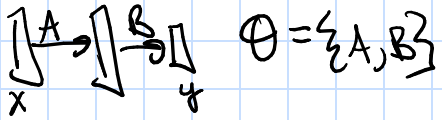
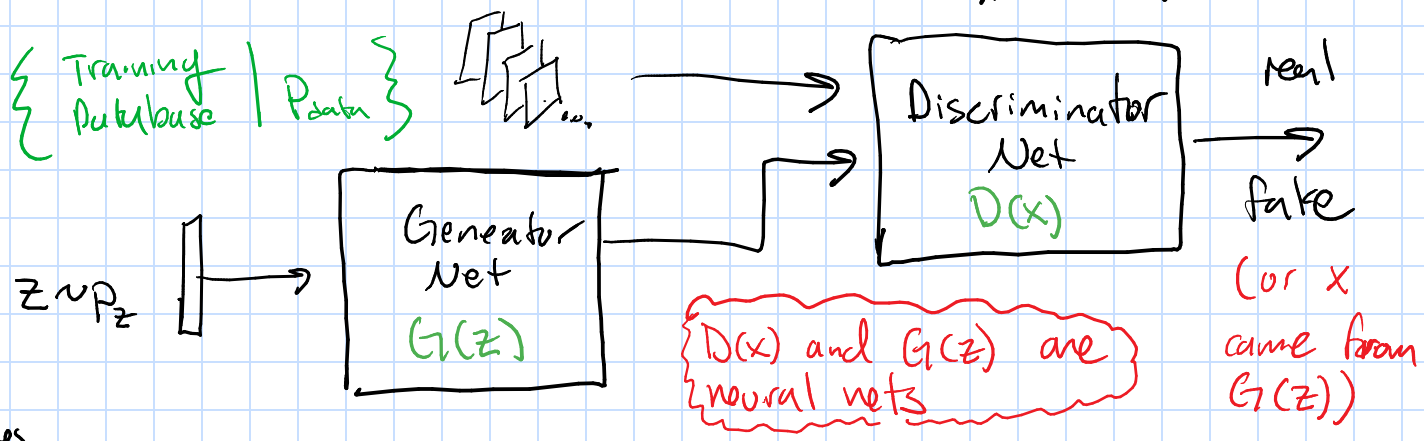
Generative Adversarial Nets (03/31/2021)





Notes

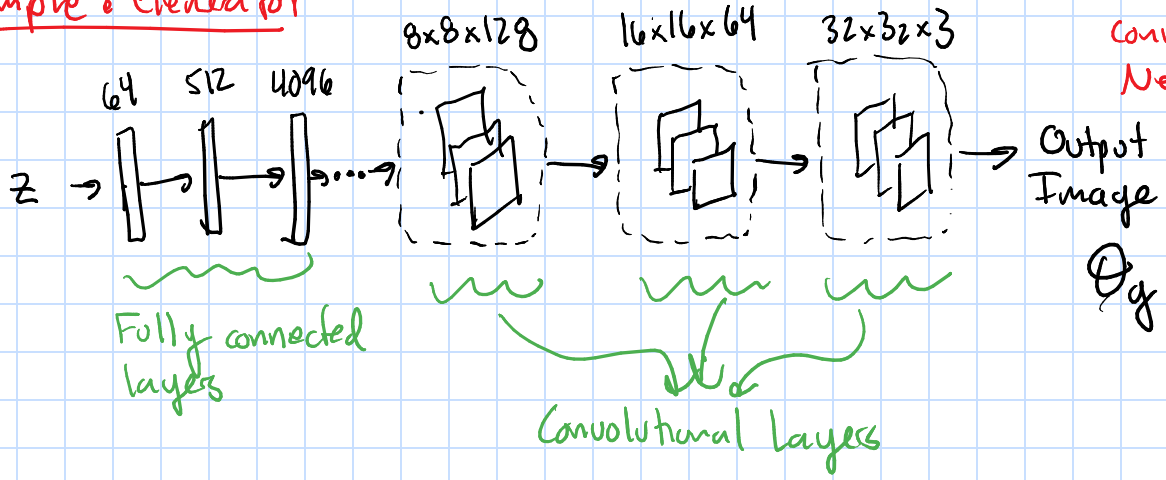
- $G(z; \theta_g)$ or $G(z)$ is a generative model that captures the distribution of the data
- $D(x; \theta_d)$ or $D(x)$ is a discriminative model that estimates the probability a sample came from p_{data} or $G(z)$
- $G(z)$ is trained to maximizes the probability that D makes a mistake
    - This is a two player game w/ G converging to the training distribution and D equal to $1/2$ everywhere

Solving this optimization task

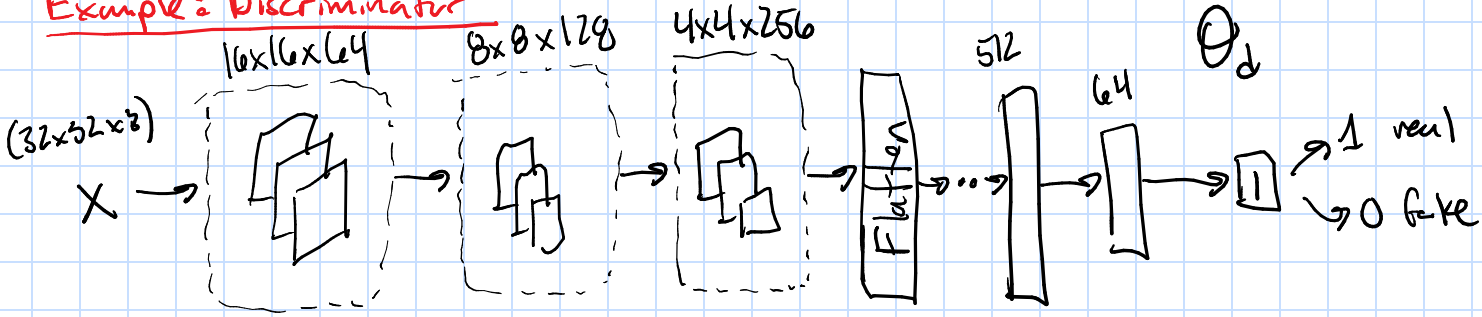
- $G(z)$ and $D(x)$ are differentiable functions
- ~~We~~ train $G(z)$ to minimize $\log(1 - D(G(z)))$ {over p_z}
- ~~We~~ train $D(x)$ to maximize $\log D(x)$ {over p_{data}}

Example: Generator

★ Transpose Convolutional Net ★



Example: Discriminator



Two player minimax w/ $V(G,D)$

$$\min_G \max_D V(G,D) = \mathbb{E}_{x \sim p_{data}}[\log D(x)] + \mathbb{E}_{z \sim p_z}[\log(1 - D(G(z)))]$$

$$\theta_g \leftarrow \theta_g + \Delta\theta_g$$

$$\theta_d \leftarrow \theta_d + \Delta\theta_d$$