ECS795P Deep Learning and Computer Vision, 2020

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 supervised learning for image classification tasks, we require a large set of annotated images, while unsupervised learning there is no such requirement. The annotation is done manually and is very time consuming. In case of supervised learning, the training data consists of a set of images with labeled/annotated information. The classifier which can be a generative or a discriminative function learns the features from given training data and generates a model to predict on unlabeled data. The only requirement in supervised learning is a huge amount of annotated images, which is the biggest challenge. In unsupervised learning, the training data is not annotated and the learning function tries to find the pattern in the training data that best describes the data, this is also called a cluster. There are many ways to improve the accuracy of the clusters, one such way is to use the output as feedback. GANs are unsupervised learning algorithms that use the feedback of supervised learning to improve the learning function of unsupervised learning algorithms.

Thus in supervised learning, we have labeled dataset $\{(\mathrm{x_i},\mathrm{y_i})\}_{\mathrm{i=1,2,3,4...n}}$ where x_i represents an instance and y_i represents the label. The task is to learn a function $f : x_j \rightarrow y_j$ which can determine y_j for an unknown x_j.

And in unsupervised learning, we have an unlabeled dataset $\{(\mathrm{x_i})\}_{\mathrm{i=1,2,3..n}}$ where x_i represents an instance. The task is to learn the structure of x_i in order to cluster an unknown instance x_j.

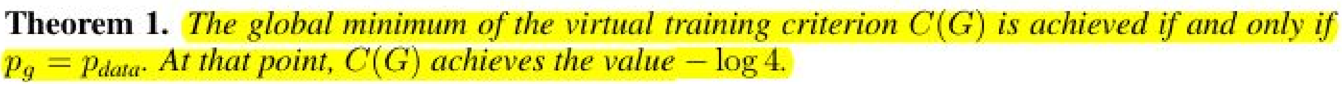
2. **What is the difference between an auto-encoder and a generative adversarial network considering (1) model structure; (2) optimized objective function; (3) training procedure on different components. (10% of CW2)**

	Auto Encoders	GAN
Model Structure	Consists of two networks encoders and decoders, Role of encoder is to map higher dimension input data x_i to latent code h_i Role of decoder is to map h_i to output r_i Objective is to learn x_i and generate r_i as close as possible Input x_i → Encoder Network f → Latent code h_i → Decoder Network g → Output r_i	Consists of two network Generator and discriminator, Role of Generator is to learn the distribution of the sample space *p(z)*. Role of a Discriminator is to learn to differentiate between real distribution *p(x)* and *p(z).* x → Generator Network → p(z) p(z),p(x) → Discriminator Network → real/ fake
Optimised Objective function	Learn encoder f and decoder g as (f,g): x → h → r Objective function $L(x, y, \theta) = -\frac{1}{M}\sum_{i=1}^{M}\left\Vert x_i - r_i\right\Vert^2$	Min Max of generator and discriminator network. Generator Network, minimise the objective such that D(G(z)) is close to 1, generate image similar to real images Discriminator Network, maximize the objective such that D(x) is close to 0 and D(G(z)) is close to 0. $\min_G \max_D V(D,G) = \underbrace{\boxed{\mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})]}}_{\text{Discrminator output for real data x}} + \underbrace{\boxed{\mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1-D(G(\boldsymbol{z})))]}}_{\text{Generator output for estimated data G(z)}}$
Training	Train both networks in sequence first encoder and then decoder in order to minimize the loss function to obtain an image same as that of	Train both Generator and Discriminator in parallel such that discriminator is trained in ascending stochastic gradient and generator in descending stochastic

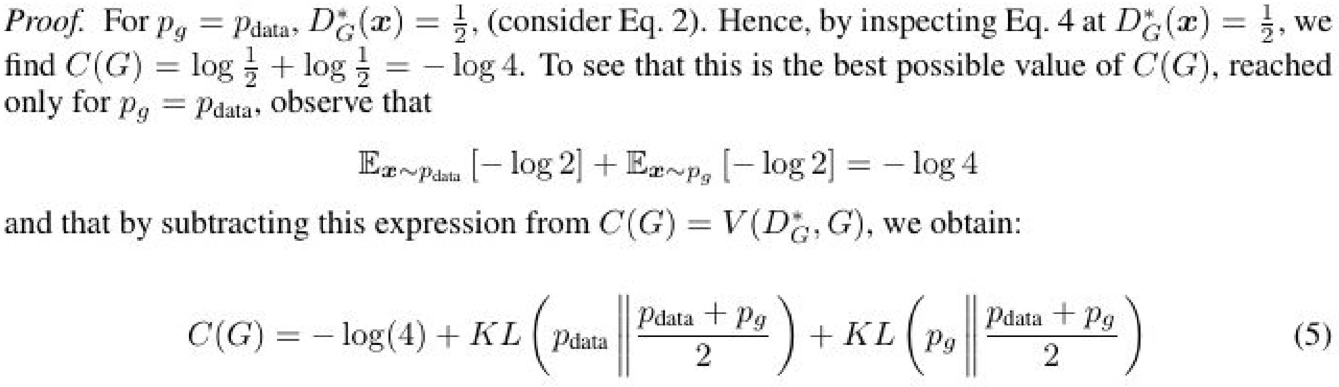
	the input image.	gradient.

3. **How is the distribution learned by the generator compared to the real data distribution when the discriminator cannot tell the difference between these two distributions? (15% of CW2)**

The convergence is achieved when the discriminator is unable to differentiate between real distribution and sample distribution by Generator. That is when D(x) = D(G(z)) = ½. Thus, when generator is $D^{\cdot}_{G}$ = ½ , p(g) = p(data) convergence is obtained. The distribution p(x) is learned in the following way.



Theorem 1. *The global minimum of the virtual training criterion $C(G)$ is achieved if and only if $p_g = p_{data}$. At that point, $C(G)$ achieves the value $-\log 4$.*



Proof. For $p_g = p_{\text{data}}$, $D^*_G(\boldsymbol{x}) = \frac{1}{2}$, (consider Eq. 2). Hence, by inspecting Eq. 4 at $D^*_G(\boldsymbol{x}) = \frac{1}{2}$, we find $C(G) = \log \frac{1}{2} + \log \frac{1}{2} = -\log 4$. To see that this is the best possible value of $C(G)$, reached only for $p_g = p_{\text{data}}$, observe that

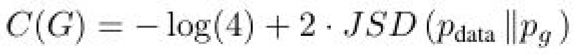
$$\mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}}\left[-\log 2\right] + \mathbb{E}_{\boldsymbol{x}\sim p_g}\left[-\log 2\right] = -\log 4$$

and that by subtracting this expression from $C(G) = V(D^*_G, G)$, we obtain:

$$C(G) = -\log(4) + KL\left(p_{\text{data}} \middle\| \frac{p_{\text{data}} + p_g}{2}\right) + KL\left(p_g \middle\| \frac{p_{\text{data}} + p_g}{2}\right) \tag{5}$$

where KL is Kullback-leibler divergence.

The Jensen–Shannon divergence between the model’s distribution and the data generating process:



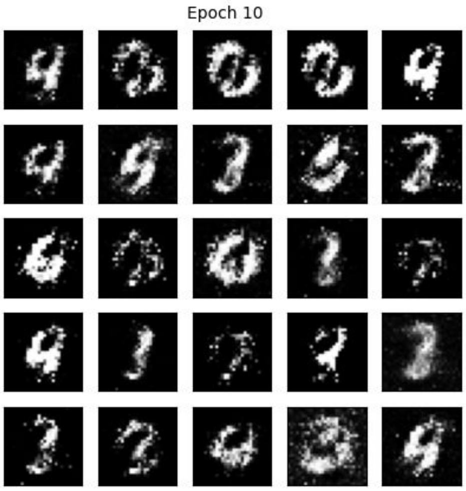
$$C(G) = -\log(4) + 2 \cdot JSD\left(p_{\text{data}} \| p_g\right)$$

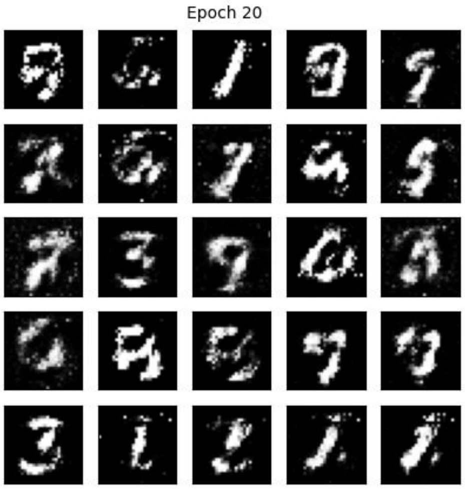
Since, the Jensen–Shannon divergence between two distributions is always non-negative and zero only when they are equal, we have shown that C^* = – log(4) is the global minimum of C(G) and that the only solution is p(g) = p(data) , i.e., the generative model perfectly replicating the data generating process.

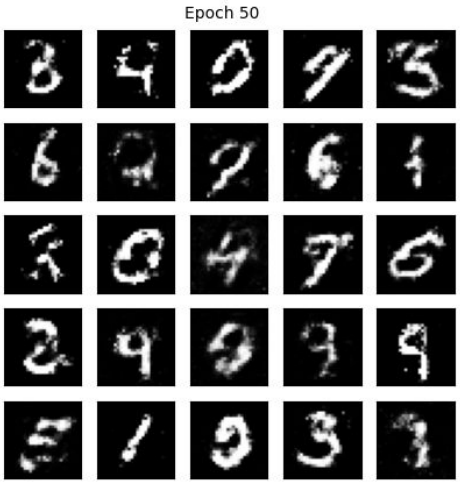
4. **Show the generated images at 10 epochs, 20 epochs,50 epochs,100 epochs by using the**

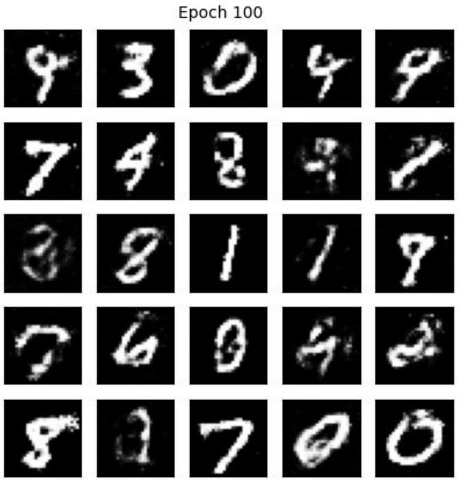
architecture required in Guidance. (15% of CW2)

With dropout set to 0.3

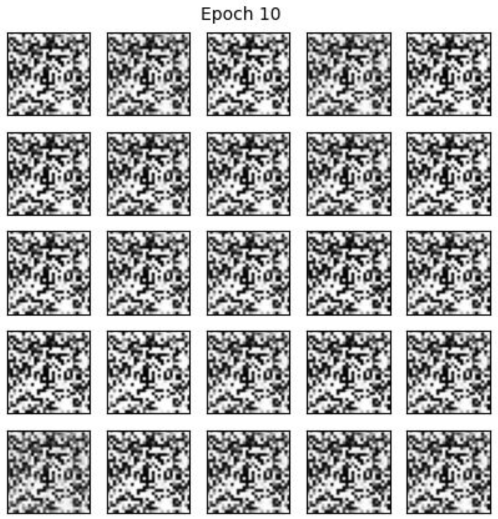


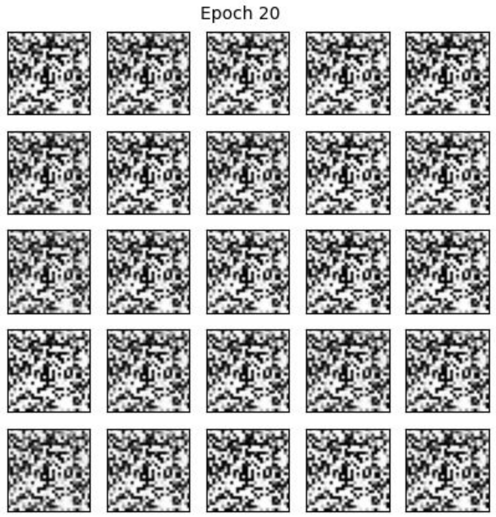


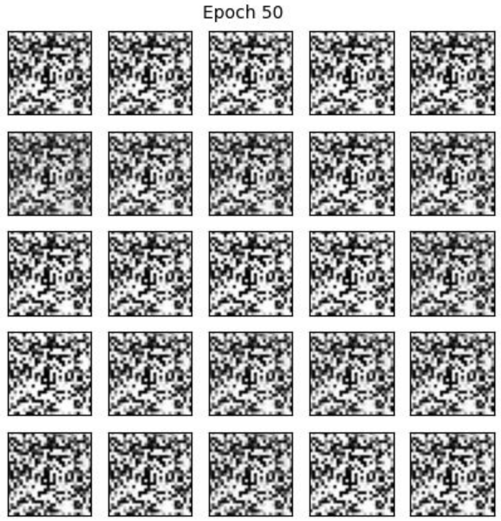


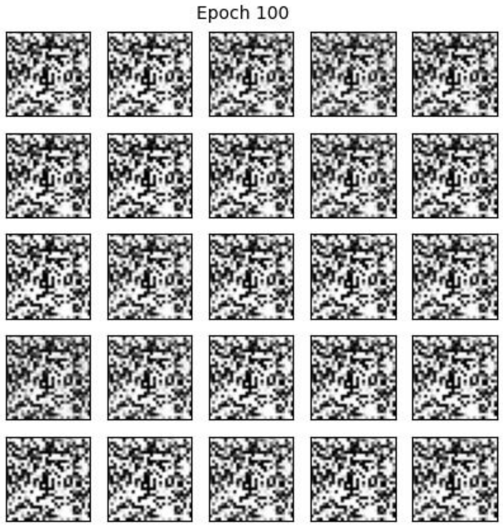


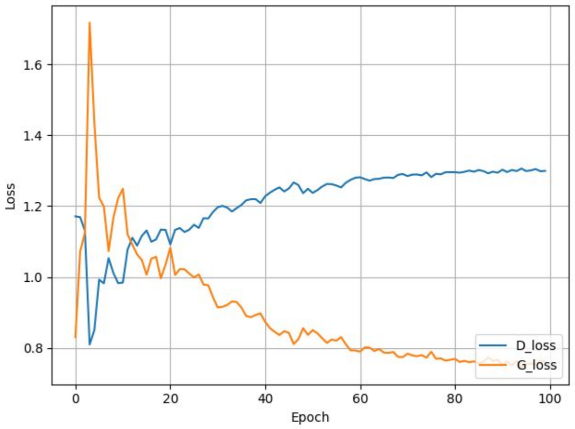
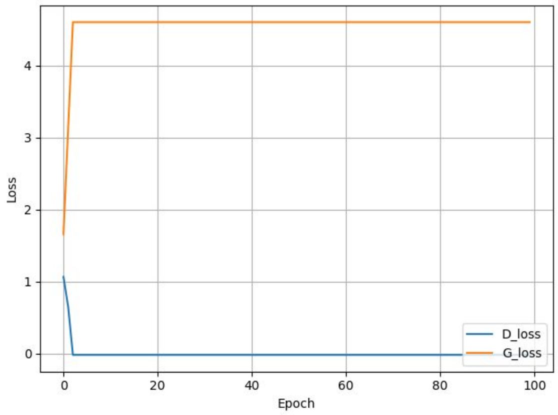
Without dropout









With dropout set to 0.3	Without dropout
	

When we set dropout to 0.3 in the discriminator network, the generator network learns the target images and it becomes difficult for the discriminator network to identify between real and generated distributions.

From above we also see that generator and discriminator networks fail to converge when there is no dropout and we get distorted images even at epoch 100