

# شبکه‌های عصبی: آشنایی

سید ناصر رضوی [www.snrazavi.ir](http://www.snrazavi.ir)

۱۳۹۷

# شبکه‌های عصبی



# شبکه‌های عصبی

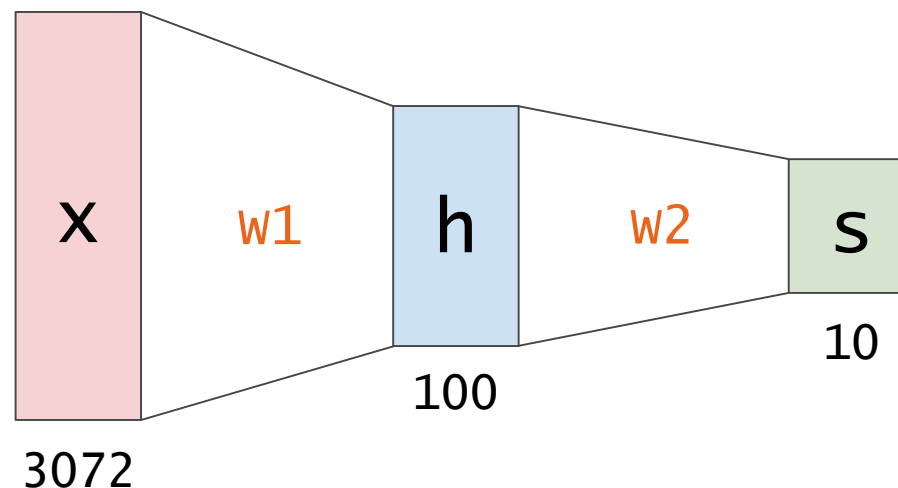
۳

$$f = Wx$$

(قبلاً) تابع امتیاز خطی:

$$f = W_2 \max(0, W_1 x)$$

(اکنون) شبکه عصبی ۲ لایه:

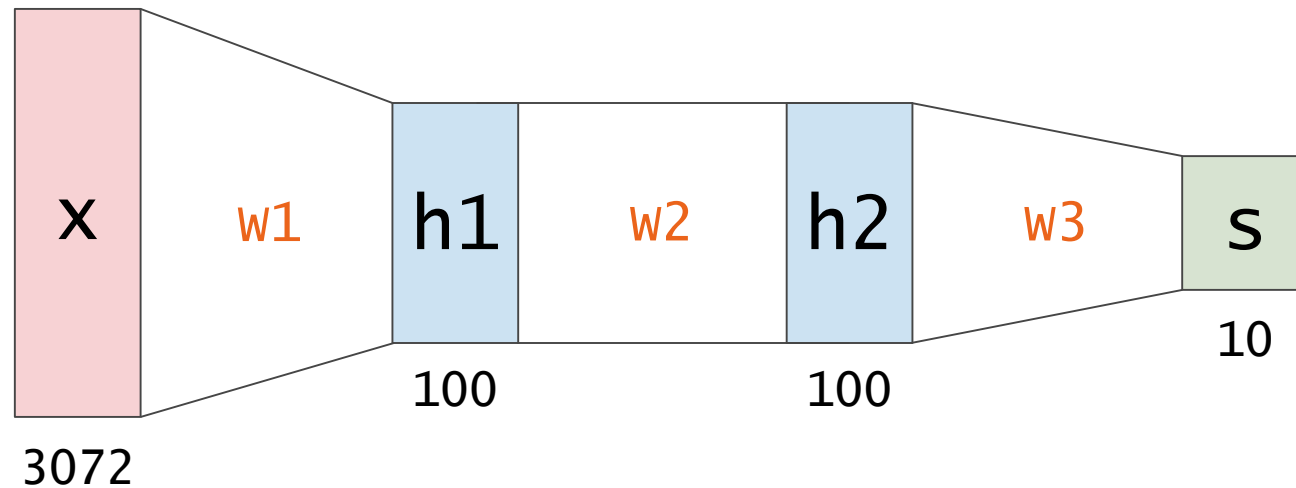


# شبکه‌های عمیق

۴

$f = Wx$  (قبلاً) تابع امتیاز خطی:

$f = W_3 \max(0, W_2 \max(0, W_1 x))$  (اکنون) یا شبکه عصبی ۳ لایه:



# شبکه‌های عصبی

۵

□ پیاده‌سازی آموزش یک شبکه عصبی ۲ لایه. [در ۱۱ خط]

```
01. X = np.array([ [0,0,1], [0,1,1], [1,0,1], [1,1,1] ])
02. y = np.array([[0,1,1,0]]).T
03. syn0 = 2*np.random.random((3,4)) - 1
04. syn1 = 2*np.random.random((4,1)) - 1
05. for j in xrange(60000):
06.     l1 = 1/(1+np.exp(-(np.dot(X,syn0))))
07.     l2 = 1/(1+np.exp(-(np.dot(l1,syn1))))
08.     l2_delta = (y - l2)*(l2*(1-l2))
09.     l1_delta = l2_delta.dot(syn1.T) * (l1 * (1-l1))
10.     syn1 += l1.T.dot(l2_delta)
11.     syn0 += X.T.dot(l1_delta)
```

from @iamtrask, <http://iamtrask.github.io/2015/07/12/basic-python-network/>

# تمرین: پیاده‌سازی یک شبکه عصبی ۲ لایه

۶

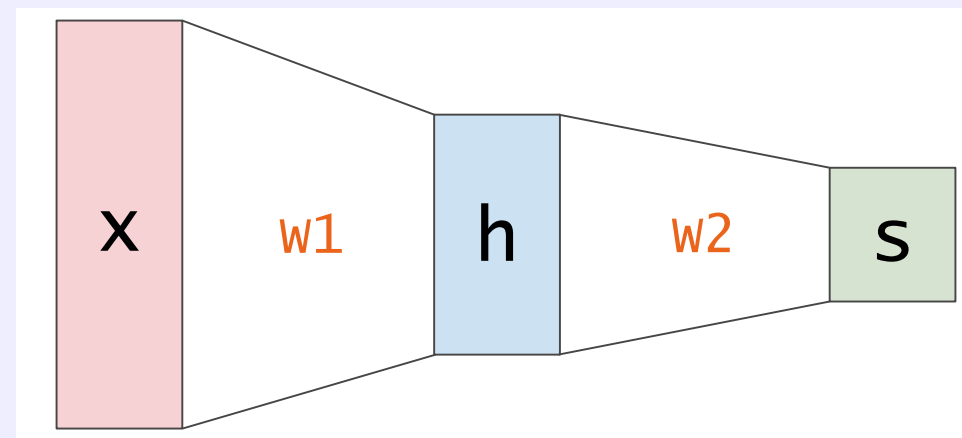
```
# receive w1, w2, b1, b2 (weights/biases), x (data)

# forward pass:

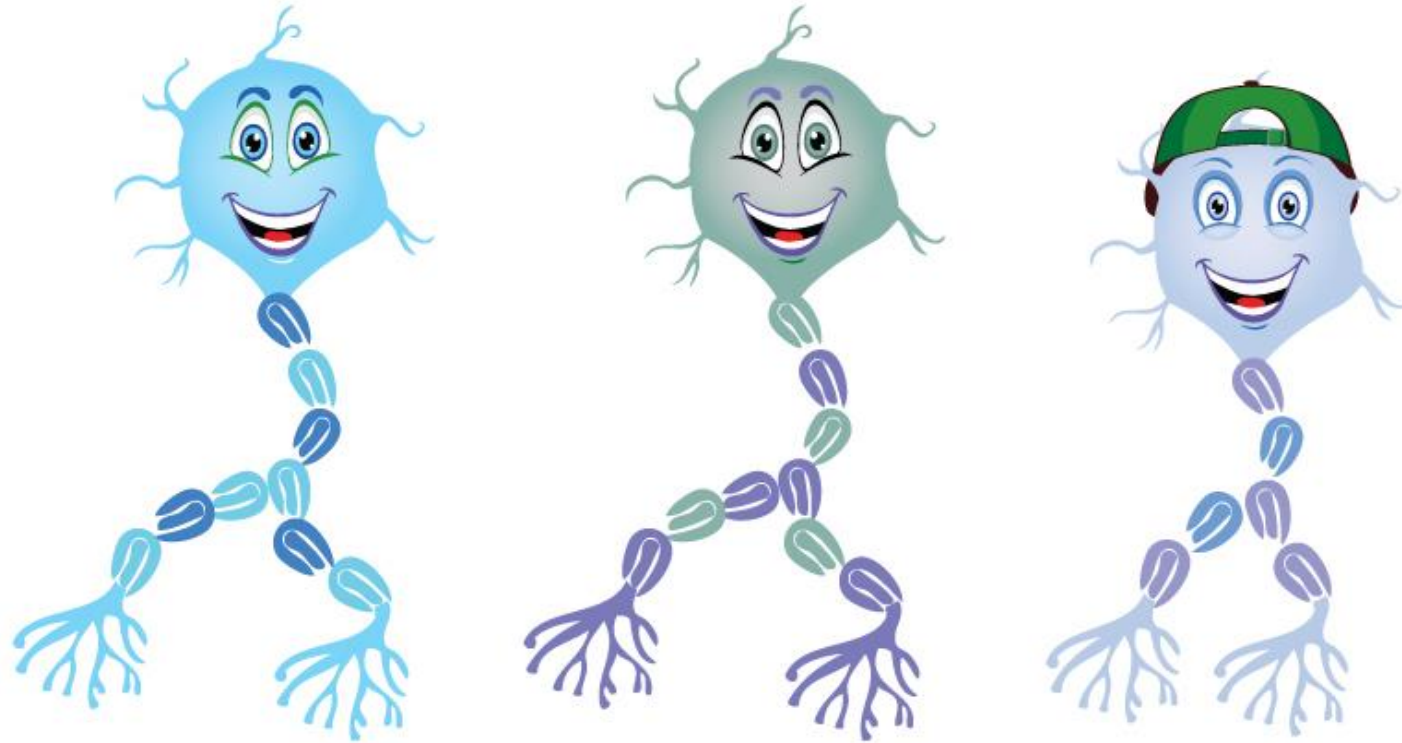
h =          #... function of x, w1, b1
scores =     #... function of h, w2, b2
loss =       #... (several lines of code to evaluate Softmax loss)

# backward pass:

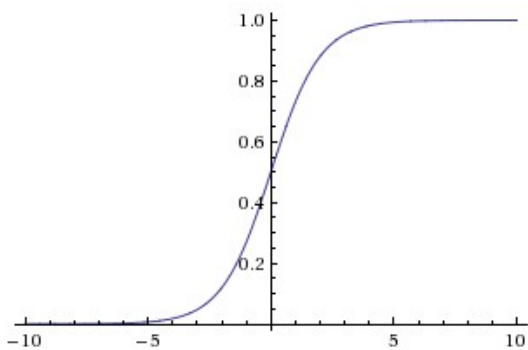
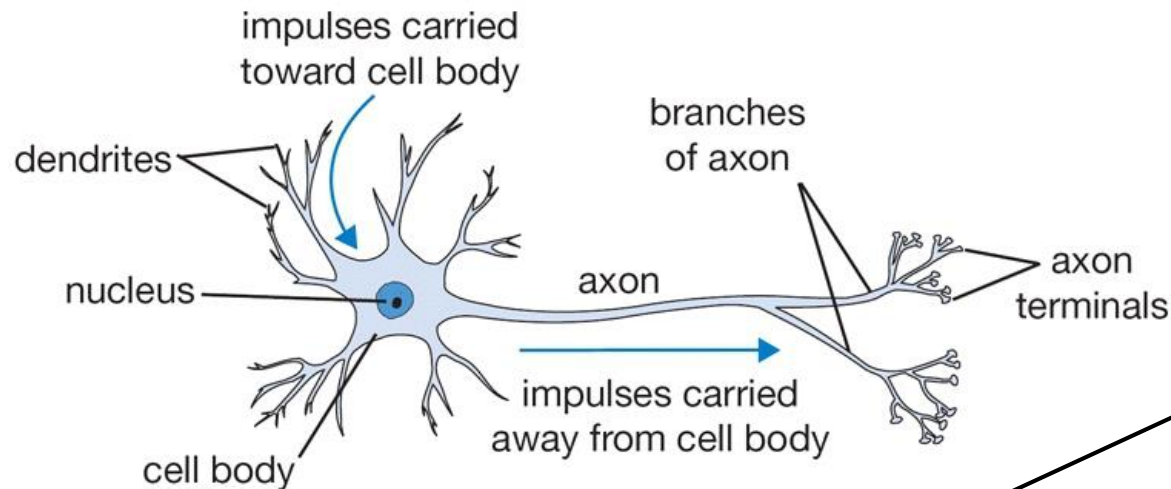
dscores =    #...
dh, dw2, db2 = #...
dw1, db1 =   #...
```



# نورون‌ها و شبکه‌های عصبی

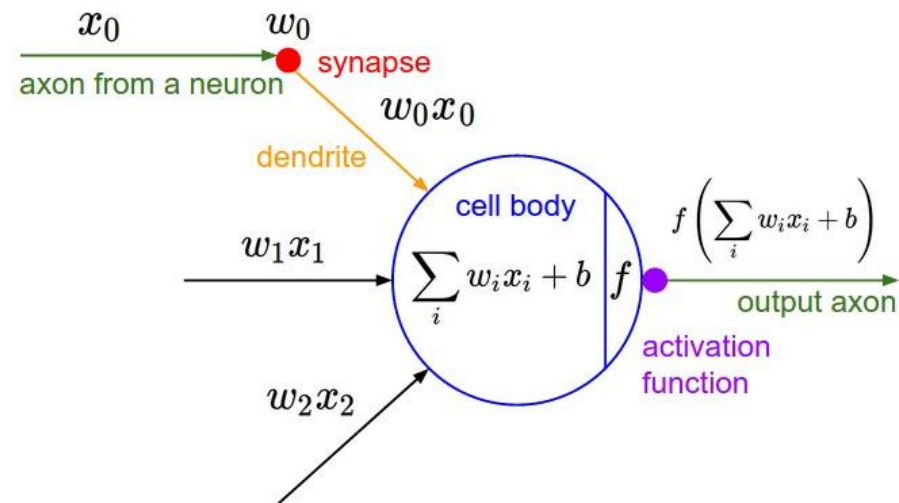


# نورون‌ها و شبکه‌های عصبی



$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

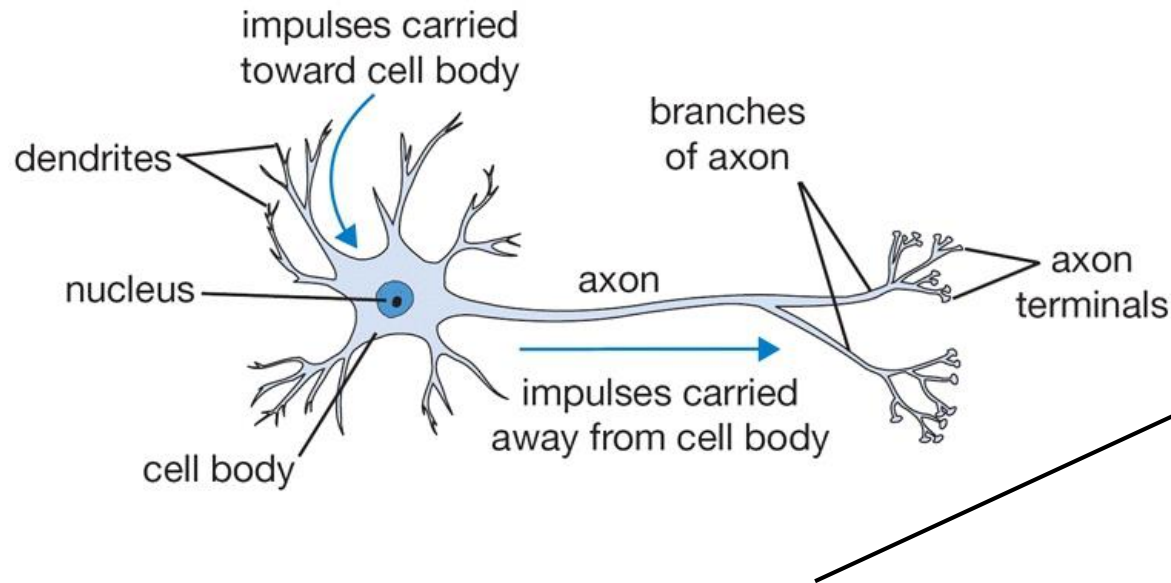
تابع فعالیت سیگموئید





# نورون‌ها و شبکه‌های عصبی

۹



```
class Neuron:
```

```
#...
```

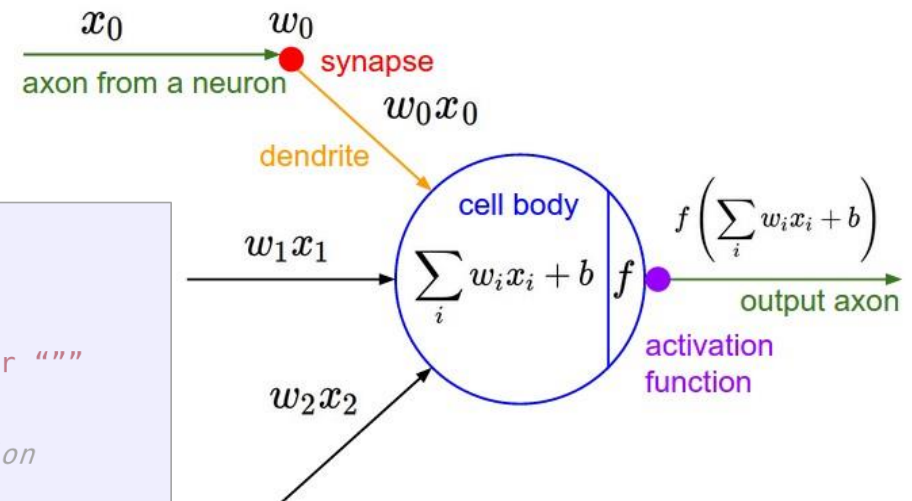
```
def neuron_tick(inputs):
```

```
    """ assume inputs and weights are 1-D numpy arrays and bias is a number """
```

```
    cell_body_sum = np.sum(inputs * self.weights) + self.bias
```

```
    firing_rate = 1.0 / (1.0 + math.exp(-cell_body_sum)) # Sigmoid function
```

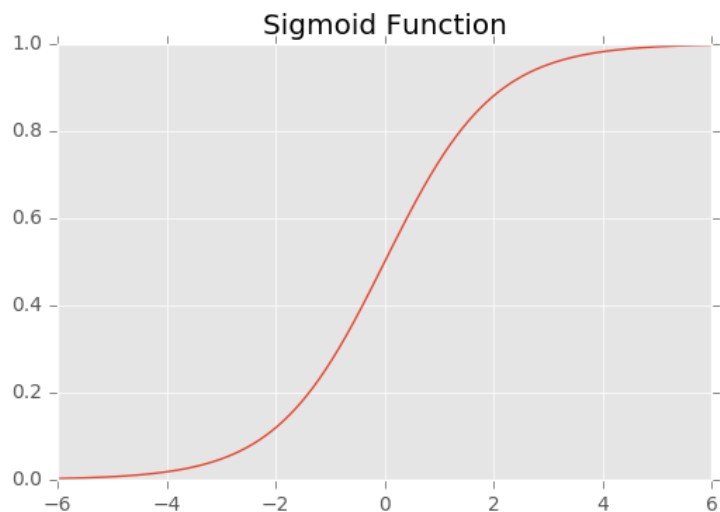
```
    return firing_rate
```



# توابع فعالیت

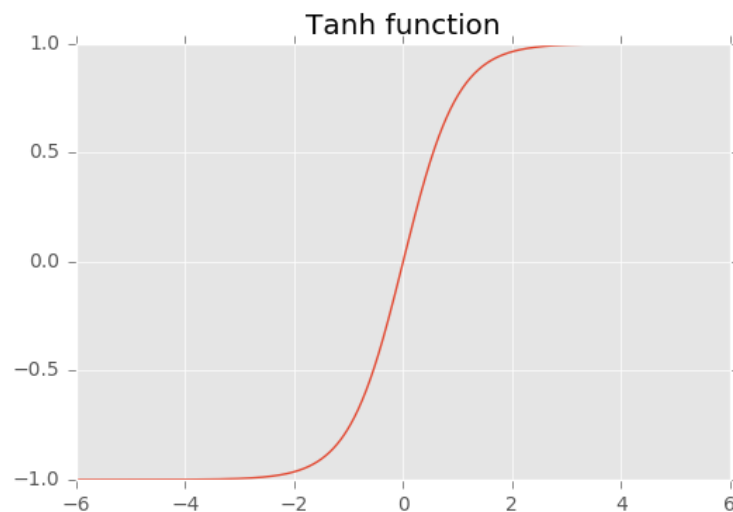
۱۰

سیگموئید



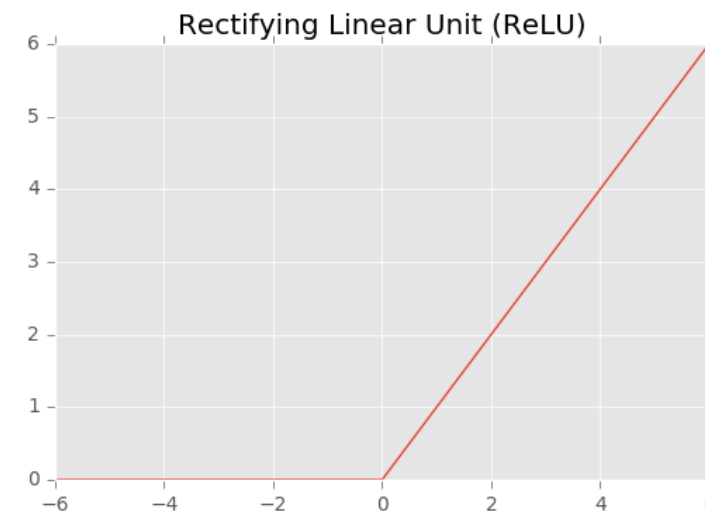
$$\sigma(x) = 1/(1 + e^{-x})$$

تانژانت هایپربولیک



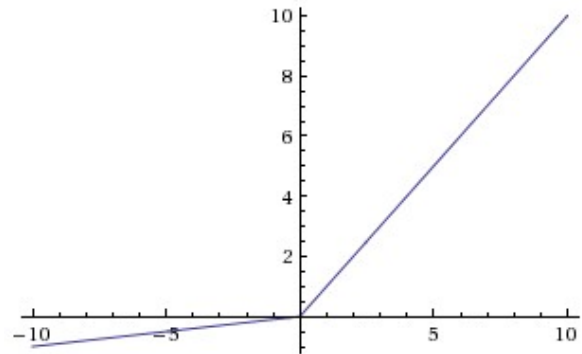
$$\tanh(x)$$

ReLU



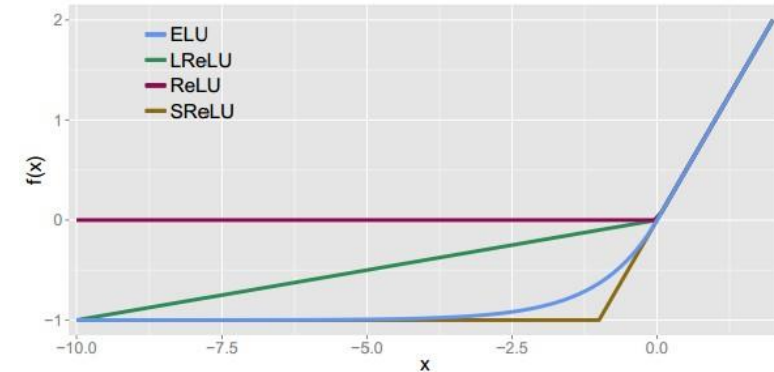
$$\max(0, x)$$

## Leaky ReLU



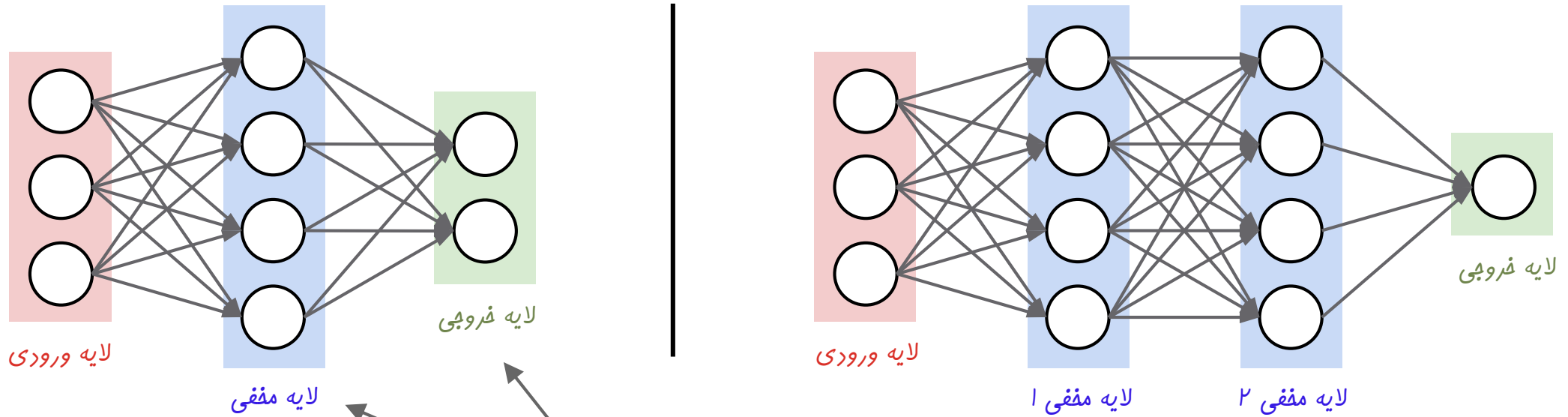
$$\max(0.1x, x)$$

## ELU



$$f(x) = \begin{cases} x, & x > 0 \\ \alpha(\exp(x) - 1), & x \leq 0 \end{cases}$$

# شبکه‌های عصبی: معماری



لایه‌های «کاملاً متصل»

شبکه عصبی ۲ لایه

[شبکه عصبی با ۱ لایه مخفی]

شبکه عصبی ۳ لایه

[شبکه عصبی با ۲ لایه مخفی]

# محاسبات روبه جلو در یک شبکه عصبی

۱۳

```
class Neuron:
```

```
    #...
```

```
    def neuron_tick(inputs):
```

```
        """ assume inputs and weights are 1-D numpy arrays and bias is a number """
```

```
        cell_body_sum = np.sum(inputs * self.weights) + self.bias
```

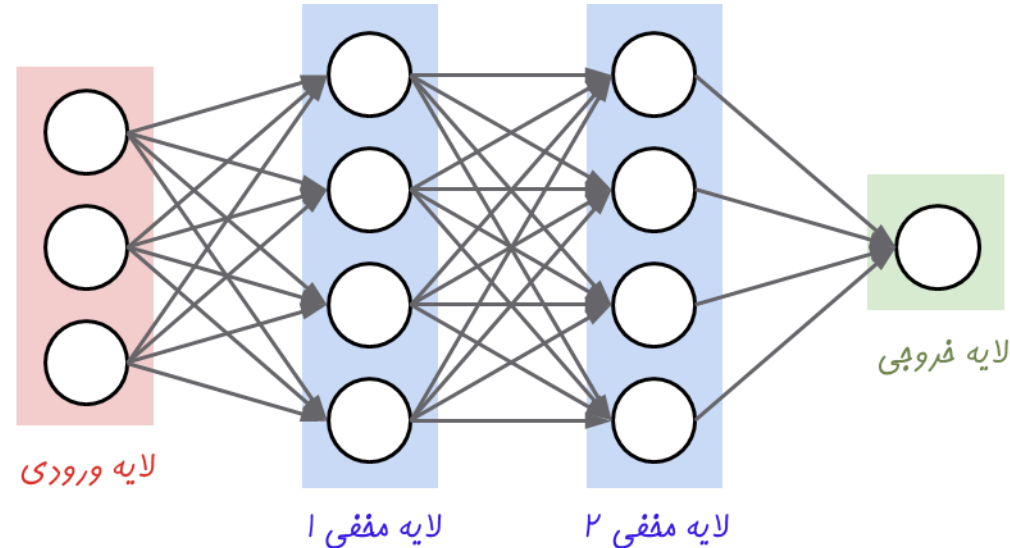
```
        firing_rate = 1.0 / (1.0 + math.exp(-cell_body_sum)) # sigmoid function
```

```
        return firing_rate
```

□ توجه. می توان محاسبات یک لایه کامل از نورون ها را به صورت کارآ پیاده سازی نمود. [پیاده سازی برداری]

# محاسبات روبه جلو در یک شبکه عصبی

۱۴



```
# forward pass of a 3-layer neural network:
```

```
f = lambda x: 1.0 / (1.0 + np.exp(-x)) # activation function (use sigmoid)
```

```
x = np.random.randn(3, 1) # random input vector of three numbers (3x1)
```

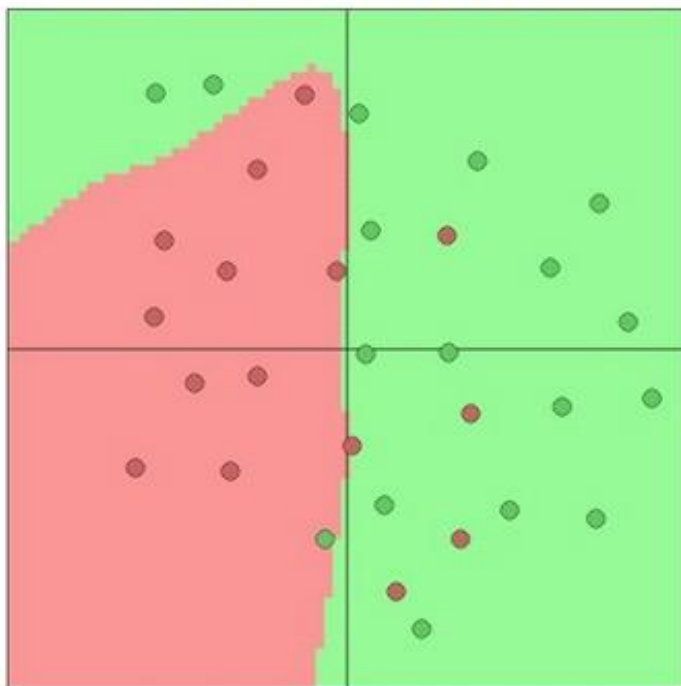
```
h1 = f(np.dot(w1, x) + b1) # calculate first hidden layer activations (4x1)
```

```
h2 = f(np.dot(w2, h1) + b2) # calculate second hidden layer activations (4x1)
```

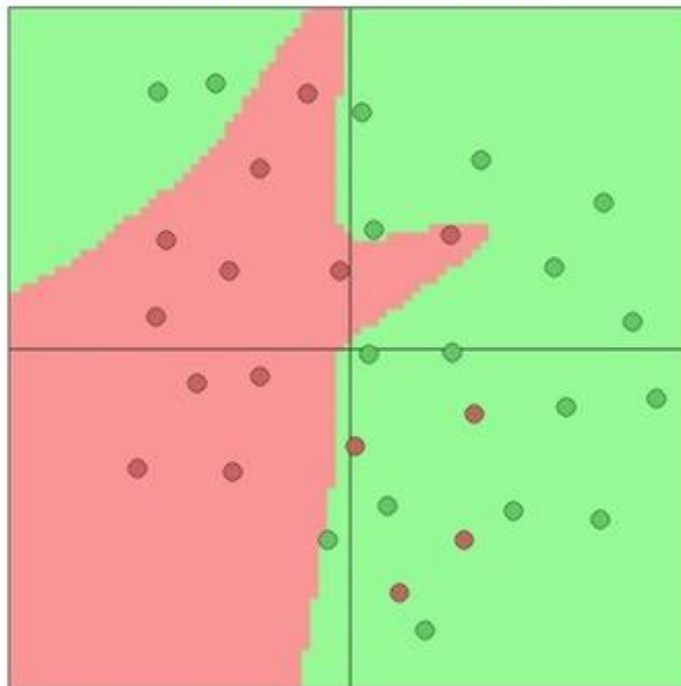
```
out = np.dot(w3, h2) + b3 # output neuron (1x1)
```

# تعیین تعداد و اندازه لایه‌ها

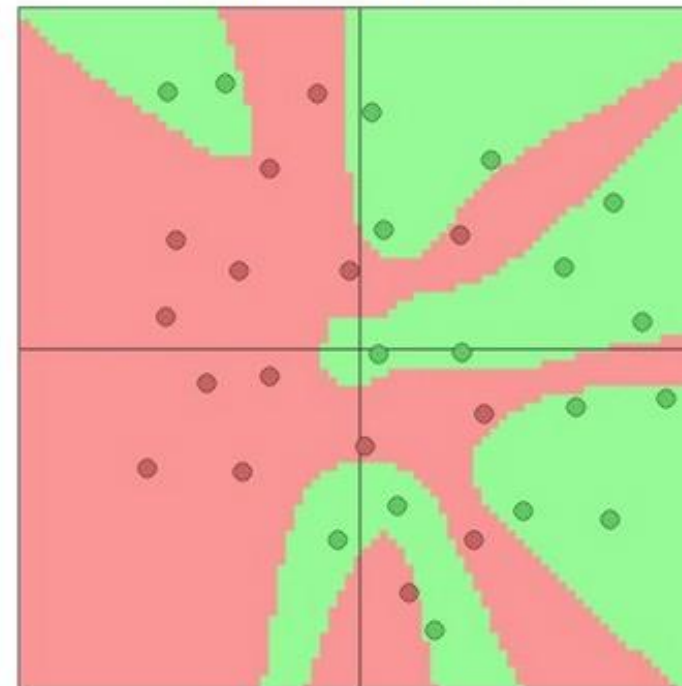
3 hidden neurons



6 hidden neurons

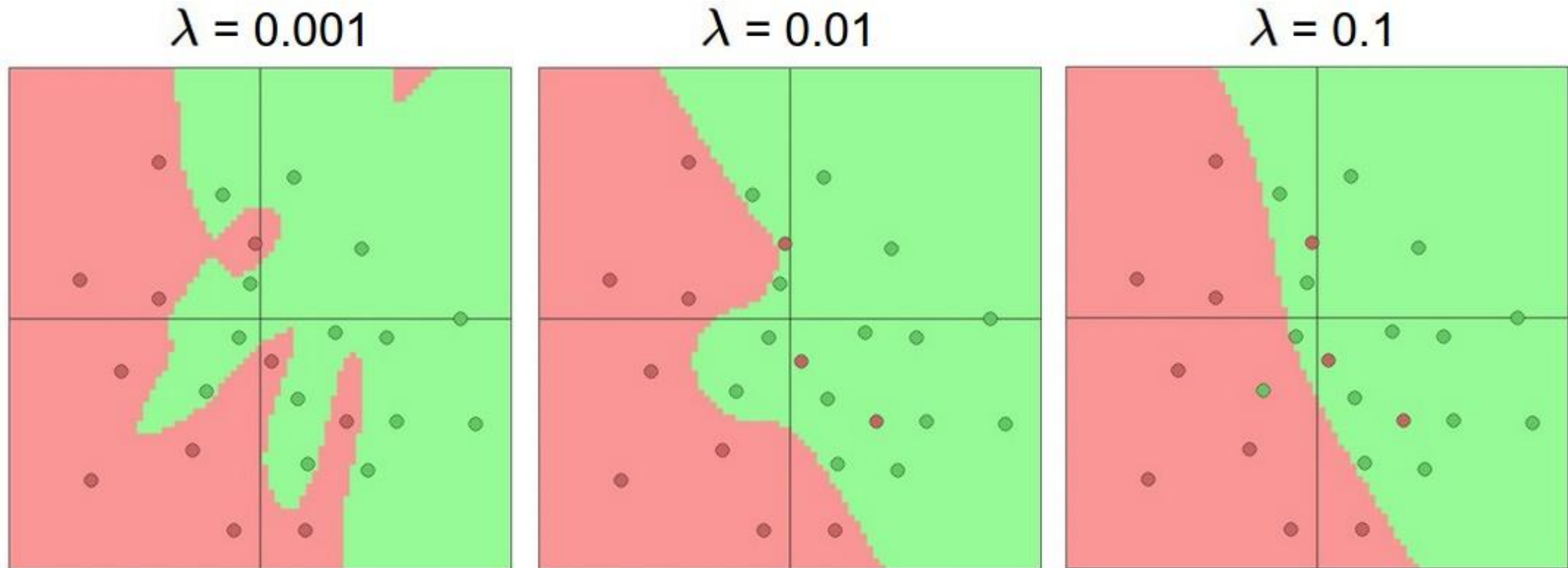


20 hidden neurons



نورون‌های بیشتر = ظرفیت بیشتر

# تعیین تعداد و اندازه لایه‌ها



□ از اندازه شبکه عصبی به عنوان تنظیم کننده استفاده نکنید و به جای آن از یک روش قوی‌تر استفاده کنید.

تنظیم L2



- در یک شبکه عصبی، نورون‌ها را در لایه‌های کاملاً متصل قرار می‌دهیم.
- استفاده از مفهوم لایه به ما امکان می‌دهد از پیاده‌سازی کارا به صورت برداری استفاده کنیم.  
[مانند ضرب ماتریسی]
- نورون‌های عصبی مصنوعی، مدل‌های بسیار ساده شده‌ای از نورون‌های زیستی هستند.
  - در واقع، شبکه‌های عصبی اصلاً عصبی نیستند!
- هر چه اندازه یک شبکه عصبی بزرگ‌تر باشد، بهتر است:
  - به شرطی که از یک تنظیم کننده قوی برای تنظیم وزن‌ها استفاده شود.

- مطالب بیشتر در مورد شبکه‌های عصبی.
- چگونگی آموزش شبکه‌های عصبی.
- تعیین مقدار مناسب وزن‌ها.

