Machine Learning Final Project Report

Qiang Qin
UFID:99696682
qin.qiang@ufl.edu

Abstract

This paper firstly introduces both GAN and VAE these two generative models. Then it compares these two models and their results, talking about the reasons for what causes the different performance on these two models generating images based on MNIST dataset.

1 Introduction of Generative Models

Informally, generative models can generate new data instances. Formally, generative models capture the joint probability $p(X, Y)$, or just $p(X)$ if there are no labels. A generative model includes the distribution of the data itself, and tells how likely a given example is. There're several types of generative models, but this paper will only talks about GAN and VAE.

1.1 GAN

A generative adversarial network (GAN) has two parts: the generator and the discriminator.

The generator learns to generate fake data. The generated instances become negative training examples for the discriminator.

The discriminator learns to distinguish the generator's fake data from real data. The discriminator penalizes the generator for producing implausible results.

When training begins, the generator produces obviously fake data, and the discriminator quickly learns to tell that it's fake. But as training goes on, the fake result generated by generator becomes more and more real as the real data that can fool the discriminator. Finally, if the generator learns well, it should be more and more difficult for discriminator to tell the difference of the real data and fake data. The accuracy of both the generator and the discriminator should goes close to 50%, which means the discriminator has 50% chance to do a bad prediction on the given data.

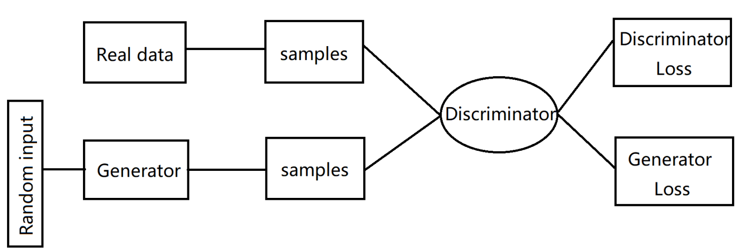

Figure 1: GAN Model

1.2 VAE

A variational autoencoder can be defined as being an autoencoder whose training is regularised to avoid overfitting and ensure that the latent space has good properties that enable generative process.

Just as a standard autoencoder, a variational autoencoder is an architecture composed of both an encoder and a decoder and that is trained to minimise the reconstruction error between the encoded-decoded data and the initial data. However, in order to introduce some regularisation of the latent space, we proceed to a slight modification of the encoding-decoding process: instead of encoding an input as a single point, we encode it as a distribution over the latent space.

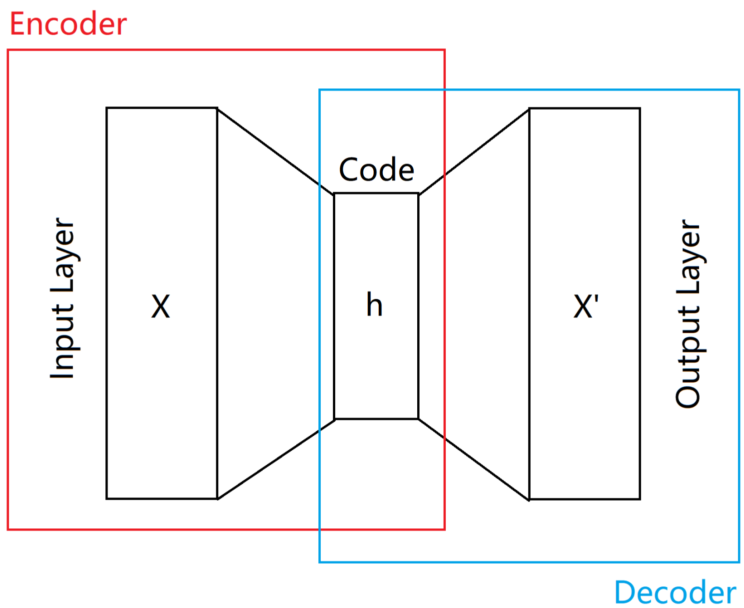

Figure 2: VAE Model

Following are steps when training a VAE network:

Firstly, the input is encoded as distribution over the latent space.

Secondly, a point from the latent space is sampled from that distribution.

After that, the sampled point will be decoded and the reconstruction error can be computed.

Finally, the VAE network will learn from the reconstrunction error backpropagated through the network.

2 My Models

2.1 GAN

In my GAN network, both the generator and the discriminator are neural networks. The generator output is connected directly to the discriminator input. Through backpropagation, the discriminator's classification provides a signal that the generator uses to update its weights.

The generate network has 6 layers, including an input layer, 4 hidden layers and an output layer. For the input layer, the number of its nodes is the same as the number of input data. For hidden layers, they have 128, 256, 512, 1024 nodes separately. For output layer, it has 28*28 nodes (the size of an image).

The discriminate layer has 4 layers, including an input layer, two hidden layers and an output layer. For the input layer, it has 28*28 nodes. For hidden layers, they have 512 and 256 nodes separately. As for the output layer, there is only one node.

These two networks are all full connected. I use Relu function as my active function for every hidden layer and use sigmoid function for the output layer of discriminate network.

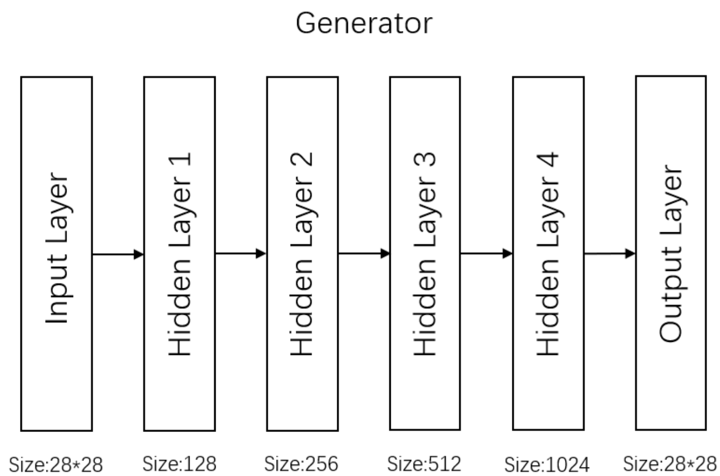

Figure 3: Generator

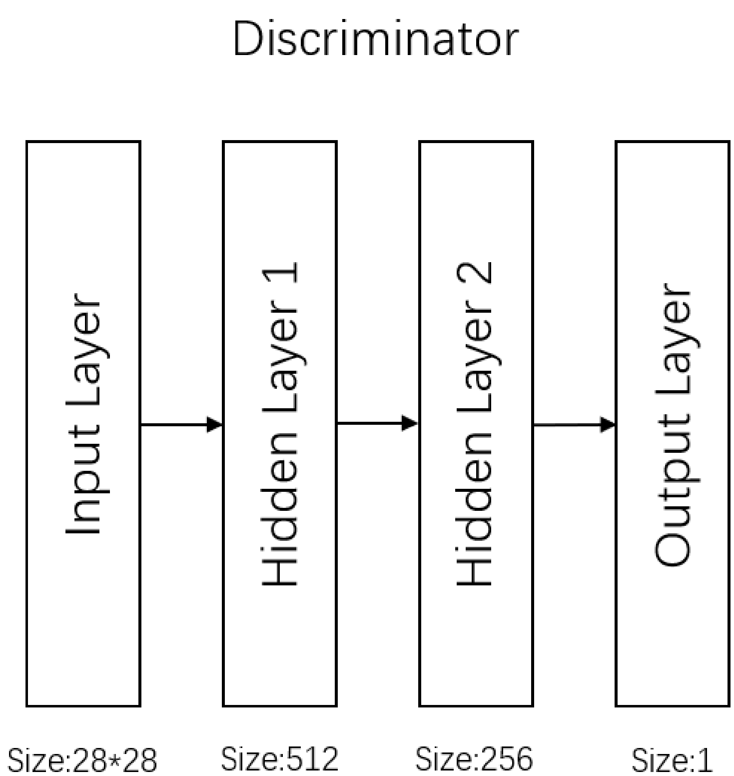

Figure 4: Discriminator

2.2 VAE

My VAE network also has two parts: encoder and decoder.

Encoder network has 4 layers, including an input layer, 2 hidden layers and an output layer. For input layer, it has 28*28 nodes. For hidden layers, they have 512 and 256 nodes separately. As for output layer, it has two nodes, which are the mean value and standard deviation.

Decoder network also has four layers, including an input layer, 2 hidden layers and an output layer. It's looks symmetric about the encoder. It has 2 nodes in input layer, which are the mean value and standard deviation. For the hidden layers, they have 256 and 512 nodes separately. As for the output layer, it has 28*28 nodes.

Between encoder and decoder there is another node z, which is the hidden variable connects generated by the mean value and standard deviation calculated by encoder and then used for decoder to generate new image.

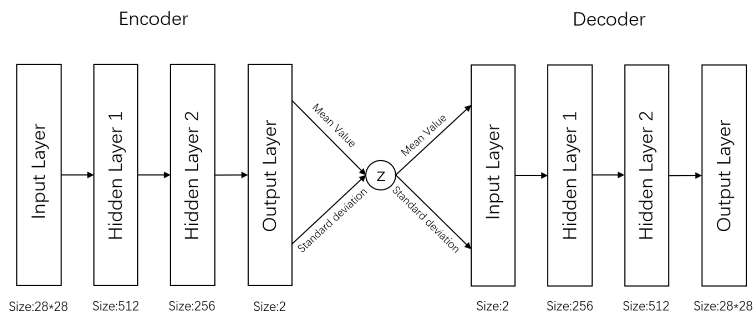

Figure 5: My VAE Network

3 Results

In this section I'll put some results generated by my GAN and VAE models.

3.1 Results from GAN

Figure 6 shows some good results generated by GAN, number 0-9.

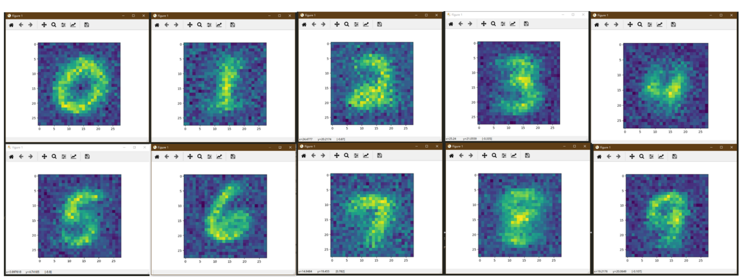

Figure 6: Results Generated by GAN

3.2 Results from VAE

Figure 7 shows some good results generated by GAN, number 0-9.

3.3 Analysis

From the results above we found that my VAE model generates blurrier but more stable data than GAN does.

Here's the difference between VAE and GAN and my understanding on this difference between outputs generated by VAE and thos generated by GAN.

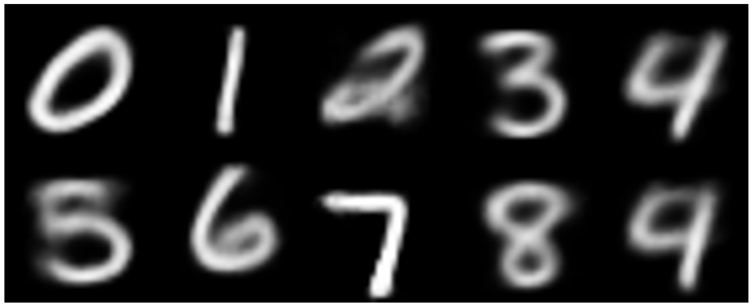

Figure 7: Results Generated by VAE

VAE learns a given distribution comparing its input to its output, this is good for learning hidden representations of data, but is pretty bad for generating new data. VAE learns an averaged representation of the data so that the output can become blurry.

While GAN use discriminators to measure the distance between the fake data and the real data. What it does is to distinguish the real data from the generated data. It receives some data as an input and returns a number between 0 and 1. 0 meaning the data is totally fake and 1 meaning it is totally real, other number means the probability of the input data comes from real set. The generators goal then is learning to convince the Discriminator into believing it is generating real data.

For my GAN model, I used multilayer perceptron for both generator and discriminator. This MLP structure doesn't work as well as CNNs. Convolution neural network works better on summarize features from difference between fake data and real data. If using CNNs for GAN, it will work better than VAE does. For my VAE model, it also uses MLP for both decoder and encoder. While since it extracts features from original data directly, it generated better outputs than my GAN model did. But since that VAE learns an average representation of data, the output can be blurrier.

4 Github Repository

`https://github.com/qinqiangdavid/CAP6610-FinalProject`

References

[1] Generative model. (2020, December 07). In Wikipedia. https://en.wikipedia.org/wiki/Generative_model

[2] Background: What is a Generative Model? (2019, May 24). Google Developers. https://developers.goog le.com/machine-learning/gan/generative

[3] Overview of GAN Structure | Generative Adversarial Networks. (2019, May 24). Google Developers. https://developers.google.com/machine-learning/gan/gan_structure

[4] Autoencoder. (2020, December 2). In Wikipedia. https://en. wikipedia.org/wiki/Autoencoder

[5] Rocca, J. (2020, July 7). Understanding Variational Autoencoders (VAEs) - Towards Data Science. Medium. https://towardsdatascience.com/understanding-variational-autoencoders-vaes-f70510919f73

[6] Sinitambirivoutin, E. (2020, June 23). An introduction to Variational Auto Encoders (VAEs). Medium. https://towardsdatascience.com/an-introduction-to-variational-auto-encoders-vaes-803ddfb623df

[7] What is the difference between Generative Adversarial Networks and Autoencoders? - Quora. (n.d.). Quora. Retrieved December 12, 2020, from https://www.quora.com/What-is-the-difference-between-Generative-Adversarial-Networks-and-Autoencoders