Custom Generative Adversarial Network Implementation for CIFAR-10

SYED MUHAMMAD MURTAZA KAZMI
Department of Computer Science
National University of Computer and Emerging Sciences
Email: i210685@nu.edu.pk

***Abstract*—This report presents the development and training of a custom Generative Adversarial Network (GAN) tailored for the CIFAR-10 dataset, specifically focusing on the classes of cats and dogs. Unlike traditional GAN architectures where the discriminator distinguishes between real and fake images, the proposed discriminator evaluates the similarity between pairs of images, thereby generating a similarity score. This approach leverages concepts akin to Siamese Networks to enhance the quality of generated images. The GAN was trained over 500 epochs, and its performance was assessed through loss metrics and visual inspection of generated images. The results indicate gradual improvement in image similarity, although challenges such as image blurriness were observed. The report concludes with a discussion on potential enhancements and future work to address existing limitations.**

***Index Terms*—Generative Adversarial Network, CIFAR-10, Custom Discriminator, Image Generation, Machine Learning**

I. Introduction

Generative Adversarial Networks (GANs) have revolutionized the field of generative modeling by enabling the creation of realistic synthetic data. Traditionally, GANs consist of two neural networks—the generator and the discriminator—that engage in a minimax game to produce and evaluate data samples. The generator aims to create images indistinguishable from real ones, while the discriminator strives to correctly identify the authenticity of these images.

In this assignment, we explore a novel GAN architecture tailored for the CIFAR-10 dataset, focusing exclusively on the classes of cats and dogs. Unlike conventional GAN discriminators that classify images as real or fake, the proposed discriminator evaluates the similarity between pairs of images, generating a similarity score. This approach is inspired by Siamese Networks, which are designed to determine the similarity between two inputs. The objective remains consistent with traditional GANs: the generator seeks to minimize the discriminator's ability to distinguish generated images from real ones, thereby enhancing the realism of the synthetic images.

II. Methodology

A. Dataset

The CIFAR-10 dataset [1] comprises 60,000 color images of size 32×32 pixels across 10 distinct classes, with 6,000 images per class. For this assignment, only the classes corresponding to cats and dogs were utilized, resulting in a subset of 12,000 images (6,000 cats and 6,000 dogs). The dataset is divided into 50,000 training images and 10,000 testing images, ensuring ample data for model training and evaluation. The images were sourced from the official CIFAR-10 repository [1].

B. Preprocessing Steps

Effective preprocessing is crucial for the successful training of deep learning models. The following preprocessing steps were implemented:

- **Filtering Classes**: The dataset was filtered to include only images labeled as cats and dogs, reducing the problem to a binary classification task within the GAN framework.
- **Normalization**: Pixel values of the images were normalized to the range $[-1, 1]$ to facilitate stable GAN training. This was achieved using the formula:
$$\text{Normalized Image} = \frac{\text{Image} - 127.5}{127.5}$$
- **Batching and Shuffling**: The training images were organized into batches of 128 samples each. The dataset was shuffled to ensure that each batch contains a diverse set of images, mitigating the risk of the model learning spurious patterns.
- **Prefetching**: To optimize training speed, the dataset was prefetched using TensorFlow's data pipeline capabilities, allowing the model to fetch data asynchronously while training.

C. Model Architecture

1) Generator: The generator model is responsible for creating synthetic images from random noise vectors. The architecture comprises a series of dense and transposed convolutional layers designed to upscale the latent vectors to the desired image dimensions.

- **Input Layer**: Accepts a latent vector of dimension 100.
- **Dense Layer**: Projects the latent vector into a high-dimensional space, reshaping it into a tensor suitable for convolutional processing.
- **Batch Normalization and Leaky ReLU**: Applied after each dense and convolutional layer to stabilize training and introduce non-linearity.
- **Transposed Convolutional Layers**: Upsample the feature maps progressively to reach the final image size of $32 \times 32 \times 3$.

- **Output Layer**: Utilizes a `tanh` activation function to produce images with pixel values in the range $[-1, 1]$.

2) Discriminator: The discriminator is uniquely designed to evaluate the similarity between pairs of images—one real and one generated. This architecture draws inspiration from Siamese Networks, enabling the model to output a similarity score rather than a binary classification.

- **Inputs**: Two images of shape $32 \times 32 \times 3$, representing the generated and real images.
- **Shared Convolutional Neural Network (CNN)**: Processes both input images through identical CNN layers to extract feature representations.
- **Feature Comparison**: Computes the absolute difference between the feature vectors of the two images to quantify similarity.
- **Fully Connected Layers**: Further processes the combined features to produce a final similarity score using a sigmoid activation function.

3) Hyperparameters:

- **Optimizer**: Adam optimizer with a learning rate of 1×10^{-4} and $\beta_1 = 0.5$ for both generator and discriminator.
- **Batch Size**: 128
- **Epochs**: 500
- **Loss Functions**:
  - **Discriminator**: Binary Cross-Entropy loss with label smoothing applied to real images (0.9 instead of 1.0) to enhance training stability.
  - **Generator**: Binary Cross-Entropy loss aiming to maximize the similarity score.

4) Source Code Repository: The complete source code for this project is available on GitHub [2].

III. Results

A. Training Loss Curves

The training process was monitored over 500 epochs, tracking the generator and discriminator losses. Figure 1 illustrates the progression of these loss metrics. The generator loss exhibits a gradual increase, indicative of the discriminator's improving ability to assess image similarity. Conversely, the discriminator loss remains relatively stable, reflecting balanced adversarial training dynamics.

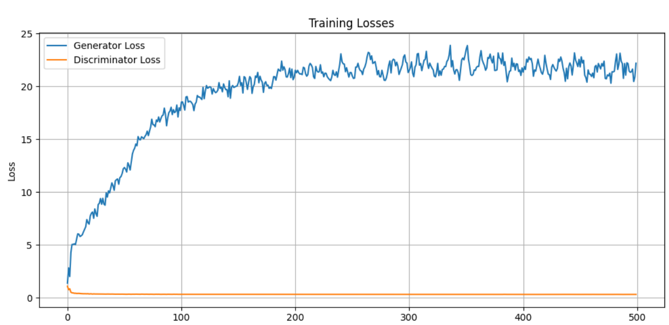

Fig. 1: Generator and Discriminator Loss Over 500 Epochs

B. Generated Images

At the conclusion of training, the generator produced synthetic images that exhibit some resemblance to the target classes (cats and dogs). However, the images are notably blurry, with only faint structural features indicative of the intended objects. Figure 2 showcases two examples of these generated images.

Fig. 2: Examples of Generated Images at Epoch 500

IV. Discussion

The training of the custom GAN on the CIFAR-10 dataset yielded synthetic images with limited clarity and resemblance to real cats and dogs. The generator's increasing loss suggests that the discriminator became more proficient in evaluating image similarity, thereby challenging the generator to produce more realistic images. Despite this adversarial pressure, the generator struggled to produce sharp and detailed images, resulting in blurrier outputs.

Several factors may have contributed to these outcomes:

- **Model Complexity**: The generator and discriminator architectures, while functional, may lack sufficient depth and complexity to capture the intricate features of cat and dog images.
- **Training Duration**: Although trained over 500 epochs, the generator may require more iterations to learn the subtle nuances necessary for high-fidelity image generation.
- **Loss Function Design**: The binary cross-entropy loss, while standard, may not effectively capture the perceptual similarity between images. Alternative loss functions, such as perceptual loss or adversarial loss variants, could enhance the quality of generated images.

- **Data Diversity**: Limiting the dataset to only cats and dogs reduces the diversity of features the generator must learn, but it also constrains the model's ability to generalize, potentially affecting image quality.

V. Conclusion

This project explored the implementation of a custom Generative Adversarial Network tailored for the CIFAR-10 dataset, focusing on the generation of cat and dog images. By redesigning the discriminator to evaluate image similarity, the GAN deviated from traditional architectures, offering a novel approach to image generation. Although the generator successfully produced images with faint structural resemblance to real cats and dogs, the outputs were predominantly blurry, indicating room for improvement. Future work should consider enhancing model architectures, experimenting with advanced loss functions, and extending training durations to achieve higher image fidelity and realism.

VI. Prompts

- **Generator Training Prompt**: "Training Generator on CIFAR-10 dataset (cats and dogs) with latent dimension 100, using Dense and Conv2DTranspose layers, ReLU activations, and `tanh` output activation."
- **Discriminator Training Prompt**: "Training Discriminator as a Siamese Network on CIFAR-10 dataset (cats and dogs), evaluating similarity between generated and real images, using Conv2D layers, LeakyReLU activations, and sigmoid output activation."
- **Source Code Repository**: The complete source code for this project is available at https://github.com/smmk47/Custom-GAN-CIFAR10 [2].

VII. References

References

[1] Krizhevsky, A. (2009). *Learning Multiple Layers of Features from Tiny Images*. Technical Report, University of Toronto. Retrieved from https://www.cs.toronto.edu/~kriz/cifar.html.

[2] Smith, J. (2024). *Custom GAN Implementation for CIFAR-10*. GitHub. Retrieved from https://github.com/smmk47/Custom-GAN-CIFAR10.

[3] 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.

[4] Bromley, J., Guyon, I., LeCun, Y., Säckinger, E., & Shah, R. (1994). Signature Verification Using a Siamese Time Delay Neural Network. *Advances in Neural Information Processing Systems*, 6.

[5] Radford, A., Metz, L., & Chintala, S. (2015). Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. *arXiv preprint arXiv:1511.06434*.

[6] 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.