**Unsupervised Learning by Generative Adversarial Network**

1. **Difference between supervised learning & unsupervised learning in image classification task?**

Within the supervised learning systems, the created architecture (algorithm) is feeded with the dataset having pre-imposed informative labels, meaning that the raw data here (images) have been tagged in advance with the labels. Labels (data labels) can be identified as the textual elements which delineates the individual data point. These labels/information acts as the features to the dataset, giving foundation and context to the machine learning model and ‘teach/supervise’ the algorithm therefore allowing it to cross-validate the efficiency between the input and output.

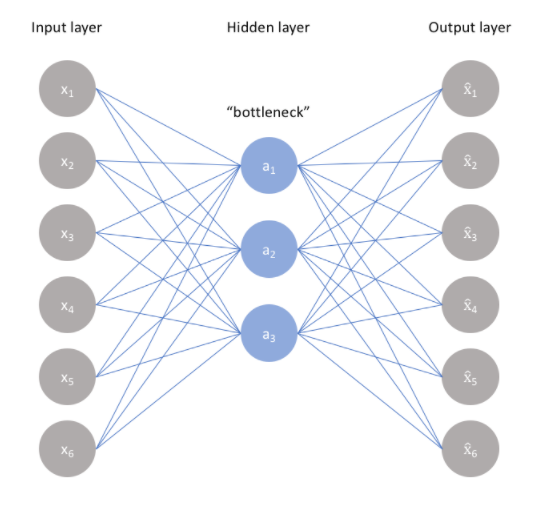
On the other hand, within unsupervised learning systems the build architecture is feeded with the unlabeled image dataset and the algorithm pipeline extracts the latent informative patterns from the dataset without any external supervision, hence unsupervised in nature.

1. **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**”

1. **Auto-Encoders.**

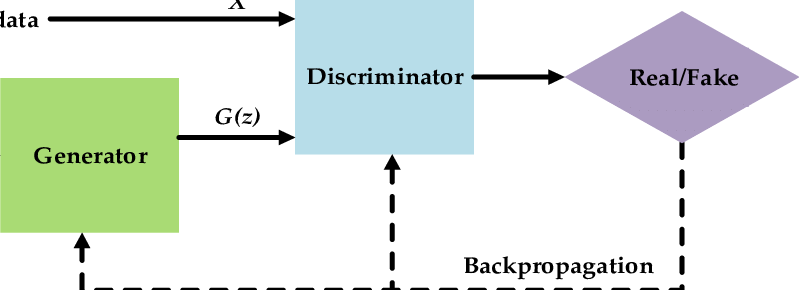
It is classified as an unsupervised learning technique where we exploit the existing neural networks to function towards the feature learning (aka representation learning). Here we constitute a neural network architecture in a way that it creates a bottleneck within the network therefore imposing a compressed and encoded realization of the data representation of the input. Therefore, enabling to correlate features and reconstruct the information/data as near as possible to the inputs.



1. **Generative Adversarial Network.**

GAN model is an unsupervised learning task in machine learning which involves exploring and learning the patterns and dependencies within the dataset so that the build pipeline generates new examples those which are plausibly could have been drawn from the original dataset.

GAN comprises of two subordinate neural pipelines known as the generator pipeline which gets trained to produce new examples, and the discriminator pipeline that classifies examples real (from the domain) or fake (generated). These models get trained together following a zero-sum game policy, until the discriminator model is fooled, meaning the generator model now is generating plausible examples.



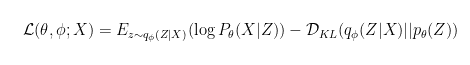
1. **Difference.**
2. Autoencoder basically reduces its input to a vector having much fewer dimensions when compared to the input data, later this model transforms it back into a tensor (higher dimension) with the same shape as it had earlier. On the other hand, GAN appears different structurally and instead of compressing the data it originally passes a low dimensional vector (as the inputs) and high dimensional data in the middle.
3. In strict technical terms they both falls under the category of unsupervised learning, but they both employs significant different approaches over the research question.
4. GAN comes under generative model category meaning it generates realistic (new) samples of a dataset. And, auto-encoders are variational in nature, and they just reconstruct their inputs and can’t generate realistic new samples.

“**OPTIMIZED OBJECTIVE FUNCTION**”

1. **Auto- Encoder.**

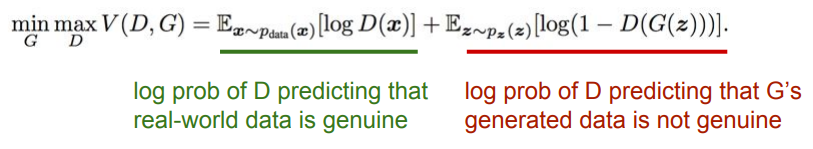
The basic idea behind an autoencoder’s representation learning procedure (encoding) for a set of data is dimensionality reduction, here model is trained to ignore the noise/nonsignificant data. Hence, auto-encoder in principle is associated to the Principal Component analysis technique.

Also, Variational autoencoder (VAE) uses probabilistic approximate distribution. Where we sample the data from posterior, and valuate the obtained expectations. Within VAE, our research problem is defined in terms of the lower bound which is expected to be maximized.



1. **Generative Adversarial Network.**

Within GAN’s architecture generator and discriminator both sports through each other as a minimax game, where the generator minimizes a two-sample test objective and the discriminator maximizes the objective.



“**TRAINING PROCEDURE ON DIFFERENT COMPONENTS**”

1. **Auto- Encoder.**

Autoencoders gets strained using supervised learning methods, and are typically are a part of a broader model that attempts to recreate the input.

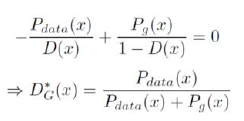
By design, autoencoders makes learning challenging by focusing more on bottleneck, from which the reconstruction of the input data is performed.

1. **Generative Adversarial Network.**

GAN works alternatingly switches between generator and discriminator, where discriminator trains for one or more epochs so as the generator while keeping the generator constant through the entire training phase. Main problem in training generator that it is for an untrained generator that produces random output.

1. **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?**

According to the Goodfelllow et al, while having a fixed generator G, the optimal discriminator D can be defined using the below shown relation

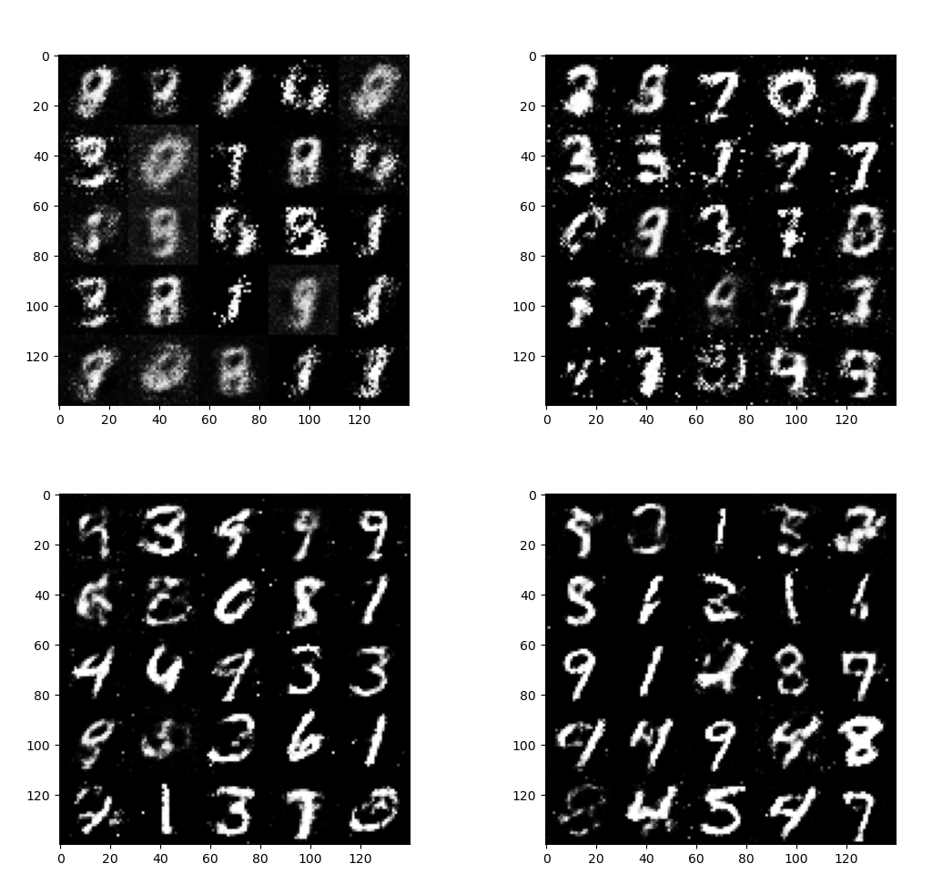


And after several iteration/epoch of calculations both discriminator and generator have enough capacity, they will reach a point at which both cannot improve because Pg will become equal to Pdata resulting in forcing Discriminator to unable to speculate between the distributions. Contrary to above case if both discriminator and generator have enough capacity, and at each Algorithm step or iteration, discriminator gets to reach its optimum given generated values, and Pg is updated so as to improve the criterion of expectation then at later stage Pg converges to Pdata giving rise to the optimum system having an efficient output.



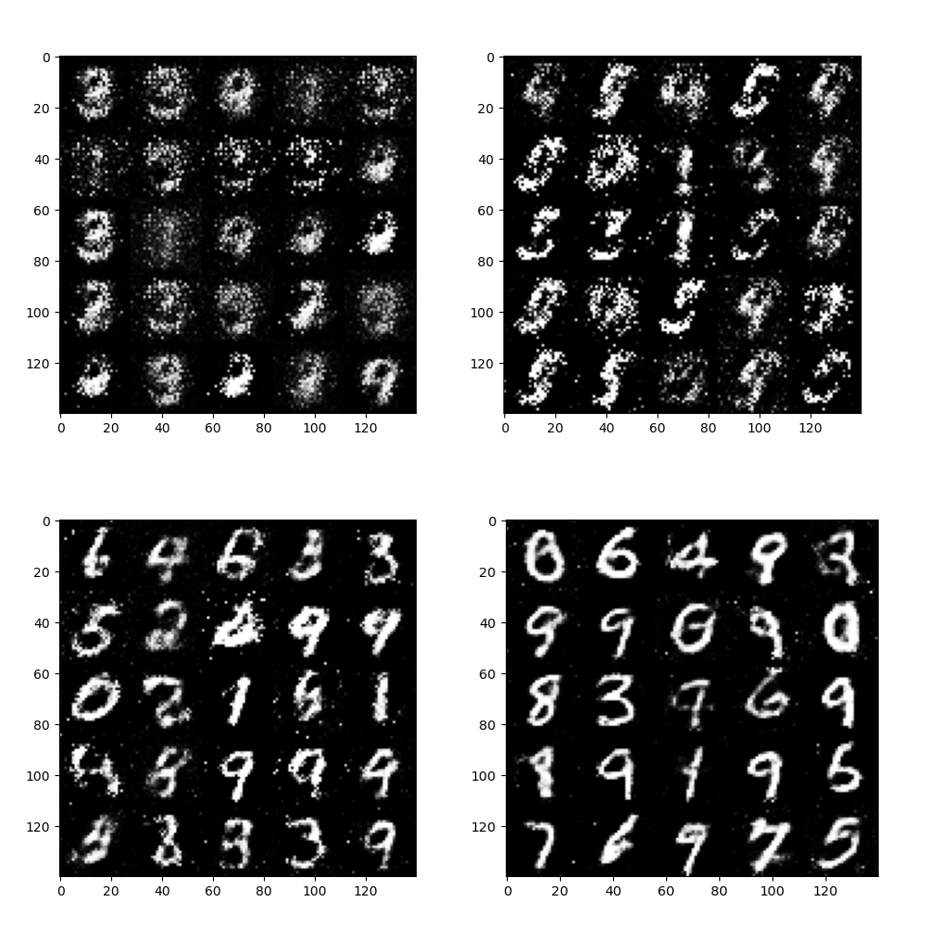
1. **Show the generated images at the 10th epoch, the 20th epoch, the 50th epoch, the 100th epoch by using the architecture required in Guideline. (10% of CW2)**

* **WITH DROPOUT**



GAN Architecture Output (Epoch Clockwise: 10, 20, 50, 100) -With Dropout

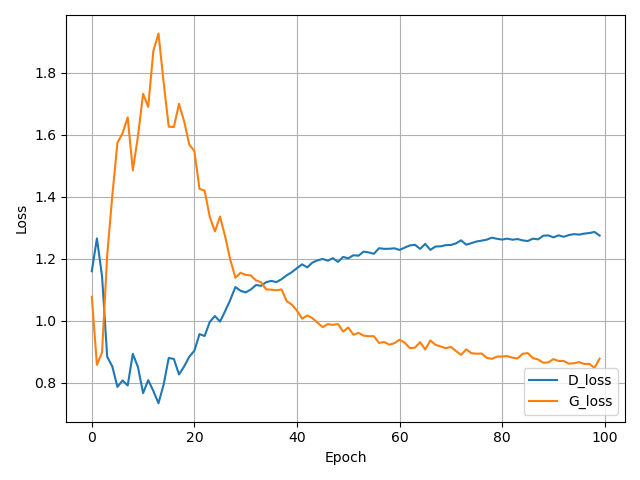
* **WITHOUT DROPOUT**



GAN Architecture Output (Epoch Clockwise: 10, 20, 50, 100) -Without Dropout

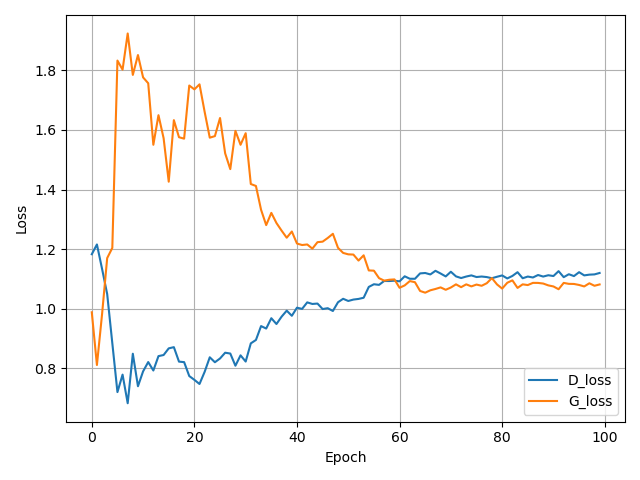
1. **Plot the loss curve during training.** (**10% of CW2**)

* **LOSS CURVE WITH DROPOUT**



GAN Loss Curve (With Dropout)

* **LOSS CURVE WITH DROPOUT**



GAN Loss Curve (With Dropout)

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. *Advances in neural information processing systems*, *27*.