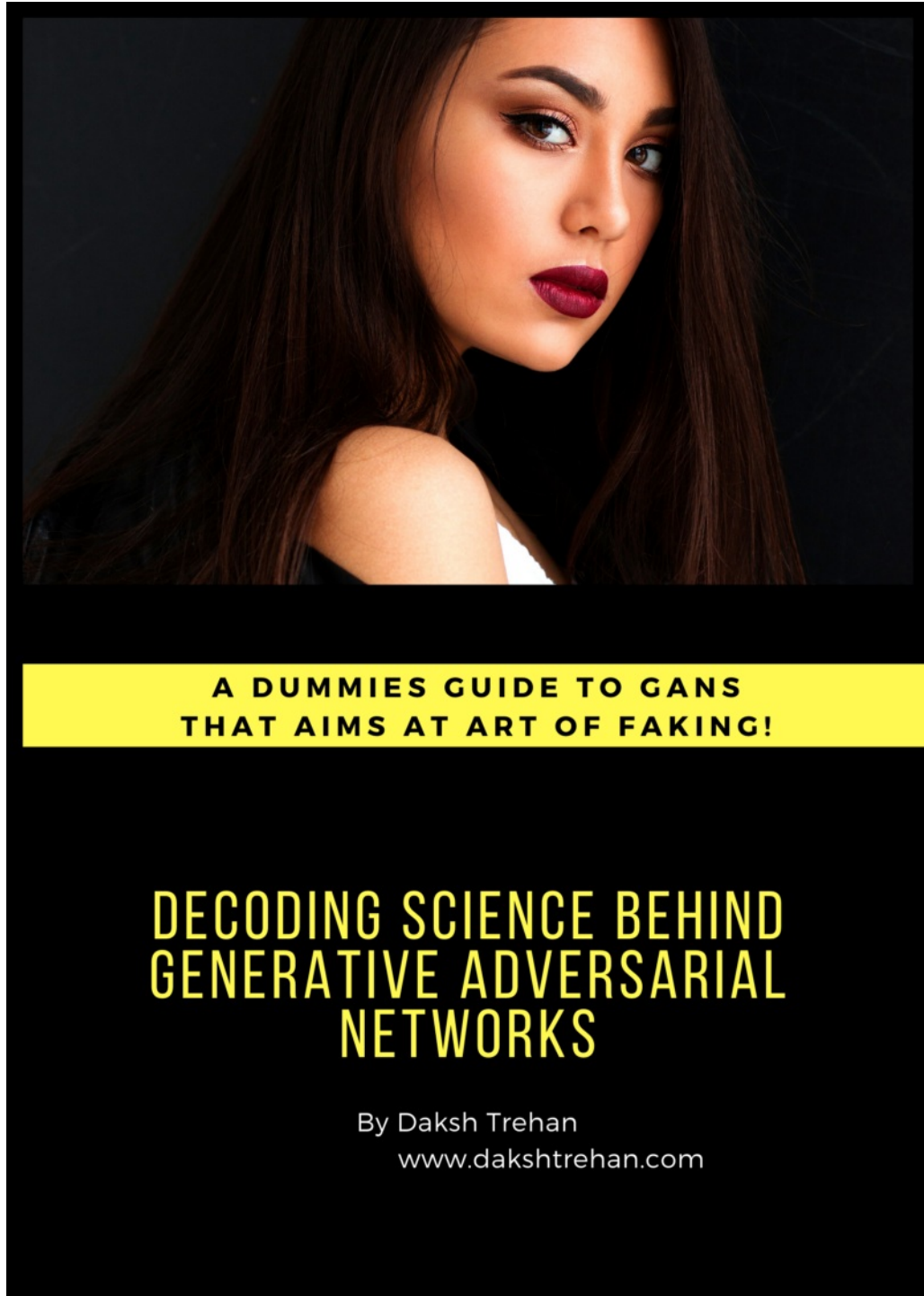


Decoding science behind Generative Adversarial Networks

A dummies guide to GANs that aims at art of faking!



Generative adversarial networks(GANs) took the Machine Learning field by storm last year with those impressive fake human-like faces.

Bonus Point* They are basically generated from nothing.

Irrefutably, GANs implements implicit learning methods where the

model learns without the data directly passing through the network, unlike those explicit techniques where weights are learned directly from the data.

Intuition



Photo by [Robert Nyman](#) on [Unsplash](#)

Okay, suppose in the city of Rio de Janeiro, money forging felonies are increasing so a department is appointed to check in these cases.

Detectives are expected to classify the legit ones and fake ones. When the officers correctly classify fake currency they are appreciated and when they make mistakes, feedback is provided by Central Bank. Now, whenever fake currencies are busted, forgers aims at circulating better fake currencies in the market. This is the recursive process, when detectives identify fake currencies, the forger learns a better way to develop a more authentic currency, similarly with the admission of more fake currencies in the market the detectives develop an even better strategy to identify fabricated currencies. Both detectives and forgers will become better at what they do by learning from each other.

This is the basic intuition behind GANs.

In GANs, there are two networks Discriminators (the detectives) and the Generator (the forger). The generator aims at creating fake data and discriminator is expected to bust fake ones. The recursive process makes both generator and discriminator to compete to perform better. They're interdependent on each other as both of them play an essential role in each other's learning process.

Another point to ponder is that both networks are constantly learning. Let's say the discriminator wrongly classified fake data as a real one, that would lead to generator learning that the current developed data is a good one. But, this is technically wrong and will be corrected by discriminator when it gets the feedback from the government about the same.

How GANs learn?

Goal: To generate new data i.e. new images, texts, speech, etc.

Generative Modeling

This is the type of modeling that involves learning trends based on given data and try to generate samples that closely resembles the provided data. A very common illustration of Generative modeling can be regarded as, predicting the sequential data.

Generative models follow joint probability distribution.

Generative



The end goal is to get X from Y.

$P(X, Y)$ can be read as the probability distribution of variables X **and** Y.

Discriminative Modeling

This type of model generally includes those algorithms that aim at classifying the data from a given trend. Analyzing an image as a cat or a dog in it can be regarded as its best example.

While discriminative models are the ones that discriminate (classify) the data from a given distribution. Telling if a given picture has a cat or a dog in it, is the simplest example of discriminative modeling.

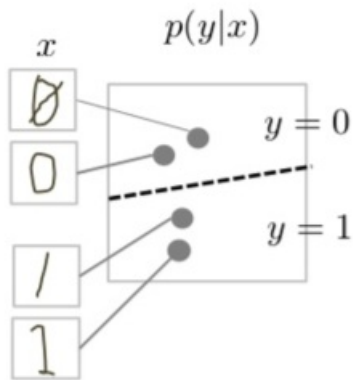
Discriminative



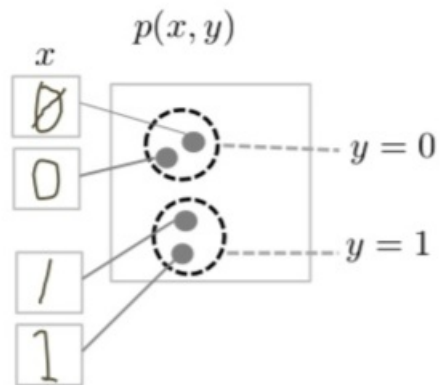
The end goal is to predict Y from X.

$P(Y|X)$ can be regarded as “Probability of event Y occurring given event X has occurred”.

• Discriminative Model



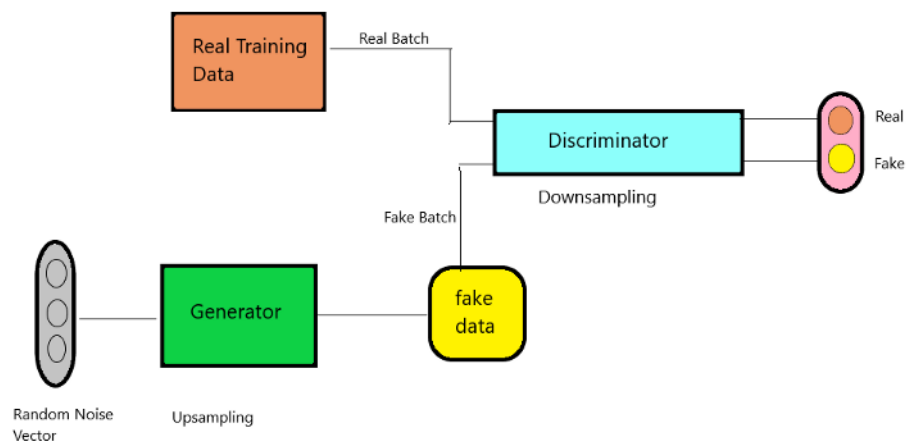
• Generative Model



Discriminative vs Generative, [Source](#)

GAN follows Generative modeling and hence it aims at generating $P(X,Y)$

Model Architecture



A typical GAN

Noise

This comes from the random distribution that can be transformed into the required data. By introducing noise, we are making our GAN robust and thus easing its process to “produce a wide variety of data, sampling from different places in the target distribution.”

Noise should be sampled from Gaussian Distribution.

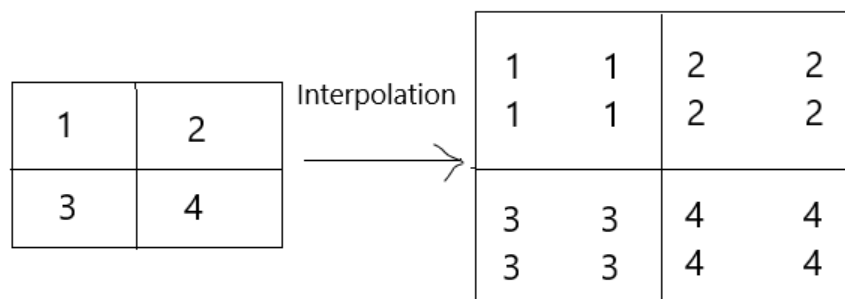
Data from the Real Distribution

This is the required distribution. We expect GAN to generate data inspired by this distribution with a little help of noise data.

Generator

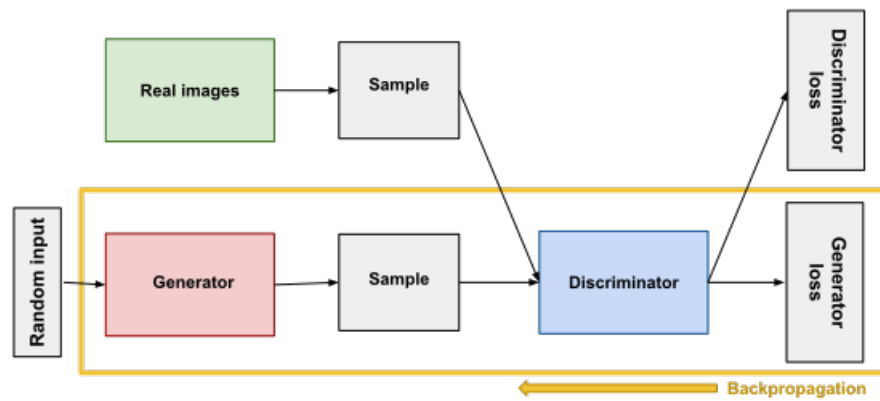
This is an essential component of GAN. It focuses on developing data following the trends of distribution.

Generators are required to upsample the input data to generate output data. The upsampling layer is a no weight layer that helps in increasing the spatial size of our input vector.



UPSAMPLING 2D

The generator creates data that would be classified by discriminator as real or fake. The generator will be fined with more loss if the data generated shares less resemblance with actual data and then using backpropagation it will try to learn from its mistakes to develop a better strategy to fool the discriminator.



Generator operation, [Source](#)

Generator training:

- Sample random noise.
- Produce output using random noise.
- Analyze loss based on the discriminator's classification.
- Backpropagate through both discriminator and generator to get gradients.
- Update weights using gradients.

Goal: To minimize the probability of detecting fake and real data.

Input accepted: Random noise

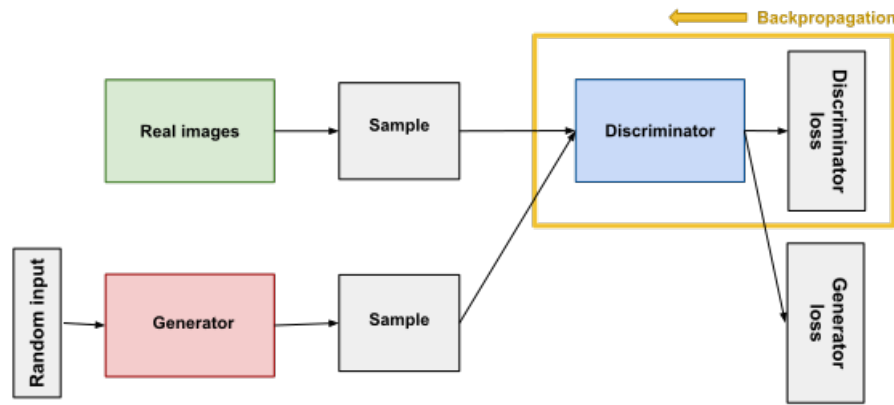
Output generated: Fake data

Discriminator

It is just a regular classifier that tries to distinguish real data from fake ones. Its network architecture depends upon the type of data it is implementing upon e.g. it might use RNN for text data, CNN for image data.

It tries to increase the number of channels thus easing the process to learn more features about data.

The discriminator tries to classify the data, it usually ignores the generator's loss and just implements a discriminator's loss. The discriminator will be fined with more loss if it misclassifies actual data with a fake one and then using backpropagation it will try to learn from its mistakes to develop a better strategy to fool generator.



Discriminator operation, [Source](#)

Discriminator training:

- Classify the input data as real or fake.
- Analyze the loss function for misclassifying.
- Update weights using backpropagation.

Goal: To maximize the probability of detection of real & fake data.

input: Data (Real+Fake)

output: Probability of data being Real or Fake.

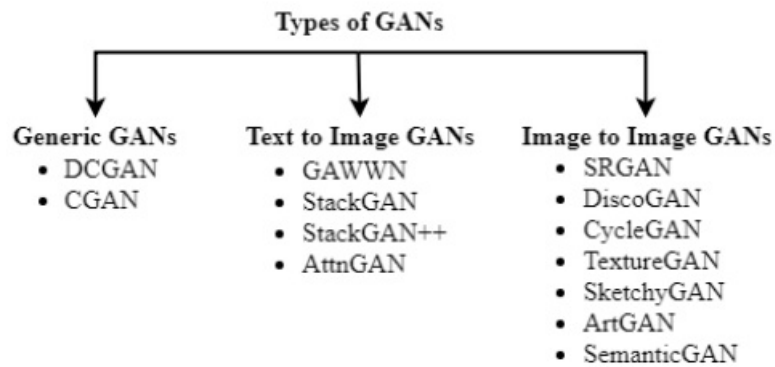
Training our model

While training our model, both networks must be trained separately. First, the discriminator is trained for several steps and then the training is shifted for generator.

If we try to train both networks simultaneously it would be like aiming for a flying bird and hence increasing our chances of failure.

As the generator gets better with training, the performance of discriminator gets struck i.e. it becomes difficult for it to classify real and fake data. If the generator succeeds perfectly to achieve its goal i.e. minimizing the probability of detection, the accuracy of discriminator swoops down to 0.5. The downgrade inaccuracy suggests that discriminator has shifted its strategy from statistical to random methods. This further leads to languish Generator's accuracy and worsening our model's performance.

Type of GAN models



GANs variations, [Source](#)

Conclusion

Hopefully, this article will help you to understand about Generative Adversarial Networks(GAN) in the best possible way and also assist you to its practical usage.

As always, thank you so much for reading, and please share this article if you found it useful!

Feel free to connect:

LinkedIn ~ <https://www.linkedin.com/in/dakshtrehan/>

Instagram ~ https://www.instagram.com/_daksh_trehan/

Github ~ <https://github.com/dakshtrehan>

Follow for further Machine Learning/ Deep Learning blogs.

Medium ~ <https://medium.com/@dakshtrehan>

Want to learn more?

[Detecting COVID-19 Using Deep Learning](#)

[The Inescapable AI Algorithm: TikTok](#)

[An insider's guide to Cartoonization using Machine Learning](#)

[Why are YOU responsible for George Floyd's Murder and Delhi Communal Riots?](#)

[Understanding LSTM's and GRU's](#)

[Recurrent Neural Network for Dummies](#)

[Convolution Neural Network for Dummies](#)

[Diving Deep into Deep Learning](#)

[Why Choose Random Forest and Not Decision Trees](#)

[Clustering: What it is? When to use it?](#)

[Start off your ML Journey with k-Nearest Neighbors](#)

[Naive Bayes Explained](#)

[Activation Functions Explained](#)

[Parameter Optimization Explained](#)

[Gradient Descent Explained](#)

[Logistic Regression Explained](#)

[Linear Regression Explained](#)

[Determining Perfect Fit for your ML Model](#)

Cheers!

[View original.](#)

Exported from [Medium](#) on August 2, 2020.