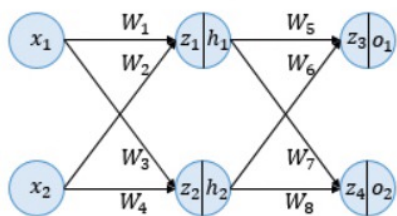


forward Backward



$$z_1 = x_1 W_1 + x_2 W_2$$

$$h_1 = \text{sig}(z_1)$$

⋮

가중치와 활성화 함수 (sig)로 다시 새로운 계층 전달한다.

Backward 는 전체 Error 를 줄이는 방향으로 W update 한다.

$$E_{\text{total}} = \frac{1}{2} (\text{tar}_1 - \text{out}_1)^2 + \frac{1}{2} (\text{tar}_2 - \text{out}_2)^2$$

$W_{5,6,7,8}$ 중 W_5 를 보자 (Chain Rule 사용)

$$\frac{\partial E_{\text{total}}}{\partial W_5} = \frac{\partial E_{\text{total}}}{\partial o_1} \times \frac{\partial o_1}{\partial z_3} \times \frac{\partial z_3}{\partial W_5} \quad \left(\begin{array}{l} \text{그 가중치의 결과,} \\ \text{sig (결과) 이용} \end{array} \right)$$

$$\frac{\partial E_{\text{total}}}{\partial o_1} = (\text{tar}_1 - \text{out}_1) \quad \frac{\partial o_1}{\partial z_3} = \text{out}_1 (1 - \text{out}_1)$$

(sig 미분 공식임!)

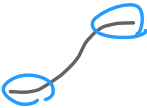
$$\frac{\partial z_3}{\partial W_5} = (h_1 W_5 + h_2 W_6)' = h_1$$

∴ $\frac{\partial E_{\text{total}}}{\partial W_5}$ 를 계산하면 $W_5' = W_5 - \alpha \frac{\partial E_{\text{total}}}{\partial W_5}$ 으로 업데이트 한다!

④ Activation Function (\Rightarrow Relu, Realy Relu 등등은 모르니 시도)

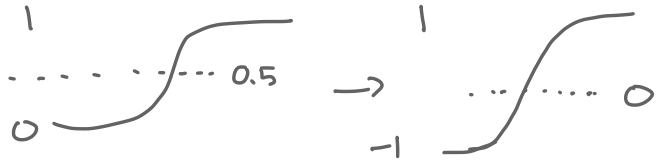
\Rightarrow linear 아니고 non linear 왜이유?

$w_3(w_2(w_1x+b) + c) + \dots$ 계속 해봤자 Layer 1개랑 똑같은

\Rightarrow Sigmoid 문제점? 

표범이 가릴지 0이 되며
가릴지 소실 (Gradient Vanishing) 현상
일어난다.

\Rightarrow hyperbolic tangent \Rightarrow Sigmoid 문제/점들 보았다



\Rightarrow 0이 중심이라 Grad Vanish 적지만 2번도 일어남

\Rightarrow ReLu 

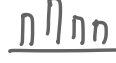

\Rightarrow 대략 반으로 간다면 하위만 함수에서 0
0이 되면 뉴런 작동 안함

\Rightarrow Leaky ReLu 

\Rightarrow 임의값 함수에도 가릴지 0이 안됨


\Rightarrow Soft Max 

Probability Distribution (확률 분포) and Generative Model

- discrete pm $P(X=1) = \frac{1}{2}$ (각사유와 같이 끊어지거나 계속 정해짐) 
- Continuous pro - (가우시안 분포와 같이 연속적인 분포) (pdf로 표현) 
- ↳ ex. 6로 표현

⇒ 키, 몸무게는 1차원 점이다. 이미지도 마찬가지로 $64 \times 64 \times 3$ 차원의 한 점 (64·64·3 픽셀 값이 있는)

으로 표현될 수 있고, 키 몸무게가 가지는 분포와 다른듯 사람 얼굴, 고양이 등도 다차원에서 한 점으로 특정 확률 분포를 가지고 있다 (multivariate probability distribution 다변량 확률 분포)

hidden layer 에서 차원이 2라고 하면  높은 차, 크의 값이 등과 같이 특정 분포를

확률 높은 변수값을 아니, 이것만 이용해서 data 생성

Generative Model

⇒ 분포를 학습해 실제와 같은 데이터를 생성 (이미지)

↳ Probability model

Discriminate



decision boundary
를 학습

Generative



data distribution
을 학습

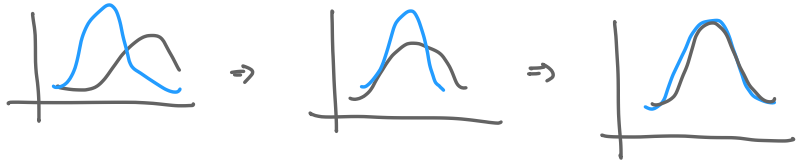
⇒ 이를 위해

Generative adversarial network

GAN 이 2014년 나왔

GAN

→ learning process



↳ learn distribution

Object Function

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

↖ 기대값 평균
↖ Noise를 뽐낼수 있는 dist

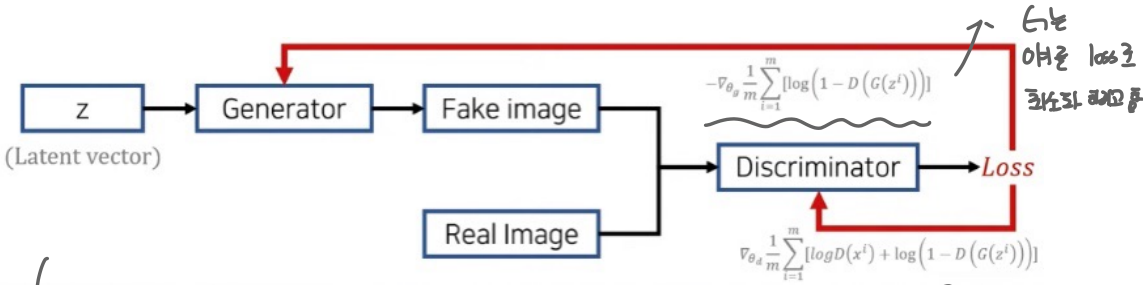
↓
 생성자는 $V(G, D)$ 낮추고자
 판별자는 $V(G, D)$ 높이려고 함

↳ $p_{data}(x)$ 는 원본 데이터 dist 이고 이중에서 2 하나를
 뽑아서 $D(x)$ 에 넣겠다

Latent vector

$G(z) =$ noise vector z 를 받아, 학습한 방법으로 new data 만들

$D(x) =$ data x 받아 얼마나 진짜인지 확률값 출력 (0 ~ 1)
real



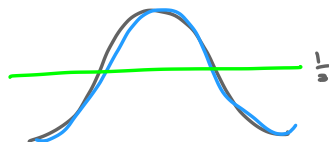
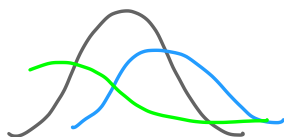
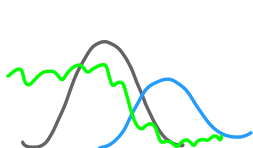
↳ 실제로는 둘다 동시에 만들고 mini batch로 다
 번갈아 가면서 학습함

↳ D는 아래 loss로 상용
 최대화하려고 최소화

• GAN 목표

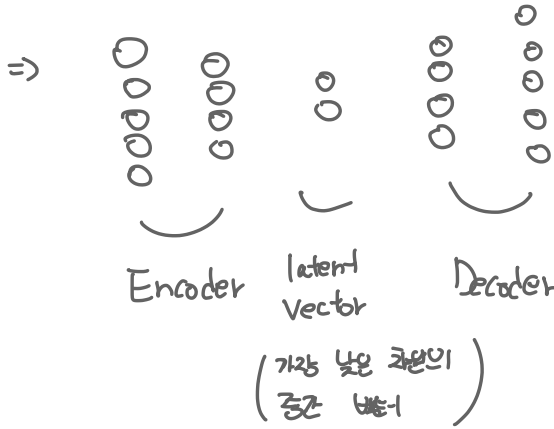
$$P_g(\text{생성자가 학습한 분포}) = P_{data}(\text{원본 데이터 분포})$$

$$D(G(z)) = \frac{1}{2} \quad (\text{생성자가 만든 것 더이상 구별 X})$$



- 원본 M
- 생성 M
- 판별 D

Auto Encoder



=> Input 이 25×25 이면
 1×625 로 해서 넣는다

Mnist data를 예시로 들면 Latent Vector를 확인해보면
숫자 1은 1개, 2는 2개, 3은 3개, 4는 4개, 5는 5개, 6은 6개, 7은 7개, 8은 8개, 9는 9개, 0은 10개

또한 Auto encoder는 Output을 확인해보면 input과 같이 비슷하게
결과를 확인할 수 있음 (조금 흐릿함)