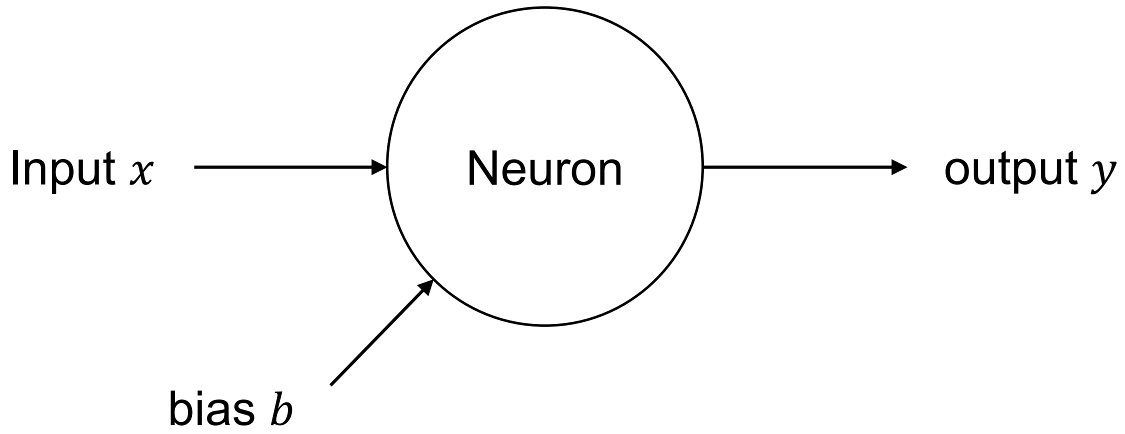
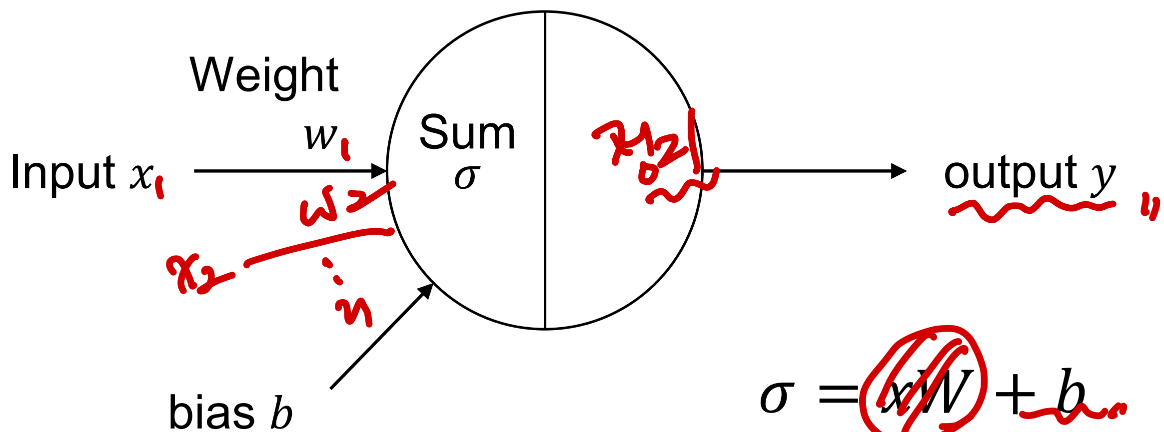
Single Neuron Training

Youngtaek Hong, PhD

Artificial Neuron



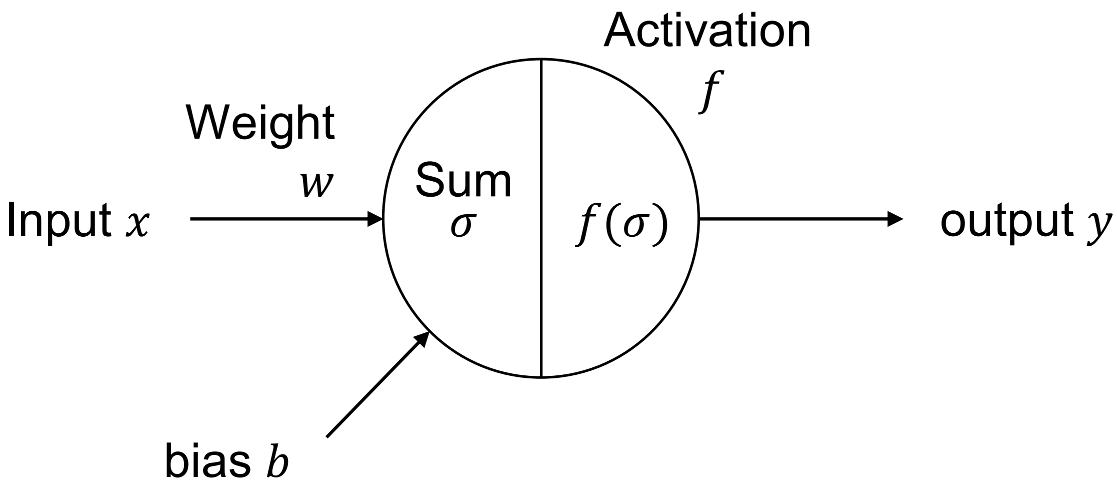
sigma



$$\sigma = xW + b$$

$$\sigma = x_1 \cdot w_1 + \cdots + x_n \cdot w_n + b$$

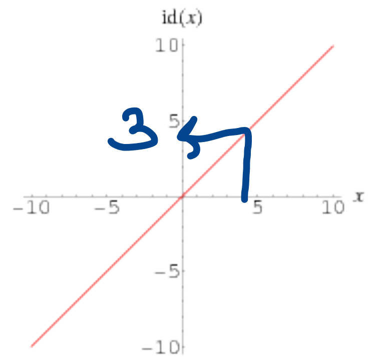
Activation



Activation
f
Weight
w
Input x
Sum
σ
$f(\sigma)$
output y
bias b

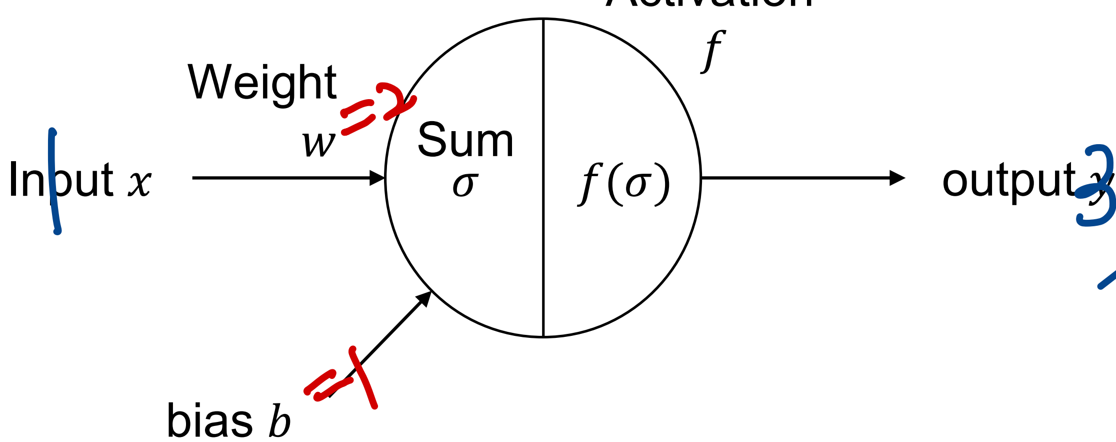
Feed forward

Activation: identity(= linear)



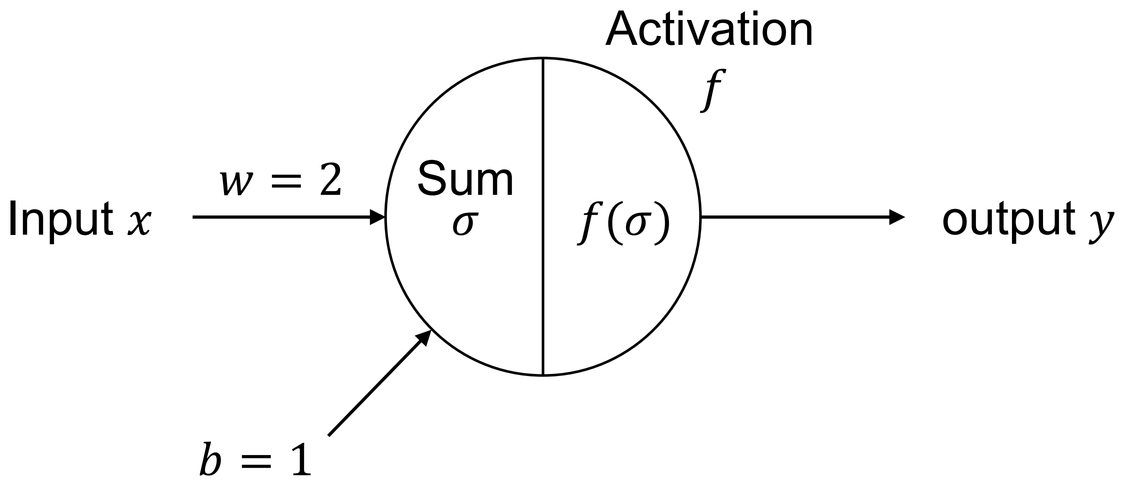
$\sigma = 1 \times 2 + 1$

$= 3$

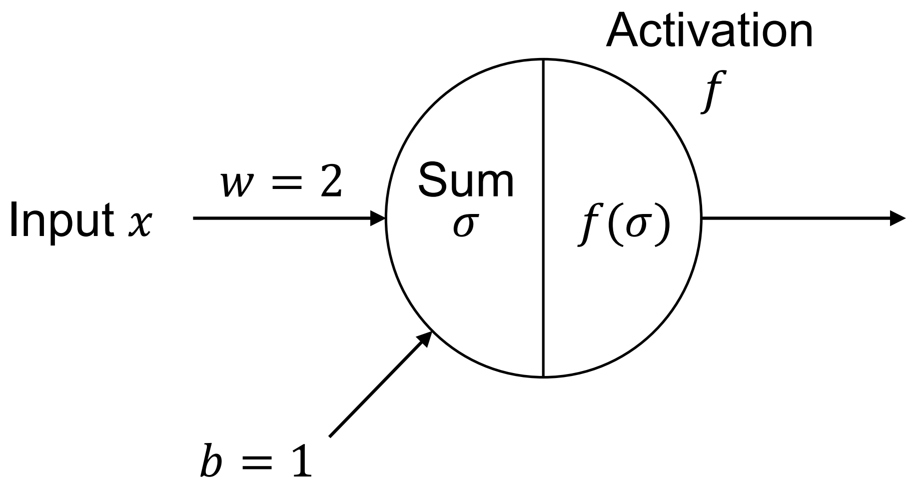


3 4

Feed forward



Feed forward



output y E target Y

$$MSE = \frac{1}{n}\sum(y - Y)^2$$

$$E = \frac{1}{2}(y - Y)^2$$

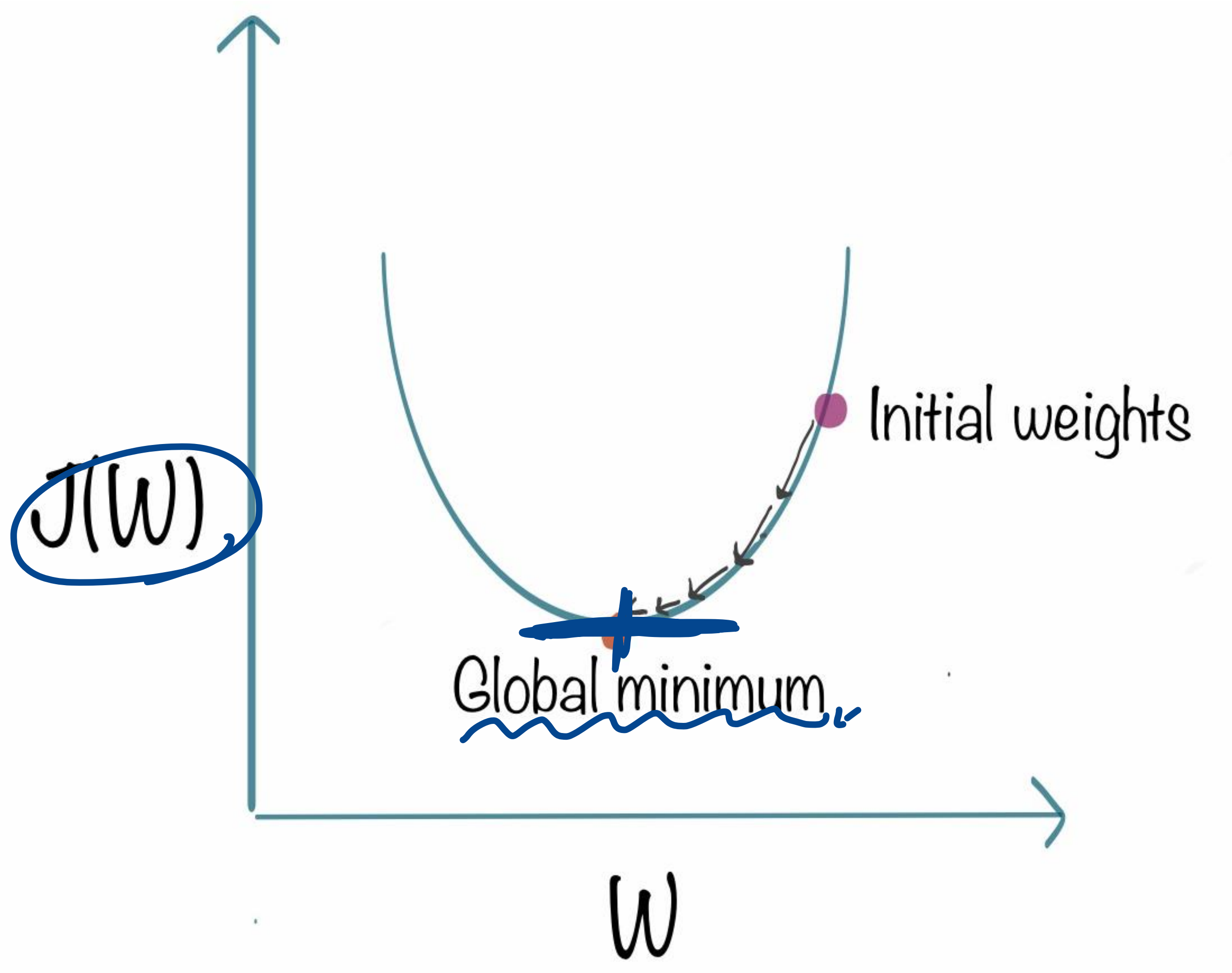
정답.

뉴런의 출력값

$$\frac{\partial E = \frac{1}{2}(y-Y)^2}{\partial y} = \frac{1}{2} \cdot 2(y-Y) \cdot 1$$

$$= (y-Y).$$ ✓

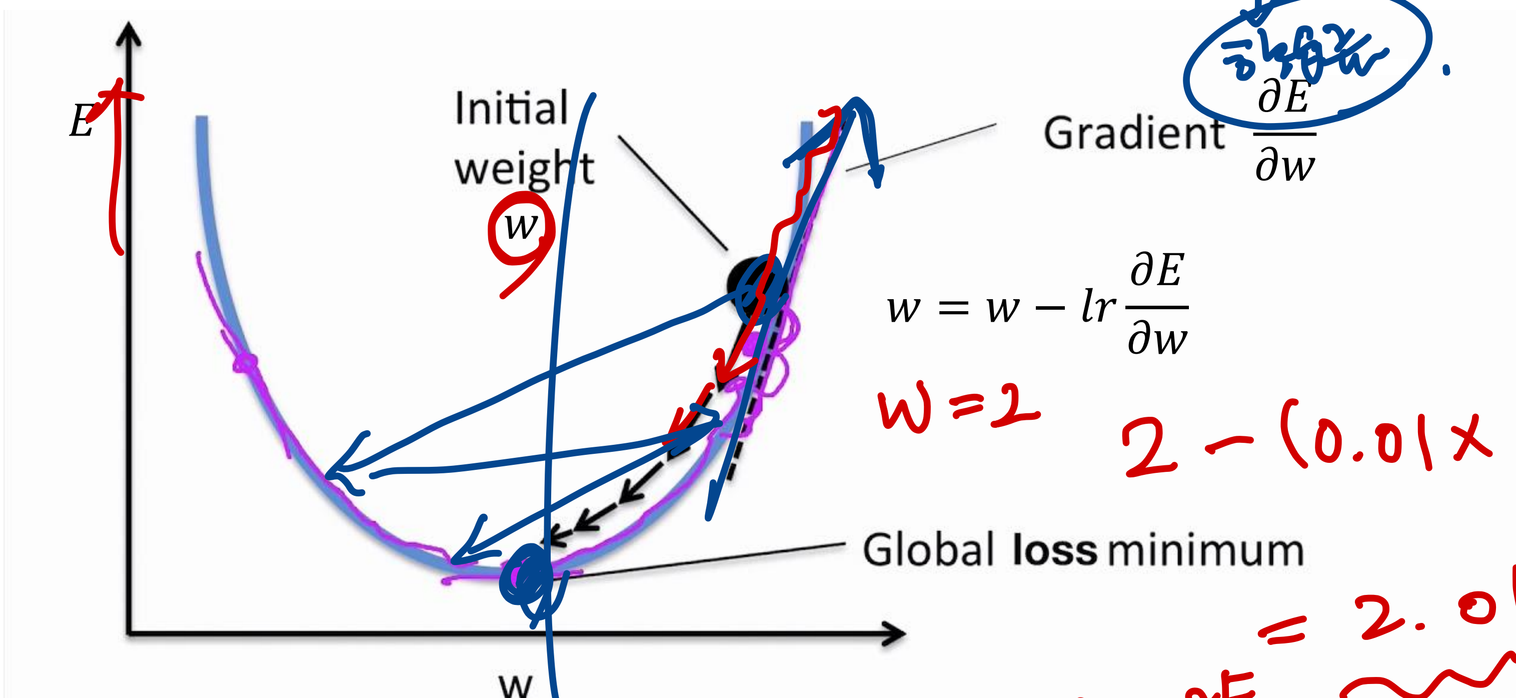
Gradient Descendent Method



Gradient Descendent Method

$$W = W - lr \frac{\partial E}{\partial W}$$

학습률



$W = 2$

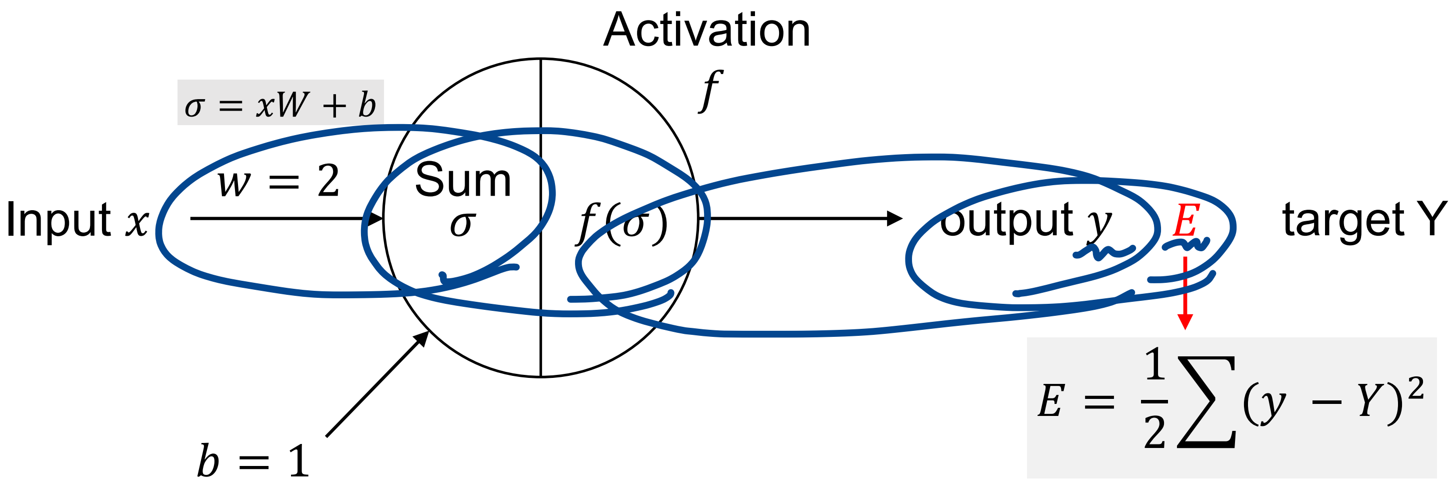
$2 - (0.01 \times -1) = 2.01$

$b = b - lr \frac{\partial E}{\partial b}$

$1 - (0.01 \times -1) = 1.01$

$2.01 \times 1 + 1.01 = 3.02$

How to control E by adjusting w



① ② ③ ④

$$\frac{\partial E}{\partial w} = \frac{\partial E}{\partial y} \times \frac{\partial y}{\partial f} \times \frac{\partial f}{\partial \sigma} \times \frac{\partial \sigma}{\partial w}$$

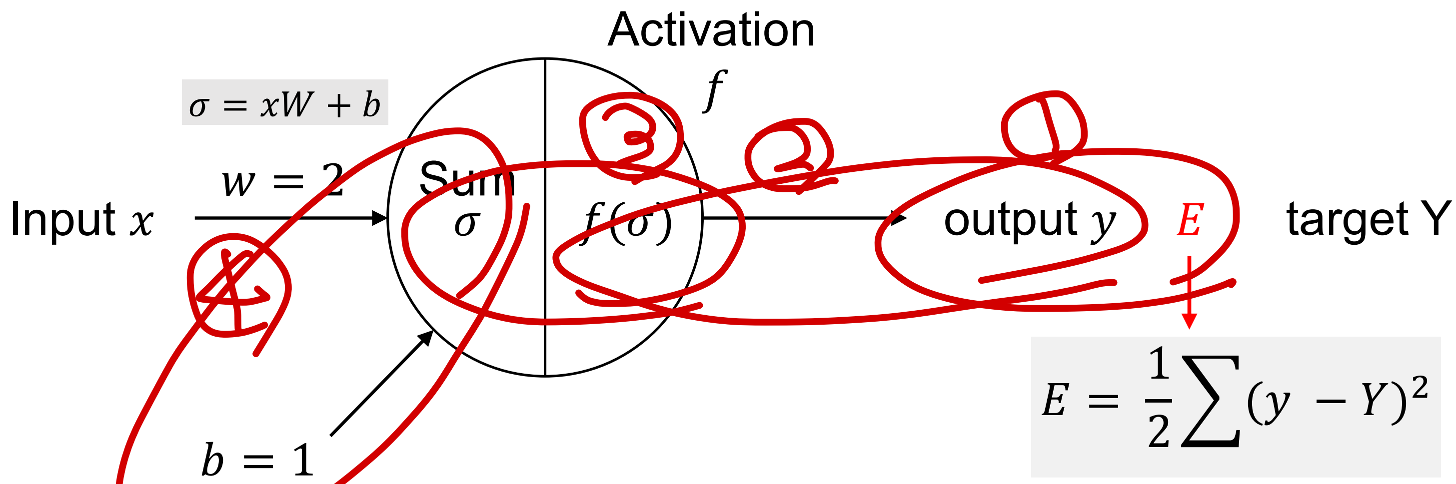
$$\sigma = w \cdot x + \cancel{b}$$

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

$$\frac{\partial E}{\partial y} = (y - Y) \qquad \frac{\partial y}{\partial f} = 1 \qquad \frac{\partial f}{\partial \sigma} = 1 \qquad \frac{\partial \sigma}{\partial w} = x$$

✓ $$\frac{\partial E}{\partial w} = (y - Y) \times x$$

How to control E by adjusting b



$$E = \frac{1}{2}\sum(y - Y)^2$$

$$\frac{\partial \sigma}{\partial b} = \cancel{w \cdot x} + \cancel{b} = 1$$

1 2 3 4

$$\frac{\partial E}{\partial b} = \frac{\partial E}{\partial y} \cdot \frac{\partial y}{\partial f} \cdot \frac{\partial f}{\partial \sigma} \cdot \frac{\partial \sigma}{\partial b}$$

$$\frac{\partial E}{\partial y} = (y - Y) \qquad \frac{\partial y}{\partial f} = 1 \qquad \frac{\partial f}{\partial \sigma} = 1 \qquad \frac{\partial \sigma}{\partial b} = 1$$

$$\frac{\partial E}{\partial b} = (y - Y) \times 1 \times 1 \times 1$$

Neuron class

```
[1] import tensorflow as tf
```

```
[2] class Neuron(object):

      def __init__(self, w, b):
        self.w = tf.Variable(w, name='weight')
        self.b = tf.Variable(b, name='bias')
        self.output = 0

        self.input  = tf.placeholder(tf.float32, shape=[1], name="input")
        self.target = tf.placeholder(tf.float32, shape=[1], name="output")

    #   Linear function = identity function
      def getActivation(self, x):
        return x

      def getActivationGradient(self, x):
        return 1.0

      def propBackWard(self):
        lr = 0.01

        grad = (self.output - self.target) * 1.0 * self.getActivationGradient(self.output)
        self.w = self.w - (lr * grad * self.input)
        self.b = self.b - (lr * grad * 1.0)

        return self.feedforward()


      def feedforward(self):
        sigma = self.w * self.input + self.b
        self.output = self.getActivation(sigma)
        return self.output
```

Neuron class

```
[3]  my_neuron = Neuron(w = 2.0, b = 1.0)

     x = [1.0]
     y = [4.0]
```

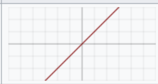
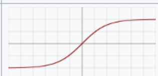
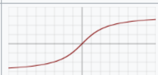
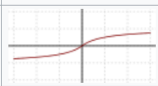
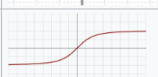
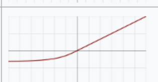
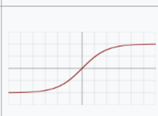
```
[4]  with tf.Session() as sess:
       sess.run(tf.global_variables_initializer())
       print (sess.run(my_neuron.feedforward(),
                       feed_dict={my_neuron.input: x,
                                  my_neuron.target:y}))

       for i in range(150):
         print (sess.run(my_neuron.propBackWard(),
                         feed_dict={my_neuron.input: x,
                                    my_neuron.target:y}))
         print (sess.run([my_neuron.w, my_neuron.b],
                         feed_dict={my_neuron.input: x,
                                    my_neuron.target:y}))
```

```
[3.]
[3.02]
[array([2.01], dtype=float32), array([1.01], dtype=float32)]
[3.0396]
[array([2.0198], dtype=float32), array([1.0198], dtype=float32)]
[3.0588078]
[array([2.029404], dtype=float32), array([1.0294039], dtype=float32)]
[3.0776315]
[array([2.0388157], dtype=float32), array([1.0388159], dtype=float32)]
[3.0960789]
[array([2.0480394], dtype=float32), array([1.0480396], dtype=float32)]
[3.1141572]
[array([2.0570786], dtype=float32), array([1.0570787], dtype=float32)]
[3.131874]
[array([2.065937], dtype=float32), array([1.0659372], dtype=float32)]
[3.1492367]
[array([2.0746183], dtype=float32), array([1.0746185], dtype=float32)]
[3.1662521]
[array([2.083126], dtype=float32), array([1.0831261], dtype=float32)]
```

```
[3.9465704]
[array([2.4732854], dtype=float32), array([1.4732851], dtype=float32)]
[3.947639]
[array([2.4738197], dtype=float32), array([1.4738194], dtype=float32)]
[3.9486861]
[array([2.4743433], dtype=float32), array([1.474343], dtype=float32)]
[3.9497125]
[array([2.4748564], dtype=float32), array([1.4748561], dtype=float32)]
[3.9507182]
[array([2.4753592], dtype=float32), array([1.475359], dtype=float32)]
```

Activation functions

Name	Plot	Equation	Derivative (with respect to x)	Range	Order of continuity
Identity		get activation gradient (x) return (getActivation (x)) · (1 − getActivation (x))			
Binary step		$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} 0 & \text{for } x \neq 0 \\ ? & \text{for } x = 0 \end{cases}$	$\{0, 1\}$	C^{-1}
Logistic (a.k.a. Sigmoid or Soft step)		$f(x) = \sigma(x) = \frac{1}{1 + e^{-x}}$ [1]	$f'(x) = f(x)(1 - f(x))$	$(0, 1)$	C^{∞}
TanH		$f(x) = \tanh(x) = \frac{(e^x - e^{-x})}{(e^x + e^{-x})}$	$f'(x) = 1 - f(x)^2$	$(-1, 1)$	C^{∞}
ArcTan		$f(x) = \tan^{-1}(x)$	$f'(x) = \frac{1}{x^2 + 1}$	$\left(-\frac{\pi}{2}, \frac{\pi}{2}\right)$	C^{∞}
ArSinH		$f(x) = \sinh^{-1}(x) = \ln\left(x + \sqrt{x^2 + 1}\right)$	$f'(x) = \frac{1}{\sqrt{x^2 + 1}}$	$(-\infty, \infty)$	C^{∞}
ElliotSig[8][9][10] Softsign[11][12]		$f(x) = \frac{x}{1 + \|x\|}$	$f'(x) = \frac{1}{(1 + \|x\|)^2}$	$(-1, 1)$	C^{1}
Inverse square root unit (ISRU)[13]		$f(x) = \frac{x}{\sqrt{1 + \alpha x^2}}$	$f'(x) = \left(\frac{1}{\sqrt{1 + \alpha x^2}}\right)^3$	$\left(-\frac{1}{\sqrt{\alpha}}, \frac{1}{\sqrt{\alpha}}\right)$	C^{∞}
Inverse square root linear unit (ISRLU)[13]		$f(x) = \begin{cases} \frac{x}{\sqrt{1+\alpha x^2}} & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} \left(\frac{1}{\sqrt{1+\alpha x^2}}\right)^3 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$	$\left(-\frac{1}{\sqrt{\alpha}}, \infty\right)$	C^{2}
Square Nonlinearity (SQNL)[10]		$f(x) = \begin{cases} 1 & : x > 2.0 \\ x - \frac{x^2}{4} & : 0 \leq x \leq 2.0 \\ x + \frac{x^2}{4} & : -2.0 \leq x < 0 \\ -1 & : x < -2.0 \end{cases}$	$f'(x) = 1 \mp \frac{x}{2}$	$(-1, 1)$	C^{2}
Rectified linear unit (ReLU)[14]		$f(x) = \begin{cases} 0 & \text{for } x \leq 0 \\ x & \text{for } x > 0 \end{cases}$	$f'(x) = \begin{cases} 0 & \text{for } x \leq 0 \\ 1 & \text{for } x > 0 \end{cases}$	$[0, \infty)$	C^{0}