ECS795P Deep Learning and Computer Vision, 2022

Course Work 2:
Unsupervised Learning by Generative Adversarial Network

1. **What is the difference between supervised learning & unsupervised learning in image classification task? (10% of CW2)**
   In image classification the images are classified into different classes, namely, either given (i.e. supervised) or extracted from the training dataset(unsupervised).

   The model in supervised learning requires a large amount of labelled training data. These labels are the classes into which the model must classify the images. The labels are generated either manually (annotated by humans) or automatically using a generative model. Based on the labelled training dataset, the model then learns a decision boundary to classify the images into their respective classes. Supervised learning is costly and may be biased. For supervised learning labelled dataset $\{(x_i,y_i)\}_{i=1,2,3,4\ldots n}$, where x_i represents an instance and y_i represents the label. The goal is to learn a function f: $x_j \rightarrow y_j$ that can calculate y_j for an unknown x_j.

   Unsupervised learning, on the other hand, does not require labelled training data. Unsupervised learning aids in understanding the underlying similarities between images and then classifying them. These models are naturally suited to dealing with large, unstructured image repositories. In unsupervised learning, we have an unlabeled dataset $\{(x_i)\}_{i=1,2,3..n}$, where x_i is an instance. The goal is to learn the structure of x_i so that you can cluster an unknown instance x_j.

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. (10% of CW2)**

   An autoencoder learns to efficiently represent the input while also learning how to reconstruct the input from its compressed form. A generative adversarial network, on the other hand, learns how to generate data that is indistinguishable from real data. The difference based on various parameters is stated as follows below:

Autoencoder	Generative adversarial network
(1) model structure	
Autoencoders are networks that learn to generate an output "r" that is as close to the input image as possible. It is made up of two networks: an encoder network and a decoder network. The encoder network generates a compressed representation of the input image in the form of a latent vector "h." The latent vector is then fed into the decoder network, which attempts to recreate the original image. Input image → Encoder Network → Latent vector (h) → Decoder Network → Output image	A generative adversarial network consists of two network as well but these networks are a generator and a discriminator network which are in a two player minimax game. GANs try to generate samples from a simple distribution. The generator tries to fool the discriminator by generating real looking images and discriminator tries to decide if the generated image is real or fake. Input image → Generator Network → Image1, Image2 → Discriminator Network → real/ fake
(2) optimized objective function	
The autoencoder's objective function is a square of the pixel-wise difference between the predicted image and the groundtruth image, as shown below. $L(x, y; \theta) = -\frac{1}{M}\sum_{i=1}^{M} \|\|x_i - r_i\|\|^2$	The GAN's objective function is shown below. The GAN's goal is a two-player minimax game in which the discriminator tries to maximise D(x), which is the probability when given the real image, as well as D(G(z), which is the probability when given a fake image. By increasing D(G(z), the generator attempts to reduce the discriminator's chances of making a mistake. $\min_{\theta_g}\max_{\theta_d}\left[\mathbb{E}_{x\sim p_{data}}\log \underbrace{D_{\theta_d}(x)}_{\text{Discriminator output for real data x}} + \mathbb{E}_{z\sim p(z)}\log(1-\underbrace{D_{\theta_d}(G_{\theta_g}(z))}_{\text{Discriminator output for generated fake data G(z)}})\right]$ • Remark: Discriminator outputs a value between 0 and 1 to denote the likelihood of being real
(3) training procedure on different components	
In autoencoders, the error is back propagated, and both networks, encoder and decoder, are simultaneously updated. The	In the case of GANs, the training procedure for *G* is designed to increase the likelihood of *D* making a mistake. Both networks in GANs are

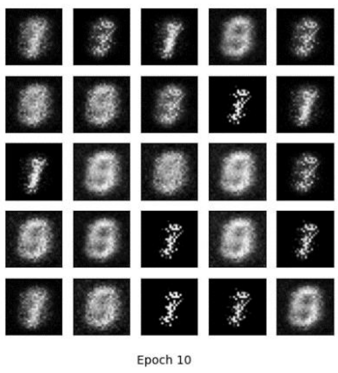
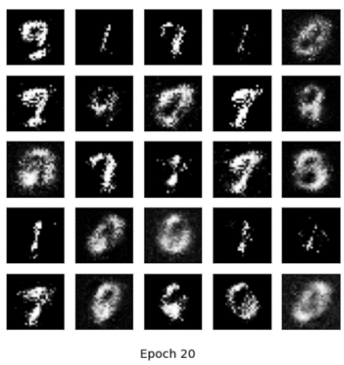
training procedure entails minimising the difference between the generated and input images.	**not updated at the same time. The discriminator's stochastic gradient is updated ascendingly, while the generator's stochastic gradient is updated descending.**

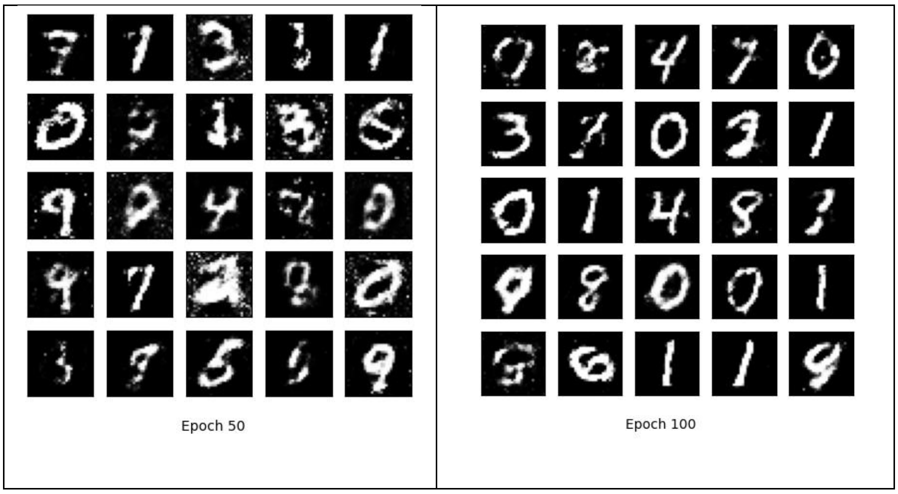
3. **How is the distribution $p_g(x)$ learned by the generator compared to the real data distribution $p_{data}(x)$ when the discriminator cannot tell the difference between these two distributions? (10% of CW2)**

When the discriminator cannot distinguish between $p_g(x)$ and $p(x)$, it indicates that the generator has learned the distribution of the real data and p_g=p_{data}. At that point, the discriminator has a 50% chance of making the correct decision, shown as follows: D*G(x) = $p_{data(x)}/p_{g(x)}+p_{data(x)}$. The global minimum of the virtual training criterion C(G) is achieved if and only if p_g = p_{data}, according to theorem 1 in [1]. C(G) reaches the value -log 4 at that point. The theorem states that C(G) = -log(4) + 2* JSD(p_{data} || p_g) and that the Jensen–Shannon divergence between two distributions is always non-negative and only zero when they are equal. As a result, when p_g = p_{data}, the global minimum of C(G) is given by C* = - log(4).

4. **Show the generated images at the 10th epoch, the 20th epoch, the 50th epoch, the 100th epoch by using the architecture required in Guideline. (10% of CW2)**

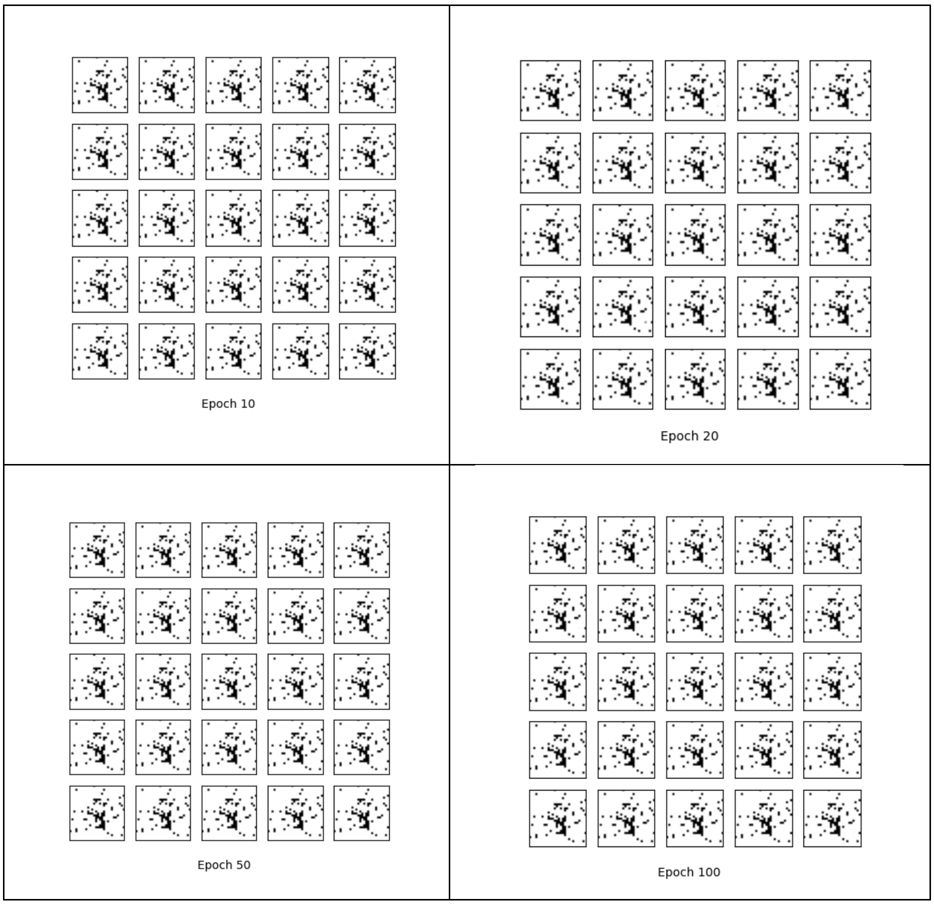
Batch size: 100, learning rate: 0.0002



Epoch 50 Epoch 100

Batch size: 256, learning rate: 0.01

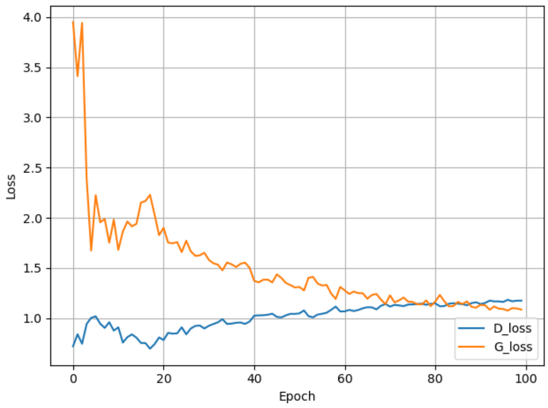


Epoch 10 Epoch 20

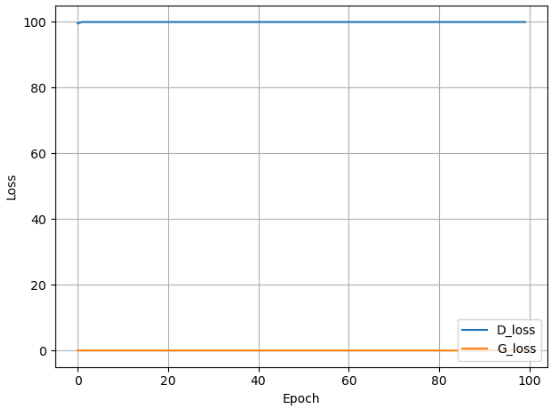
Epoch 50 Epoch 100

5. **Plot the loss curve during training.** **(10% of CW2)**

   The dropout is set to 0.3, learning rate = 0.0002, batch size= 100



The dropout is set to 0.3, learning rate = 0.01, batch size= 256



References:

(1) Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. and Bengio, Y., 2014. Generative adversarial nets. In Advances in neural information processing systems (pp. 2672-2680).