

UDACITY Deep Learning
1강



Bio Keras

"From Machine Learning
To Deep Learning"

1. What is Deep Learning?



DEEP LEARNING



LOTS OF DATA

+



COMPLEX PROBLEMS
TO SOLVE



PERCEPTION



RECOGNIZING IMAGES



WHAT PEOPLE
ARE SAYING



HELPING ROBOTS INTERACT
WITH THE WORLD

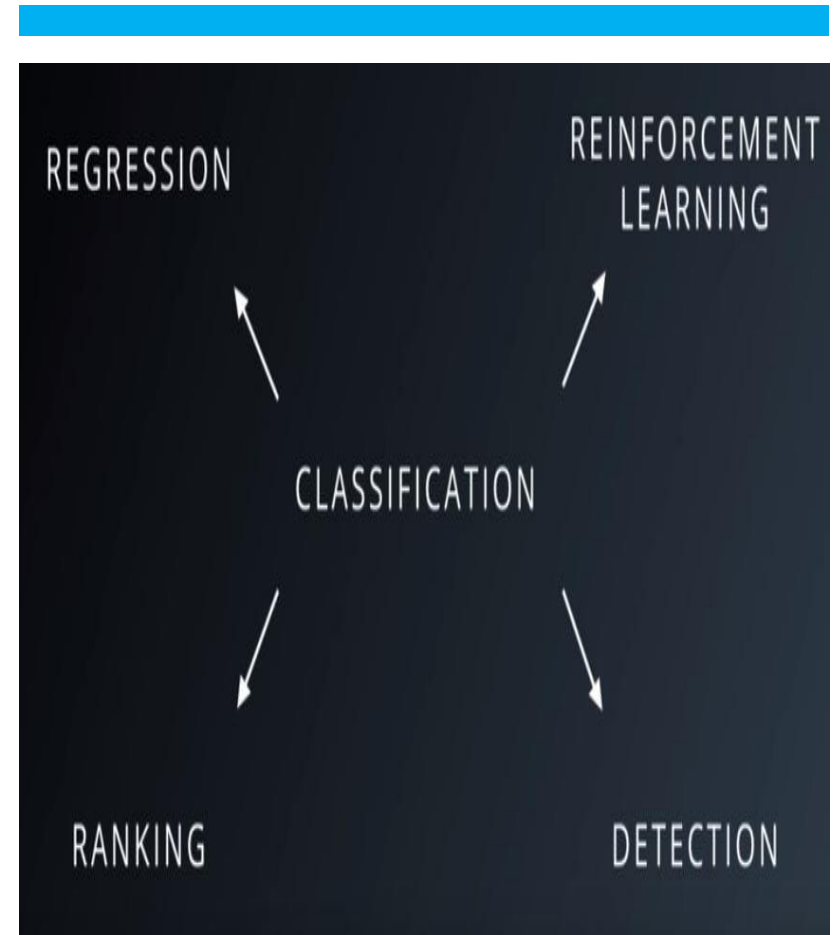


COMPUTER VISION

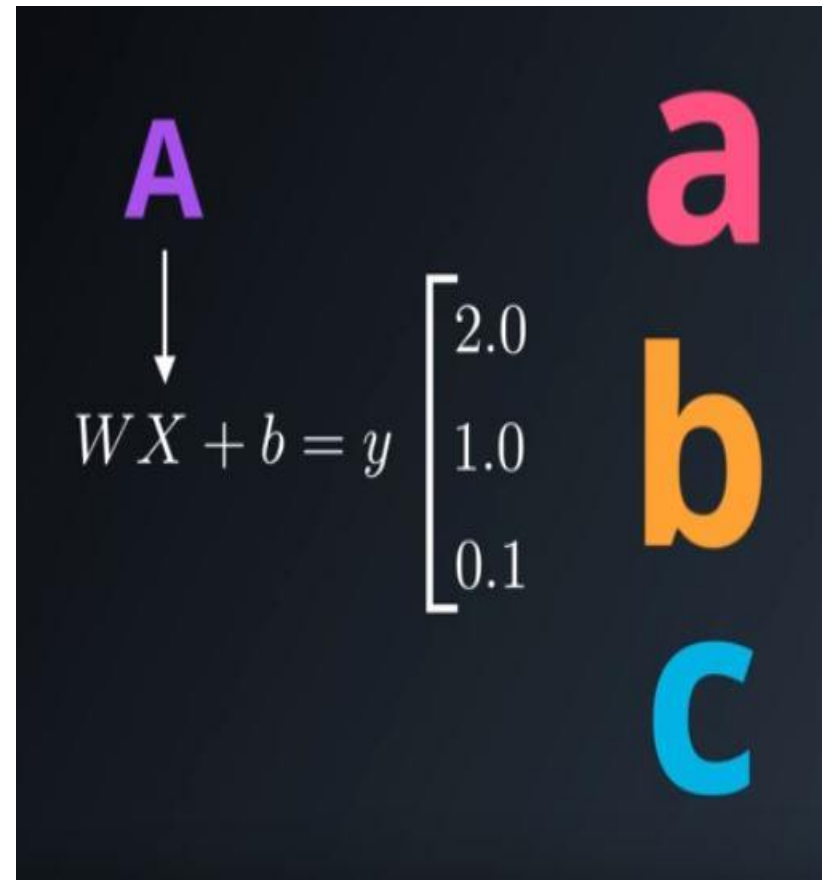
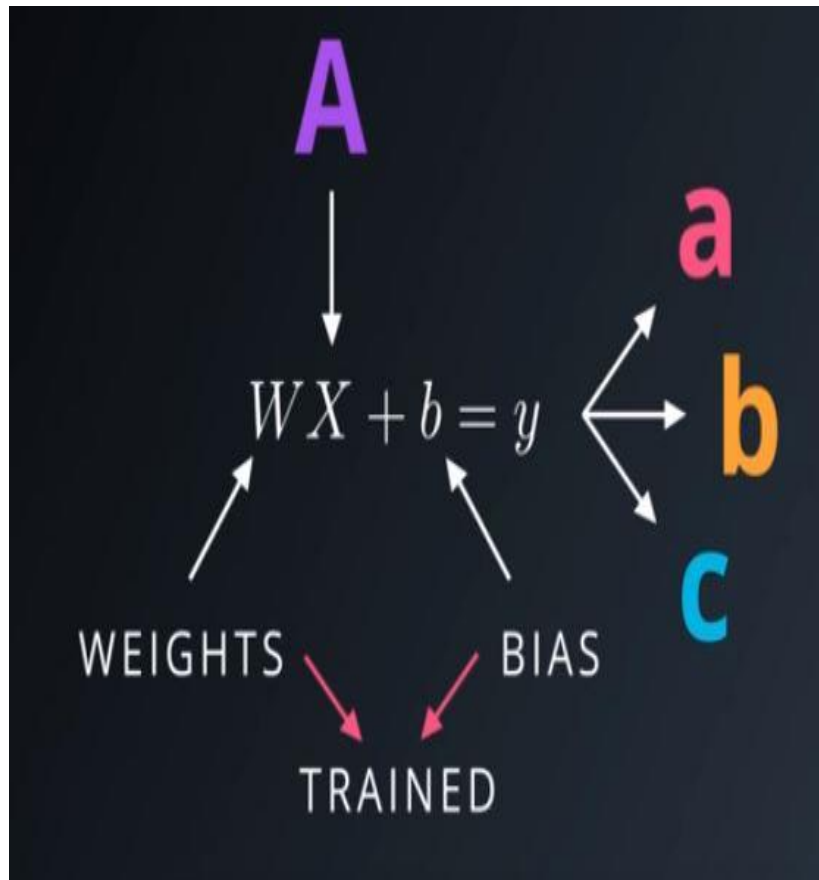


SPEECH RECOGNITION

2. Supervised Classification



2. Supervised Classification



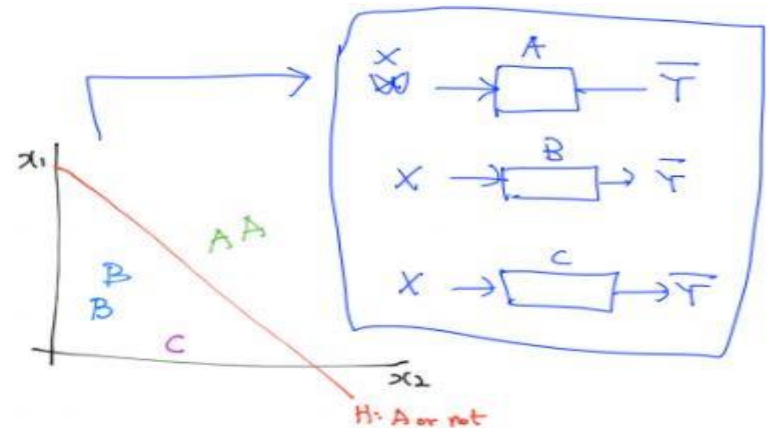
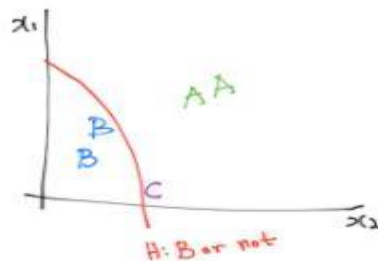
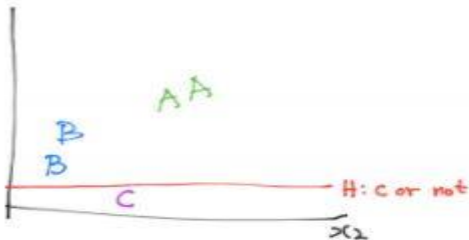
2. Supervised Classification



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

A, B, C 세 개가 있을 때

- A or B, C
- B or A, C
- C or A, C 이렇게 세개의 Classifier가 있으면 가능 하다.

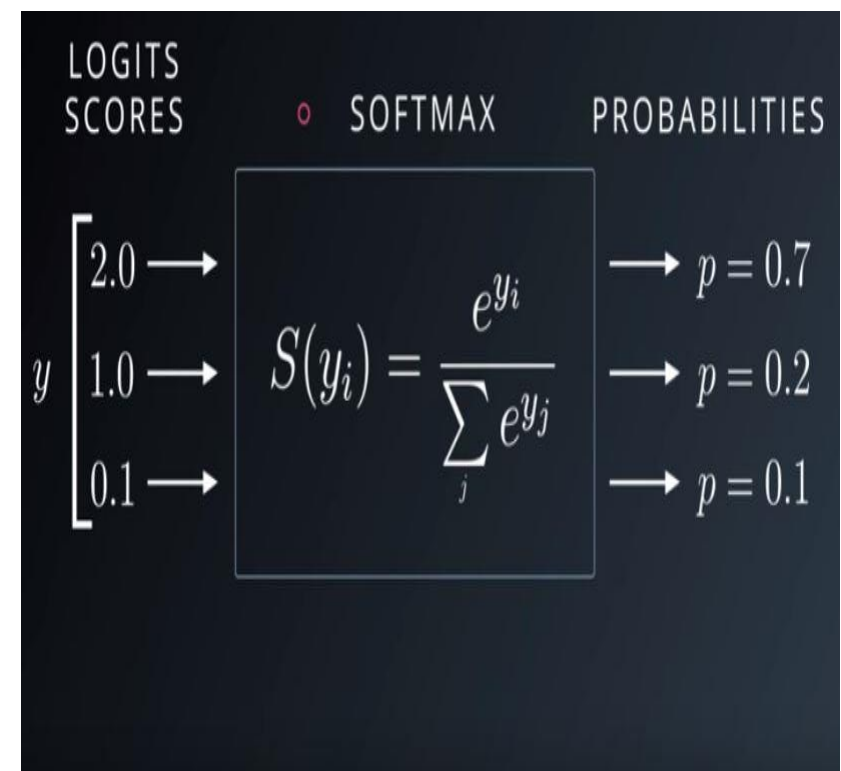
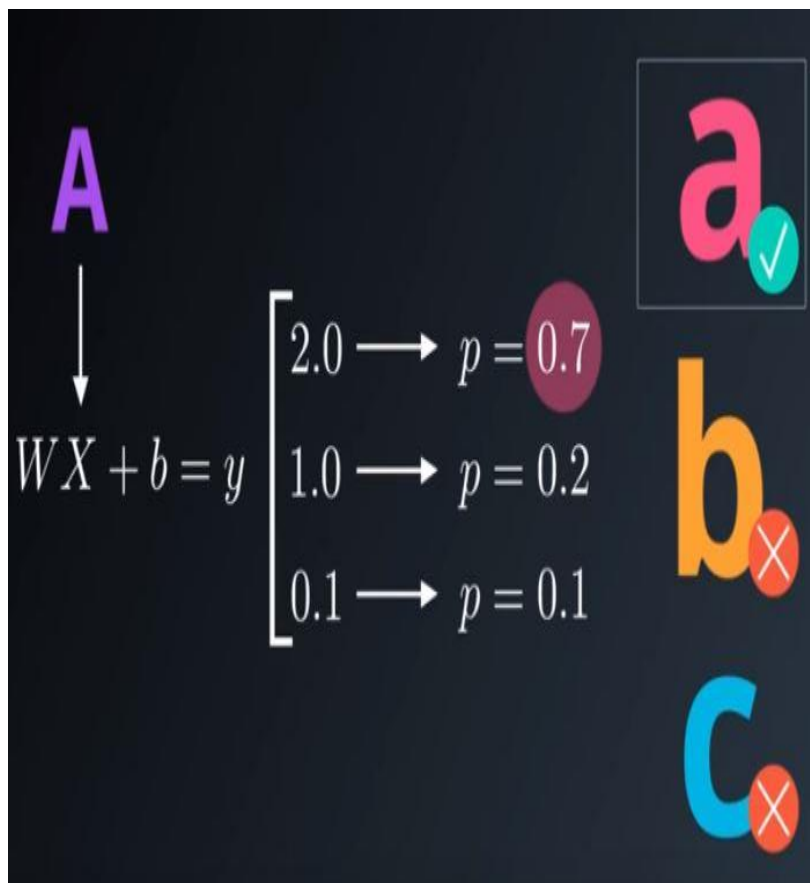


3. Softmax



1. $0 \leq y_i \leq 1 (i = 1, 2, \dots, n)$

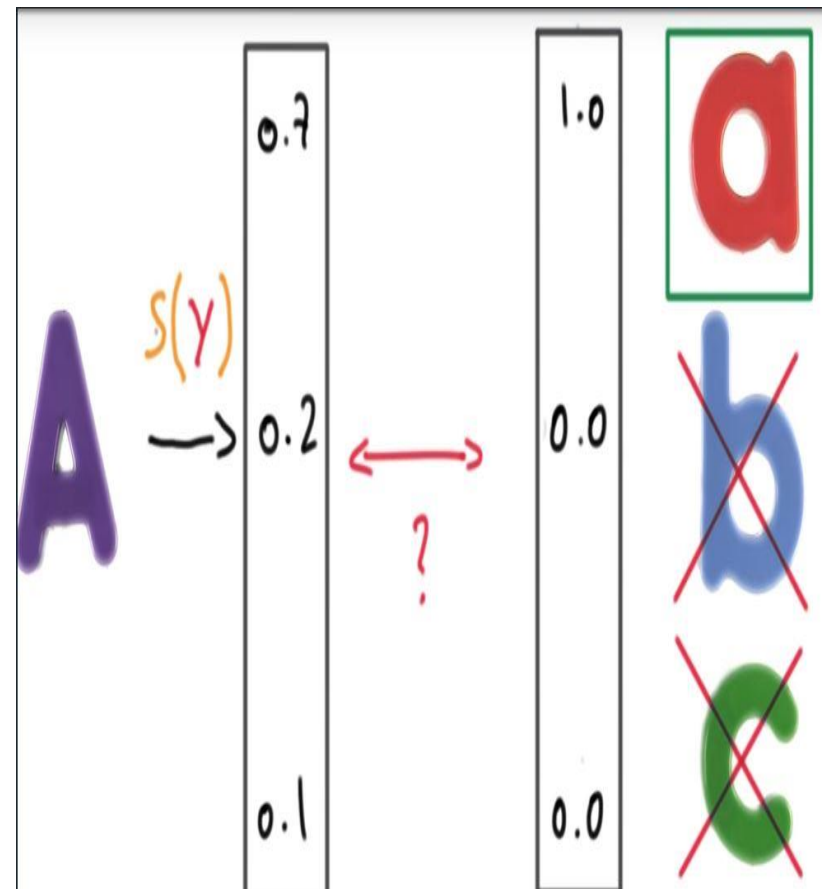
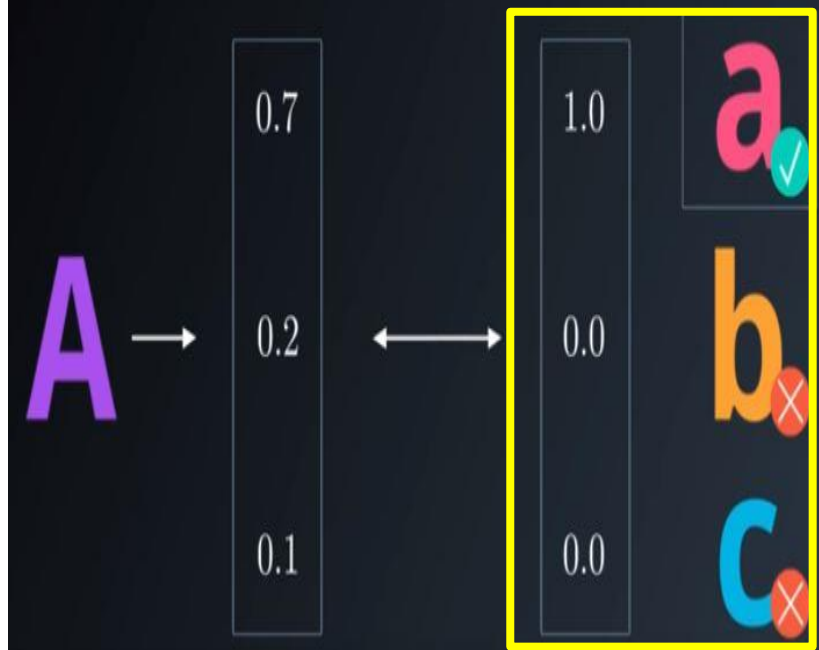
2. $\sum S(y_i) = 1$



4. Cross – Entropy



◦ 'ONE-HOT' ENCODING

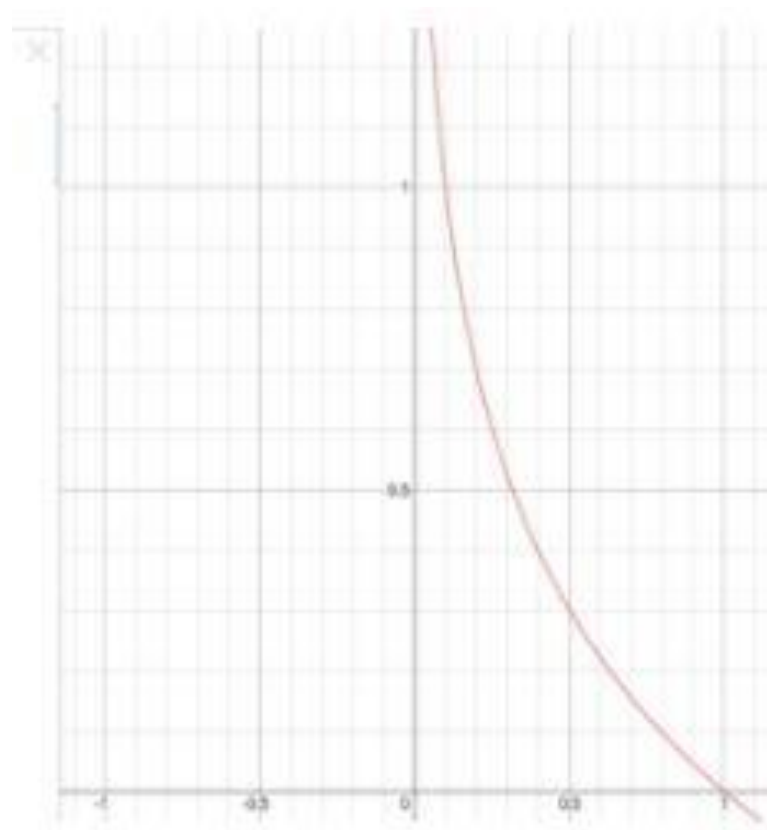
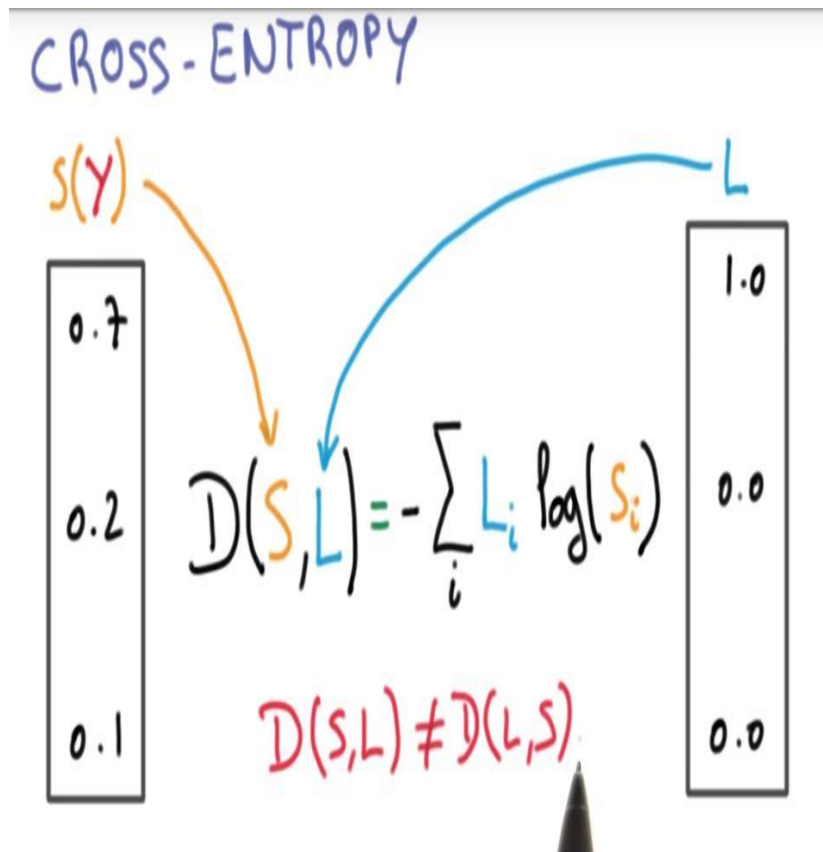


4. Cross - Entropy

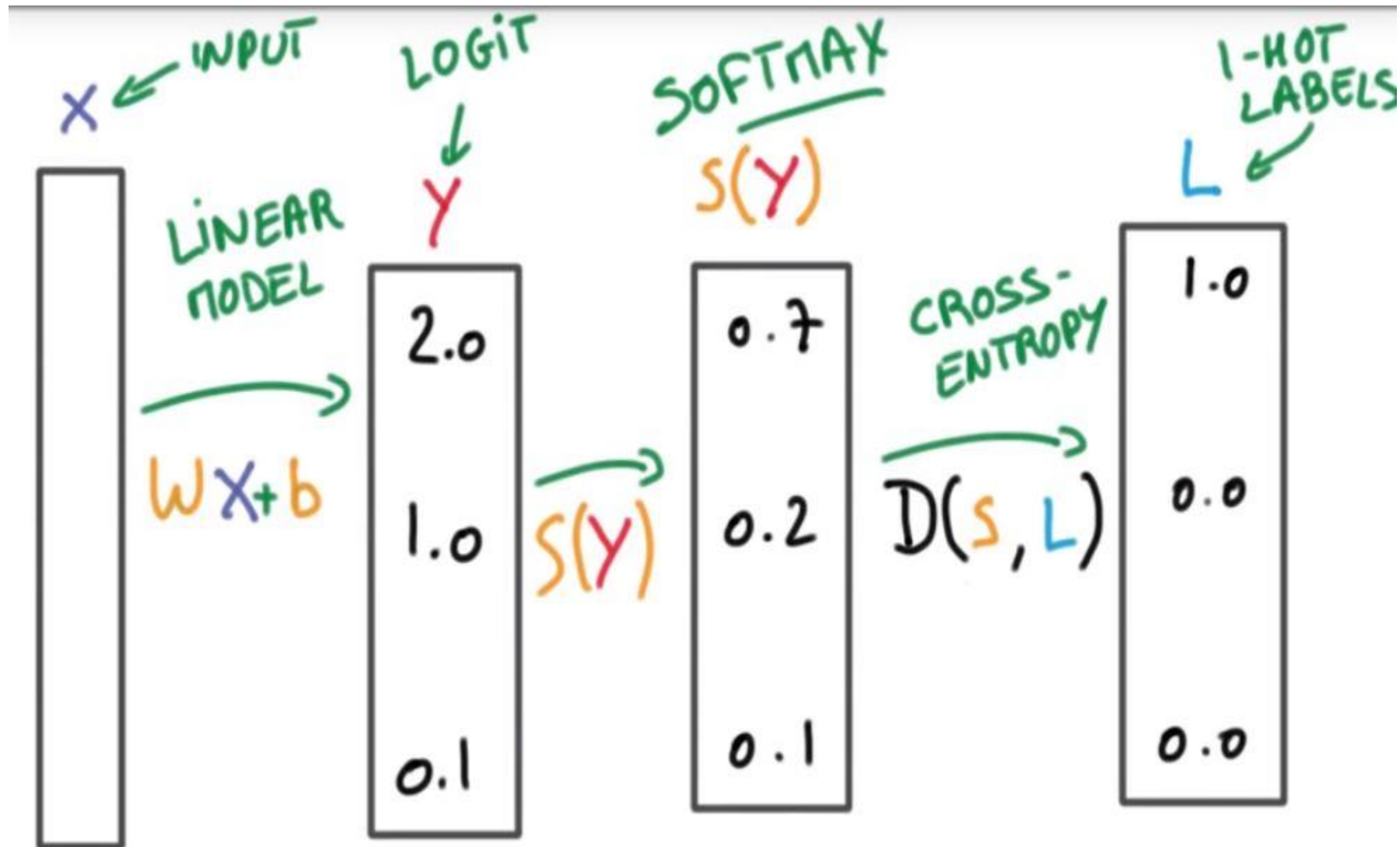


$$-\sum L_i \log S(y_i) = -\sum L_i \log S(\hat{y}_i)$$

$y = -\log(x)$ 의 그래프



5. Recapitulation



6. Optimization



$$D(S(wx + b), L)$$

$$D(A, a) \quad D(A, b)$$

$$\mathcal{L} = \frac{1}{N} \sum_i D(S(wx_i + b), L_i)$$

TRAINING SET

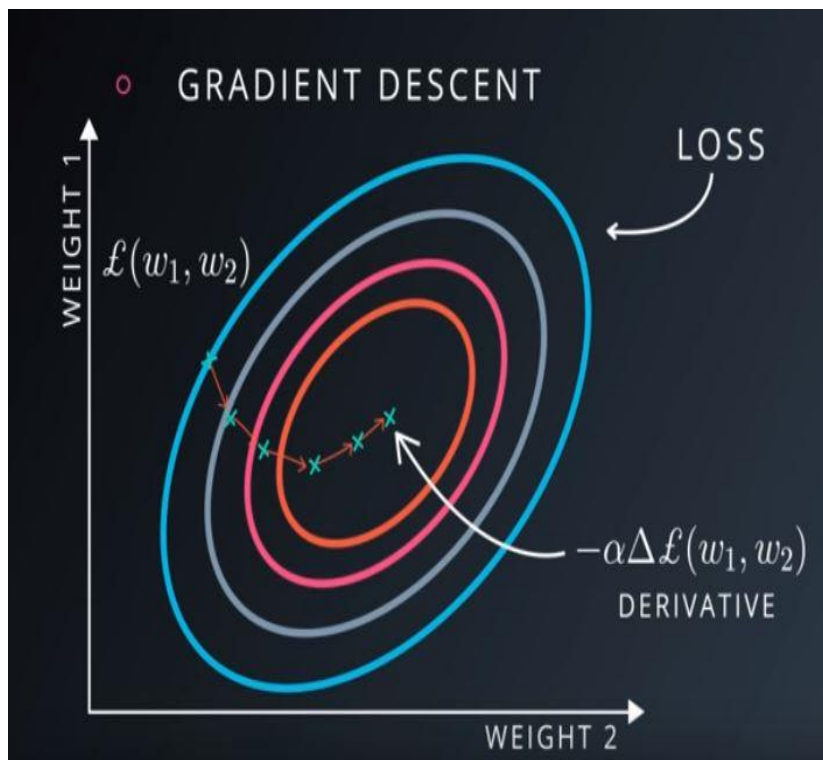
LOSS = AVERAGE CROSS-ENTROPY

$$\mathcal{L} = \frac{1}{N} \sum_i D(S(wx_i + b), L_i)$$

BIG MATRIX!!

BIG SUM!!

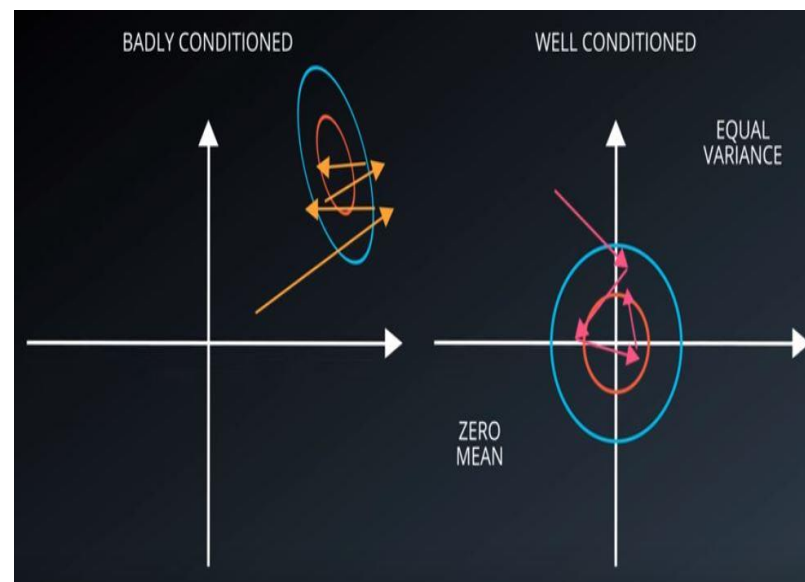
6. Optimization



원점수(raw score)를 z점수로 변환하기

$$z = \frac{X - \mu}{\sigma} \quad (\text{전집 (population) 자료의 경우})$$

$$z = \frac{X - \bar{X}}{s} \quad (\text{표본 (sample) 자료의 경우})$$



6. Optimization



WEIGHT INITIALIZATION

$$D(S(WX + b), L)$$

$W_0?$ $b_0?$



LOGITS
SCORES

SOFTMAX

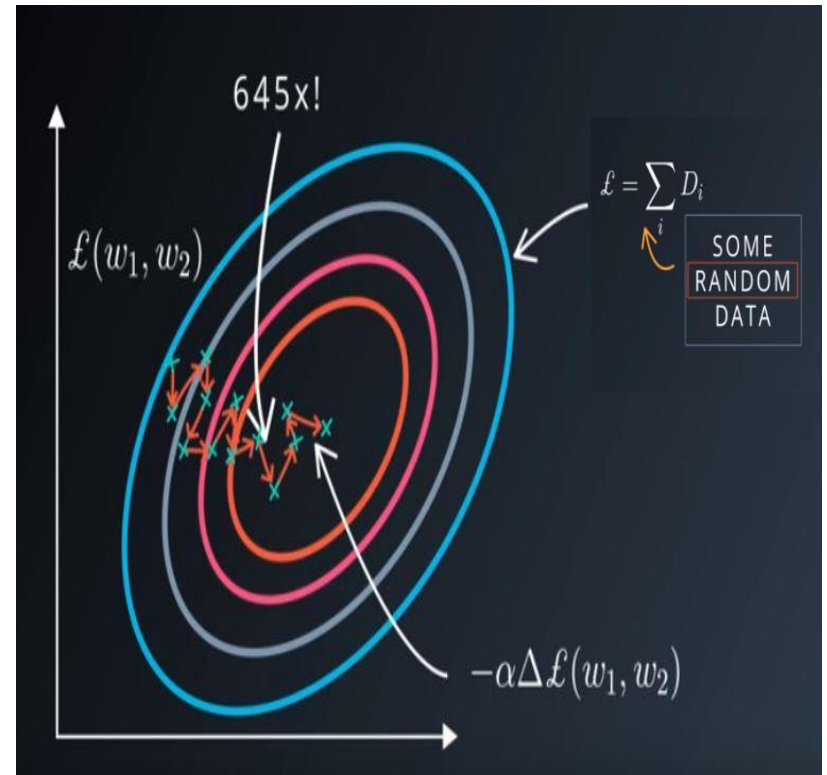
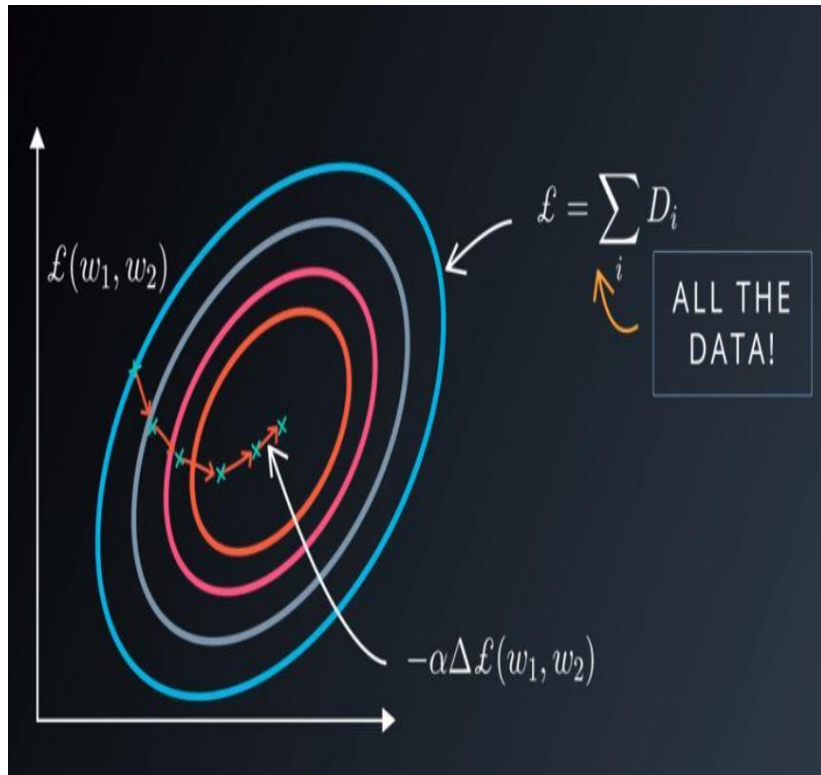
PROBABILITIES

y	$\begin{bmatrix} 2.0 \rightarrow \\ 1.0 \rightarrow \\ 0.1 \rightarrow \end{bmatrix}$	$S(y_i) = \frac{e^{y_i}}{\sum_j e^{y_j}}$	$\begin{matrix} \rightarrow p = 0.7 \\ \rightarrow p = 0.2 \\ \rightarrow p = 0.1 \end{matrix}$
-----	---	---	---

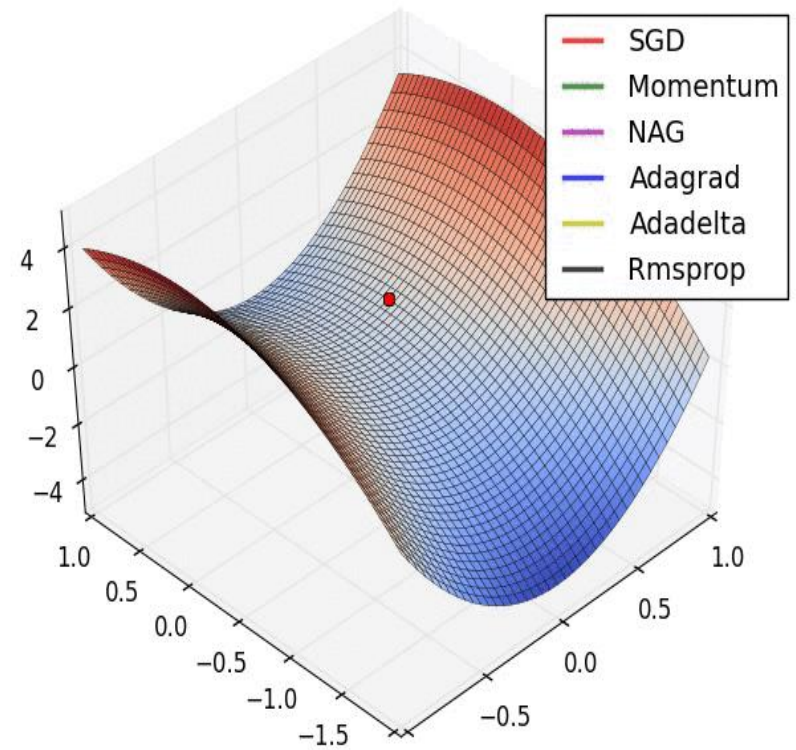
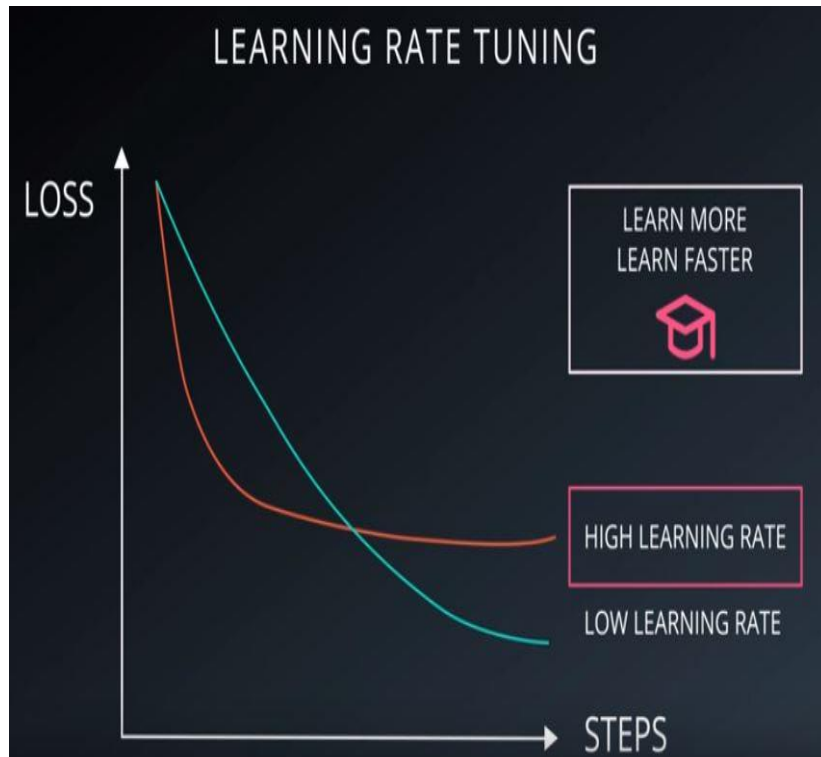


22. Measuring Performance

6. Stochastic Gradient Descent



6. Stochastic Gradient Descent



6. Stochastic Gradient Descent



SGD "BLACK MAGIC"

HYPER-PARAMETERS

~~INITIAL LEARNING RATE~~

~~LEARNING RATE DECAY~~

~~MOMENTUM~~

BATCH SIZE

WEIGHT INITIALIZATION

ADAGRAD

Remember

KEEP CALM AND
LOWER YOUR
LEARNING RATE



Momentum

Momentum 방식은 말 그대로 Gradient Descent를 통해 이동하는 과정에 일종의 '관성'을 주는 것이다. 현재 Gradient를 통해 이동하는 방향과는 별개로, 과거에 이동했던 방식을 기억하면서 그 방향으로 일정 정도를 추가적으로 이동하는 방식이다. 수식으로 표현하면 다음과 같다. v_t 를 time step t 에서의 이동 벡터라고 할 때, 다음과 같은 식으로 이동을 표현할 수 있다.

$$v_t = \gamma v_{t-1} + \eta \nabla_{\theta} J(\theta)$$

$$\theta = \theta - v_t$$

이 때, γ 는 얼마나 momentum을 줄 것인지에 대한 momentum term으로서, 보통 0.9 정도의 값을 사용한다. 식을 살펴보면 과거에 얼마나 이동했는지에 대한 이동 항 v 를 기억하고, 새로운 이동항을 구할 경우 과거에 이동했던 정도에 관성항만큼 곱해준 후 Gradient를 이용한 이동 step 항을 더해준다. 이렇게 할 경우 이동항 v_t 는 다음과 같은 방식으로 정리할 수 있어, Gradient들의 지수평균을 이용하여 이동한다고도 해석할 수 있다.

$$v_t = \eta \nabla_{\theta} J(\theta)_t + \gamma \eta \nabla_{\theta} J(\theta)_{t-1} + \gamma^2 \eta \nabla_{\theta} J(\theta)_{t-2} + \dots$$