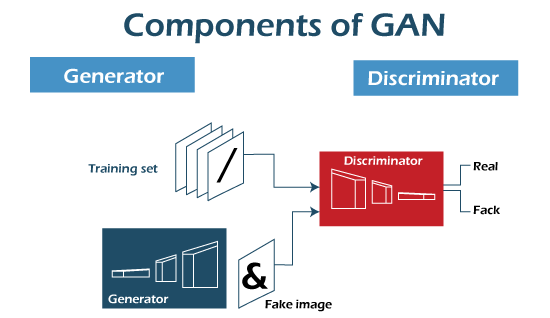
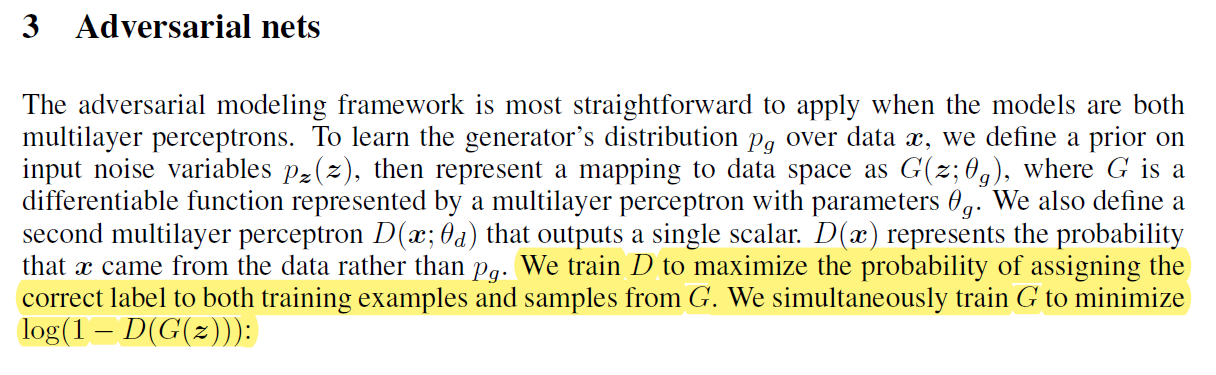
**GANs**

Generative Adversarial Network





Two-player minimax

* To train **D**, we need to maximize the probability of assigning the correct label to both training examples and samples from
* To train **G**, we need to minimize .

Therefore, the following loss functions are defined:

Jensen Shannon Divergence

Symmetrized

Pseudocode done.

**Conditional GAN**

Context

**Cycle GAN**

Write pseudocode

2 generators and 2 discriminators

A to B, B to A

**Wasserstein GAN**

It uses Wasserstein Distance instead of Jensen-Shannon Divergence. More stable and better properties.

Wasserstein Distance:

“It's a metric that measures how much "mass" from one probability distribution needs to be moved to transform it into another, while minimizing the transportation cost.”

Find 1-Lipschitz (bounded slopes stabilization)

Weight Clipping

Discriminator weights are limited [-c, c]

**Style GAN**

Introduces Mapping Network (maps noise z to intermediate latent space W)

"AdaIN" (Adaptive Instance Normalization) injects style information into each layer. Controls details.

Backbone network: sort latent space