Neural Networks

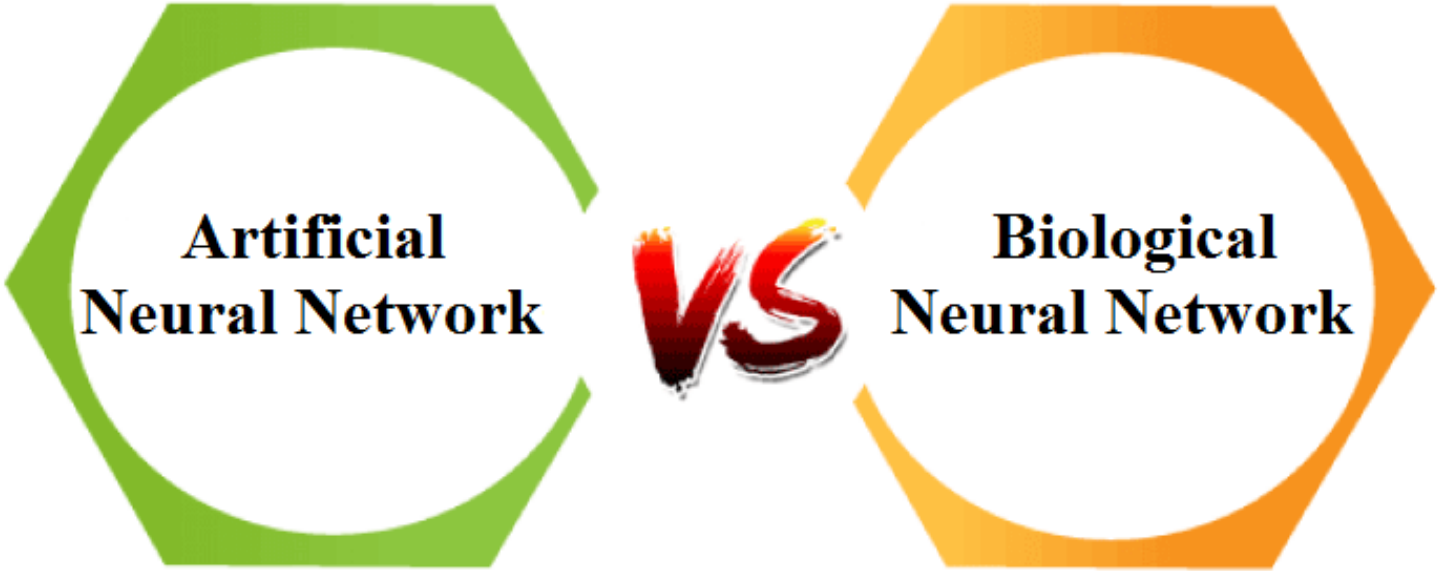
شما در حال حاضر،در حال استفاده از یک سیستم عصبی بیولوژیکی پیچیده هستید، جهت فهم مطالب هستید.

این سیستم دو درصد از وزن بدن شما را تشکیل می دهد

و بیش از ۲۰ درصد کل اکسیژن بدن را مصرف می کند.

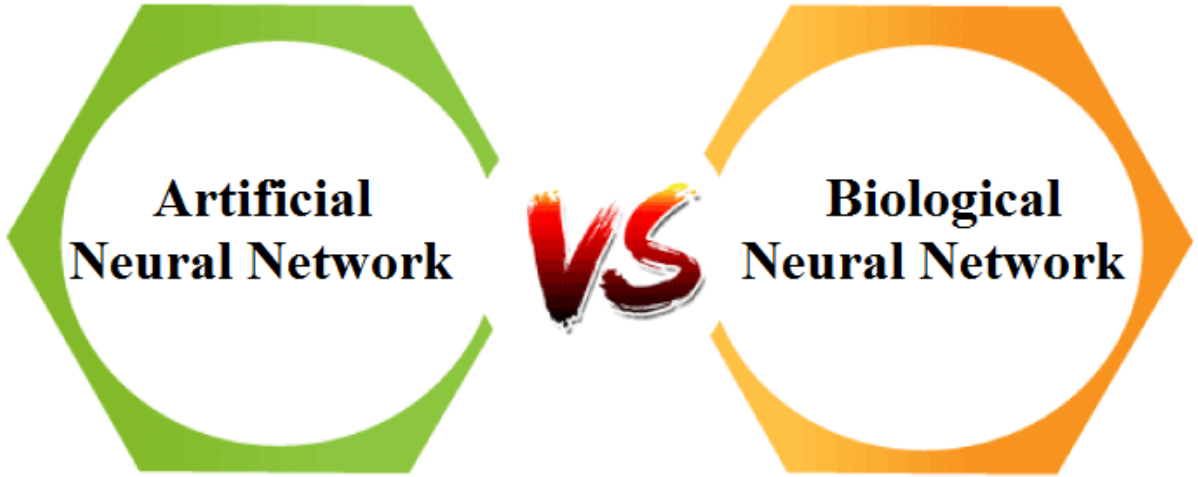
- سیستم عصبی شما یکی از سریع ترین و دقیق ترین سیستم های محاسباتی دنیاست.

حجم عظیمی از اطلاعات و سیگنال ها جهت این کار و در طی زمانی کمتر از چند صدم ثانیه بایستی جمع آوری و محاسبه شود.



پیاده سازی ویژگی های شگفت انگیز مغز در یک سیستم عصبی مصنوعی همیشه وسوسه انگیز و مطلوب بوده است.

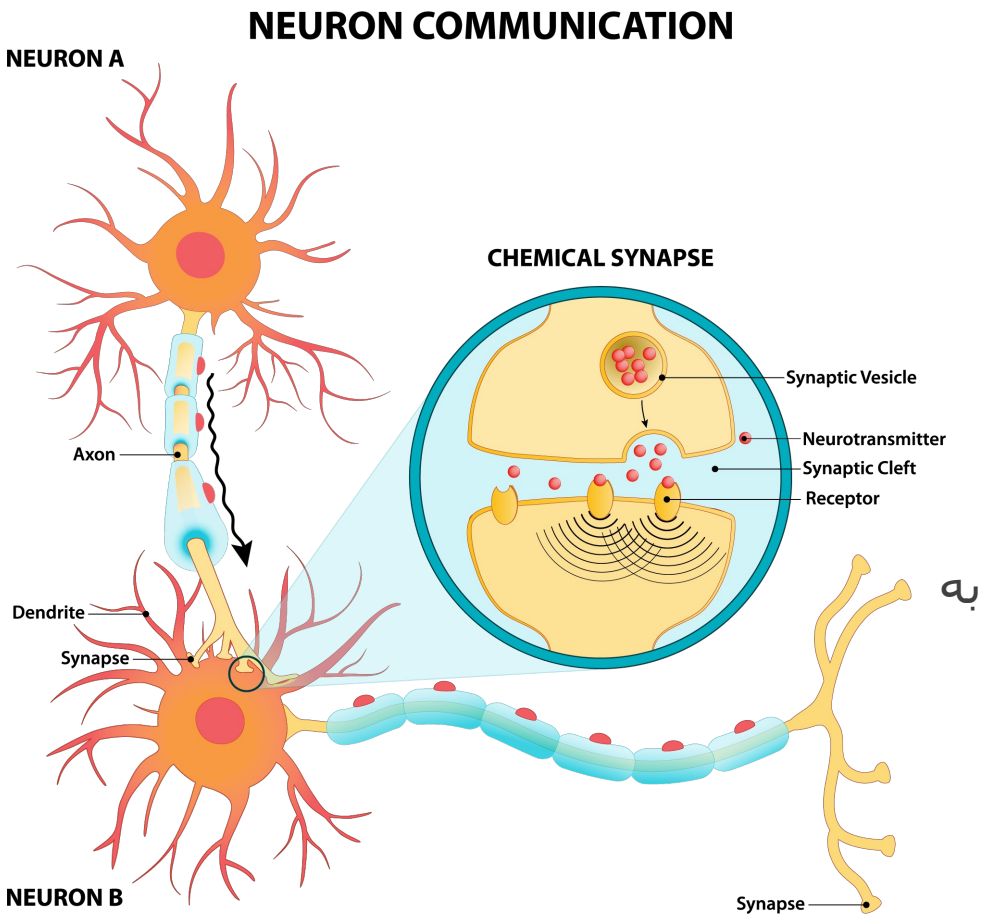
اما سالها تلاش محققین از سال ۱۹۱۱ثابت می کند که مغز بشر دست نیافتنی است.



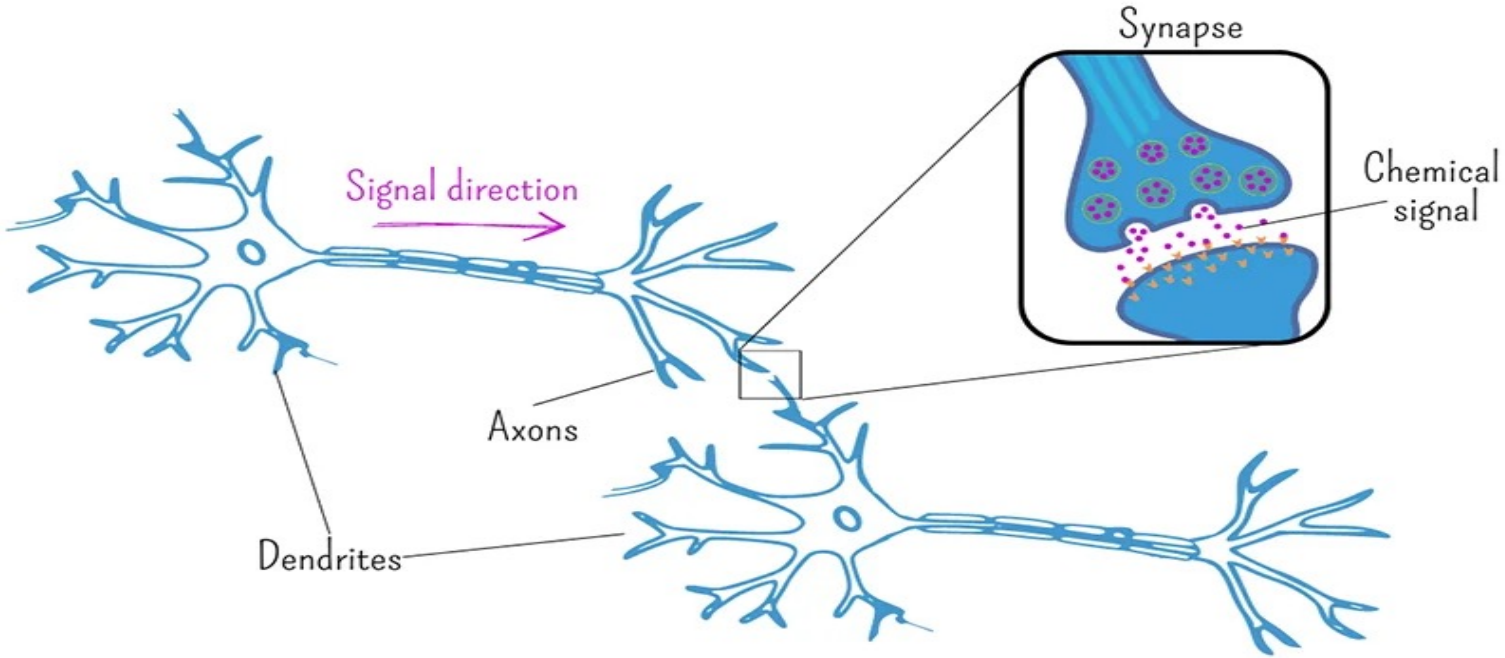
اگر چه نرون های بیولوژیکی از نرون های مصنوعی که توسط های مدارهای الکتریکی ساخته می شوند، بسیار کندترهستند (یک میلیون) ،اما عملکرد مغز خیلی سریعتر از یک کامپیوتر معمولی است.

علت این پدیده بیشتر به دلیل ساختار کاملا موازی نرون های بیولوژیکی است. در سیستم های بیولوژیکی همه نرون های بطور همزمان کار می کنند و پاسخ می دهند.

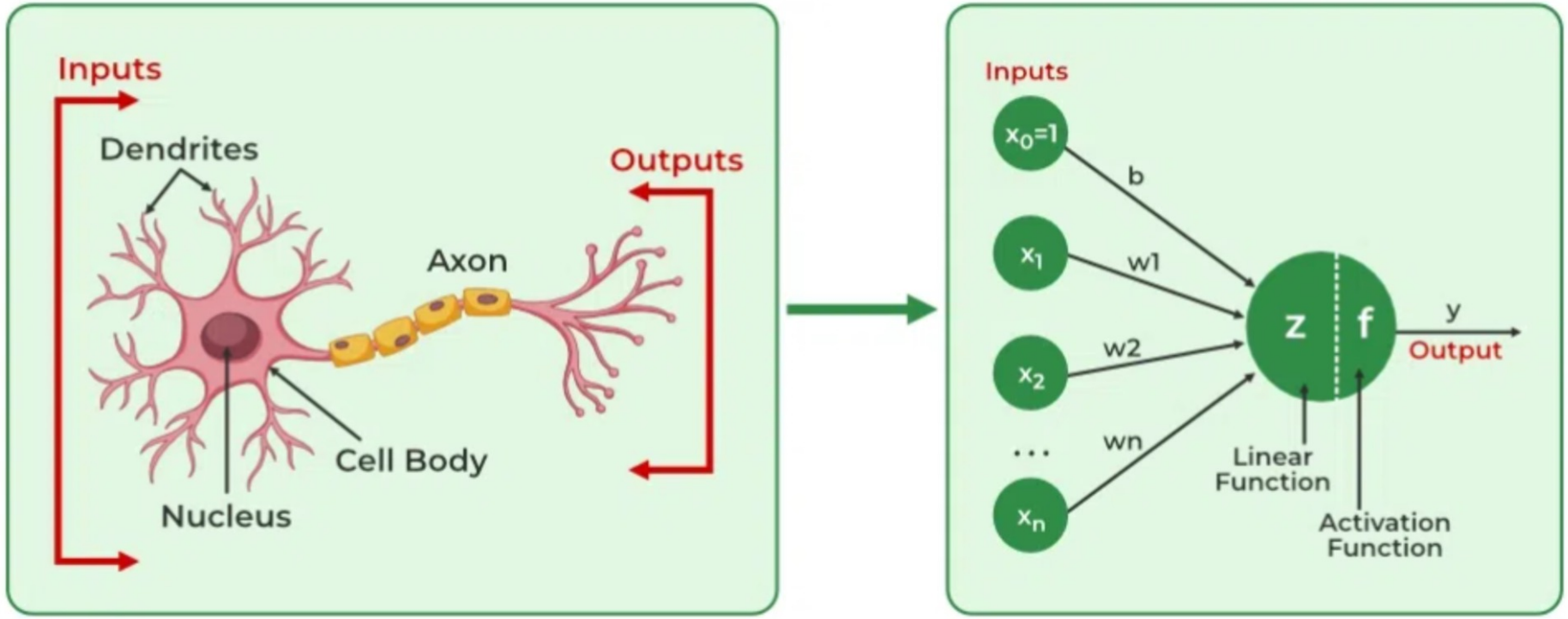
ساختار نرون



- هسته : انرژی لازم برای فعالیت های نرون را فراهم می کند.
- دندریت : دریافت کننده سیگنال های الکتروشیمیایی
- اکسون : سیگنال های الکتروشیمیایی نرون را به سایر نرون ها منتقل می کند.
- سیناپس : محل تلاقی اکسون یک نرون به دندریت نرون دیگر



مغز شما یک سیستم پردازش اطلاعات با ساختار موازی است که از۱۰۰ تریلیون ($۱۰^{۱۱}$) نرون به هم متصل با $۱۰^{۱۶}$ ارتباط تشکیل شده است.



شبکه های عصبی مصنوعی الگوریتم هایی هستند که سعی می کنند از ساختار شبکه ای ،

نحوه پردازش موازی و شیوه یادگیری مغز الگو بگیرند.

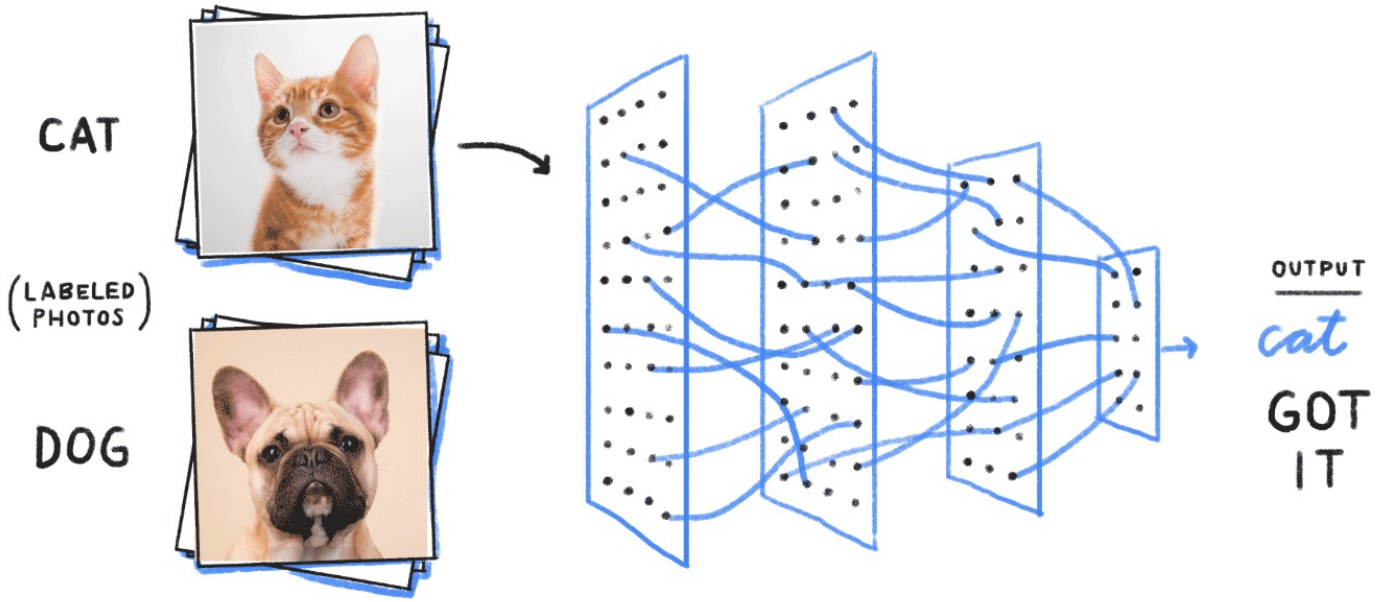
یادگیری به معنای ایجاد ارتباطات جدید بین نرون ها یا تنظیم مجدد ارتباطات موجود بین آنهاست.

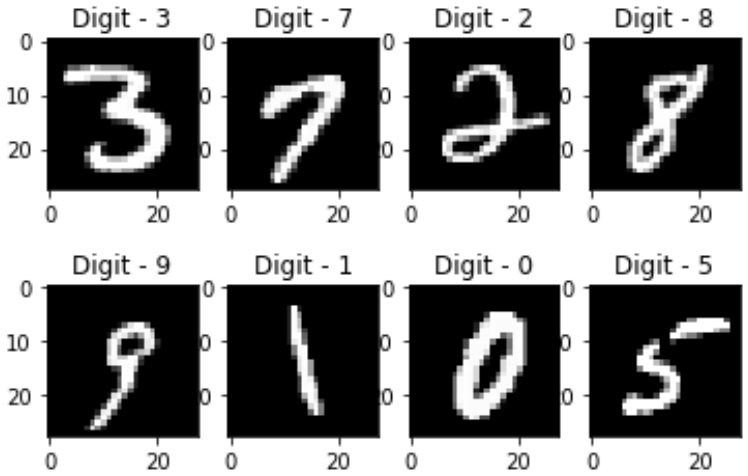
سوال:

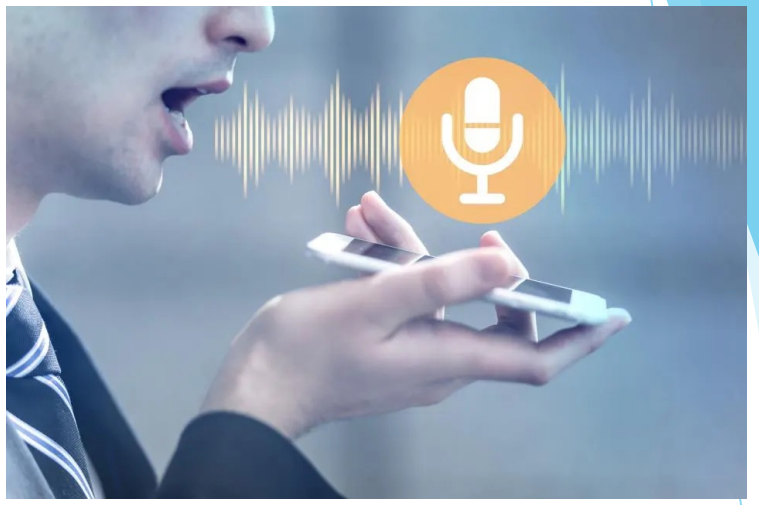
- آیا می توان یک شبکه کوچک از نرون های مصنوعی ساخت، به طوری که جهت حل مسایل پیچیده (که همان یادگیری نگاشت هاست) آموزش پذیر باشد؟

بهترین کاربرد شبکه های عصبی مصنوعی

زمانی است که تعداد feature های شما زیاد باشد و به همین دلیل سایر مدل های یادگیری ماشین روی داده شما عملکرد خوبی نداشته باشند.

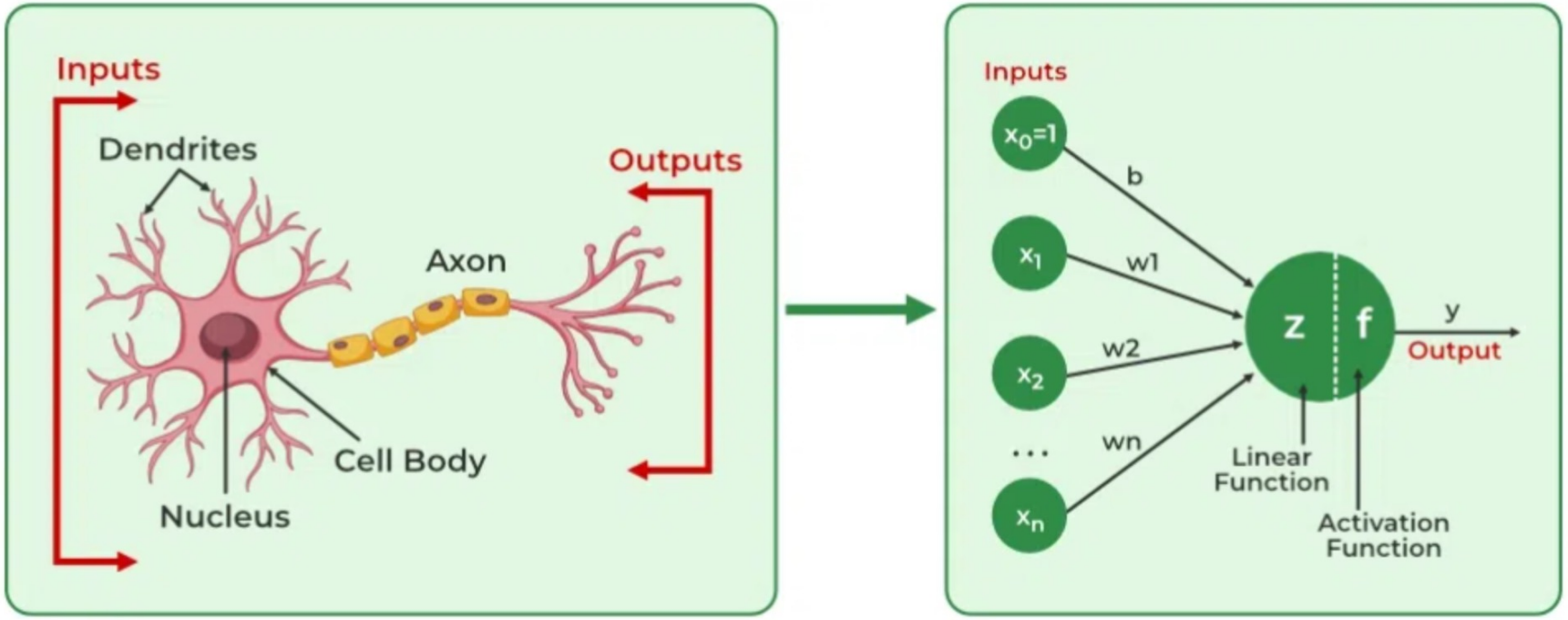



Digit - 3
Digit - 7
Digit - 2
Digit - 8
Digit - 9
Digit - 1
Digit - 0
Digit - 5

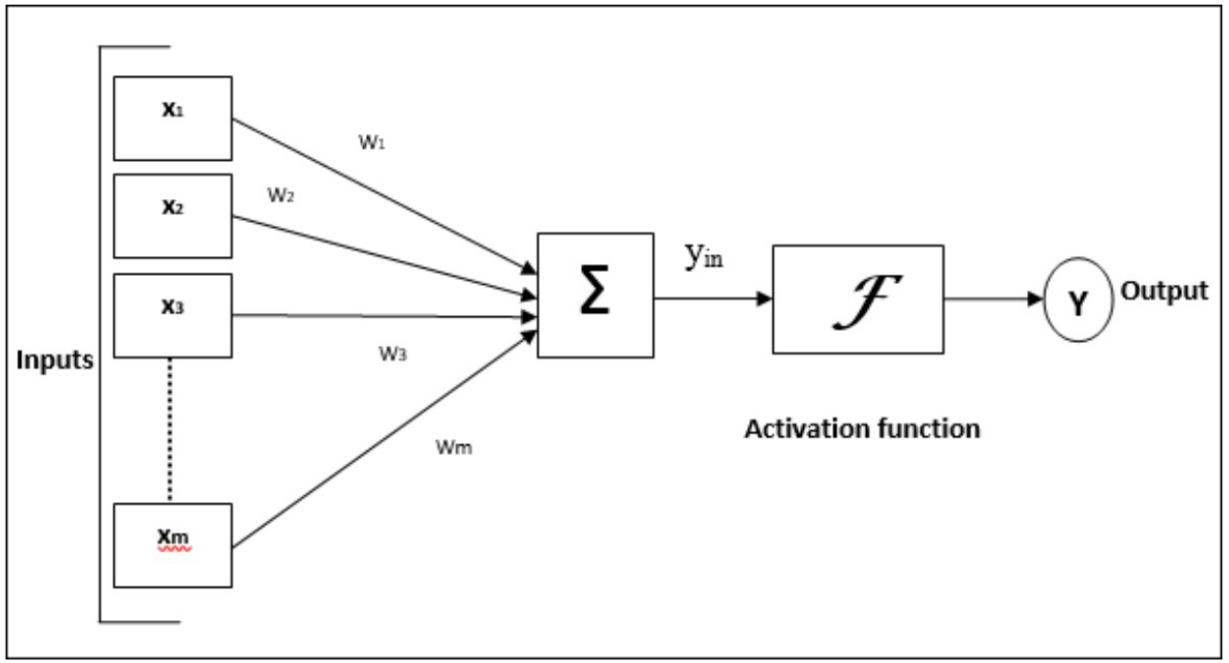



Happy
Surprise
Angry
Sad

Neuron model



Neuron model



$$y = f(\sum_{i=1}^{n} w_i x_i + b)$$

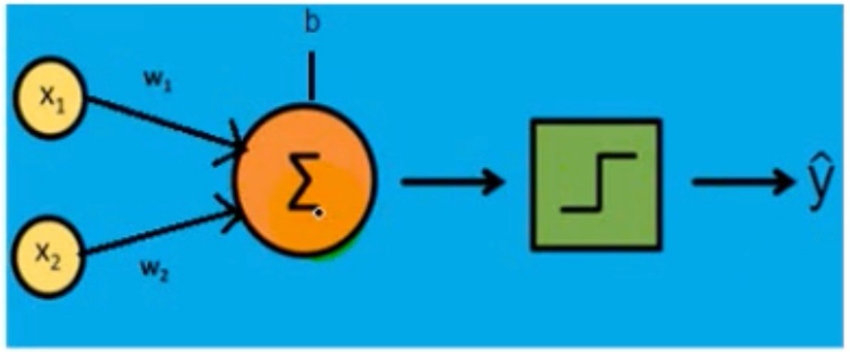
Here if is a sign function

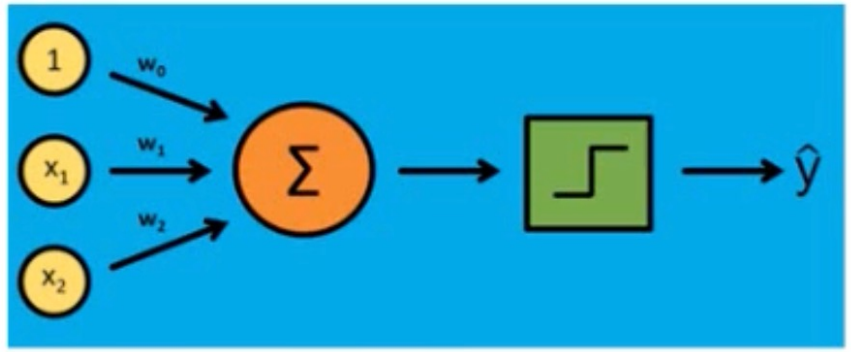
As we see single neuron is a linear classifier

Perceptron

شبکه های پرسپترون در سال ۱۹۵۸ توسط روزنبلات معرفی شد.

پرسپترون یک نرون ساده است که داده ها را به صورت خطی به دو کلاس تعریف می کند.

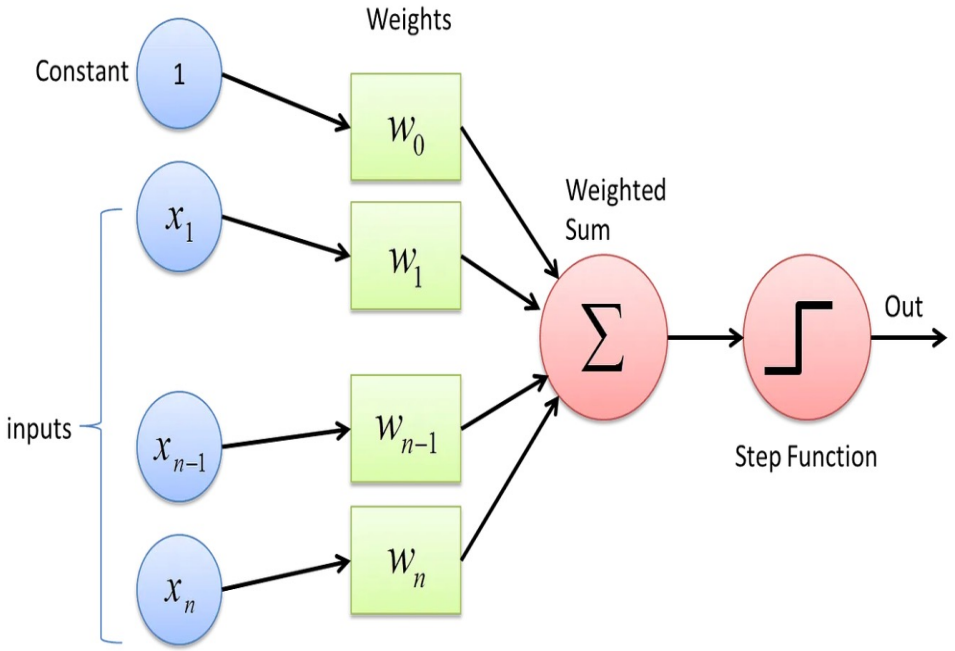




$$y = f(\sum_{i=1}^{n} w_i x_i + b)$$

$$y = f(\sum_{i=0}^{n} w_i x_i)$$

Single layer perceptron



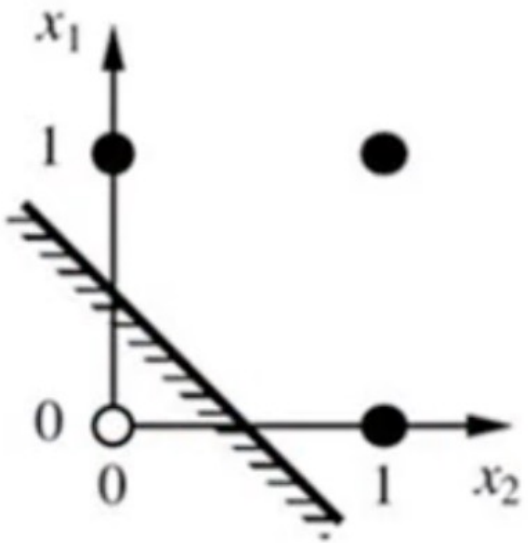
The perceptron consists of 4 parts.

1. Input values or One input layer
2. Weights and Bias
3. Net sum
4. Activation Function

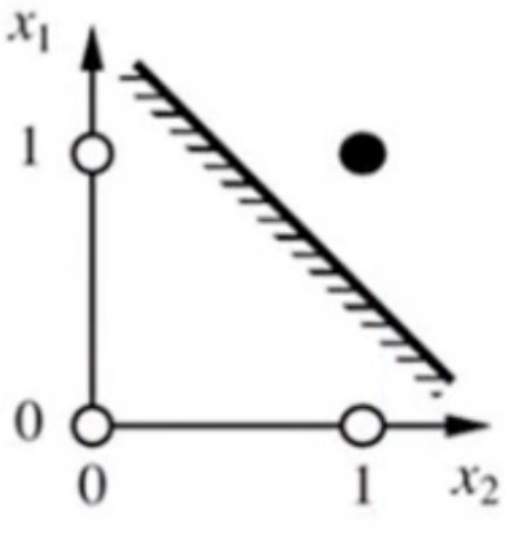
$$\begin{cases} a = 0 & n < 0 \\ a = 1 & n > 0 \end{cases}$$

تابع محرک در پرسپترون تابع پله ای یا sign است

پیاده سازی گیت های منطقی با پرسپترون

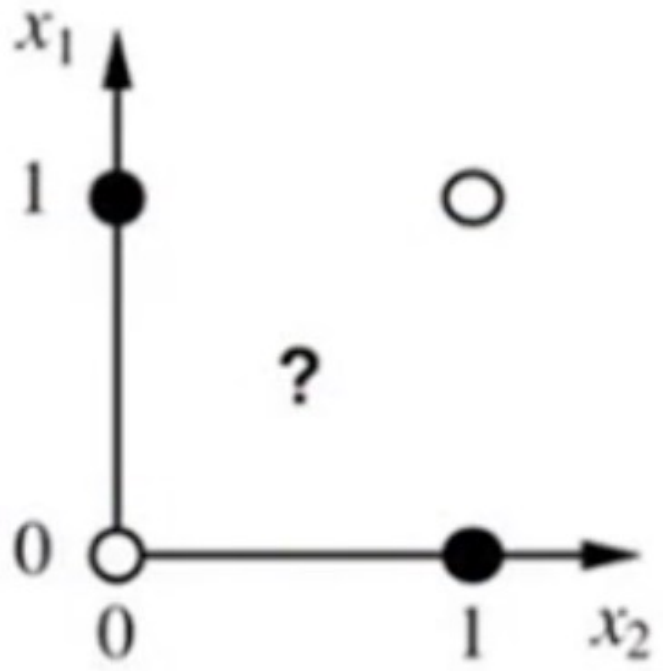


x_1 or x_2



x_1 and x_2

پیاده سازی گیت های منطقی با پرسپترون



x_1 xor x_2

قانون آموزش پرسپترون (قانون دلتا)

برای پیدا کردن وزن های پرسپترون ، از یک مقدار تصادفی اولیه شروع می کنیم و بعد با استفاده از قانون دلتا وزن ها را بروزرسانی می کنیم.

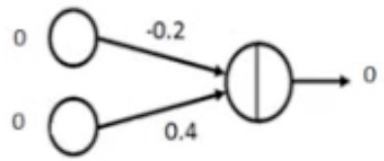
$$w_{i_{\text{جدید}}} = w_{i_{\text{قدیم}}} + \alpha\, e\, x_i$$

e = t−y

t : مقدار واقعی

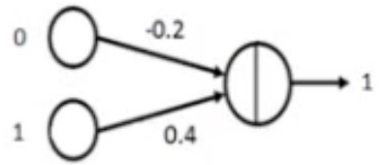
y : مقدار پیش بینی شده

پیاده سازی OR با پرسپترون تک لایه



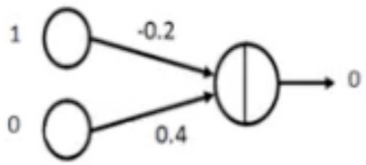
$$y = f(0\times -0.2 + 0\times 0.4) = f(0) = 0$$

$$e = 0 - 0 = 0$$



$$y = f(0\times -0.2 + 1\times 0.4) = f(0.4) = 1$$

$$e = 0 - 0 = 0$$

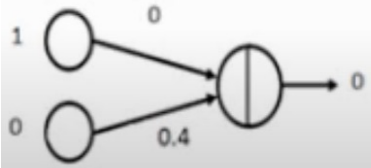


$$y = f(1\times -0.2 + 0\times 0.4) = f(-0.2) = 0$$

$$e = 1 - 0 = 1$$

$$w_1 = w_1 + 1\times 0.2\times x_1 = -0.2 + 0.2 = 0$$

$$w_2 = w_2 + 1\times 0.2\times x_2 = 0.4 + 0 = 0.4$$



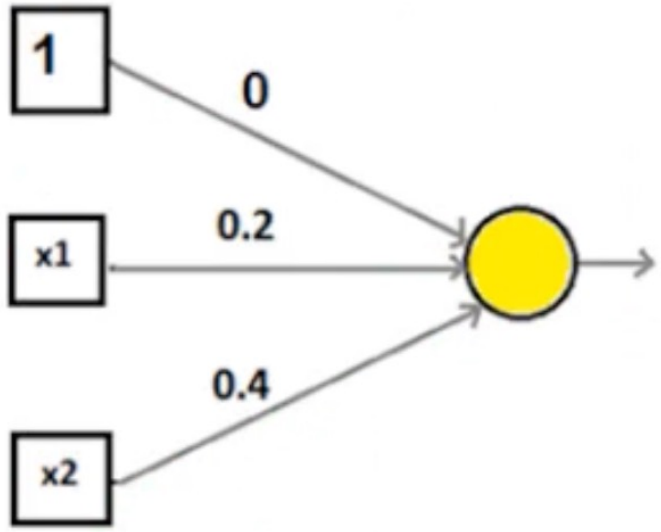
$$y = f(1\times 0 + 1\times 0.4) = f(0.4) = 1$$

$$e = 1 - 1 = 0$$

پیاده سازی OR با پرسپترون تک لایه

epoch	x1	x2	t	y	e	w1	w2
1	0	0	0	0	0	-0.2	0.4
	0	1	1	1	0	-0.2	0.4
	1	0	1	0	1	0	0.4
	1	1	1	1	0	0	0.4
2	0	0	0	0	0	0	0.4
	0	1	1	1	0	0	0.4
	1	0	1	0	1	0.2	0.4
	1	1	1	1	0	0.2	0.4
3	0	0	0	0	0	0.2	0.4
	0	1	1	1	0	0.2	0.4
	1	0	1	1	0	0.2	0.4
	1	1	1	1	0	0.2	0.4

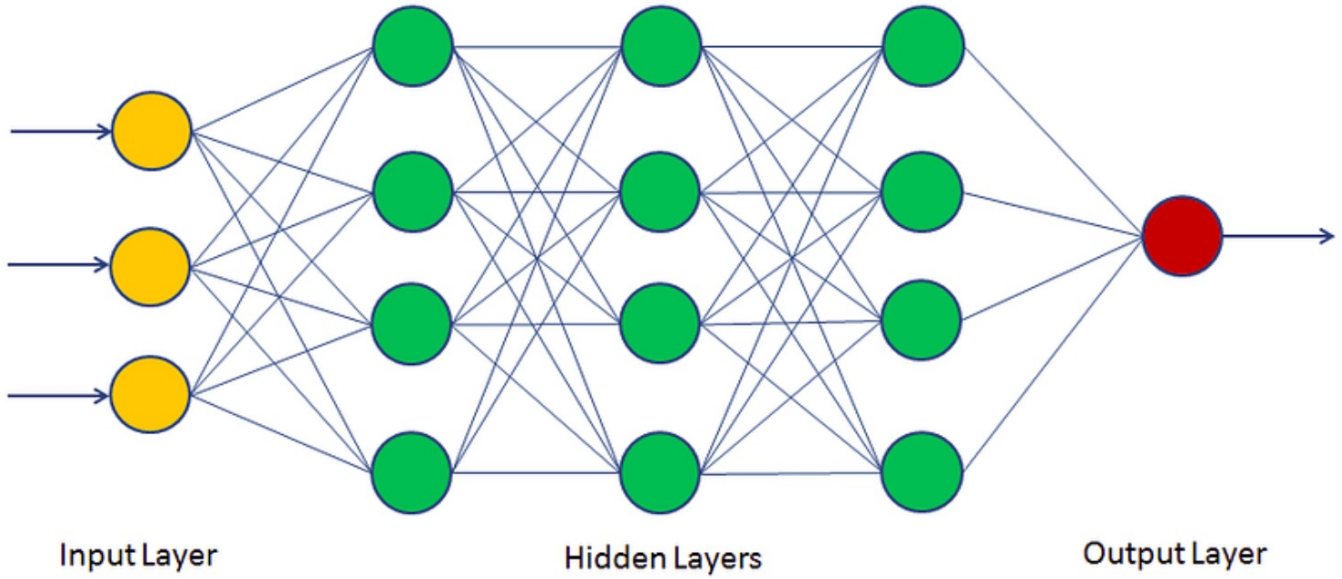
پیاده سازی OR با پرسپترون تک لایه



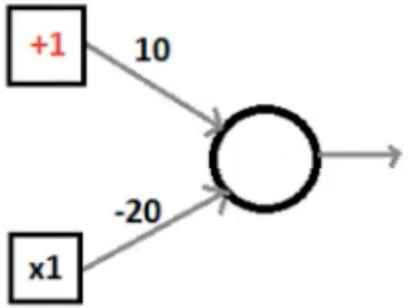
$$h_\theta(x) = f(0 + 0.2x_1 + 0.4x_2)$$

Multi layer perceptron (MLP)

- ✓ همان طور که دیدیم شبکه های عصبی پرسپترون تک لایه فقط قادر به تفکیک کلاس هایی هستند که به صورت خطی جدایی پذیر هستند.
- ✓ اما می دانیم که بیشتر مسایل دنیای واقعی به صورت خطی جدایی پذیر نیستند به همین دلیل از الگوریتم پرسپترون چند لایه برای انها استفاده می کنیم.

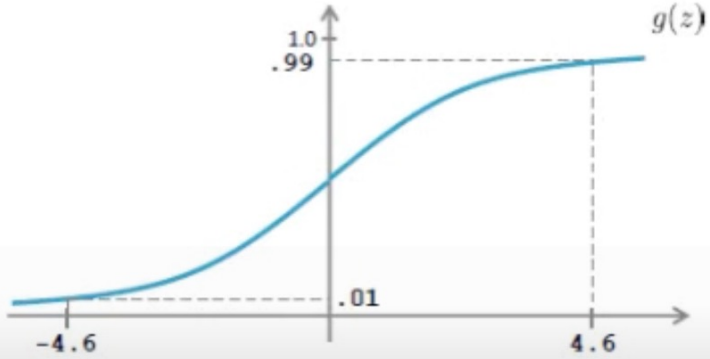


پیاده سازی گیت NOT

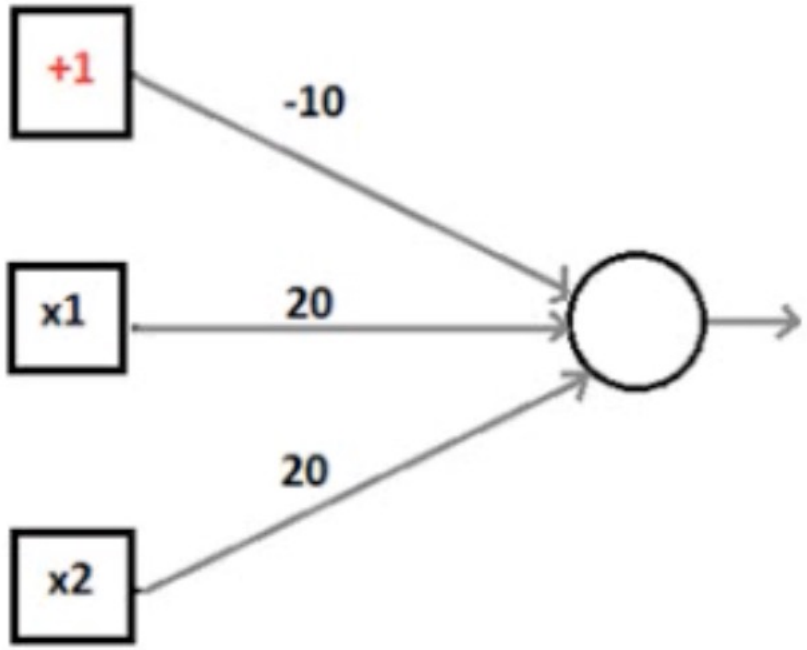


$$h_\theta(x) = g(10 - 20x_1)$$

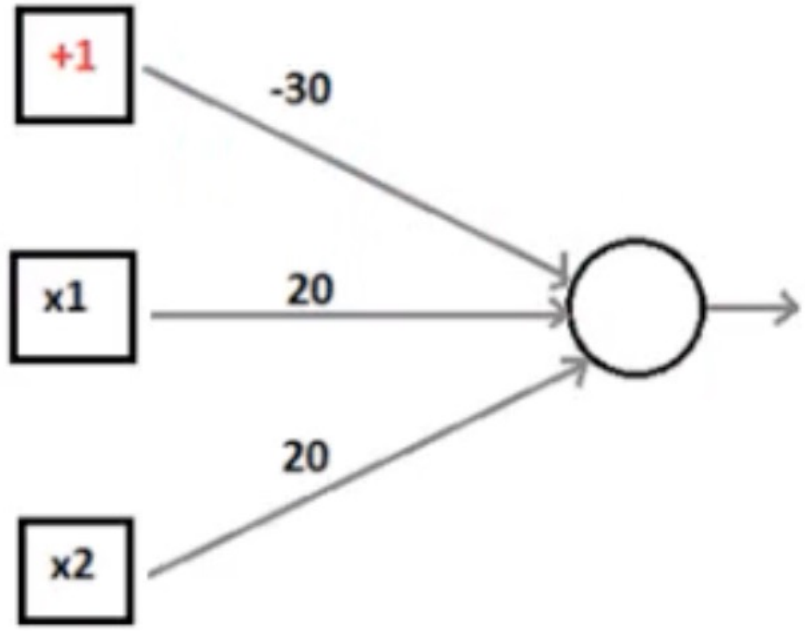
x1	h(x)
0	g(10)=1
1	g(-10)=0



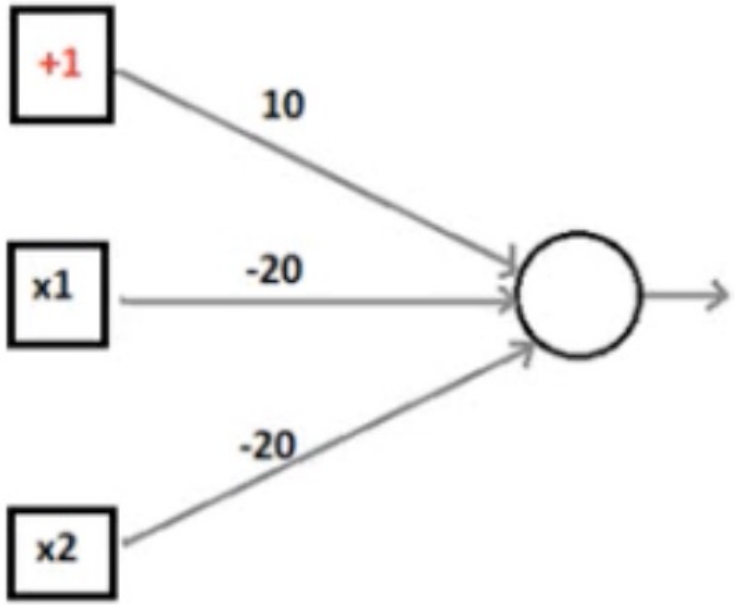
پیاده سازی گیت OR



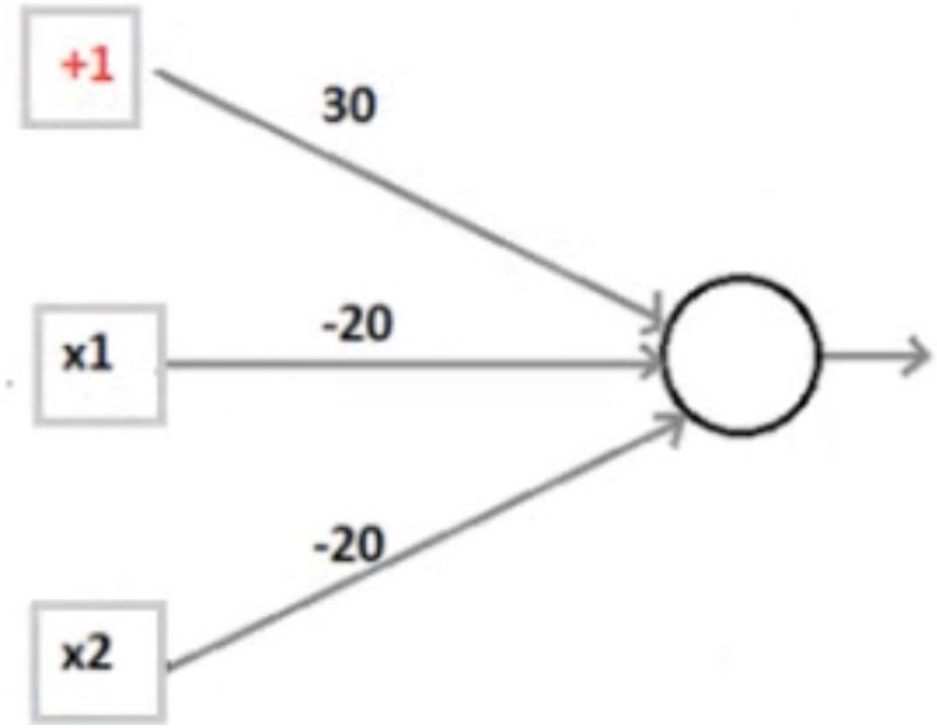
پیاده سازی گیت AND



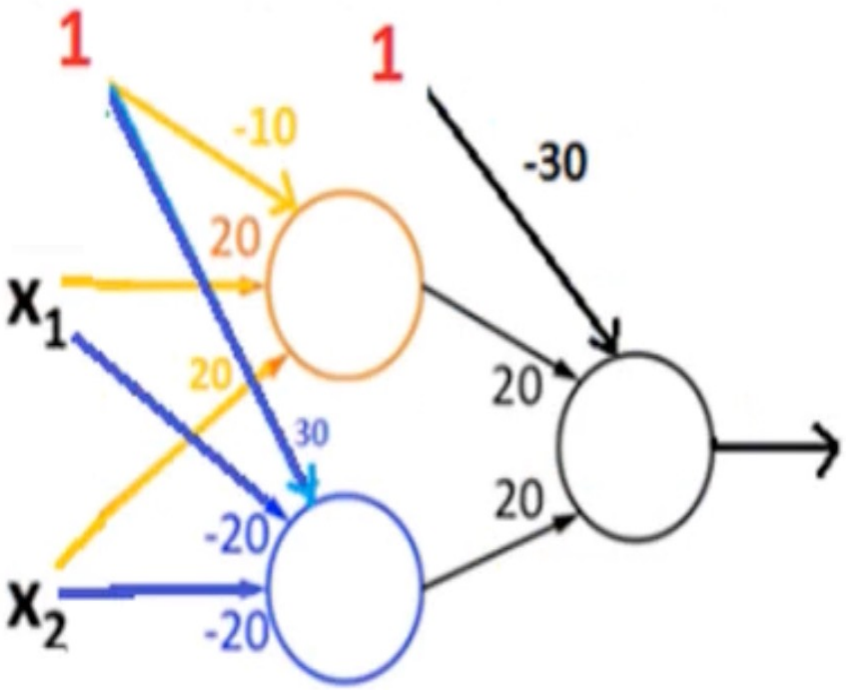
$$(not\ \ x_1)\ \ AND\ \ (not\ \ x_2)$$

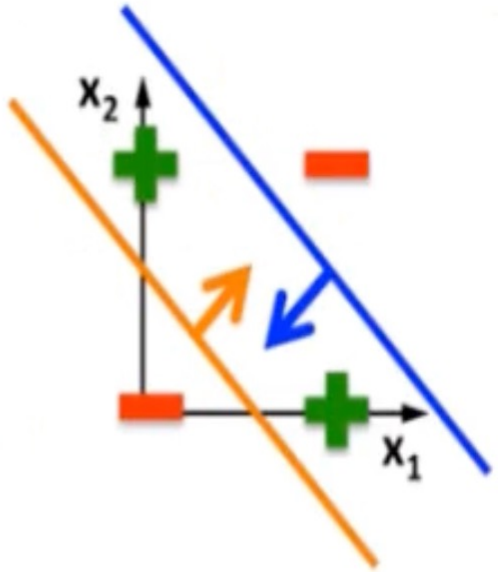


(not x_1) AND (not x_2)



پیاده سازی گیت XOR





پیاده سازی گیت XNOR