[Cross-Entropy Function : Softmax Regression's loss function]

ex)

softmax → $\hat{y} = a = \begin{bmatrix} 0.1 \\ 0.2 \\ 0.3 \\ 0.4 \end{bmatrix}$

$$loss(a, y) = -\sum_{j=1}^{4} y_j \log(\hat{y}_j)$$

$y = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$ → target (one-hot vectored)

* general form

y = 실제값 (target)

K = 클래스 개수

y_j = 실제값 원-핫 벡터의 j번째 인덱스

$a_j = \hat{y}_j$ = 샘플 데이터가 j번째 클래스일 확률

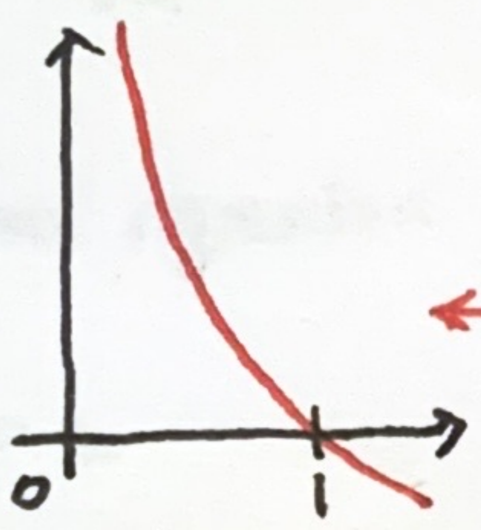
[cross-entropy function]

$$loss(a, y) = -\sum_{j=1}^{K} y_j \log(a_j)$$

$$\text{cost function} = -\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{K} y_j \log(\hat{y}_j)$$

* cross-entropy 함수가 softmax regression의 loss function으로서 적합한 이유

$-\log(z)$



$$-\sum_{j=1}^{K} y_j \cdot \log(\hat{y}_j) = \sum_{j=1}^{K} \left[(y_j) \cdot (-\log(\hat{y}_j))\right]$$

i)
- $g(z) = \text{softmax}(z) = \hat{y} = \begin{bmatrix} 0.1 \\ 0.2 \\ 0.3 \\ 0.4 \end{bmatrix}$
- target $y = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$
- $loss(\hat{y}, y) = -(0 \cdot \log 0.1 + 0 \cdot \log 0.2 + 0 \cdot \log 0.3 + 1 \cdot \log 0.4)$

  $= -\log 0.4 \approx \underline{0.39}$ ⇒ loss 값 작음

더 직관적인 예시

$y = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$

① $\hat{y} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ → 올바르게 예측

$loss(\hat{y}, y) = -(1 \cdot \log 1 + 0 \cdot \log 0) = -\log 1 = \underline{0}$

② $\hat{y} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$ → 완전히 틀리게 예측

$loss(\hat{y}, y) = -(1 \cdot \log 0 + 0 \cdot \log 1) = -\log 0 = \underline{\infty}$

- $g(z) = \text{softmax}(z) = \hat{y} = \begin{bmatrix} 0.1 \\ 0.2 \\ 0.3 \\ 0.4 \end{bmatrix}$
- target $y = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$
- $loss(\hat{y}, y) = -(1 \cdot \log 0.1 + 0 \cdot \log 0.2 + 0 \cdot \log 0.3 + 0 \cdot \log 0.4)$

  $= -\log 0.1 = \underline{1}$ ⇒ loss 값 큼