

Lab-06 Softmax Classification

$$P(\text{class} = i) = \frac{e^i}{\sum e^i}$$

$z = \text{torch.FloatTensor}([1, 2, 3])$

hypothesis = F. Softmax(z , dim=0) 각각 값에 대한 확률 얻어줌.

Cross Entropy : 두 개의 확률 분포가 얼마나 다른지를 나타내는 값

$$H(P, Q) = -E_{x \sim p(x)}(\log Q(x)) = -\sum_x p(x) \log Q(x)$$

Cross Entropy Loss (Low-Level): $L = \frac{1}{N} \sum_{i=1}^N - \log(\hat{y}_i)$

One-hot encoding: $y_one_hot = torch.zeros(hypothesis)$

$$y_one_hot.scatter_cl, y.unsqueeze(1,1)$$
$$\text{Cost} = (y_{\text{one hot}} * -\text{torch.log}(\text{hypothesis}))$$

$$\text{Sum}(\text{dim}=1). \text{mean}()$$

- Low-level

$$(y_one_hot * -\text{torch.log}(F.\text{Softmax}(z, \text{dim}=1))).\text{sum}(\text{dim}=1).\text{mean}()$$

High-level Negative Log Likelihood

$$F.\text{nn_loss}(F.\text{log_softmax}(z, \text{dim}=1), y)$$
$$F_{\text{cross_entropy}}(z, y)$$

```
model = SoftmaxClassifierModel()
```