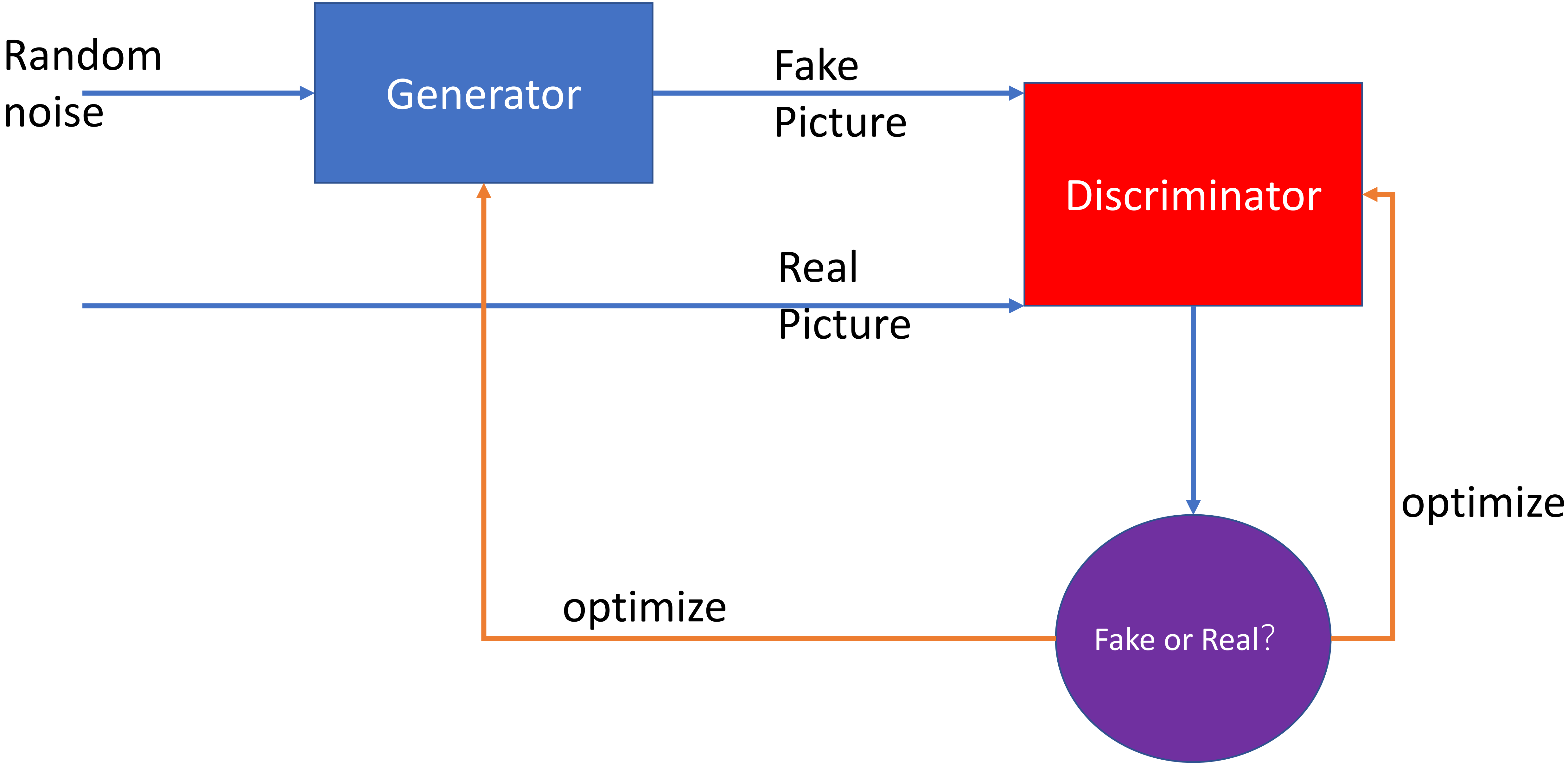
Generative Adversarial Nets

- Brief Explanation
- Thorough Explanation
- Mathematical Explanation


Random
noise
Generator
Fake
Picture
Discriminator
Real
Picture
optimize
optimize
Fake or Real?

Generator

How computer generates complicit random variable obeying a certain distribution?

Inverse Transform Method

$$CDF_X(x) = \mathbb{P}(X \leq x)$$

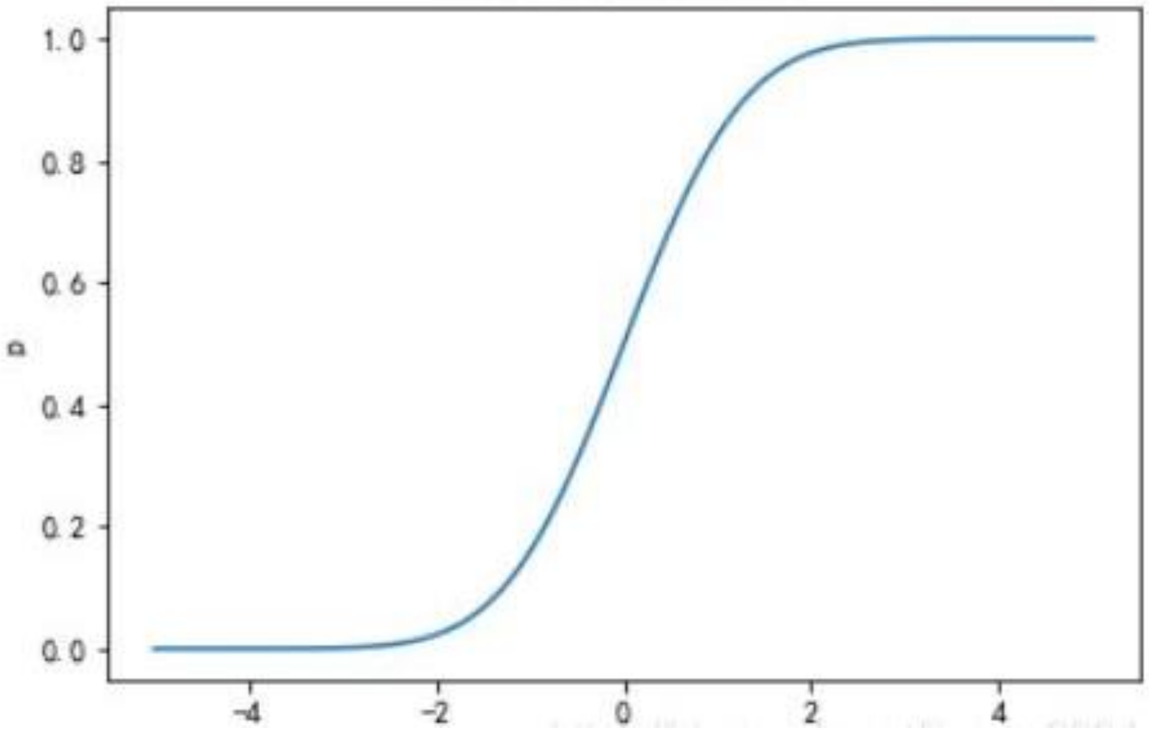


Figure out the probability density function $CDF_X(x)$

Generate a **(fake)** random variable u ~ U(0,1)

$CDF_X(x) =$ u

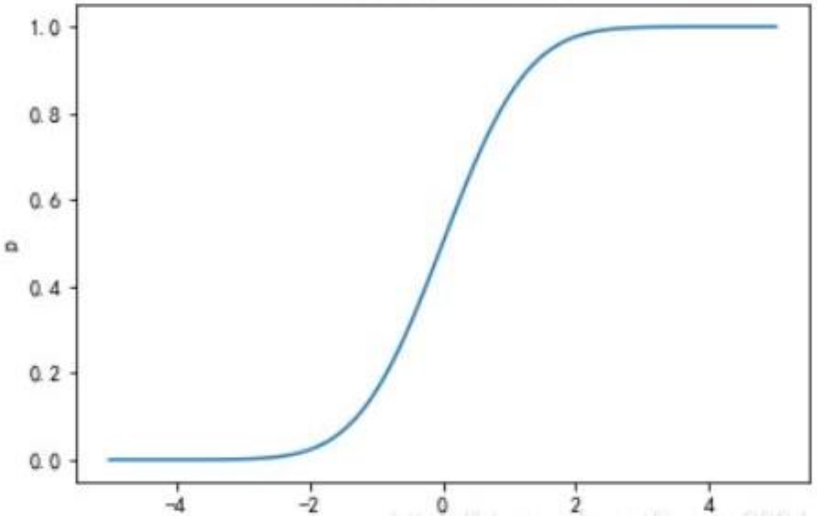
Solve x as the random value

x $= CDF_X^{-1}(U)$

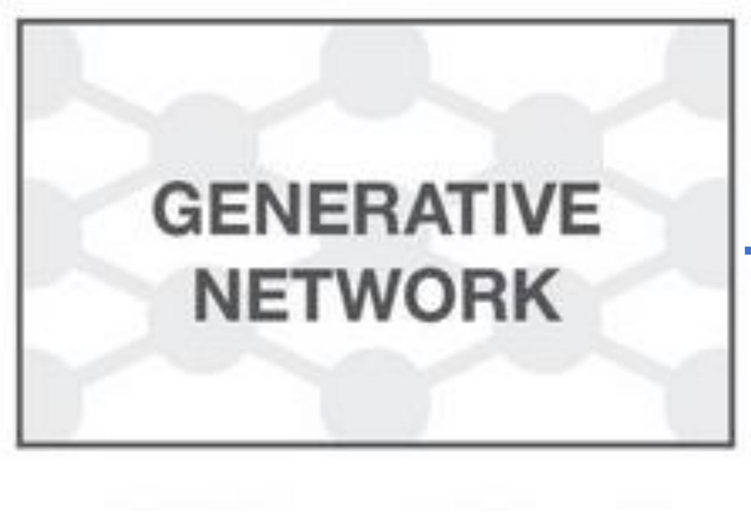
But what about the complicit distribution whose probably density function cannot be figured out?

Neural Network!

x $= CDF_X^{-1}(U)$ $\longrightarrow$ x = network(U)



Simple random variable

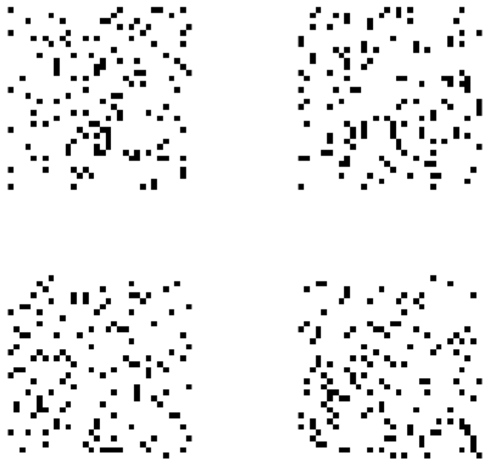


Complicit Random variable Obeying certain distribution

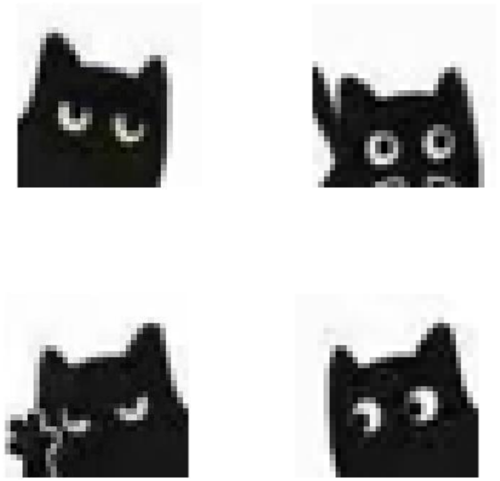
one pixel obeying Normal distribution(or any other distribution)

Two pixels obeying Two-dimensional Normal distribution

n pixels obeying n-dimensional normal distribution



32 * 32 pixels obeying
1024-dimensional
normal distribution



32 * 32 pixels obeying
Black cat distribution

Use neural network to fit(learn) the “black cat distribution”

Generate “black cat distribution” using simple distribution

(fake) random noise
obeying simple distribution

Fake picture obeying the
fitted “black cat distribution”

GAN-Core

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

Real picture

Fake picture

character	meaning
D( )	discriminator
$p_{data}(x)$	real picture data set
$p_z(z)$	distribution of simple (fake) random variable
X	real picture
Z	random noise
G(Z)	fake pictures generated on the random noise

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

Algorithm : training GAN nets

---

for e in epoch

sample ***m*** real pictures $\{x^{(1)}, x^{(2)}, \dots \dots x^{(m)}\}$

sample ***m*** random noise $\{z^{(1)}, z^{(2)}, \dots \dots z^{(m)}\}$

update discriminator by **gradient ascend**:

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D\left(\boldsymbol{x}^{(i)}\right) + \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right)\right]$$

sample ***m*** random noise $\{z^{(1)}, z^{(2)}, \dots \dots z^{(m)}\}$

update generator by **gradient descend**:

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right)$$

end

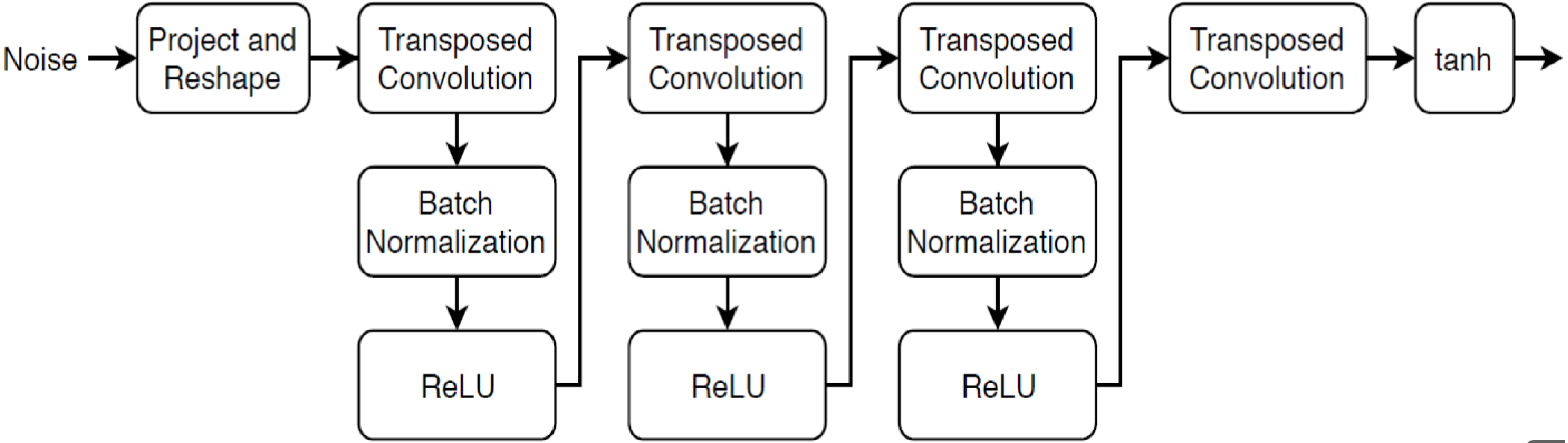
Why this equation?

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

Can reach global optimality of $p_g = p_{data}$

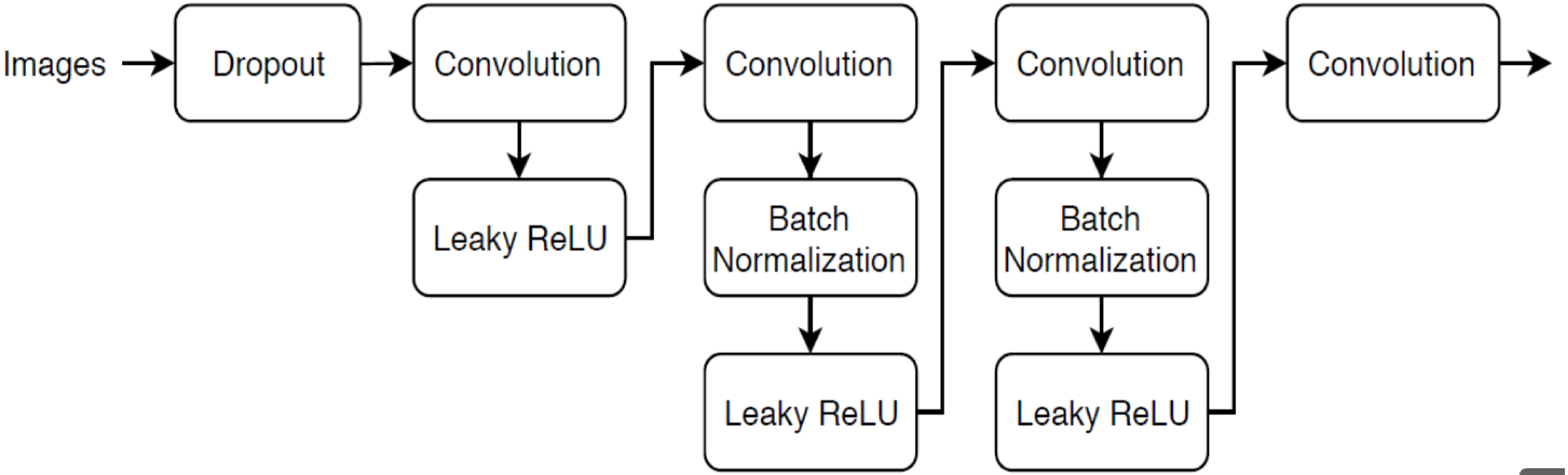
proof

Generator



Noise
Project and Reshape
Transposed Convolution
Batch Normalization
ReLU
Transposed Convolution
Batch Normalization
ReLU
Transposed Convolution
Batch Normalization
ReLU
Transposed Convolution
tanh

discriminator



Real Images

Fake Images

Real Images

Fake Images