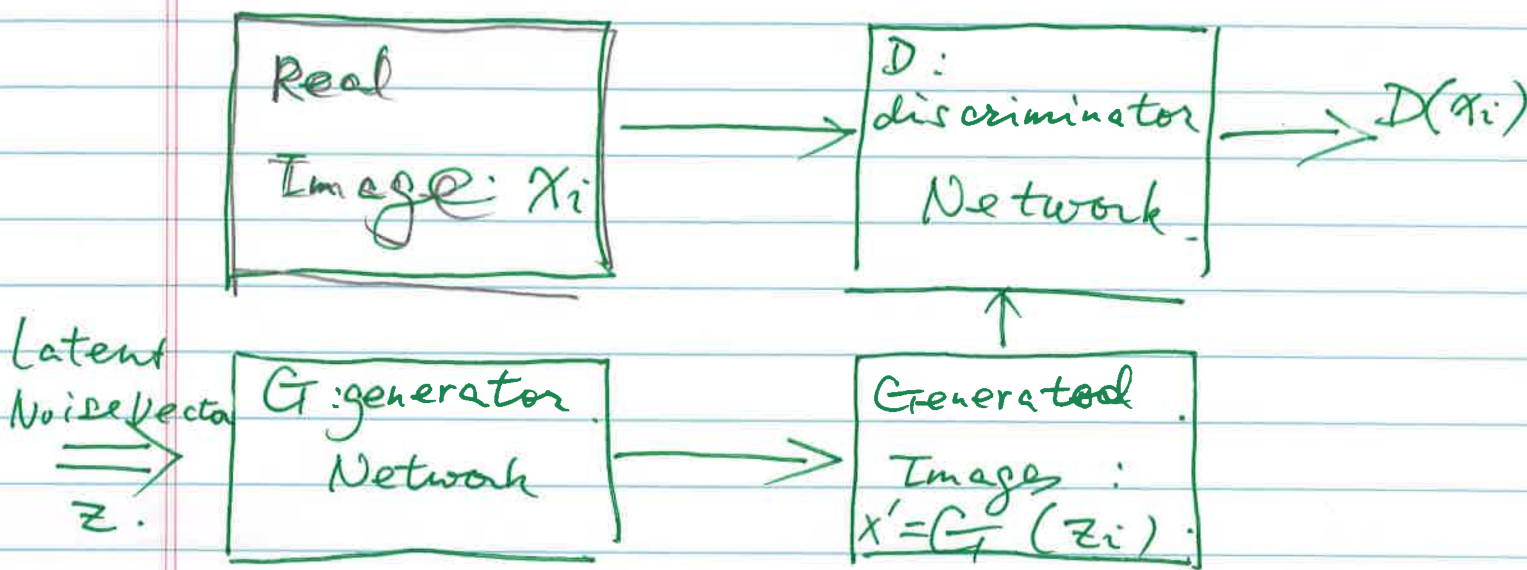
# Derivation of GAN from basic Probability Theory — Yu Liang




## I: Definition: & Basic Term.

① $D(x)$: the probability of image $x$ is identified "Real" ($D: R^n \to [0,1]$).

$\Rightarrow 1 - D(x)$: the probability of image $x$ is identified as "Fake".

② $\mathbb{E}$: $\begin{cases} \text{cross\_Entropy} \\ \text{or} \\ \text{mean value.} \end{cases}$

③ $G(z)$: the image generated from latent vector $z$.

④ $z \sim P_{generator}$ :

generator's Distribution

$x \sim P_{data}$ :

The Real data Distribution

## II : Training of Discriminator .

### II . 1 : Input & Output Data

| | Input | | Model's output ($Y$) | Label ($Y'$) |
|---|---|---|---|---|
| Real Image | $x_1$ | $X_1$ | $D(X_1)$ | 1 |
| | $X_2$ | $X_2$ | $D(X_2)$ | 1 |
| | ⋮ | ⋮ | ⋮ | ⋮ |
| | $x_m$ . | $X_m$ | $D(X_m)$ | 1 |
| Latent vector | $z_1$ | $G(z_1)$ | $D(G(z_1))$ | 0 |
| | $z_2$ | $G(z_2)$ | $D(G(z_2))$ | 0 |
| | ⋮ | ⋮ | ⋮ | ⋮ |
| | $z_m$ | $G(z_m)$ | $D(G(z_m))$ | 0 |

Define ① P (All correct) as the probability that Discriminator made correct judgement over all input data set (Real Images + generated data).

$$P(\text{All\_correct}) = P(\text{correct\_on\_Real\_image}) * P(\text{correct\_on\_generated\_image})$$

$$= \prod_{i=1}^{m} P(\text{correct\_on\_Real\_image}_i) * \prod_{i=1}^{m} P(\text{correct\_on\_generated\_image}_i \text{ is fake}).$$

$$= \prod_{i=1}^{m} P(x_i) * \prod_{i=1}^{m} \left(1 - D(G(z_i))\right)$$

Apply logarithm on both side of equation

$\log P(\text{All-correct})$

$= \sum_{i=1}^{m} \log P(x_i) + \sum_{i=1}^{m} \log(1 - D(G(z_i)))$.

$\Downarrow$

$\boxed{\frac{1}{m} \log P(\text{all-correct})} \leftarrow V(G)$.

$= \underbrace{\frac{1}{m} \sum_{i=1}^{m} \log P(x_i)}_{\text{the prediction accuracy about real images.}} + \underbrace{\frac{1}{m} \sum_{i=1}^{m} \log(1 - D(G(z_i)))}_{\text{the prediction Accuracy about generated image.}}$

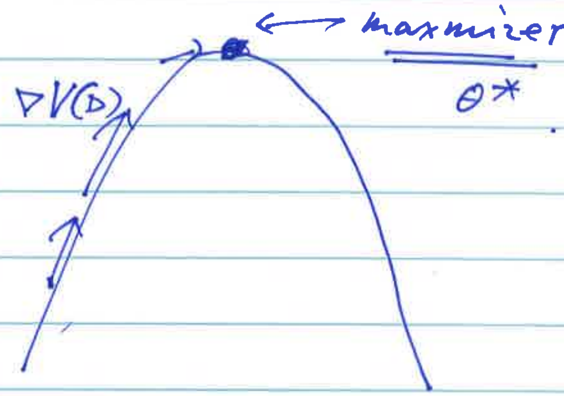
Define $\underbrace{V(D)}_{\text{objective function}} = \underbrace{\frac{1}{m} \log P(\text{all-correct})}_{\text{the discriminator's prediction Accuracy over whole data set}}$.

The discriminator is configured by

$$\underbrace{\max_{\theta} \; V(D)}$$ ($\theta$ is [weights, bias])

~~if we use gradient method~~.

Gradient Ascending Method is used




$$\nabla V(D) = \nabla_{\theta} \Big[ \frac{1}{m} \sum_{i=1}^{m} \log D(x_i) + \frac{1}{m} \sum_{i=1}^{m} \log(1 - D(G(z_i))) \Big]$$

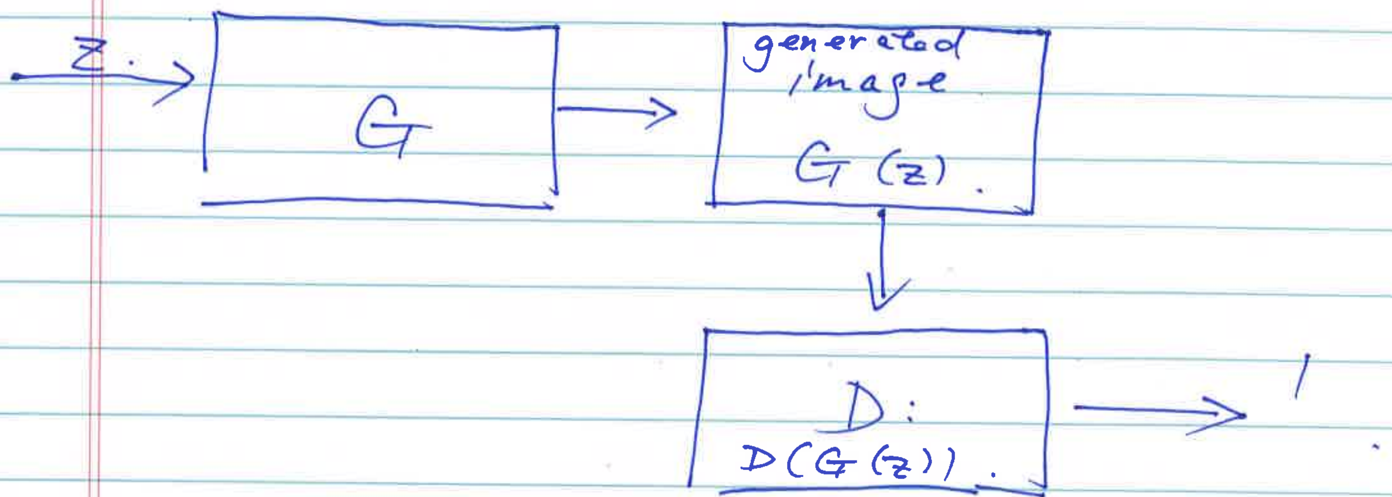
$$= \nabla_{\theta} \Big( \mathbb{E}_{x \sim P_{data}(x)} \log D(x) + \mathbb{E}_{z \sim P_z(z)} \log(1 - D(G(z))) \Big).$$

$\theta$ is updated via

$$\theta = \theta + \alpha \nabla V(D)$$

training of discriminator is completed!

## III. Training of Generator : G




| Input | Model's output | Label |
|---|---|---|
| $z_1$ | $D(G(z_1))$ | 1 |
| $z_2$ | $D(G(z_2))$ | 1 |
| $\vdots$ | $\vdots$ | $\vdots$ |
| $z_m$ | $D(G(z_m))$ | 1 |

what generator expected

Loss Function:

$$\mathcal{L}_\theta = \left\| -\begin{bmatrix} D(G(z_1)) \\ D(G(z_2)) \\ \vdots \\ D(G(z_m)) \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} \right\| = \left\| \begin{bmatrix} 1 - D(G(z_1)) \\ 1 - D(G(z_2)) \\ \vdots \\ 1 - D(G(z_m)) \end{bmatrix} \right\|.$$

To keep consistent with the Discriminator Objection function $V(D)$.

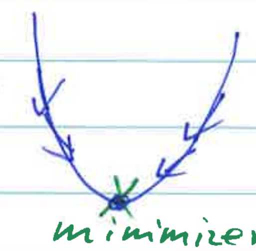
Apply logrithm on both side.

$$V(G) = \log \mathcal{L}_\theta = \left\| \begin{bmatrix} \log(1-D(G(z_1))) \\ \log(1-D(G(z_2))) \\ \vdots \\ \log(1-D(G(z_n))) \end{bmatrix} \right\|_1$$

$$= \underbrace{\sum_{i=1}^{m} \log(1-D(G(z_i)))}.$$

Generator is configured by

$$\min_\theta V(G).$$




using gradient decent.

$$\boxed{\theta = \theta - \alpha \cdot \nabla_\theta V(G)}.$$