Lecture 6 : General Adverserial Networks 🌱

Class: GAN

Date: 07.35 AM, Today

Author: Lasal Hettiarachchi

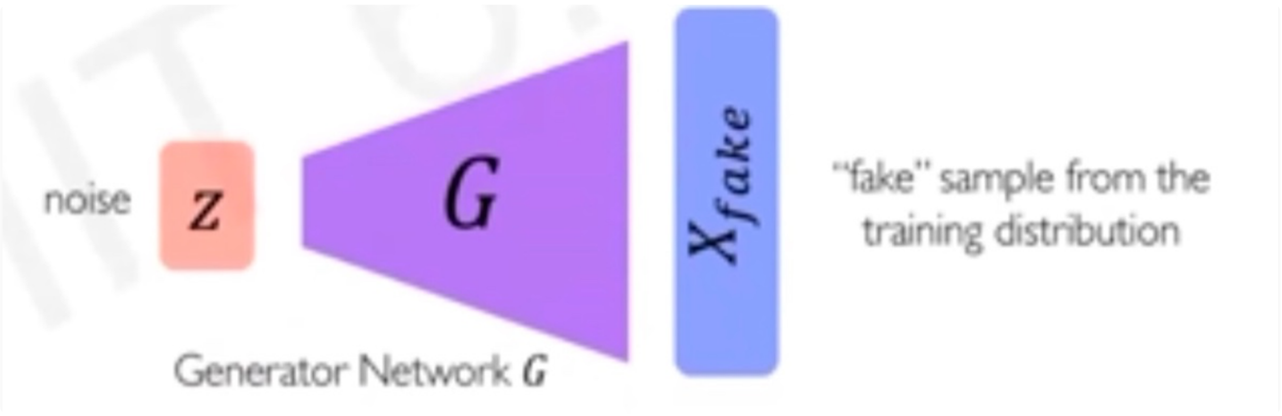
Key learnings:

- Stack autoencoders
- Denoising autoencoders
- Variational autoencoders

idea : don't explicitly model density, and instead just sample to generate new instances. (Unlike in auto encoders we do not have the encoding part to determine Z. Instead we use a random(noise) value and try to generate the output)

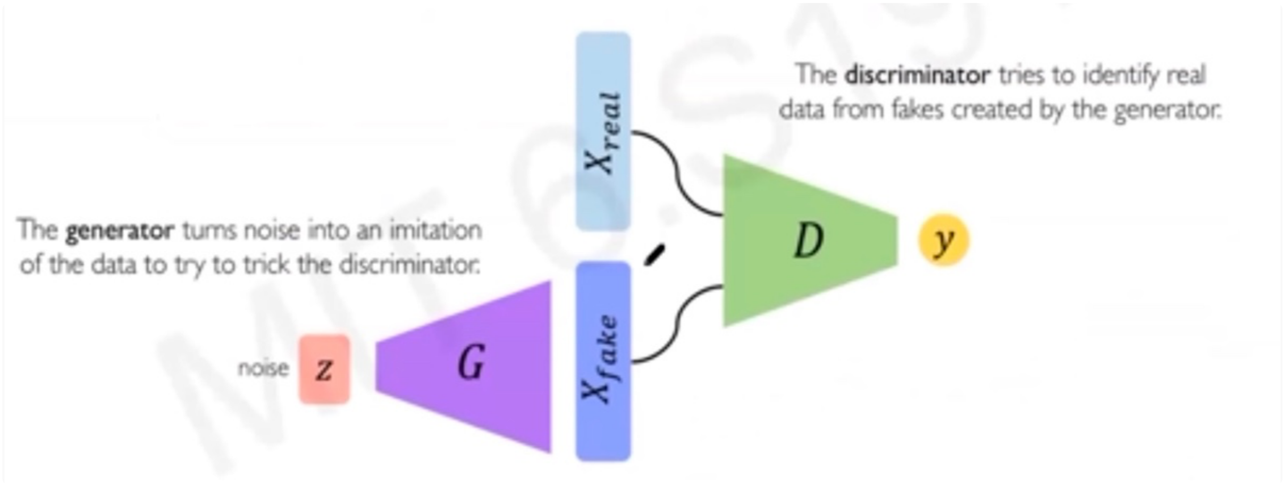
Problem: want to sample from complex distribution - can't do this directly!

Solution: sample from something simple (noise), leam a transformation to the training distribution.



GAN

GANs solve this problem by combining generators and discriminators. these 2 neural networks complete one another.


The **discriminator** tries to identify real data from fakes created by the generator.
The **generator** turns noise into an imitation of the data to try to trick the discriminator.
noise
z
G
X_{fake}
X_{real}
D
y

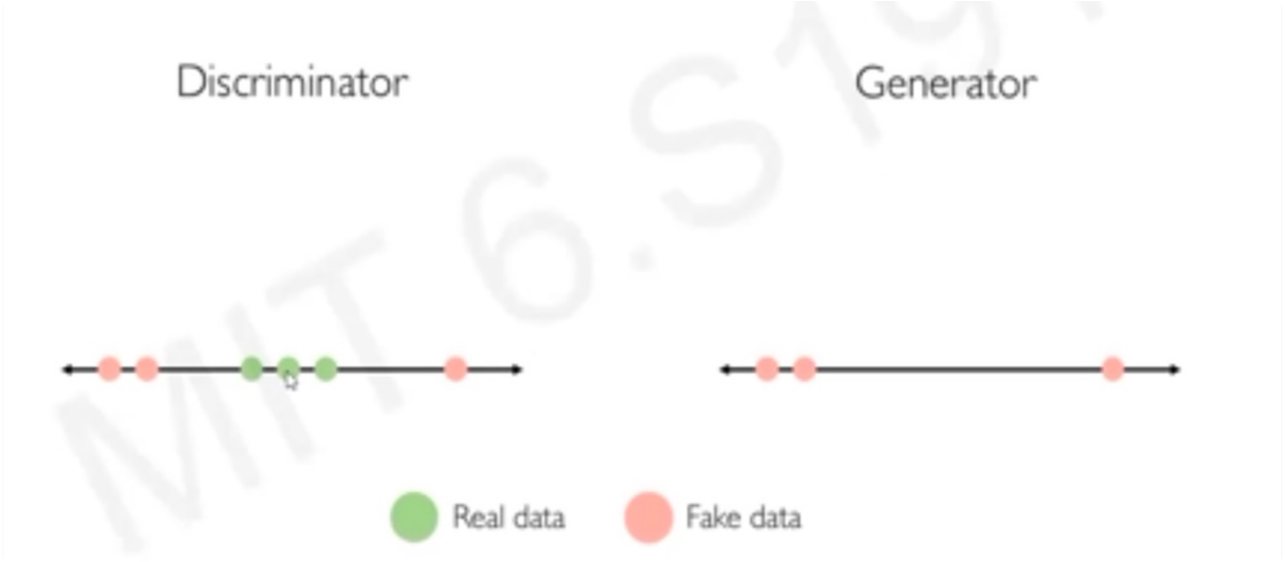
Discriminator : Takes the Xreal and Xgenerated and compares the 2 and gives out the vector of loss values.

After the loss is calculated, the gan will update the parameters in D and G by back proporgation.

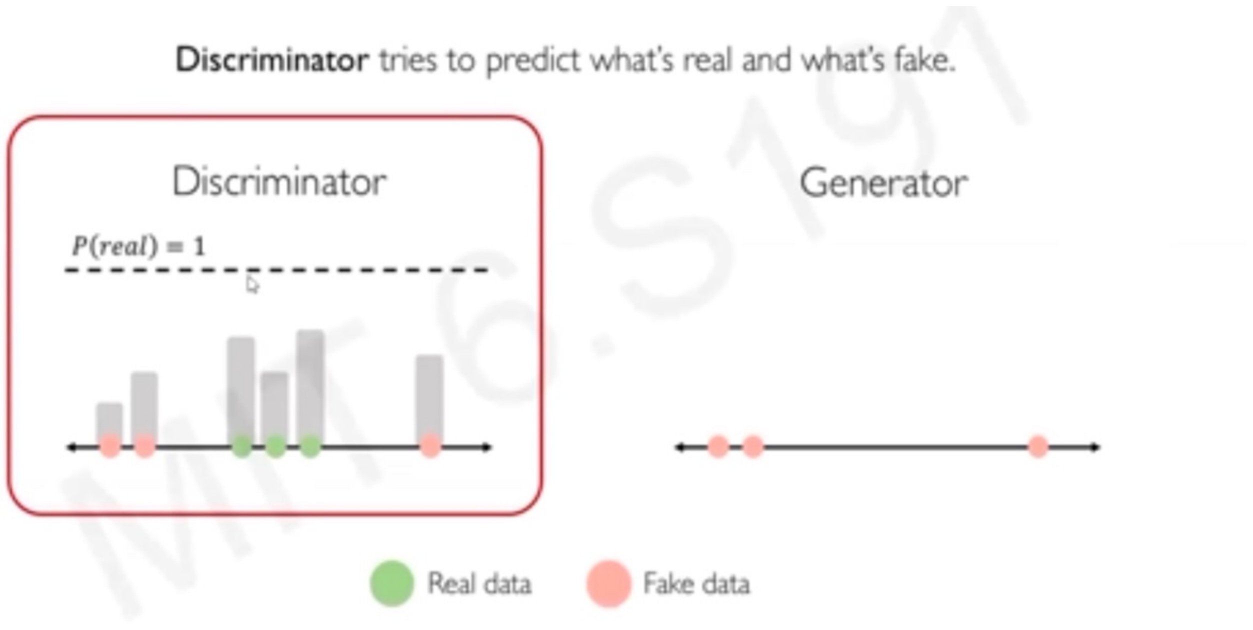
So what is the loss that is calculated?

Intuition behind GANs

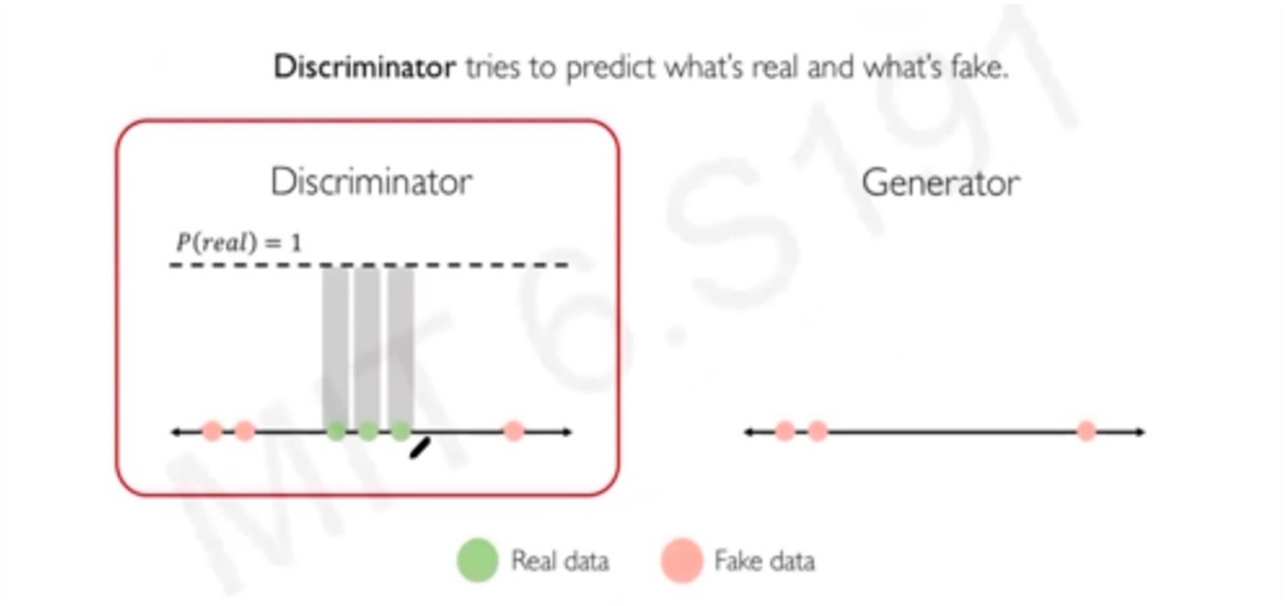
Discriminator looks at both real data and fake data created by the generator.



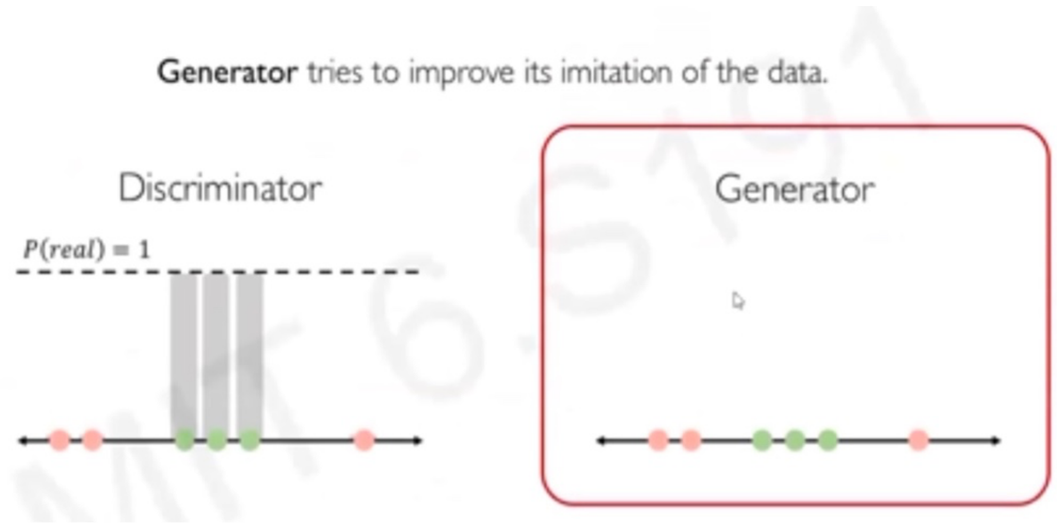
Discriminator then learns to predict what the real values are and whats fake.



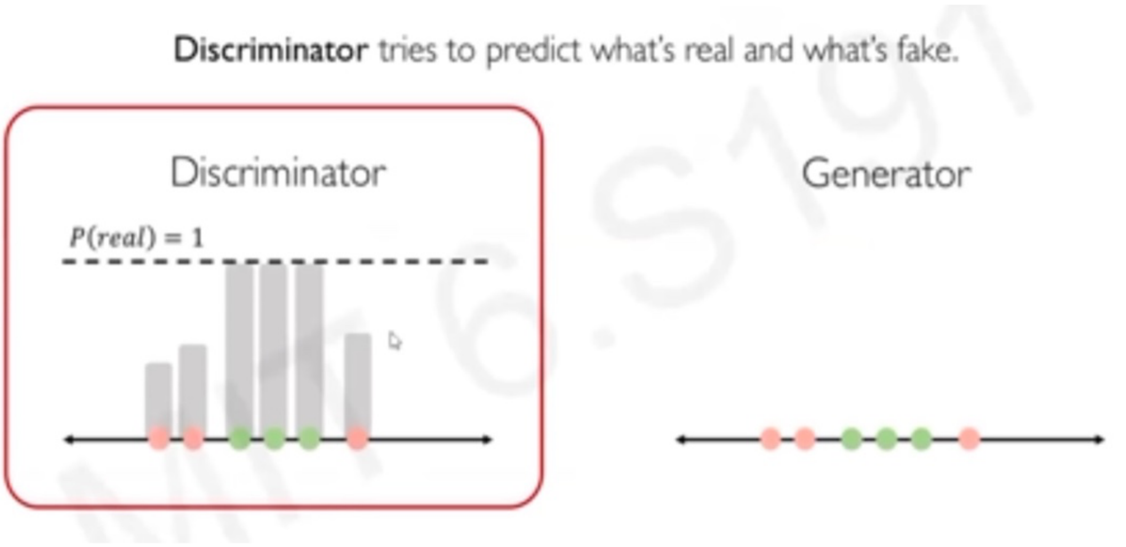
After several iterations the Discriminator will learn to give the real ones the probabolity of 1 and the fake ones the probability of 0.



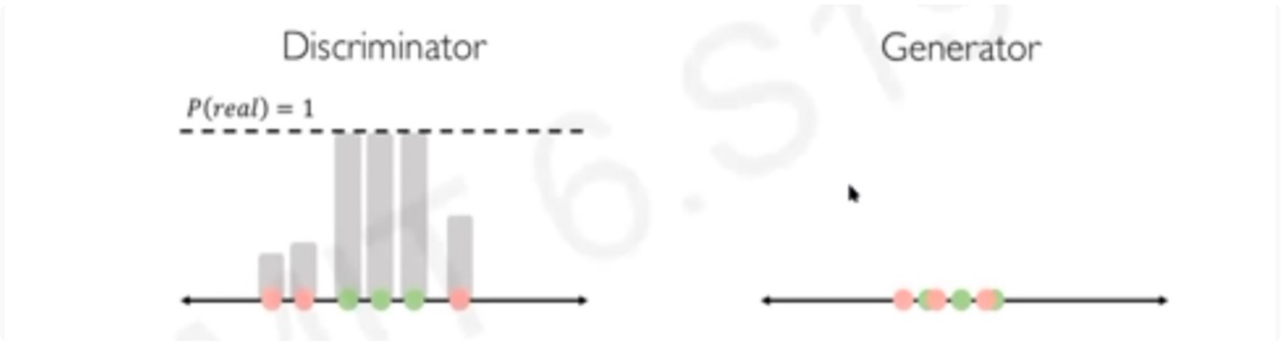
We define the loss ST the above process can be achieved. After that the discriminator knows how to differentiate real data from fake data. Next it is needed to train the generator to bring the fake data close to the real data.



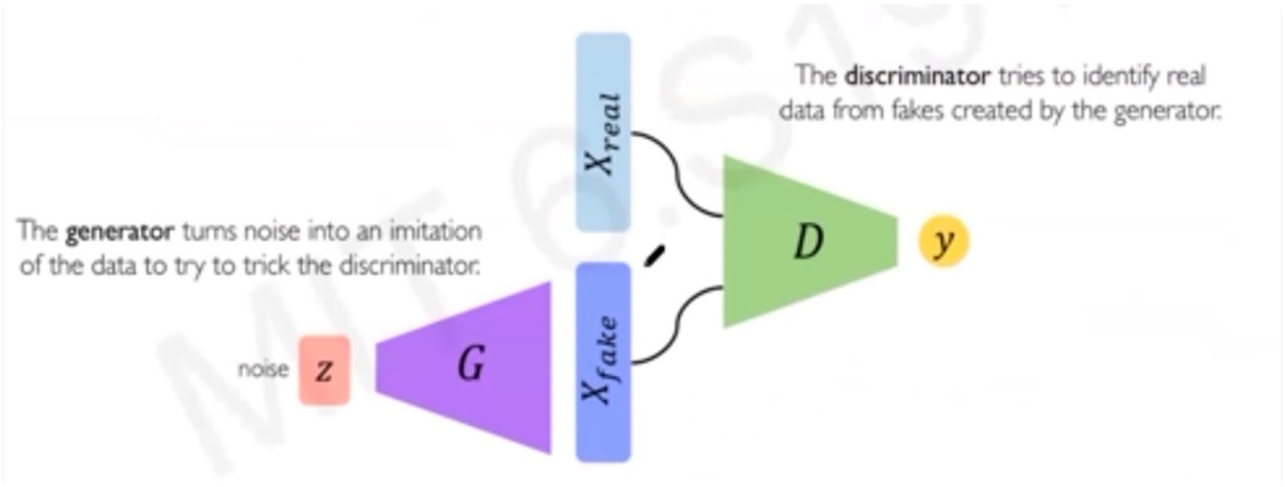
After the fake data is brought closer , again the discriminator predicts the real values and the fake values



And similar to the previous steps, the generator moves the points closer.



So what is the loss that is calculated?


The discriminator tries to identify real data from fakes created by the generator.
The generator turns noise into an imitation of the data to try to trick the discriminator.
noise
z
G
X_{real}
X_{fake}
D
y

What should the discriminator do ?

- It should give 1 to the real img and 0 to the fake ones.
- If the discriminator actually does a correct job, the loss should be 0

Thus the loss can be given as,

LD = (1 - Xr) + G(Xf)

This can be used as the obj to update parameters of D

What should the Generator do ?

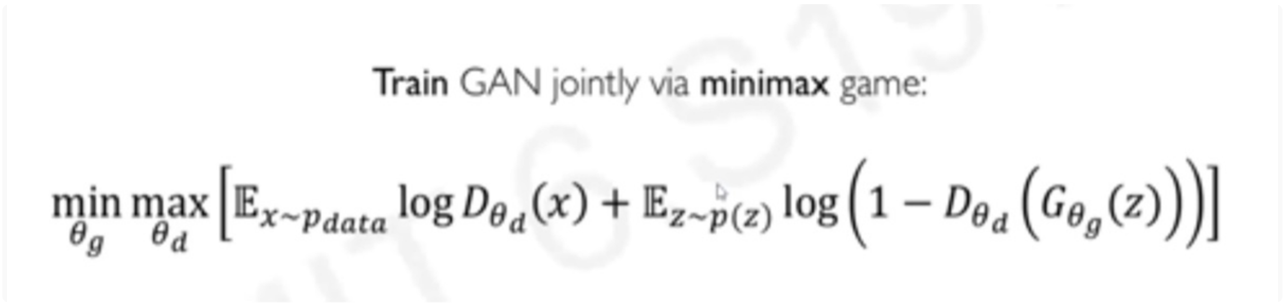
- To convince the discriminator that the generated image is a real image. which means that it should try to get the value of Xf close to 1.

Thus the loss can be given as

LG = 1 - G(Xf)

G is usually the sigmoid func. If the discriminator has learned well it will give 1 as the value

Considering both the above loss values togerther and taking to account the cross enntropy values (thus the log). The log function can be given as,



Train GAN jointly via **minimax** game:

$$\min_{\theta_g} \max_{\theta_d} \left[\mathbb{E}_{x \sim p_{data}} \log D_{\theta_d}(x) + \mathbb{E}_{z \sim p(z)} \log \left(1 - D_{\theta_d}\left(G_{\theta_g}(z)\right)\right)\right]$$

After training the generator can just be used to generate the data.

Advantages:

- Once training is done, generation process is faster
- Inference rate is also high

Disadvantages:

- There are no ways to compare between different GANs to see whats better
- Its hard to use the output to re-engineer the z values.
- Non-convergence: the model parameters oscillate, destabilize and never converge,
- Mode collapse: the generator collapses which produces limited varieties of samples,
- Diminished gradient: the discriminator gets too successful that the generator gradient vanishes and learns nothing,
- Unbalance between the generator and discriminator causing overfitting.
- Highly sensitive to the hyperparameter selections.

Applications of GANs

- Editing photos (black white → colour)
- AR applizations
- Generate environments for games