January, 2025

ROB313

Optional Work 1

Conditional Generative Adversarial Network

Name: **Sebti Adam**

This is the title of my project

Abstract

This project employs a Conditional GAN to generate images from the Fashion MNIST dataset. The model was evaluated on test data using innovative analysis techniques. Insights gained demonstrate the effectiveness of Conditional GANs in image synthesis.

1 Dataset Overview

Fashion-MNIST is a dataset of fashion images from Zalando. It consists of **60,000** samples for training and **10,000** for testing. Each example is a **28x28** grayscale image labeled into one of 10 distinct classes.

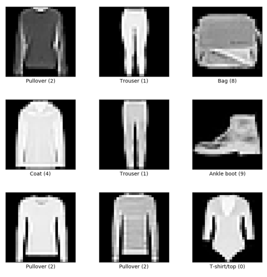


Figure 1: Fashion MNIST samples

2 Conditional GAN Architecture

The model has a simple architecture, it contains a generator and a discriminator, both conditioned on class labels. The generator creates images based on noise and class labels, while the discriminator distinguishes real images from generated ones, considering both the images and their labels.

Training The cGan has been trained using a low learning rate of 0.0001, to compensate epochs has been fixed to 200.

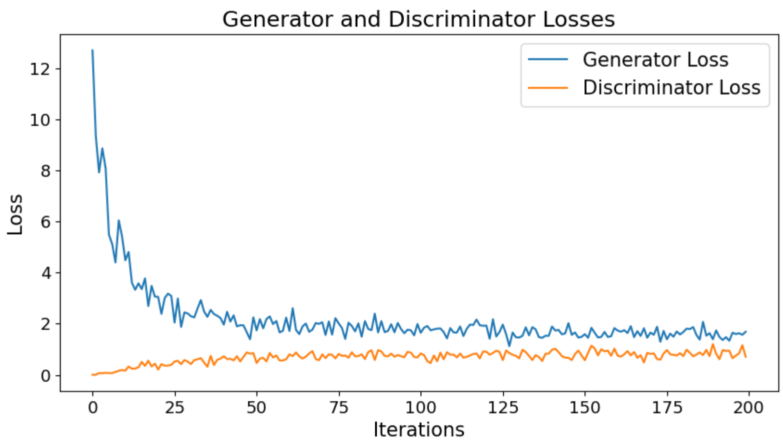


Figure 2: Training loss of the cGAN

The discriminator's loss increases slightly over time at the beginning because the generator starts producing more realistic images, making the discriminator's task harder. The generator starts with a high loss, as its initial outputs are poor. As it becomes better at fooling the discriminator, its loss decreases considerably.

Results Generated images : For most items an epoch of 200 seems to be an overkill as the generated images look fine, but still we have some exceptions. For example the in early epochs the model failed to reproduce correctly a bag confusing it with a t-shirt. A downside of raining the model for such long periods is encountering mode collapse problems.

Evaluation To evaluate the models performance, the test dataset has been been compared with the generated images with an FID score.

Epochs	**FID**
10	123.59
50	82.66
100	64.59
150	54.88
200	**50.62**

Training for longer periods did in fact improve the generation quality as the FID score keeps decreasing, but longer training means also an increased computational cost and time.

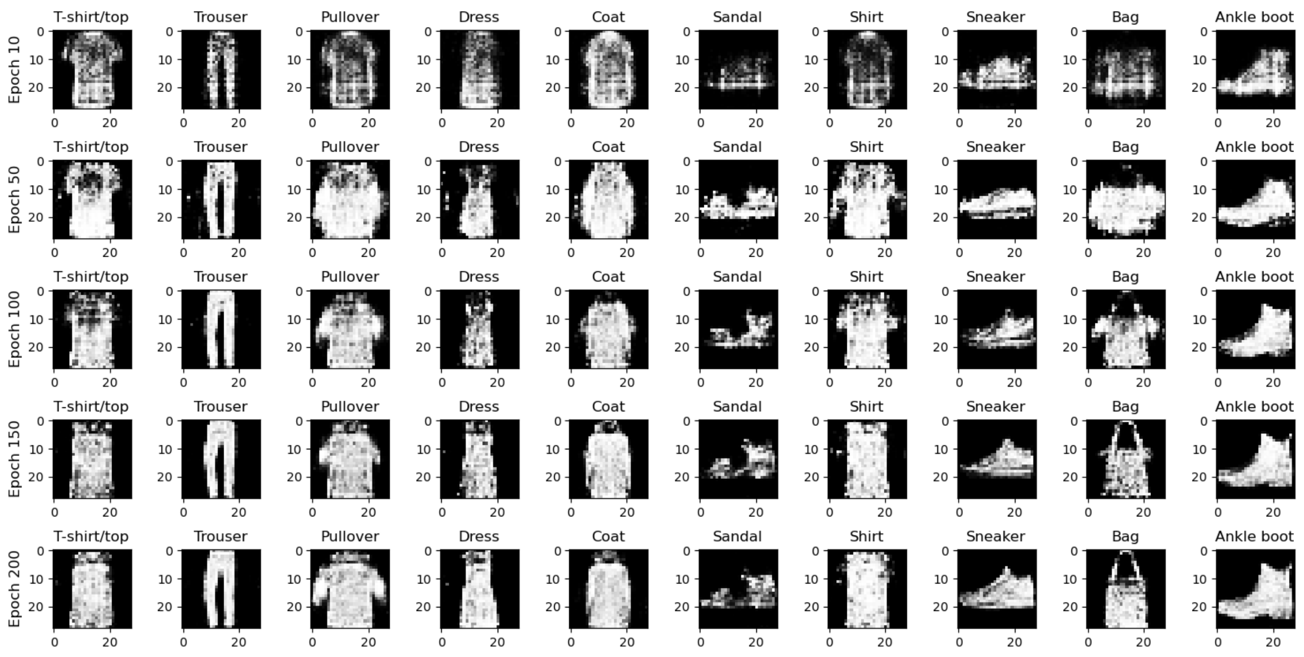


Figure 3: Generated Samples

The cGAN in order to be efficient is also required to generate diverse data closely representing the real distribution of the training data. Indeed one main problem of GAN architecture models is ttat they often suffer from mode collapse : the model sticks only to certain classes.

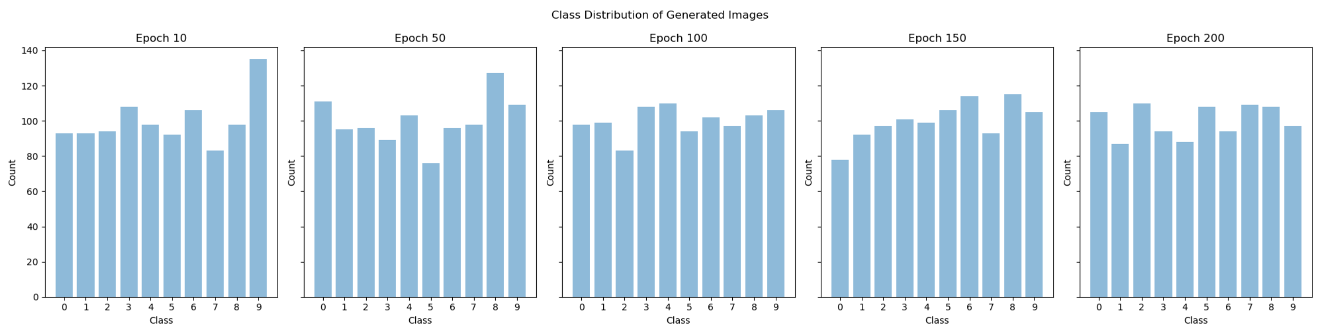


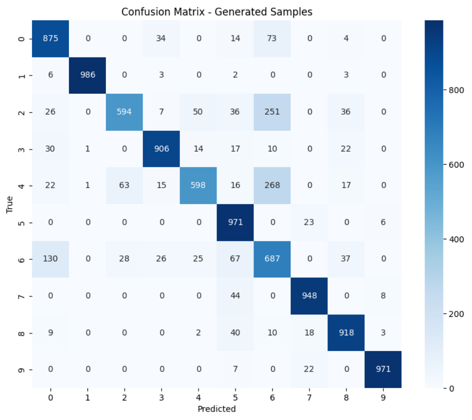
Figure 4: Generated classes distribution

Over 1000 samples the class distribution seems to be balanced with low error. As opposed as expected the large number of epochs did not lead to collapse mode.

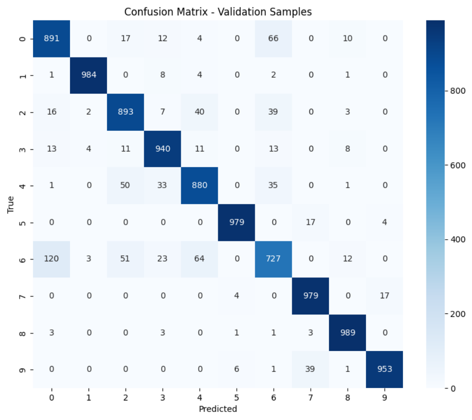
3 CGAN and Classifiers

Classifier Performance To further test the generated images quality, an image classifier has been trained on the Fashion-MNIST train dataset. Then the classifier performance has been tested on bot the real validation set and a generated validation set.

Dataset	Accuracy
Real Data	**92.15**
Generated Data	85.41



(a) Generated validation data



(b) Real validation data

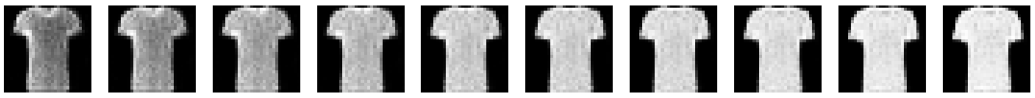
Figure 5: Confusion Matrix

The classification of the generated data fails because the model fails to correctly differentiate the shirt, coat and pullover classes. These 3 classes have quite similar representation that why the cGAN struggles to capture their distribution.

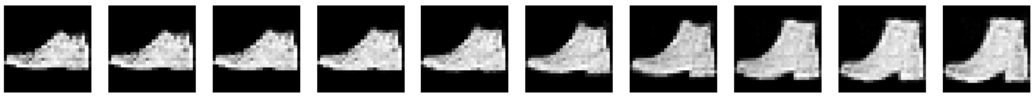
Data Augmentation We trained the same classifier on both synthetic and real data. The accuracy worsens we have now 79% of right predictions. This is due to the poor quality of the generated data, the CGAN needs further improvement to be used as a synthetic data generator.

4 Latent Space Exploration

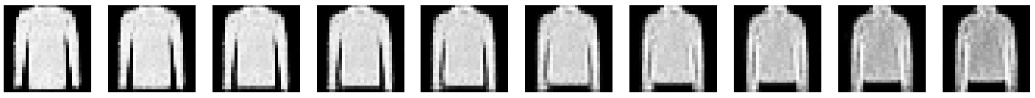
Exploration on the same class Changing the latent space for the same class label shows the transition of generated images for the sames class.



(a)



(b)



(c)

Figure 6: Latent space interpolation

Small changes in the latent space input can lead to a much different output as it's the case for the boot class.

Latent Space Clustering Visualizing the latent space clustering can give insight into how well-separated the different classes are in the latent space and how well the model captures class-specific features.

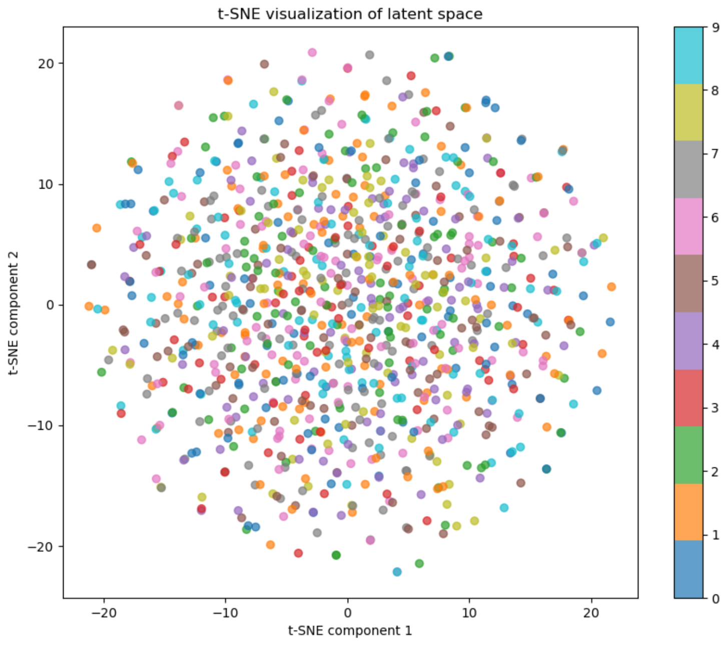


Figure 7: Latent space clustering

No clear separation is distinguishable between the different classes.