What is a Generative Adversarial Neural Network?

This was a type of network model proposed by Ian Goodfellow back in 2014.

This model has two distinct networks, tagged the *Generator* and the *Discriminator.* These networks try to outsmart each other continuously until one of them (usually the Generator) produces an output matching the desired level of accuracy.

How this works is that the Generator uses an arbitrary input to generate an output whose dimensions match those of a given set of samples, and the Discriminator has to determine if a given input to it is generated by the Generator or is from the actual sample set.

Ideally after sufficient training, the Generator should be able to produce outputs from input noise which resembles closely to that of the given sample set.

How is it trained?

As mentioned earlier, the Generator model is made to generate some kind of sample whose dimensions match that of a given sample set. This set of generated samples is mixed with the actual samples and fed to the discriminator. Now, the discriminator (through supervised learning) is trained (Generator weights kept fixed) such that it can accurately predict if a given imput is of the generated data set or the given sample set.

Once the Discriminator is trained enough, it is frozen, and the generator is trained to try and decieve the discriminator model into believing that it's generated sample is a true one.

This alternating process is usually done until the Generator can produce outputs that share a required amount of similarity to the real sample set.

Why is this a such an interesting model?

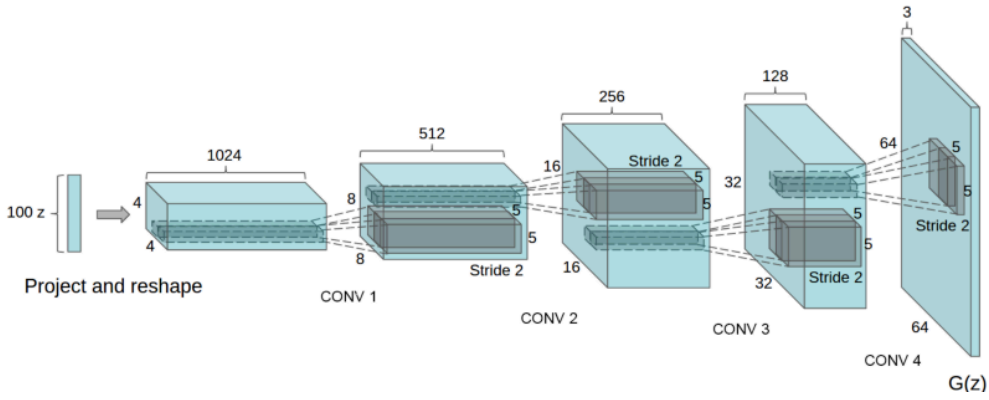
This is very interesting as the very nature of a GAN prevents any sort of overfitting from taking place, as if there is an overfit among the sample set, the discriminator will exploit that to differentiate and thus prevent the generator from remaining that way.

This model can be used to synthesise datasets and affect the field of accurate content generation heavily. If done well, we can make the GAN mimic new samples of the original dataset, thus helping us train a better model for that dataset.

Variants of this model have been modelled where they can be used for image enhancement, de-noising, and even generation. This may drive a lot of classical Image processing domains obsolete and usher a new era of image processing.

Our Model

We have used the model architecture borrowed from the very successful 'DCGAN' structure where many convolutional layers are stacked one after the other without pooling and aggregation.



We used convolution layers to upscale the 1D noise input to a 3channel 64x64 image which matches the size of the sample set which we intend to imitate.

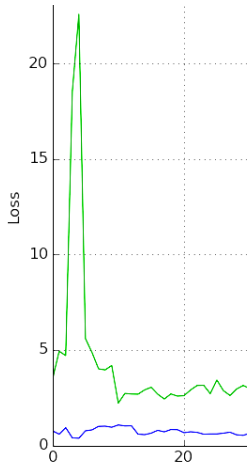
By converting the inputs of both the generator and the discreminator to matrix form, it increases scalability allowing us to train the entire model on a bigger dataset without bogging down the process units

Results:

Input dataset examples :

Output obtained after 25 epochs :

As can be seen, even after a little training, we are generating completely new face like images from nothing but an input noise. On further training we can get better and better results.



—> Generator Loss
—> Discriminator Loss