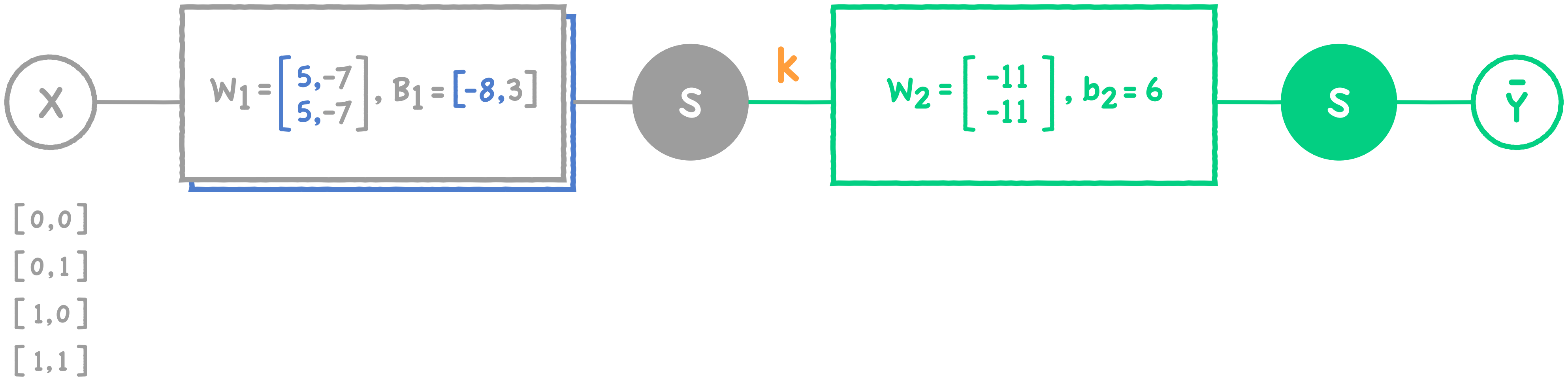
LECTURE 9-2

BACKPROPAGATION

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

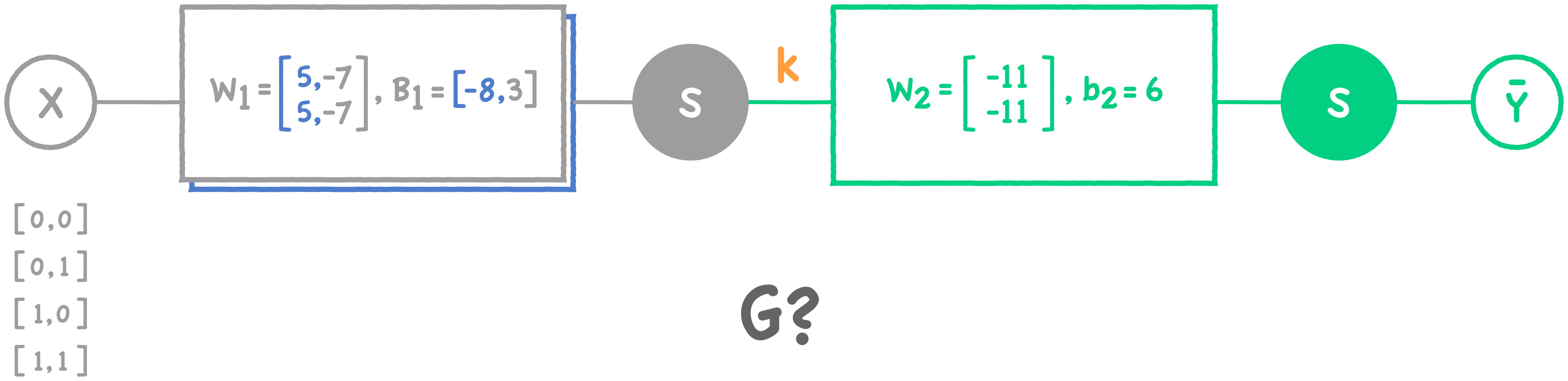
Neural Network(NN)



X
$W_1 = \begin{bmatrix} 5,-7 \\ 5,-7 \end{bmatrix}, B_1 = [-8,3]$
S
k
$W_2 = \begin{bmatrix} -11 \\ -11 \end{bmatrix}, b_2 = 6$
S
$\bar{Y}$
[0,0]
[0,1]
[1,0]
[1,1]

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

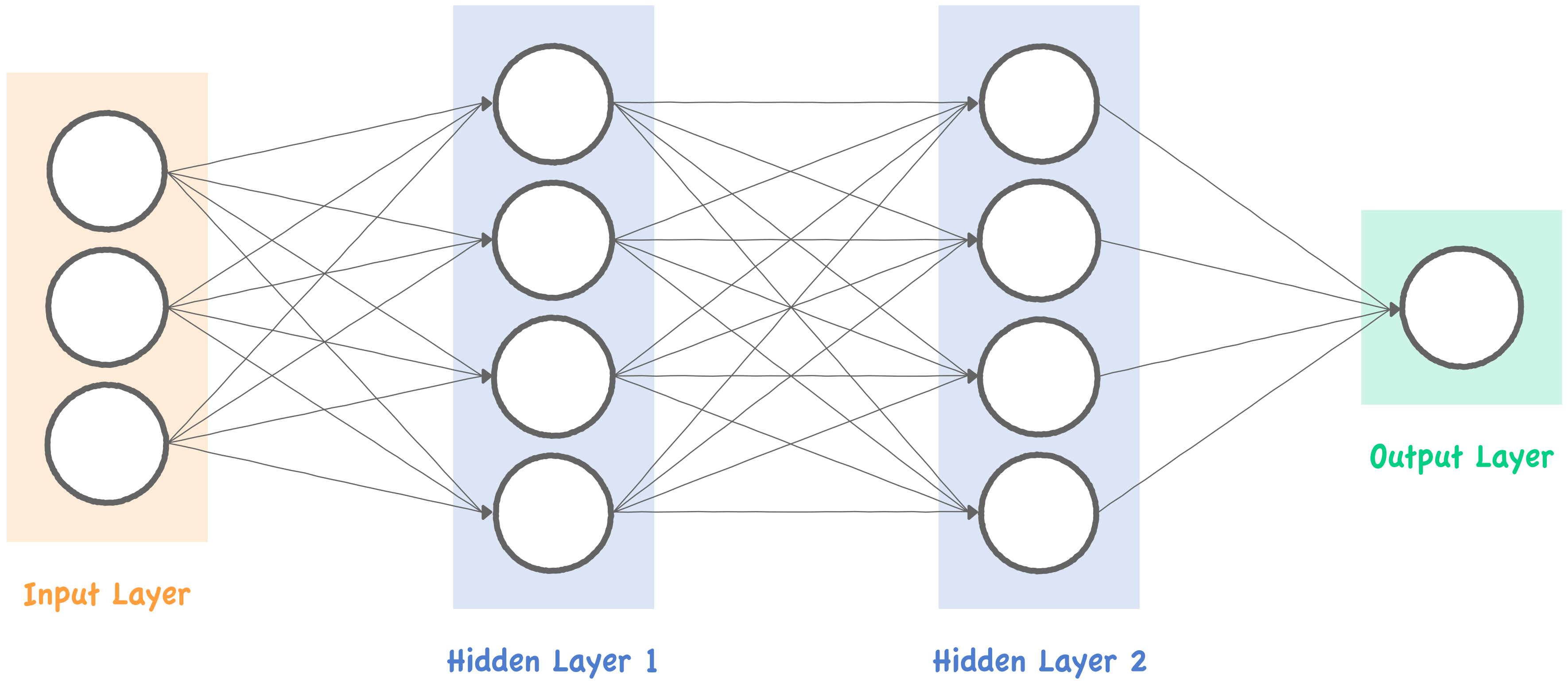
Neural Network(NN)


X
$W_1 = \begin{bmatrix} 5,-7 \\ 5,-7 \end{bmatrix}, B_1 = [-8,3]$
S
k
$W_2 = \begin{bmatrix} -11 \\ -11 \end{bmatrix}, b_2 = 6$
S
$\bar{Y}$
[0,0]
[0,1]
[1,0]
[1,1]

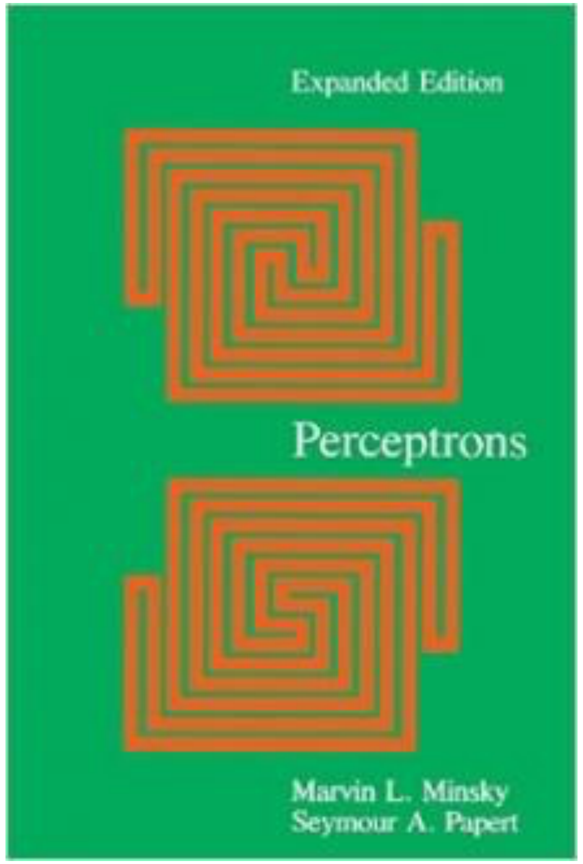
G?

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

Derivation



Perceptrons (1969)



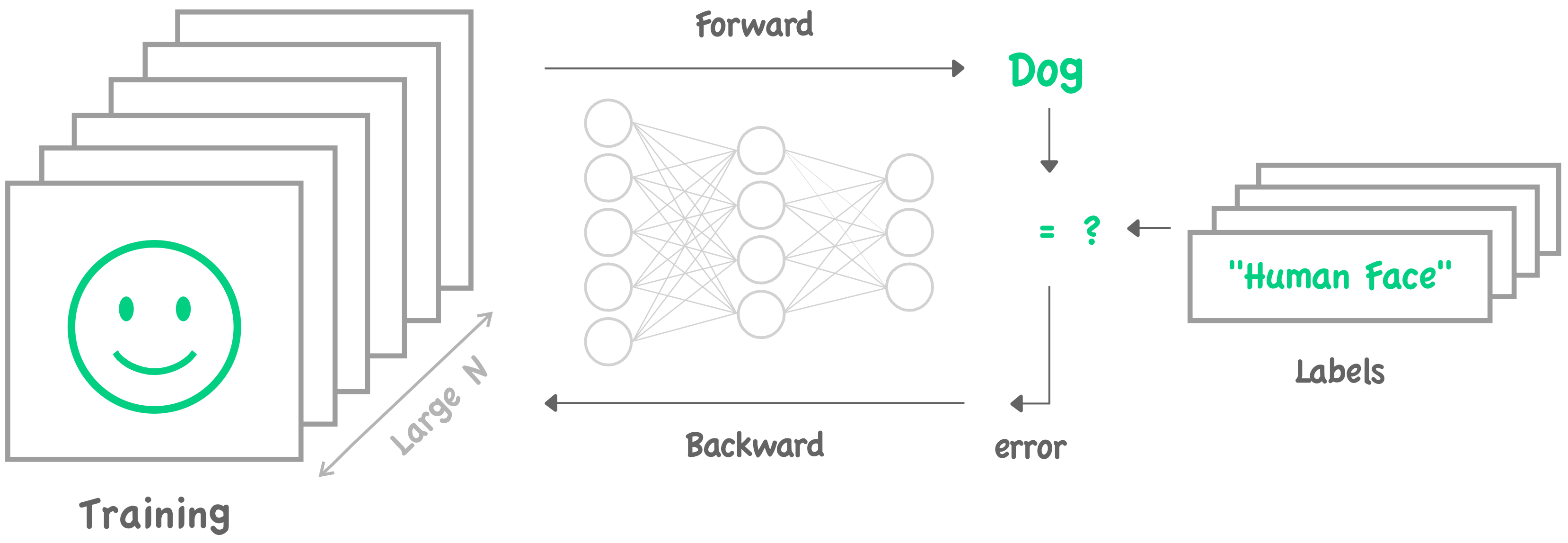
- We need to use MLP, multilayer perceptrons (multilayer neural nets)

- No one on earth had found a viable way to train MLPs good enough to learn such simple functions.

Perceptrons (1969)
by Marvin Minsky, founder of the MIT AI Lab

Backpropagation

1974, 1982 by Paul Werbos,
1986 by Hinton



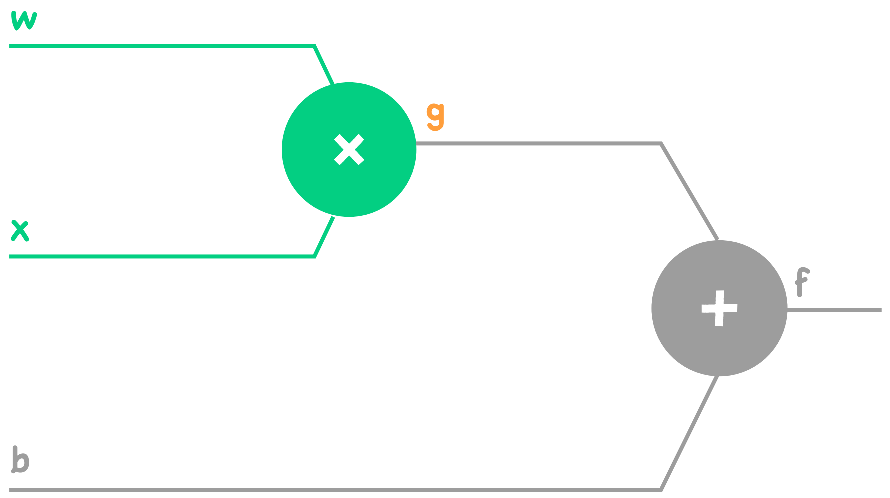
https://devblogs.nvidia.com/parallelforall/inference-next-step-gpu-accelerated-deep-learning/

Backpropagation (Chain Rule)

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

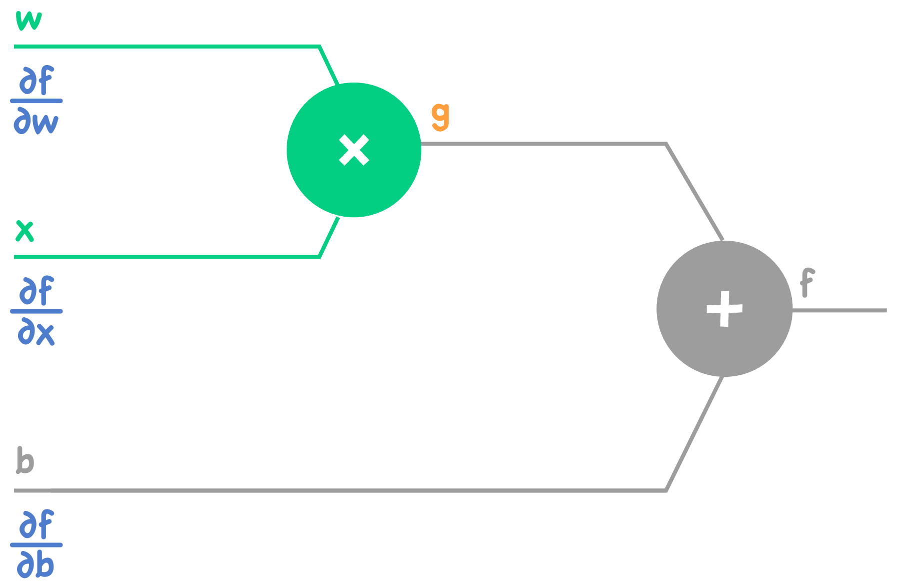
Backpropagation (Chain Rule)

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



Backpropagation (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

$f(x) = 3$

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

$f(x) = x+3$

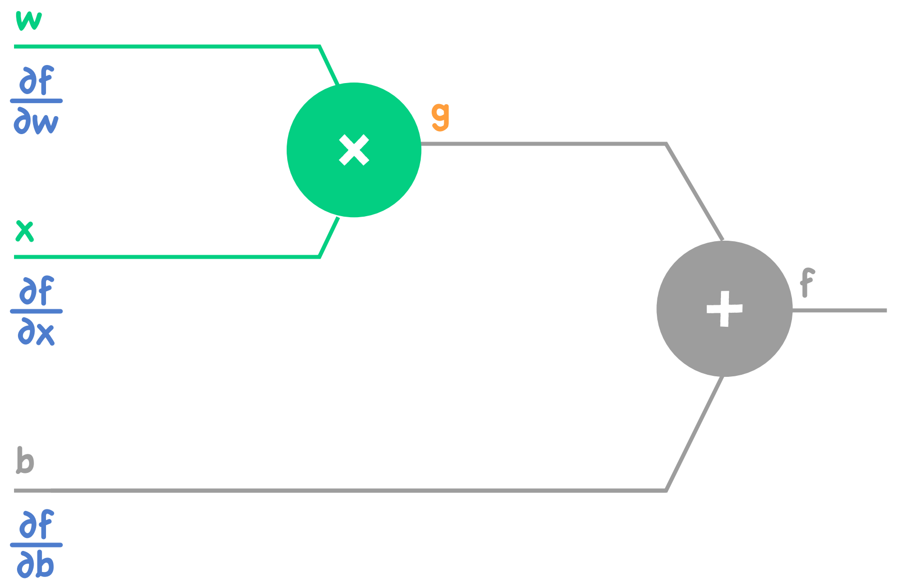
$f(x,y) = x+y, \frac{\partial f}{\partial x}$

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

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

Backpropagation (Chain Rule)

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

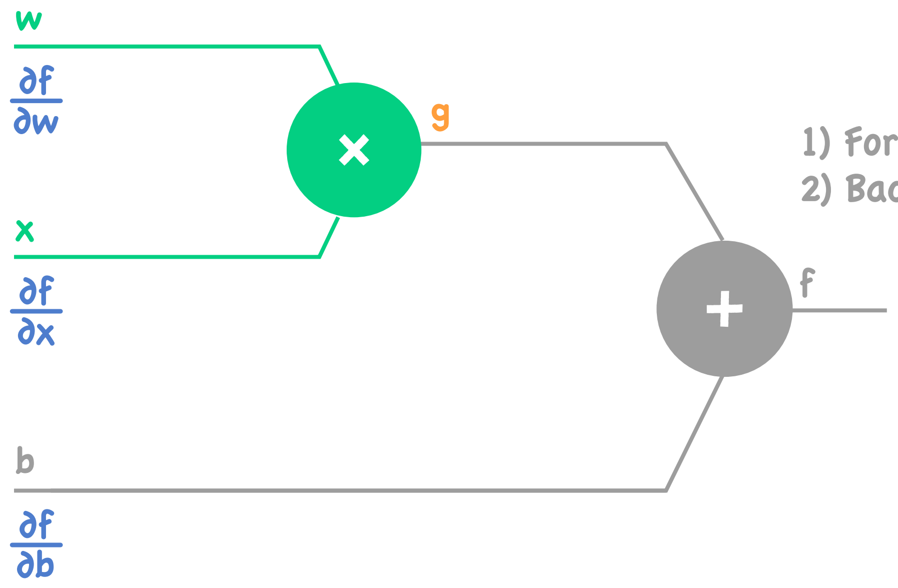


w
$\frac{\partial f}{\partial w}$
x
$\frac{\partial f}{\partial x}$
b
$\frac{\partial f}{\partial b}$
×
g
+
f

http://cs231n.stanford.edu/

Backpropagation (Chain Rule)

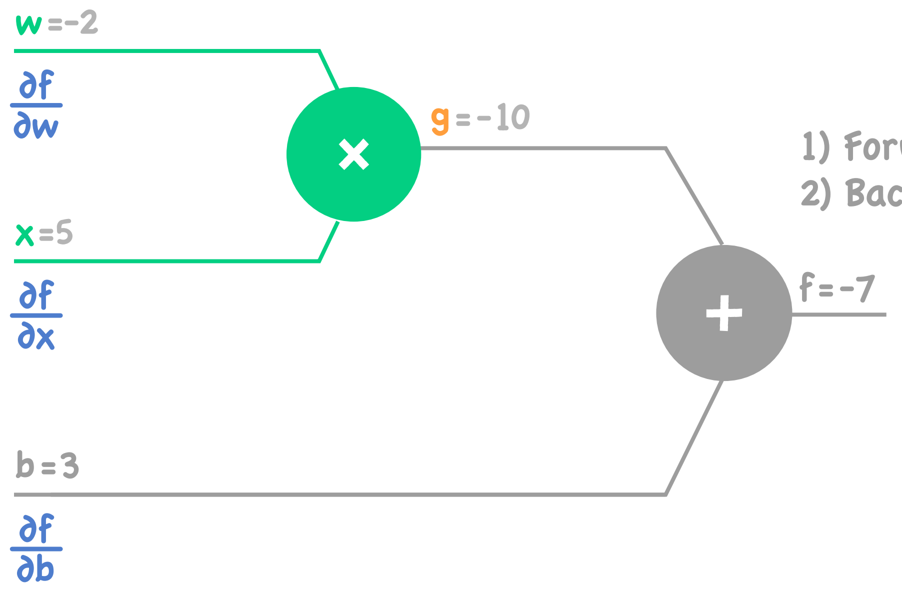
$f = wx + b, \quad g = wx, \quad f = g + b$


w
$\frac{\partial f}{\partial w}$
x
$\frac{\partial f}{\partial x}$
b
$\frac{\partial f}{\partial b}$
×
g
+
f

1) Forward (w = -2, x = 5, b = 3)
2) Backward

Backpropagation (Chain Rule)

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



1) Forward (w=-2, x=5, b=3)
2) Backward

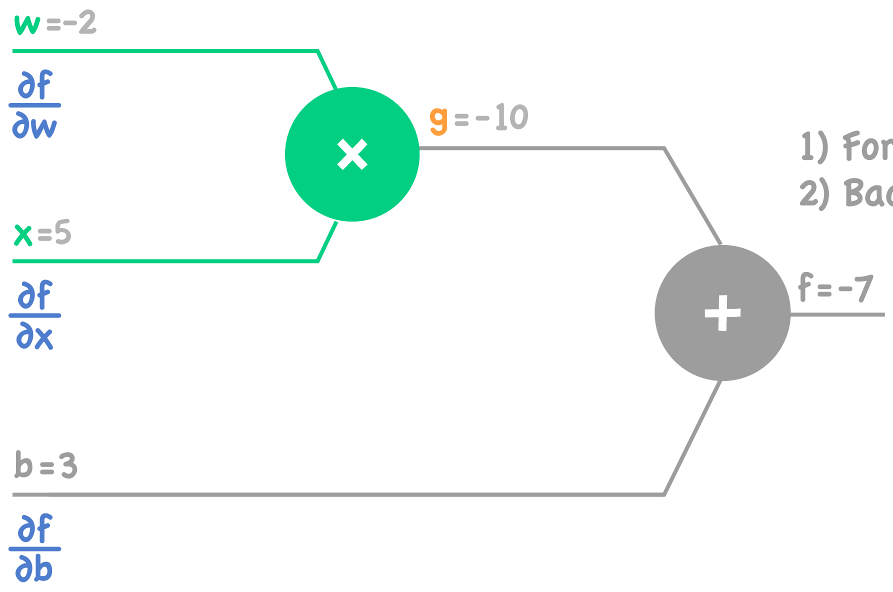
http://cs231n.stanford.edu/

Backpropagation (Chain Rule)

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

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

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



$w=-2$
$\frac{\partial f}{\partial w}$
$x=5$
$\frac{\partial f}{\partial x}$
×
$g=-10$
+
$f=-7$
$b=3$
$\frac{\partial f}{\partial b}$

1) Forward (w = -2, x = 5, b = 3)
2) Backward

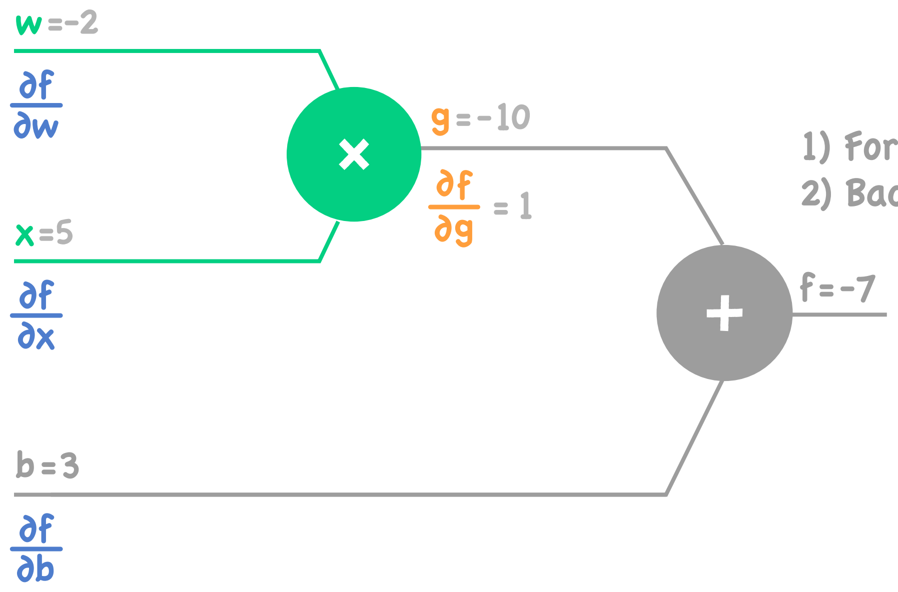
http://cs231n.stanford.edu/

Backpropagation (Chain Rule)

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

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

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



w=-2
$\frac{\partial f}{\partial w}$
x=5
$\frac{\partial f}{\partial x}$
×
g=-10
$\frac{\partial f}{\partial g} = 1$
+
f=-7
b=3
$\frac{\partial f}{\partial b}$

1) Forward (w=-2, x=5, b=3)
2) Backward

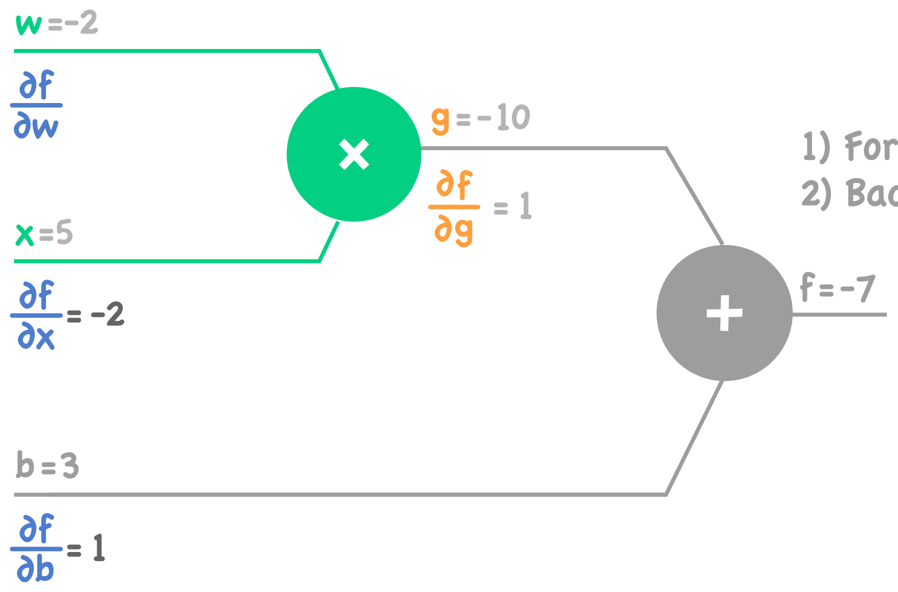
http://cs231n.stanford.edu/

Backpropagation (Chain Rule)

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

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

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



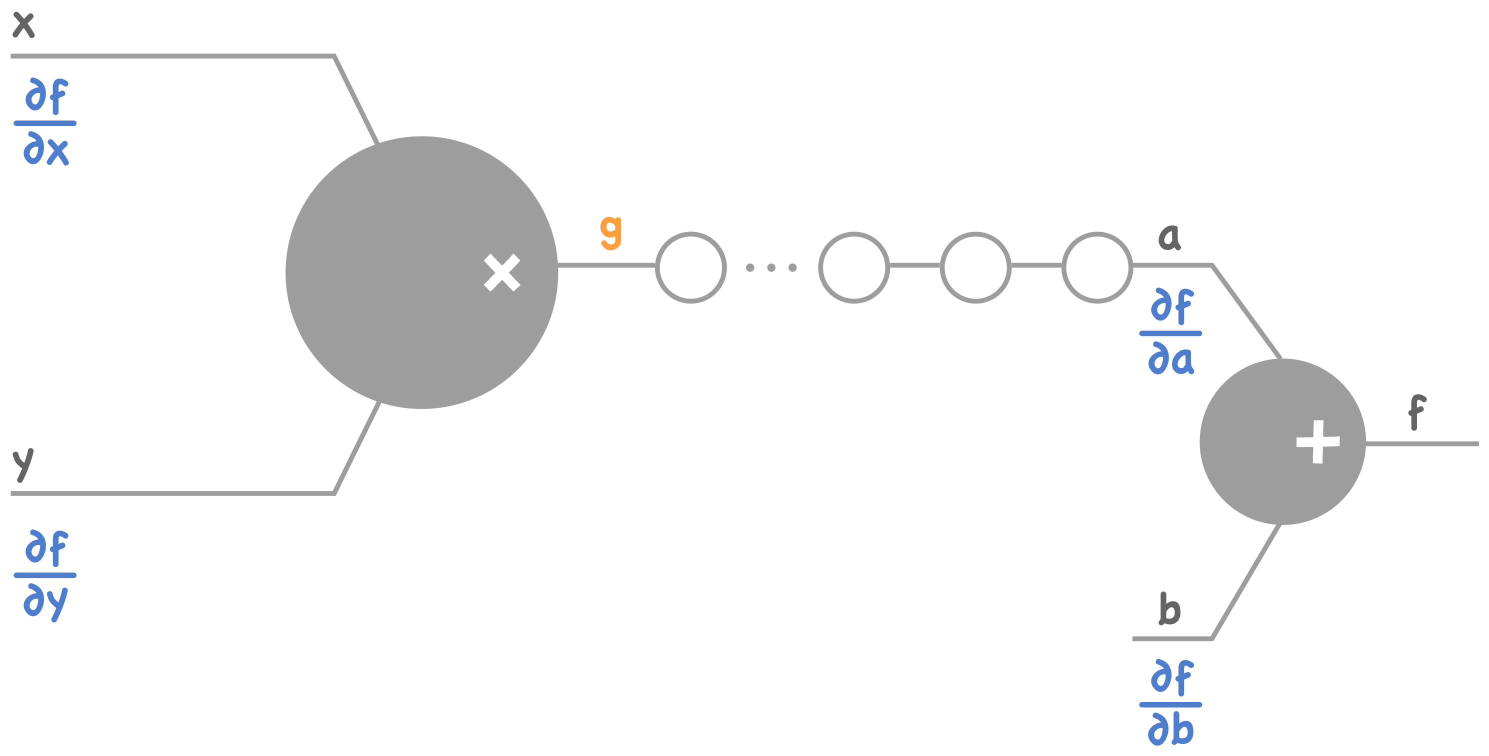
1) Forward (w=-2, x=5, b=3)
2) Backward

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

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

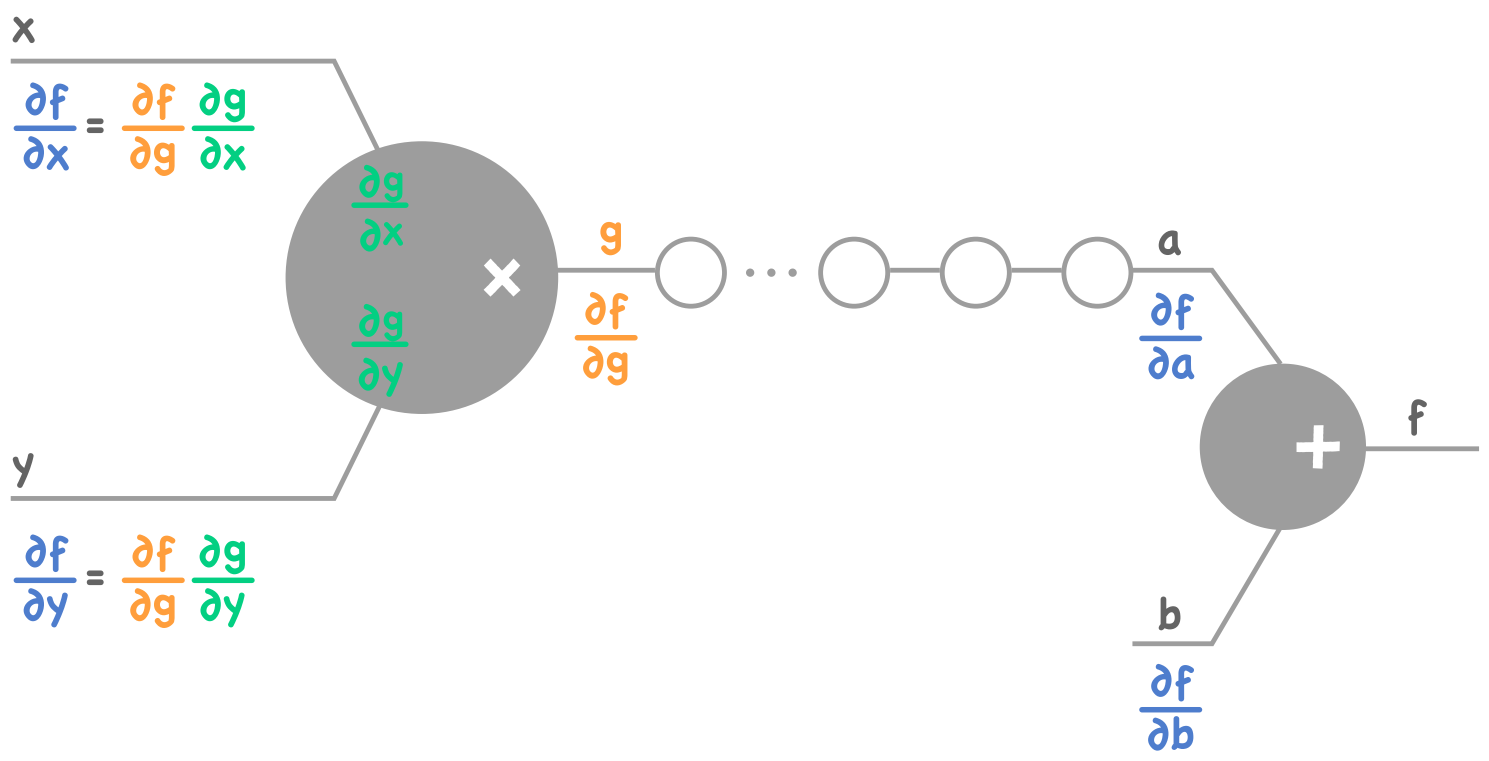
http://cs231n.stanford.edu/

Backpropagation (Chain Rule)


x
$\frac{\partial f}{\partial x}$
y
$\frac{\partial f}{\partial y}$
×
g
a
$\frac{\partial f}{\partial a}$
+
f
b
$\frac{\partial f}{\partial b}$

http://cs231n.stanford.edu/

Backpropagation (Chain Rule)


x
$\frac{\partial f}{\partial x} = \frac{\partial f}{\partial g}\frac{\partial g}{\partial x}$
$\frac{\partial g}{\partial x}$
$\frac{\partial g}{\partial y}$
×
g
$\frac{\partial f}{\partial g}$
y
$\frac{\partial f}{\partial y} = \frac{\partial f}{\partial g}\frac{\partial g}{\partial y}$
a
$\frac{\partial f}{\partial a}$
+
f
b
$\frac{\partial f}{\partial b}$

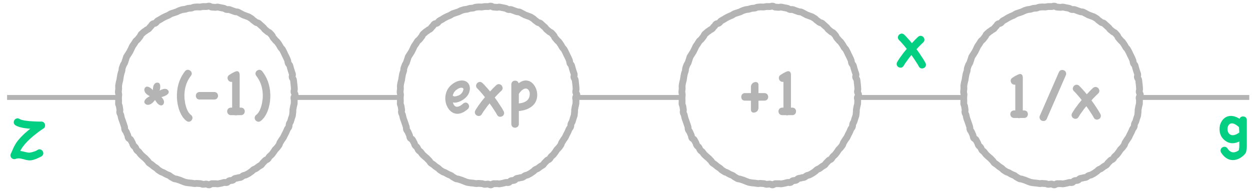
http://cs231n.stanford.edu/

Sigmoid

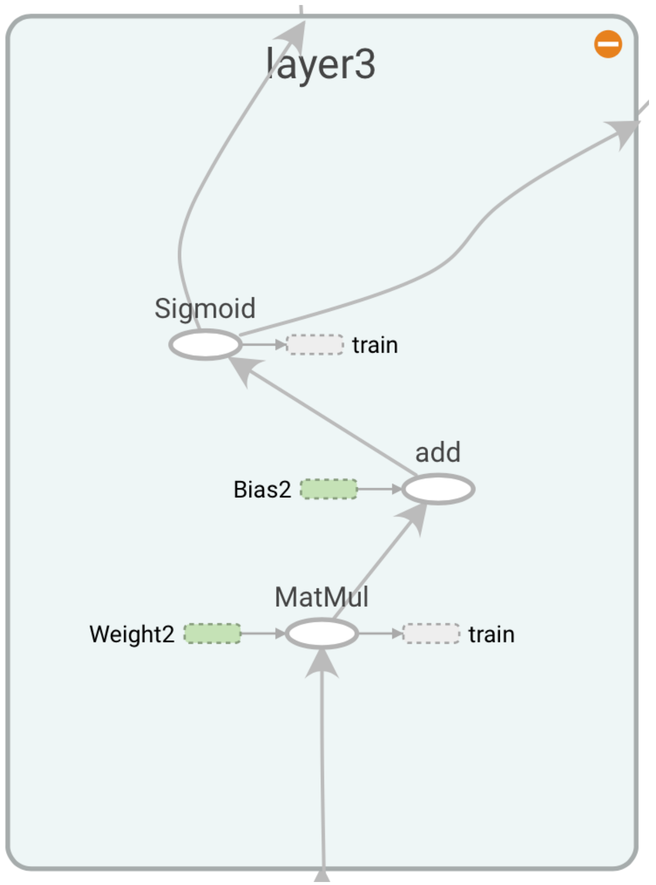
$$g(z) = \frac{1}{1+e^{-2}}$$

Sigmoid

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



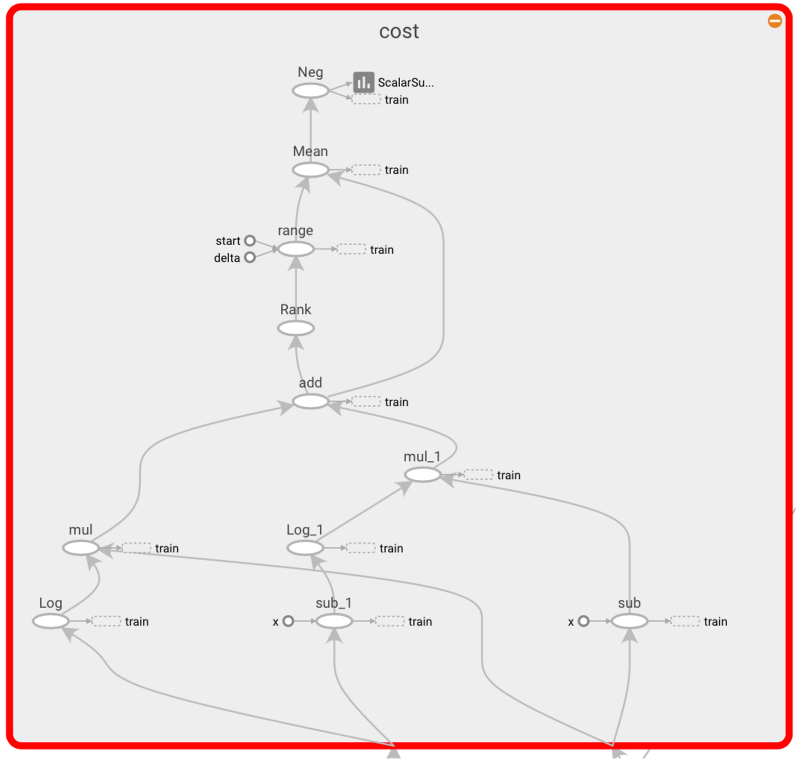
Back Propagation in TensorFlow



[TensorBoard]

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

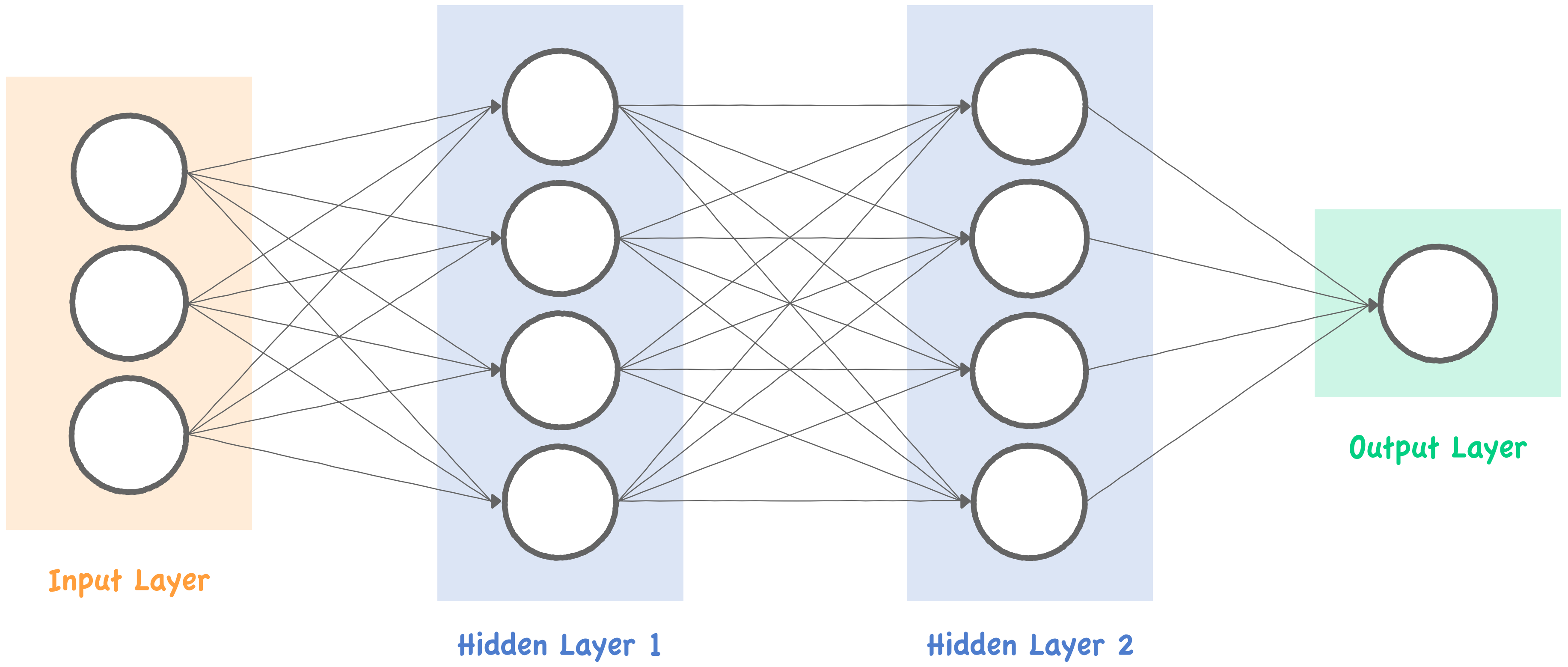
Back Propagation in TensorFlow



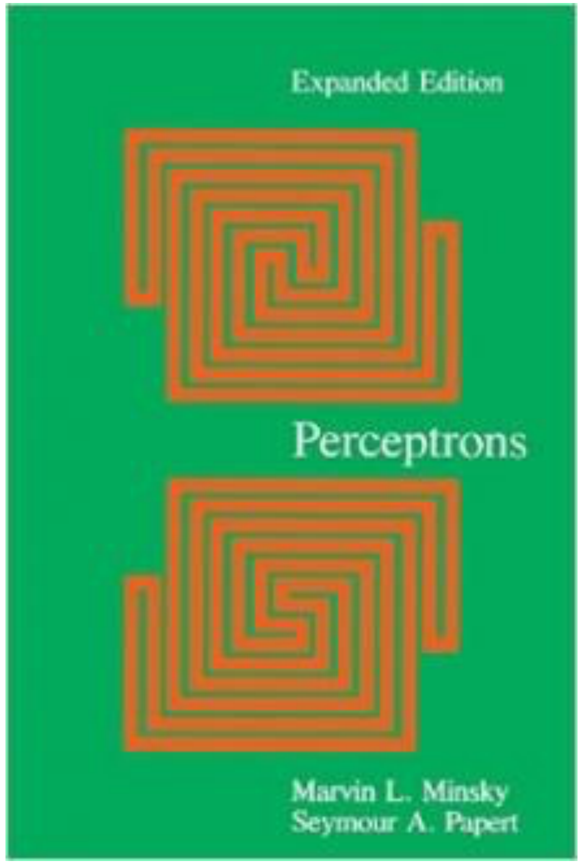
[TensorBoard]

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

Backpropagation



Perceptrons (1969)



- 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.

Perceptrons (1969)
by Marvin Minsky, founder of the MIT AI Lab

NEXT LECTURE

ReLU