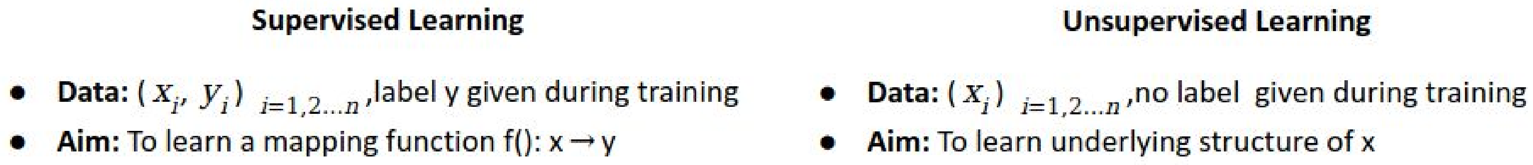
ECS795P Deep Learning and Computer Vision, 2020

Coursework 2: Unsupervised Learning by Generative Adversarial Network

1. **What is the difference between supervised learning & unsupervised learning in image classification tasks? (10% of CW2)**

   Supervised learning requires a lot of labelled/annotated training images. The system needs to be provided with lots of images that have to be labelled manually. Thus, the dataset consists of images with their annotations. This approach is used to generate a trained classifier model that can predict the features of unlabelled data/images. Traditional supervised learning process consumes a lot of time and resources and it is a big challenge to get the data. Whereas, unsupervised learning does not require such labelling and annotations. The model itself tries to find an underlying structure of the data to describe the data and build a probabilistic model from all the unknown images by extracting features and patterns on its own. This generates clusters for each particular type of data. By interpreting the clusters, the developer can understand the situation and also find ways to improve the accuracy of the model, one of which is to use the output of the supervised algorithm as a feedback. GANs are unsupervised learning algorithms that make use of the supervised loss in order to improve the accuracy of the unsupervised model.



Supervised Learning

- **Data:** (x_i, y_i) $_{i=1,2\ldots n}$,label y given during training
- **Aim:** To learn a mapping function f(): x → y

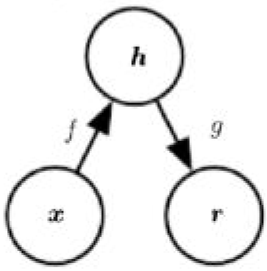
Unsupervised Learning

- **Data:** (x_i) $_{i=1,2\ldots n}$,no label given during training
- **Aim:** To learn underlying structure of x

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)**

The difference between auto-encoder and a generative adversarial network considering various aspects is as follows:

1. **Model structure:**
    - Auto-Encoder consists of two networks, called Encoder and Decoder that learn to generate output data as close to input data as possible. The encoder network maps input to the latent code h and the decoder network maps h to the reconstructed input.
    - Generative adversarial network also has two networks Generator and Discriminator, that have to generate samples belonging to a sample distribution. Generator tries to fool the discriminator by generating fake images as close to the real image(ground truth) as possible. Whereas the discriminator learns to distinguish between the generated images(fake) and the ground truth image(real). Both models aim to improve their performance.
2. **Optimized objective function:**
    - Auto-Encoder:
        - ○ Learn encoder f and decoder g as



$$\textbf{(f, g): x} \rightarrow \textbf{h} \rightarrow \textbf{r}$$

- ○ Objective function:

$$L(x, y; \theta) = -\frac{1}{M}\sum_{i=1}^{M} ||x_i - r_i||^2$$

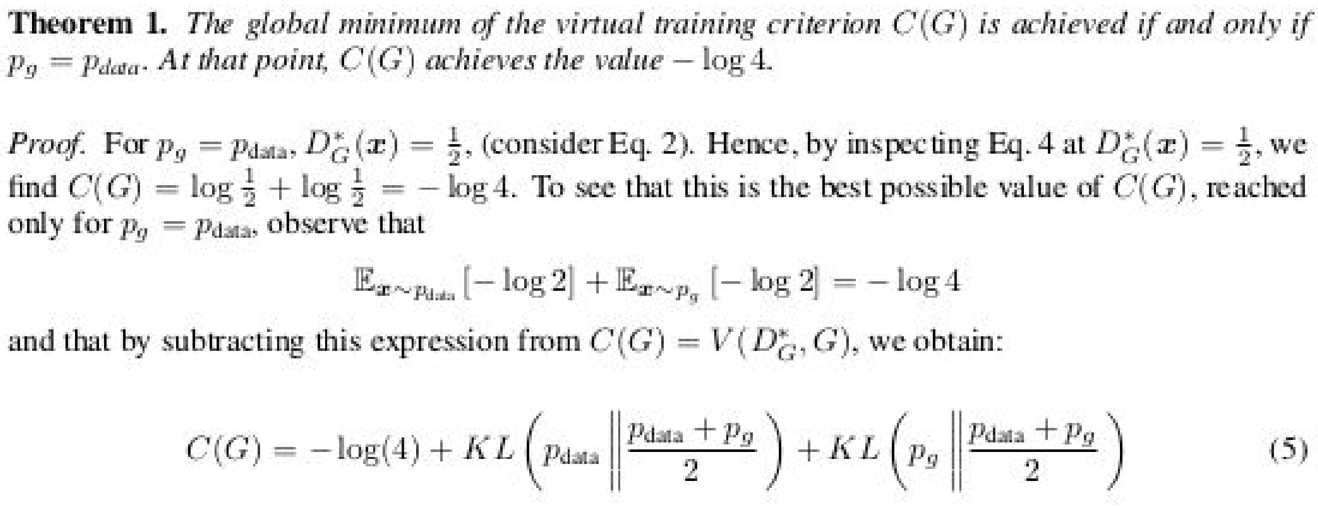
- GANs follow a two-player minimax game.
    - ○ Generator minimizes the objective D(G(z)) is close to 1, i.e. generate images G(z) close to the real image X.
    - ○ Discriminator maximizes the objective such that D(x) is close to 0 and D(G(z)) is close to 0 for fake images G(z).
    - ○ Objective Function:

$$\min_{\theta_g}\max_{\theta_d}\left[\mathbb{E}_{x\sim p_{data}}\log \underbrace{D_{\theta_d}(x)}_{\text{Discriminator output for real data x}} + \mathbb{E}_{z\sim p(z)}\log(1-\underbrace{D_{\theta_d}(G_{\theta_g}(z)))}_{\text{Discriminator output for generated fake data G(z)}}\right]$$

3. **Training procedure on different components:**
    - Auto-Encoder trains both the networks chronologically, the encoder first and then the decoder to make the output image same as the input by minimising the loss function.
    - GAN trains both the networks parallely. Generator is trained in the descending stochastic gradient while the Discriminator is trained with an ascending stochastic gradient.

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? (15% of CW2)

GAN reaches convergence when Discriminator cannot distinguish between the real ground truth images and the images generated by the generator. This occurs when D(x) = D(G(z)) = $\frac{1}{2}$. Thus, for the given generator the optimal discriminator is $D *_G = 1/2$. The convergence can be reached only when p(g) = p(data). 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 $-\log 4$.*

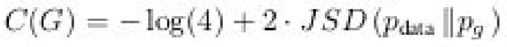
Proof. For $p_g = p_{\text{data}}$, $D^*_G(\boldsymbol{x}) = \frac{1}{2}$, (consider Eq. 2). Hence, by inspecting Eq. 4 at $D^*_G(\boldsymbol{x}) = \frac{1}{2}$, we find $C(G) = \log\frac{1}{2} + \log\frac{1}{2} = -\log 4$. To see that this is the best possible value of $C(G)$, reached only for $p_g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

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

where KL is the Kullback–Leibler divergence. We recognize in the previous expression the Jensen–Shannon divergence between the model's distribution and the data generating process:

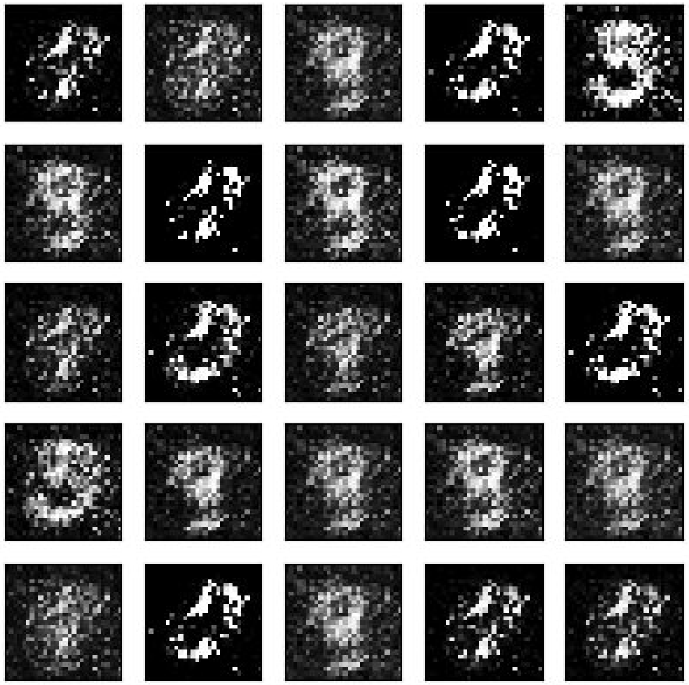


$$C(G) = -\log(4) + 2 \cdot JSD\left(p_{\text{data}} \| p_g\right)$$

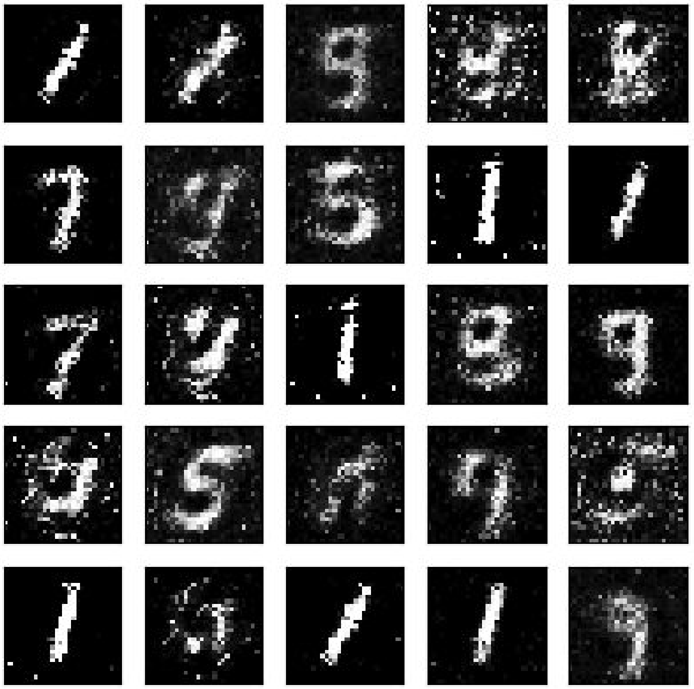
Since the Jensen–Shannon divergence between two distributions is always non-negative and zero only when they are equal, we have shown that C∗ = − log(4) is the global minimum of C(G) and that the only solution is p_g = p_{data} , i.e., the generative model perfectly replicating the data generating process.

4. **Show the generated images at 10 epochs, 20 epochs,50 epochs,100 epochs by using the architecture required in Guidance. (15% of CW2)**

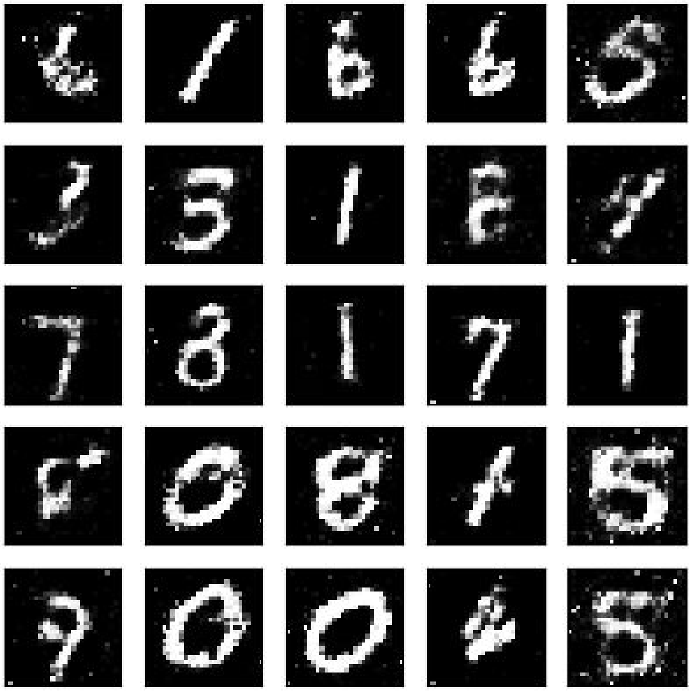
- **With dropout 0.3:**



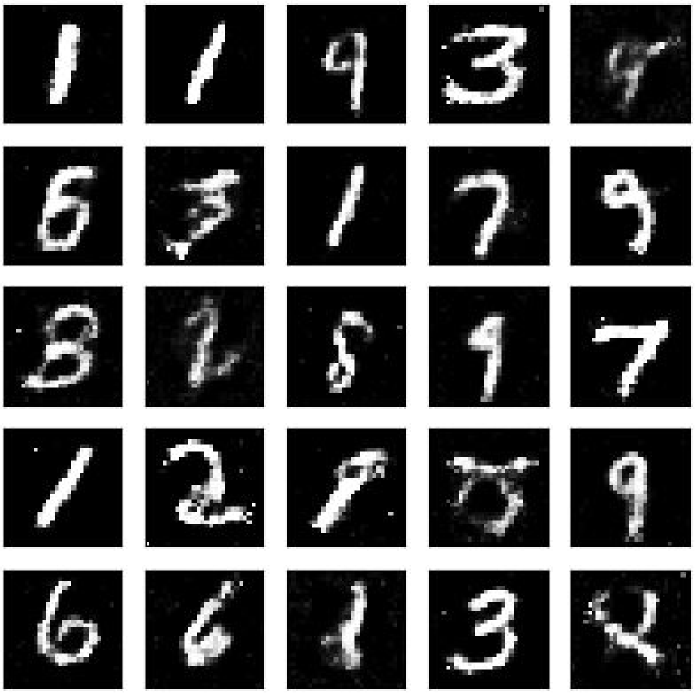
Epoch 10



Epoch 20

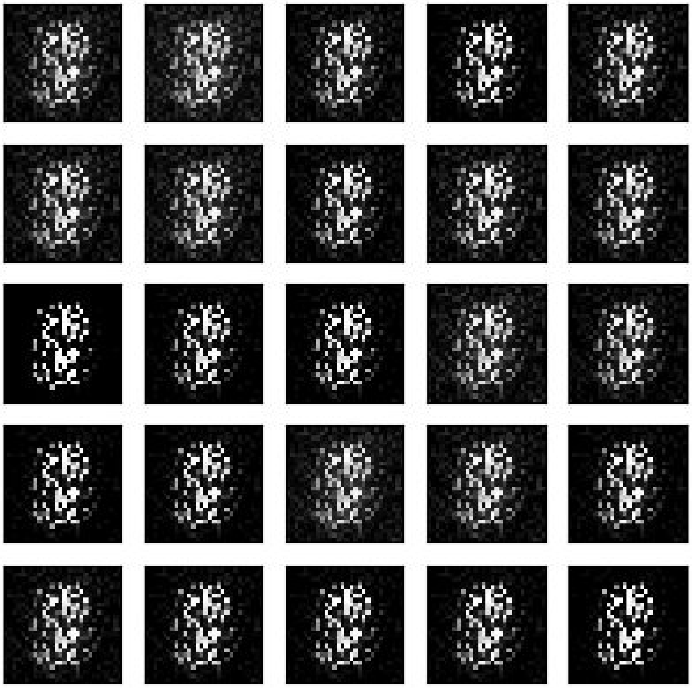


Epoch 50

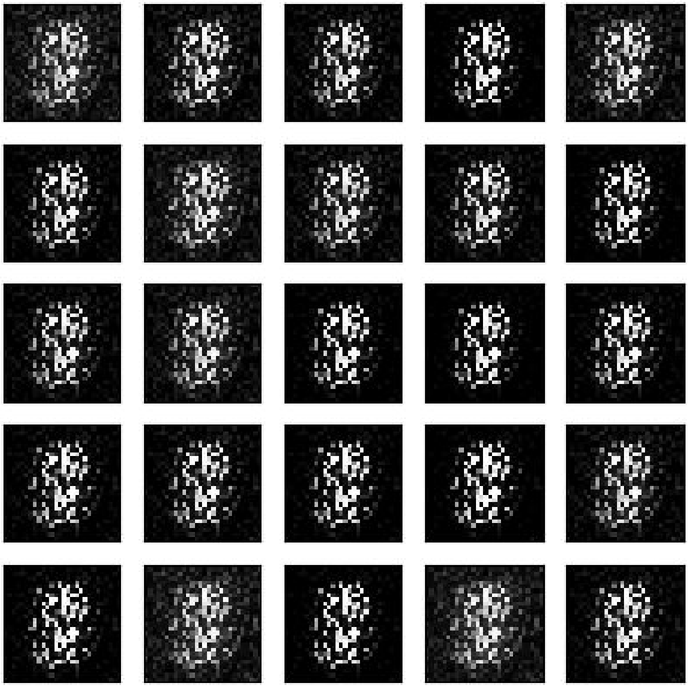


Epoch 100

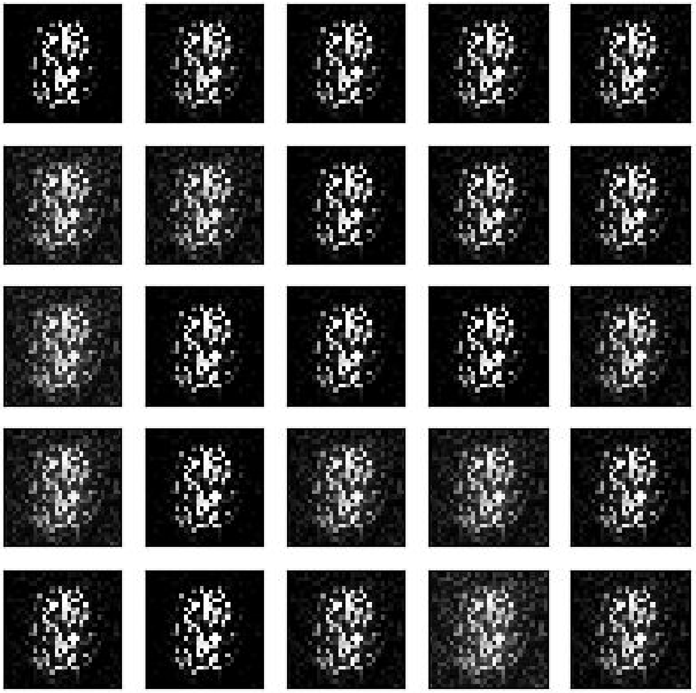
- **Without dropout**



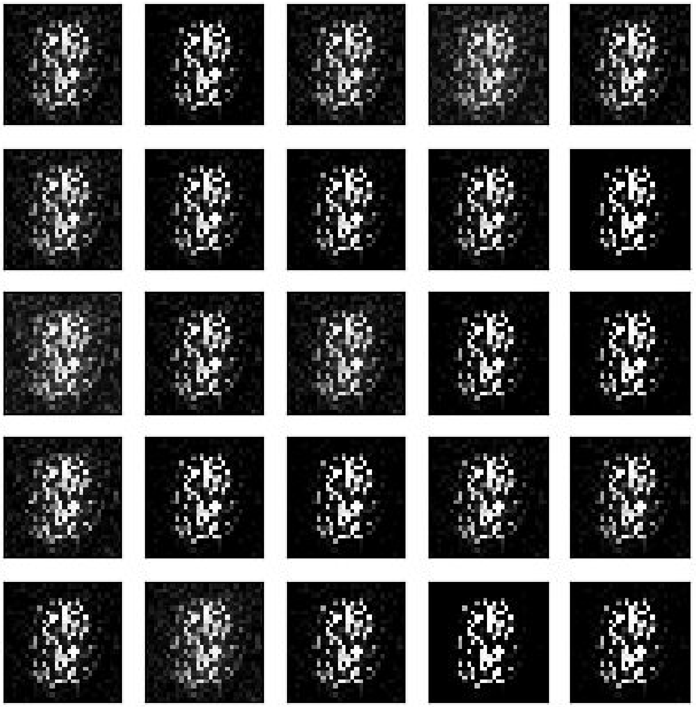
Epoch 10



Epoch 20



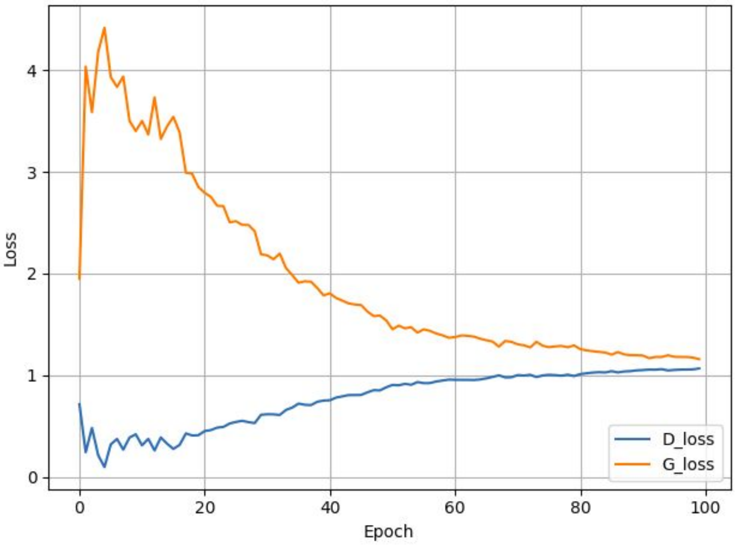
Epoch 50



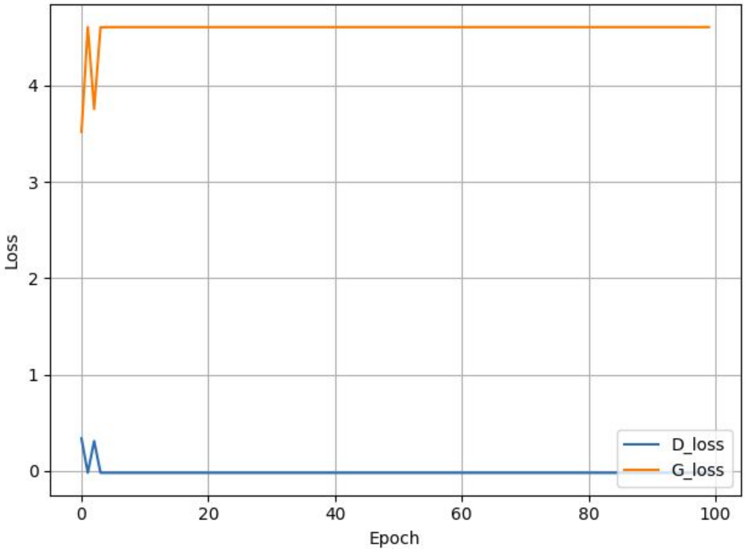
Epoch 100

Training Histograms:

With dropout 0.3



Without dropout



The generator is able to learn the distribution and fool the discriminator while it becomes difficult for discriminator to distinguish between real and fake images when providing a dropout of 0.3. As for the model without dropout, we cannot get satisfactory images even after 100 epochs. It is evident from the histogram that the model is not at all accurate without the dropout, and it fails to converge.