

Week02 발표

Softmax Classification

이경선

Softmax classification

Softmax

Cross Entropy

Low-level Implementation

High-level Implementation

Training Example

Softmax classification

Binary
classification

Logistic regression

Sigmoid function

$$y = \frac{1}{1 + e^{-z}}$$
$$= \frac{1}{1 + e^{-(\mathbf{w}^T \mathbf{x} + b)}}$$

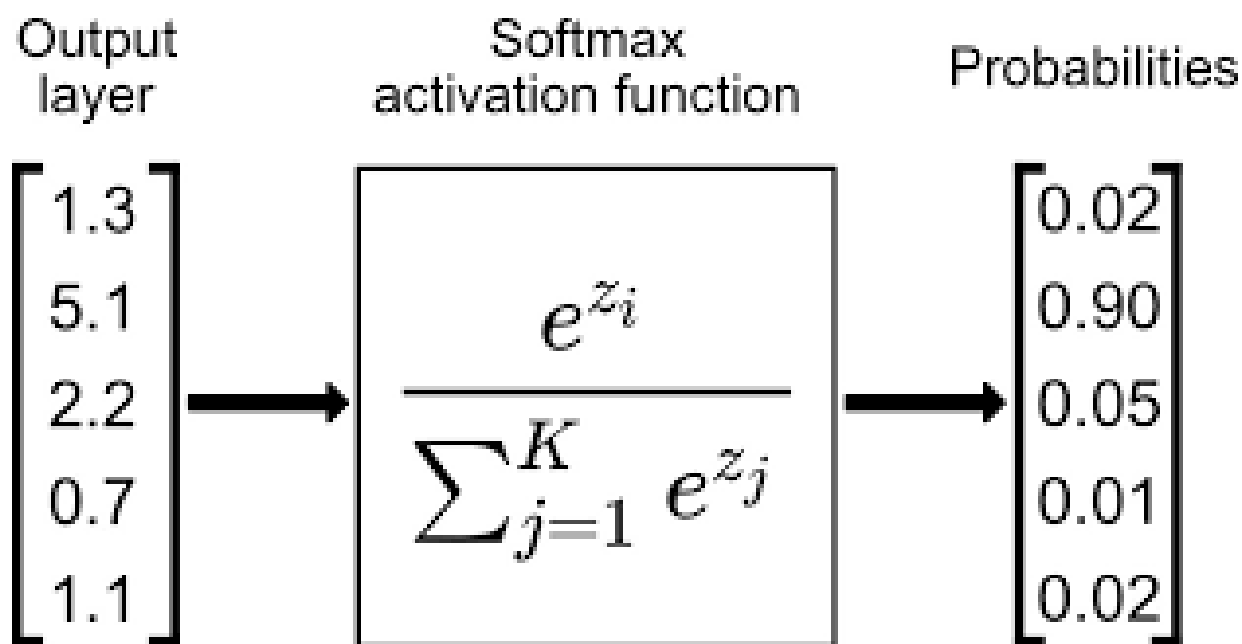
Multi-class
classification

Softmax regression

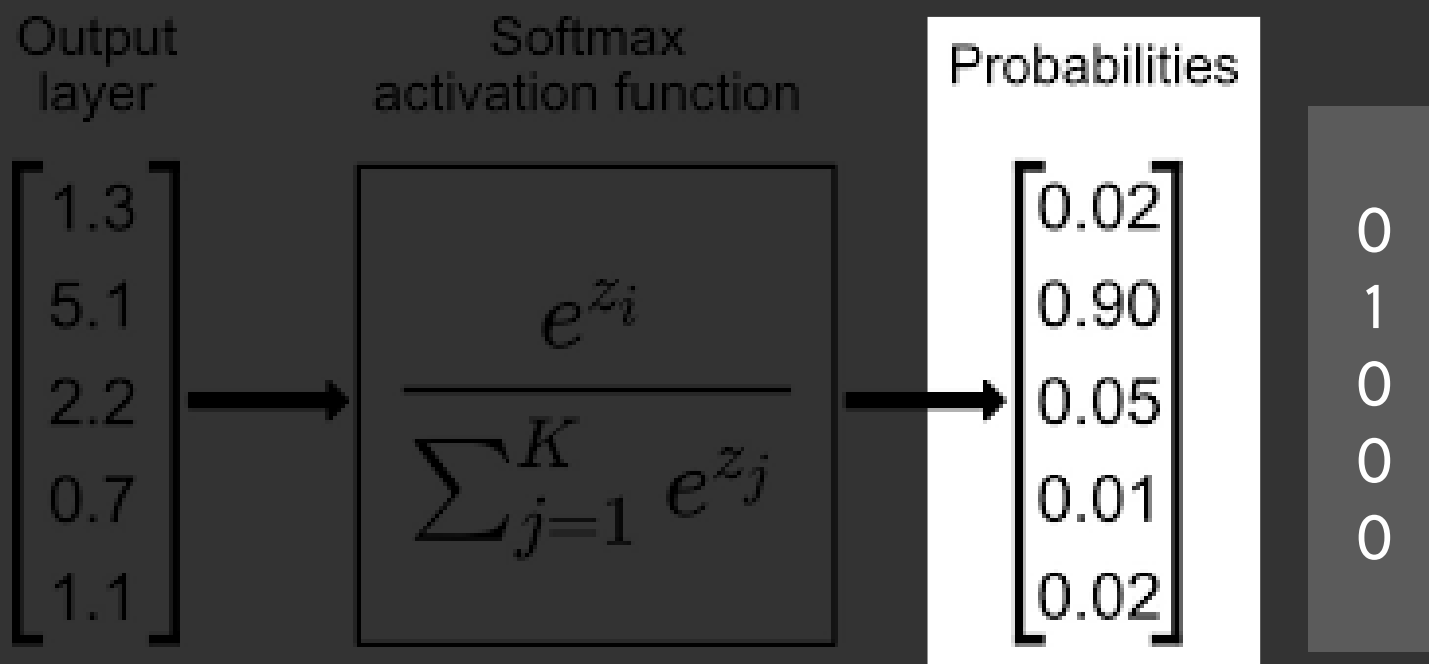
Softmax function

$$\phi(x_k) = \frac{\exp(x_k)}{\sum_{i=1}^n \exp(x_i)}$$

Softmax classification



Softmax classification



A,B,C,D,E로 분류
행렬의 원소는 각 알파벳일 확률

Softmax

모두를 위한 머신러닝/딥러닝 강의

모두를 위한 머신러닝과 딥러닝의 강의

알파고와 이세돌의 경기를 보면서 이제 머신 러닝이 인간이 잘 한다고 여겨진 직관과 의사 결정능력에서도 충분한 데이터가 있으면 어느정도 또는 우리보다 더 잘할수도 있다는 생각을 많이 하게 되었습니다. Andrew Ng 교수님이 말씀하신것 처럼 이런 시대에 머신 러닝을 잘 이해하고 잘 다룰수 있다면 그야말로 "Super Power"를 가지게 되는 것이 아닌가 생각합니다.

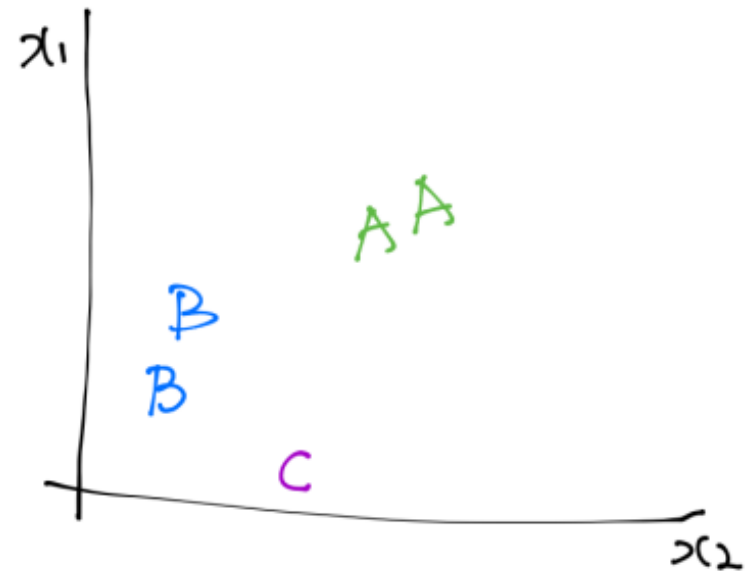
더 많은 분들이 머신 러닝과 딥러닝에 대해 더 이해하고 본인들의 문제를 이 멋진 도구를 이용해서 풀수 있게 하기위해 비디오 강의를 준비하였습니다. 더 나아가 이론에만 그치지 않고 최근 구글이 공개한 머신러닝을 위한 오픈소스인 TensorFlow를 이용해서 이론을 구현해 볼수 있도록 하였습니다.

수학이나 컴퓨터 공학적인 지식이 없이도 쉽게 볼수 있도록 만들려고 노력하였습니다.

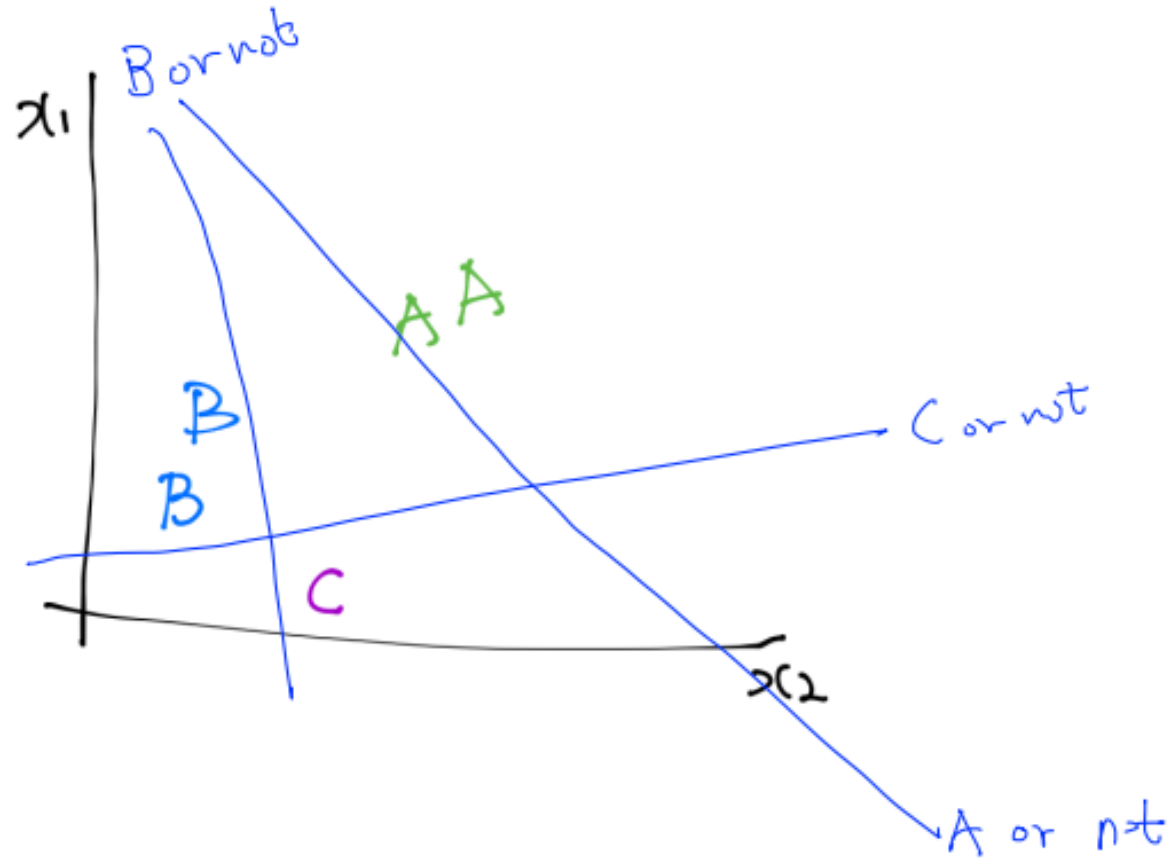
모두를 위한 머신러닝/딥러닝 강의
(hunkim.github.io)

Multinomial classification

x1 (hours)	x2 (attendance)	y (grade)
10	5	A
9	5	A
3	2	B
2	4	B
11	1	C



Multinomial classification



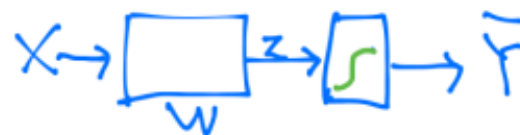
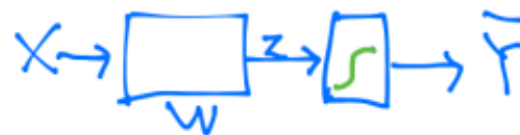
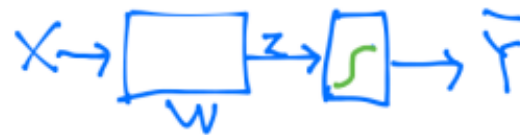
3개의 Binary classification

Multinomial classification

$$\begin{bmatrix} w_1 & w_2 & w_3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = [w_1 x_1 + w_2 x_2 + w_3 x_3]$$

$$\begin{bmatrix} w_1 & w_2 & w_3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = [w_1 x_1 + w_2 x_2 + w_3 x_3]$$

$$\begin{bmatrix} w_1 & w_2 & w_3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = [w_1 x_1 + w_2 x_2 + w_3 x_3]$$



Multinomial classification

$$\begin{bmatrix} w_1 & w_2 & w_3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = [w_1 x_1 + w_2 x_2 + w_3 x_3]$$

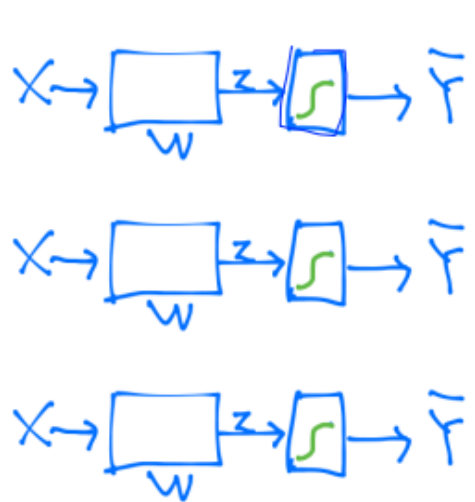


$$\begin{bmatrix} w_{A1} & w_{A2} & w_{A3} \\ w_{B1} & w_{B2} & w_{B3} \\ w_{C1} & w_{C2} & w_{C3} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} =$$

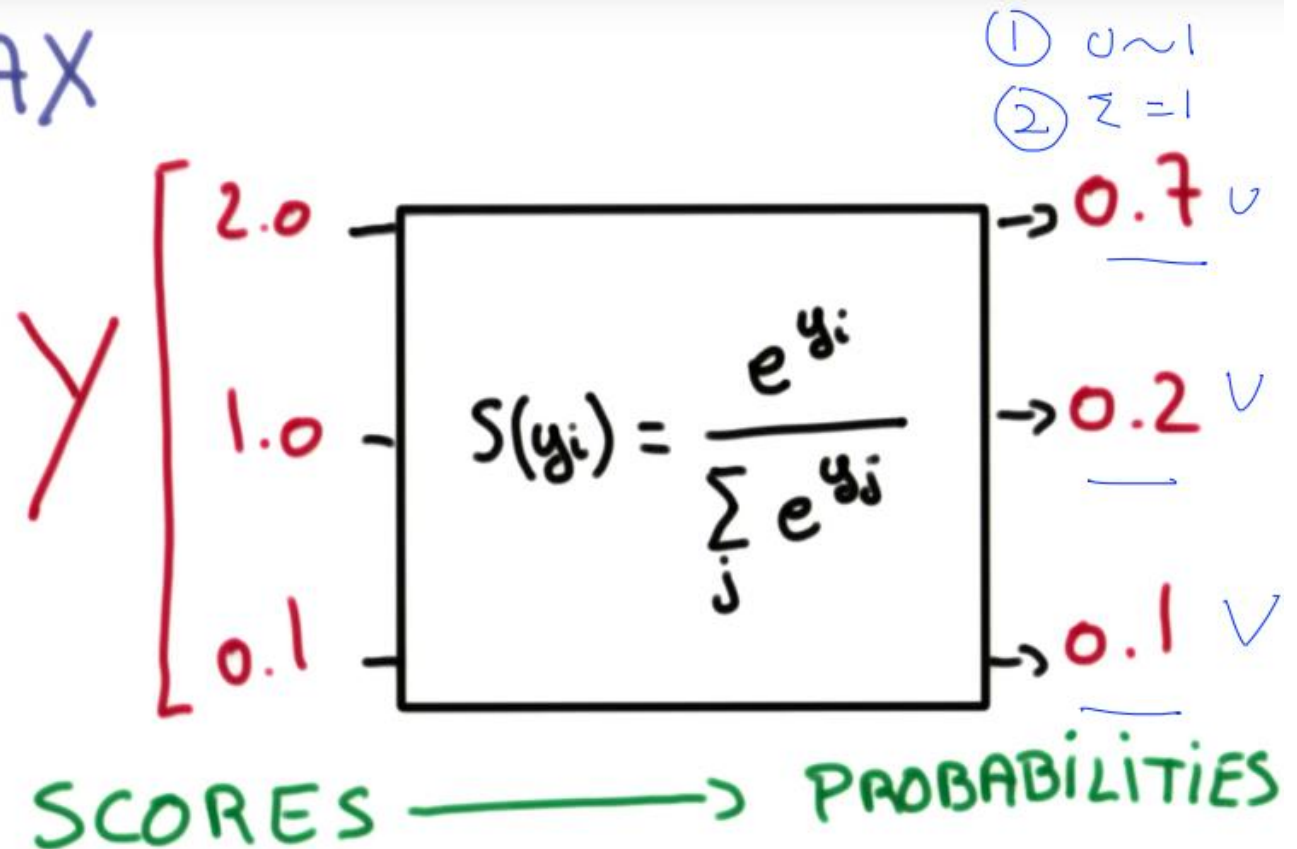


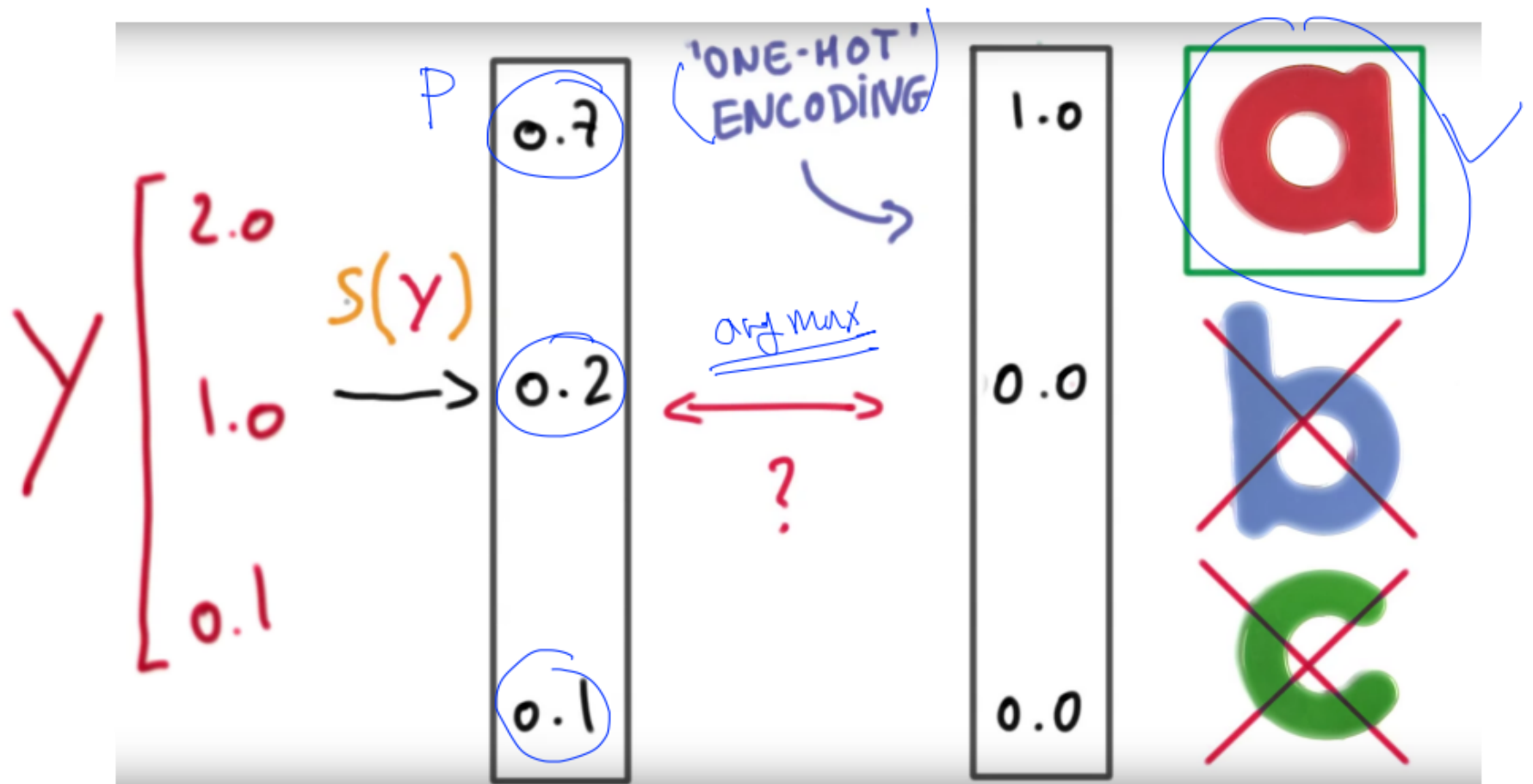
Where is sigmoid?

$$\begin{bmatrix} w_{A1} & w_{A2} & w_{A3} \\ w_{B1} & w_{B2} & w_{B3} \\ w_{C1} & w_{C2} & w_{C3} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} w_{A1}x_1 + w_{A2}x_2 + w_{A3}x_3 \\ w_{B1}x_1 + w_{B2}x_2 + w_{B3}x_3 \\ w_{C1}x_1 + w_{C2}x_2 + w_{C3}x_3 \end{bmatrix} = \begin{bmatrix} \bar{y}_A \\ \bar{y}_B \\ \bar{y}_C \end{bmatrix}$$



SOFTMAX

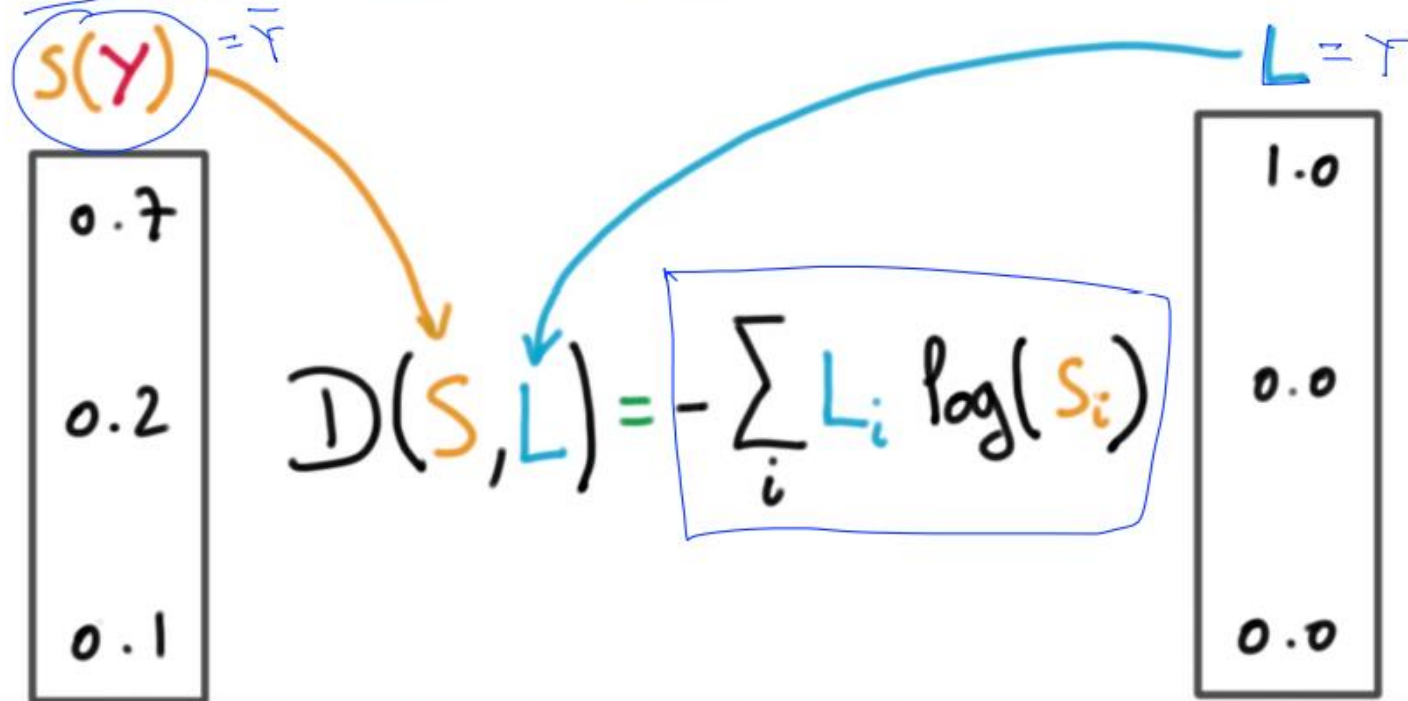




<https://www.udacity.com/course/viewer#!/c-ud730/l-6370362152/m-637981181>

Cost function

CROSS - ENTROPY



<https://www.udacity.com/course/viewer#!/c-ud730/l-6370362152/m-637981181>

Cross Entropy: 원래 확률과 추정 확률 사이의 차이

x1 x2 x3 x4 y

```
[12]: x_train = [[1, 2, 1, 1], 2
              [2, 1, 3, 2], 2
              [3, 1, 3, 4], 2
              [4, 1, 5, 5], 1
              [1, 7, 5, 5], 1
              [1, 2, 5, 6], 1
              [1, 6, 6, 6], 0
              [1, 7, 7, 7]] 0
#x_train (m,4) m은 샘플 갯수
#y_train (m,)
y_train = [2, 2, 2, 1, 1, 1, 0, 0]
x_train = torch.FloatTensor(x_train)
y_train = torch.LongTensor(y_train)
```

2=병아리
1=송아지
0=강아지

[illegible]

Epoch 1000/1000 Cost: 0.399962

```
tensor([[1.5171e-03, 2.3140e-02, 9.7534e-01],  
        [2.2182e-02, 2.1552e-01, 7.6229e-01],  
        [1.9853e-05, 3.0629e-01, 6.9369e-01],  
        [4.6897e-05, 5.0356e-01, 4.9639e-01],  
        [6.6322e-01, 2.9114e-01, 4.5647e-02],  
        [3.4381e-01, 6.5618e-01, 3.1115e-06],  
        [8.7086e-01, 1.2901e-01, 1.2303e-04],  
        [9.4367e-01, 5.6321e-02, 8.7894e-06]], grad_fn=<SoftmaxBackward>)
```

Epoch 1000/1000 Cost: 0.399962

tensor([[1.5171e-03, 2.3140e-02, 9.7534e-01],
[2.2182e-02, 2.1552e-01, 7.6229e-01],
[1.9853e-05, 3.0629e-01, 6.9369e-01],
[4.6897e-05, 5.0356e-01, 4.9639e-01],
[6.6322e-01, 2.9114e-01, 4.5647e-02],
[3.4381e-01, 6.5618e-01, 3.1115e-06],
[8.7086e-01, 1.2901e-01, 1.2303e-04],
[9.4367e-01, 5.6321e-02, 8.7894e-06]])

2
2
2
1
1
1
0
0

grad_fn=<SoftmaxBa

ckward>)