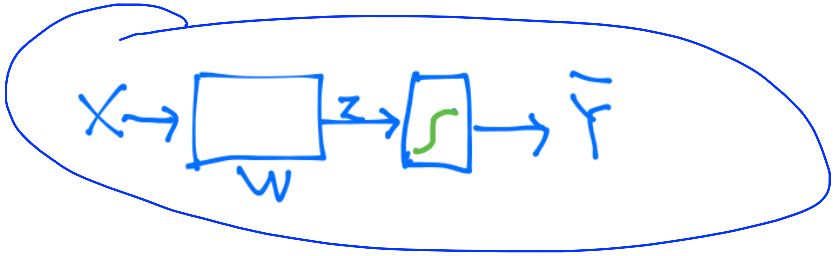
Lecture 9-1

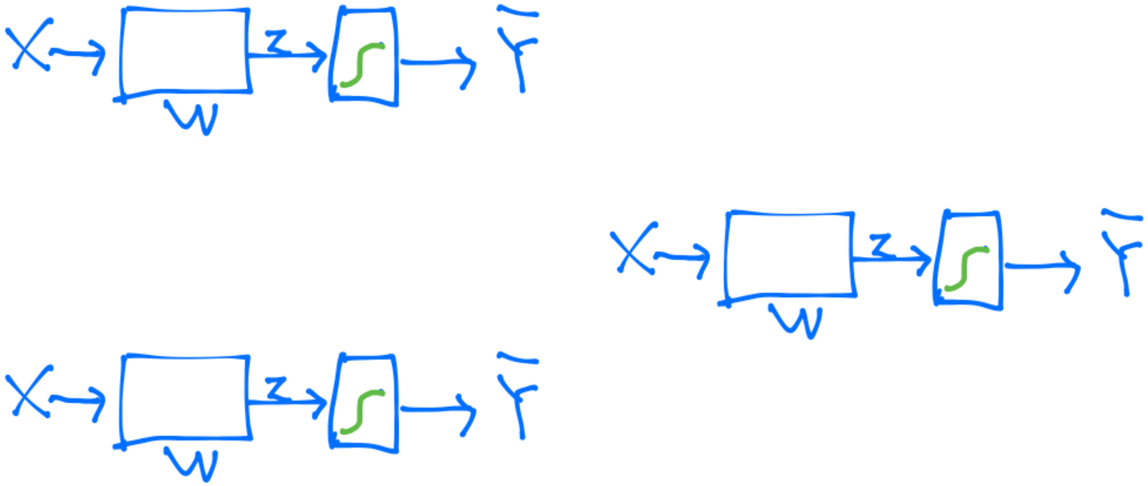
Neural Nets(NN) for XOR

Sung Kim <hunkim+mr@gmail.com>

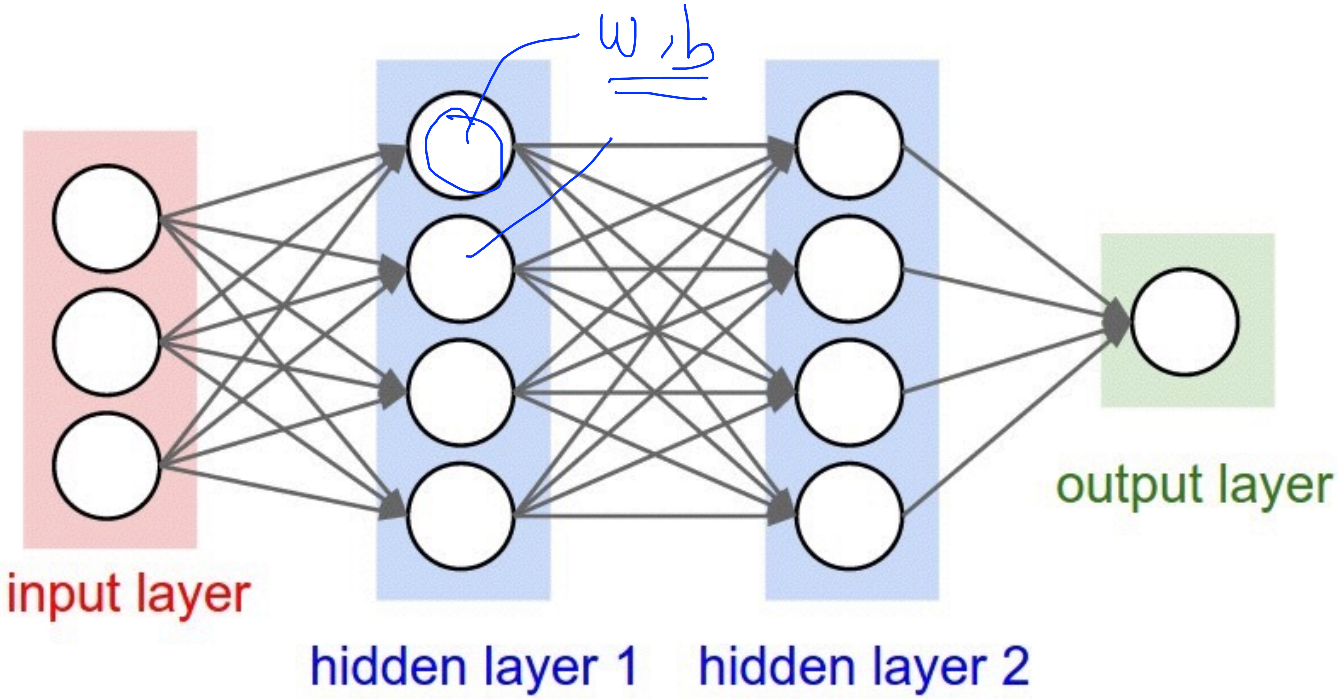
One logistic regression unit cannot separate XOR



Multiple logistic regression units



Neural Network (NN)
"No one on earth had found a viable way to train*"

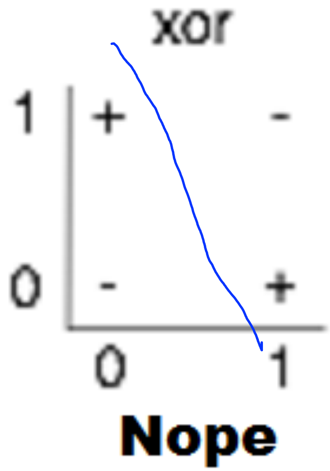


*Marvin Minsky

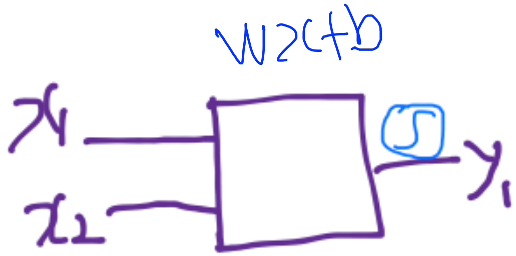
http://cs231n.github.io/convolutional-networks/

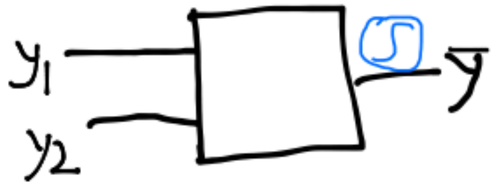
XOR using NN

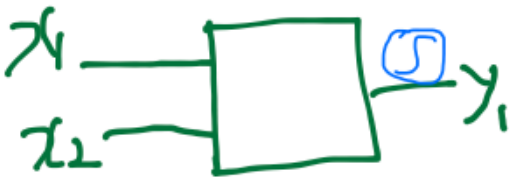
x_1	x_2	XOR
0	0	0 (−)
0	1	1 (+)
1	0	1 (+)
1	1	0 (−)



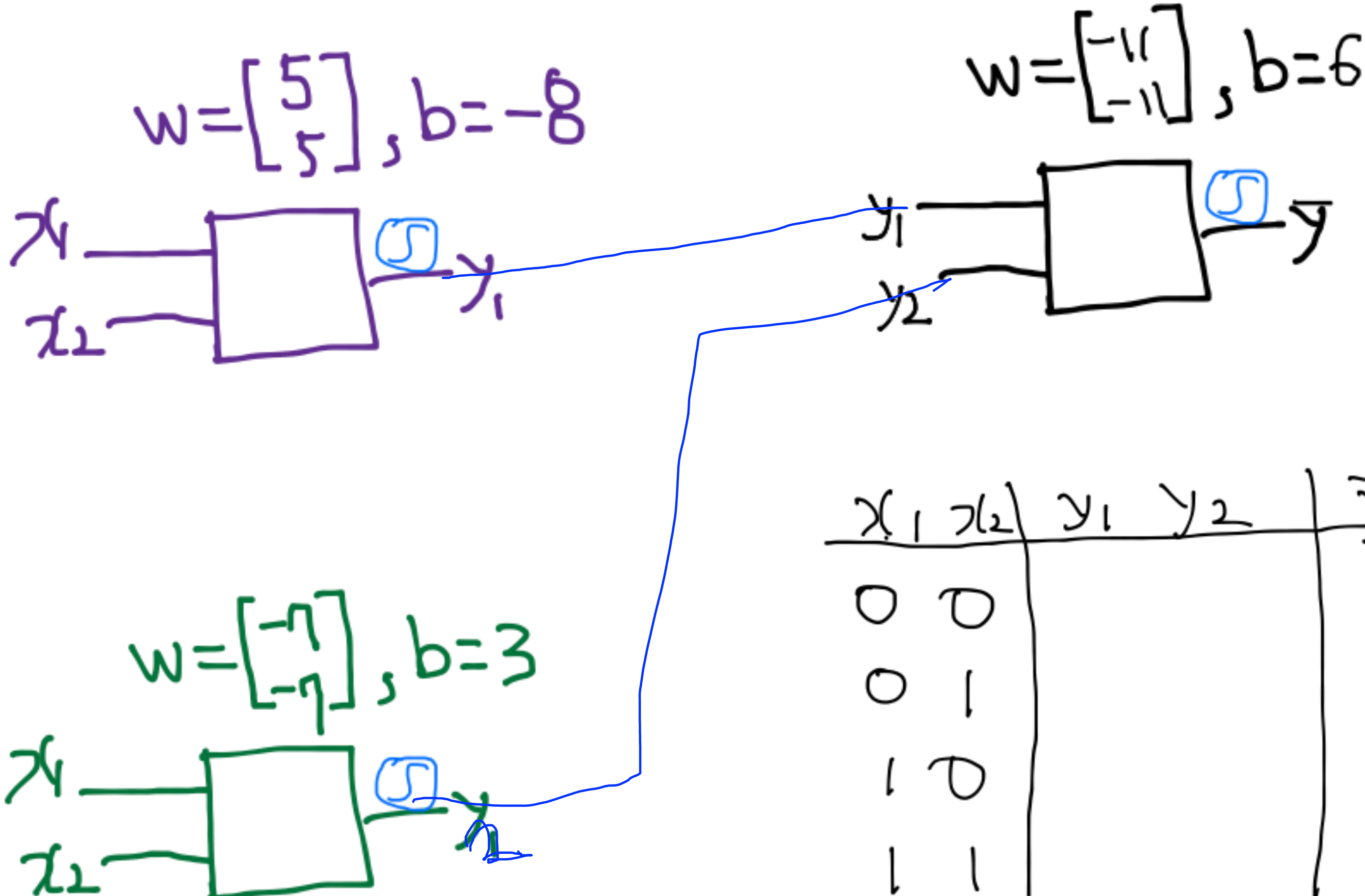
Neural Net



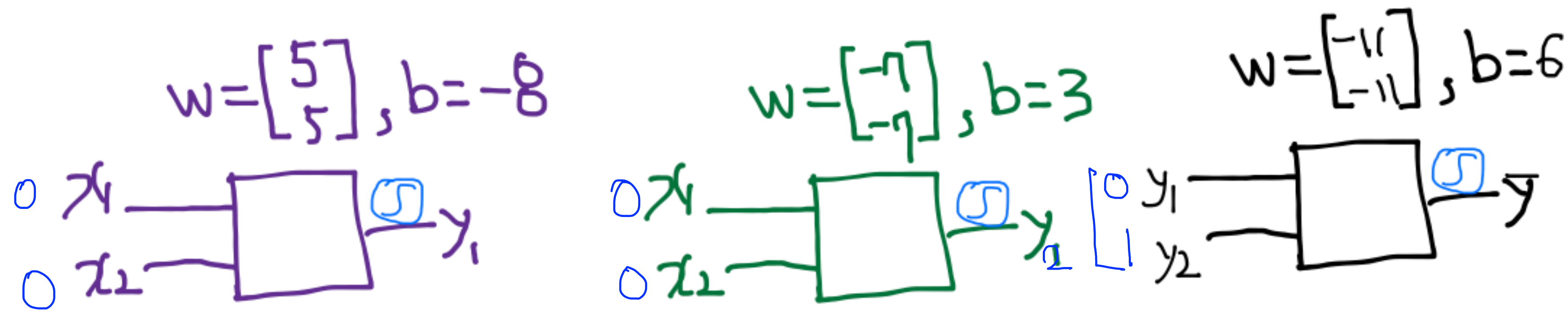


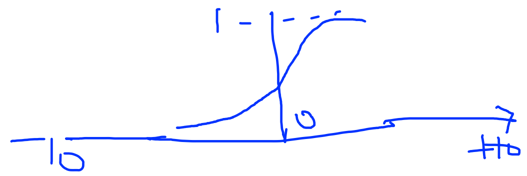


Neural Net



x_1	x_2	y_1	y_2	$\bar{y}$	XOR
0	0				0
0	1				1
1	0				1
1	1				0





$[0\ 0]\begin{bmatrix}5\\5\end{bmatrix}-8=-8,\quad y_1=S(-8)=0$

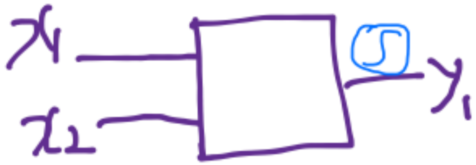
$[0\ 0]\begin{bmatrix}-7\\-7\end{bmatrix}+3=3,\quad y_2=S(3)=1$

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

$\bar{Y}=S(-5)=0$

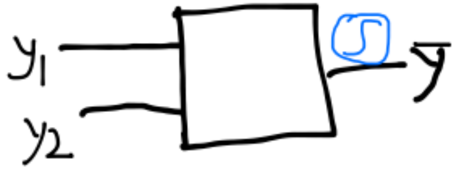
x_1	x_2	y_1	y_2	$\bar{y}$	XOR
0	0	0	1	0	= 0
0	1				1
1	0				1
1	1				0

$w = \begin{bmatrix} 5 \\ 5 \end{bmatrix}, b = -8$



$w = \begin{bmatrix} -7 \\ -7 \end{bmatrix}, b = 3$

$w = \begin{bmatrix} -11 \\ -11 \end{bmatrix}, b = 6$



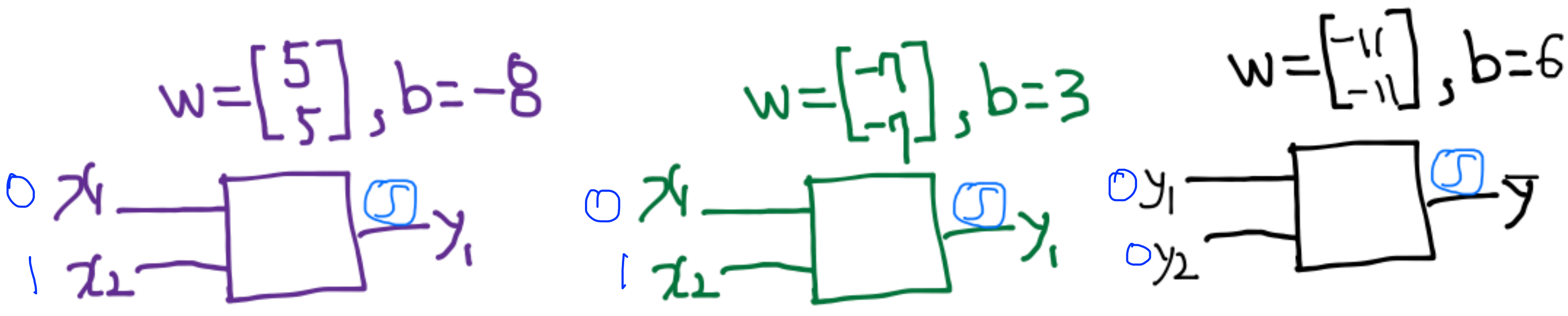
$[0\ 0]\begin{bmatrix} 5 \\ 5 \end{bmatrix} - 8 = 0 + 0 - 8 = -8,\ \text{sigmoid}(-8) = 0$

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

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

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

x_1	x_2	y_1	y_2	$\bar{y}$	XOR
0	0	0	1	0	0
0	1				1
1	0				1
1	1				0



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

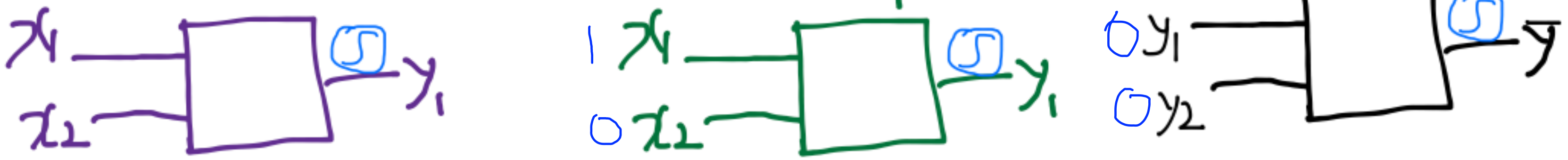
$[0\ 1]\begin{bmatrix}-7\\-7\end{bmatrix}+3 = 0+-7+3 = \underline{-4}$, sigmoid $(-4)=0$

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

sigmoid$(6)=1$

x_1	x_2	y_1	y_2	$\bar{y}$	XOR
0	0	0	1	0	0
0	1	0	0	1	1
1	0				1
1	1				0

$w=\begin{bmatrix}5\\5\end{bmatrix}, b=-8$ $\quad$ $w=\begin{bmatrix}-7\\-7\end{bmatrix}, b=3$ $\quad$ $w=\begin{bmatrix}-11\\-11\end{bmatrix}, b=6$



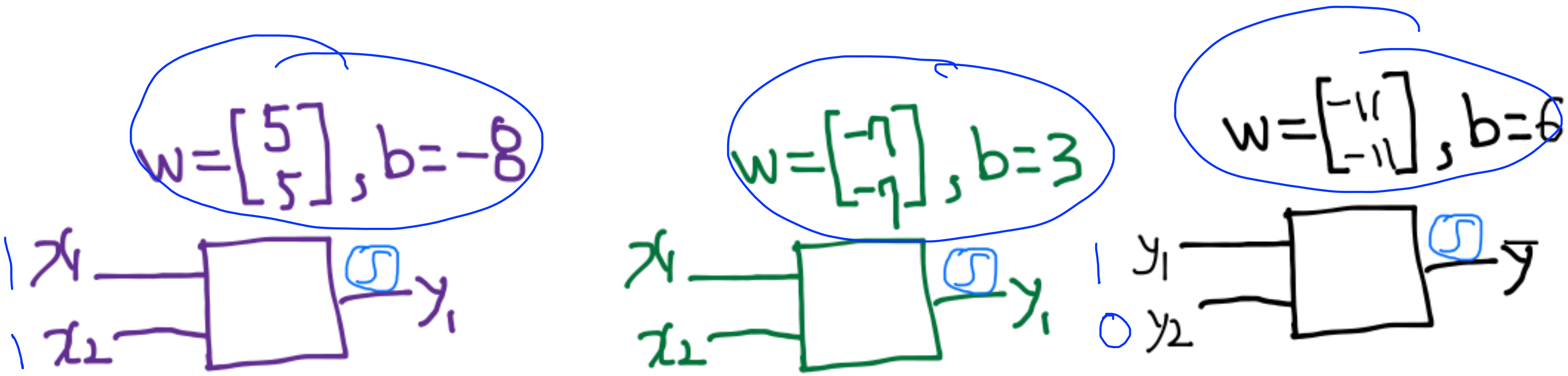
$[1\ 0]\begin{bmatrix}5\\5\end{bmatrix}-8 = 5+0-8 = \underline{\underline{-3}},\ \text{sigmoid}(-3) = \underline{\underline{0}}$

$[1\ 0]\begin{bmatrix}-7\\-7\end{bmatrix}+3 = -7+0+3 = \underline{\underline{-4}},\ \text{sigmoid}(-4) = \underline{\underline{0}}$

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

$\text{sigmoid}(6) = 1$

x_1	x_2	y_1	y_2	$\hat{y}$	XOR	
0	0	0	1	0	0	✓
0	1	0	0	1	1	✓
✓ 1	0	0	0	1	1	✓
1	1				0	?



$[1 \ \ 1]\begin{bmatrix}5\\5\end{bmatrix} - 8 = 5+5-8 = 2$, $\mathrm{sigmoid}(2) = 1$

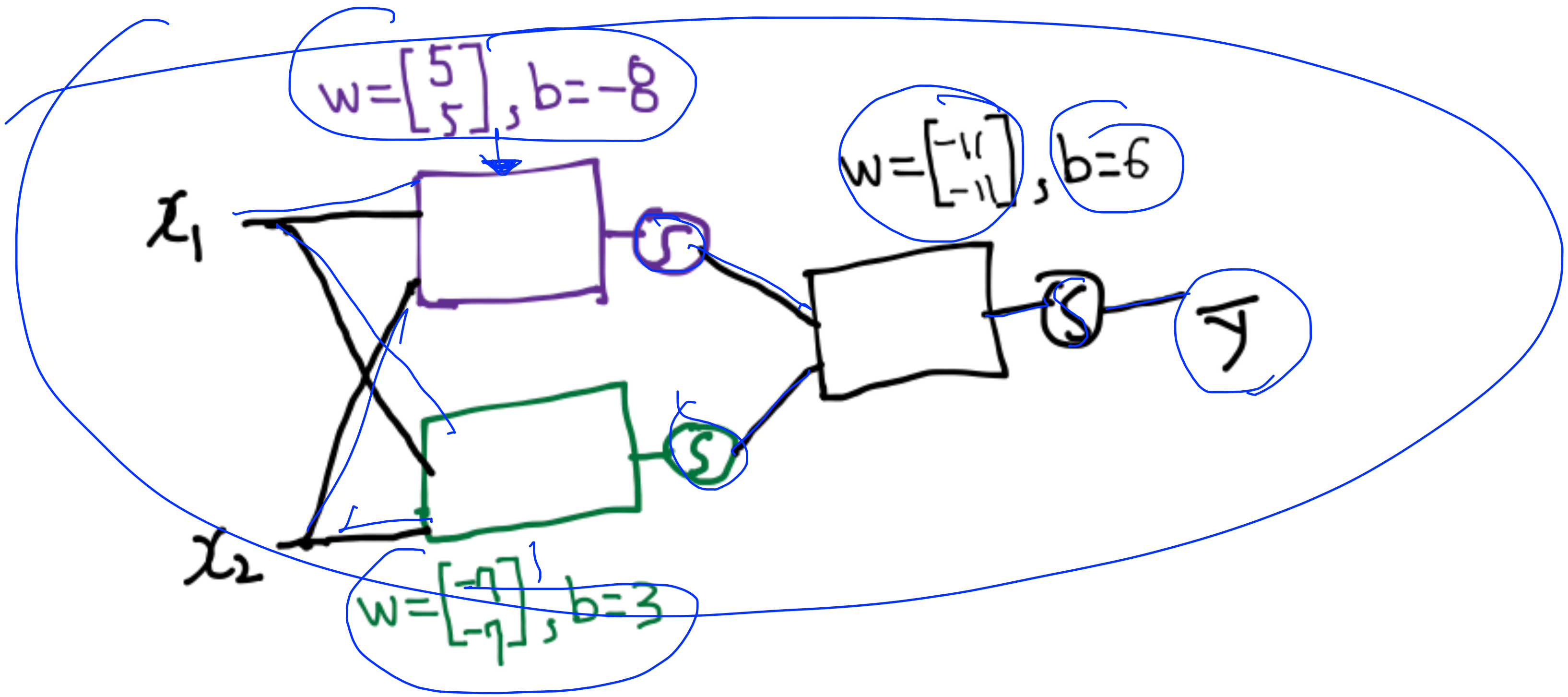
$[1 \ \ 1]\begin{bmatrix}-7\\-7\end{bmatrix} + 3 = -7+-7+3 = -11$, $\mathrm{sigmoid}(-11) = 0$

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

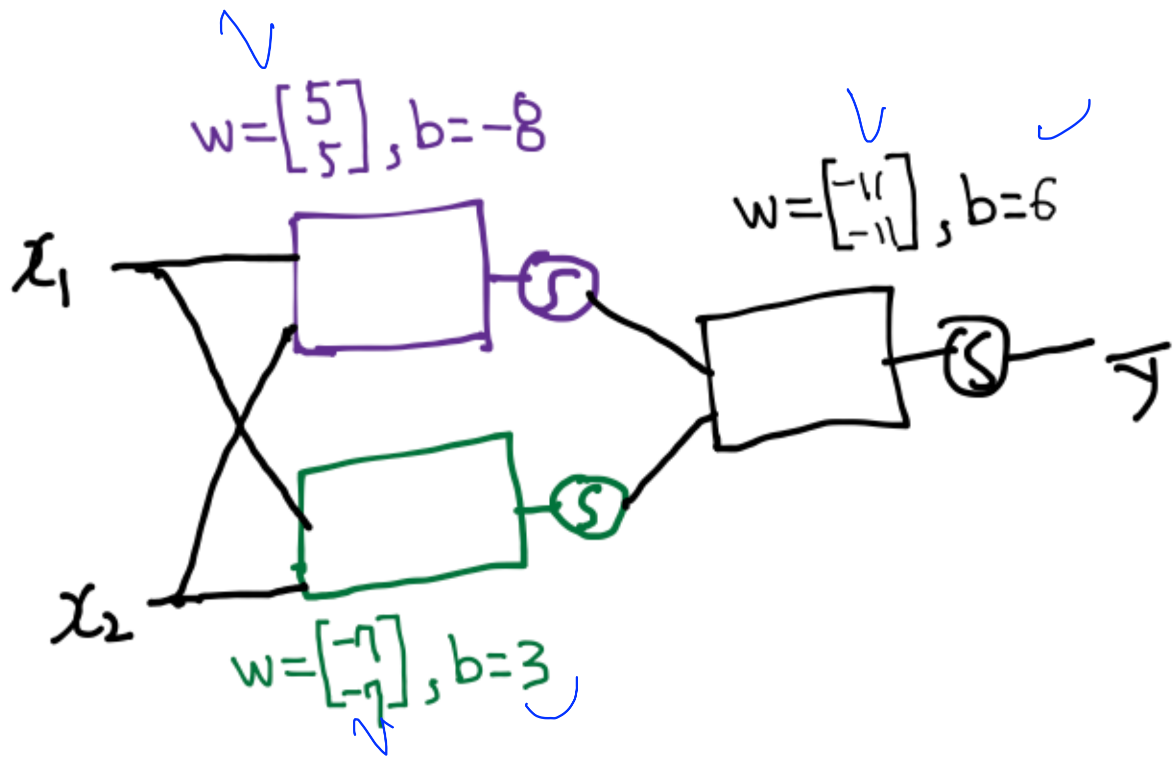
$\mathrm{sigmoid}(-5) = 0$

x_1	x_2	y_1	y_2	$\bar{y}$	XOR	
0	0	0	1	0	0	✓
0	1	0	0	1	1	✓
1	0	0	0	1	1	✓
1	1	1	0	0	0	✓

Forward propagation

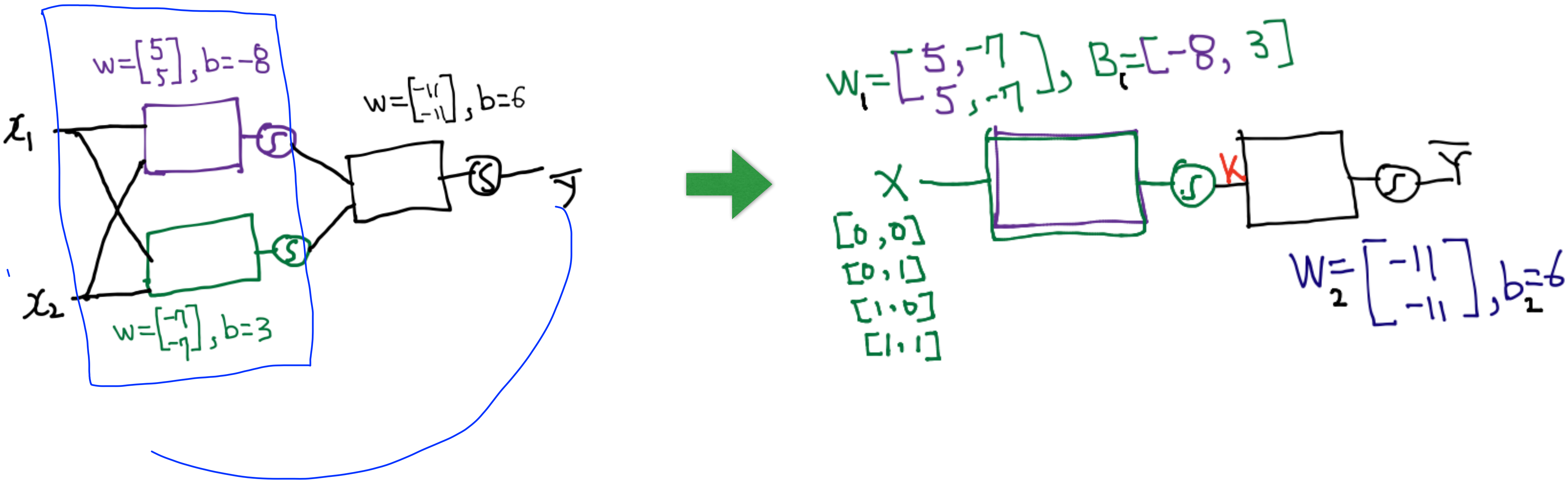


Forward propagation



Can you find another W and b for the XOR?

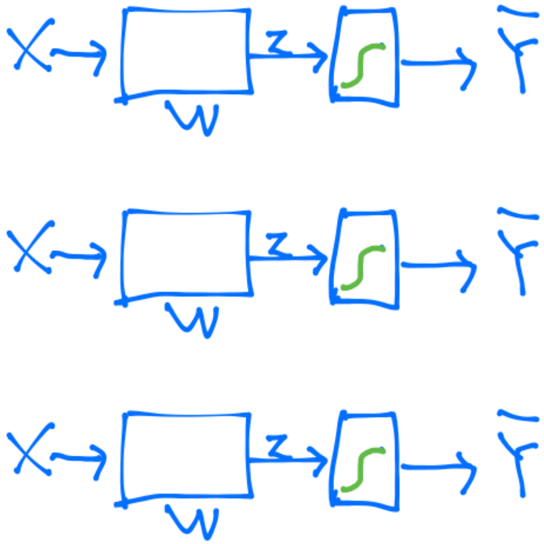
NN



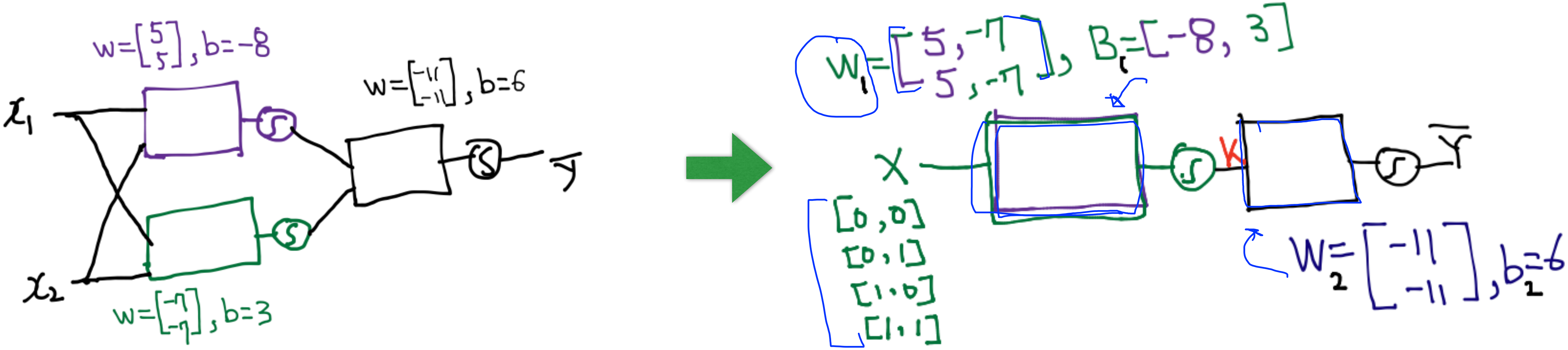
w=[5 5], b=-8
w=[-11 -11], b=6
x_1
x_2
w=[-7 -7], b=3
Y
W_1=[5, -7
5, -7], B_1=[-8, 3]
X
[0,0]
[0,1]
[1,0]
[1,1]
K
Y
W_2=[-11
-11], b_2=6

Recap: Lec 6-1 Multinomial classification

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



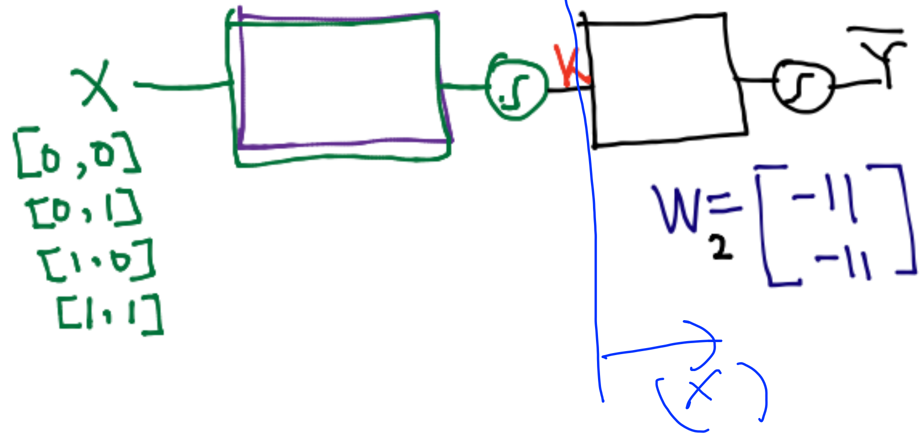
NN



How can we learn W, and b from trading data?

NN

$W_1 = \begin{bmatrix} 5, -7 \\ 5, -7 \end{bmatrix}$, $B_1 = [-8, 3]$



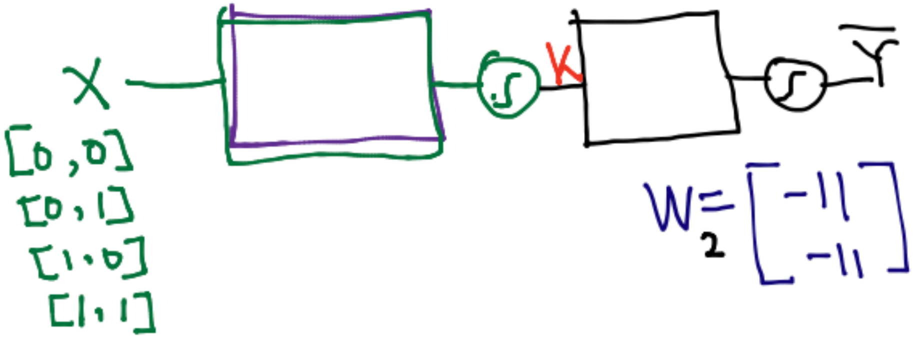
$W_2 = \begin{bmatrix} -11 \\ -11 \end{bmatrix}$, $b_2 = 6$

$K(X) = \mathrm{sigmoid}(XW_1 + B_1)$

$\bar{Y} = H(X) = \mathrm{sigmoid}(K(X)W_2 + b_2)$

NN

$W_1 = \begin{bmatrix} 5, -7 \\ 5, -7 \end{bmatrix}$, $B_1 = [-8, 3]$



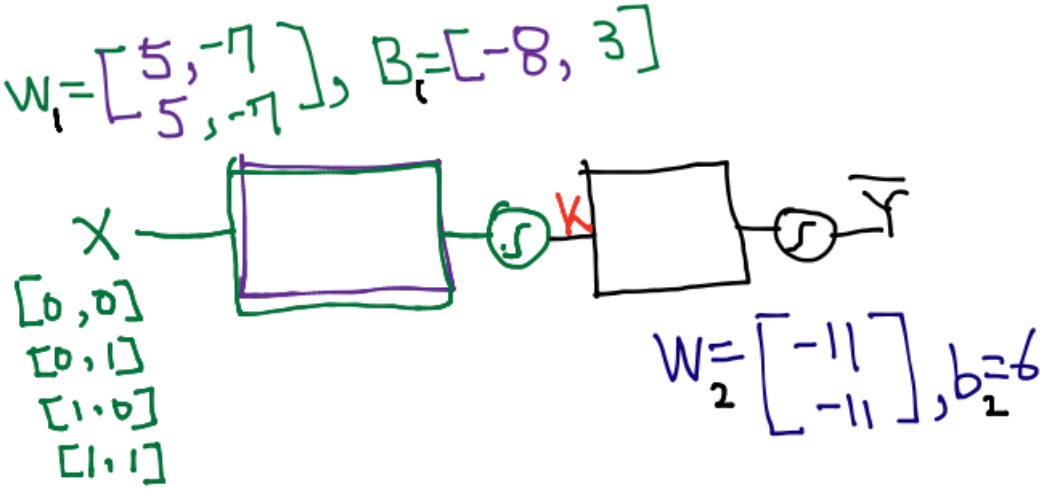
$W_2 = \begin{bmatrix} -11 \\ -11 \end{bmatrix}$, $b_2 = 6$

$K(X) = \text{sigmoid}(X W_1 + B_1)$

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

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

NN



$$K(x) = \mathrm{sigmoid}(XW_1 + B_1)$$

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

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

How can we learn W1, W2, B1, b2 from training data?

Next

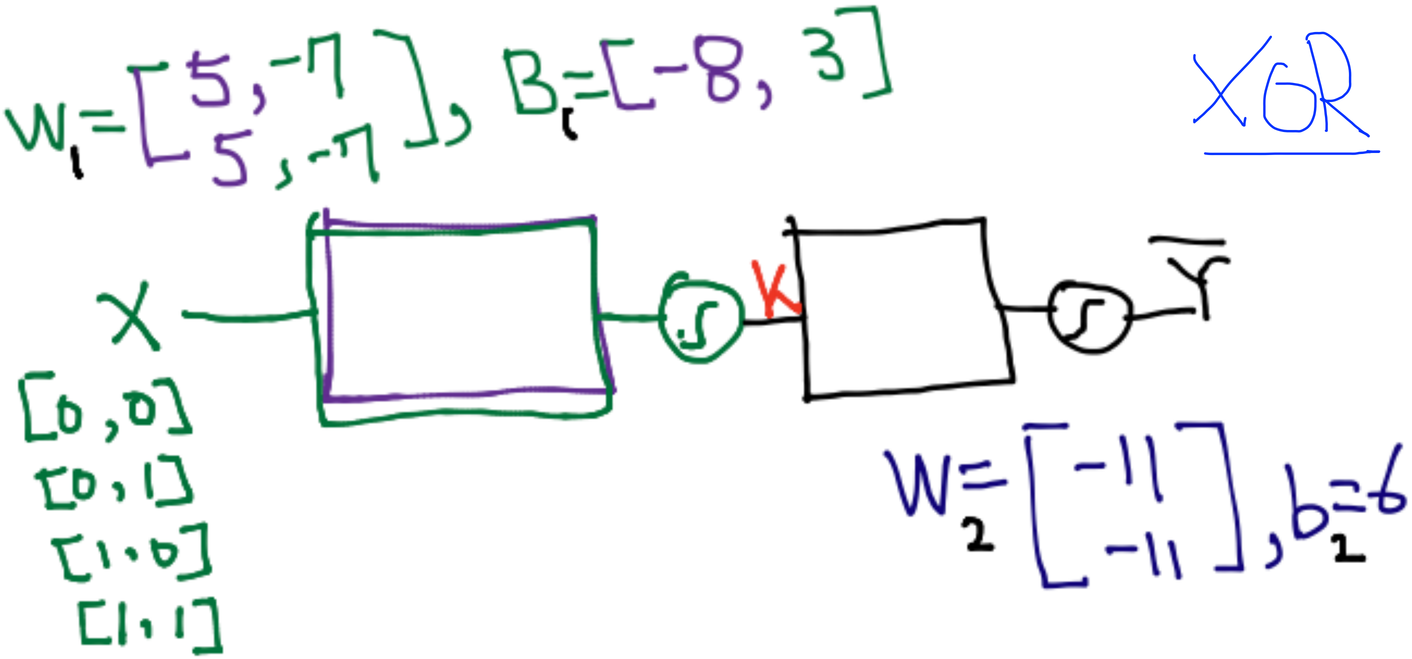
Backpropagation

Lecture 9-2

Backpropagation

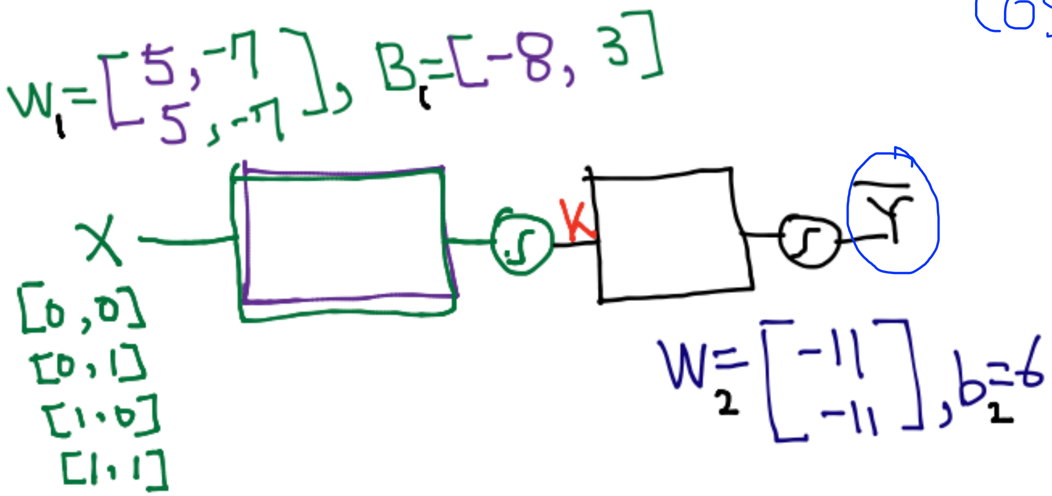
Sung Kim <hunkim+mr@gmail.com>

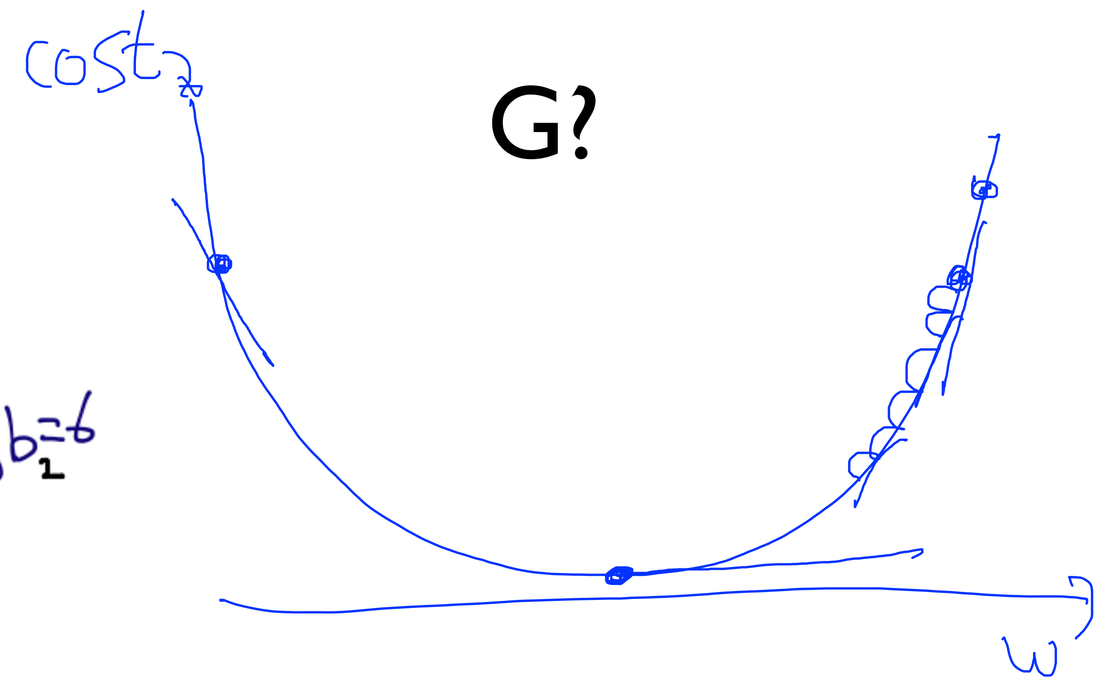
Neural Network (NN)



How can we learn W1, W2, B1, b2 from training data?

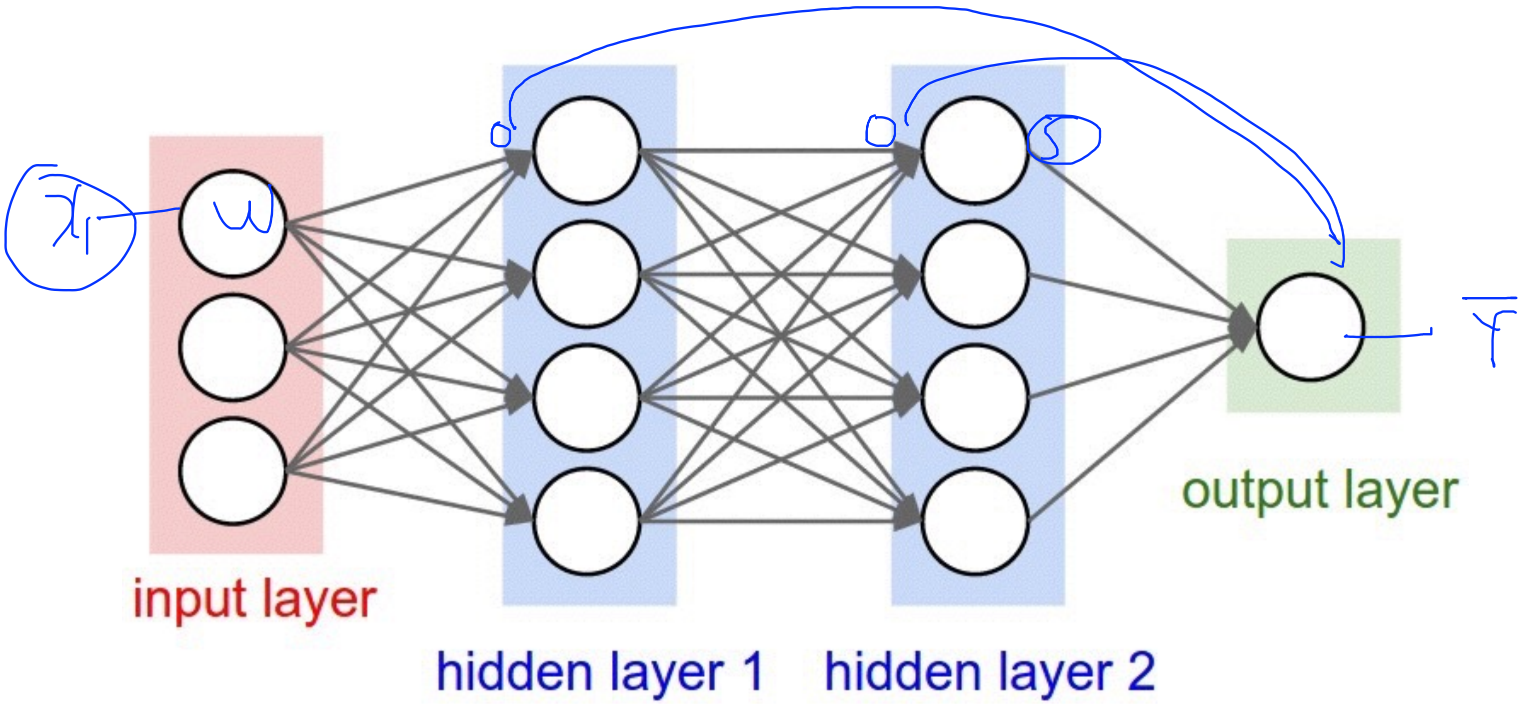
How can we learn W1, W2, B1, b2 from training data?





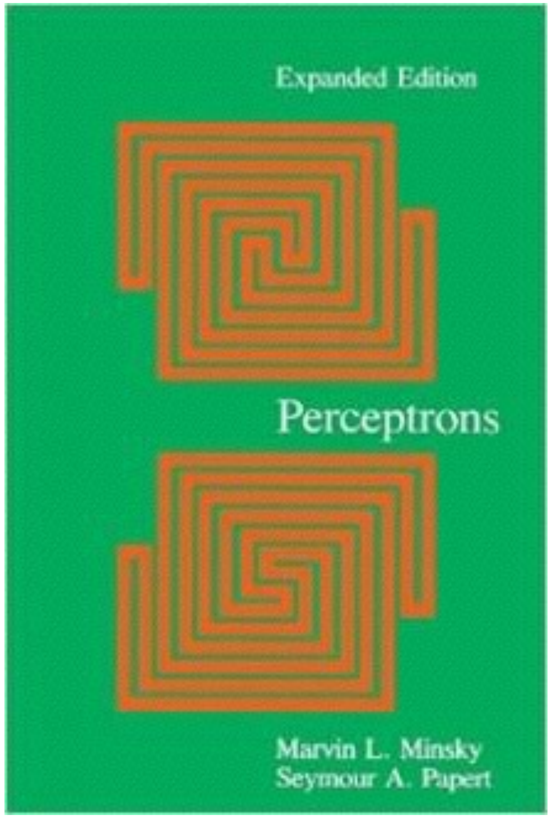
G?

Derivation



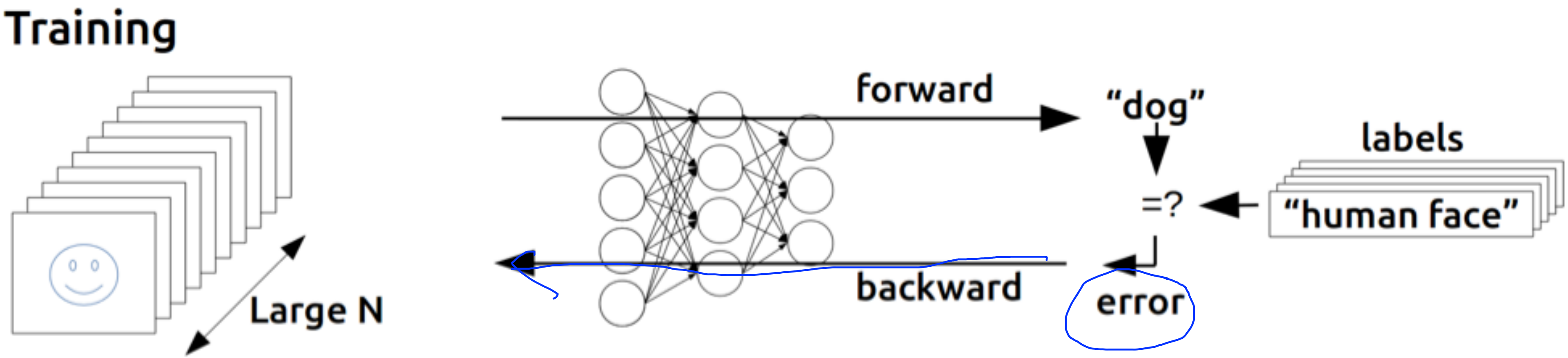
Perceptrons (1969)

by Marvin Minsky, founder of the MIT AI Lab



- We need to use MLP, multilayer perceptrons (multilayer neural nets)
- No one on earth had found a viable way to train MLPs good enough to learn such simple functions.

Backpropagation
(1974, 1982 by Paul Werbos, 1986 by Hinton)



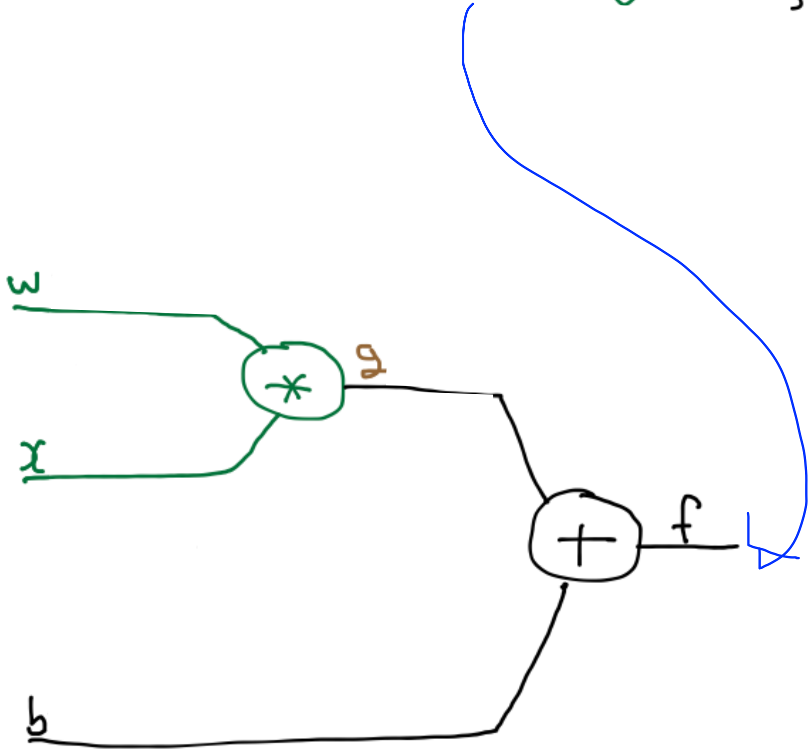
https://devblogs.nvidia.com/parallelforall/inference-next-step-gpu-accelerated-deep-learning/

Back propagation (chain rule)

$f = \boxed{\underbrace{wx}_{g} + b}$, $g = wx$; $f = g + b$

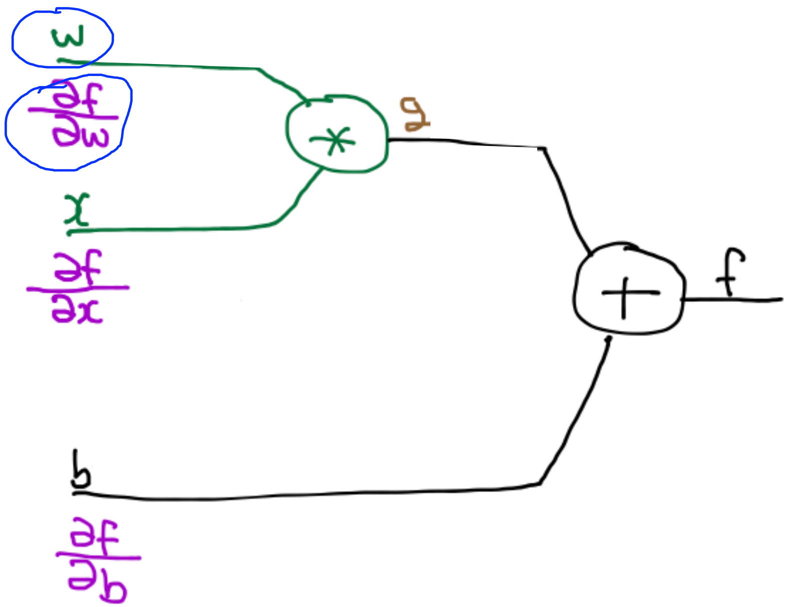
Back propagation (chain rule)

$f = wx + b, \quad g = wx \; ; \quad f = g + b$



Back propagation (chain rule)

$f = wx + b, \; g = wx, \; f = g + b$



Basic derivative

$$\frac{d}{dx}f(x) = \lim_{\Delta x \to 0} \frac{f(x + \Delta x) - f(x)}{\Delta x}$$

$f(x) = 3$

$f(x) = x$

$f(x) = 2x$

https://ko.wikipedia.org/wiki/%EB%AF%B8%EB%B6%84

Partial derivative: consider other variables as constants

$f(x) = 2x$

$f(x, y) = xy, \frac{\partial f}{\partial x}$

$f(x, y) = xy, \frac{\partial f}{\partial y}$

https://ko.wikipedia.org/wiki/%EB%AF%B8%EB%B6%84

Partial derivative: consider other variables as constants

$f(x) = 3$

$f(x) = 2x \quad f(x) = x + x$

$f(x) = x + 3$

$f(x, y) = x + y, \frac{\partial f}{\partial x}$

$f(x, y) = x + y, \frac{\partial f}{\partial y}$

chain rule

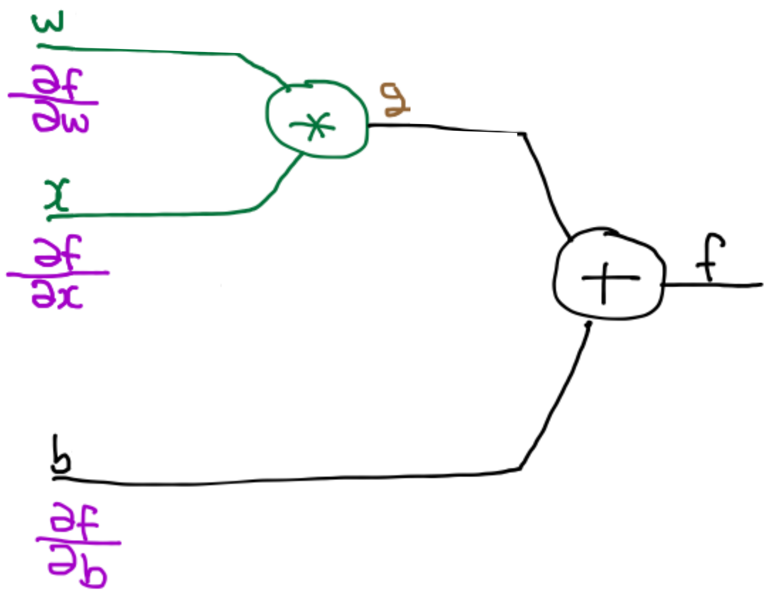
$f(g(x))$

$$\boxed{\frac{\partial f}{\partial x} = \frac{\partial f}{\partial g} \cdot \frac{\partial g}{\partial x}}$$

https://ko.wikipedia.org/wiki/%EB%AF%B8%EB%B6%84

Back propagation (chain rule)

$f = wx + b$, $g = wx$, $f = g + b$



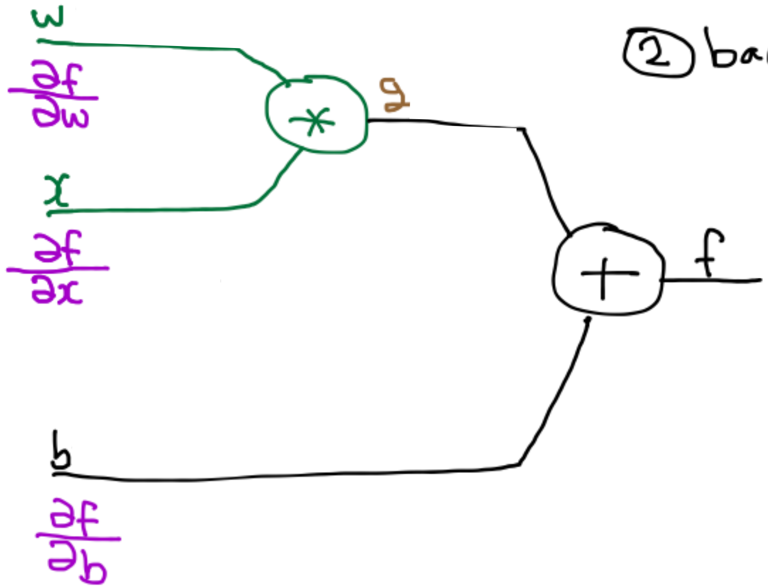
http://cs231n.stanford.edu/

Back propagation (chain rule)

$f = wx + b$, $g = wx$, $f = g + b$

① forward ($w = -2$, $x = 5$, $b = 3$)

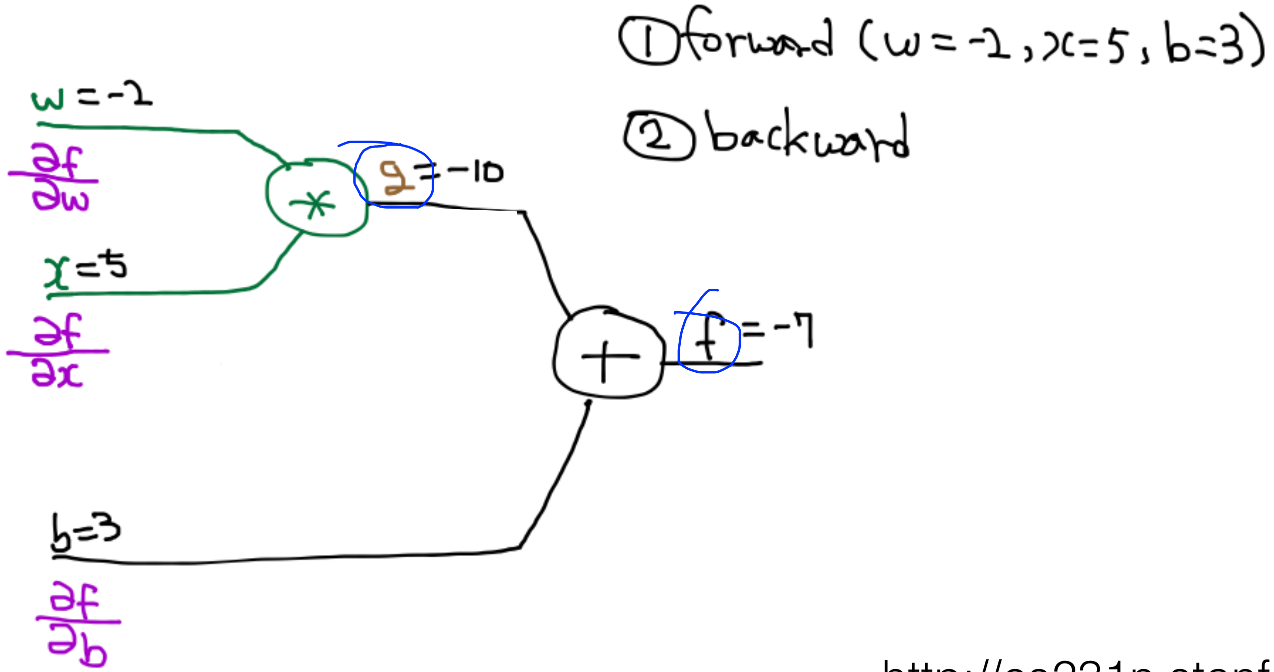
② backward



http://cs231n.stanford.edu/

Back propagation (chain rule)

$f = wx + b, \quad g = wx, \quad f = g + b$



http://cs231n.stanford.edu/

Back propagation (chain rule)

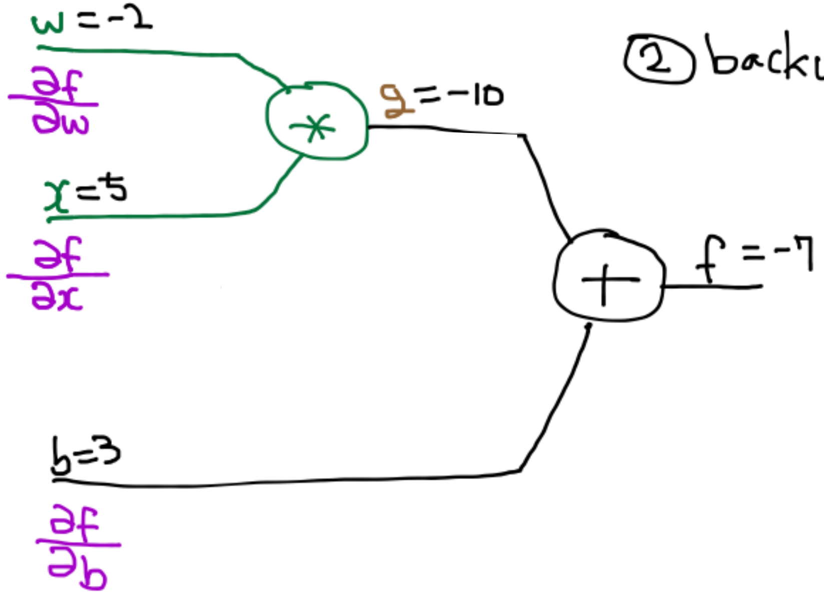
$f = wx + b$, $g = wx$, $f = g + b$

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

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

① forward ($w=-2$, $x=5$, $b=3$)

② backward



http://cs231n.stanford.edu/

Back propagation (chain rule)

$f = wx + b$, $g = wx$, $f = g + b$, $\frac{\partial f}{\partial g} = 1$, $\frac{\partial f}{\partial b} = 1$

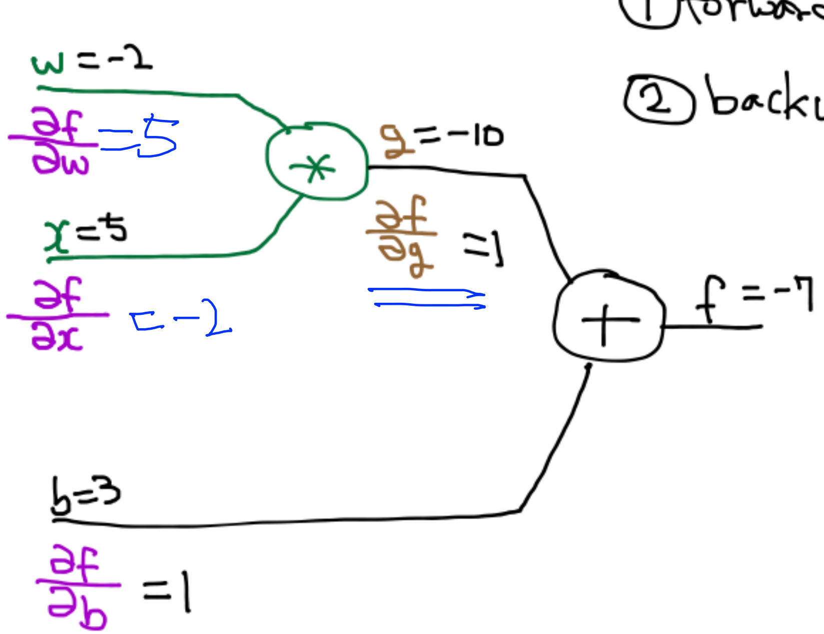
$\frac{\partial g}{\partial w} = x$, $\frac{\partial g}{\partial x} = w$

① forward ($w=-2$, $x=5$, $b=3$)

② backward

$$\frac{\partial f}{\partial w} = \frac{\partial f}{\partial g} \cdot \frac{\partial g}{\partial w} = 1 * 5 = 5$$

$$\frac{\partial f}{\partial x} = \underbrace{\frac{\partial f}{\partial g}}_{=1} * \underbrace{\frac{\partial g}{\partial x}}_{=-2} = -2$$



http://cs231n.stanford.edu/

Back propagation (chain rule)

$f = wx + b, \quad g = wx, \quad f = g + b$

$\frac{\partial f}{\partial g} = 1, \quad \frac{\partial f}{\partial b} = 1$

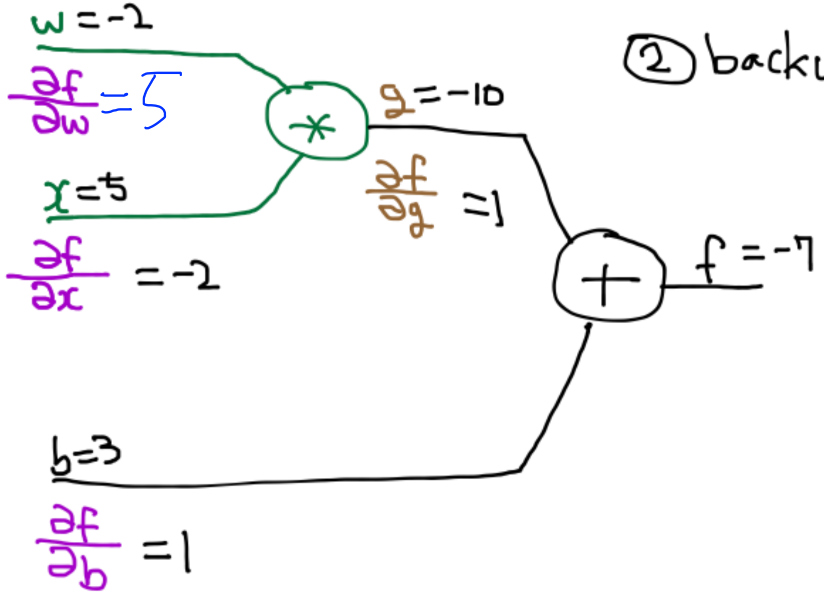
$\frac{\partial g}{\partial w} = x, \quad \frac{\partial g}{\partial x} = w$

① forward $(w=-2, x=5, b=3)$

② backward

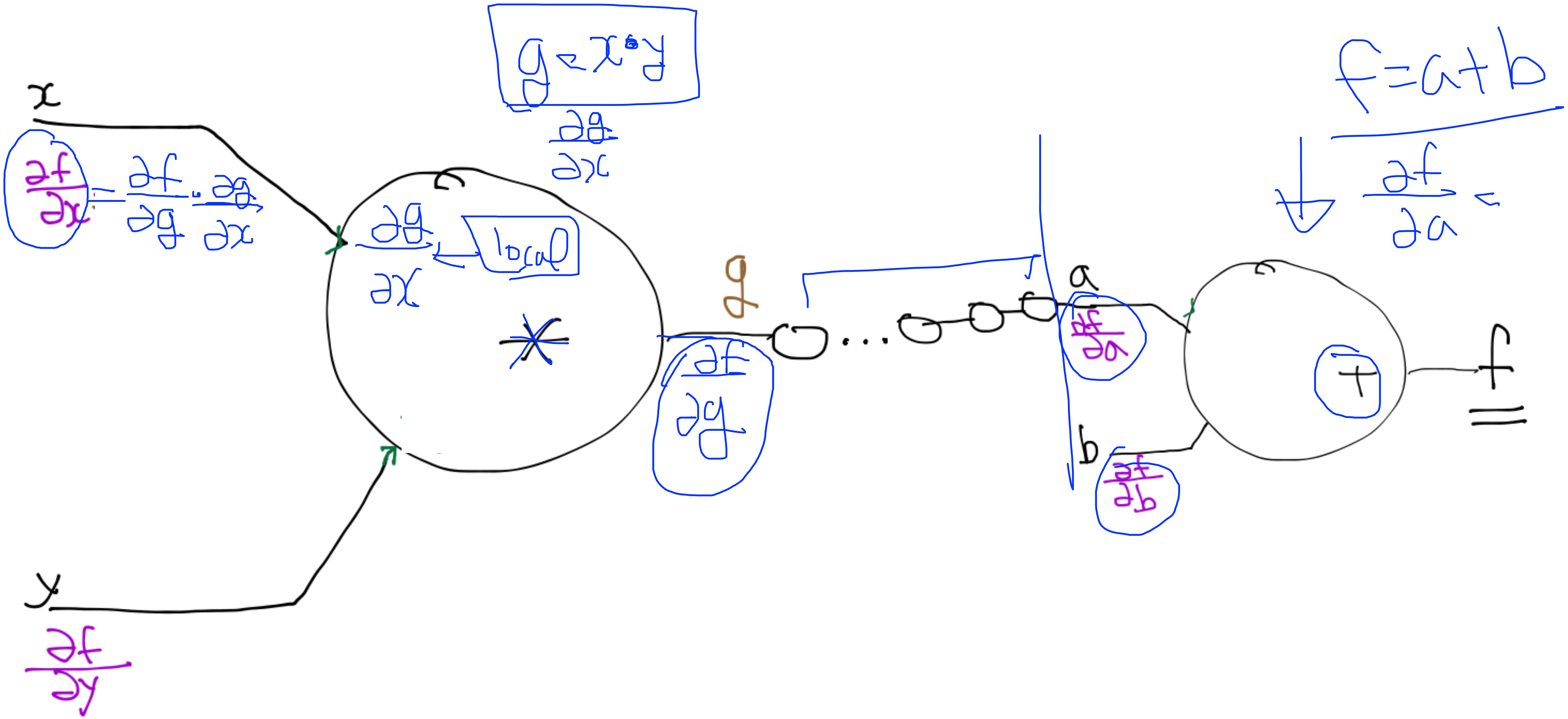
$\frac{\partial f}{\partial x} = \frac{\partial f}{\partial g}\frac{\partial g}{\partial x} = 1 * w = -2$

$\frac{\partial f}{\partial w} = \frac{\partial f}{\partial g}\frac{\partial g}{\partial x} = 1 * x = 5$



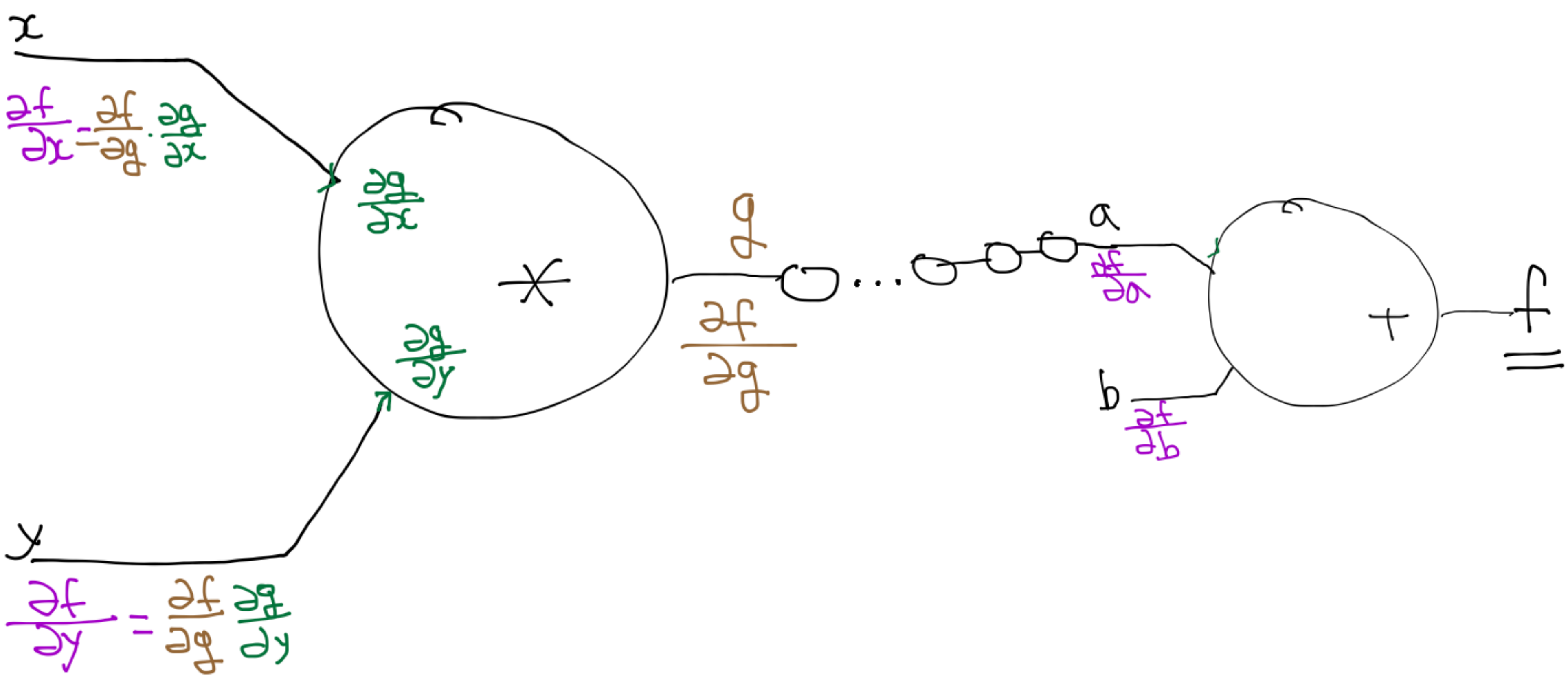
http://cs231n.stanford.edu/

Back propagation (chain rule)



http://cs231n.stanford.edu/

Back propagation (chain rule)



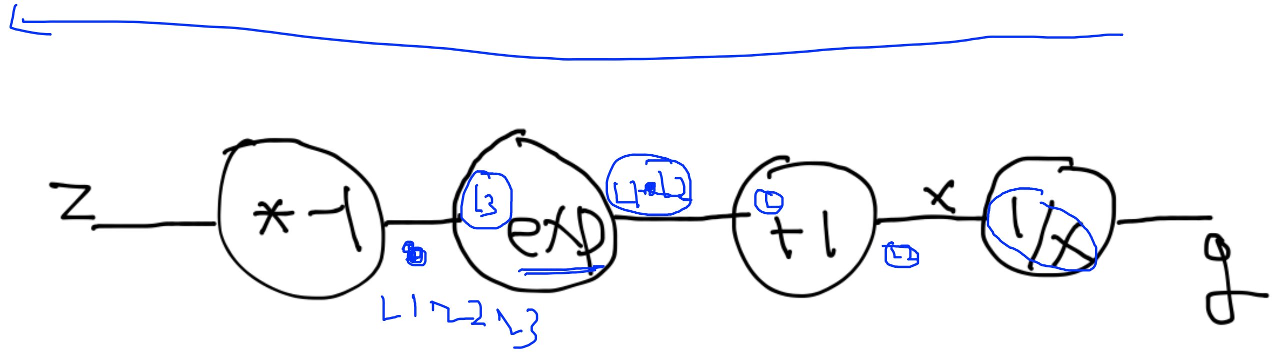
http://cs231n.stanford.edu/

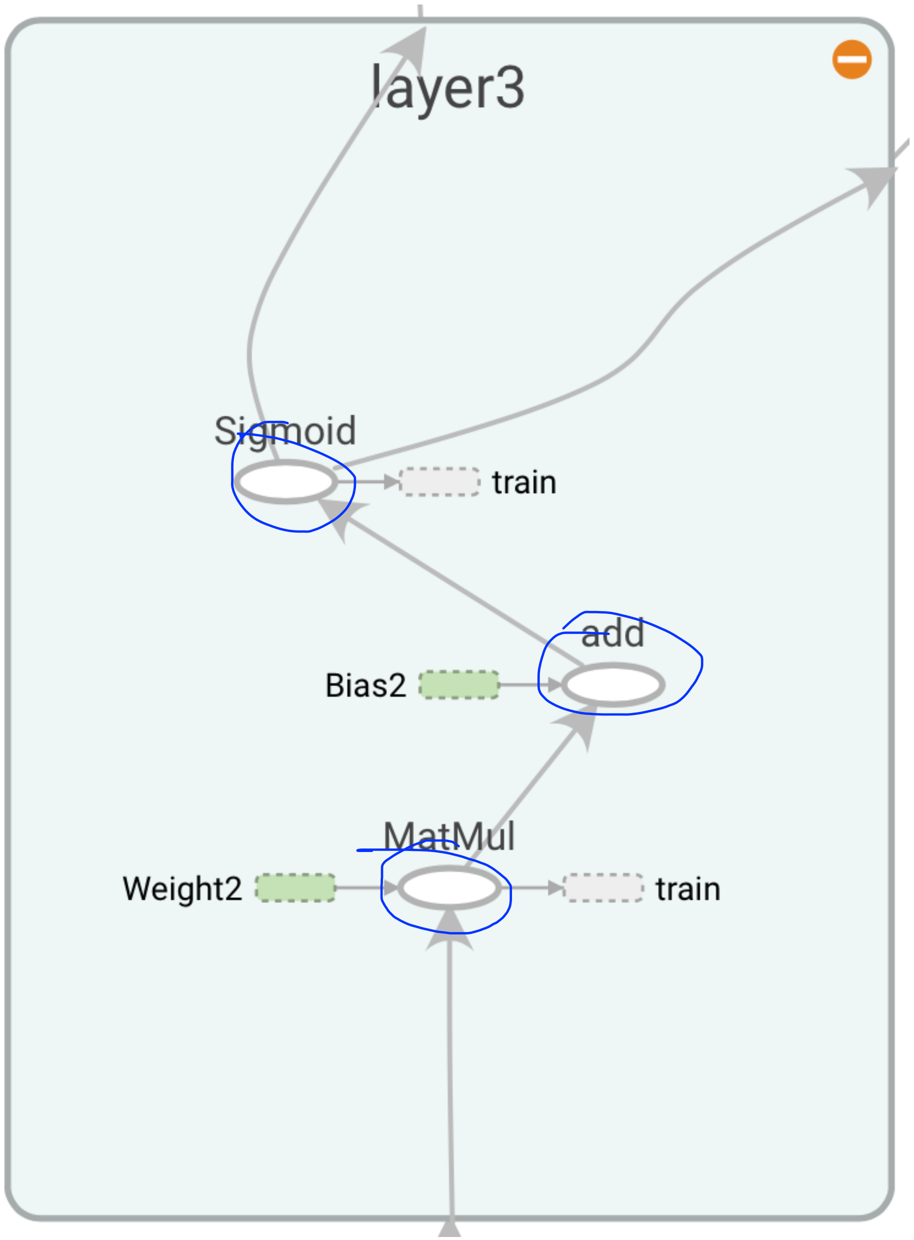
Sigmoid

$$g(z) = \frac{1}{1+e^{-z}} \qquad \frac{\partial g}{\partial z}$$

Sigmoid

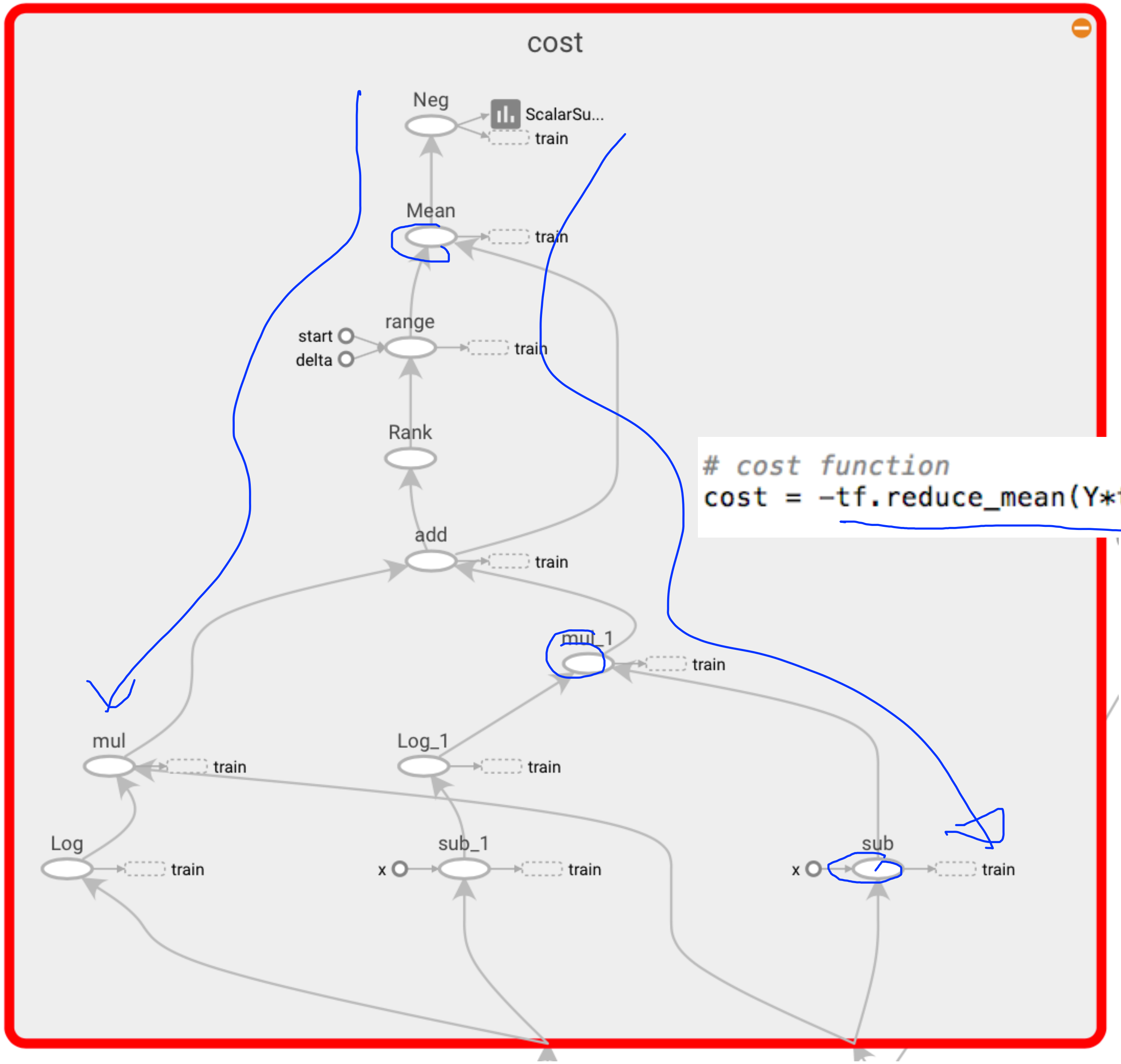
$$g(z) = \frac{1}{1+e^{-z}}$$





Back propagation in TensorFlow TensorBoard

```
hypothesis = tf.sigmoid(tf.matmul(L2, W2) + b2)
```

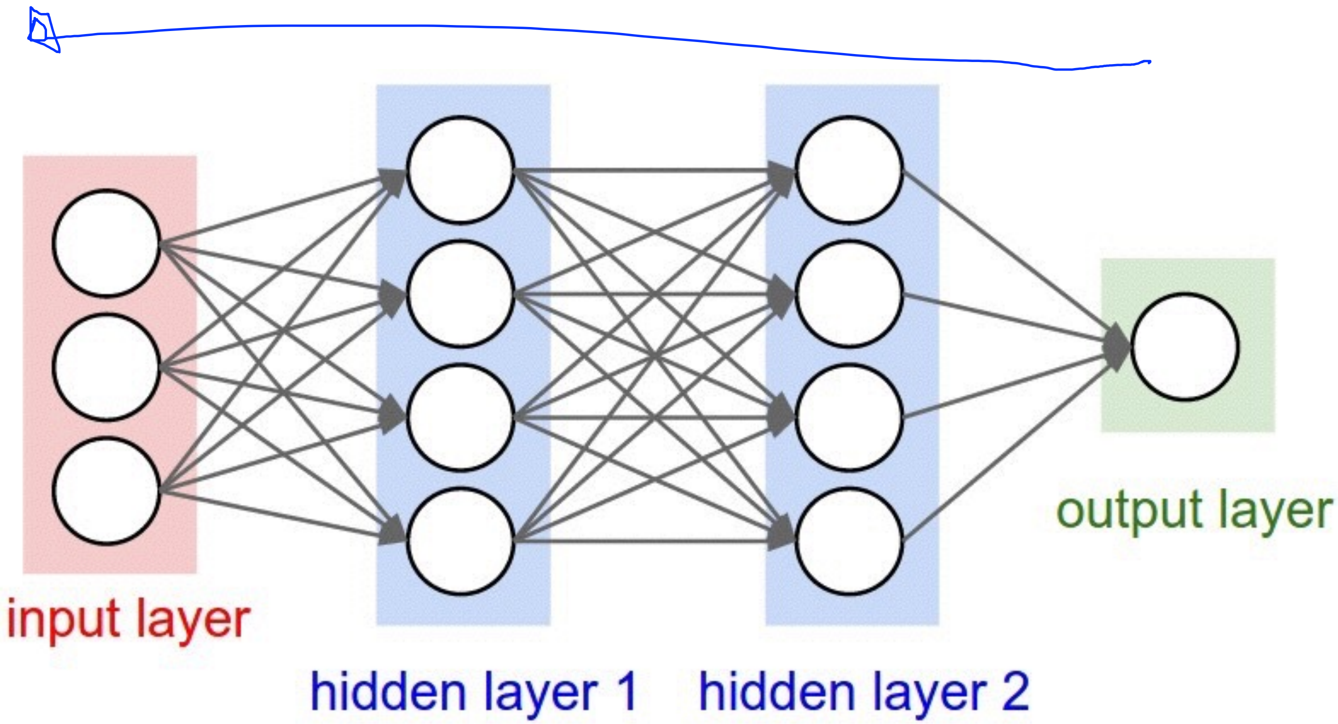


Back propagation in TensorFlow

TensorBoard

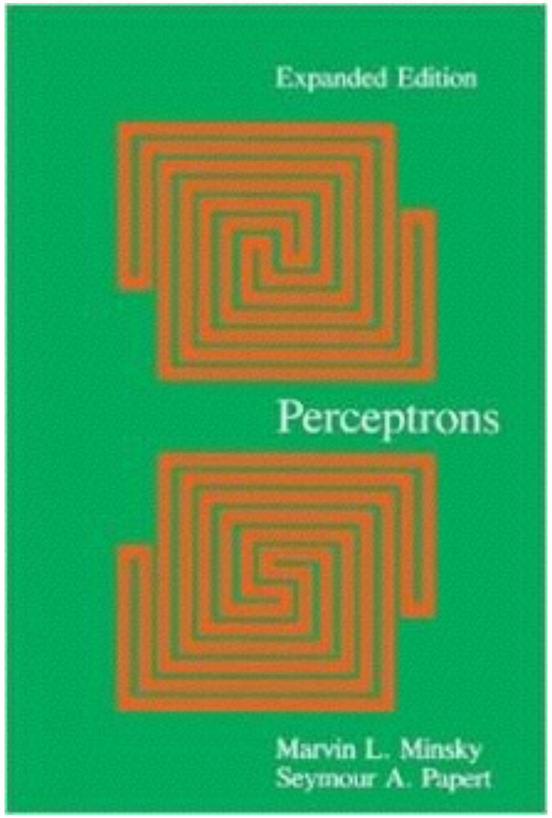
```
# cost function
cost = -tf.reduce_mean(Y*tf.log(hypothesis) + (1-Y)*tf.log(1-hypothesis))
```

Back propagation



Perceptrons (1969)

by Marvin Minsky, founder of the MIT AI Lab



- We need to use MLP, multilayer perceptrons (multilayer neural nets)
- No one on earth had found a viable way to train MLPs good enough to learn such simple functions.

Next

ReLU