$P_\theta(x) = p(x;\theta) = p(x|\theta)$

확률분포 θ에서 x라는 값이 가지는 확률

$P_\theta(y|x) = p(y|x;\theta) = p(y|x,\theta)$

확률분포 θ가 있을 때 x가 주어졌을 때 y의 확률값

MLE

$\theta = \{w_1, b_1, w_2, b_2 \cdots\}$

Gradient Ascent를 통해

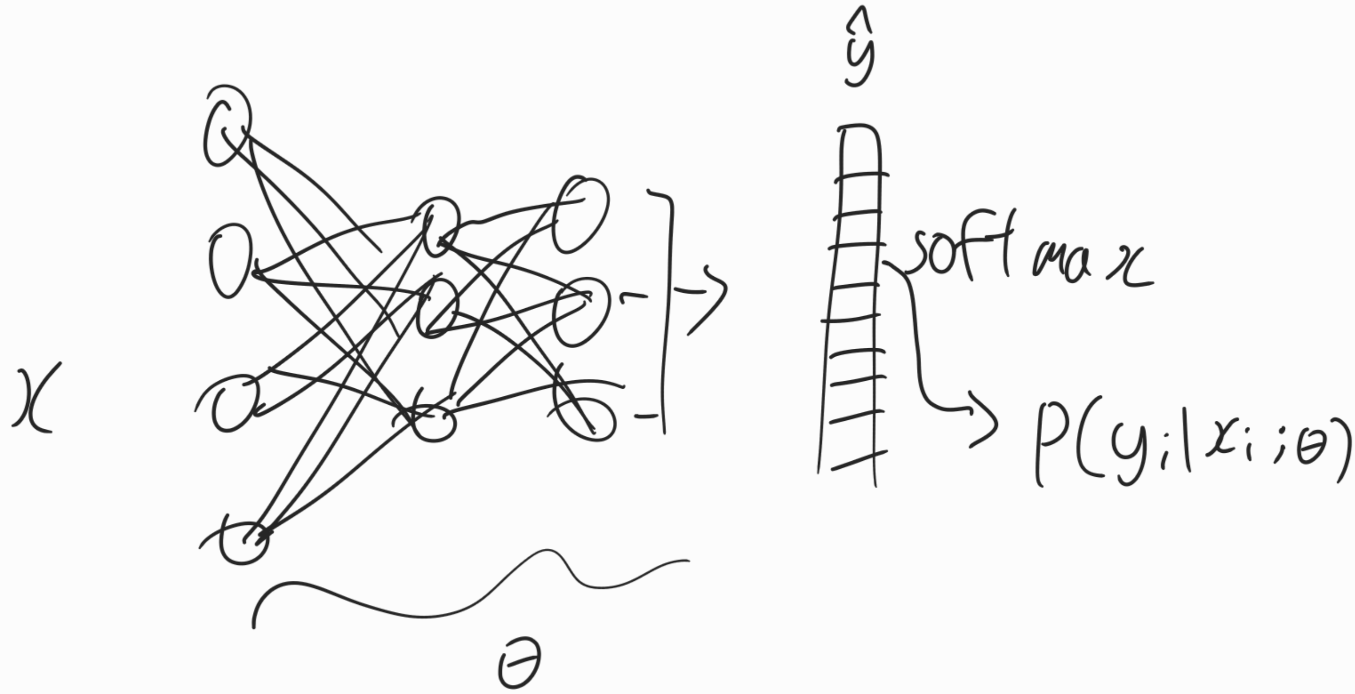
도출

하지만 대부분 딥러닝 프레임워크는 Gradient decent만 지원

No problem -1 곱하면 됨

따라서 Maximization에서 minimization 문제로접근

Negative log Likelihood (NLL)



$$-\sum_{i=1}^{N} \log P(y_i|x_i;\theta) = -\sum_{i=1}^{n} y_i^T \cdot \log \hat{y}_i$$

cross entropy Loss

자 MLE

$P(y_i|x_i;\theta)$

확률 분포 θ (DL에서는 weight) 에서 X가 주어졌을 때 Y의 확률

한번의 시행에서 사건 A가 일어날 확률을 p라 할 때, 이 시행을 n회 반복한 독립시행에서 사건 A가 r번 일어날 확률은

MLE $L(\theta) =$ ${}_nC_r p^r (1-p)^{n-r}$ $\left(\prod_{i=1}^{N} P(y_i|x_i;\theta)\right)$

해당 관찰 결과가 발생할 확률

여기에 X을 +로 바꾸기 위해 Data underflow를 방지하기 위해

$\log L(\theta)$ 로 Maximum likelyhood로 변환

$\sum_{i=1}^{N} \log P(y_i|x_i;\theta)$

(이 식을 최대로 만드는 θ가 가능도함수 $L(\theta)$도 최대로 만듦)

MLE의 목적 $L(\theta)$가 가장 큰 곳의 θ 즉 $L(\theta)$를 미분했을 때 값이 0이 되는 θ를 찾는것

$$\hat{\theta} = \underset{\theta \in \Theta}{\arg\max} \sum_{i=1}^{N} \log P(y_i|x_i;\theta)$$

그런데 이 식은 Gradient Ascent를 통해 가중치 조절

⇓

$$\hat{\theta} = \underset{\theta \in \Theta}{\arg\min} - \sum_{i=1}^{n} \log P(y_i|x_i;\theta)$$ 로 변환

→ Gradient descent

Model 의 결과물 $\hat{y}$

X가 파라미터 θ를 거쳐서 나온 출력 $\hat{y}$

→ 확률

$P(y_i|x_i;\theta)$

→ 확률 분포 θ (DL에서는 weight) 에서 X가 주어졌을 때 Y의 확률

$$\hat{y}_i = P(y_i|x_i;\theta)$$

$$-\sum_{i=1}^{N} \log P(y_i|x_i;\theta) = -\sum_{i=1}^{N} \log \hat{y}_i$$

자 여기서 softmax

→ 출력 각 클래스 별로 확률 제공

→ 입력 $\begin{bmatrix}0\\0\\0\\1\end{bmatrix}$ one-hot vector

$$\hat{y}_i = \underbrace{y_i^T}_{\text{i번째 1 나머지 0 해당 index만 생존}} \times \hat{y}_i$$

$$-\sum_{i=1}^{N} \log P(y_i | x_i ; \theta) = \underbrace{-\sum_{i=1}^{N} y^T_i \cdot \log \hat{y}_i}_{\text{cross entropy loss}}$$

평균내는건 상수