## GAN:

Generative Adversarial Networks (GANs) were introduced in 2014 by Ian J. Goodfellow and co-authors. GANs perform unsupervised learning tasks in machine learning. It consists of 2 models that automatically discover and learn the patterns in input data.

The two models are known as Generator and Discriminator.

 GANs consists of two ANN or CNN models:

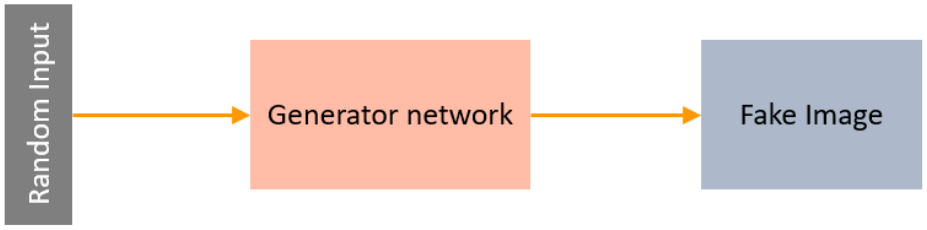
1. Generator Model: Used to generate new images which look like real images.
2. Discriminator Model: Used to classify images as real or fake

### The Generator Model

The Generator Model generates new images by taking a fixed size random noise as an input. Generated images are then fed to the Discriminator Model.

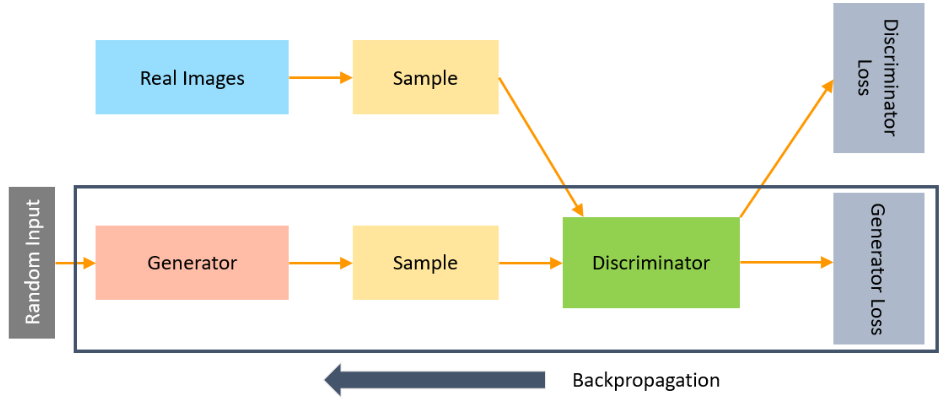
The main goal of the Generator is to fool the Discriminator by generating images that look like real images and thus makes it harder for the Discriminator to classify images as real or fake.

A Generator in GANs is a neural network that creates fake data to be trained on the discriminator



* The main aim of the Generator is to make the discriminator classify its output as real.
* generator loss, which penalizes the Generator for failing to dolt the discriminator

The backpropagation method is used to adjust each weight in the right direction by calculating the weight's impact on the output. It is also used to obtain gradients and these gradients can help change the generator weights.



### The Discriminator Model

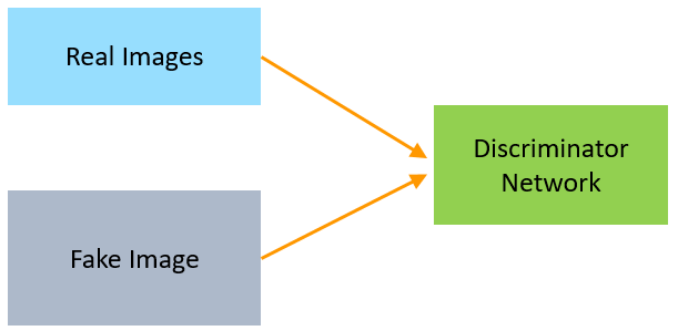
The Discriminator Model takes an image as an input (generated and real) and classifies it as real or fake.

Generated images come from the Generator and the real images come from the training data.

The discriminator model is the simple binary classification model.

The Discriminator is a neural network that identifies real data from the fake data created by the Generator. The discriminator's training data comes from different two sources:

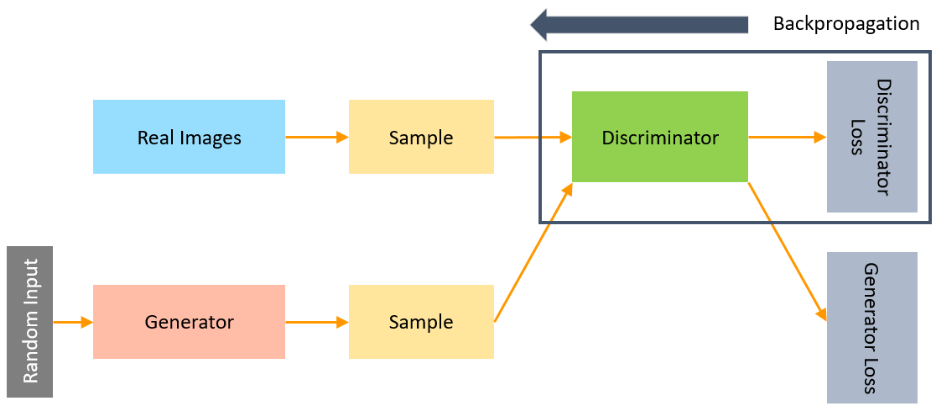
* The real data instances, such as real pictures of birds, humans, currency notes, etc., are used by the Discriminator as positive samples during training.
* The fake data instances created by the Generator are used as negative examples during the training process.



While training the discriminator, it connects to two loss functions. During discriminator training, the discriminator ignores the generator loss and just uses the discriminator loss.

In the process of training the discriminator, the discriminator classifies both real data and fake data from the generator. The discriminator loss penalizes the discriminator for misclassifying a real data instance as fake or a fake data instance as real.

The discriminator updates its weights through backpropagation from the discriminator loss through the discriminator network.



## How Do GANs Work?

GANs consists of two neural networks. There is a Generator G(x) and a Discriminator D(x). Both of them play an adversarial game. The generator's aim is to fool the discriminator by producing data that are similar to those in the training set. The discriminator will try not to be fooled by identifying fake data from real data. Both of them work simultaneously to learn and train complex data like audio, video, or image files.

The Generator network takes a sample and generates a fake sample of data. The Generator is trained to increase the Discriminator network's probability of making mistakes

## IMG_256

## Steps for Training GAN

* Define the problem
* Choose the architecture of GAN
* Train discriminator on real data
* Generate fake inputs for the generator
* Train discriminator on fake data
* Train generator with the output of the discriminator

Let us now look at the different types of GANs.

Vanilla GANs: Vanilla GANs have a min-max optimization formulation where the Discriminator is a binary classifier and uses sigmoid cross-entropy loss during optimization. The Generator and the Discriminator in Vanilla GANs are multi-layer perceptrons. The algorithm tries to optimize the mathematical equation using stochastic gradient descent.

**Deep Convolutional GANs (DCGANs)**: DCGANs support convolution neural networks instead of vanilla neural networks at both Discriminator and Generator. They are more stable and generate better quality images. The Generator is a set of convolution layers with fractional-strided convolutions or transpose convolutions, so it up-samples the input image at every convolutional layer. The discriminator is a set of convolution layers with strided convolutions, so it down-samples the input image at every convolution layer.

**Conditional GANs**: Vanilla GANs can be extended into Conditional models by using extra-label information to generate better results. In CGAN, an additional parameter ‘y’ is added to the Generator for generating the corresponding data. Labels are fed as input to the Discriminator to help distinguish the real data from the fake generated data.

**Super Resolution GANs**: SRGANs use deep neural networks along with an adversarial network to produce higher resolution images. SRGANs generate a photorealistic high-resolution image when given a low-resolution image.

# Amazing Applications of GAN

Let us discuss some amazing applications of GANs other than image generation.

## Image to Image Translation

demonstrates GANs as many images to image translation tasks.  
[https://editor.analyticsvidhya.com/uploads/73176img-to-img-translation.PNG](https://editor.analyticsvidhya.com/uploads/73176img-to-img-translation.PNG" \t "https://www.kaggle.com/general/_blank)

## Text to Image Translation

demonstrates a way to generate images from text.  
[https://editor.analyticsvidhya.com/uploads/37782text-image-translation.PNG](https://editor.analyticsvidhya.com/uploads/37782text-image-translation.PNG" \t "https://www.kaggle.com/general/_blank)

## Photos to Emojis

Types of GAN:

- Standard: GAN, DCGAN.

- Conditional: cGAN, SS-GAN, InfoGAN, ACGAN.

- Loss: WGAN, WGAN-GP, LSGAN.

- Image Translation: Pix2Pix, CycleGAN.

- Advanced GANs: BigGAN, PG-GAN, StyleGAN.

- Other: StackGAN, 3DGAN, BEGAN, SRGAN, DiscoGAN, SEGAN

**Recommendation to add below things to improve model**

- Downsample Using Stride Convolutions

- Upsample Using Stride Convolutions

- Use LeakyReLU

- Use Batch Normalization

- Use Gaussian Weight Initialization

- Use Adam Stochastic Gradient Descent

- Scale Images to the Range [-1,1]

- Use a Gaussian Latent Space

- Separate Batches of Real and Fake Images

- Use Label Smoothing

- Use Noisy Labels

Original sources: [How to Train a GAN? Tips and tricks to make GANs work, ganhacks, PyTorch](https://github.com/soumith/ganhacks)

GAN Applications

Here is another great blog, discussing the application of GAN by [Jonathan Hui](https://medium.com/@jonathan_hui) - [GAN — Some cool applications of GAN](https://medium.com/@jonathan_hui/gan-some-cool-applications-of-gans-4c9ecca35900), namely:

- Create Anime characters

- Pose Guided Person Image Generation

- CycleGAN

- StarGAN

- PixelDTGAN

- Super-resolution

- The progressive growing of GANs

- StyleGAN2

- High-resolution image synthesis

- GauGAN

- Text to Image Synthesis

- Face synthesis

- Image inpainting

- Learn Joint Distribution

- DiscoGAN

- Pix2Pix

- DTN

- Texture synthesis

- Image editing (IcGAN)

- Face aging (Age-cGAN)

- DeblurGAN

- Neural Photo Editor

- Object detection

- Image blending

- Video generation

- Generate 3D objects

- Music generation

- Medical (Anomaly Detection)