

$$P(h|D) = \frac{\underset{\text{Likelihood}}{P(D|h)} \underset{\text{Prior}}{P(h)}}{\underset{\text{Evidence}}{P(D)}}$$

h를 주면 D가 나올 확률

$$P(h|D)P(D) = P(D|h)P(h) = P(h, D)$$

$$\hat{h} = \operatorname{argmax} P(h|D)$$

$$= \operatorname{argmax} \frac{P(D|h)P(h)}{P(D)}$$

$$= \operatorname{argmax} P(D|h)P(h) \Rightarrow \text{Bayesian} \text{은 정}$$

↗ Prior      ↗ Likelihood

prior와 likelihood를 통해 앞으로의 uncertainty까지 고려함

↳ 이를 통한 overfitting 등의 문제 해결

# Kullback-Leibler Divergence

두 블포(시그모이드)의 차를 측정

$KL(P||Q)$ 는  $KL(Q||P)$ 의 반

$$KL(P||Q)$$

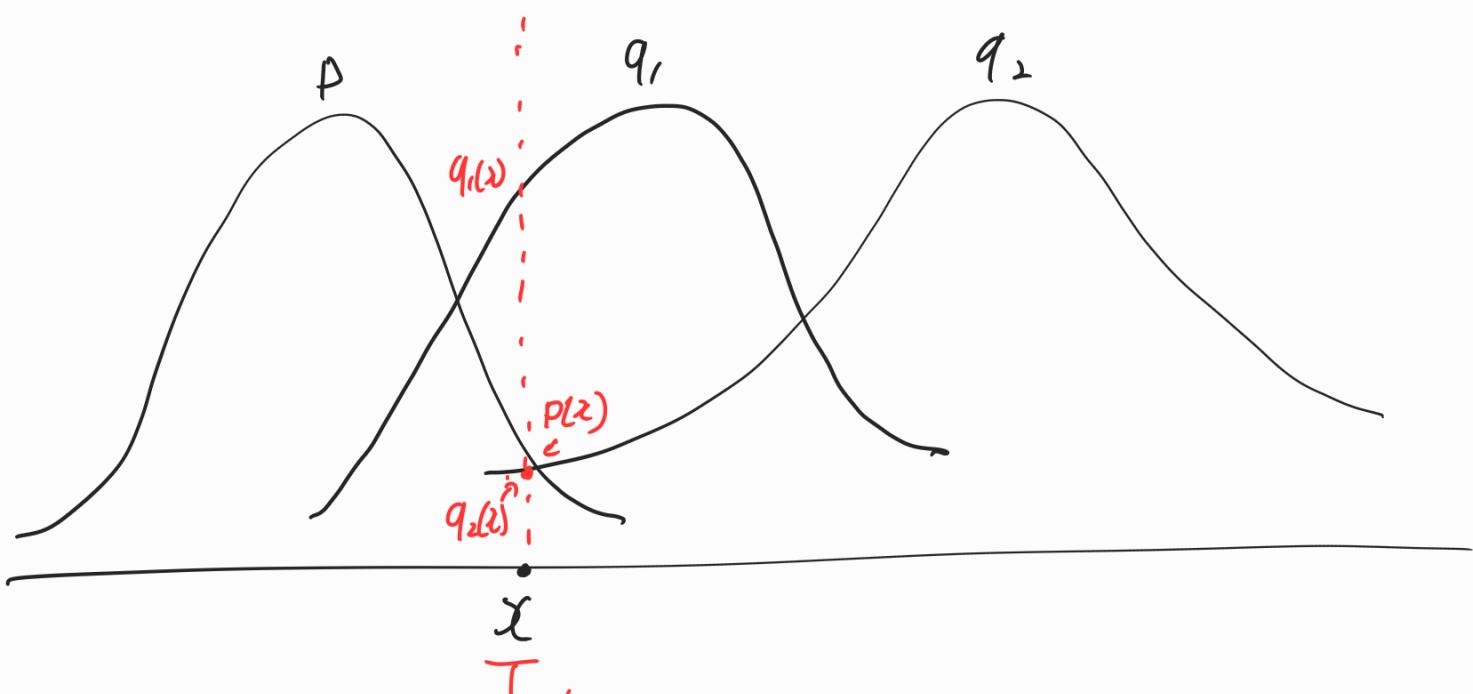
$\hookrightarrow P$ 라는 운포와  $Q$ 라는 운포의 차를 측정

$P$ 관점에서 측정

$P$ 에서 생기는 확률

Data 시그모이드

$$- \mathbb{E}_{x \sim p(x)} \left[ \log \frac{q(x)}{p(x)} \right]$$



$$\log \frac{q_1(x)}{p(x)} > \log \frac{q_2(x)}{p(x)}$$

$$-\log \frac{q_1(x)}{p(x)} < \boxed{-\log \frac{q_2(x)}{p(x)}}$$

더 크다 즉  $q_1$ 이  $q_2$ 보다  
P에 가깝다

KL()는 불포함 더 다를수록 절 가까울수록  
큰 값을 가진다

DL에서 사용

$$KL(P || P_{\theta})$$

↑

잘 모사할수록 값 0에 가까울수록 잘 초(적합)

$$L(\theta) = -\mathbb{E}_{x \sim p(x)} \left[ \mathbb{E}_{y \sim p(y|x)} \left[ \log \frac{p_{\theta}(y|x)}{p(y|x)} \right] \right]$$

↳  $p(x)$ 에 대해  
 x생성  
 ↳  $x$ 에 대해  $y$ 를  
 뽑아내는 확률에서  $y$ 생성

$$D = \{ (\overrightarrow{x_i}, \overrightarrow{y_i}) \}_{i=1}^N \quad N \in \mathbb{N}$$

$$L(\theta) \approx -\frac{1}{N \cdot k} \sum_{i=1}^N \sum_{j=1}^k \log \frac{P_\theta(y_{i,j}|x_i)}{P(y_{i,j}|x_i)}$$

$x_{\frac{1}{2}} N_{\frac{1}{2}} y_{\frac{1}{2}} k$   
설 풀기 해제를

$$\approx -\frac{1}{N} \sum_{i=1}^k \log \frac{P_\theta(y_i|x_i)}{P(y_i|x_i)}, \text{ if } k=1.$$


---

$$\hat{\theta} = \underset{\theta \in \Theta}{\operatorname{argmin}} L(\theta) \quad L(\theta) \text{ 는 } \bar{x} \text{ 를 } \hat{y} \text{로 예측하는 } \theta \text{ 찾기}$$

Gradient descent

# information

- 불확실성을 미지수로 하는 것

information  $\uparrow$  불확실성이-다

information  $\downarrow$  확실성이-다

$$I(x) = -\log p(x)$$

$p(x)$ 가 0이거나 1일 때

낮을수록  $\infty$ 에 가깝다

불확실성이-다 (확률이 낮다)

$$-\log \text{high}$$

확률이 높을수록 불확실성이-다

확률이 높을수록 정보량이 많다

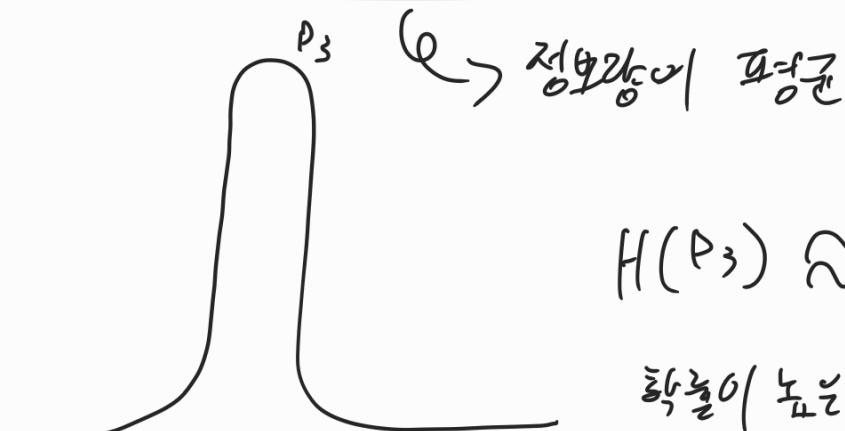
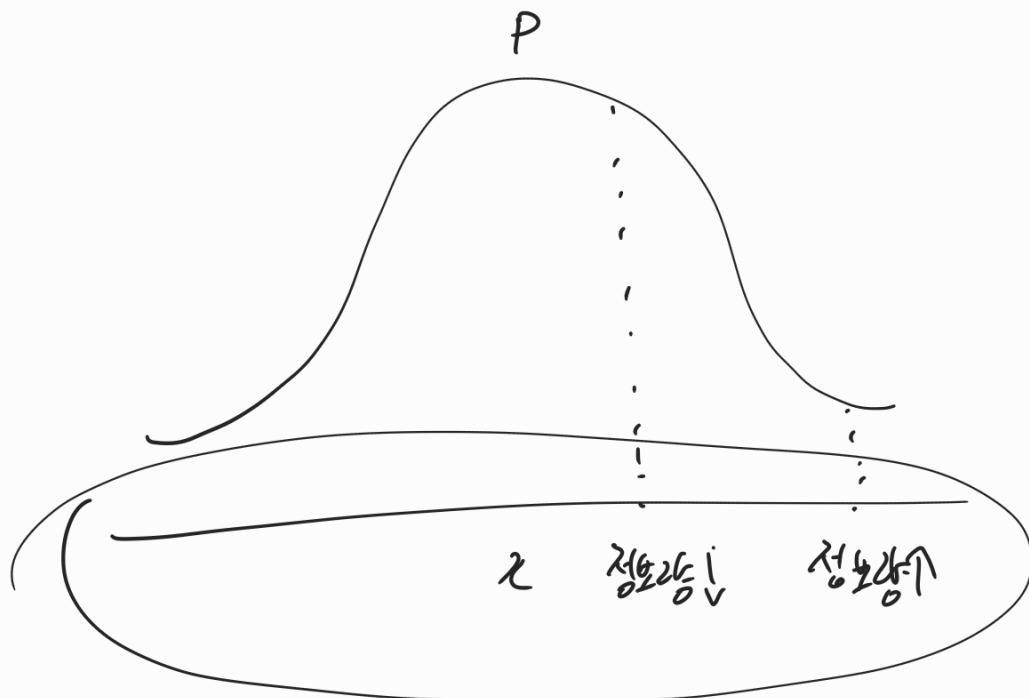
(확률이 높으면 너도 알고 나도 알고 정보량이 ↓)

$P(x)$ 가 0인 경우 0 (and 사실이 CT)

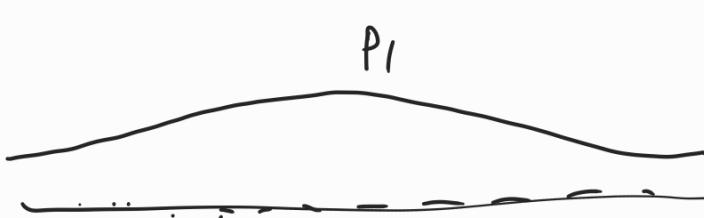
# Entropy

- 정보량의 기대값의 평균
- 불포화 평균적인 uncertainty
  - 불포화 형태를 예측

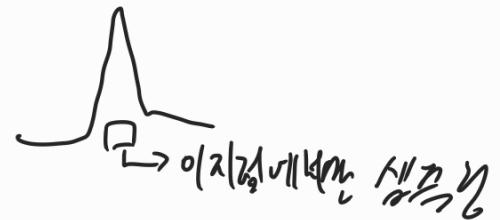
$$H(P) = -E_{x \sim P(x)} [\log P(x)]$$



$x \wedge x \dots \times \times$



Entropy가 클수록  $\text{flat}(\text{분포}(?))$  분포를 가지고  
낮을수록 편중한 분포를 갖진다



$P_3$ 는 불확실성이 낮다

↳ 어떤일이 일어날지 예상하다.

$P_1$ 은 불확실성이 높다

↳ 어떤일이 일어날지 모른다.

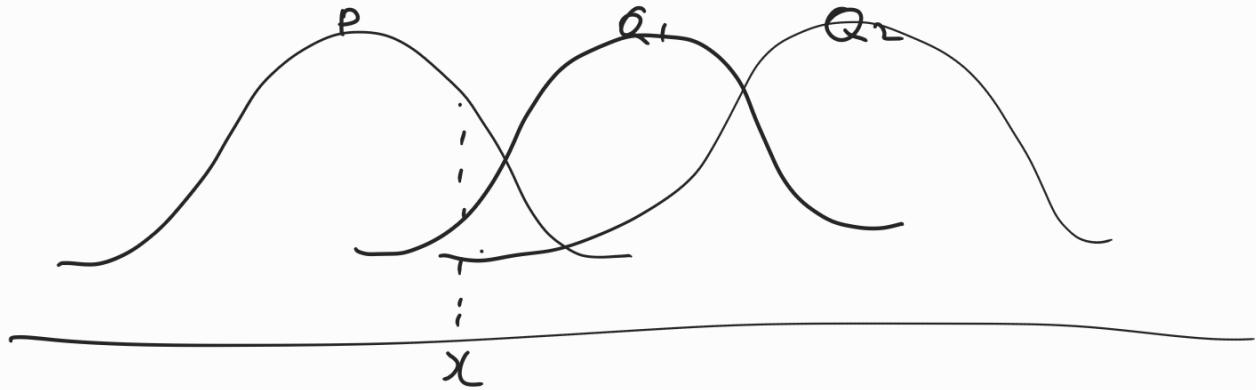
$$\text{불확실성} = \text{정보량(Entropy)} \uparrow = \text{확률} \downarrow$$

## Cross Entropy

- 분포  $P$ 의 관점에서 본 분포  $Q$ 의 정보량의 평균
- 비슷할수록 작은값

$$H(P, Q) = - \mathbb{E}_{x \sim P(x)} [\log Q(x)] = - \int P(x) \log Q(x) dx$$

$$\approx - \frac{1}{n} \sum_{i=1}^N \log Q(x_i)$$



$$Q_1(x) > Q_2(x)$$

$$\log Q_1(x) > \log Q_2(x)$$

$$-\log Q_1(x) < \log Q_1(x)$$

**Classification**에서 **Cross Entropy** 사용하여 최적화

$P_{\theta}$ 와  $P_{\theta}$ 를 비교하여 이를 최소화

ML의 정의, Entropy와 수식은?

$$-\frac{1}{N} \sum_{i=1}^N y_i \cdot \log y_i$$

$$KL(P||P_\theta) = -E_{x \sim P(x)} \left[ \log \frac{P_\theta(x)}{P(x)} \right]$$

$$= - \int P(x) \log \frac{P_\theta(x)}{P(x)} dx$$

$$\log \frac{y}{x} = \log y - \log x$$

$$= - \underbrace{\int P(x) \log P_\theta(x) dx}_{\text{crossentropy}} + \int P(x) \log P(x) dx$$

$$= H(P, P_\theta) - H(P)$$

$P_\theta$ 를 미분

$$\nabla_\theta KL(P||P_\theta) = \nabla_\theta (H(P, P_\theta)) - \nabla_\theta H(P)$$

$\overbrace{\text{Cross Entropy}}$

$\overbrace{\text{H}}^{\Theta \text{ 없는 간제한}} \text{ 상수 추적}$

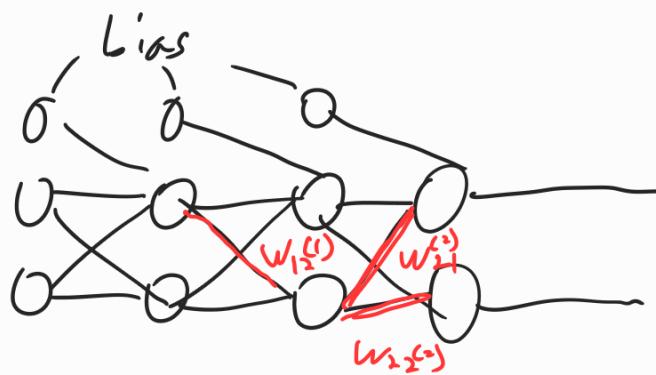
$\overbrace{\text{H}}^{\Theta \text{ 있는 간제한}} \text{ 미분 가능}$

$KL$ 과 cross entropy는 미분함수 같다.

$KL$ -divergence (or crossentropy)가 최소화로

Gradient descent 수행

# Back propagation



정답(1이면 정답)

$$x_1^{(3)} = g(s_1^{(2)}), s_1^{(2)} = w_{01}^{(2)} + w_{11}^{(2)}x_1^{(2)} + w_{21}^{(2)}x_2^{(2)}$$

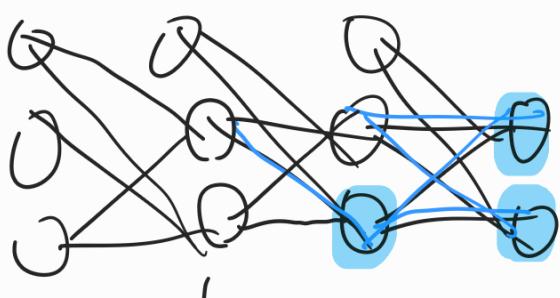
$$x_2^{(3)} = g(s_2^{(2)}), s_2^{(2)} = w_{02}^{(2)} + w_{12}^{(2)}x_1^{(2)} + w_{22}^{(2)}x_2^{(2)}$$

\* Chain rule

$$\frac{df(x)}{dx} = \frac{df(x)}{du} \cdot \frac{du}{dx}$$

$$\frac{\partial f}{\partial w_{11}^{(2)}} = \frac{\partial f}{\partial x_1^{(3)}} \cdot \frac{\partial x_1^{(3)}}{\partial s_1^{(2)}} \cdot \frac{\partial s_1^{(2)}}{\partial w_{21}^{(2)}}$$

$$\frac{\partial f}{\partial w_{22}^{(2)}} = \frac{\partial f}{\partial x_2^{(3)}} \cdot \frac{\partial x_2^{(3)}}{\partial s_2^{(2)}} \cdot \frac{\partial s_2^{(2)}}{\partial w_{22}^{(2)}}$$



→ 예전에 배운 딥러닝의 차이점은 뒤에서 다룬다

$$\frac{\partial f}{\partial w_{12}^{(1)}} = \left[ \frac{\partial f}{\partial X_1^{(3)}} \cdot \frac{\partial X_1^{(3)}}{\partial S_1^{(2)}} \right] \cdot \left[ \frac{\partial S_1^{(2)}}{\partial X_2^{(1)}} \cdot \frac{\partial X_2^{(1)}}{\partial w_{12}^{(1)}} \right]$$

지정한  $S_1^{(2)}$

$$+ \left[ \frac{\partial f}{\partial X_2^{(3)}} \cdot \frac{\partial X_2^{(3)}}{\partial S_2^{(2)}} \right] \cdot \left[ \frac{\partial S_2^{(2)}}{\partial X_1^{(1)}} \cdot \frac{\partial X_1^{(1)}}{\partial w_{12}^{(1)}} \right]$$

$\overbrace{S_2^{(2)}}$       공통

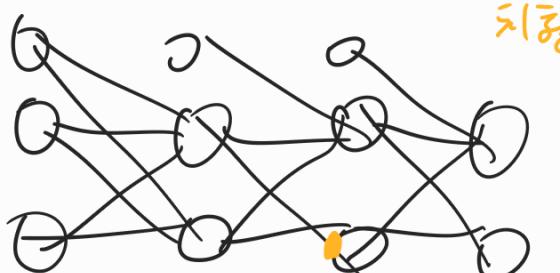
$$= [S_1^{(2)}, S_2^{(2)}] \begin{bmatrix} w_{21}^{(2)} \\ w_{22}^{(2)} \end{bmatrix} \frac{\partial X_2^{(2)}}{\partial S_2^{(1)}} X_1^{(1)}$$

$\vec{s}^{(2)}$        $\vec{w}^{(2)}$

$$= [S_2^{(2)\top}, w_2^{(2)} \cdot \frac{\partial X_2^{(2)}}{\partial S_2^{(1)}}] \cdot X_1^{(1)}$$

$$\epsilon \frac{\partial f}{\partial S_2^{(1)}} = S_2^{(1)}$$

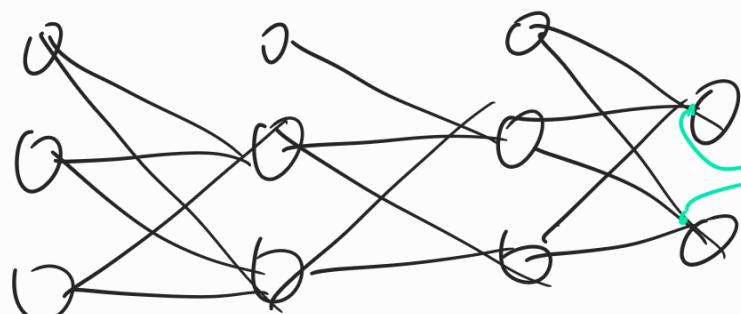
지정한



$$\frac{\partial f}{\partial w_{ij}^{(l)}} = \delta_j^{(l)} c_i^{(l)}$$

$$\delta_j^{(l+1)} \leftarrow \frac{\partial f}{\partial S_j^{(l)}}$$

이걸 구하기 위해 전 노드를 계산해 가고



최종에는

$$\delta_j^{(l)} \text{가 표시되는 } l$$

거꾸로 계산해나간다.