Generative Adversarial Models

Pranabendu Misra

Advanced Machine Learning 2022

Generative Adversarial Network

- Another type of *Generative* models, like autoencoders.

Generative Adversarial Network

- Another type of *Generative* models, like autoencoders.
- **The generation step is at the heart of this model**.

Generative Adversarial Network

- Another type of *Generative* models, like autoencoders.
- **The generation step is at the heart of this model**.

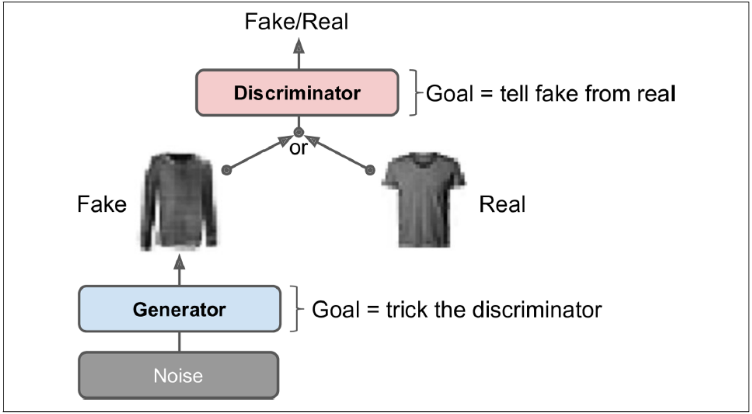


Figure 17-15. A generative adversarial network

Generative Adversarial Network

- Based on a turn-based-game between 2 players

Generative Adversarial Network

- Based on a turn-based-game between 2 players
- **Generator**: A Neural Network which given some *random noise* it generates a *fake datapoint*.
- **Discriminator**: A Neural Network that given an image determines if it is real or fake.

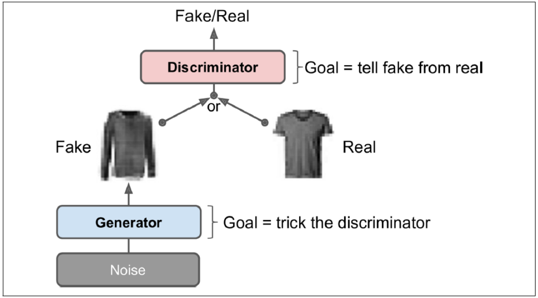
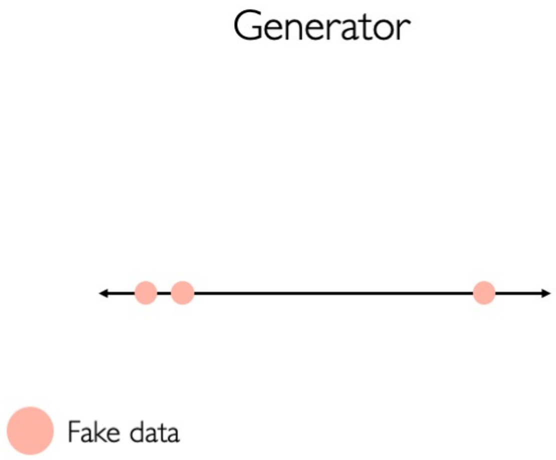


Figure 17-15. A generative adversarial network

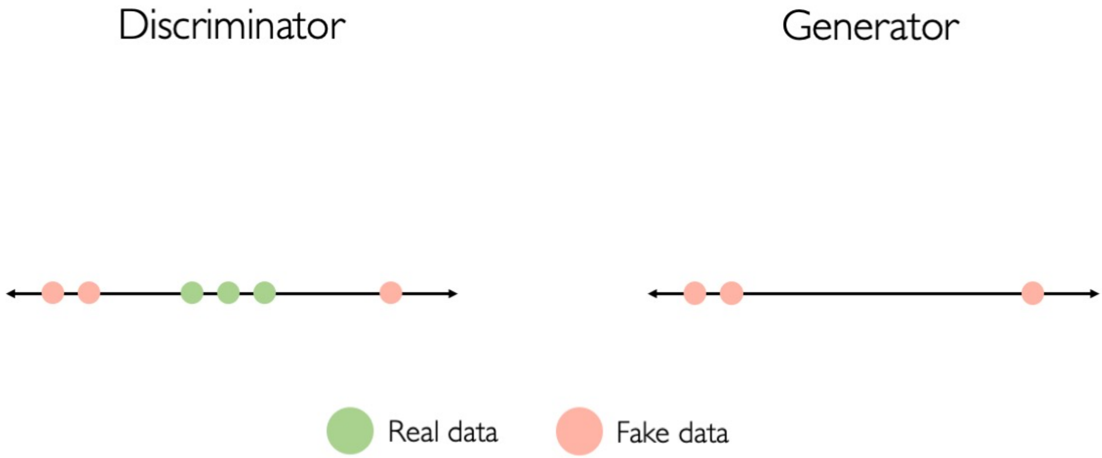
This is a game-theoretic modeling of the problem.

- Generator and Discriminator are both trying to outwit each other.
- In this process, the Generator learns to produce very good *fake data*.

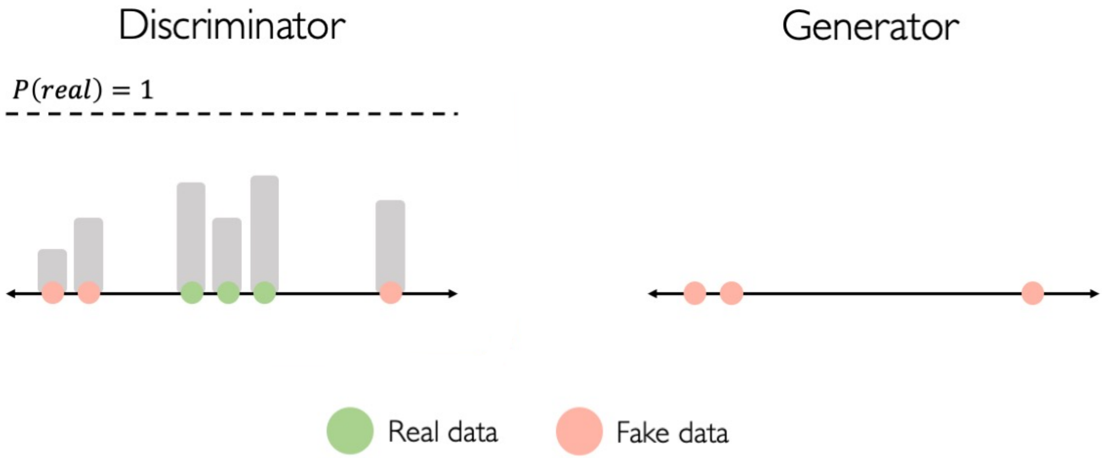
GAN: Learning



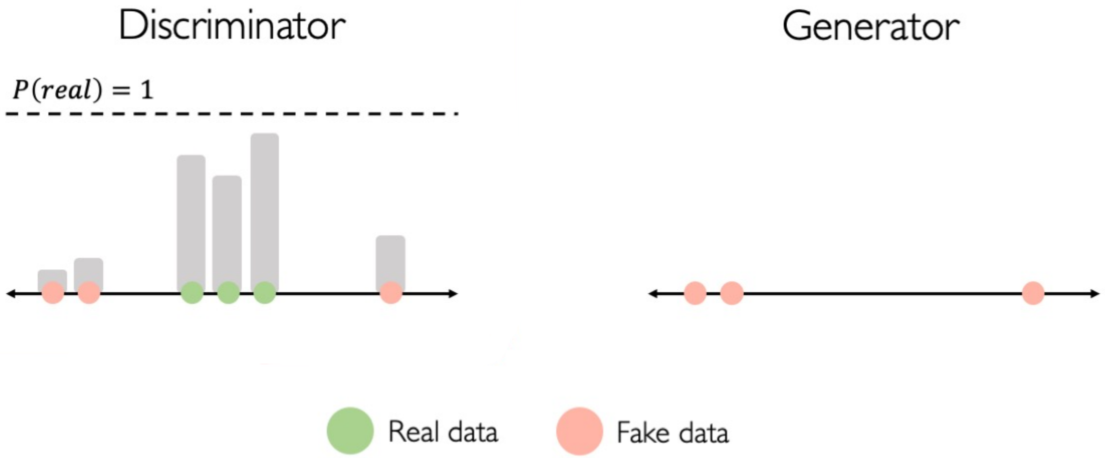
GAN: Learning



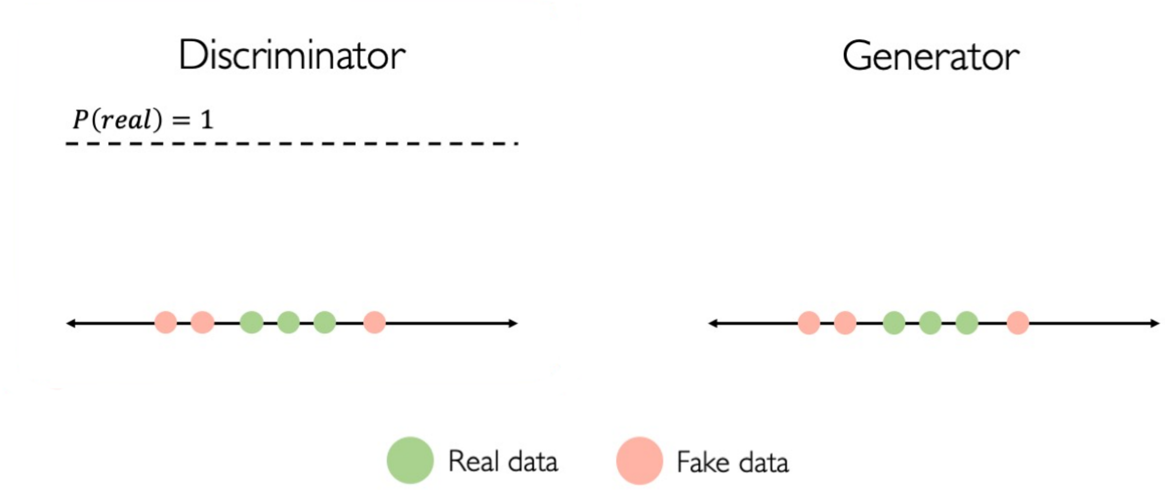
GAN: Learning



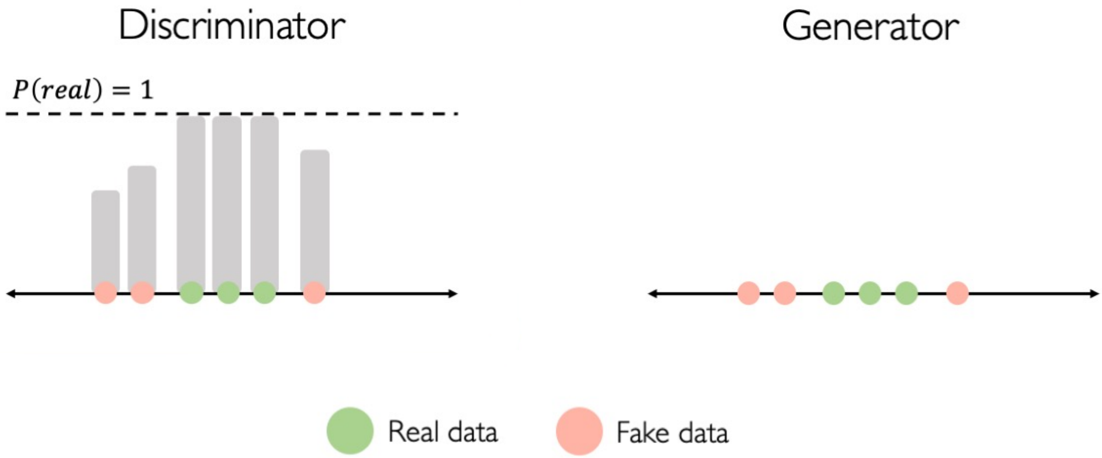
GAN: Learning



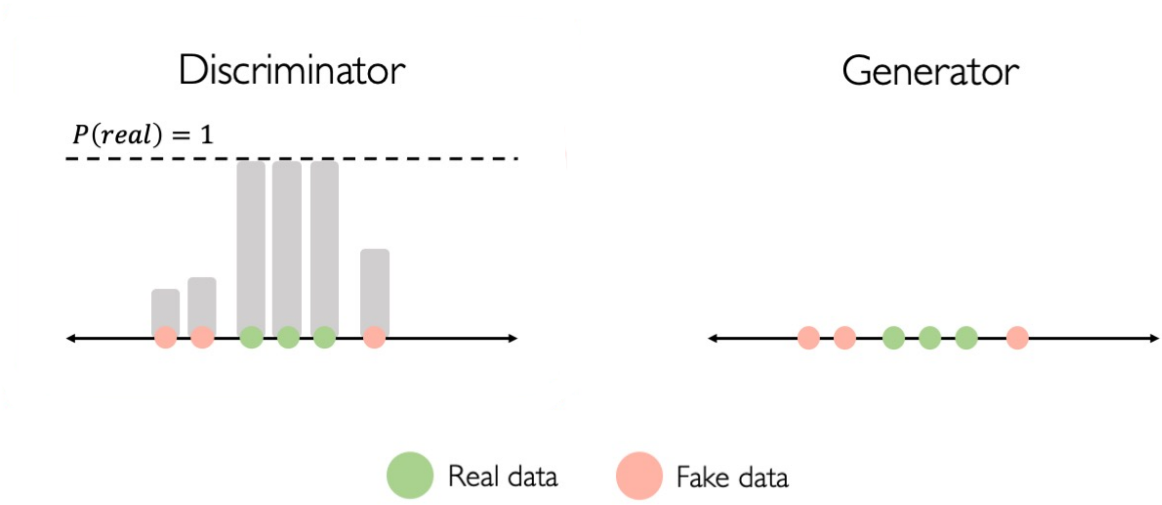
GAN: Learning



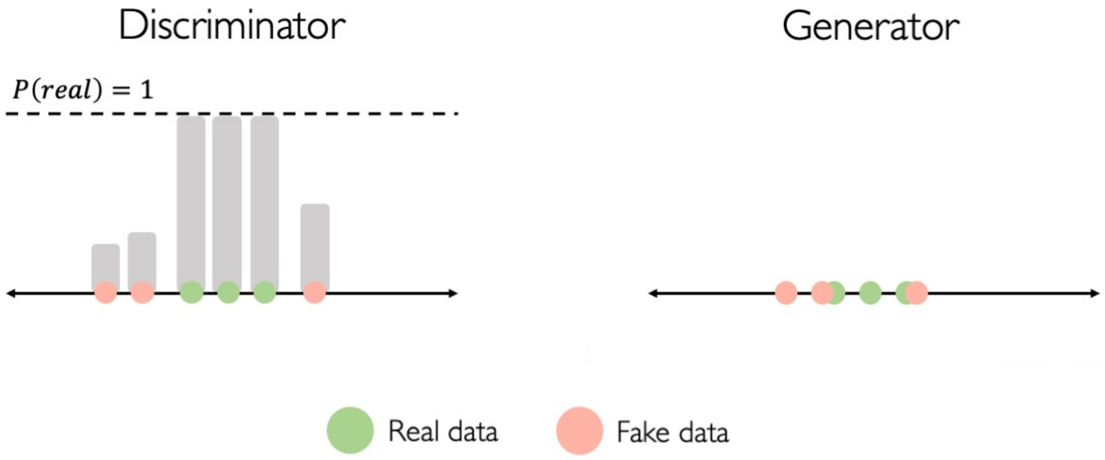
GAN: Learning



GAN: Learning



GAN: Learning



- **Generator**: A Neural Network which given some *random noise* it generates a *fake datapoint*.
- **Discriminator**: A Neural Network that given an image determines if it is real or fake.

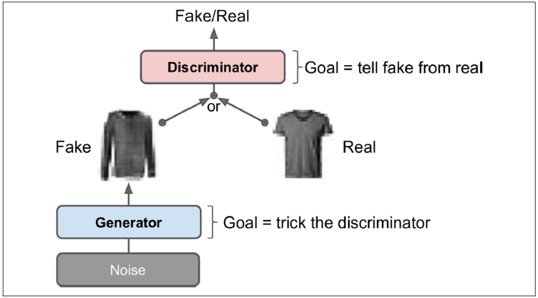


Figure 17-15. A generative adversarial network

Generative Adversarial Network: Loss Functions and Training

- A *Sampler* S samples a vector z in the *latent space* using a normal distribution.
- The Generator G maps z to a data point.

$$\hat{x} = G(z)$$

- The Discriminator D gets $\{\hat{x}, x\}$ where x is a training sample, i.e. a real datapoint.
- D outputs a probability for both $\hat{x}$ and x of them being real (i.e. not generated by G)

$$\hat{y} = D(\hat{x})$$

$$y = D(x)$$

- Note that $\hat{y}$ and y are probabilities

Generative Adversarial Network: Loss Functions and Training

- Discriminator's Loss:

$$\log(D(G(z))) + (1 - \log(D(x)))$$

- Generator's Loss:

$$1 - \log(D(G(z)))$$

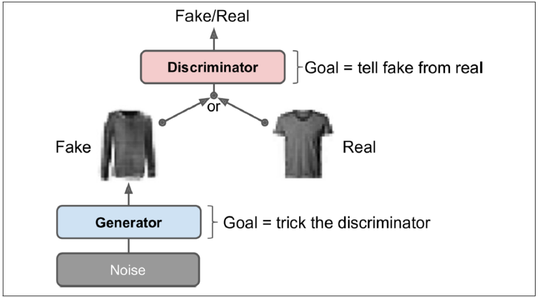


Figure 17-15. A generative adversarial network

Generative Adversarial Network: Loss Functions and Training

- Discriminator's Loss:

$$\log(D(G(z))) + (1 - \log(D(x)))$$

- Generator's Loss:

$$1 - \log(D(G(z)))$$

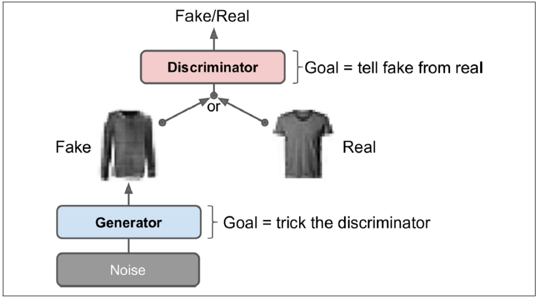


Figure 17-15. A generative adversarial network

- We train in rounds
- The Generator produces a batch of fake data $G(z)$
- A batch of real data x and $G(z)$ are both run through the discriminator, and loss $\log(D(G(z))) + (1 - \log(D(x)))$ is computed.
- We then update D via a gradient descent step, keeping G fixed.

- Discriminator's Loss:

$$\log(D(G(z))) + (1 - \log(D(x)))$$

- Generator's Loss:

$$1 - \log(D(G(z)))$$

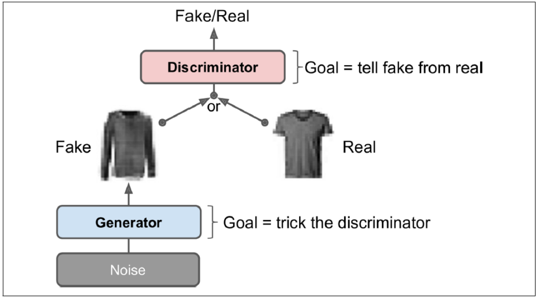


Figure 17-15. A generative adversarial network

- Next, the generator then produces a new batch of fake data $G(z')$
- These are processed by the discriminator to get the loss $\log(D(G(z')))$.
- We then update G by a gradient descent step, keeping D fixed.

Generative Adversarial Network: Loss Functions and Training

- Training GANs can be difficult, and we have to be very careful
- The training succeeds if we reach a good equilibrium between the Generator and the Discriminator.

Generative Adversarial Network: Loss Functions and Training

- Training GANs can be difficult, and we have to be very careful
- The training succeeds if we reach a good equilibrium between the Generator and the Discriminator.

Some issues that may arise:

- *Mode Collapse:* The Generator only produces a small set of distinct images.
- *Convergence Failure:* The Generator produces poor quality images even after training for a long time. Could happen because the opponent network always counters whatever improvements you make, blocking every direction of progress in gradient descent
- *Losses don't indicate progress:* Even as the generator is improving, so is the discriminator. So the loss of the Generator may keep increasing even though it is getting better.

GAN: Example

GANs for image synthesis: latest results

References:

These slides are based on:

- http://introtodeeplearning.com/slides/6S191_MIT_DeepLearning_L4.pdf
- Generative Adversarial Networks, Goodfellow at al, Communications of ACM, November 2020