУРЕ. Україна. 61166, Харків, просп. Науки, 14

Віддруковано в редакційно-видавничому відділі ХНУРЕ
61166, Харків, просп. Науки, 14