Architecture

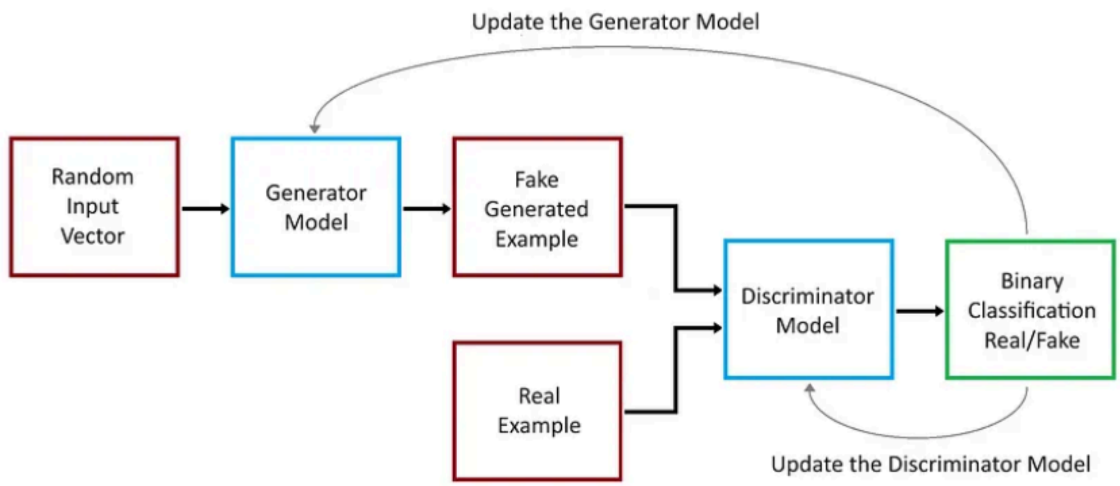

Image by the author

A Generative Adversarial Network (GAN) consists of two neural networks, namely the Generator and the Discriminator, which are trained simultaneously through adversarial training.

Generator: This network **takes random noise as input and produces data (like images)**. Its goal is **to generate data that's as close as possible to real data**.
Discriminator: This network t**akes real data and the data generated by the Generator as input and attempts to distinguish between the two**. It outputs the probability that the given data is real.

Generator activation functions : LeakyReLU or ReLU for all layers, except last layer, we use tanh activation

Activation Functions: ReLU (Rectified Linear Unit) or Leaky ReLU are common choices for activation functions in the generator. These functions introduce non-linearity to the model, enabling it to generate complex data

Full activation = https://arxiv.org/pdf/1603.05027
GANs with ResNet : https://arxiv.org/pdf/1707.04881

Discriminator Loss

The discriminator's goal is to correctly classify real samples as real and fake samples (produced by the generator) as fake. Its loss is typically represented as:

$$L_D = -\frac{1}{2}\mathbb{E}_{x\sim p_{\text{data}}(x)}[\log(D(x))] - \frac{1}{2}\mathbb{E}_{z\sim p_z(z)}[\log(1 - D(G(z)))]$$

where

$$\mathbb{E}_{x\sim p_{\text{data}}(x)}[f(x)] \approx \frac{1}{N}\sum_{i=1}^{N} f(x_i)$$

$$\mathbb{E}_{z\sim p_z(z)}[f(z)] \approx \frac{1}{M}\sum_{i=1}^{M} f(z_i)$$

Generator Loss

The generator's goal is to produce samples that the discriminator incorrectly classifies as real. Its loss is typically represented as:

$$L_G = -\frac{1}{2}\mathbb{E}_{z\sim p_z(z)}[\log(D(G(z)))]$$

This term penalizes the generator when the discriminator correctly identifies its outputs as fake.

Combined Loss

The combined GAN Loss, often referred to as the minimax loss, is a combination of the discriminator and generator losses. It can be expressed as:

$$L_{\text{GAN}} = \min_G \max_D L_D + L_G$$

This represents the adversarial nature of GAN training, where the generator and the discriminator are in a two-player minimax game. The discriminator tries to maximize its ability to classify real and fake data correctly, while the generator tries to minimize the discriminator's ability by generating realistic data.

DCGANs datasets : https://www.tensorflow.org/datasets/catalog/lsun
We trained DCGANs on three datasets, Large-scale Scene Understanding (LSUN) (Yu et al., 2015),Imagenet-1k and a newly assembled Faces dataset. Details on the usage of each of these datasets are given below.

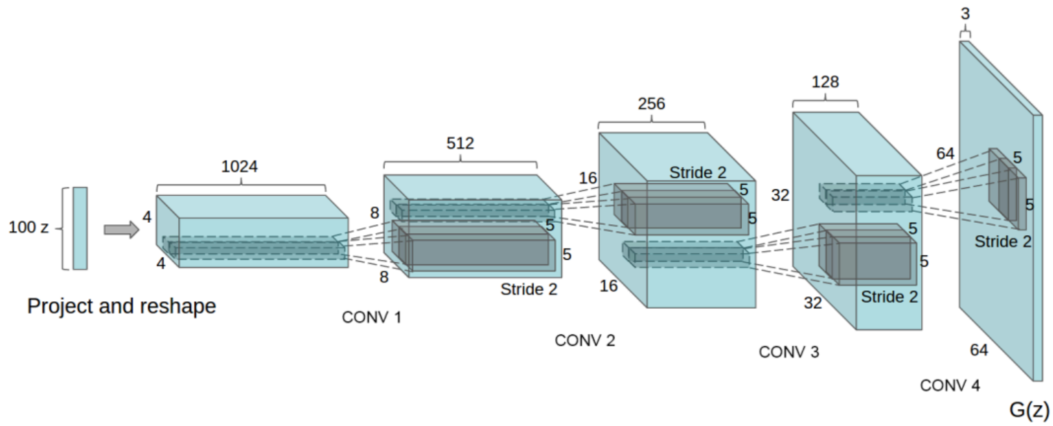

Figure 1: DCGAN generator used for LSUN scene modeling. A 100 dimensional uniform distribution Z is projected to a small spatial extent convolutional representation with many feature maps. A series of four fractionally-strided convolutions (in some recent papers, these are wrongly called deconvolutions) then convert this high level representation into a 64×64 pixel image. Notably, no fully connected or pooling layers are used.

When the relationship between the input and the output is uncertain, the neural networks are used. They are categorized as supervised, unsupervised and reinforcement learning. **The Generative Adversarial Networks (GANs) are a class of unsupervised learning used for generative modelling**. Through iterative process, t**he generative model generates new data and are derived from a realistic data and the randomly generated noise which are similar to the existing data**. **In each iteration, the loss is calculated and back propagated to obtain an optimized model**. Figure 1 illustrates how a CycleGAN translated the brush art into a photograph.