Conditional Generative Adversarial Networks

Parth Sagar Hasabnis, Master of Science in ECE[1]

Abstract

This document outlines the progress of the project - conditional GANs for the ECE 50024: Machine Learning Course.

1. Homework 2

Conditional Generative Adversarial Networks (cGANs) addresses a shortcoming of traditional GANS. Traditional GANs generate data by randomly sampling from a latent space and then using a generator network to transform that random input into a new data point. However, the generated data does not follow any given condition, and hence the output can be unpredictable. cGANs, on the other hand, enable the generation of data that is conditioned on a specific input or attribute.

Machine Learning and Deep Learning are the premier technologies in today's world, and both of them run on one single currency - data. Hence, the generation of data has become an important problem, as this data can be used to train autonomous vehicles, smart speakers and virtual assistants, wireless communications channels, and solve many more problems. Using cGANs, the generation of all kinds of data has become possible. We can generate conditioned data that was previously difficult or expensive to collect in the real world. For example – the performance and handling of an autonomous vehicle on a slippery road is an experiment that is difficult and dangerous to conduct in real life, but it is essential for autonomous vehicles to be trained on such conditions. Now we can generate artificial testing data using cGANs, and train vehicles using this artificial data.

There exist multiple implementations of cGANs on the internet, and different people have tried to implement them in their own methods. I shall start playing around with a PyTorch implementation trained on the MNIST dataset [1].

While I have created discriminator-like classifier networks, I am not familiar with the architecture of Generator networks. Hence my next step would be to learn about them and develop a traditional GAN before I move towards creating a cGAN.

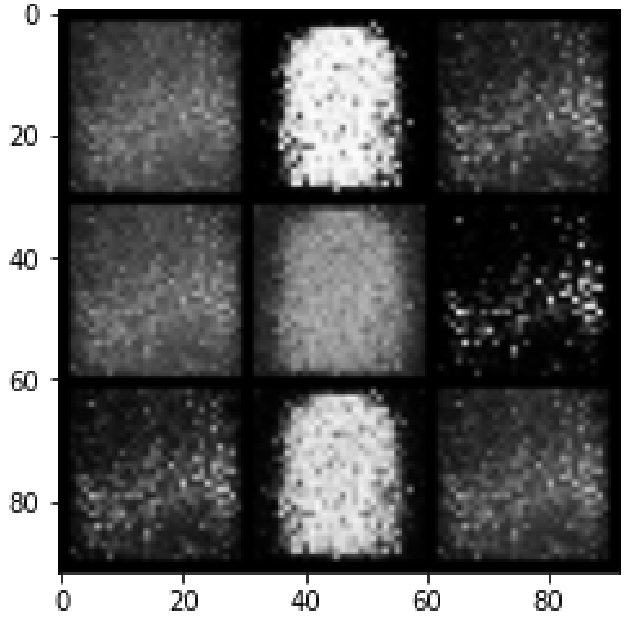
Figure 1. Boot

Figure 2. T-shirt

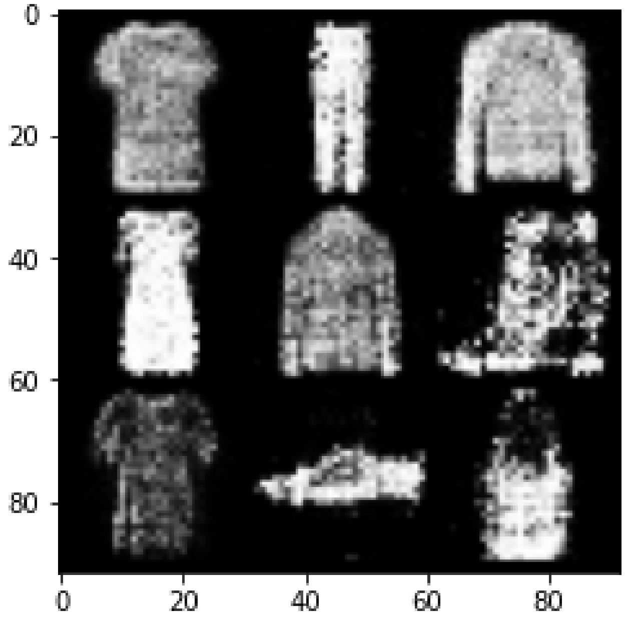
2. Homework 3

I had a few issues with installing the Cuda version of PyTorch on my local device, and in the end decided to go with the CPU implementation. I was successful in running the implementation on my device. This implementation of CGANS uses the MNSIT fashion data set to learn the different types of clothing articles from 28x28 pixel images (shown above).

The CGAN was trained for 30 epochs and has resulted in effective generation of new data from the dataset.



Epoch 1



Epoch 30

3. Acknowledgements

[1] https://github.com/qbxlvnf11/conditional-GAN