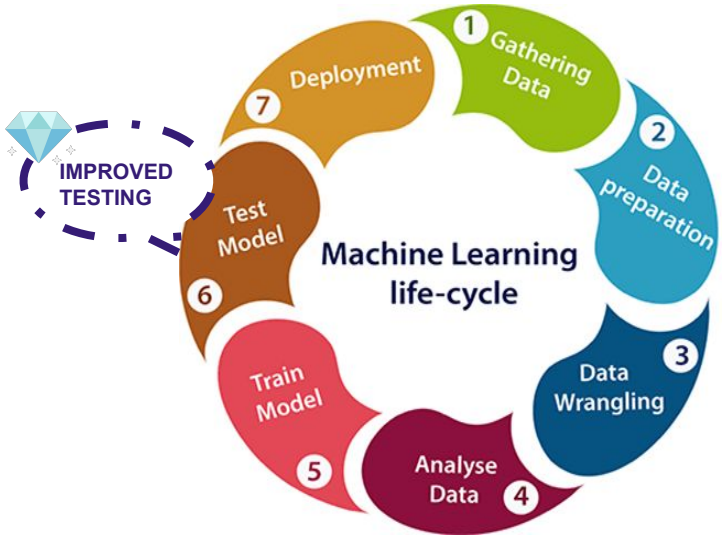
Evaluating Performance of GANs using Data Augmentation Techniques

Umang Sharma, Annamalai Chockalingam
Introduction to Deep Learning Systems - Final Project
New York University (NYU)

Executive Summary



1. Learnt to effectively train a GAN
2. Evaluated the performance of a GAN using various metrics, through both standard and novel approaches
3. Found issues and major gaps in the manner in which GANs are evaluated

Introduction & Background Information

Problem Motivation

Generator Adversarial Networks (GANs) are widely used in computer vision to generate images. Use cases include: Image Translation, Image Generation (Real/Fake). Also, GANs can be used for data augmentation when there is a large class imbalance in a dataset or to generate a larger dataset.

Problem: Although GANs are a popular method in computer vision, **techniques to evaluate the performance of these GANs have lagged behind**. **Some quantitative and qualitative metrics have emerged**, but these **metrics don't holistically capture** the true performance of GANs.

Current State:

- Assessment of GANs often are **qualitative**, and **may require human assessment** to **understand the quality** of GAN (expensive to do in practice)
- Commonly used quantitative measures are: **Inception Score and Frechet-Inception Distance (FID)**, have shortcomings, and **don't provide a holistic view** of GAN quality

How might we...

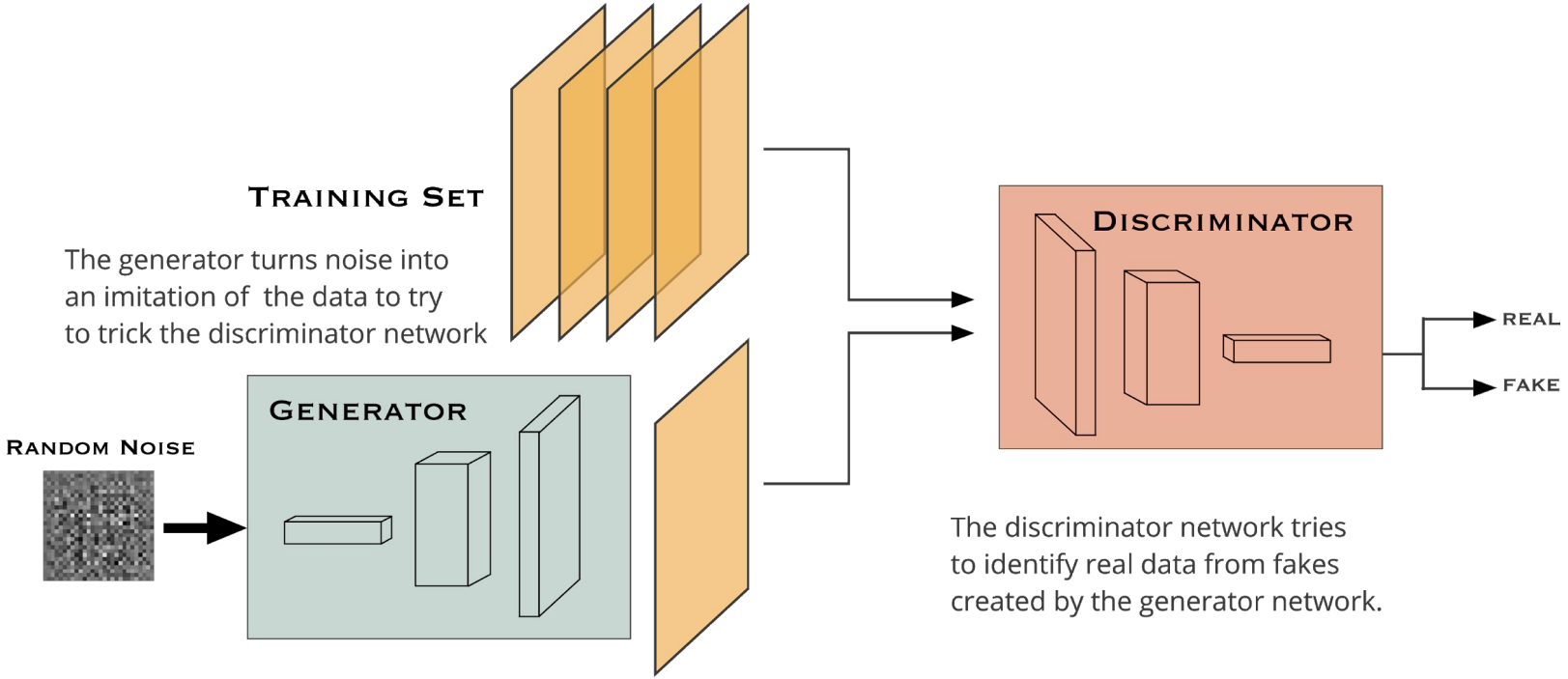
- **Quantitatively measure the performance** of a GAN to remove human bias, manual inspection.
- **Overcome shortcomings** of standard GAN metrics of Inception Score and Frechet-Inception Distance to **provide a holistic view quantitatively**

Key Outcomes:

1. To **evaluate the performance of GANs** specifically on dataset augmentation, using **GAN-Test Score** and **GAN-Train Score**
2. **Identify relationships** between this novel approach and other standard GAN metrics
3. **Explore effects of hyperparameters to tune a GAN**, improving performance of data augmentation

Overview of GANs

GANs are deep neural net architectures composed of competing neural networks. The models are trained by alternatively optimizing two objective loss functions. Generator learns to produce samples resembling real images, while discriminator learns to discriminate between real and fake data.These models have proven to generate good data.



Standard Methods to Evaluate GANs

Inception Score

1. KL-divergence between conditional and marginal label distributions over a dataset
2. Assesses image diversity through measurement of the distribution of dataset
3. $$\mathrm{IS}(G) = \exp\left(\mathbb{E}_{\mathbf{x} \sim p_g} D_{KL}(p(y|\mathbf{x}) \parallel p(y)) \right),$$

★ *Ranges from 0-∞. Ideally, score should be close to number of output classes. CIFAR-10 dataset* ***Inception Score 8.7***

Frechet Inception Distance (FID)

1. Wasserstein-2 Distance between multivariate Gaussians fitted to data embedded into a feature space
2. Relative measure evaluating the distribution of generated images with distribution of real images
3. $$FID = \|\mu_r - \mu_g\|^2 + T_r(\Sigma_r + \Sigma_g - 2(\Sigma_r \Sigma_g)^{1/2})$$

★ *Ranges from 0-∞. Ideally, this score should be as low as possible*

Drawbacks & Shortcomings

These standard metrics don't holistically capture the true performance of the dataset in a quantitative fashion

★ Impact between image diversity and quality not measured
★ Human judgement does not correspond with these metrics
★ Inception score is naive, and only measures diversity of dataset to output classes, rather than measuring true image quality
★ FID is a crude metric and does not capture fine details. Uses feature maps of inception network to calculate the FID Score. FID was originally designed to work on large datasets of natural scenes
★ Original inception networks don't inherently support grayscale images. Inception network may not provide good performance on complex datasets

Proposed Solution

Improved Measure to Evaluate GANs

In 2018 Shmelkov et al. proposed an alternate method; GAN-train and GAN-test scores to quantitatively measure the performance of a GAN

Using image classification techniques:

- **GAN-train approximates the diversity (recall) of the image**
- **GAN-test approximates the quality (precision) of the image**

GOALS:

1. To **evaluate the performance of GANs** specifically on dataset augmentation use-cases, using **GAN-Test Score** and **GAN-Train Score**
2. **Identify relationships** between this novel approach and other standard GAN metrics
3. **Explore effects of hyperparameters to tune a GAN**, improving performance of data augmentation

Technical Challenges

1. Finding the right set of hyperparameters, that enables the GAN to provide good data augmentation
2. Relationships between GAN-train, GAN-test, FID Score, Inception Score are weak and unreliable
3. Performance on MNIST and CIFAR-10 varies significantly, which required extensive hyperparameter tuning for each dataset
4. Datasets explored are not complex enough to investigate the nuanced drawbacks of Inception and FID Score
5. Compute resource constraints resulting in difficulties to obtain vast data to analyze

Approach

Used a three-step approach to evaluate the performance of GANs using inception score, frechet-inception distance (FID), GAN-test score, GAN-train. Further relationships between metrics, and hyperparameters to tune a GAN effectively are explored in this approach.

1 Augment data using a GAN

Use **DCGAN** architecture to construct a GAN for MNIST and CIFAR10 datasets

Tune hyperparameters such as number of **epochs**, **batch size**, **learning rate of discriminator & generator networks**, and **generator-discriminator ratio**

2 Evaluate performance of the generated dataset

Compute Inception Score & **Frechet-Inception Distance** through the construction of an inception network, and extracting relevant features and metrics

Create a **Resnet-32 model** architecture to obtain GAN test and GAN-train scores.

- **Train with GAN generated data** & **test on original dataset**. **Accuracy** of model on test set is **GAN-train score**
- **Train with original dataset** & **test with GAN generated data**. **Accuracy** of model on test set is **GAN-test score**

3 Analyze results of the experiment & explore relationships between metrics

Solution Diagram

INPUTS

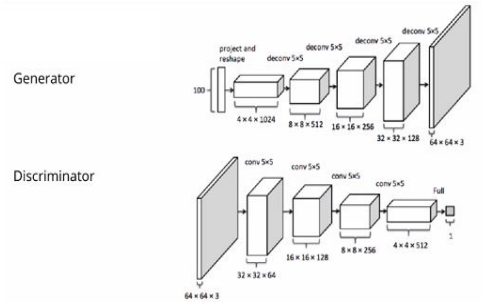
CIFAR10 (50K Images)

Hyperparameters

AUGMENTOR

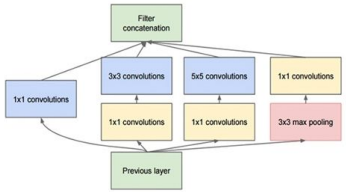
DCGAN Network

10K Generated Images

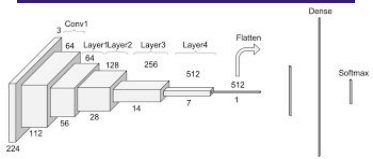


EVALUATE

STANDARD METRICS ENGINE (INCEPTION-V3 NETWORK)



GAN SCORE CALCULATOR (RESNET-32)



OUTPUTS

FID SCORE
INCEPTION SCORE

GAN-TRAIN SCORE
GAN-TEST SCORE

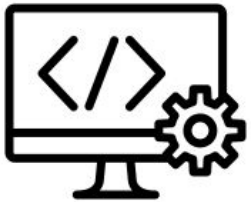
Implementation Details

INFRASTRUCTURE

GCP VM's based on debian Deep Learning VM images with 1 GPU (V100/P100 depending on test run) to run all 3 modules of the experiment (DCGAN Network, Standard Metrics Engine, GAN Score Calculator)

DATASET

CIFAR-10



SOFTWARE

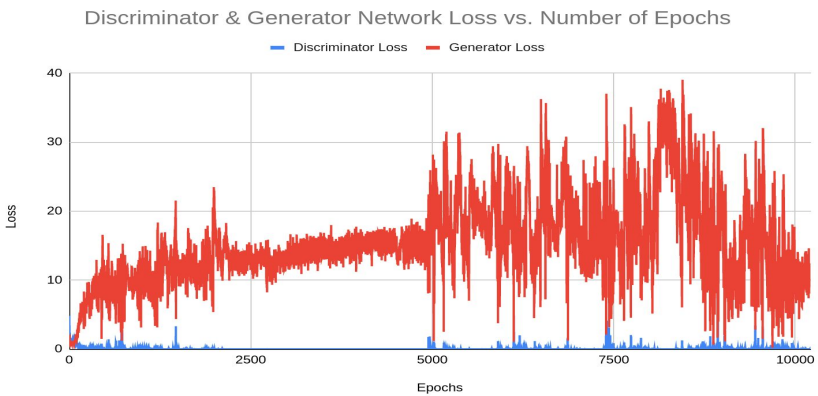
- Pytorch
- Tensorflow & Keras
- Numpy
- Matplotlib
- Scikit-Learn
- Other Standard Python Libraries

EXPERIMENT & OBSERVATIONS

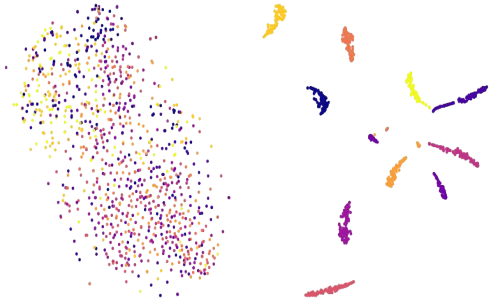
Experiment # 1

HYPERPARAMETERS

- Learning Rate = 1e-2
- Batch Size = 128
- Generator-Discriminator Ratio = 1
- Epochs = 10,200



t-SNE plot



METRICS

- Inception Score = 1.54 ± 0.035
- FID Score = 15.646
- GAN-train score = 0.1412
- GAN-test score = 0.1107

INSIGHTS

- Having an equal ratio of generator to discriminator from the start gives poor results
- Learning rate of 0.01 is too high for the models to learn
- This FID score is not low, and doesn't indicate true performance of network

Experiment # 2

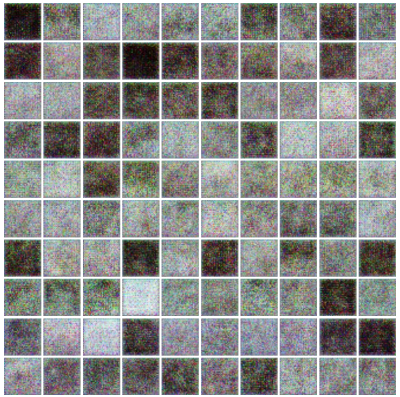
HYPERPARAMETERS

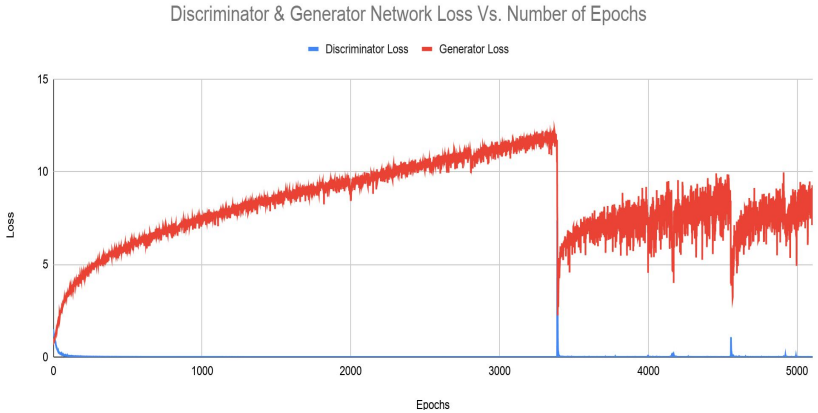
Discriminator LR = 1e-4
Generator LR = 1e-6

Batch Size = 128

Generator-Discriminator Ratio = 3

Epochs = 5,100





t-SNE plot

METRICS

Inception Score = 1.55 ± 0.047

FID Score = 22.413

GAN-train score = 0.1325

GAN-test score = 0.1061

INSIGHTS

- The generator needs a higher learning rate to produce meaningful images
- Lower learning rates lead to poor convergence
- A higher train ratio of discriminator vs generator leads to a better representation of the underlying data distribution.

Experiment # 3

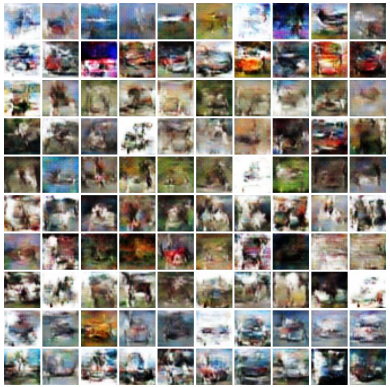
HYPERPARAMETERS

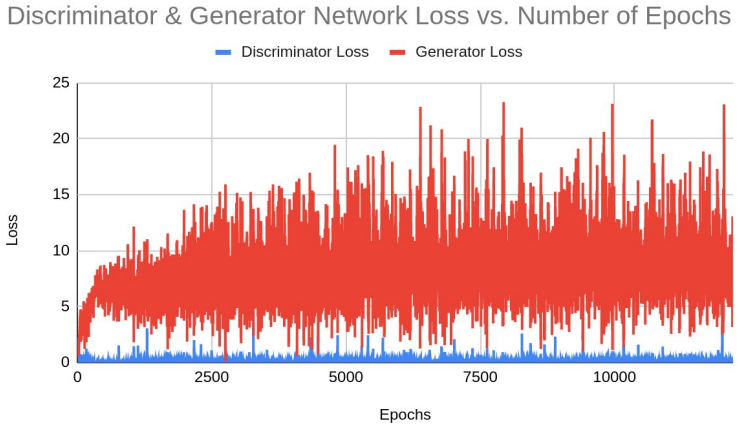
Discriminator LR = 1e-3
Generator LR = 1e-4

Batch Size = 128

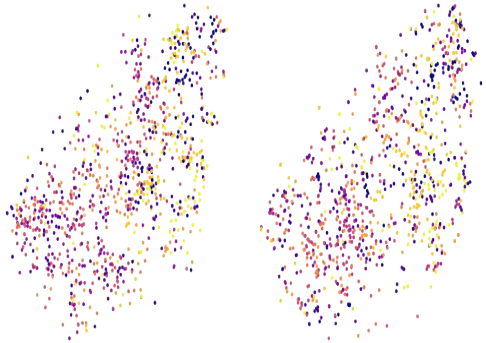
Generator-Discriminator Ratio = 3

Epochs = 12,200





t-SNE plot



METRICS

Inception Score = 2.277 ± 0.0748

FID Score = 11.03

GAN-train score = 0.1115

GAN-test score = 0.1258

INSIGHTS

This learning rate & gen/disc ratio learns quite well, training for more epochs may result in good performance

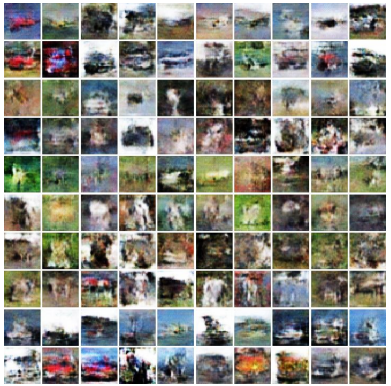
t-NSE plot indicates that distribution of data in both gen and disc networks are getting more equal

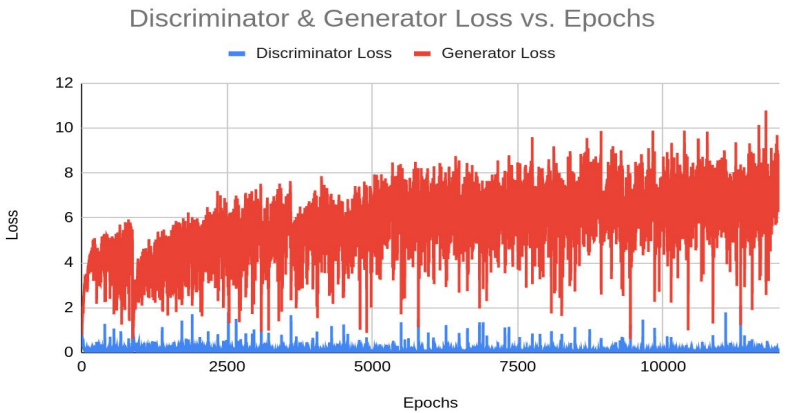
FID score is not low, and doesn't indicate true performance of network

Experiment # 4

HYPERPARAMETERS

- Learning Rate = 1e-4
- Batch Size = 128
- Generator-Discriminator Ratio = 5
- Epochs = 12000





t-SNE plot

METRICS

- Inception Score = 2.461 ± 0.0689
- FID Score = 10.58
- GAN-train score = 0.147
- GAN-test score = 0.136

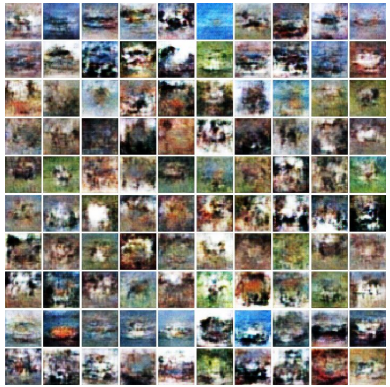
INSIGHTS

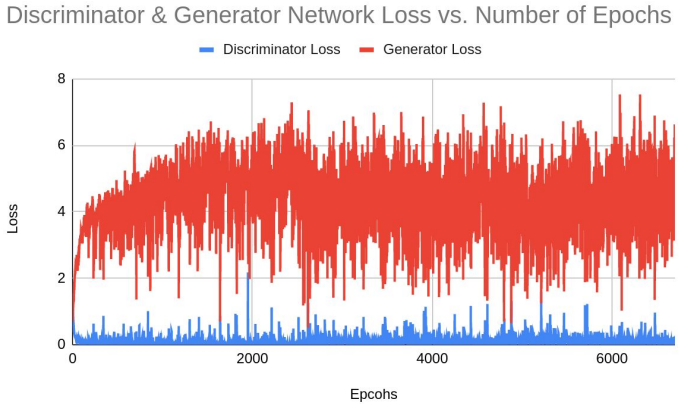
- Higher gen/disc ratio improves performance
- t-SNE plot indicates that distribution of data in both gen and disc networks are getting more equal
- The generator training loss is more stable compared to the experiment with a lower disc/gen train ratio

Experiment # 5

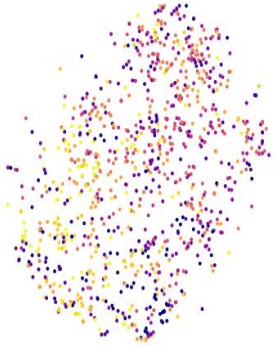
HYPERPARAMETERS

- Learning Rate = 1e-4
- Batch Size = 256
- Generator-Discriminator Ratio = 5 decayed to 2 after 3000 epochs
- Epochs = 6,000





t-SNE plot



METRICS

- Inception Score = 1.988 ± 0.03197
- FID Score = 12.046
- GAN-train score = 0.1176
- GAN-test score = 0.1026

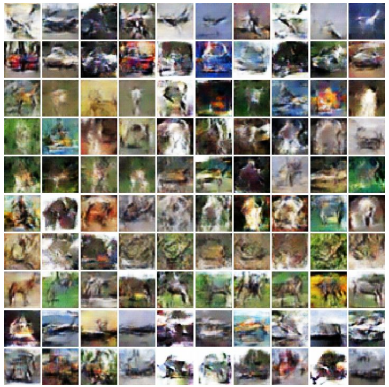
INSIGHTS

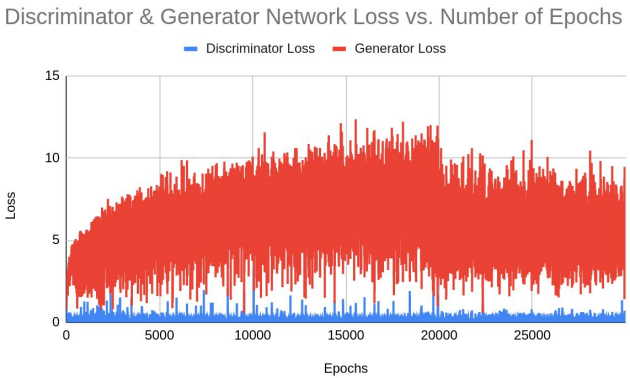
- Decaying the generator-discriminator ratio after a certain number of epochs produces better results
- Decreasing the number of epochs leads to worse performance.
- Whilst producing a diverse set of images, the realistic quality of the images is not very high.

Experiment # 6

HYPERPARAMETERS

- Learning Rate = 1e-4
- Batch Size = 64
- Generator-Discriminator Ratio = 5 decayed to 2 after 20000 epochs
- Epochs = 30,000





t-SNE plot

METRICS

- Inception Score = 3.12 ± 0.073
- FID Score = 9.732
- GAN-train score = 0.12
- GAN-test score = 0.10

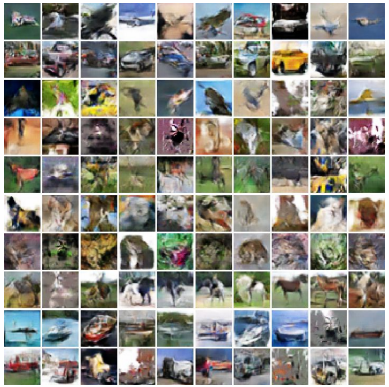
INSIGHTS

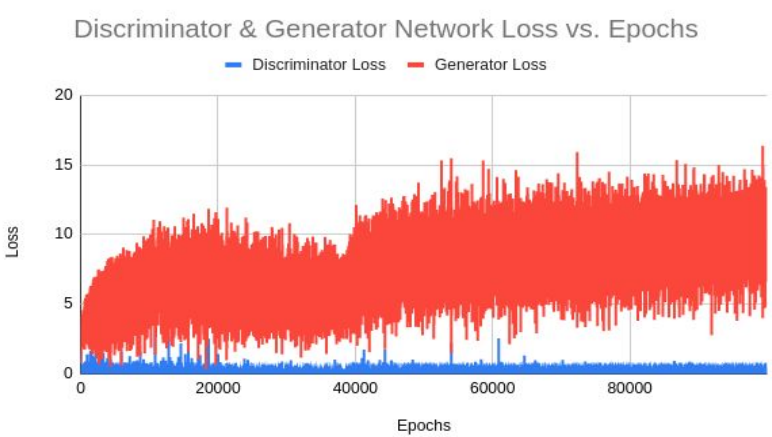
- Having a low batch size leads to instability in training of generator
- Higher the number of epochs, better the model captures the underlying data distribution
- This FID score is not low, and doesn't indicate true performance of network

Experiment # 7

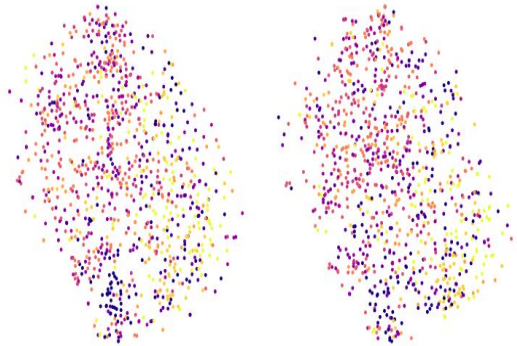
HYPERPARAMETERS

- Learning Rate = 1e-4
- Batch Size = 128
- Generator-Discriminator Ratio = 5 -> 4(5000) -> 3 (10000) -> 2 (20000)
- Epochs = 100,000





t-SNE plot



METRICS

- Inception Score = 3.84 ± 0.113
- FID Score = 7.772
- GAN-train score = 0.15
- GAN-test score = 0.14

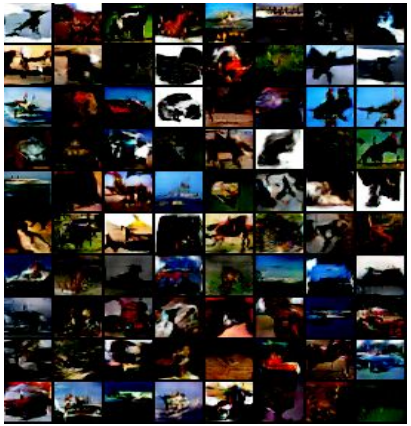
INSIGHTS

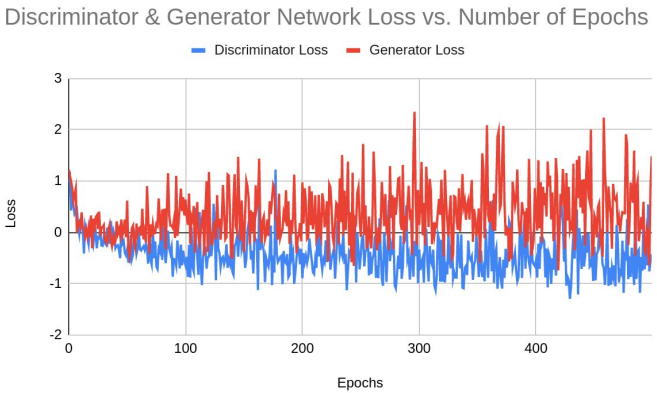
- Training for a large number of epochs further improves the results
- t-SNE plot indicates that increasing the number of epochs allow the model to mimic the dataset better
- FID score and inception scores are worse than previous experiments even though generated images are of better quality

Experiment #8

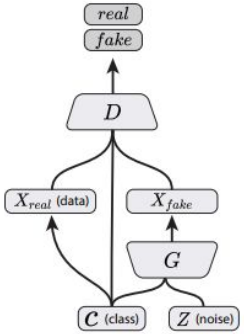
HYPERPARAMETERS

Learning Rate = 1e-4	Batch Size = 100	Generator-Discriminator Ratio = 1	Epochs = 500

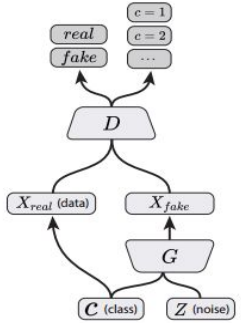




DCGAN



Auxiliary Classifier GAN (ACGAN)



METRICS

Inception Score = 2.261 ± 0.078	FID Score = 11.593	GAN-train score = 0.21	GAN-test score = 0.14

INSIGHTS

- ACGANs are better suited to deal with conditional inputs as required by the experimental setup.
- AC GAN generated dataset produces images of superior quality leading to higher GAN-train score.
- The auxiliary conditional aids the discriminator training, thus improving the overall performance of the model.

Summary of Results

TRIAL	HYPERPARAMETERS					METRICS			
	Learning Rate					Standard Metrics		GAN Score Approach	
	Discriminator	Generator	Batch Size	Iterations	Discriminator-Generator Ratio	Inception Score	FID Score	GAN train Score	GAN test score
1	1.00E-02	1.00E-02	128	10,200	1	1.54	15.646	0.14120	0.11070
2	1.00E-04	1.00E-06	128	5,100	3	1.55	22.413	0.13250	0.10610
3	1.00E-03	1.00E-04	128	12,200	3	2.277	11.03	0.11150	0.12580
4	1.00E-04	1.00E-04	128	12000	5	2.461	10.58	0.14700	0.13600
5	1.00E-04	1.00E-04	256	6000	5->2 (last 3000 iters)	1.988	12.046	0.11000	0.10260
6	1.00E-04	1.00E-04	64	30000	5->2 (last 10000 iters)	3.12	9.732	0.12000	0.10000
7	1.00E-04	1.00E-04	128	100000	5 -> 4(5000) -> 3 (10000) -> 2 (20000)	3.84	7.772	0.15000	0.10000
8	1.00E-04	2.00E-04	100	500 epochs	5 -> 4(5000) -> 3 (10000) -> 2 (20000)	2.26	11.593	0.21000	0.14000

GAN test & train scores calculated on resnet-32 (80 epochs, lr=0.001, batch size = 100) with 10K generated images

Outcomes of Experiment

Initial phases of training needs to train discriminator more, and the remaining training should focus on improving the generator networks performance

GAN test & train scores clearly indicate poor performance of the network, whilst FID score doesn't quantitatively illustrate the same

FID score is a deceiving metric & inception score is not holistic enough

All metrics are to be looked at holistically, to measure how good a GAN really is. Some level of qualitative measurement may be required

Learning Rate & Ratio of Generator-to-Discriminator are key to obtain good performance in a GAN

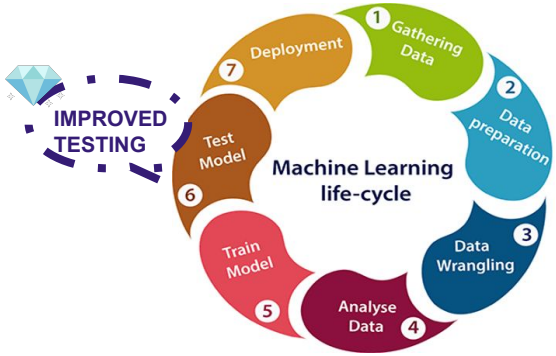
While GANs might produce diverse images, the quality of the images is not as realistic as ones in the original dataset, which is reflected by the GAN test scores

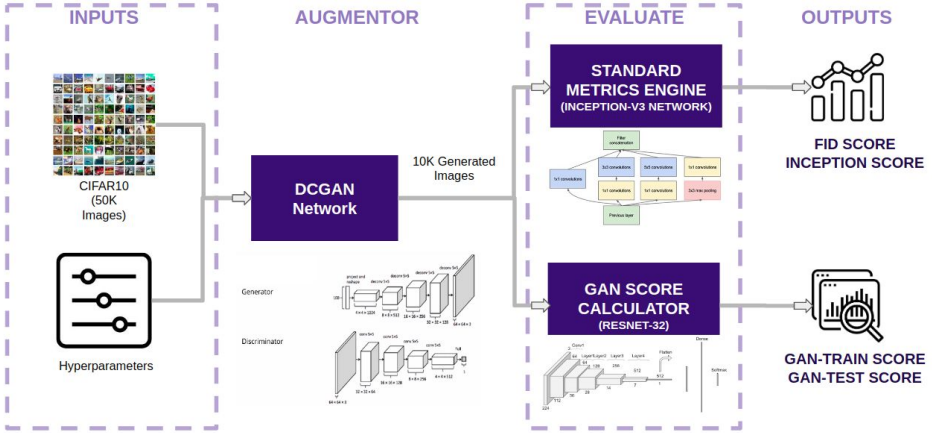
Training of networks to obtain better GAN test & GAN train may have significant impact on obtaining these scores (hyperparameter & architecture decisions)

GAN architectures suited to mimic natural images can generate more realistic images, leading to higher GAN-train scores. Experiment #8 shows shows improvement in generated images & GAN-train score.

Conclusion

Summary





Outcomes:

- In this experiment, all GAN test & train scores clearly indicate poor performance of the network, whilst FID score doesn't quantitatively illustrate performance
- FID score is a deceiving metric & inception score is not holistic enough. All metrics are to be looked at in holistically, to measure how good a GAN really is.
- Further advancement is needed in measuring performance of a GAN quantitatively, perhaps GAN-test & GAN-train score can address some issues
- Training of networks to obtain better GAN test & GAN train may have significant impact on obtaining these scores (hyperparameter & architecture decisions)

Immediate Improvements:

- Better GAN architectures (WGAN-GP, SNGAN) would generate more realistic images which would in turn lead to higher GAN-train and GAN-test scores
- 10K images used for training ResNet-32 while calculating GAN-test scores. Increasing the size of the dataset should lead to better performance.
- Hyperparameter tuning on ACGAN and DCGAN architectures can lead to more realistic and diverse samples which would lead to better GAN-train and test scores

References

1. Improved Techniques for Training GANs: https://arxiv.org/abs/1606.03498

2. How good is my GAN: https://arxiv.org/pdf/1807.09499.pdf

3. Evaluate GAN's: https://beyondminds.ai/advances-in-generative-adversarial-networks-gans/

4. Pros and Cons of GAN Evaluation Measures: https://arxiv.org/abs/1802.03446

Image References:

5. https://towardsdatascience.com/image-generation-in-10-minutes-with-generative-adversarial-networks-c2afc56bfa3b

6. https://www.kaggle.com/c/cifar-10

7. https://www.javatpoint.com/machine-learning-life-cycle

LINK TO GITHUB REPO