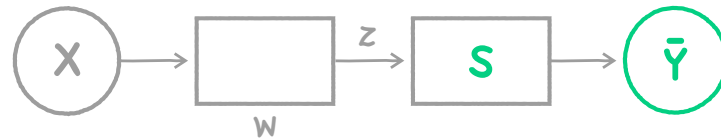


LECTURE 9-1

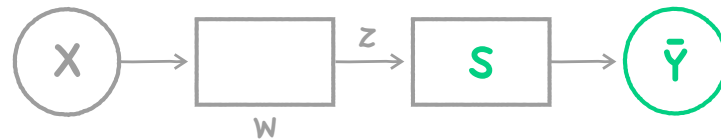
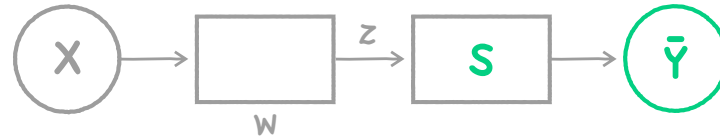
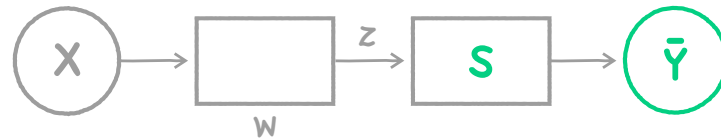
NEURAL NETS(NN) FOR XOR

Sung Kim <hunkim+ml@gmail.com>
<http://hunkim.github.io/ml>

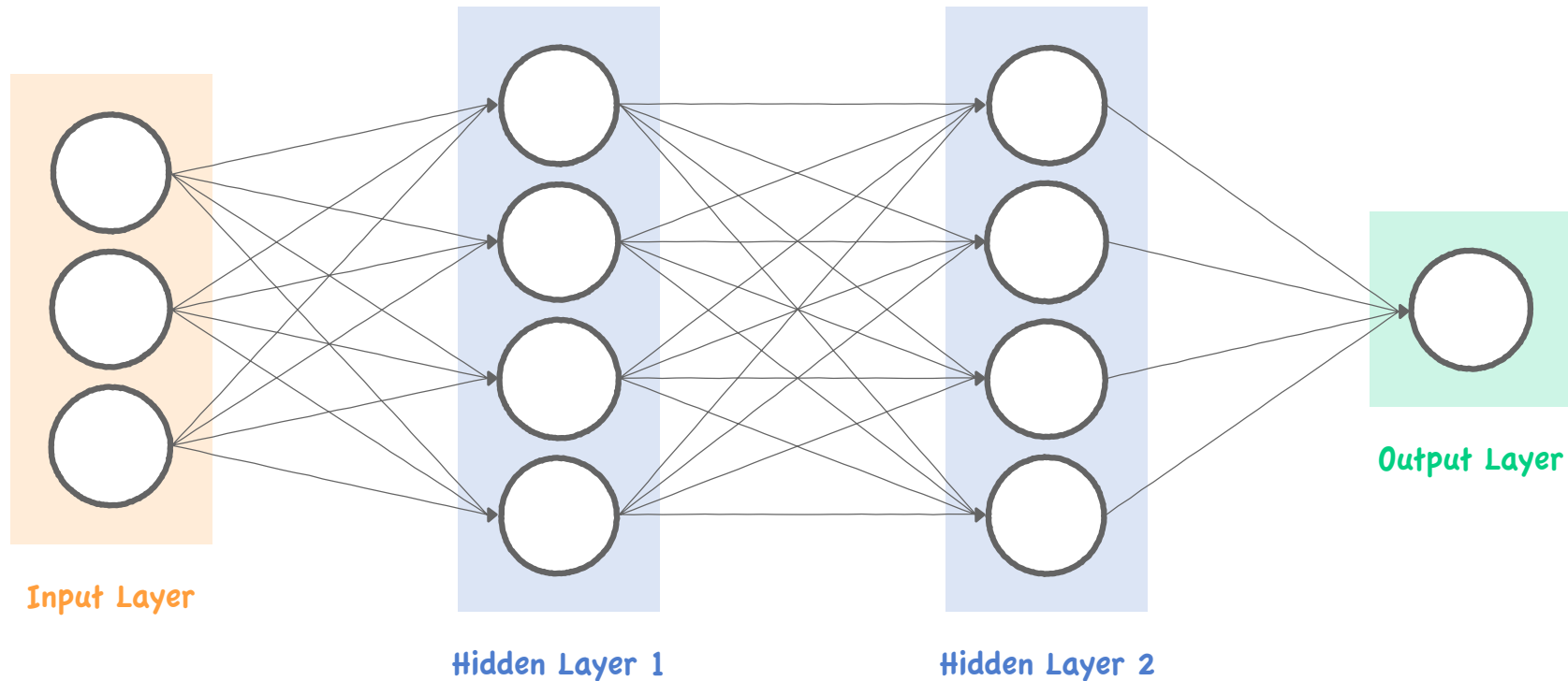
One Logistic Regression Unit Cannot Separate XOR



Multiple Logistic Regression Units



Neural Network(NN)



“No one on earth had found a viable way to train”

<Marvin Minsky>

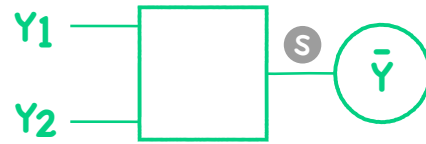
<http://cs231n.github.io/convolutional-networks>

XOR Using NN

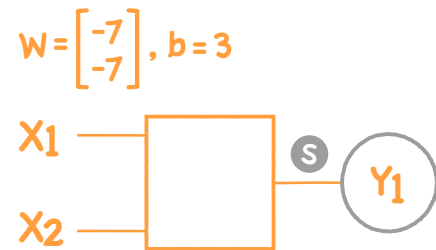
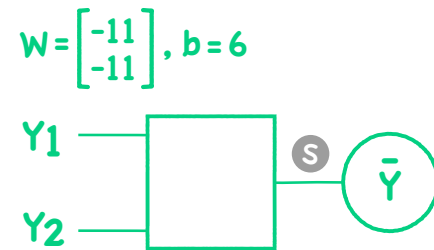
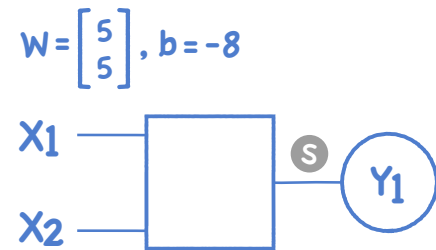
x_1	x_2	XOR
0	0	0(-)
0	1	1(+)
1	0	1(+)
1	1	0(-)



Neural Net

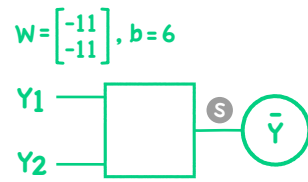
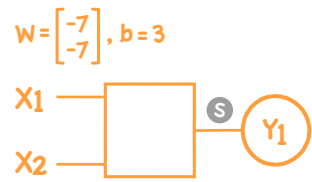
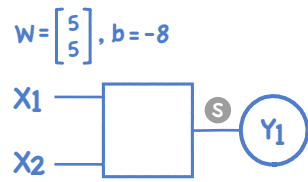


Neural Net



X_1	X_2	Y_1	Y_2	\bar{Y}	XOR
0	0				0
0	1				1
1	0				1
1	1				0

Neural Net

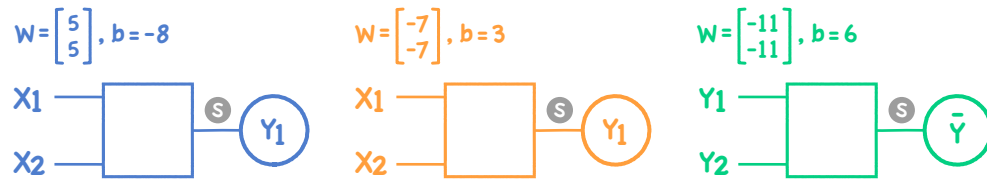


$$\begin{bmatrix} 0 & 0 \end{bmatrix} \begin{bmatrix} 5 \\ 5 \end{bmatrix} - 8 =$$

$$\begin{bmatrix} 0 & 0 \end{bmatrix} \begin{bmatrix} -7 \\ -7 \end{bmatrix} + 3 =$$

X_1	X_2	Y_1	Y_2	\bar{Y}	XOR
0	0				0
0	1				1
1	0				1
1	1				0

Neural Net



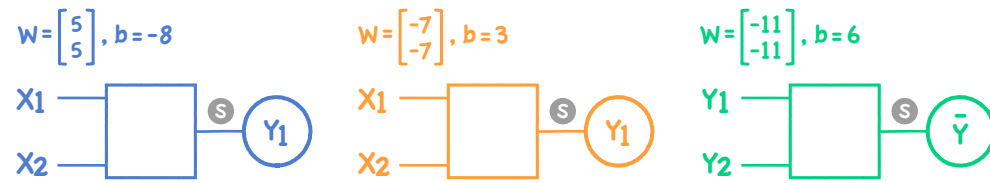
$$\begin{bmatrix} 0 & 0 \end{bmatrix} \begin{bmatrix} 5 \\ 5 \end{bmatrix} - 8 = 0 + 0 - 8 = -8, \text{Sigmoid}(-8) = 0$$

$$\begin{bmatrix} 0 & 0 \end{bmatrix} \begin{bmatrix} -7 \\ -7 \end{bmatrix} + 3 = 0 + 0 + 3 = 3, \text{Sigmoid}(3) = 1$$

$$\begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} -11 \\ -11 \end{bmatrix} + 6 = 0 + (-11) + 6 = -5 \\ \text{Sigmoid}(-5) = 0$$

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

Neural Net



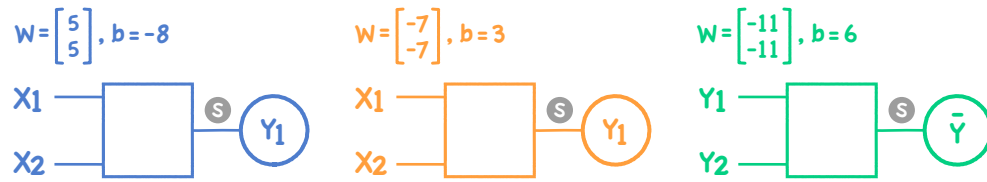
$$\begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} 5 \\ 5 \end{bmatrix} - 8 = 0 + 5 - 8 = -3, \text{Sigmoid}(-3) = 0$$

$$\begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} -7 \\ -7 \end{bmatrix} + 3 = 0 + (-7) + 3 = -4, \text{Sigmoid}(-4) = 0$$

$$\begin{bmatrix} 0 & 0 \end{bmatrix} \begin{bmatrix} -11 \\ -11 \end{bmatrix} + 6 = 0 + 0 + 6 = 6, \text{Sigmoid}(6) = 1$$

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

Neural Net



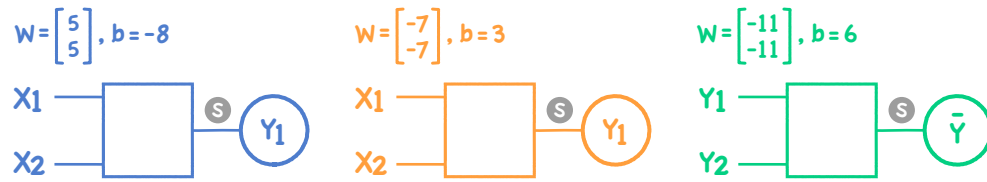
$$\begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} 5 \\ 5 \end{bmatrix} - 8 = 5 + 0 - 8 = -3, \text{Sigmoid}(-3) = 0$$

$$\begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} -7 \\ -7 \end{bmatrix} + 3 = (-7) + 0 + 3 = -4, \text{Sigmoid}(-4) = 0$$

$$\begin{bmatrix} 0 & 0 \end{bmatrix} \begin{bmatrix} -11 \\ -11 \end{bmatrix} + 6 = 0 + 0 + 6 = 6, \text{Sigmoid}(6) = 1$$

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	0	0	1	1
1	1				0

Neural Net



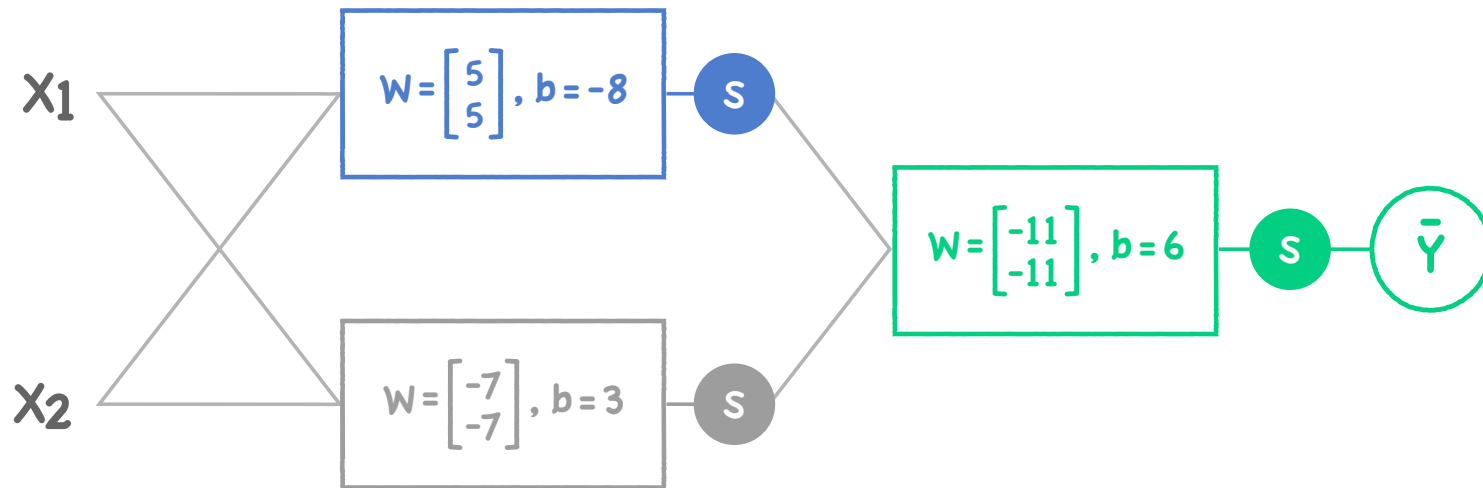
$$\begin{bmatrix} 1 & 1 \end{bmatrix} \begin{bmatrix} 5 \\ 5 \end{bmatrix} - 8 = 5 + 5 - 8 = 2, \text{Sigmoid}(2) = 1$$

$$\begin{bmatrix} 1 & 1 \end{bmatrix} \begin{bmatrix} -7 \\ -7 \end{bmatrix} + 3 = (-7) + (-7) + 3 = -11, \text{Sigmoid}(-11) = 0$$

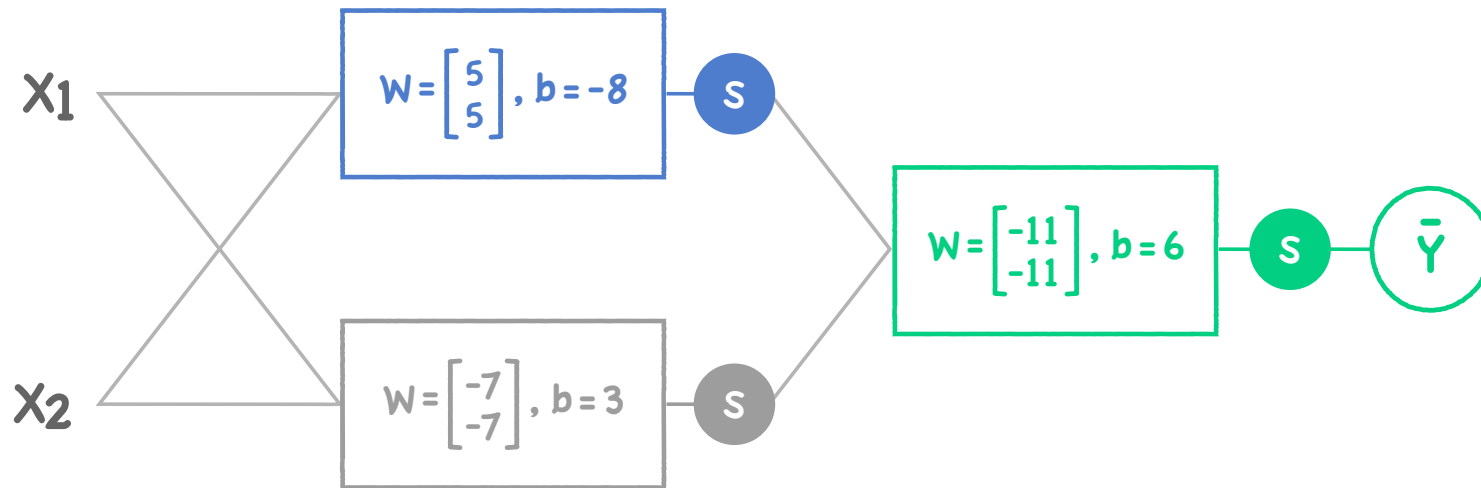
$$\begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} -11 \\ -11 \end{bmatrix} + 6 = (-11) + 0 + 6 = -5, \text{Sigmoid}(-5) = 0$$

X_1	X_2	Y_1	Y_2	\bar{Y}	XOR
0	0	0	1	0	0
0	1	0	0	1	1
1	0	0	0	1	1
1	1	1	0	0	0

Forward Propagation

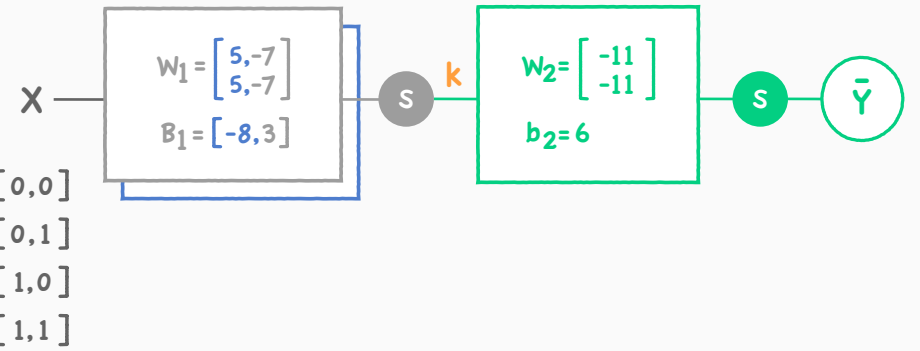
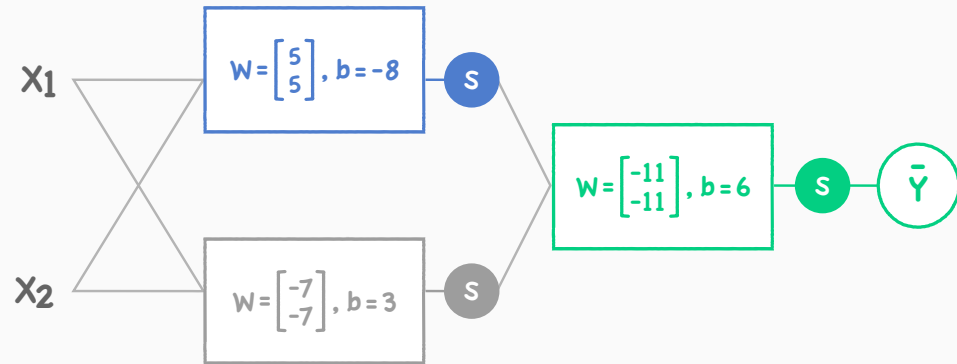


Forward Propagation



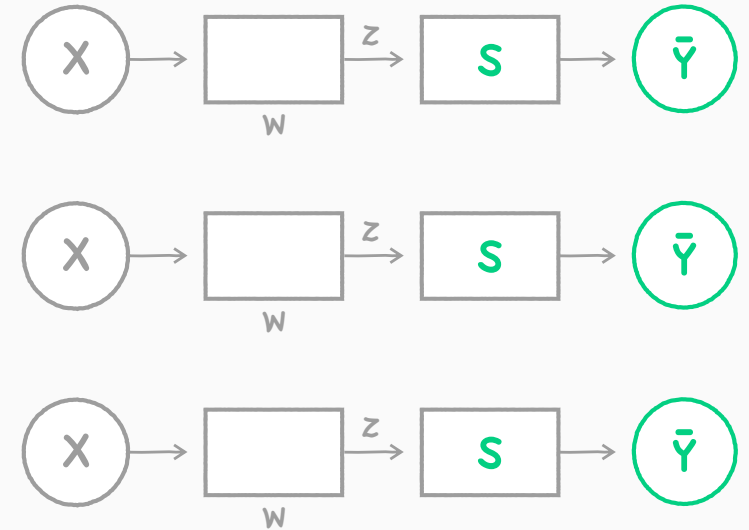
Can you find another W and b for the XOR?

NN

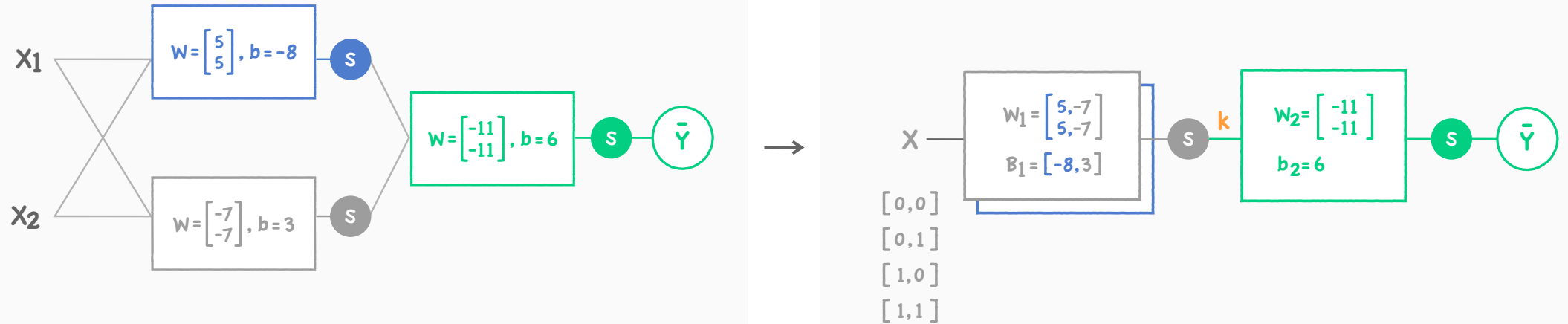


Recap: Lec 6-1 Multinomial Classification

$$\begin{bmatrix} W_{A1} & W_{A2} & W_{A3} \\ W_{B1} & W_{B2} & W_{B3} \\ W_{C1} & W_{C2} & W_{C3} \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ X_3 \end{bmatrix} = \begin{bmatrix} W_{A1}X_1 + W_{A2}X_2 + W_{A3}X_3 \\ W_{B1}X_1 + W_{B2}X_2 + W_{B3}X_3 \\ W_{C1}X_1 + W_{C2}X_2 + W_{C3}X_3 \end{bmatrix} = \begin{bmatrix} \bar{Y}_A \\ \bar{Y}_B \\ \bar{Y}_C \end{bmatrix}$$

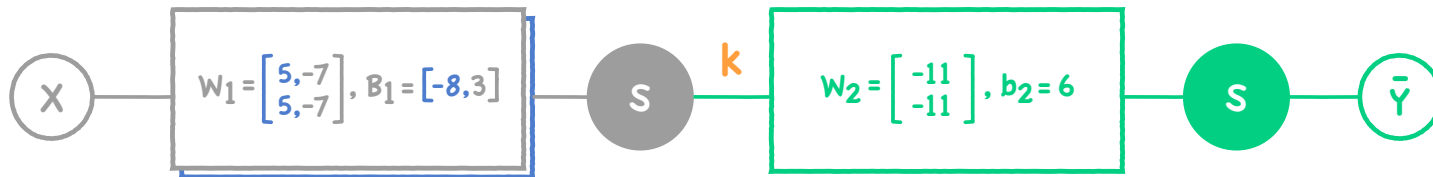


NN



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

NN

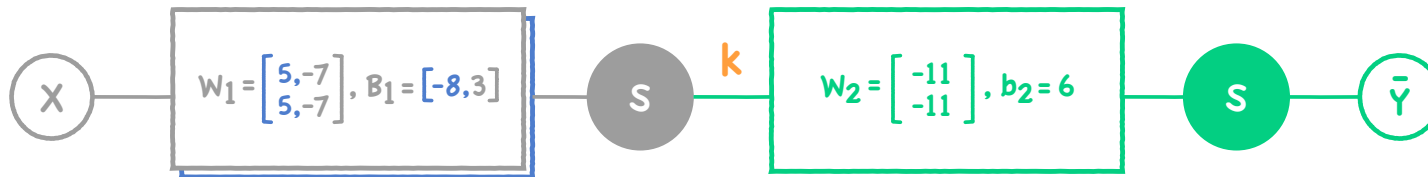


$[0, 0]$
 $[0, 1]$
 $[1, 0]$
 $[1, 1]$

$$K(X) = \text{Sigmoid}(XW_1 + B_1)$$
$$\bar{Y} = \#(X) = \text{Sigmoid}(K(X)W_2 + b_2)$$

NN

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

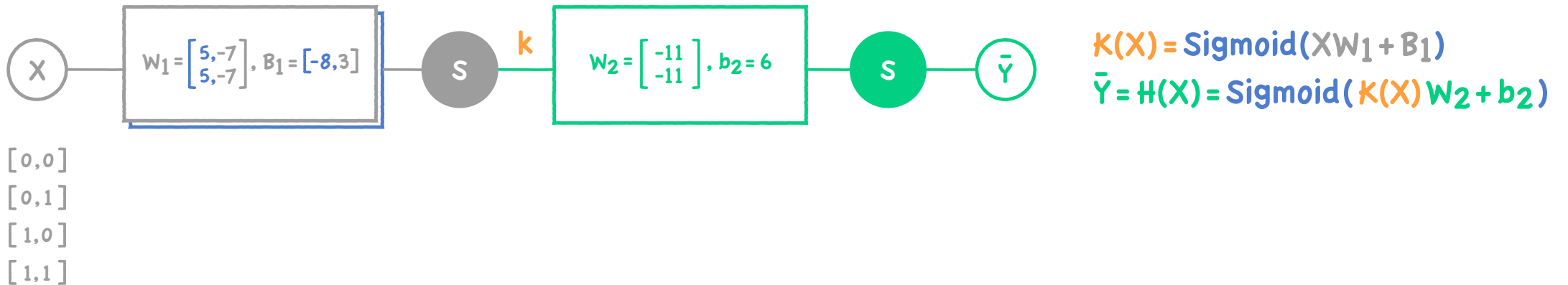


$$K(X) = \text{Sigmoid}(XW_1 + B_1)$$
$$\bar{Y} = \text{H}(X) = \text{Sigmoid}(K(X)W_2 + b_2)$$

$\begin{bmatrix} 0,0 \\ 0,1 \\ 1,0 \\ 1,1 \end{bmatrix}$

NN

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



How can we learn W_1, W_2, B_1, b_2 from training data?

NEXT LECTURE

BACKPROPAGATION