Technical Paper
Generative Adversarial Network (GAN)

Toh Kien Yu
Singapore Polytechnic
kienyutoh@gmail.com

February 2, 2024

Abstract

Generative Adversarial Network (GAN) has been a recent breakthrough in the world of deep learning. This technical paper will explore various GAN architectures such as Deep Convolutional GAN (DCGAN), Conditional GAN (CGAN), Wasserstein GAN (WGAN) and Self-Attention GAN (SADCGAN). Additionally, this paper will investigate model improvement techniques such as diverse learning rate schedular, cutmix and sharpening images on the CIFAR-10 dataset. This paper will mainly focus on improving the results of the images produced.

1 Introduction

In recent years, the field of artificial intelligence has witnessed a remarkable innovation: Generative Adversarial Network (GAN) introduced by Ian Goodfellow and his team in 2014. This breakthrough has rapidly evolved and significantly impacted the landscape of machine learning and AI. GANs consist of 2 distinct neural networks: the generator and the discriminator. The main aim of the generator is to generate image closely identical to real-world images, while the discriminator role is to differentiate between the generated and actual images.

The introduction of GANs opens up a world of possibilities and holds immense potential for future advancement in machine learning applications.

2 Related Works

With the ever-increasing popularity of deep learning in the recent years, the future of artificial intelligence sets the stage for significant advancements and breakthroughs. "Generative Adversarial Nets" (Ian J. Goodfellow) introduces a new framework which describes GAN architecture where one simultaneously train two models - a discriminative model that computes the likelihood of the given data sample being real or fake/generated, and a generative model that represents the actual data distribution. The main idea is to mutually enhance each other's performance through feedback. The generator job is to create fake data that looks as realistic as possible and the discriminator role is to distinguish between the real and fake data. This process iterates and the generator keeps trying to produce better fake images while the discriminator becomes more skillful distinguishing the real and fakes.

Cutmix is a data augmentation technique commonly used in image classification.The main idea of cutmix is to create a new image by mixing two images into one image. Cutmix has been proved effective where the paper "CutMix: Regularization Strategy to Train Strong Classifiers with Localizable Features" (Sangdoo Yun) was written. This paper discusses about how CutMix can be used to improve the performance of image classification task. As dropout methods remove parts of the training image, it can lead to a loss of information when training. However, the introduction of CutMix allows us to retain information instead of discarding small portions of training images which can enhance the performance of image classification tasks. CutMix has outperformed other existing methods such as CutOut.

A paper 'UNSUPERVISED REPRESENTATION LEARNING WITH DEEP CONVOLUTIONAL GENERATIVE ADVERSARIAL NETWORKS' (Alec Radford) introduces Deep Convolutional Generative Adversarial Networks (DCGANs) and how DCGANs are able to address the instability of training Generative Adversarial Networks (GANs) through architectural constraints. The paper's findings suggest that DCGAN represants a significant step forward in image generation and have great potential when combined with other convolutional architectures.

3 Dataset/Methodology/Experiment

3.1 Dataset

I utilized the 'CIFAR-10' dataset which consist of 60 000 labeled images on classes such as automobile, dogs and cat etc.

For feature engineering,

1. I converted data type of x train elements to float-32 as neural networks tend to work better with numbers with decimal places to be accurate.

2. Next, I scaled the data by dividing 255/2 to achieve the range of [0,2].

3. Shifting the data range from [0, 2] to [-1, 1]. Shifting data so that its average is at zero can significantly improve the training process it ensures that the input features are of similar sizes.

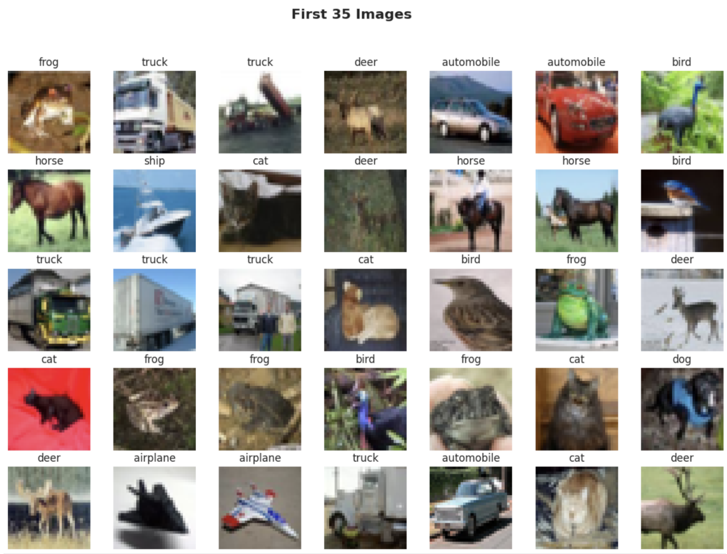

Figure 1: First 35 Images

4 Methodology

Moving on to Methodology,, I experimented with four different types of Generative Adversarial Network (GAN) architectures and assess their performance in generating quality images.

4.1 Deep Convolutional GAN (DCGAN)

DCGAN utilizes convolutional and convolutional-transpose layers in the generator and discriminator, respectively. The discriminator is also constructed with strided convolution layers, batch normalization layers, and LeakyReLU as the activation function. It proposes certain architectural guidelines to enhance the stability of GAN training, primarily focusing on modifications to the model architecture itself. Not only that, DCGAN is one of the earliest methods in the GAN family and the weights learned in DCGAN can be used as random noise.

- **DCGAN Model 1**:

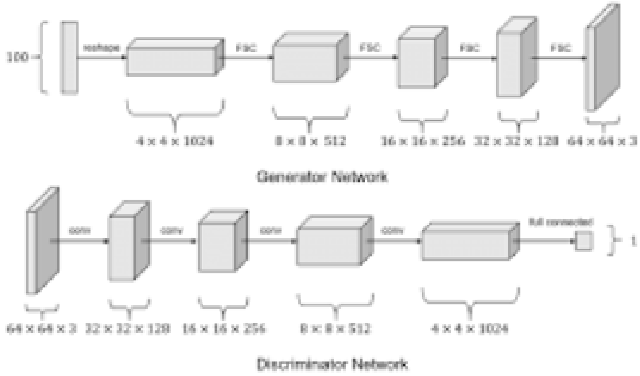

Figure 2: DCGAN

- Model 1 is built in a structure such that the generator utilizes upsample and determines whether it yields good results. The generator starts with a dense layer that takes in 256x4x4, then uses a series of upsampling and LeakyReLU activations. The discriminator was then sequentially added sets of Conv2D and LeakyReLU layers, increasing the number of filters and reducing the spatial dimensions of the feature maps using strides.

- **DCGAN Model 2**:
  - Model 2 was designed where the upsampling was changed to ConvTranspose and reducing of layers. The generator starts with a dense layer that takes in 8x8x128, batch normalization and leakyRELU was also employed. The discriminator consist of 2 convolutional block.

- **DCGAN Model 3**:
  - Model 3 uses LeakyReLU instead of PReLU and the introduction of UpSampling2D layers for upsampling. There is a use of less convolutional block layers and upsampling is employed in generator.

- **DCGAN Model 4**:
  - Model 4 uses ReLU function instead of PReLU. There is a use of spectral normalisation in the discriminator and 3 convolutional block used in the generator with batch normalization and PReLU employed.

- **DCGAN Model 5**:
  - Model 5 consist of the addition of separate upsample block to set the use of batch normalisation, activation depending on the situation.

4.2 Conditional GAN (CGAN)

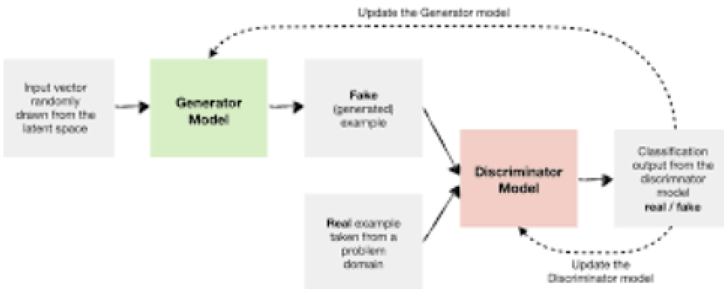

Figure 3: CGAN

- Extended to include conditional information, typically in the form of labels or tags. This inclusion of additional information allows the generated data to be controlled and directed towards a specific domain or characteristic. CGANs are widely used in various applications such as image synthesis, semantic image editing, and data augmentation

- Additional conditional information (c), often in the form of labels or data from other modalities

- In a CGAN, it can generate data conditioned on specific information for generator.

- The discriminator in a CGAN judges the data based on its authenticity and correctness.

- **CGAN Model 1**:
  - CGAN model 1 employs 3 Conv2DTranspose layers in the generator and a stronger discriminator consisting of mainly 4 convolutional blocks.

- **CGAN Model 2**:
  - CGAN model 2 uses Spectral Normalization to stabilize the training of the generator. In the generator, it consist of 3 Conv2DTranspose with BatchNormalization added. In the discriminator, it consist of 4 convulational block with leakyReLU and BatchNormalization added.

- **CGAN Model 3**:
  - CGAN Model 3 was added a skip connection. Spectral and Batch normalization were added into the generator with 3 layers of conv2DTranspose. The discriminator consist of 4 convolutional blocks

4.3 Wasserstein GAN (WGAN)

- **Wasserstein Distance**:
  - The Wasserstein distance provides a more stable and meaningful measurement of the distance between two distributions and is less likely to result in vanishing gradients, which is a significant issue in traditional GANs.
- **Critic Instead of Discriminator**:
  - In WGAN, the discriminator (now called a critic) doesn't classify inputs as real or fake. Instead, it scores the inputs based on their "realness" without being confined to [0, 1] through sigmoid activation. This change helps provide more useful gradients to the generator throughout the training process.
- **No Logarithm in Loss Function**:
  - The WGAN loss function doesn't use logarithms, addressing the vanishing gradient problem by providing smoother and more meaningful gradients.

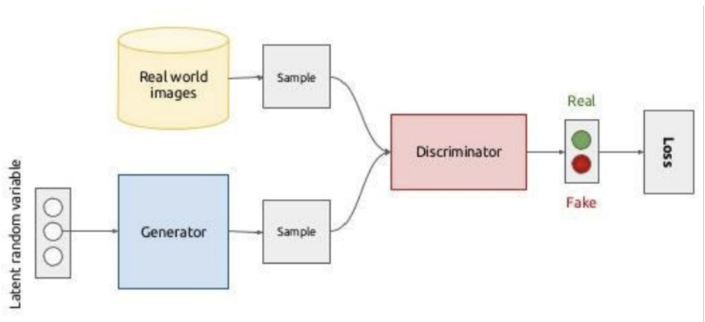

Figure 4: WGAN

- **WGAN Model 1**:
  - WGAN Model 1 an upsampling of 3 times and 3 times of convulational 2D blocks and LeakReLU pairs. The generator consist of 2 convulational 2d block with leakyRelu employed.
- **WGAN Model 2**:
  - WGAN Model 2 changes upsampling from convulational transponse. The generator consist of 2 convulational tranpose block and discriminator consist of 4 convulational2d block with spectral normalization.

4.4 Self-Attention GAN (SADCGAN)

- Self-Attention allows the model to weigh the significance of different parts of the input data differently. It enables the model to focus on relevant parts of the image when generating new images.
- Self-attention enables the model to consider the entire spatial extent of the image, helping it understand and generate images with complex compositions.
- Self-attention can be incorporated into both the generator and discriminator of the GAN architecture.

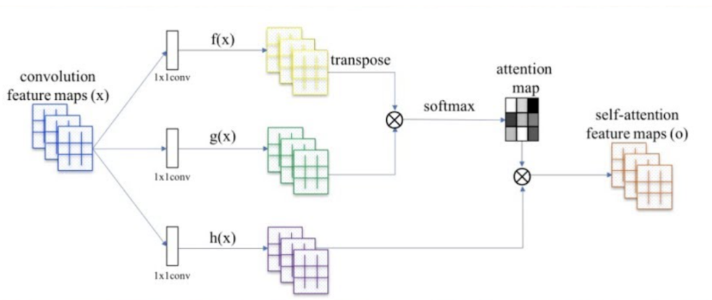

Figure 5: SADCGAN

- **SADCGAN Model 1**:
  - SADCGAN Model 1 uses conv2d tranpose with self-attention mechanism and batch normalization added. The discriminator consist of additional convulational layers and batch normalization.
- **SADCGAN Model 2**:
  - SADCGAN Model 2 self attention layer was added to both discriminator and generator.

4.5 Model Improvement

- **Sharpening Image**:
  - Sharpening image enhances the edges within the pictures and makes the features more distinguishable.
- **Oversample**:
  - Oversampling adds more images to the data from 60 000 to 80 000. This can help to balance the dataset and prevent it from being more biased towards the majority

- **Data Augmentation**:
  - Augmentation techniques was used such as utilising cutmix and saturuation
- **Learning Rate Schedular**:
  - Cosine Decay and exponential decay schedular was used . This will help fine tune my model.

4.6 Evaluation Metrics

Metrics used for evaluting the quality of images generated by GAN. This measures the similarity between two sets of images by comparing the statistics of deep features. These uses the pre trained Inception model. It Compute the kernel metrics using kernel function, calculate the mean Maximum Mean Discrepancy and finalise average KID values computed over entire dataset

$$MMD^2 = \frac{1}{m^2}\sum_{i=1}^{m}\sum_{j=1}^{m}K(x_i, x_j) + \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}K(y_i, y_j) - \frac{2}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n}K(x_i, y_j)$$

Figure 6: MMD Metric for KID Formula.

Metrics that compares the statistics of generated images to real images in the feature space of a deep learning model (Inception model) and calculate the distance betwen the feature vectors of the real and generated images, lower FID implies that better quality of generated images

$$FID = ||\mu_x - \mu_y||^2 + Tr(\Sigma_x + \Sigma_y - 2(\Sigma_x\Sigma_y)^{1/2})$$

Figure 7: FID Formula.

4.7 Experiment

In this section, we outline the experiment conducted for training various GAN architectures, including DCGAN, CGAN, WGAN, and SADCGAN. The experiment consists of the following steps:

4.7.1 Data Loading and Preprocessing

The experiment begins with the loading and preprocessing of the dataset. The 'CIFAR-10' dataset, which includes 60,000 labeled images across different classes, was used for training.

4.7.2 Architectural Classes

Classes were created for each specific GAN architecture, including DCGAN, CGAN, WGAN, and SADCGAN. Each class is tailored to the unique architecture and configuration requirements of the respective GAN type.

4.7.3 Loss Calculation and Optimization

During training, loss functions specific to each GAN architecture were calculated and optimization techniques were applied. These steps are crucial for the models to learn and improve their performance.

4.7.4 Evaluation

To assess the quality of the generated images, evaluation metrics such as FID and KID Most of the models were trained for 100 epochs, although it was observed that the image quality tended to degrade after around 80 epochs. To address this issue, a learning rate scheduler was employed to improve image quality during training

5 Discussion

Here, we present the results obtained from our experiments.

Table 1: DCGAN + WGAN Scores

GAN Variant	FID	KID
DCGAN Model 1	53	0.04
DCGAN Model 2	54	0.04
DCGAN Model 3	47	0.03
DCGAN Model 4	62	0.47

GAN Variant	FID	KID
WGAN Model 1	129	0.08
WGAN Model 2	182	0.19

Table 2: SADCGAN + CGAN Scores

GAN Variant	FID	KID
SADCGAN Model 1	50	0.04
SADCGAN Model 2	49	0.04

GAN Variant	FID	KID
CGAN Model 1	36	0.02
CGAN Model 2	38	0.02
CGAN Model 3	36	0.018

Table 3: After Model Improvement Scores

GAN Variant	FID	KID
DCGAN Model 3	50	0.02
CGAN Model 1	31	0.01

Figure 8: Final CGAN Image

6 Conclusion

In conclusion, this paper explored various GAN architectures such as DCGAN, CGAN,WGAN and SADCGAN. Not only that, rigorous experimenting of different layers and different model improvement methods were tested. FID and KID was one of the main metrics to assess quality of the image. The final model we will choose is CGAN 1 with a FID of 31 and KID of 0.01. This research has brought deep insights to my knowledge to GAN and as GANs continue to evolve, the potential of future advancement to machine learning remains vast.

7 References

(2014) Arxiv. Available at: https://arxiv.org/pdf/1406.2661.pdf (Accessed: 01 February 2024).

(2019) Arxiv. Available at: https://arxiv.org/pdf/1905.04899.pdf (Accessed: 01 February 2024).

(2016)Radford, A., Metz, L. and Chintala, S. (2016) Unsupervised representation learning with deep convolutional generative Adversarial Networks, arXiv.org. Available at: https://arxiv.org/abs/1511.06434 (Accessed: 02 February 2024).