Generate Anime Characters with DCGAN

1st Ti-Wen Chen
Institute of International Business
National Taiwan University
Taipei, Taiwan
r10724050@ntu.edu.tw

***Abstract*—In this research, I utilized the Anime GAN Lite dataset available on Kaggle and developed a DCGAN (Deep Convolutional Generative Adversarial Network) architecture. This endeavor successfully culminated in a project capable of autonomously generating anime images.**

***Index Terms*—Deep Convolutional Generative Adversarial Network**

I. Introduction

Since the inception of the Generative Adversarial Network (GAN) framework by Goodfellow et al. [1] in 2014, the landscape of machine learning and deep learning research has undergone a significant transformation. The focus has expanded from merely analyzing, processing, and predicting data to the more complex and nuanced task of data generation. This groundbreaking shift has opened up unprecedented possibilities for application across a multitude of fields, including but not limited to computer vision (CV), natural language processing (NLP), and various sectors such as business, finance, medicine, and engineering. The versatility and adaptability of GANs have fostered a burgeoning field of research, leading to the emergence of a vast array of GAN-related studies, models, and frameworks, affectionately referred to as the ”GAN zoo.” The profound impact of GANs extends beyond the academic and theoretical realms, influencing practical applications and innovation in industry and research. The ability of GANs to generate realistic, high-fidelity data has catalyzed advancements in areas such as image and voice generation, synthetic data production for training machine learning models, and even the creation of novel artistic and creative works.

In this context, the project I undertook aims to harness the capabilities of the Deep Convolutional Generative Adversarial Network (DCGAN) architecture, a pivotal variant within the GAN family known for its effectiveness in generating high-quality images. By utilizing the Anime GAN Lite dataset, sourced from the renowned Kaggle platform, this project specifically focuses on the generation of anime images. The successful implementation of this project underscores the potential of GANs to revolutionize content creation across various domains, setting a precedent for future innovations in the field.

The structure of the remainder of this paper is as follows. Section 2 commences with an introduction to the data utilized in this study. Section 3 introduces the DCGAN model that used in this study. Section 4 presents the results. Finally, Section 5 provides the conclusion of this study.

II. Data

A. Data Sources

The dataset employed in this project is the Anime GAN Lite dataset[1], a comprehensive collection that is hosted on the renowned data science platform, Kaggle. This particular dataset contains over 20,000 anime images. Figure 1 shows some samples in the dataset.

Fig. 1. Anime GAN Lite Dataset image samples

III. Methodology

A. Generative Adversarial Network

Algorithm 1: Generative Adversarial Network

$\mathbf{D} \leftarrow discriminator$; $\mathbf{G} \leftarrow generator$;
for *each epoch* **do**
  $\boldsymbol{z} \leftarrow random\ noise$;
  $\boldsymbol{x} \leftarrow real\ data$;
  update the discriminator by increasing:

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \{\log \mathbf{D}(x_i) + \log[1 - \mathbf{D}(\mathbf{G}(z_i))]\}$$

  update the generator by decreasing:

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log[1 - \mathbf{D}(\mathbf{G}(z_i))]$$

end

In this project, both the generator and the discriminator are constructed using deep convolutional networks, which is why this architecture is referred to as DCGAN. In the following sections, I will introduce how the generator and discriminator are structured within this project.

[1]https://www.kaggle.com/datasets/prasoonkottarathil/gananime-lite/data

B. CNN Architecture

The generator in a DCGAN architecture aims to map latent space vectors to the data space, generating new data samples that mimic the distribution of the real data. It typically consists of a series of transposed convolutional layers that progressively upsample the input latent vector to the desired output size, integrating non-linear activation functions and normalization methods to enhance the generation process.

Conversely, the discriminator acts as a binary classifier, differentiating between real data samples and fake samples produced by the generator. It comprises several convolutional layers that downsample the input image, employing non-linear activations and normalization to process the data efficiently.

Both networks leverage the convolutional layers' hierarchical structure to capture and utilize complex patterns in the data, a characteristic that significantly contributes to the DCGAN's robustness and effectiveness in generating high-quality synthetic images. The integration of these deep convolutional networks within the GAN framework facilitates a powerful model capable of learning detailed and intricate data representations, thereby driving the success of generative tasks in this project.

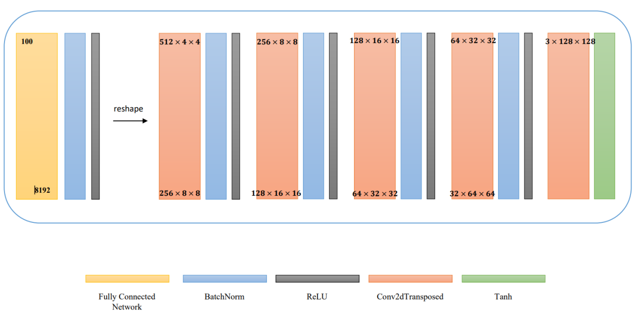

Fig. 2. Generator Architecture

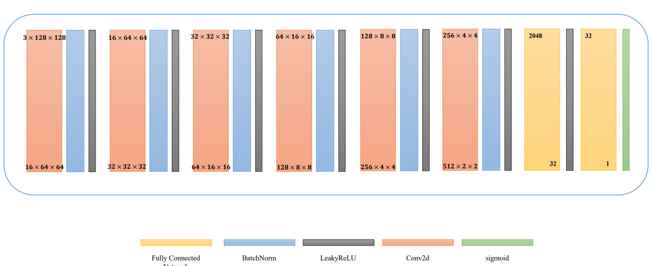

Fig. 3. Discriminator Architecture

IV. Results

In this detailed analysis, we examine the progression of a neural network model, specifically focusing on the adversarial dynamics between the generator and discriminator across a substantial training duration of 150 epochs. The graphical representation delineated in Figure 4 provides a comprehensive elucidation of the loss trajectories associated with both entities throughout the training process.

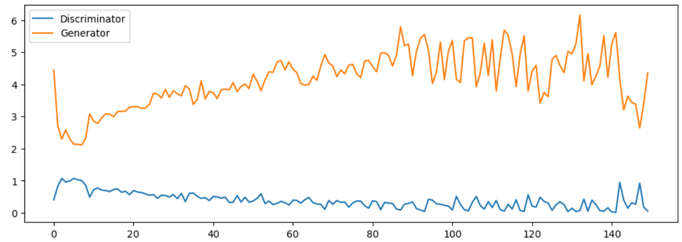

Fig. 4. Generator and Discriminator Loss Trajectories

In contrast to traditional neural network training paradigms, where a consistent decrement in loss is typically indicative of learning and convergence, the adversarial framework presents a more complex interplay. Herein, both the discriminator and the generator are engaged in a continuous strategic contest, evolving their parameters in response to the other's adaptations. This dynamic interaction results in a non-linear loss trajectory, lacking a predictable path of decline, as one might expect in simpler, non-adversarial models.

Upon a meticulous examination of the loss patterns, it becomes evident that there is no monolithic trend towards minimization. This observation underscores the nuanced equilibrium sought in adversarial training, where the generator's enhancement in generating plausible outputs is counteracted by the discriminator's sharpened acuity in distinguishing genuine from generated data. Consequently, the observed loss for each component does not exhibit a steady descent but rather fluctuates in response to the iterative enhancements of its counterpart.

A particularly noteworthy aspect emerges beyond the 100th epoch, where an increase in the loss volatility for the generator is observed. This phase marks a critical juncture in the training process, potentially indicating heightened competition between the adversarial counterparts. The increased volatility is not merely a numerical artifact but resonates with tangible outcomes, notably impacting the quality of the images synthesized by the generator. Such a trend necessitates a deeper inquiry into the learning dynamics at play, suggesting that the generator may be exploring more diverse or complex strategies to deceive the discriminator, thereby encountering varied success across iterations.

In Figure 5, I have sampled a consistent set of noise inputs and used them to generate images across various epochs using the corresponding generator states. The figure reveals that, post epoch = 100, there is no discernible advancement in the quality of the generated outputs; in fact, some of the anime image results appear to be of lesser quality compared to others.

This observation suggests a plateau in the learning process of the generator beyond the 100th epoch, indicating that the model may have reached its optimization limit under the current training parameters or could be experiencing overfitting or mode collapse issues. Typically, in the domain of GANs, the generator should improve over time, producing increasingly

Fig. 5. Generated images under same noises with different epochs training

Fig. 6. Epoch 125

refined images. However, the stagnation or decline in image quality suggests that beyond a certain point, further training does not necessarily equate to better results, and can even lead to degradation in the fidelity of generated images.

Such a phenomenon warrants a closer investigation into the training dynamics post-epoch 100. It may be beneficial to analyze the learning rates, loss function behavior, and other hyperparameters to identify potential causes for this stagnation. Additionally, implementing techniques such as checkpointing and early stopping based on a validation metric could prevent overtraining and preserve the model quality at its peak performance state.

Figure 6 presents the generative outcomes of the model at epoch 125, which is identified as the best-performing instance across all epochs. The acknowledgement of epoch 125 as a performance peak also implies the practicality of implementing early stopping or model checkpointing strategies. These approaches can prevent unnecessary computational expenditure on further training past the point of optimal performance and can also mitigate risks associated with model regression or overfitting in subsequent epochs. This methodology ensures resource-efficient training and captures the model state that offers the most utility for generative tasks or further applications.

Conclusion

In this project, I trained a DCGAN model to generate anime images. Although the generator was capable of producing commendable anime images, there were observable deficiencies in its performance. Notably, the images exhibited excessive noise, and the training outcomes were not consistently stable. Contrary to expectations, prolonged training did not correlate with improved performance, and extended training durations were observed.

To achieve enhanced training results, a deeper network architecture and an increased number of parameters would be requisite. Such modifications invariably lead to the construction of larger models, which are becoming progressively more demanding in terms of computational resources. These requirements have now surpassed the capabilities of standard personal computing setups.

References

[1] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... Bengio, Y. (2014). Generative adversarial nets. Advances in neural information processing systems, 27.