Cara Evangeline Chandra Mohan
190539421

Course Work 2: Unsupervised Learning by Generative Adversarial Network

1.What is the difference between supervised learning & unsupervised learning in image classification task?

Supervised Learning:

In supervised learning, to know the class of an image we require many training images along with it's labels. The training data consists of {(x, y)}, where x is the input and y is the target output.

Ex: ImageNet dataset, has 1.3 million training images and contains fine-grained and specific classes. To minimize the error rate in classifying an image, a supervised Convolutional neural network was developed which was very successful because of the large amount of labelled data present and further many other supervised algorithms like VGGNet, GoogLeNet were developed to progress in this field

Unsupervised Learning:

We need large amount of labelled training data for supervised learning task. This process is both tedious and costly to obtain. In unsupervised learning, the training data consists of {x}, ie. only the inputs are observed and there are no target outputs. Unlike supervised learning, unsupervised learning does not have explicit task-specific goals

2.What is the difference between an auto-encoder and a generative adversarial network considering (1) model structure; (2) optimized objective function; (3) training procedure on different components.

Model Structure:

Auto-encoder:

Autoencoders are a specific type of feed forward neural networks where the input to the encoder is the same as the output from the decoder. They compress the given input into a lower-dimensional feature in encoder and then reconstruct the output from this representation using decoder.

Generative Adversarial Networks:

In GAN, Generator is pitted against an adversarial network called Discriminator. Generator's objective is to generate data that is very similar to the training data. Discriminator's objective is to identify if the data coming from generator and original dataset is real or fake.

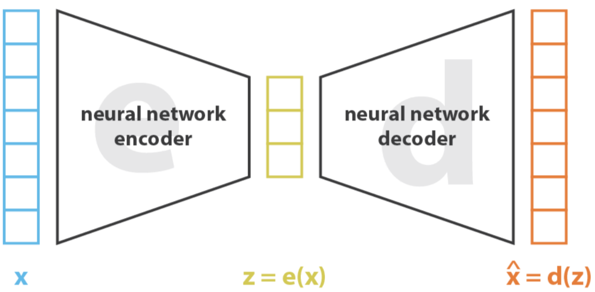


Fig: Auto-Encoder

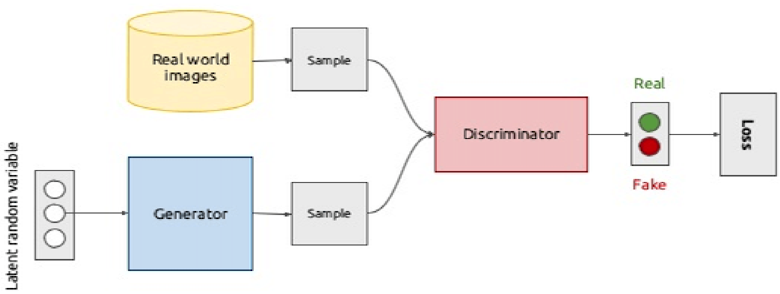


Fig: Generative Adversarial Network

Cara Evangeline Chandra Mohan
190539421

Optimized Objective Function:

Auto-encoder:

$\min(\text{cost}) = |x - \hat{x}|^2 = |x - d(z)|^2 = |x - d(e(x))|^2$

Generative Adversarial Networks:

$$\min_G \max_D V(D, G) = \mathbb{E}_{\boldsymbol{x} \sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z} \sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))].$$

Training procedure:

Auto-encoder:

- Autoencoder neural network is an unsupervised learning algorithm that applies backpropagation, setting the target values to be equal to the inputs and learns automatically from the data example.
- Encoder learns to reduce the input dimensions and compress the input data into the encoded representation.
- Decoder learns to reconstruct the data from the encoded representation

Generative Adversarial Networks:

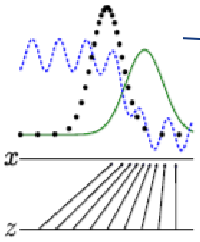
- Generator takes in random noise and returns a fake image. It learns until it can fake the discriminator by showing an image similar to the real image. It gets the discriminator's Real or Fake classification for generator output. Then it calculates the loss from discriminator classification, backpropogates through both the discriminator and generator to obtain gradients and atlast uses gradients to change only the generator weights.
- Discriminator takes in both real and fake images and returns probabilities between 0 and 1, where 1 represents a prediction of authenticity and 0 represents fake. The discriminator loss penalizes the discriminator for misclassifying a real instance as fake or a fake instance as real and updates its weights through backpropagation from the discriminator loss via the discriminator network.

3.How is the distribution $p_g(x)$ learned by the generator compared to the real data distribution p(x) when the discriminator cannot tell the difference between these two distributions?

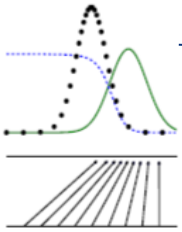
GANs are trained by simultaneously updating the discriminative distribution so that it discriminates between samples from the data generating distribution p(x) from those of the generative distribution p_g (x)

Description of the graph : The lower horizontal line is the domain from which z is sampled. The horizontal line above z is part of the domain of x. The upward arrows show how the mapping x = G(z) imposes the non-uniform distribution p_g on transformed samples. G contracts in regions of high density and expands in regions of low density of p_g.

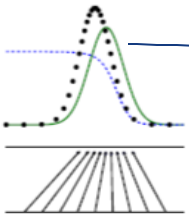
Cara Evangeline Chandra Mohan
190539421



p_g is similar to p(x) and D is a partially accurate classifier.



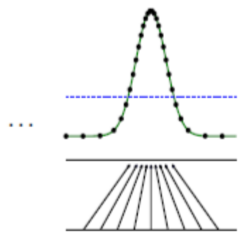
D is trained to discriminate fake samples from real samples belonging to the p(x) distribution



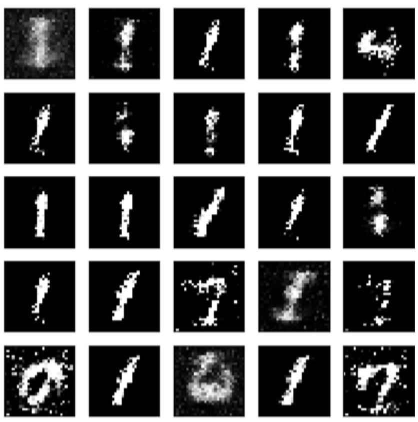
G is updated and the gradient of D is used to force G(z) to flow in a region which has a similar distribution to p(x)

= p_g
. . . . . = p(x)
.... = Discriminator distribution

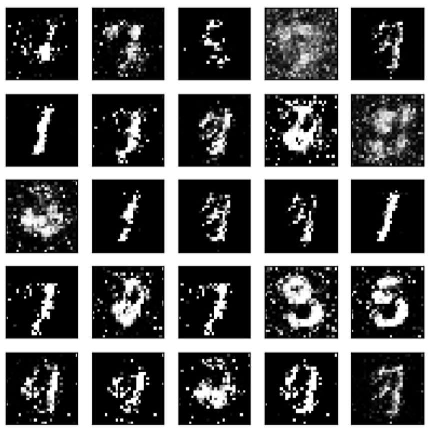
After training the algorithm for n number of times GAN attains a point where p_g = p(x) and you cannot improve the model any better and D(x) = 0.5



4.Show the generated images at 10 epochs, 20 epochs,50 epochs,100 epochs by using the architecture required in Guidance.



Epoch 10



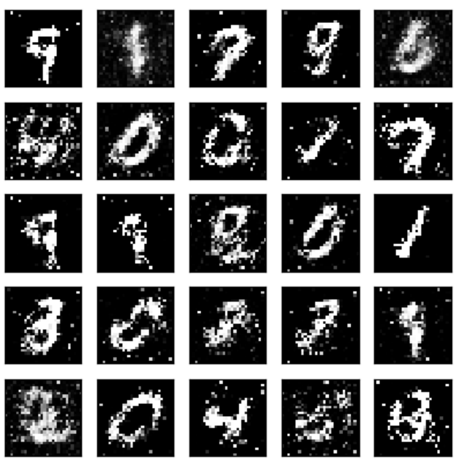
Epoch 20

Epoch 10:
loss_d = 0.927
loss_g = 1.264

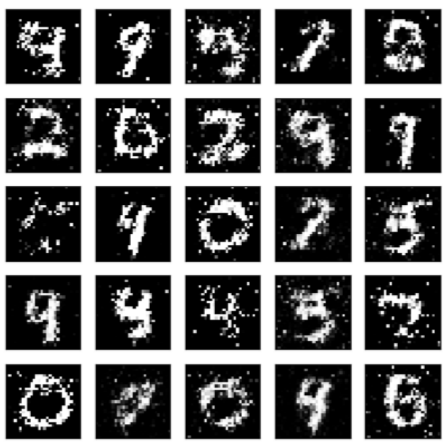
Epoch 20:
loss_d = 1.262
loss_g = 1.826

Epoch 50:
loss_d = 1.337
loss_g = 0.708

Epoch 100:
loss_d = 1.347
loss_g = 0.673



Epoch 50



Epoch 100