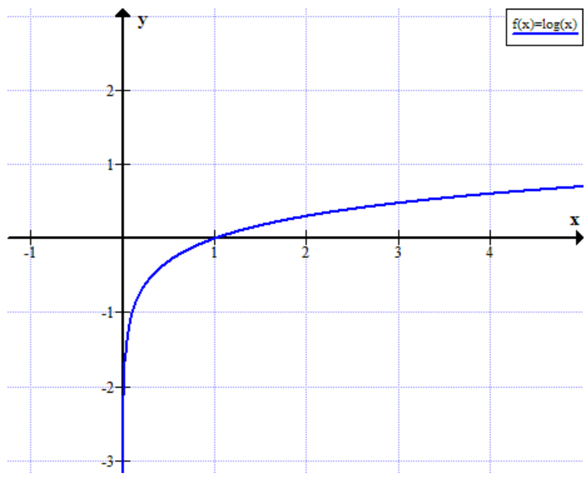
Cross Entropy Loss, Softmax Loss, NT-Xent Loss

Writer: Chaeeun Ryu

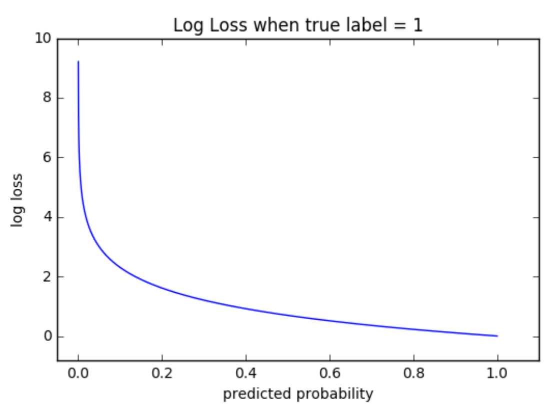
Cross-entropy loss (i.e., *log loss*) measures the performance of a classification model whose output is a probability value between 0 and 1. Cross-entropy loss increases as the predicted probability diverges from the actual label.



log 그래프: 1 이하에선 음수이다.

$$\text{Loss} = -\sum_{i=1}^{\substack{\text{output}\\\text{size}}} y_i \cdot \log \hat{y}_i$$

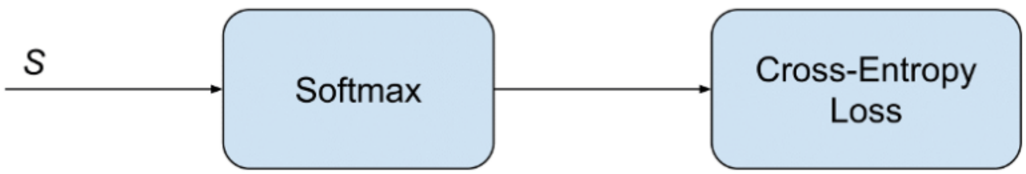
Implication: Loss(y_i가 1이고 y_hat_i가 0.7일 때) < Loss(y_i가 1이고 y_hat_i가 0.2일 때)



Softmax Loss

Softmax Loss (i.e., *categorical cross-entropy loss*): a Softmax Activation + Cross-Entropy Loss

- Softmax: activation function that outputs the probability for each class and these probabilities will sum up to one.
- Cross Entropy loss: just the sum of the negative logarithm of the probabilities.



$$f(s)_i = \frac{e^{s_i}}{\sum_j^C e^{s_j}} \quad CE = -\sum_i^C t_i log(f(s)_i)$$

In the specific (and usual) case of Multi-Class classification the labels are **one-hot, so only the positive class C_p keeps its term in the loss**. There is only one element of the Target vector t which is not zero t_i=t_p. So discarding the elements of the summation which are zero due to target labels, we can write:

$$CE = -log\left(\frac{e^{s_p}}{\sum_j^C e^{s_j}}\right)$$

Softmax loss = $-log(\frac{exp(S_p)}{\sum_j^C exp(S_j)})$

S_p: CNN score for the positive class (softmax) ex.) score for the image being dog

Supervised NT-Xent loss (Khosla et al. 2020)

the normalized temperature-scaled cross entropy loss

It is a modification of the multi-class N-pair loss with addition of the **temperature parameter (τ) to scale the cosine similarities**:

$$\mathcal{L}(\mathbf{z_i}, \mathbf{z_j}) = -\log \frac{\exp(\mathbf{z_i} \mathbf{z_j} / \tau)}{\sum_{k=1}^{2N} \mathbb{1}_{k \neq i} \exp(\mathbf{z_i} \mathbf{z_k} / \tau)}$$

Self-supervised NT-Xent loss

An appropriate temperature parameter can **help the model learn from hard negatives**. In addition, they showed that the **optimal temperature differs on different batch sizes and number of training epochs. (hard negatives: 실제로는 negative 인데 positive 라고 잘못 예측하기 쉬운 데이터,** hard negative는 마치 positive처럼 생겨서 예측하기 어려움**)**

$$\mathcal{L}(\mathbf{z_i}, \mathbf{z_j}) = \frac{-1}{2N_{y_i} - 1} \sum_{j=1}^{2N} \log \frac{\exp(\mathbf{z_i} \mathbf{z_j} / \tau)}{\sum_{k=1}^{2N} \mathbb{1}_{k \neq i} \exp(\mathbf{z_i} \mathbf{z_k} / \tau)}$$

Supervised NT-Xent loss

참고:

https://ml-cheatsheet.readthedocs.io/en/latest/loss_functions.html#:~:text=Cross-entropy loss%2C or log,diverges from the actual label.

https://jamesmccaffrey.wordpress.com/2022/04/11/an-example-of-normalized-temperature-scaled-cross-entropy-loss/

https://blog.naver.com/PostView.nhn?isHttpsRedirect=true&blogId=sogangori&logNo=221073537958&parentCategoryNo=&categoryNo=6&viewDate=&isShowPopularPos

https://towardsdatascience.com/additive-margin-softmax-loss-am-softmax-912e11ce1c6b#:~:text=In short%2C Softmax Loss is,negative logarithm of the probabilities. (설명 엄청 잘 되어있음)

https://gombru.github.io/2018/05/23/cross_entropy_loss/