Training of Multi-Layer NN: Cost Functions SSE vs. CE

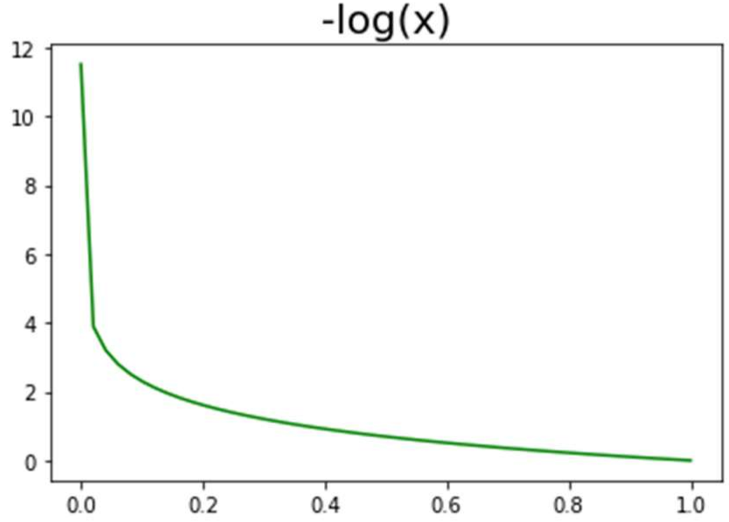
Information Content = Surprisal

- Quantity of Information(QI, 정보량), h(A)
- h(A):=－logP(A)

where P(A)=Probability of Event A

0<=P(A)<=1

Therefore h(A)=(0,∞].



If P(A)=0.99, h(A)=0.01

P(B)=0.01, h(B)=4.61

QI is higher for low probability of event B, i.e., higher surprises (= 확률이 낮은 사건 B의 정보량이 높음, 즉 놀람의 정보가 높음)

Entropy

- Entropy: Average QI (평균정보량), $\mathrm{H}(X) = \mathrm{E}[\mathrm{I}(X)] = \mathrm{E}[-\log(\mathrm{P}(X))].$
  - Reflects average surprises (=불확실성 정보 반영, 평균 놀람 정보).
- Random Variable, X then QI is given as H[X]

  $H[X] \coloneqq -\sum_{i=1}^{N} p_i \log(p_i)$ pi:=P(X=xi)
- Ex.) if X=0, 1

H[X]=－[P(X=0)logP(X=0)+P(X=1)logP(X=1)]

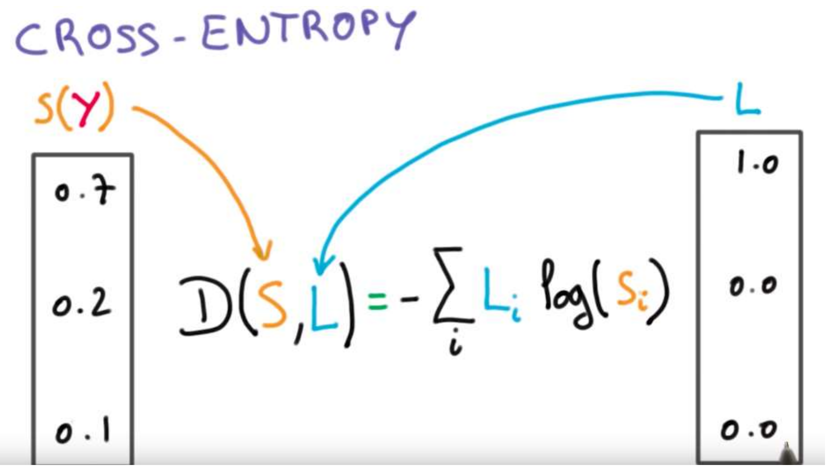
Case 1 P(X=0)=0.5, P(X=1)=0.5
H[X]=－(0.5log0.5+0.5log0.5)=0.69.

Case 2 P(X=0)=0.8, P(X=1)=0.2
H[X]=－(0.8log0.8+0.2log0.2)=0.50.

Case 3 P(X=0)=1, P(X=1)=0
H[X]=－(1log1+0log0)=0. Entropy is zero when one outcomes is certain to occur

Case 3: Low H, as low surprises (=불확실성(놀람)이 적을 수록 값이 적음)
Case 1: High H, as high surprises (=불확실성(놀람)이 클수록 값이 큼)

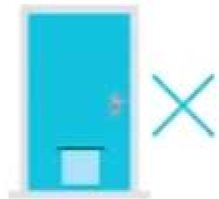
Cross Entropy



- *CE* measures the distance (or difference) between L (target) and S (output)
- Good model, lower CE (L=S), bad model high CE
- Minimize CE => Better Model => Better Solution

CE Example

CE Example

			
P(gift)	0.8	0.7	0.1
P(no gift)	0.2	0.3	0.9

CE Example

			Probability	-ln(Probability)
🎁 0.8	🎁 0.7	🎁 0.1	0.056	2.88
🎁 0.8	🎁 0.7	× 0.9	0.504	0.69
🎁 0.8	× 0.3	🎁 0.1	0.024	3.73
× 0.2	🎁 0.7	🎁 0.1	0.014	4.27
🎁 0.8	× 0.3	× 0.9	0.216	1.53
× 0.2	🎁 0.7	× 0.9	0.126	2.07
× 0.2	× 0.3	🎁 0.1	0.006	5.12
× 0.2	× 0.3	× 0.9	0.054	2.92

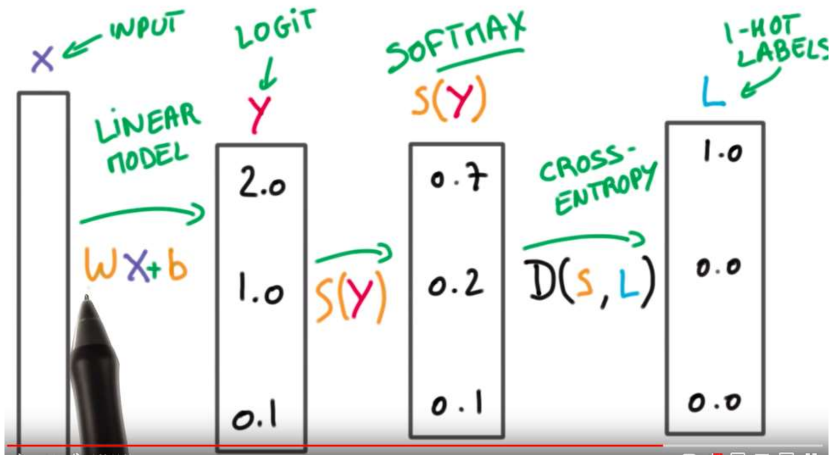
Consider $-\sum_{i=1}^{N} Log p_i$

- Lower, good choice (model)
- Higher, bad choice (model)

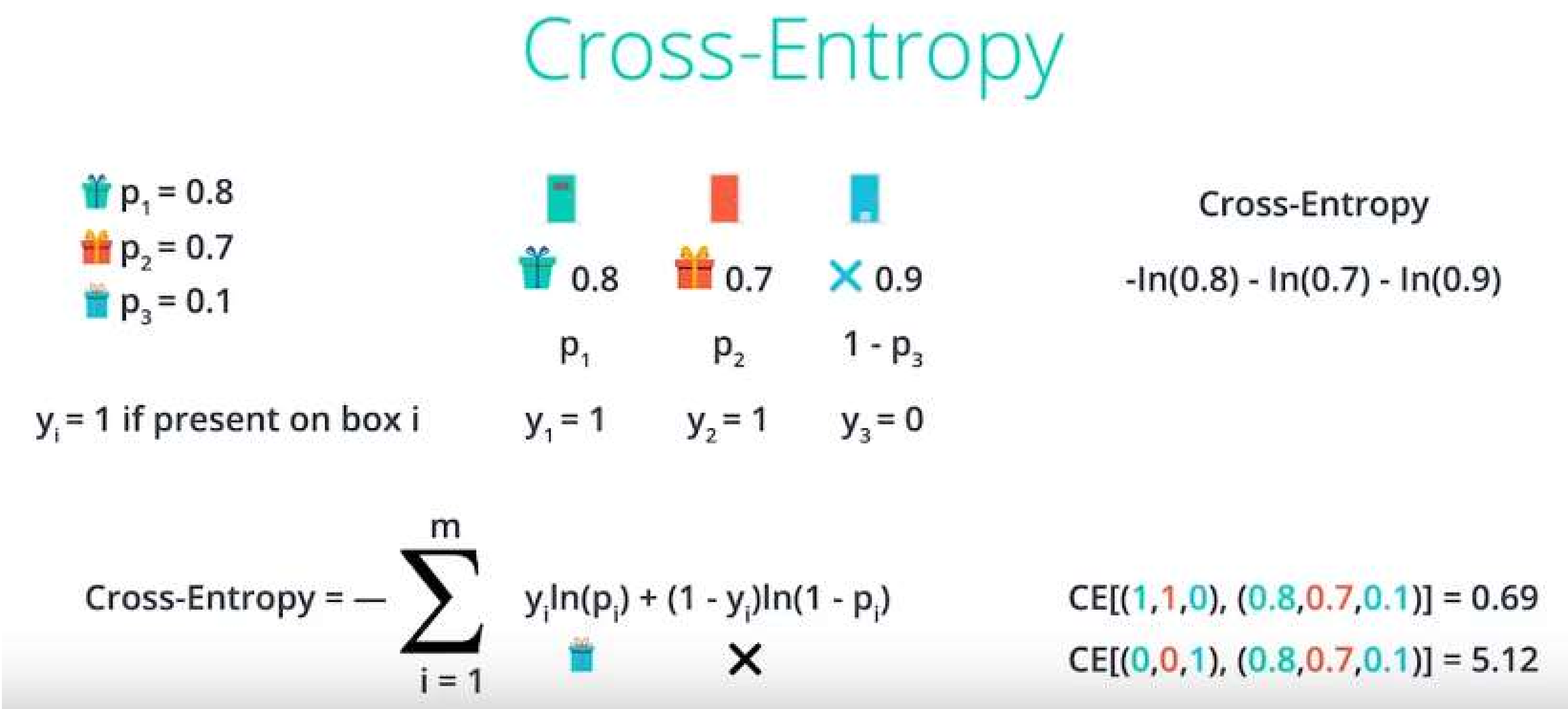
Binary Cross Entropy

$$CE(t, y) \coloneqq -\sum_{i=1}^{N} t_i \ln(y_i) + (1 - t_i)\ln(1 - y_i)$$

- *CE* measures the distance (or difference) between t (target or **L**) and y (output or **S**)
- **Consider ti=1 or ti=0 (as in textbook)**
- Good model, lower CE (t=y, or L=S)
- bad model, higher CE (t≠y, or L≠S)
- Minimize CE => Better Model => Better Solution



Example



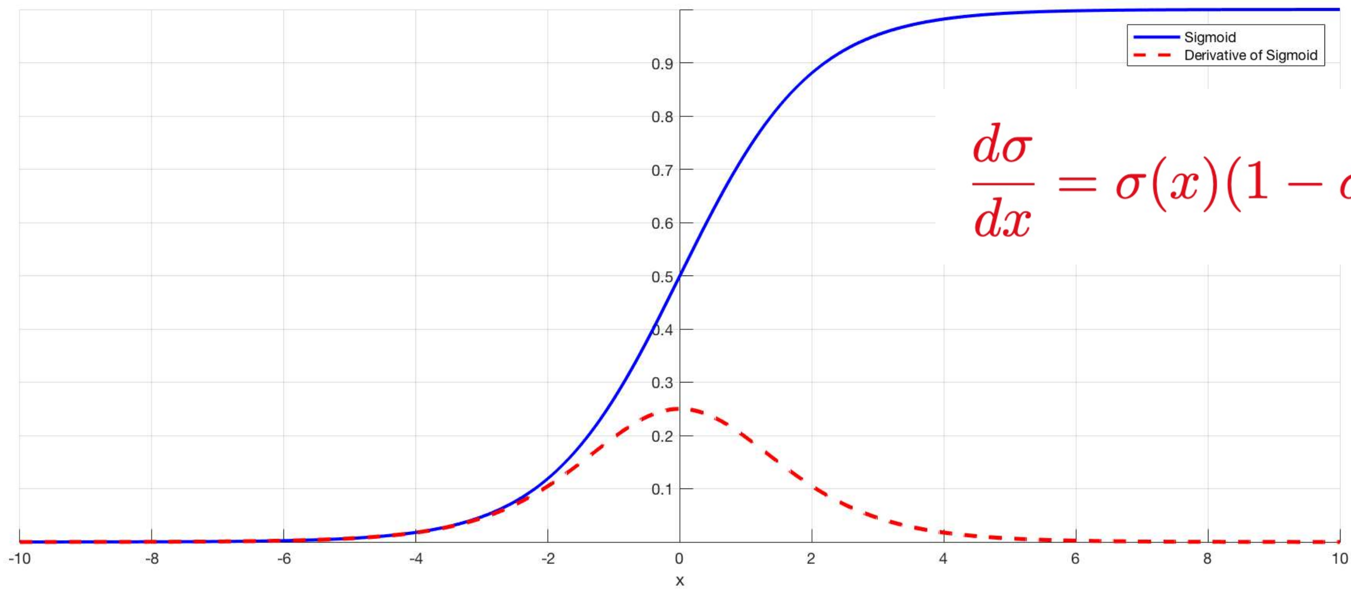
- Low CE, good match, good model
- High CE, bad match, bad model

Cross-Entropy as Cost Function

Improving the way neural networks learn

- Read about CE in English at http://neuralnetworksanddeeplearning.com/chap3.html

- Read the CE Handout in Korean



$$\frac{d\sigma}{dx} = \sigma(x)(1 - \sigma(x))$$

Let's denote the sigmoid function as $\sigma(x) = \frac{1}{1+e^{-x}}$.

The derivative of the sigmoid is $\frac{d}{dx}\sigma(x) = \sigma(x)(1-\sigma(x))$.

Here's a detailed derivation:

$$\begin{aligned}
\frac{d}{dx}\sigma(x) &= \frac{d}{dx}\left[\frac{1}{1+e^{-x}}\right] \\
&= \frac{d}{dx}\left(1+\mathrm{e}^{-x}\right)^{-1} \\
&= -(1+e^{-x})^{-2}(-e^{-x}) \\
&= \frac{e^{-x}}{(1+e^{-x})^2} \\
&= \frac{1}{1+e^{-x}} \cdot \frac{e^{-x}}{1+e^{-x}} \\
&= \frac{1}{1+e^{-x}} \cdot \frac{(1+e^{-x})-1}{1+e^{-x}} \\
&= \frac{1}{1+e^{-x}} \cdot \left(\frac{1+e^{-x}}{1+e^{-x}} - \frac{1}{1+e^{-x}}\right) \\
&= \frac{1}{1+e^{-x}} \cdot \left(1 - \frac{1}{1+e^{-x}}\right) \\
&= \sigma(x) \cdot (1-\sigma(x))
\end{aligned}$$