

Neural Networks

k-Nearest Neighbor on images **never used**.

- distance metrics on level of whole images can be very unintuitive

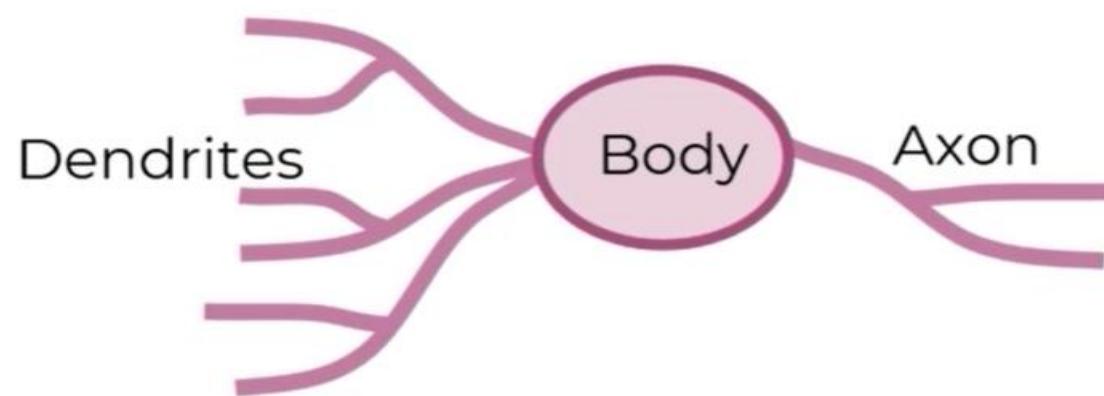


(all 3 images have same L2 distance to the one on the left)

مقدمه ای بر شبکه های عصبی

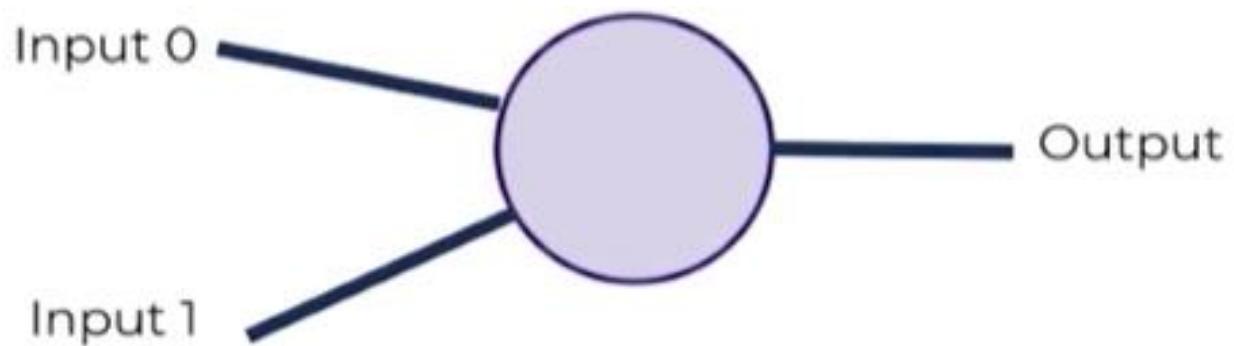
- Neurons and Activation Functions
- Cost Functions
- Gradient Descent
- Backpropagation

The biological neuron:



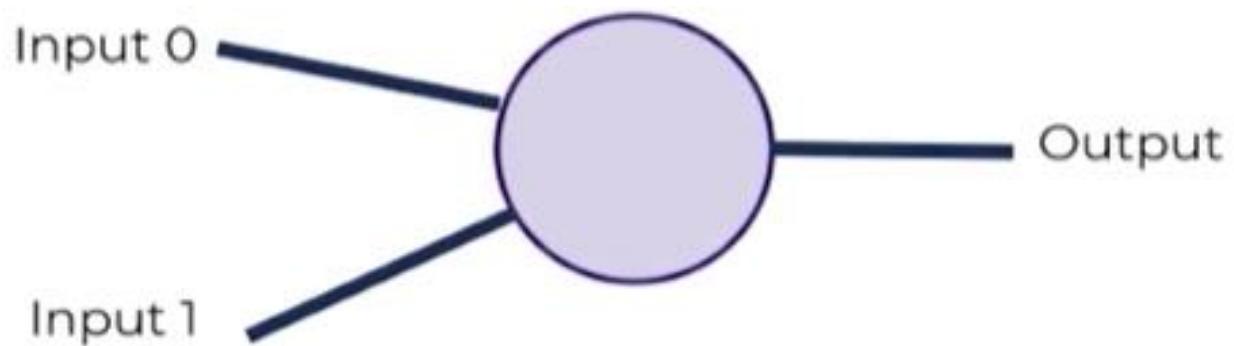
پرسپترون

- نورون های مصنوعی هم ورودی و خروجی دارند.
- این مدل ساده را پرسپترون می نامند.



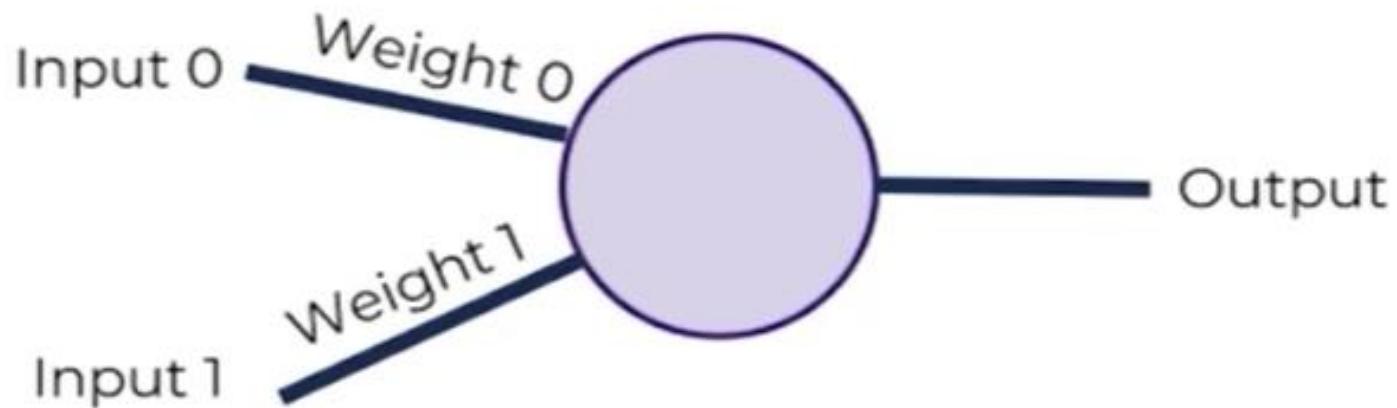
پرسپترون

- ورودی، مقادیر ویژگی ها (features) هستند.



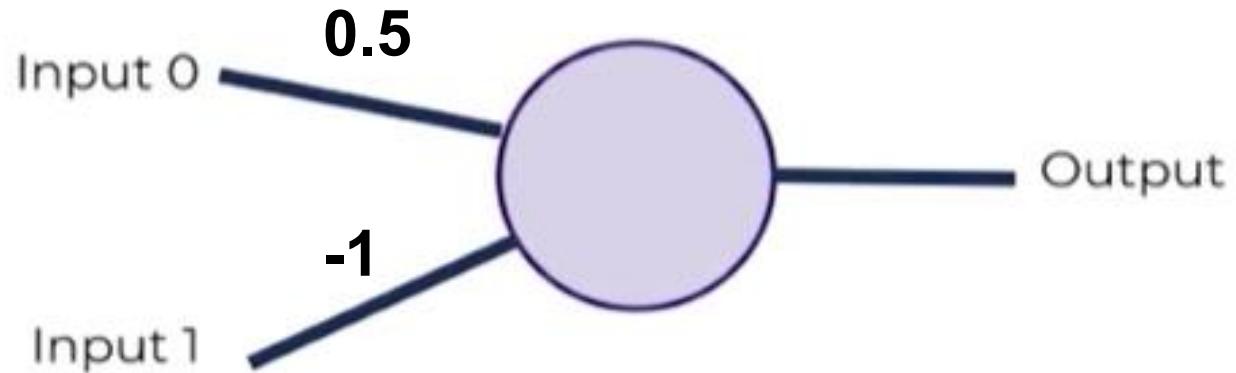
پرسپترون

- ورودی ها در وزن ها ضرب می شوند.



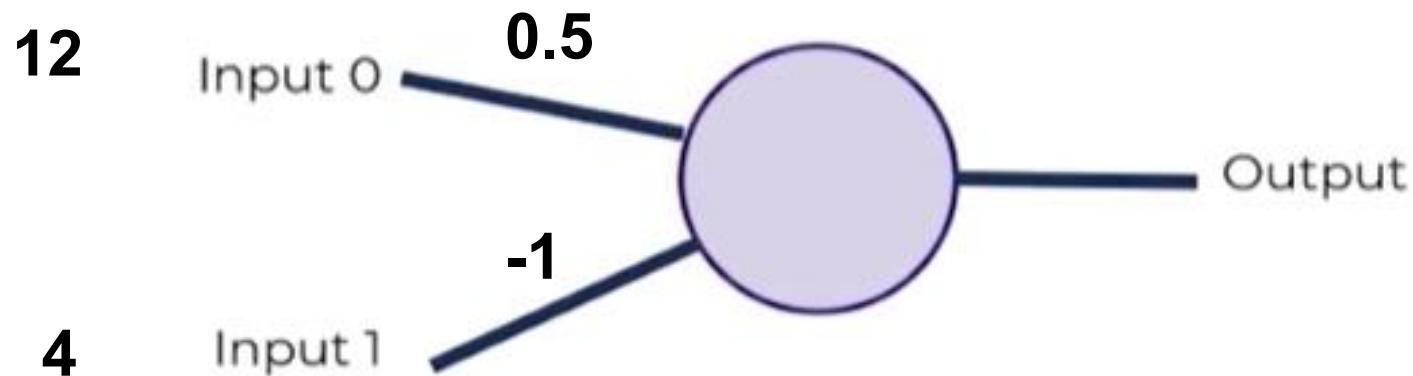
پرسپترون

- وزن ها به صورت تصادفی و رندوم مقداردهی اولیه می شوند.



پرسپترون

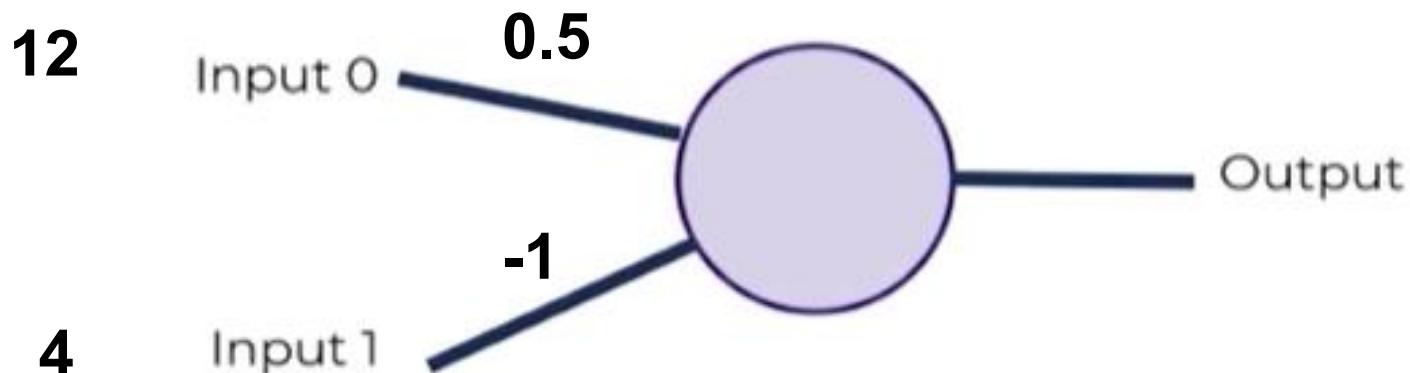
- ورودی ها در وزن های متناظر ضرب می شوند.



پرسپترون

- ورودی ها در وزن های متناظر ضرب می شوند.

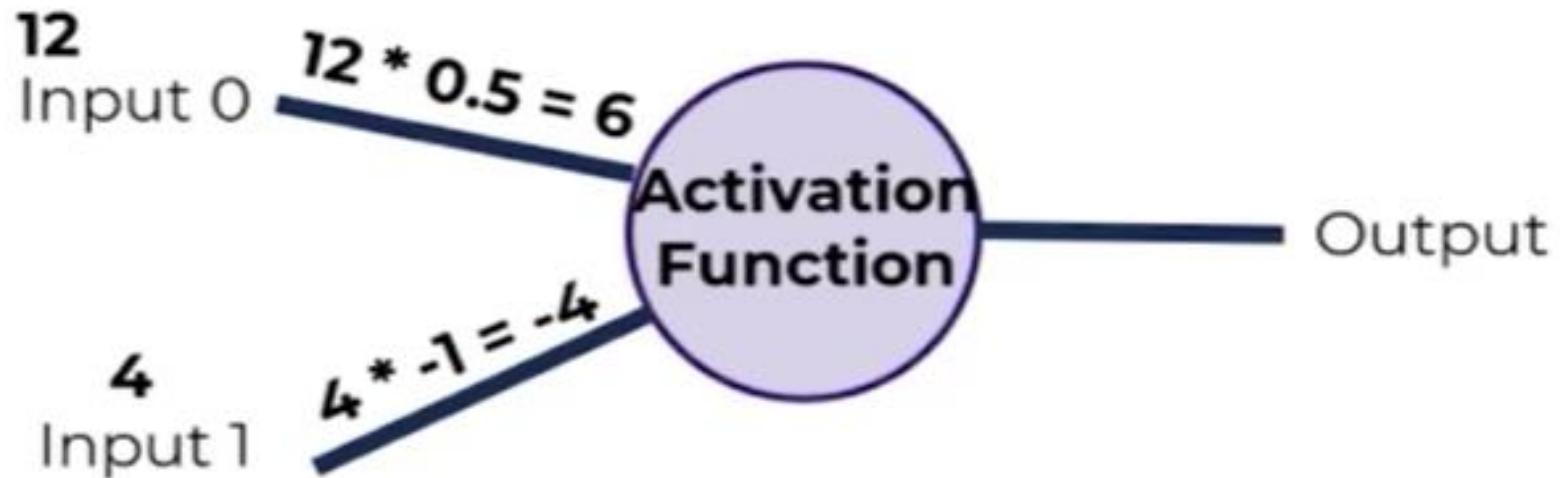
$$12 * 0.5 = 6$$



$$4 * -1 = -4$$

پرسپترون

- یک تابع فعالیت بر روی نتایج اعمال می گردد.



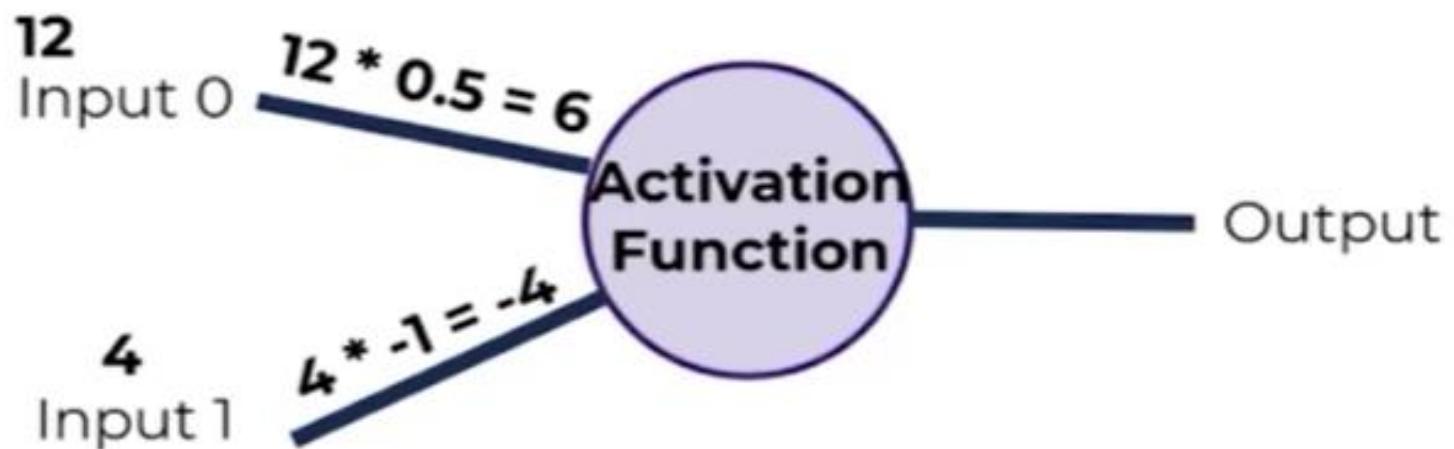
پرسپترون

- انتخاب های زیادی برای تابع فعالیت داریم.

- ساده ترین تابع فعالیت:

- اگر مجموع مثبت بود: ۱

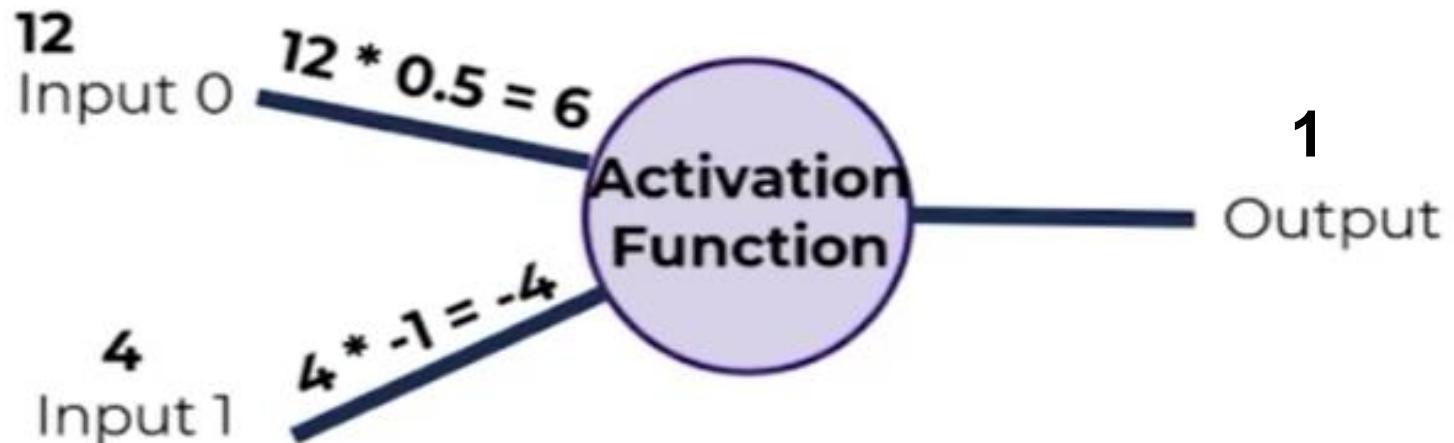
- در غیر اینصورت: ۰



پرسپترون

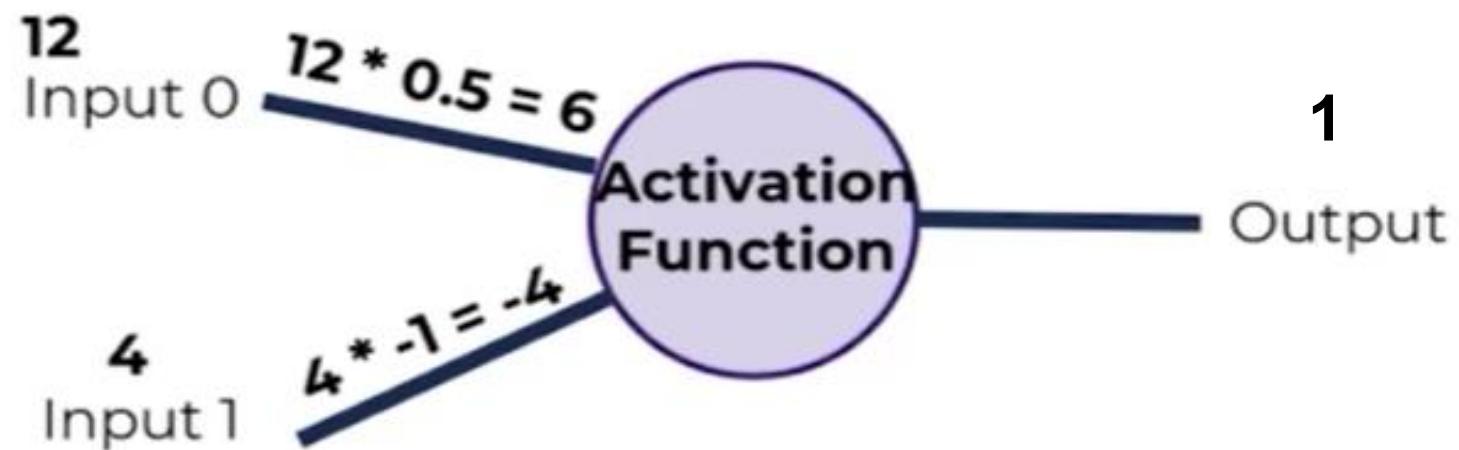
- در این مثال داریم:
 $6 - 4 = 2$

- پس خروجی برابر است با ۱



پر سپترون

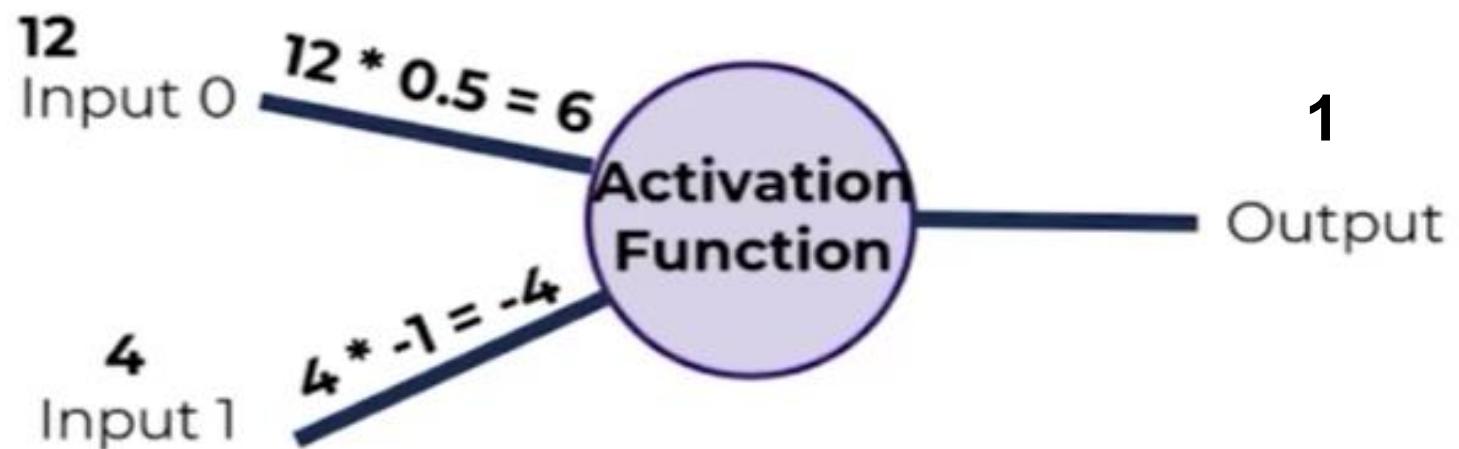
یک مشکل ! اگر ورودی ۰ باشد ؟ ■



پرسپترون

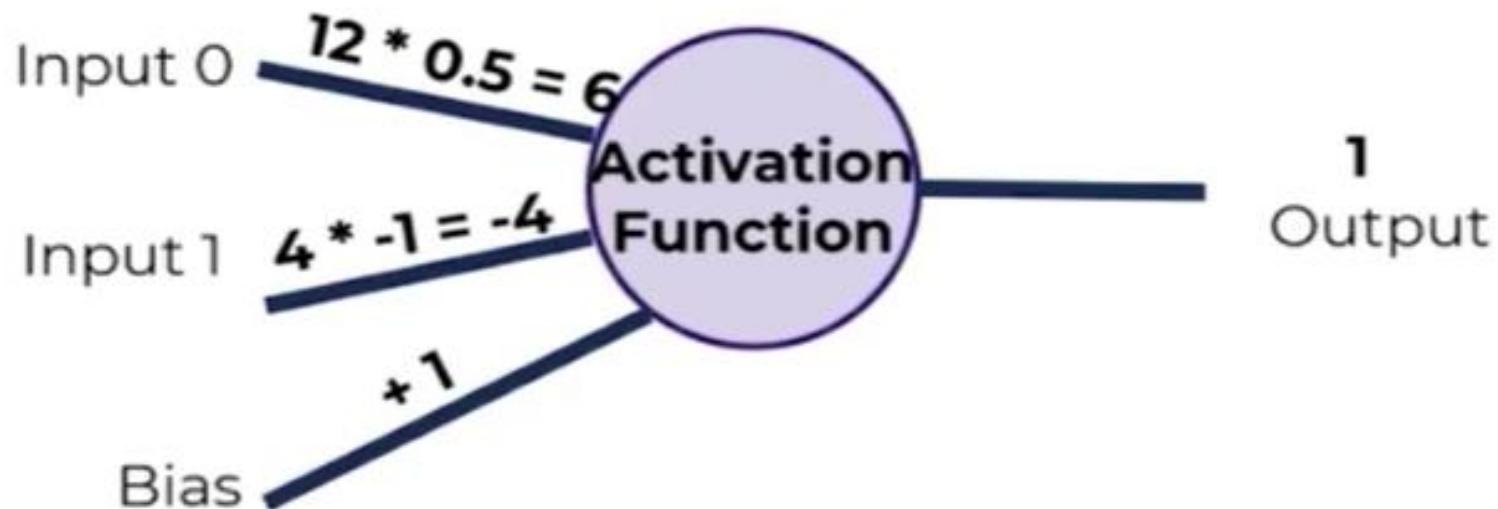
- یک مشکل ! اگر ورودی ۰ باشد ؟!

- مستقل از وزن ها خروجی ۰ می شود!



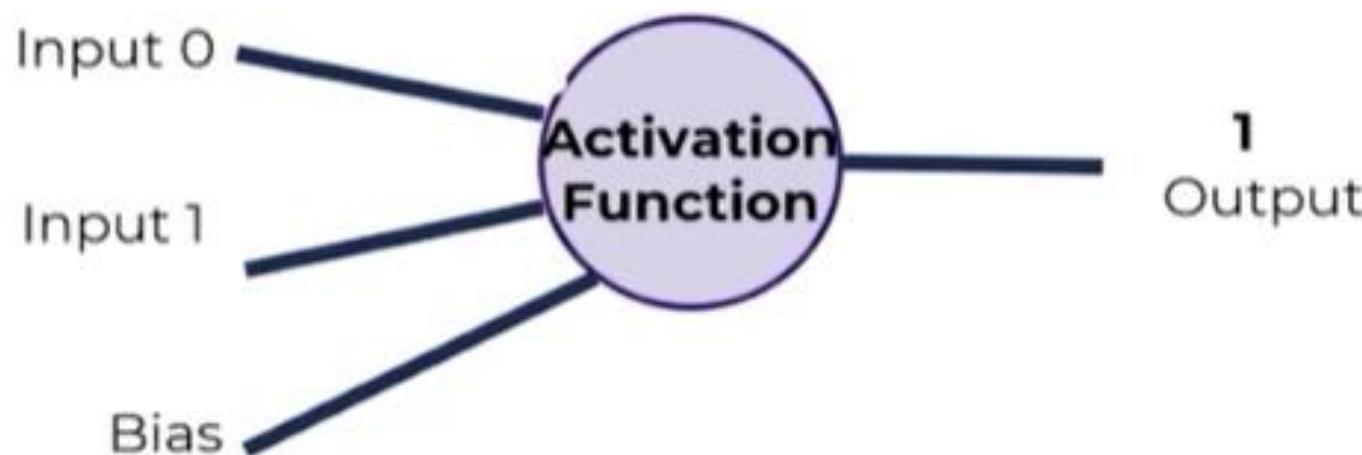
پرسپترون

- این مشکل را با افزودن بایاس رفع می کنیم.



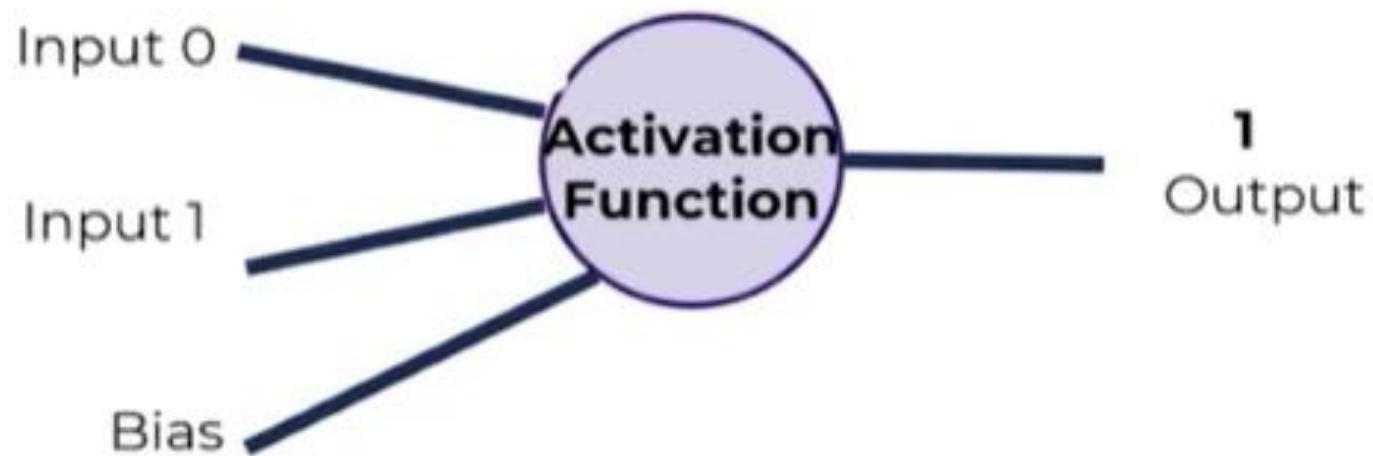
پرسپترون

- اگر بخواهیم پرسپترون را به صورت ریاضی بنویسیم...



پرسپترون

اگر بخواهیم پرسپترون را به صورت ریاضی بنویسیم... ■



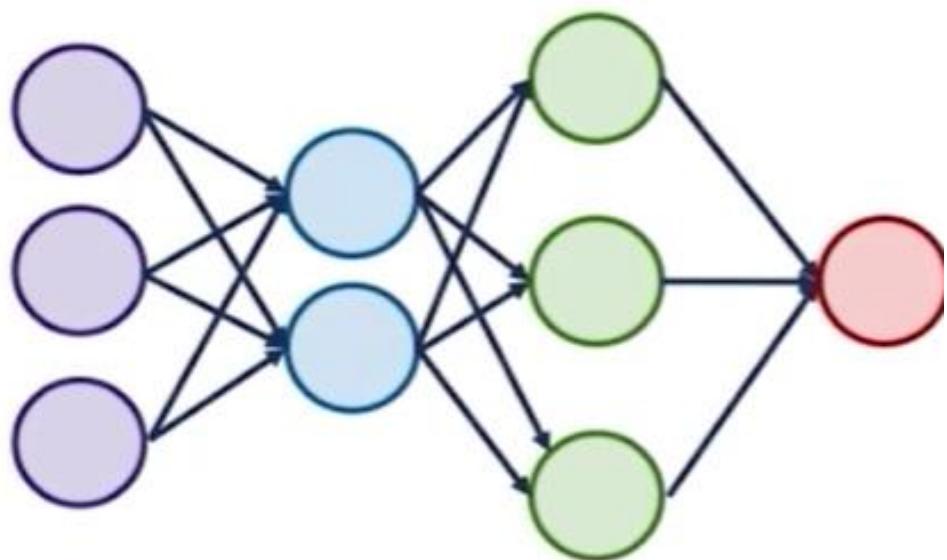
$$\sum_{i=0}^n w_i x_i + b$$

Introduction to Neural Networks

مقدمه ای بر شبکه های عصبی

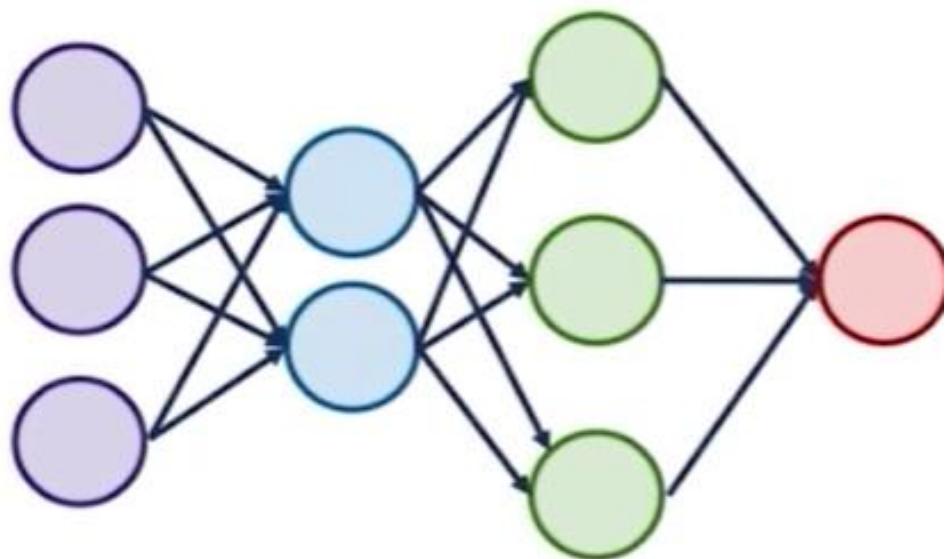
مقدمه ای بر شبکه های عصبی

- شبکه ای از چندین پرسپترون



مقدمه ای بر شبکه های عصبی

- شبکه ای از چندین پرسپترون



Input Layer

2 hidden Layer

Output Layer

مقدمه ای بر شبکه های عصبی - لایه ها

۱. لایه ورودی:

- مقادیر حقیقی داده ها

۲. لایه پنهان:

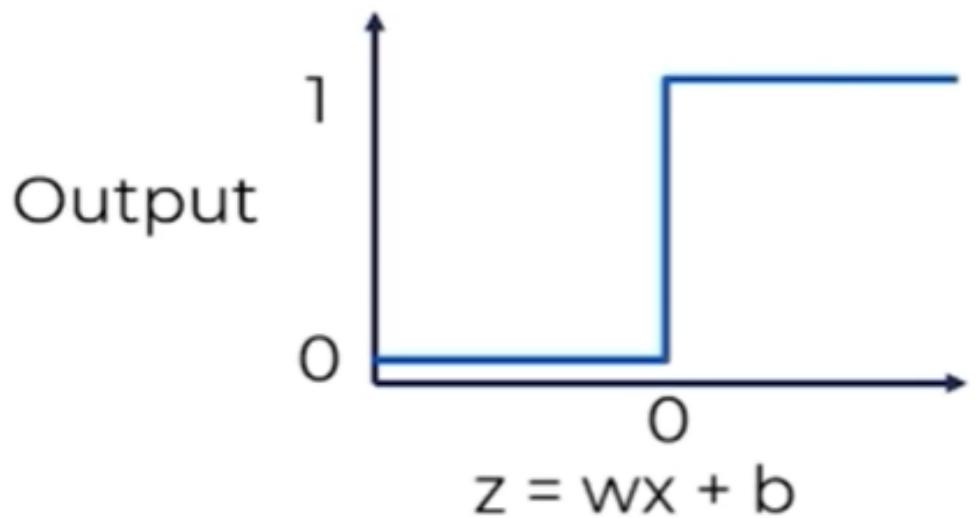
- لایه های بین ورودی و خروجی
- به ۳ یا بیشتر لایه معمولاً شبکه عمیق می گوییم.

۳. لایه خروجی:

- تخمین نهایی شبکه از خروجی است.

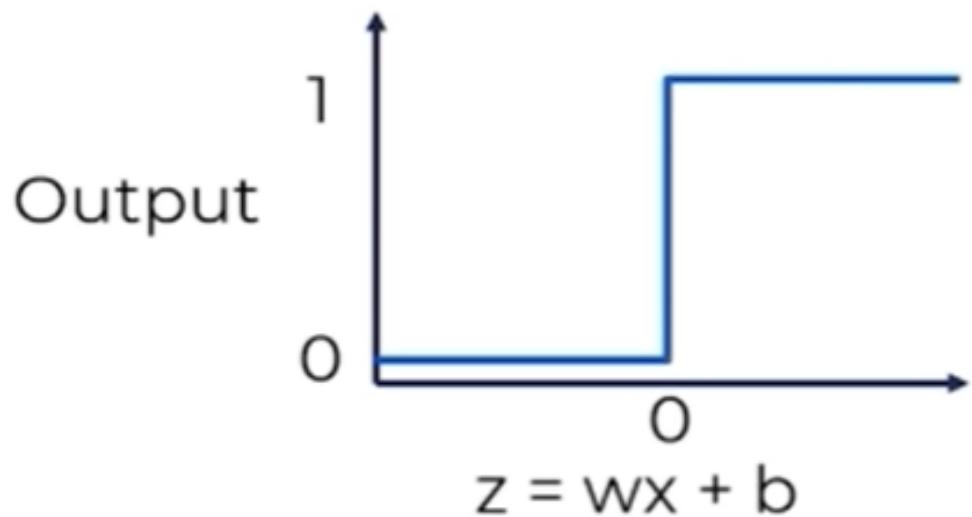
مقدمه ای بر شبکه های عصبی - تابع فعالیت

تابع فعالیت ساده ای که تاکنون دیدیم.



مقدمه ای بر شبکه های عصبی - تابع فعالیت

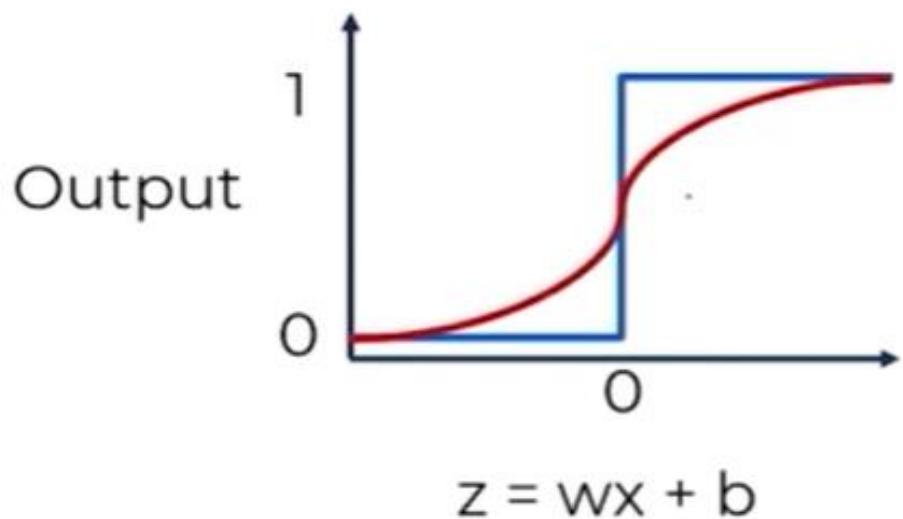
تابع فعالیت ساده ای که تاکنون دیدیم.



تغییرات کوچک منعکس نمی شوند.

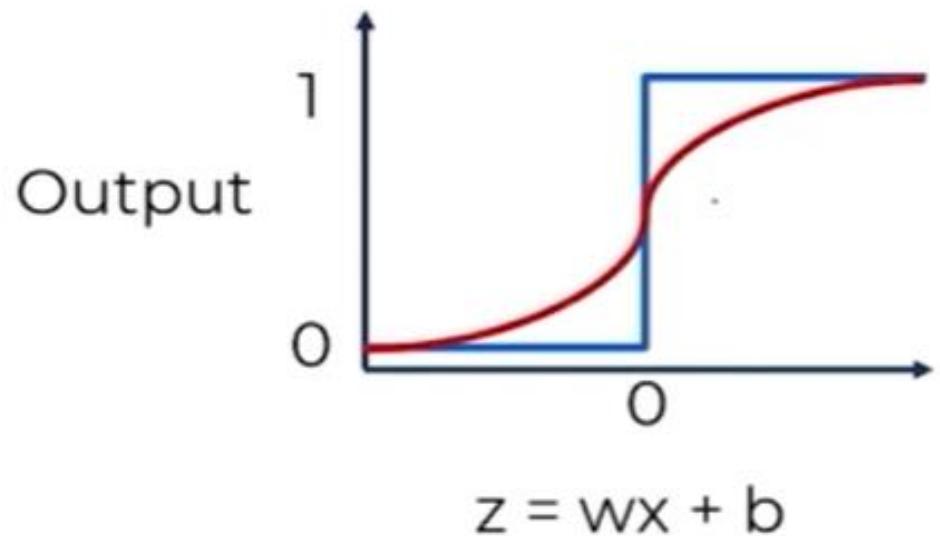
مقدمه ای بر شبکه های عصبی - تابع فعالیت

اگر تابع پویاتر داشتیم بهتر میشد، مثل تابع قرمز رنگ!



مقدمه ای بر شبکه های عصبی - تابع فعالیت

تابع سیگموئید

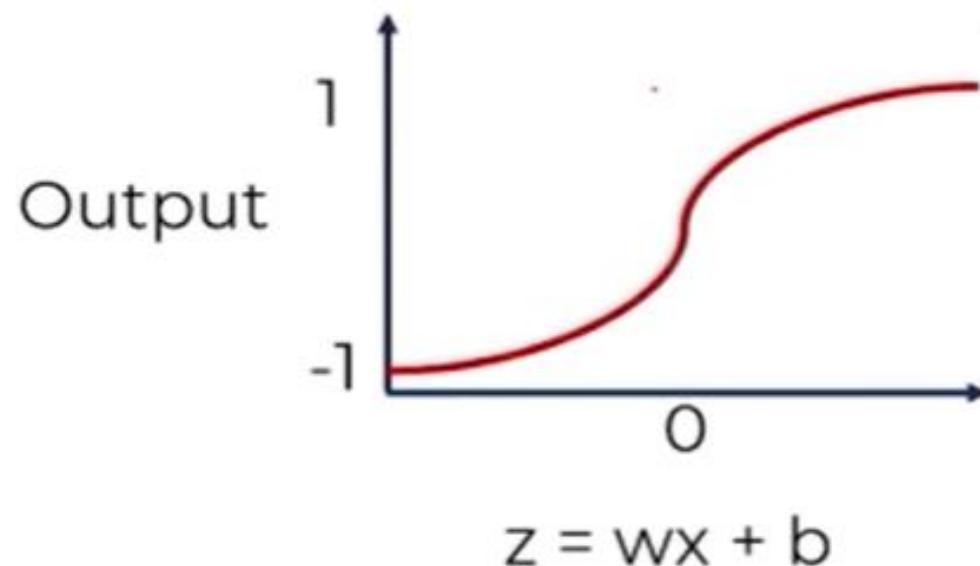


$$f(x) = \frac{1}{1 + e^{-(x)}}$$

مقدمه ای بر شبکه های عصبی - تابع فعالیت

بسته به کار ممکن است از توابع فعالیت متفاوتی استفاده کنیم.

- Hyperbolic Tangent: $\tanh(z)$



$$\cosh x = \frac{e^x + e^{-x}}{2}$$

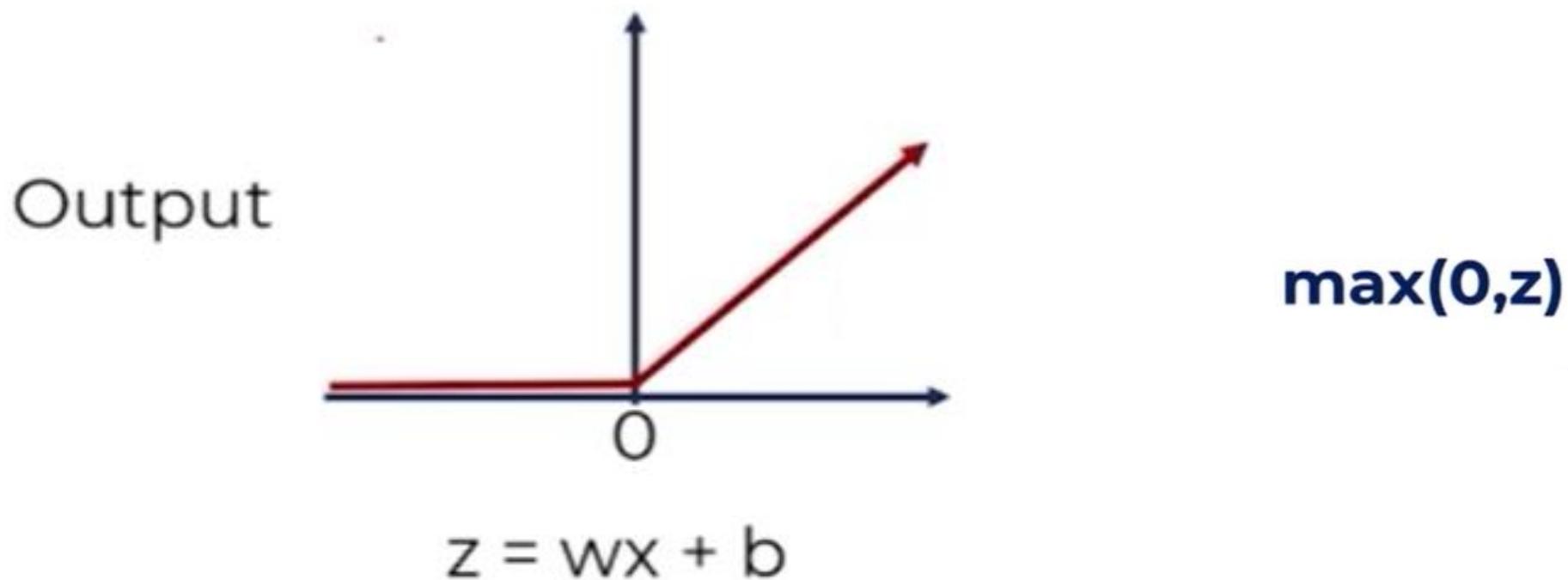
$$\sinh x = \frac{e^x - e^{-x}}{2}$$

$$\tanh x = \frac{\sinh x}{\cosh x}$$

مقدمه ای بر شبکه های عصبی - تابع فعالیت

واحد یکسوساز خطی **ReLU**

- Rectified Linear Unit

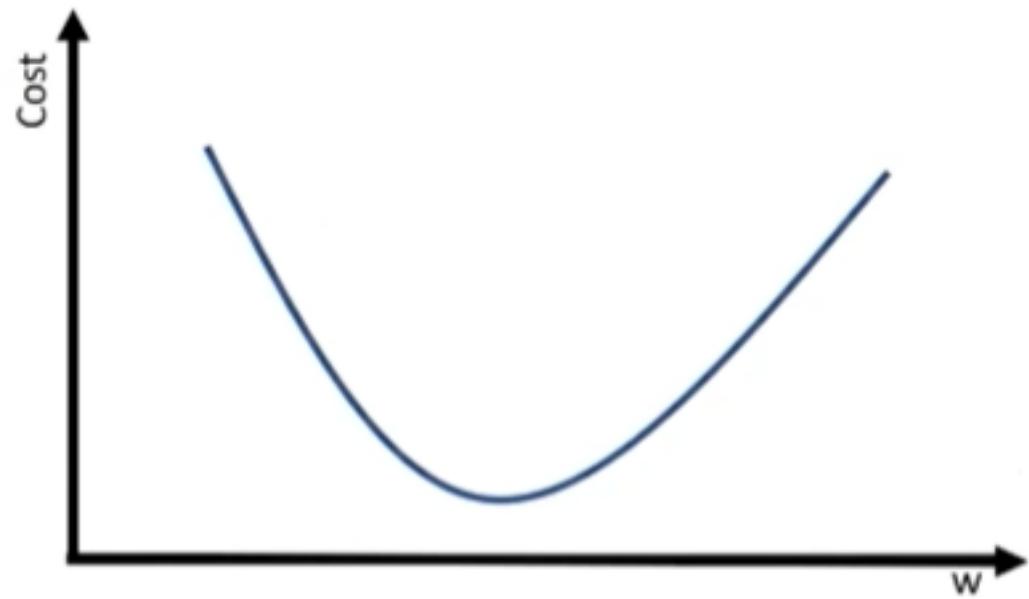


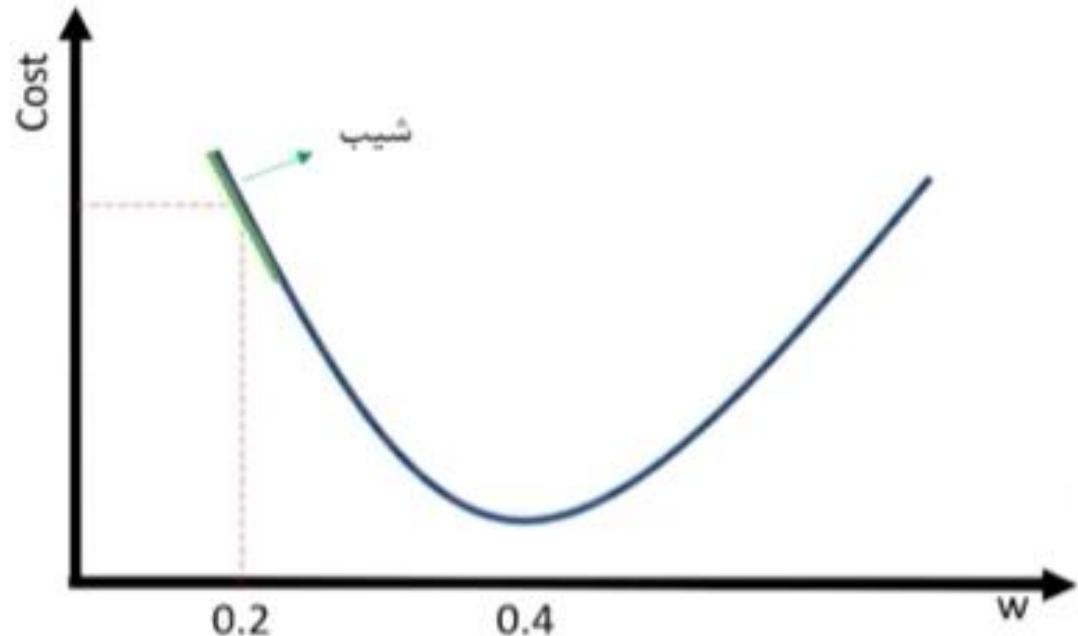
Gradient Descent and Backpropagation

پس انتشار خطای و گرادیان کاهشی

پس انتشار خطأ و گرادیان کاهشی

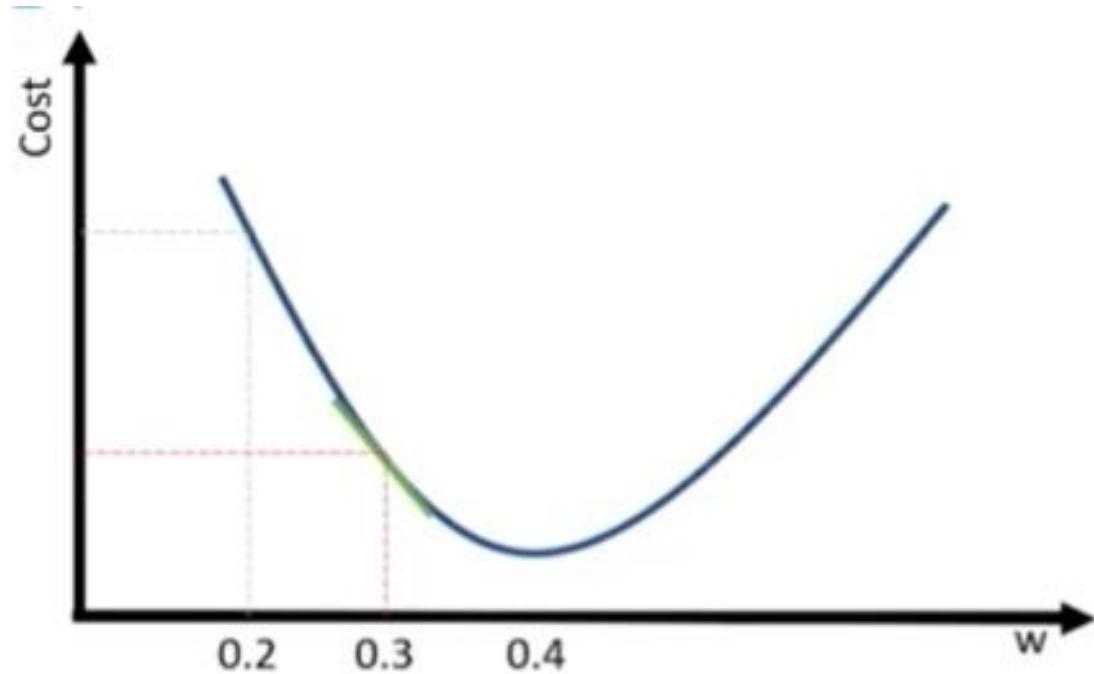
- گرادیان کاهشی یا Gradient descent یک الگوریتم بهینه سازی برای یافتن حداقل یک تابع است.
- برای یافتن کمینه‌ی محلی یک تابع با استفاده از این الگوریتم، گام‌هایی متناسب با منفی گرادیان تابع در محل فعلی برداشته خواهد شد.





$$W = W - \text{شیب}$$

$$W = 0.2 - (-0.1)$$



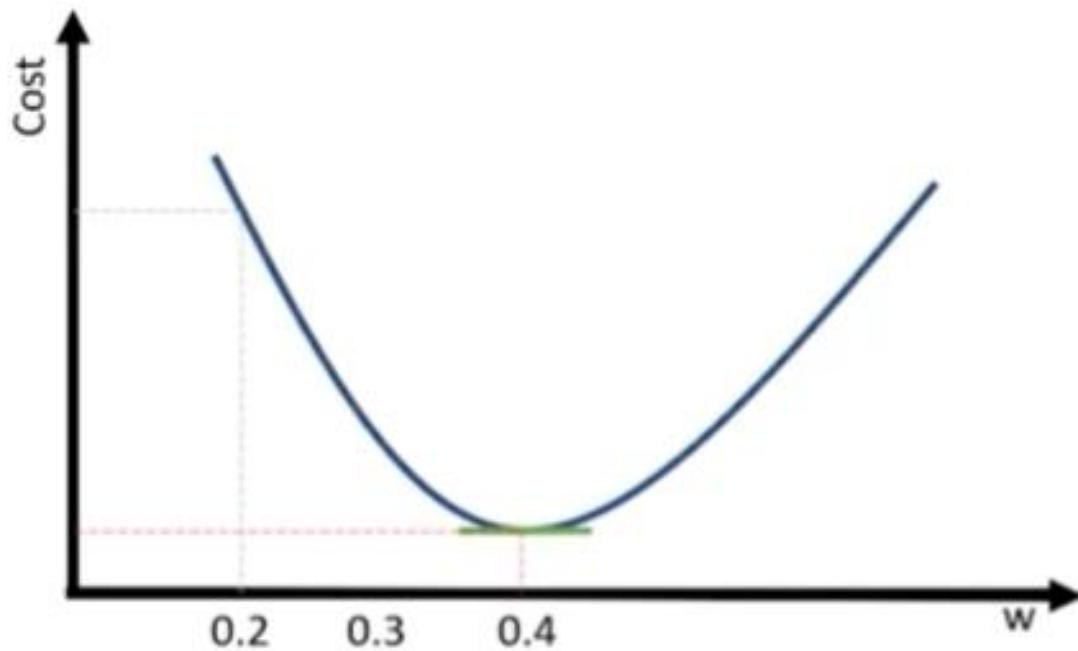
$$W = W^* - \frac{\lambda}{2}$$

$$W = 0.2 - (-0.1)$$

$$W = 0.3$$

$$W = W^* - \frac{\lambda}{2}$$

$$W = 0.3 - (-0.1)$$



$$W = W - \text{نسبة}$$

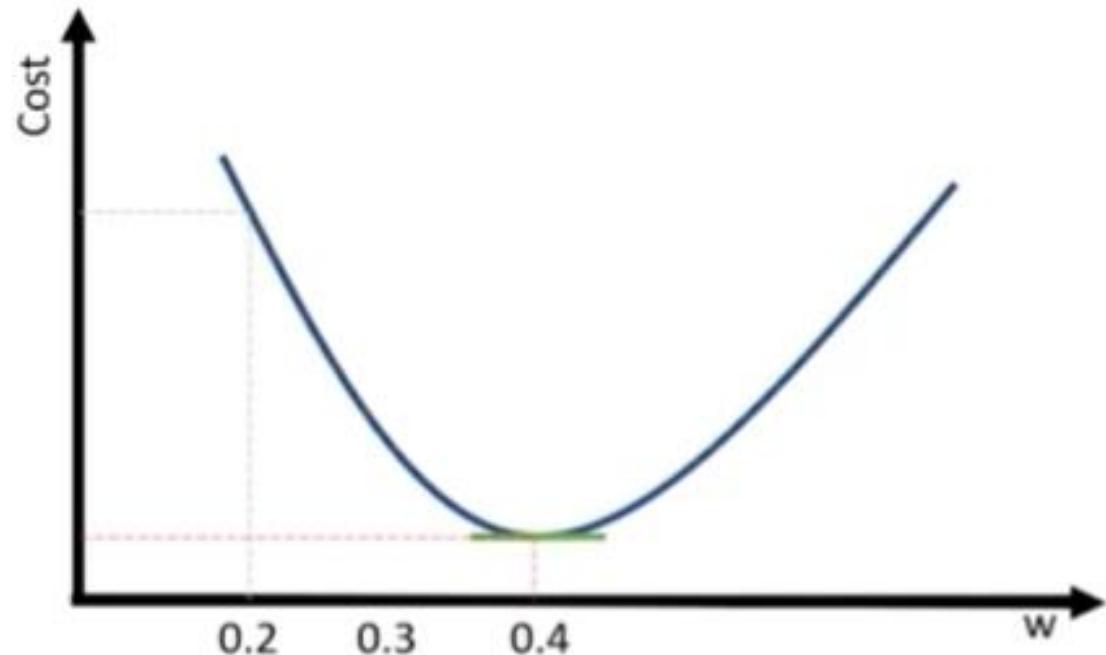
$$W = 0.2 - (-0.1)$$

$$W = 0.3$$

$$W = W - \text{نسبة}$$

$$W = 0.3 - (-0.1)$$

$$W = 0.4$$



$$W = W - \text{شيء}$$

$$W = 0.2 - (-0.1)$$

$$W = 0.3$$

$$W = W - \text{شيء}$$

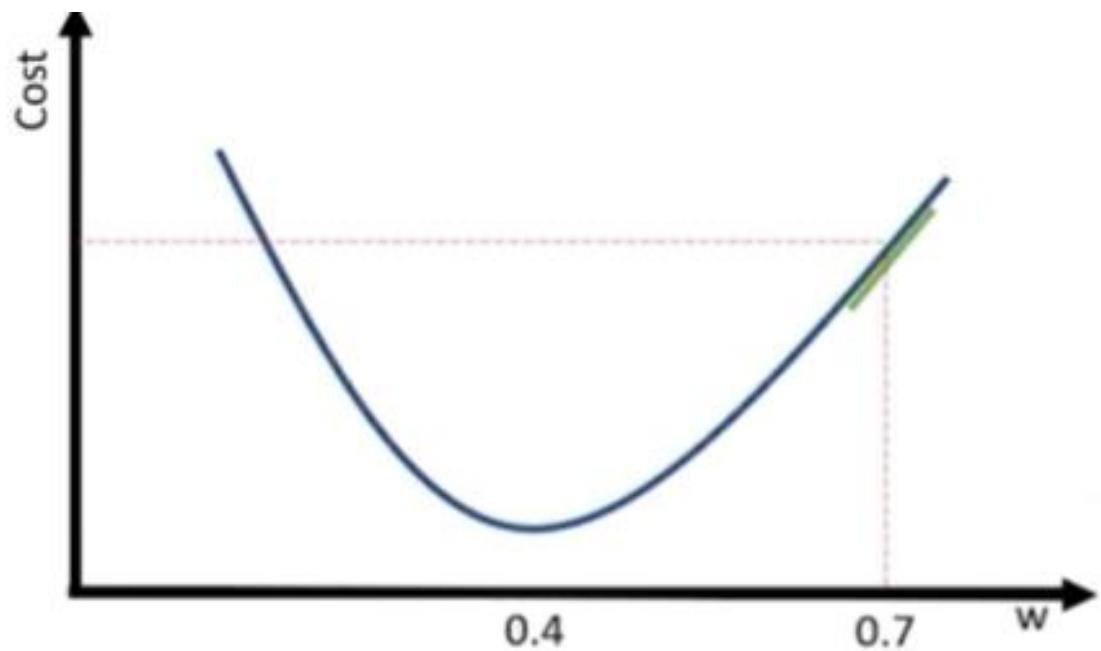
$$W = 0.3 - (-0.1)$$

$$W = 0.4$$

$$W = W - \text{شيء}$$

$$W = 0.4 - 0$$

$$W = 0.4$$

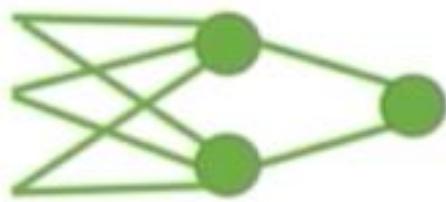


$$W = W^* - \frac{1}{2} \gamma$$

$$W = 0.7 - 0.3$$

$$W = 0.4$$

چرا Gradient decent و نه derivative decent



$$C(W_1, W_2, W_3, \dots, W_n)$$

$$C'$$

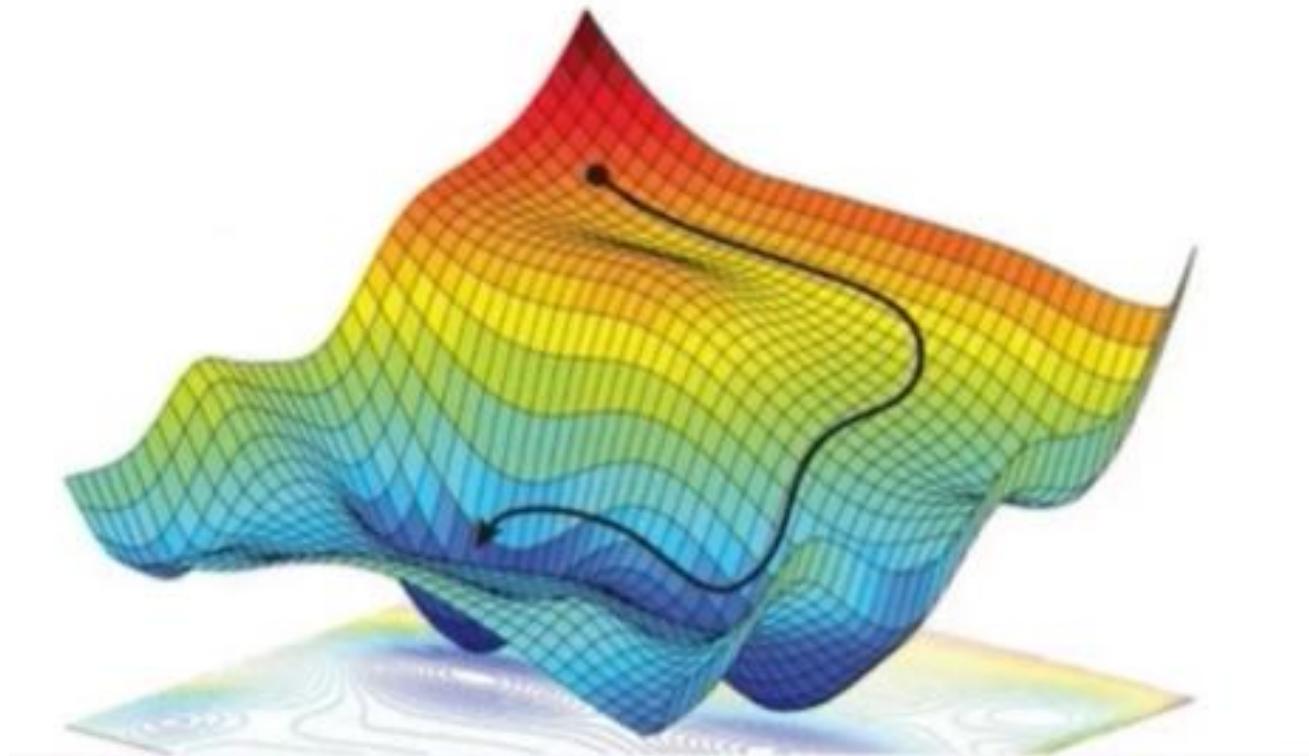
$$\frac{\partial C}{\partial W_1}$$

$$\frac{\partial C}{\partial W_2}$$

⋮

$$\frac{\partial C}{\partial W_3}$$

چرا Gradient decent و نه derivative decent

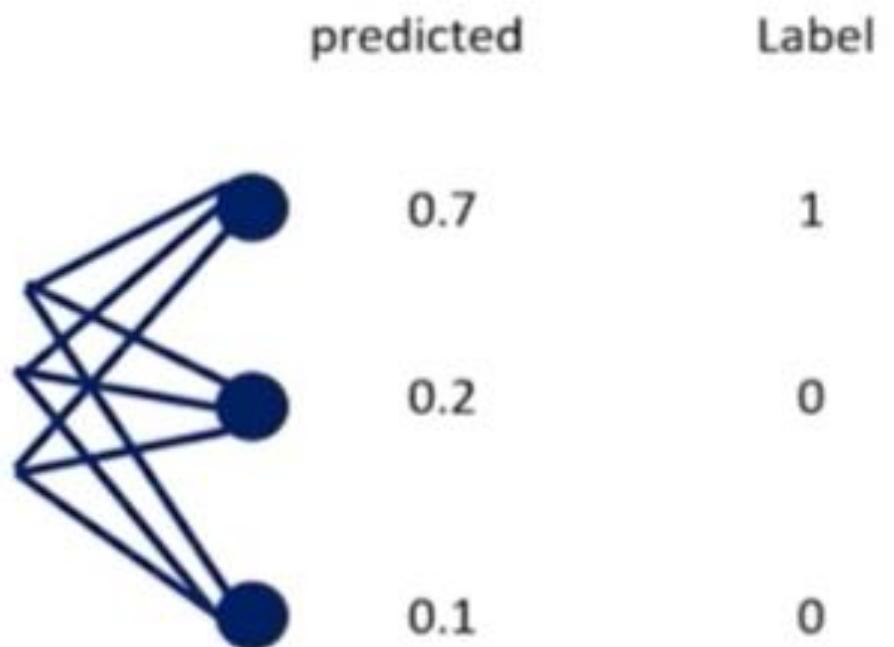


$$\frac{\partial C}{\partial W_1}$$

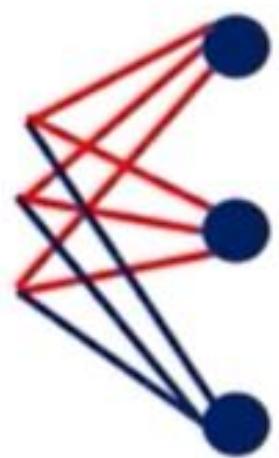
$$\frac{\partial C}{\partial W_2}$$

⋮

$$\frac{\partial C}{\partial W_3}$$



predicted	Label
0.7	1
0.2	0
0.1	0



predicted

Label

0.7

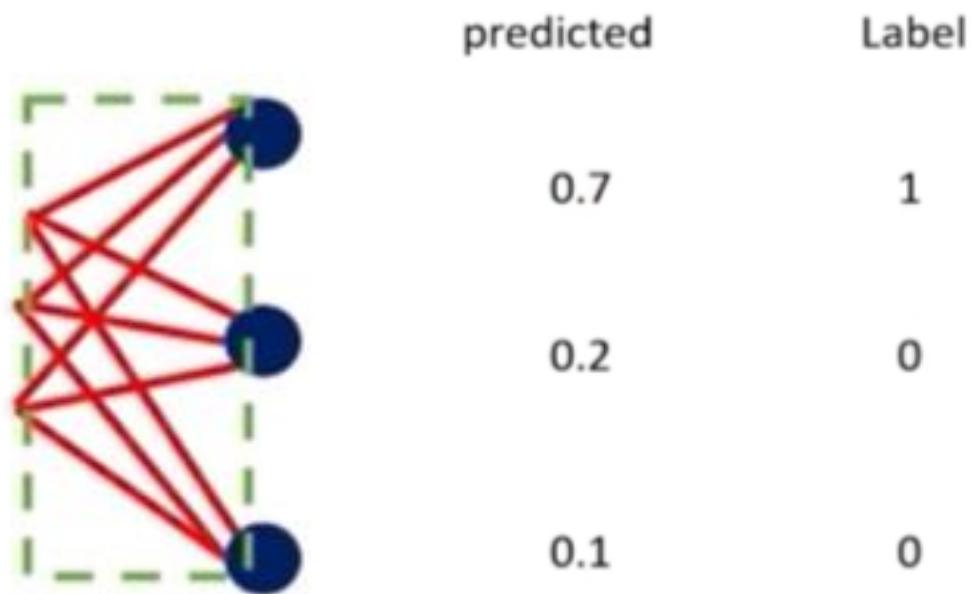
1

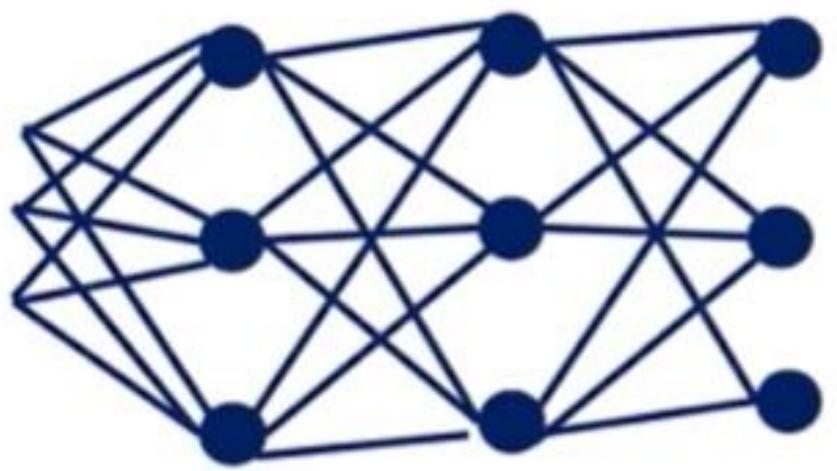
0.2

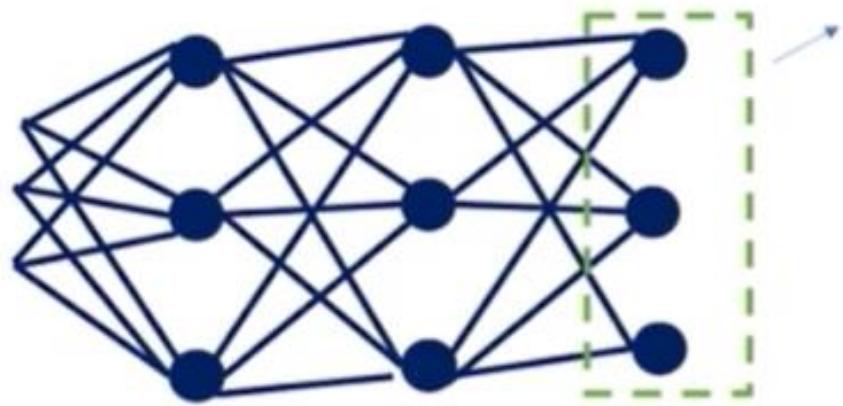
0

0.1

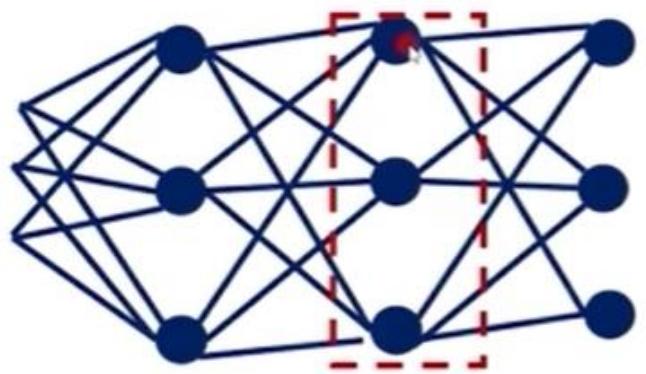
0



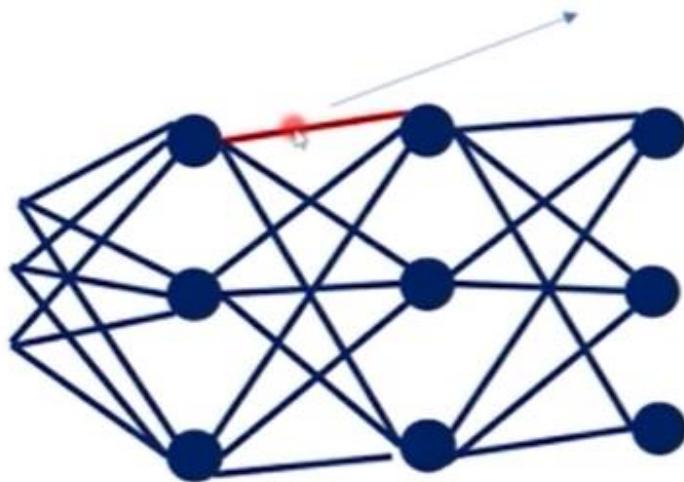


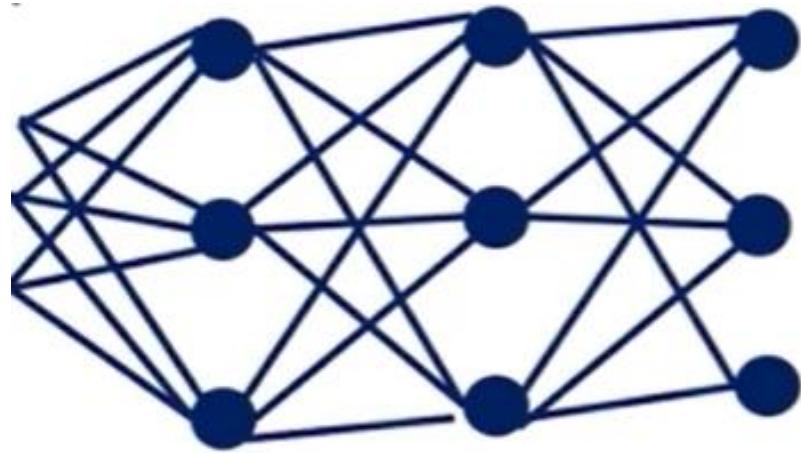


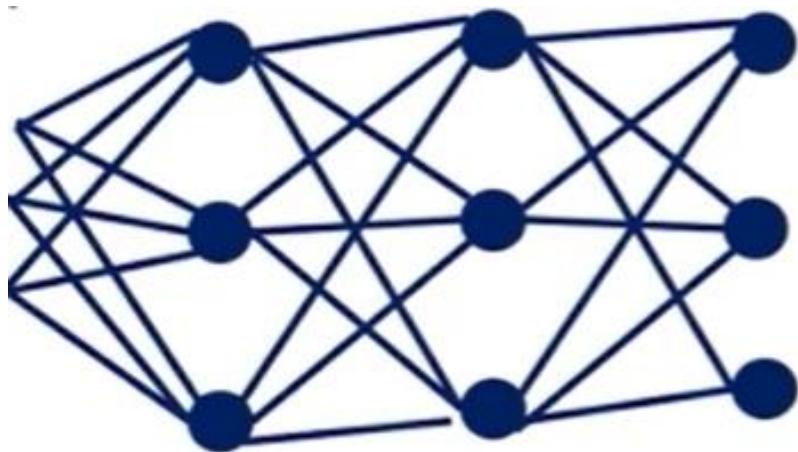
فقط برای لایه آخر علاوه بر خروجی خود نورون ها خروجی
مورد انتظار (لیبل) را نیز داریم



به دنبال روشی هستیم که گرادیان این وزن‌ها را نسبت به
تابع خطایی نهایی حساب کنیم.

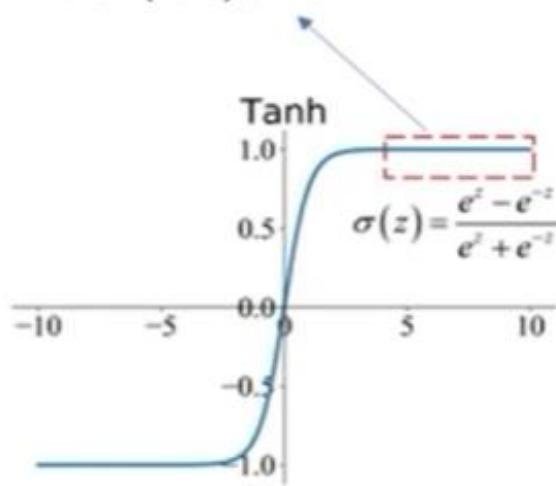
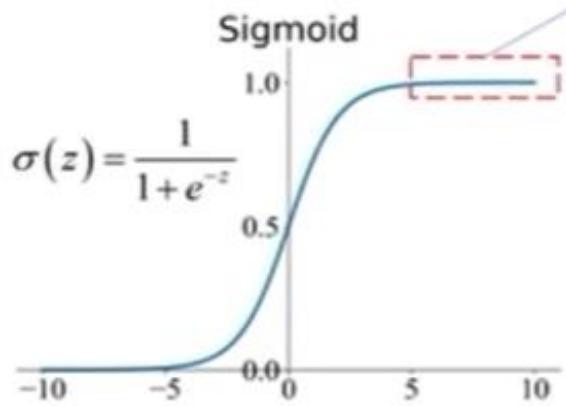


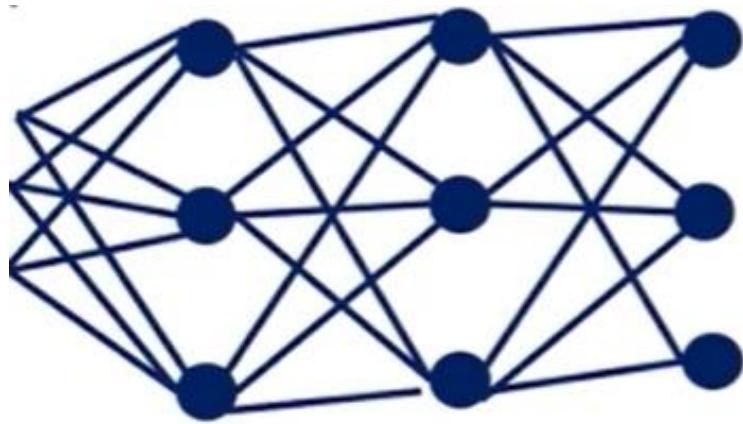

$$\square \times \square \times \square \times \dots$$



$\square \times \square \times \square \times \dots$

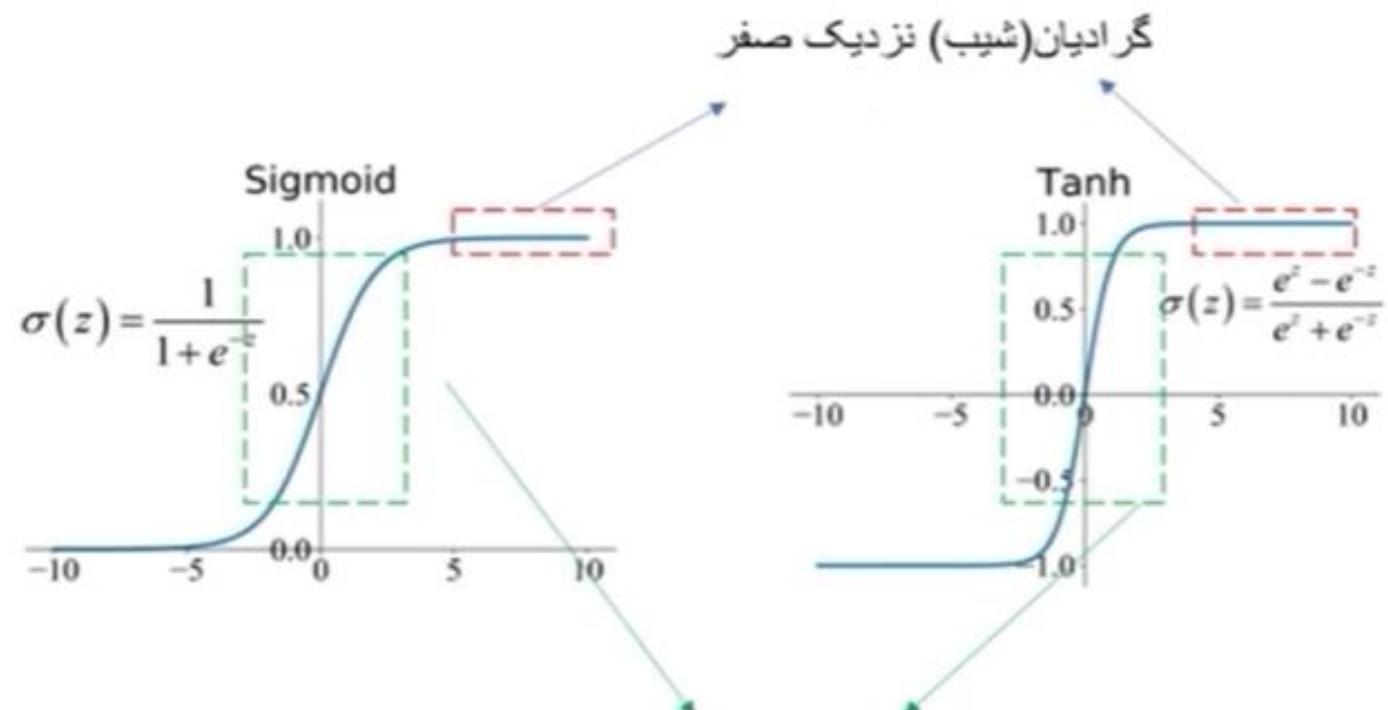
گرادیان(شیب) نزدیک صفر





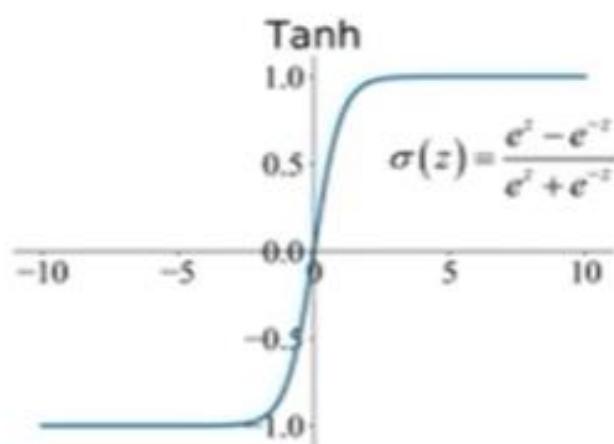
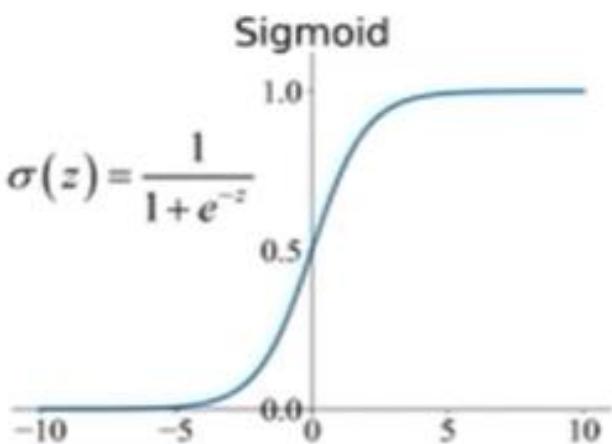
$$W = W - 0.01 \times 0.005 \times 0.01 \times \dots \times 0.1$$

حاصل ضرب تعداد زیادی عدد کوچک نزدیک صفر است



گرادیان مناسب و غیر صفر فقط در برخی از نقاط تابع

گرادیان ۱ و غیر نزدیک صفر در قسمت مثبت



$$\text{ReLU}(z) = \begin{cases} z, & z > 0 \\ 0, & \text{otherwise} \end{cases}$$

