Generative Adversarial Networks

S. Dakurah
03/11/20

Outline

Outline

What are GANs?

- A class of ML techniques
- Involves two simultaneously trained models adversarial to each other
- One (generator) is trained to generate fake data whiles the other (discriminator) is trained to discern between fake and real

(1) Generative
  - This captures the whole purpose of this class of ML techniques - generate data

(2) Adversarial
  - This speaks to the competition between the two models

(3) Networks
  - This simply points to the class of ML models[1] commonly used

[1] ANN & it's variants

How GANs work at a high level

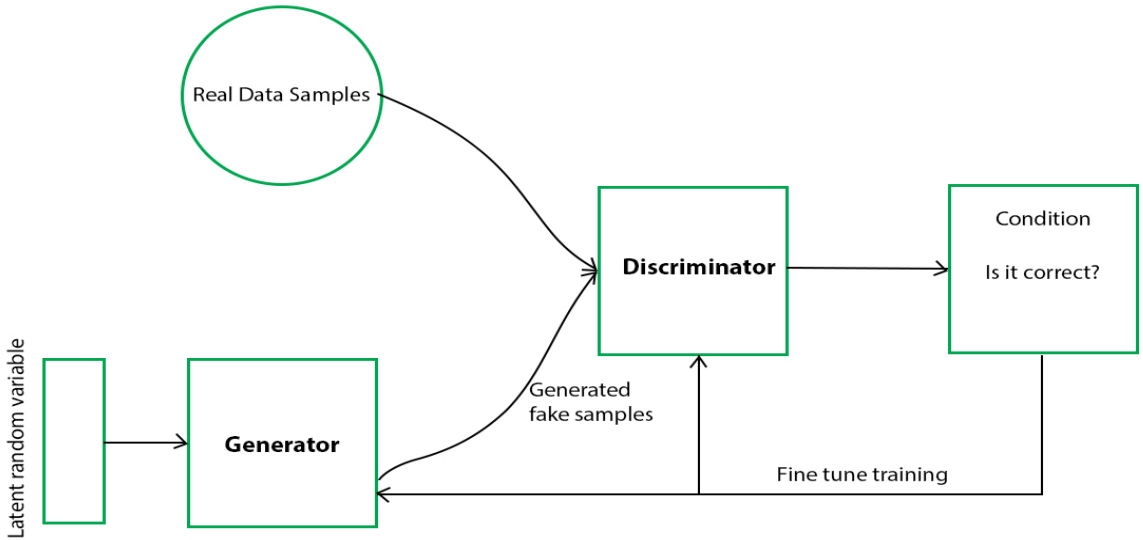

Figure: Illustration of a GAN Training Process

(1) Generator
- **Input:** A vector of random inputs - no structure initially
- **Output:** Fake examples that are as close as possible to the real ones

(2) Discriminator
- **Input:** Real examples or fake examples
- **Output:** Probability that the input example is real

Outline

Training Setup

- Generator & Discriminator are represented by differentiable functions[2]
- The two functions train[3] on the discriminators loss, but each has it's own cost function
- The Discriminator aims at minimizing it's loss, whiles the Generator aims at maximizing the Discriminators loss

- The Generator will only learn to capture the data distribution of the training samples.
  e.g. If you want to generate car images, the training sample should be made of car images

[2] ANN & it's variants
[3] Using backpropagation

Cost Function

- Let $J^{(G)}$ AND $J^{(D)}$ be the cost function for the Generator and Discriminator
- GANs differ from traditional NN in two key respects
  (1) The cost function of a traditional NN is exclusively made-up of the parameters of that network.
  This is not the case for GANs.
  e.g. the Generators cost function will be of the form $J^{(G)}\left(\theta^{G}, \theta^{D}\right)$
    - θ^{G} and θ^{D} are the set of parameters for the generator and discriminator
  (2) A traditional neural network can tune all it's parameters in the training process.
  This is not the same for GANs as each network can only tune it's parameters

Training Process

- Let $D(x)$ and $D(G(z))$ be the discriminators output for a real example x and a random vector z
- $D(x) = P(truelabel|x) \longrightarrow maximize$ and $D(G(z)) = P(truelabel|G(z)) \longrightarrow minimize$ for a Discriminator Network
- $D(G(z))$ should be maximized for a Generator network
- Assuming a cross entropy loss/log loss[4] function:

[4] Measure the performance of a classification model whose output is a probability value between 0 and 1

Training Process

- At Discriminator D:
  - $Dloss_x = log(D(x))$
  - $Dloss_{G(z)} = log(1 - D(G(z)))$
  - $Dloss = log(D(x)) + log(1 - D(G(z)))$
- At Generator G:
  - $Gloss_x = log(1 - D(G(z)))$

- You can average over all training examples to get the cost function
- The discriminator network runs twice, whiles the generator runs once
- The stopping criterion will be when the two networks reach a **Nash equilibrium**

Outline

References

1. GANs in Action $\sim$ Jakub Langr & Vladimir Bok
   - Chapter 1, 2, & 3