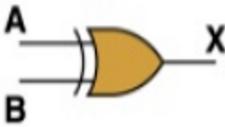


8. Neural Network (인공 신경망)

2019년 4월 7일 일요일 오후 8:53

1. XOR

XOR data set

Boolean Expression	Logic Diagram Symbol	Truth Table															
$X = A \oplus B$		<table border="1"><thead><tr><th>A</th><th>B</th><th>X</th></tr></thead><tbody><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></tbody></table>	A	B	X	0	0	0	0	1	1	1	0	1	1	1	0
A	B	X															
0	0	0															
0	1	1															
1	0	1															
1	1	0															

x_1	x_2	x_{OR}
0	0	0 (-)
0	1	1 (+)
1	0	1 (+)
1	1	0 (-)

xor

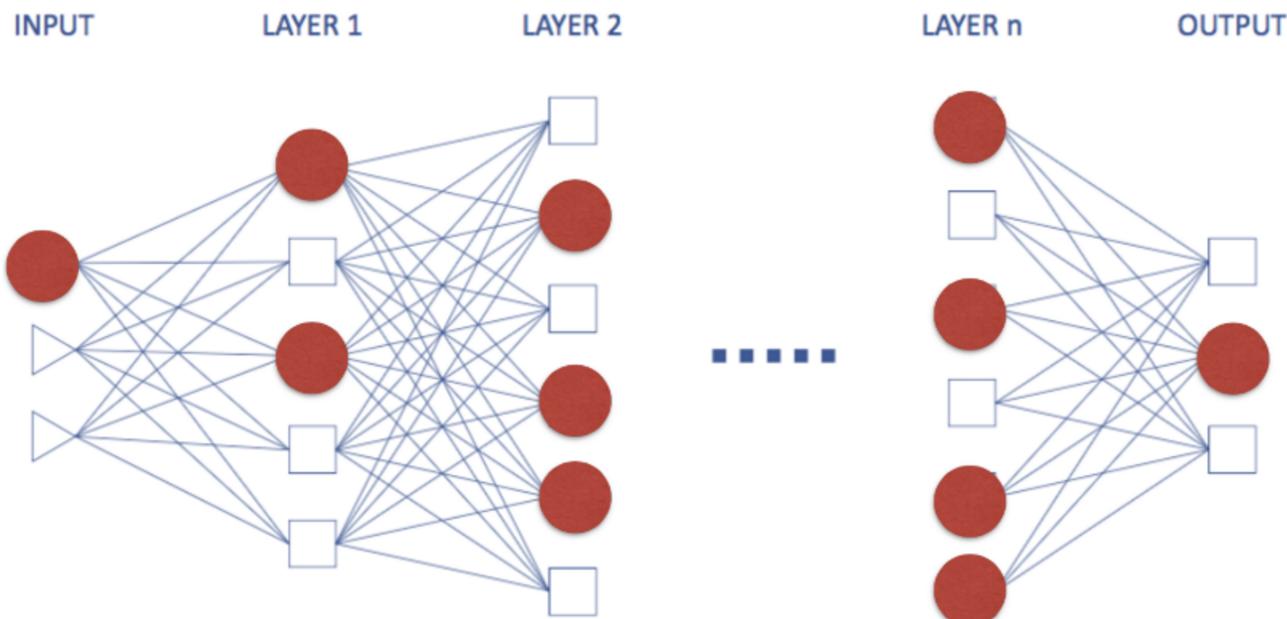
	+	-
1	-	-
0	-	+

Nope

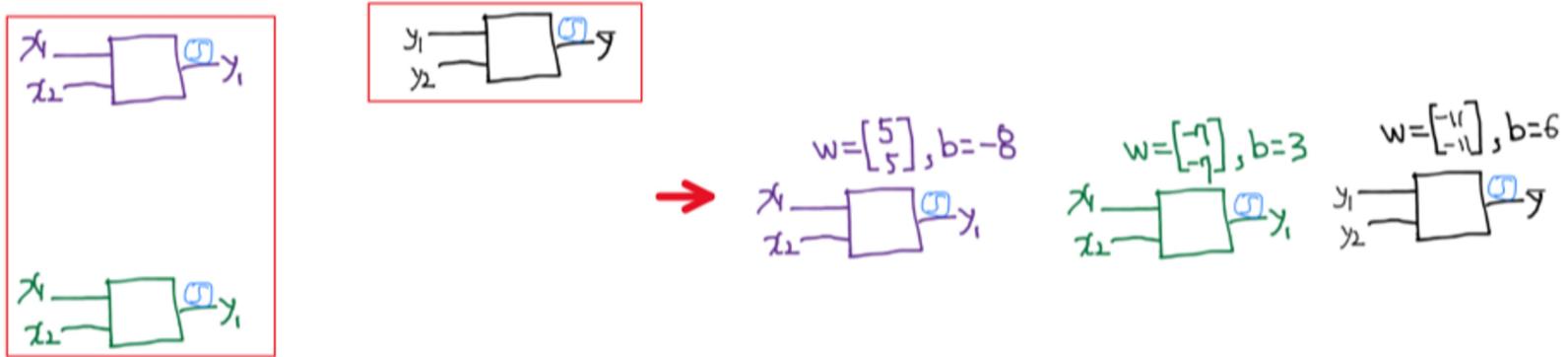
- Logistic regression으로 해결 불가능
- Neural nets가 더 Deep, Wide 해질 수록 예측이 더 정확해짐

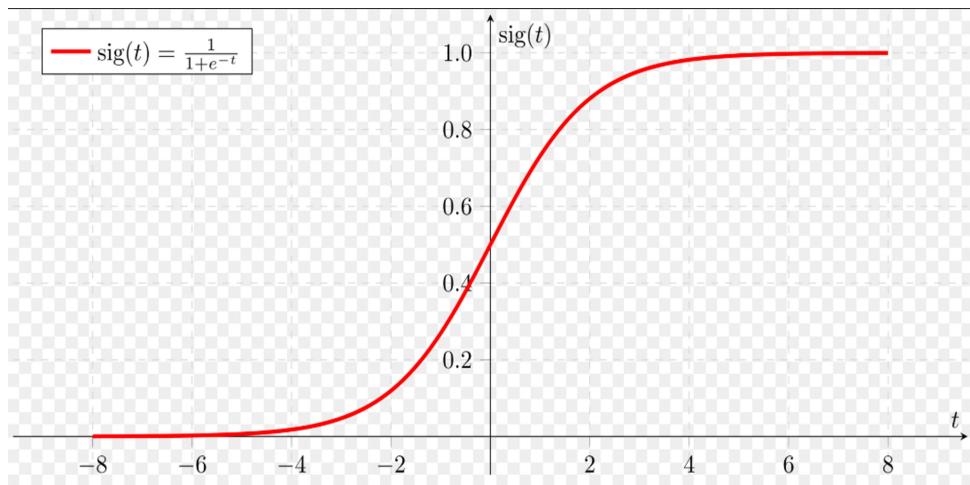
2. NN

- neural network는 사람의 뇌를 본 따서 만든 머신러닝 모델이다
- full name은 artificial neural network이지만, 일반적으로 neural network라고 줄여서 부른다



Neural Net





$$[0 \ 6] \begin{bmatrix} 5 \\ 5 \end{bmatrix} - 8 = 0 + 0 - 8 = -8, \text{ sigmoid}(-8) = 0$$

$$[0 \ 0] \begin{bmatrix} -7 \\ -7 \end{bmatrix} + 3 = 0 + 0 + 3 = 3, \text{ sigmoid}(3) = 1$$

$$[0 \ 1] \begin{bmatrix} -11 \\ -11 \end{bmatrix} + 6 = 0 + -11 + 6 = -5$$

$$\text{sigmoid}(-5) = 0$$

x_1	x_2	y_1	y_2	\bar{y}	x_{OR}
0	0	0	1	0	0
0	1			0	1
1	0			1	1
1	1			1	0

$$[1 \ 0] \begin{bmatrix} 5 \\ 5 \end{bmatrix} - 8 = 5 + 0 - 8 = -3, \text{Sigmoid}(-3) = 0$$

$$[1 \ 0] \begin{bmatrix} -1 \\ -1 \end{bmatrix} + 3 = -1 + 0 + 3 = -4, \text{Sigmoid}(-4) = 0$$

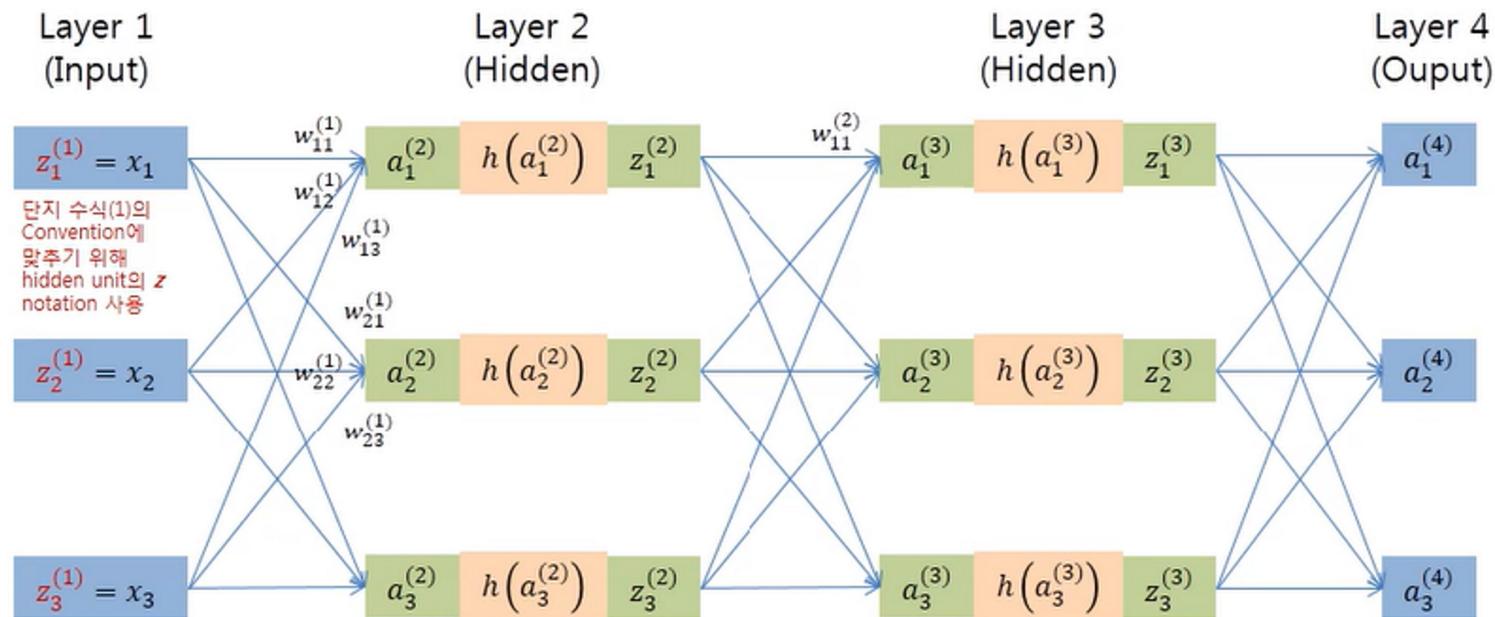
$$[0 \ 0] \begin{bmatrix} -1 \\ -1 \end{bmatrix} + 6 = 0 + 0 + 6 = 6$$

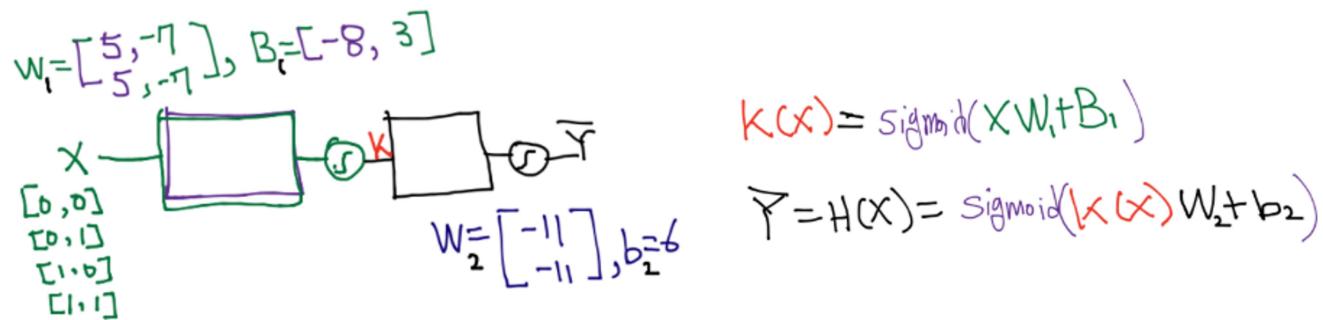
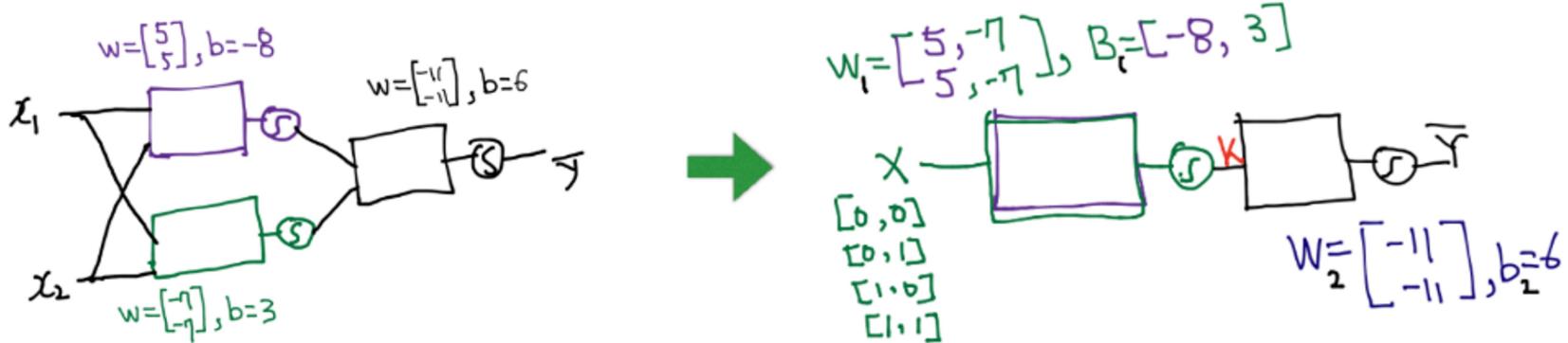
Sigmoid(6) = 1

x_1	x_2	y_1	y_2	\bar{y}	$x \oplus R$
0	0	0	1	0	0
0	1	0	0	1	1
1	0	0	0	1	1
1	1	1	1	0	0

3. Forward Propagation

Feed-forward Network





$$K(x) = \text{sigmoid}(xW_1 + B_1)$$

$$\bar{Y} = H(x) = \text{sigmoid}(K(x)W_2 + b_2)$$

4. Hypothesis

```
# NN
K = tf.sigmoid(tf.matmul(X, W1) + b1)
hypothesis = tf.sigmoid(tf.matmul(K, W2) + b2)
```

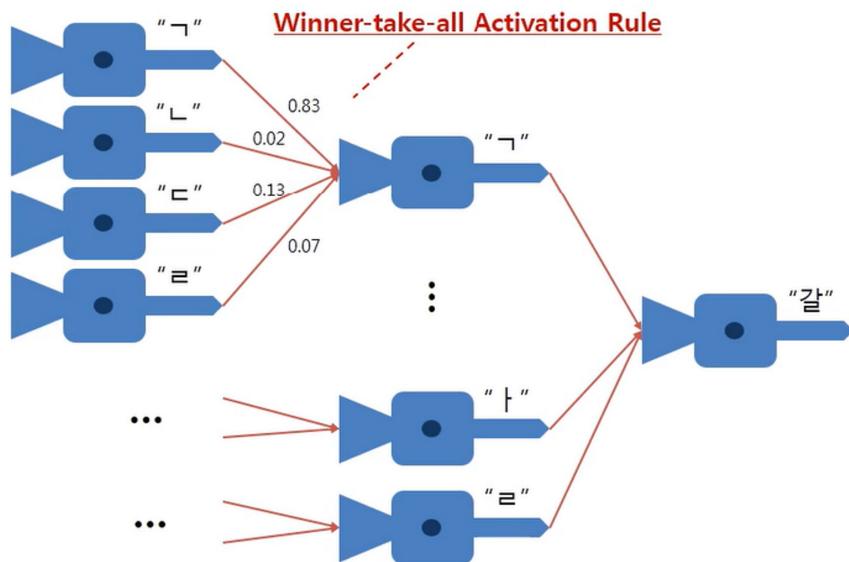
1. Definition

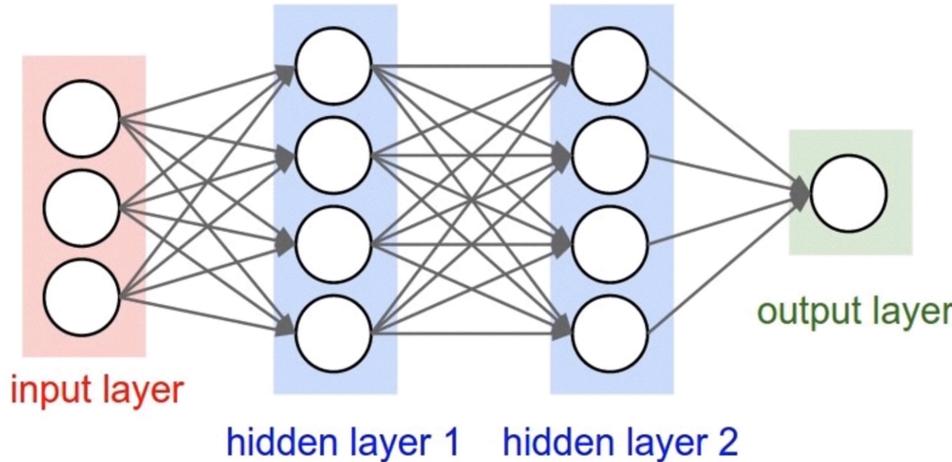
- Model of Neural Network: neuron, synapse, activation function :

먼저 뉴런들이 node이고, 그 뉴런들을 연결하는 시냅스가 edge인 네트워크를 만드는 것이 가능하다. 각각의 시냅스의 중요도(weight) 가 다를 수 있으므로 edge마다 weight를 따로 정의하게 되면 아래 그림과 같은 형태로 네트워크를 만들 수 있다. (출처: [위키](#))

Neuron of Biological Neural Network

- 예시 - 뉴런의 활성화 (Activation) 을 통한 문자 인식 모델





- **Back propagation**

마지막으로 이제 weight를 어떻게 찾을 수 있는지 weight parameter를 찾는 알고리즘에 대해 알아보자.

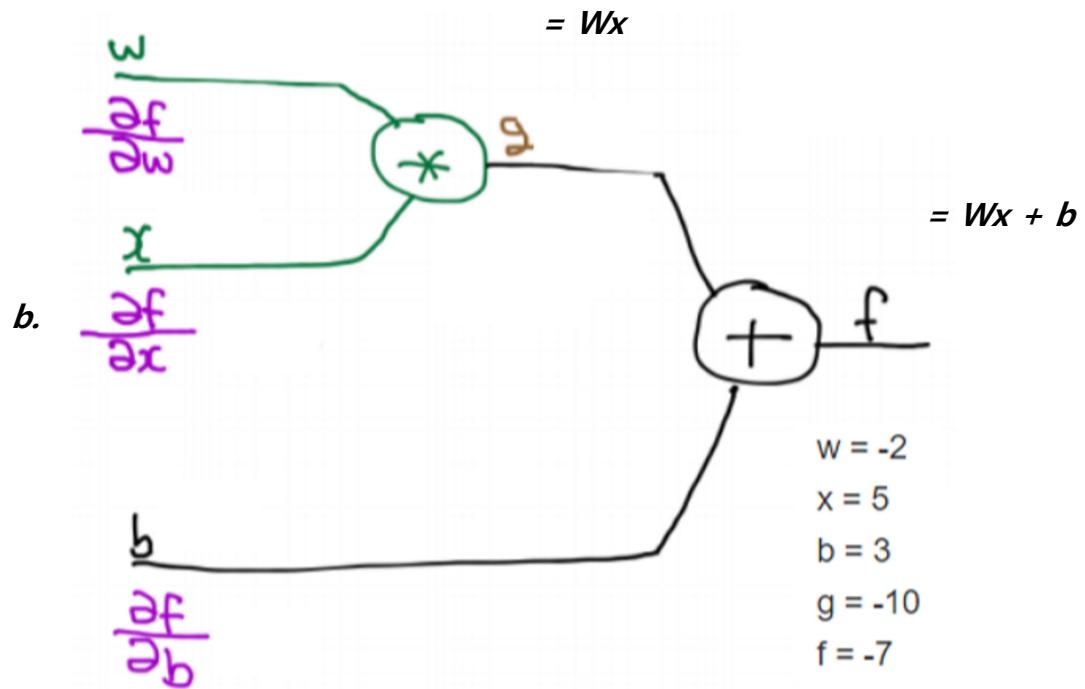
먼저 한 가지 알아두어야 할 점은 activation function들이 non-linear하고, 이것들이 서로 layer를 이루면서 복잡하게 얹혀있기 때문에 neural network의 weight optimization이 non-convex optimization이라는 것이다.

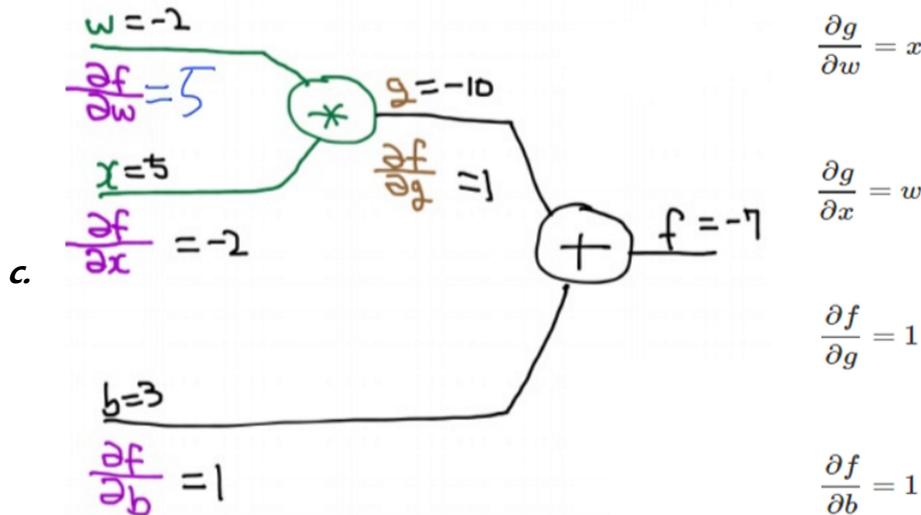
따라서 일반적인 경우에 neural network의 parameter들의 global optimum을 찾는 것은 불가능하다. 그렇기 때문에 보통 gradient descent 방법을 사용하여 적당한 값까지 수렴시키는 방법을 사용하게 된다.

Neural network (이 글에서는 multi-layer feed-forward network)의 parameter를 update하기 위해서는 backpropagation algorithm이라는 것을 주로 사용하는데, 이는 단순히 neural network에서 gradient descent를 chain rule을 사용하여 단순화시킨 것에 지나지 않는다

- $f = Wx + b, g = Wx, f = g + b$

- forward ($w=-2$, $x=5$, $b=3$)
- backward





$$\frac{\partial g}{\partial w} = x$$

$$\frac{\partial g}{\partial x} = w$$

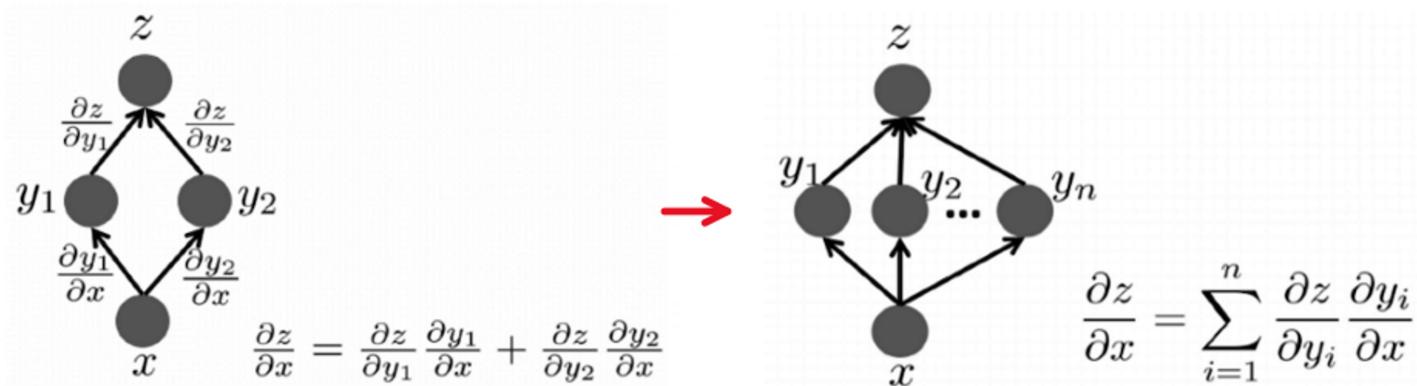
$$\frac{\partial f}{\partial g} = 1$$

$$\frac{\partial f}{\partial b} = 1$$

- 위 값을 해석해보면 결국 1이라는 것은 1:1 영향 관계를 나타내고 5는 5배 만큼 영향을 미친다는 것이다.

즉 $b=3$ 에서 $b=4$ 로 증가시킬 경우 f 의 값은 -6이 된다. 1만큼 차이가 생긴다.

반대로 w 를 -3으로 변경 시키면 f 는 -12되므로 -5만큼 값이 증가하게 된다. 5배의 영향을 미치는 것을 알 수 있다.



- o *output -> hidden k, hidden k -> hidden k-1, hidden 2 -> hidden 1, hidden 1 -> input*의 과정을 거치면서 계속 weight가 update되는 것이다.

2. Hypothesis

3. Cost function

Cost function

$$cost(W) = \frac{1}{m} \sum c(H(x), y)$$

$$J(\theta) = \frac{1}{m} \sum_{i=1}^m \text{Cost}(h_\theta(x^{(i)}), y^{(i)})$$

$$\begin{aligned} \text{Cost}(h_\theta(x), y) &= -\log(h_\theta(x)) && \text{if } y = 1 \\ \text{Cost}(h_\theta(x), y) &= -\log(1 - h_\theta(x)) && \text{if } y = 0 \end{aligned}$$

$$C(H(x), y) = y \log(H(x)) - (1 - y) \log(1 - H(x))$$

4. Gradient descent algorithm