GAN Architecture
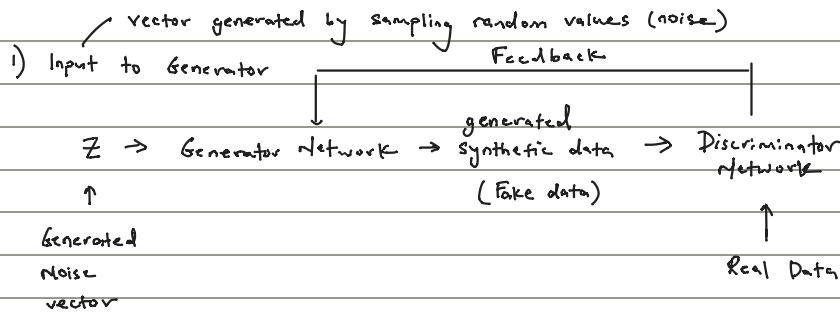
1. Generator network : Generate output convincing enough to fool discriminator
   └ Improves on feedback provide by Generator

2. Discriminator network : Try its best to identify synthetic data
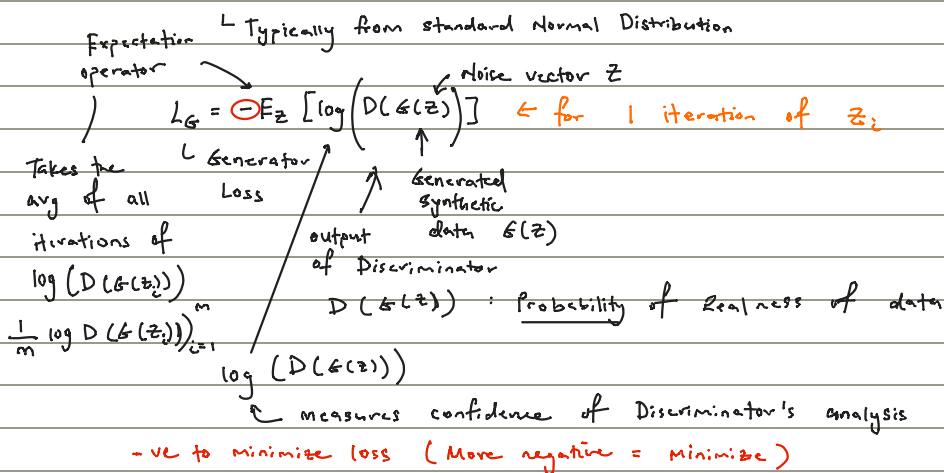   └ Provides feedback to Generator

GAN Flow   Noise (Z)

vector generated by sampling random values (noise)

1) Input to Generator

Feedback

$Z \rightarrow$ Generator Network $\rightarrow$ generated synthetic data $\rightarrow$ Discriminator Network

(Fake data)

Generated Noise vector

Real Data

2) Repeated iterations to produce synthetic data == Real data

How Generator works (Cross Entropy of fake data)

1) Noise vector Z
   └ Typically from standard Normal Distribution

Expectation operator

Noise vector Z

$$L_G = \ominus E_Z \left[\log \left(D(G(Z)) \right) \right] \quad \leftarrow \text{ for } 1 \text{ iteration of } Z_i$$

└ Generator Loss

Takes the avg of all iterations of $\log(D(G(z_i)))$

$\frac{1}{m} \log D(G(z_i))\big|_{i=1}^{m}$

output of Discriminator

Generated Synthetic data $G(Z)$

$D(G(Z))$: Probability of Realness of data

$\log (D(G(Z)))$
 └ measures confidence of Discriminator's analysis

– ve to Minimize loss (More negative = Minimize)

How Discriminator works (Cross entropy of Real & Fake data)

To minimize loss

confidence of discriminator (for real data

Probability of synthetic being fake

∴ $D(G(Z))$ is probability of synthetic being real

$$L_D = \ominus E_x \left[\log(D(x)) \right] + E_Z \left[\log(1 - D(G(Z))) \right]$$

Expectation operator (Avg of all values)

Real data x