УДК 004.855 DOI: https://doi.org/10.30837/IYF.IIS.2024.094

FORMATION OF ALTERNATIVE APPROACHES TO NOISE GENERATION IN GAN NETWORKS

Bilokon V. A.
Scientific supervisor – Ph.D., prof. Ryabova N. V.
Kharkiv National University of Radio Electronics
(61166, Kharkiv, Nauki Ave., 14, Dep. of Artificial Intelligence)
e-mail: vasyl.bilokon@nure.ua

This work discusses the problem of developing alternative approaches to generating noise in generative adversarial networks (GAN). Various noise generation techniques are important for training GAN networks as they help improve the quality of the generated data and the stability of training. This article provides an overview of current noise generation methods and discusses an approach based on the Pandas library in the Python programming language for generating, storing, and mixing noises.

In the modern world, neural networks are widely used in many fields, including computer vision, natural language processing, and the generation of a wide range of content such as:

- images – photorealistic images of people, landscapes, etc;
- texts – articles, news, poems;
- music – melodies, sounds;
- video – short videos, animation, special effects.

The capabilities of GANs will only expand with developing technologies and computing machines. It is worth noting that GANs do not always generate flawless content. Images may be blurry or have distortions, text may not make sense, and music may be cacophonous. At the same time, GANs are a powerful tool that can be used to create new and original content.

Our research work will be aimed at studying an exercise of image generation using GAN networks. Since the process of training GAN networks often faces the problem of generating high-quality noise, special attention will be paid to the study of noise generation methods that are necessary for the diversity and stability of neural network training.

Generative adversarial network (GAN) is a deep neural network model that consists of two main components: a generator and a discriminator. The generator is responsible for generating new data samples by simulating the distribution of the training set, while the discriminator tries to guess and distinguish the generated data from the real data [1].

The architecture of the GAN network consists of several layers that ensure the task of data generation. Training GANs is a competition process between the generator and the discriminator, and successful training requires the right balance between the diversity of the generated data and its realism.

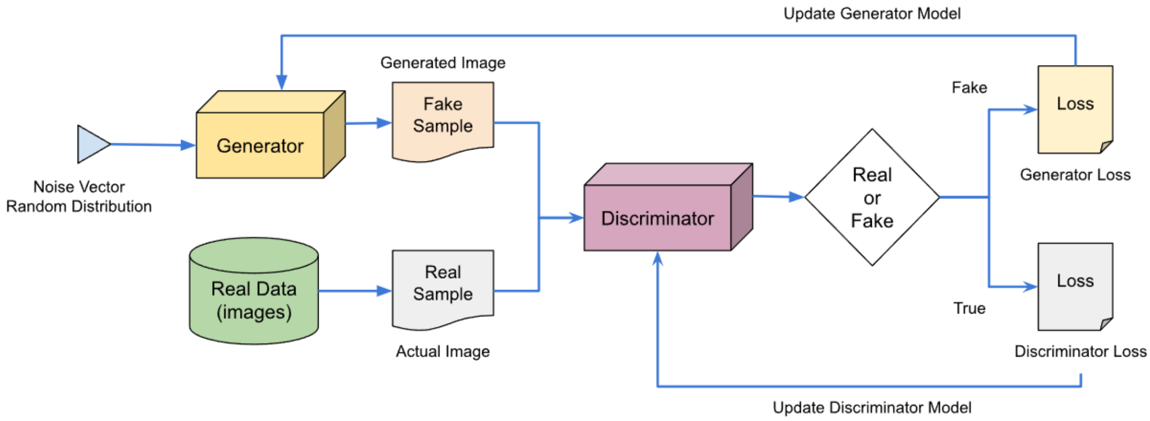

Fig. 1 – Generic architecture of GAN network

Here is an overview of GAN network work logic:

1) The generator analyzes the training set and identifies data attributes.

2) The discriminator neural network also analyzes the initial training data and distinguishes between the attributes independently.

3) The generator modifies some data attributes by adding noise (or random changes) to certain attributes.

4) The generator passes the modified data to the discriminator.

5) The discriminator calculates the probability that the generated output belongs to the original dataset.

6) The discriminator gives some guidance to the generator to reduce the noise vector randomization in the next cycle.

To train the discriminator, both labeled images are used – generated by the generator and real images. The discriminator learns to identify images as real or generated and is trained using a loss function. Typically, the generator and discriminator are trained alternately. This training approach is similar to a two-player adversarial game, the generator seeks to maximize the discriminator's loss, and the discriminator seeks to minimize its own loss [2].

In our work, a noise vector extracted from a random distribution is used to train the generator. The result of the generator will be an image of a certain size. We use an array of size 224 x 224 x 3.

There are several approaches to generating noise in GANs, including adding random vectors to the generator inputs, applying random distortions to data samples, and using data augmentation to increase the diversity of the training set.

1) Adding random vectors. These random vectors, known as noise vectors or hidden variables, add variety to the generation process and allow you to create different variations of the output. Advantages are ease of implementation and the ability to manage the variety of generated data. A disadvantage is a possible loss of semantic connection between input data and output data.

2) Random distortion. Another approach is to apply random distortions to the data samples before passing them to the generator. These distortions may

include changes in brightness, contrast, rotations, shifts, etc. This technique helps the generator explore different aspects of the data and create more diverse samples. An advantage is taking into account various aspects of data variability such as lighting, viewing angle, etc. A disadvantage is may distort the semantic content of data, especially in the case of complex objects.

3) Data Augmentation. It is a process of creating new data samples by applying various transformations to existing samples. In the context of GANs, data augmentation can be used to increase the diversity of the training set and improve the quality of the generated data. This approach may involve random rotations, cropping, resizing, etc. Benefits: Increases the diversity of the training set, helps prevent overfitting, and improves the model's ability to generalize. A disadvantage is that it may require additional computing resources to generate large amounts of augmented data.

4) Using conditions or class labels. To control data generation and train the generator, you can use conditions or class labels that indicate what data types should be generated. For example, when generating images, you can specify what object class should be in the image, allowing you to create samples with certain characteristics. An advantage is allowing to control of data generation following specified characteristics. A disadvantage is it requires annotated data with class labels, which can be a labor-intensive process [2, 4].

To implement these approaches, in our work, we use the Pandas library in the Python programming language. The Pandas library has extensive capabilities for working with data, including its generation and transformation. We can use Pandas to create different types of noise data such as gaussian noise, random distortion, and data augmentation. In addition, Pandas provides convenient tools for storing and mixing generated noise, which allows you to effectively use them in the process of training GAN networks [3].

This article aimed at the problem of developing alternative approaches to generating noise in GAN networks. We reviewed the basic concepts of GAN network architecture, various noise generation methods and suggested using the Pandas library to implement them. In the future, noise generation methods will be explored that can improve the quality of images generated by GAN networks.

References:

1. Goodfellow I., "Generative Adversarial Nets" // Advances in Neural Information Processing Systems. 2014. arXiv:1406.2661.

2. Snell J., "Learning to Generate Images with Perceptual Similarity Metrics" // IEEE International Conference on Image Processing (ICIP). 2017 arXiv:1511.06409. Pages 4277–4281.

3. Chen Daniel Y., "Pandas for Everyone: Python Data Analysis", Addison-Wesley Professional; 2nd edition, 2022. 512 p.

4. Bodyanskiy Y., Antonenko T. Deep neural network based on generalized neo-fuzzy neurons and its learning based on backpropagation //Artificial Intelligence. – 2021. – T. 26. – №. 1. – C. 32-41.