

# دوره‌های AI

## جلسه اول

محمد رضا بر جیان



# هوش مصنوعی

تعریفات مختلفی برای هوش مصنوعی گفته شده است که چند تا شو میخوینیم.

01

مانند انسان فکر کردن  
ماشین، مانند انسان رفتار  
کردن ماشین، منطقی فکر  
کردن، منطقی رفتار کردن

02

هنر ساختن ماشین هایی که  
کارهایی انجام می دهند که  
آن کارها توسط انسان با فکر  
کردن انجام می شوند

03

مطالعه برای ساختن  
کامپیوترها برای انجام  
کارهایی که فعل انسان  
آن ها را بهتر انجام می  
دهد

04

مطالعه برای ساختن  
کامپیوترها برای انجام  
کارهایی که فعل انسان  
آن ها را بهتر انجام می  
دهد



# Turing Test

پردازش زبان طبیعی 01

نمایش دانش 02

استدلال خودکار 03

یادگیری ماشین 04



# روش‌های اصلی AI

Symbolic AI

01

Machine Learning(ML)

02



## Symbolic AI

در این روش یکسری قوانین ایجاد میشه و سیستم  
براساس اونها به ورودی‌ها پاسخ میده

### این روش برای کجا مناسب نیست؟

این روش برای مواردی که قوانین مشخص و کم  
هستند مناسب است ولی این روش برای مواردی  
که سیستم پیچیده است مناسب نیست.

مانند  
ترجمه زبان، ماشین خودران، تشخیص تصویر،  
تشخیص گفتار  
گاهی قوانین ناشناخته هستند

# Machine Learning(ML)

- improve their performance P
- at some task T
- with experience E

## Example

پیش بینی قیمت سهام  
(task)

با قیمت های گذشته  
(Experience)

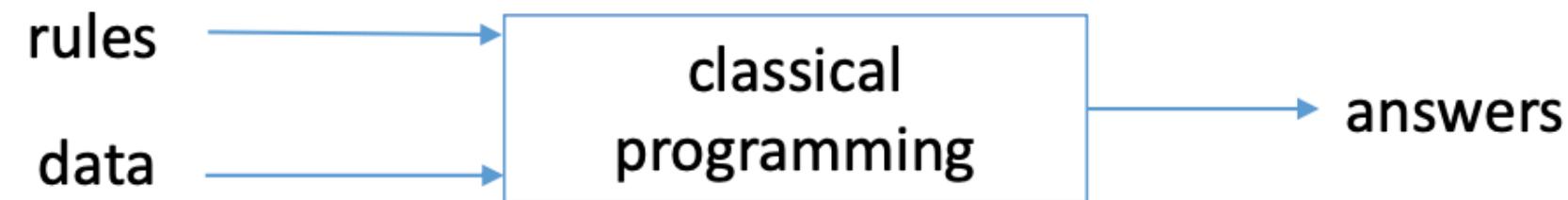
با دقت بالا  
(performance)

تشخیص هرزنامه بودن ایمیل  
(task)

با بررسی ایمیل هرزنامه های گذشته  
(experience)

با دقت بالا  
(performance)

# Machine Learning(ML)



- قواعد و وضعیت بیمار به سیستم داده می شود و نوع بیماری تعیین شود.
- قواعد بازی به سیستم داده می شود و سیستم شطرنج بازی می کند.



- یادگیری ماشین از حوزه های پیشرو در چند سال اخیر است زیرا:
- الگوریتم ها پیشرفت کرده اند.
- یادگیری ماشین به سخت افزار قوی و داده زیاد نیاز دارد که این هردو در سال های اخیر فراهم شده است.

# انواع یادگیری

- ▶ یادگیری تحت نظارت (Supervised Learning)
- ▶ یادگیری بدون نظارت (Unsupervised Learning)
- ▶ یادگیری نیمه نظارت (Semi-supervised Learning)
- ▶ یادگیری تقویتی (Reinforcement Learning)



- تشخیص ایمیل‌های اسپم
- خوشبندی مشتریان
- طبقه‌بندی تصاویر با داده‌های کم برچسب‌دار
- آموزش یک ربات برای بازی شطرنج



# Classification

یکی از انواع مسائل در یادگیری ماشین است که هدف آن دسته‌بندی داده‌ها به دسته‌های از پیش تعریف شده است. به عبارت ساده‌تر، در مسائل کلاسیفیکیشن، مدل یاد می‌گیرد که ورودی‌ها را به یکی از چندین دسته ممکن اختصاص دهد.

درخت تصمیم (Decision Tree)

قضیه بیز (Bayes)

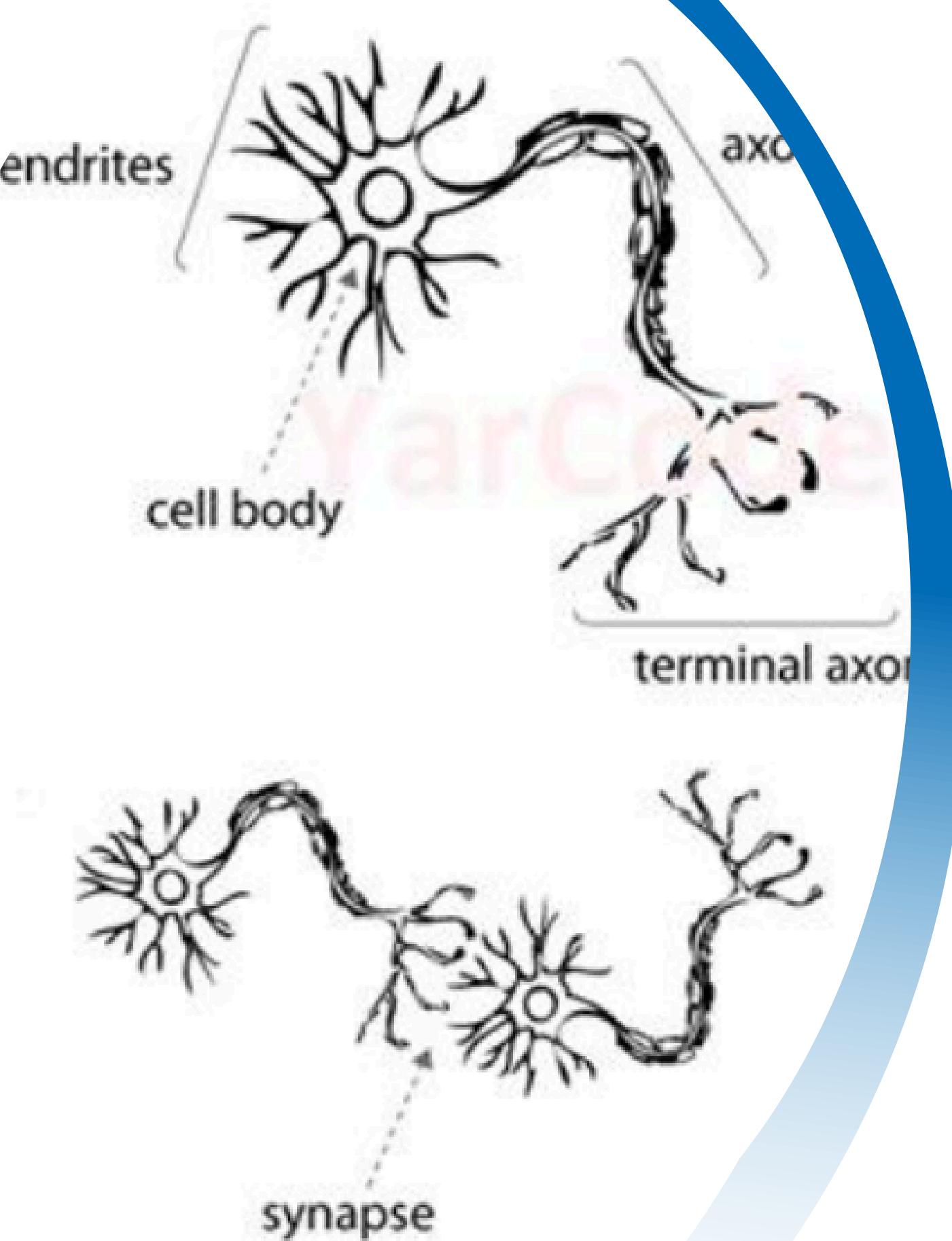
ماشین بردار پشتیبان (SVM)

k-نزدیک‌ترین همسایه (K-Nearest Neighbors)

شبکه‌های عصبی (Neural Networks)

جنگل تصادفی (Random Forest)

# شبکه عصبی مصنوعی



سلول های عصبی را یاخته عصبی یا نرون (neuron) گویند

مغز حدوداً 2 درصد وزن بدن است و حدود 20 درصد اکسیژن را مصرف می کند

مغز  $10^{11}$  نرون دارد

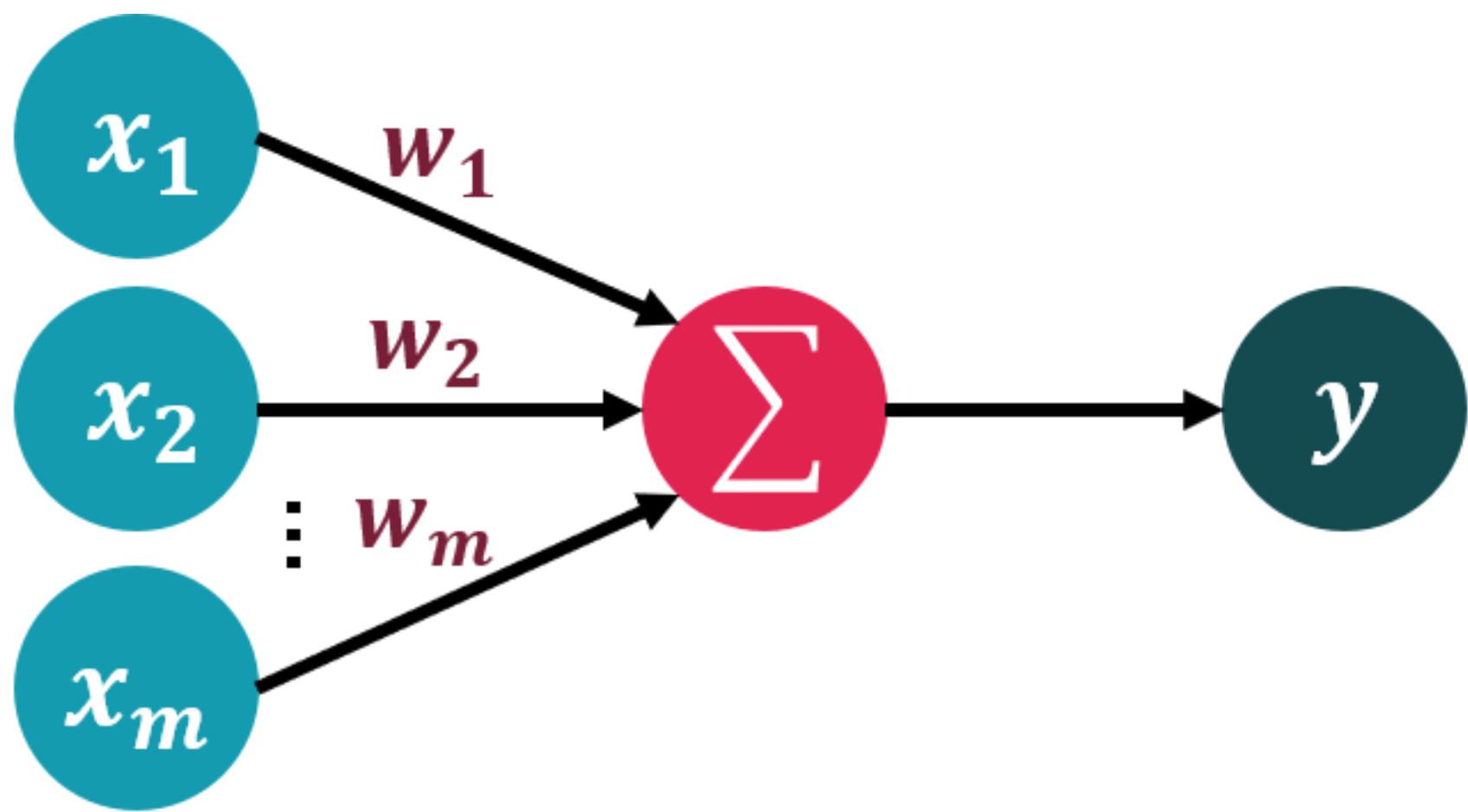
. با فرایند الکتروشیمیایی با یکدیگر ارتباط دارند

. نرون ها در اندازه و شکل های مختلف وجود دارند

هر نرون یا به نرون های دیگر متصل است یا به سلول های دیگر

هر نرون ممکن به 10000 نرون دیگر متصل باشد

ورودی های دندریت با ورودی های  $\times$  نشان داده می شوند.  
مواد در فاصله سیناپسی باعث تاثیر روی ورودی می شود. در شبکه عصبی مصنوعی با وزن های  $W$  نشان داده می شود.  
همه محرك ها از دندریت به سمت هسته رفته و هسته را تحت تاثیر قرار می دهند.  
در شبکه عصبی مصنوعی ورودی هادر نرون تجمیع می شوند.  
خروجی هسته از طریق اکسون منتقل می شود و اکسون روی این سیگنال تاثیر می گذارند و سپس به سمت نرون بعدی منتقل می کند.  
در شبکه عصبی مصنوعی نیز در نهایت نتیجه از سمت دیگر خارج می شود.



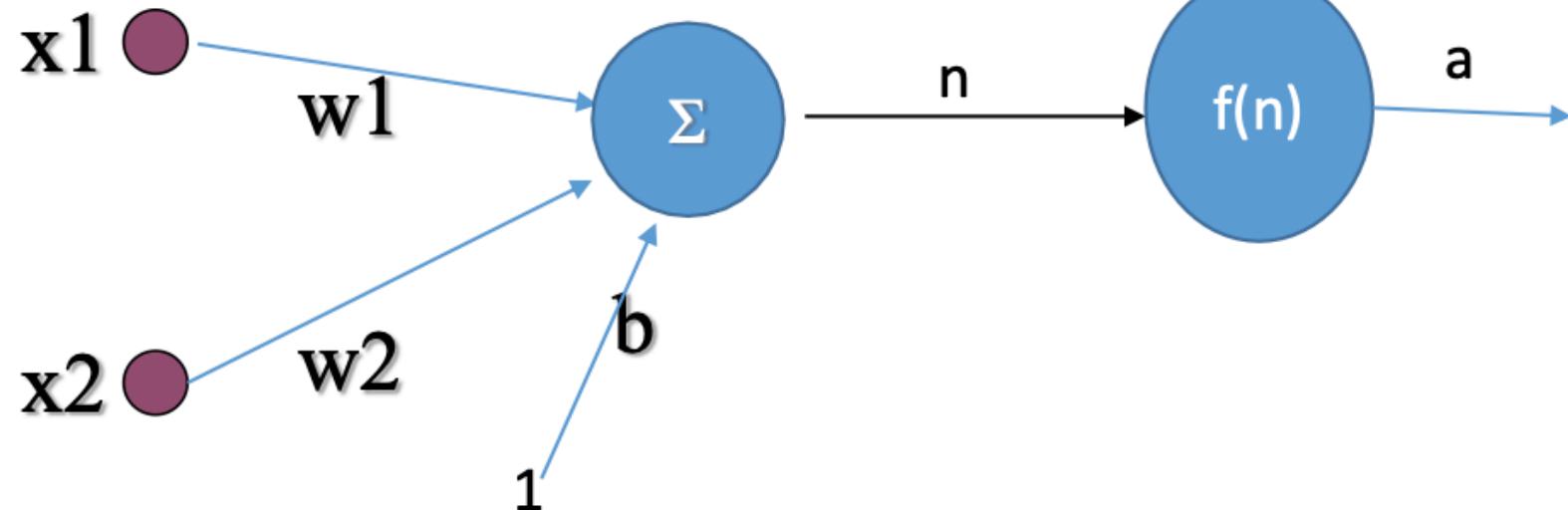
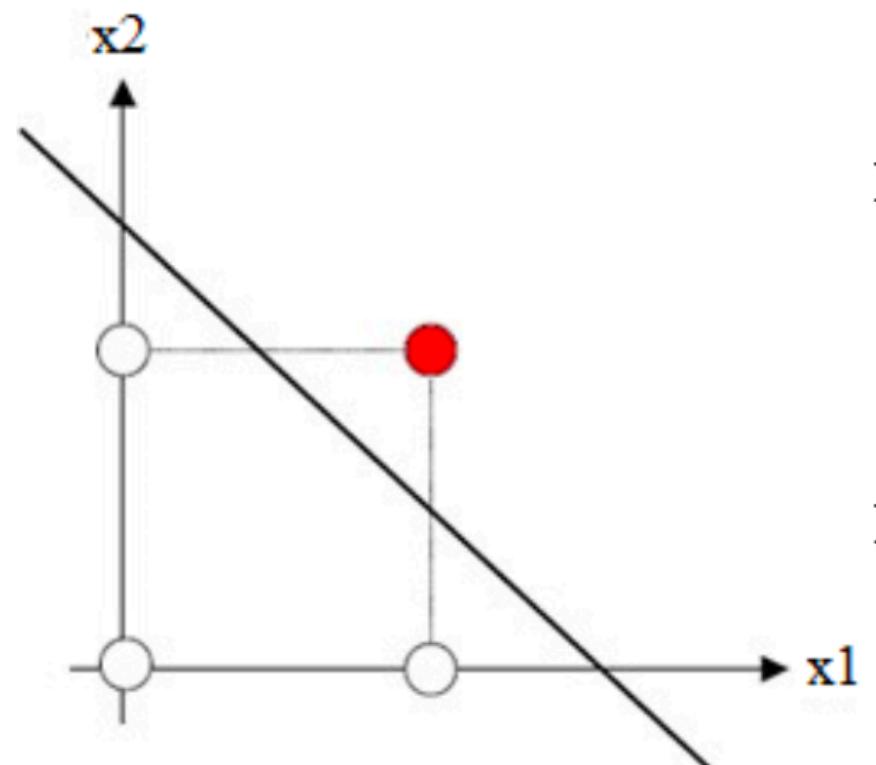
وزن ورودی

جمع

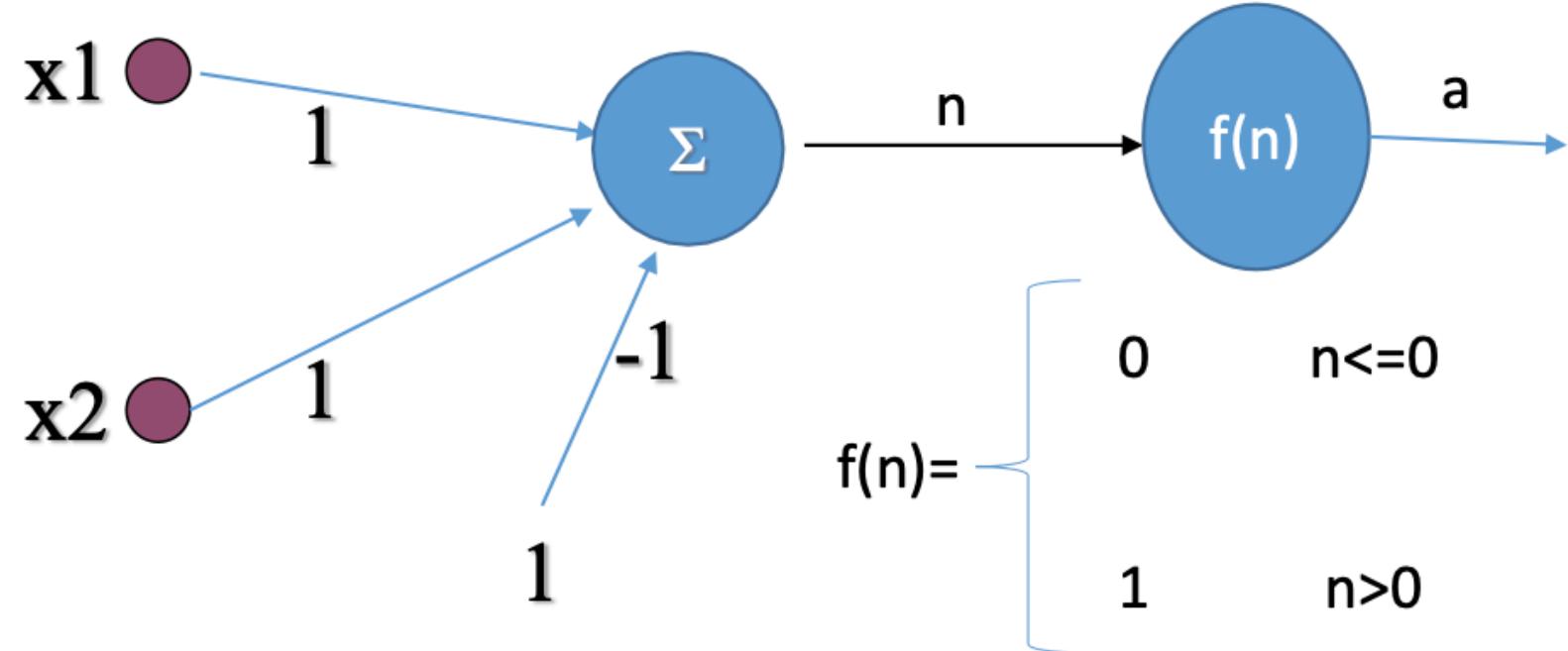
خروجی

# مثال: مدل نرون AND

$x_1$	$x_2$	$x_1$ and $x_2$
1	1	1
1	0	0
0	1	0
0	0	0



$x_1$	$x_2$	$n$	$x_1$ and $x_2$
1	1	1	1
1	0	0	0
0	1	0	0
0	0	-1	0



آیا وزن ها منحصر بفردند؟  
خیر

# یادگیری

فرآیند محاسبه وزن ها و بایاس با استفاده از داده های ورودی و پاسخ ها (اهداف) را یادگیری گویند.

# HEBB LEARNING ALGORITHM

# Hebb

الگوریتم هب چیست؟

یکی از اولین الگوریتم و راحت ترین روش آموزش شبکه عصبی هست. میتوانیم چند ورودی {۰-۱} یا {۰۰-۱۱} داشته باشیم اما خروجی ما یه دونه هست {۰۱-۰۲}

این قانون بیان میکند که "سلول‌های عصبی که به صورت همزمان فعال می‌شوند، ارتباطات قوی‌تری بین خود ایجاد می‌کنند." در واقع، اگر دو نورون با هم فعال شوند، اتصالات سیناپسی بین آنها تقویت می‌شود. این اصل به عنوان مبنای برخی از الگوریتم‌های یادگیری در شبکه‌های عصبی در نظر گرفته شده است.

01

02

03

04

معایب:

شبکه هب با یک نرون میتواند مثال هایی را یاد بگیرد که با یک خط قابل جدا کردن است. همیشه همگرا نخواهد شد.

خروجی اگر صفر باشد بخاطر ضرب خروجی در ورودی، الگوریتم یاد نخواهد گرفت. به دلیل بالا ورودی ۰-۱ بهتر از ۰ است.

# مراحل

01

تمام وزن هار به همراه بایاس را  
صفر میدهیم(تعداد وزن به  
اندازه تعداد داده های ورودی ما  
هست)

[ 0 0 0 ]

02

- آرایه مربوط به ورودی را به  
همراه بایاس بسازید.

$$x_1 = \{ 1,1,1 \}$$

$$x_2 = \{ -1,1,1 \}$$

$$x_3 = \{ 1,-1,1 \}$$

$$x_4 = \{ -1,-1,1 \}$$

$$y = \{ 1,-1,-1,-1 \}$$

03

- اپدیت با استفاده از قانون  
هب

$$\Delta w = xy$$

$$w(\text{new}) = w(\text{old}) + \Delta w$$



$$\Delta w = xy$$

$$w(\text{new}) = w(\text{old}) + \Delta w$$

$$w(\text{new}) = w(\text{old}) + x_1 y_1 = [0\ 0\ 0] + [1\ 1\ 1] \cdot [1] = [1\ 1\ 1]$$

$$\text{Second: } w(\text{new}) = [1\ 1\ 1] + [-1\ 1\ 1] \cdot [-1] = [2\ 0\ 0]$$

$$\text{Third : } w(\text{new}) = [2\ 0\ 0] + [1\ -1\ 1] \cdot [-1] = [1\ 1\ -1]$$

$$\text{Fourth: } w(\text{new}) = [1\ 1\ -1] + [-1\ -1\ 1] \cdot [-1] = [2\ 2\ -2]$$

So, the final weight matrix is [2 2 -2]

## Perceptron learning rules

یک معادله خط مانند

$$ax_1 + bx_2 + c = 0$$

این معادله شامل دو متغیر  $a, b$  هست که باهم بردار ضرایب رو تشکیل میدن.

مثلا بردار ضرایب در بالا  $a, b$  هستند.

یکی از خواص بردار ضرایب، به دست آوردن شب خط است که من شود -  $a/b$

اگر یک نقطه داشته باشیم و بردار ضرایب را با این نقطه جمع کنیم، بردار ضرایب به این نقطه نزدیک میشود. در نتیجه خط من چرخد.

**p = داده‌های ورودی**

**t = هدف برای نمونه**

**a = خروجی شبکه**

**e = t - a**

**w(new) = w(old) + ep**

**b(new) = b(old) + e**

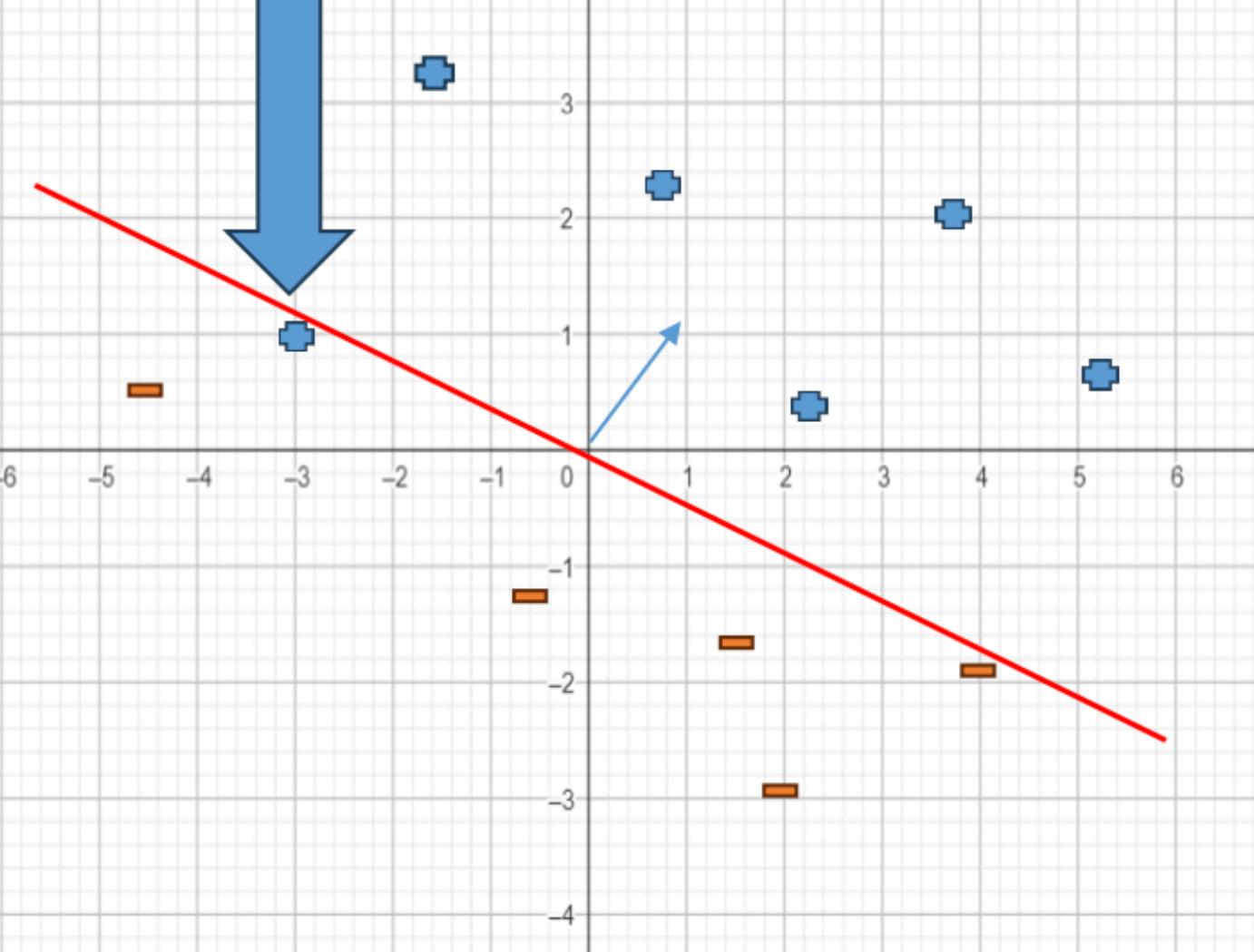
یک شرط خواهیم داشت تا زمانی که بگیم -

به ازای هر ورودی، خواهیم داشت

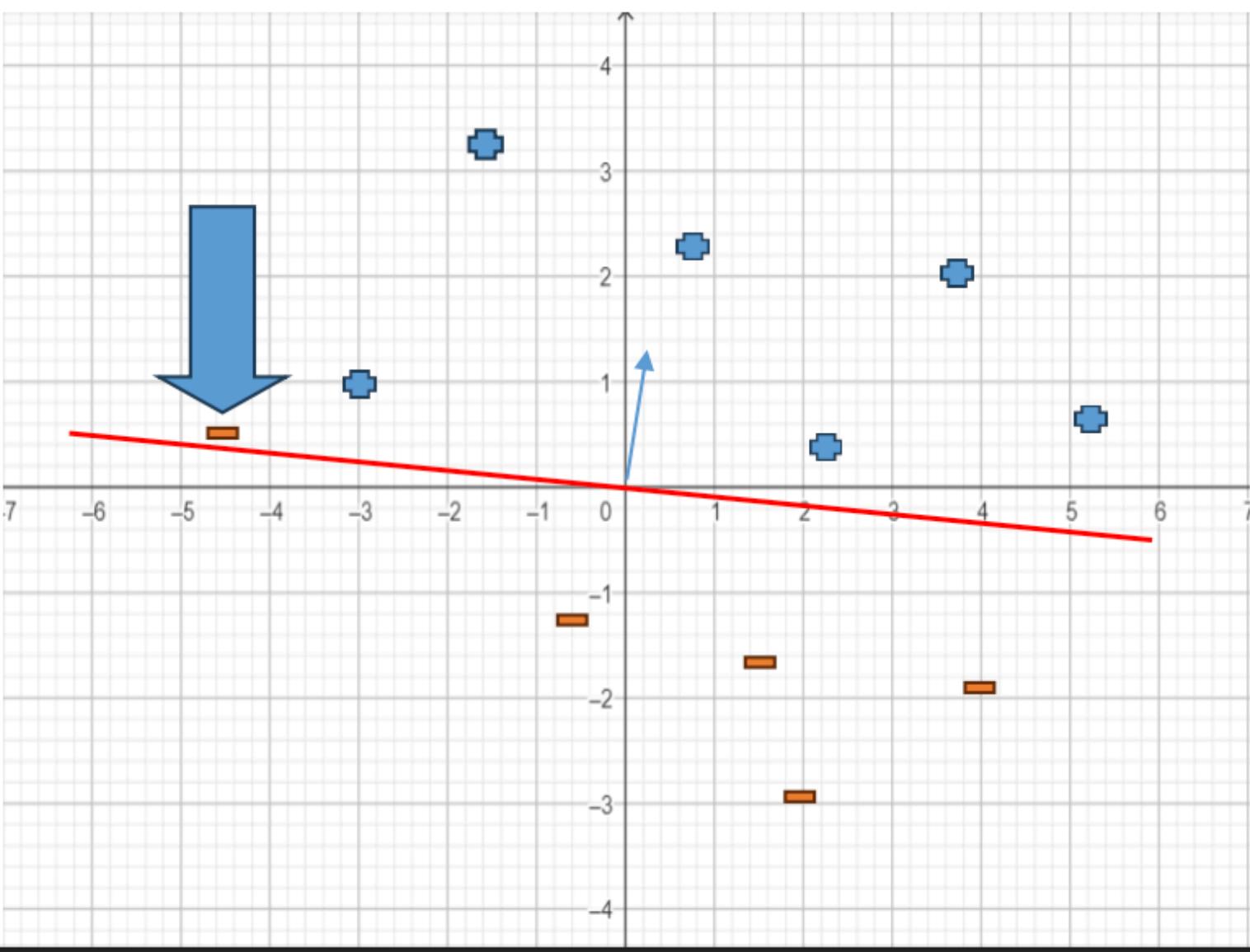
**n=wp+b**

اگر n بزرگتر از . بود ا در غیر این صورت ا

نحوه تغییر خط با توجه نقاط مثبت که اشتباه در سمت اشتباه خط قرار گرفته است.



نتیجه تغییر خط



برای سادگی  $b$  صفر در نظر گرفته می شود.

# Example

<b>p</b>	<b>Target(t)</b>
1	2
-1	0
0	-1

Hard Limit	$a = 0 \quad n < 0$ $a = 1 \quad n \geq 0$		hardlim
Symmetrical Hard Limit	$a = -1 \quad n < 0$ $a = +1 \quad n \geq 0$		hardlims

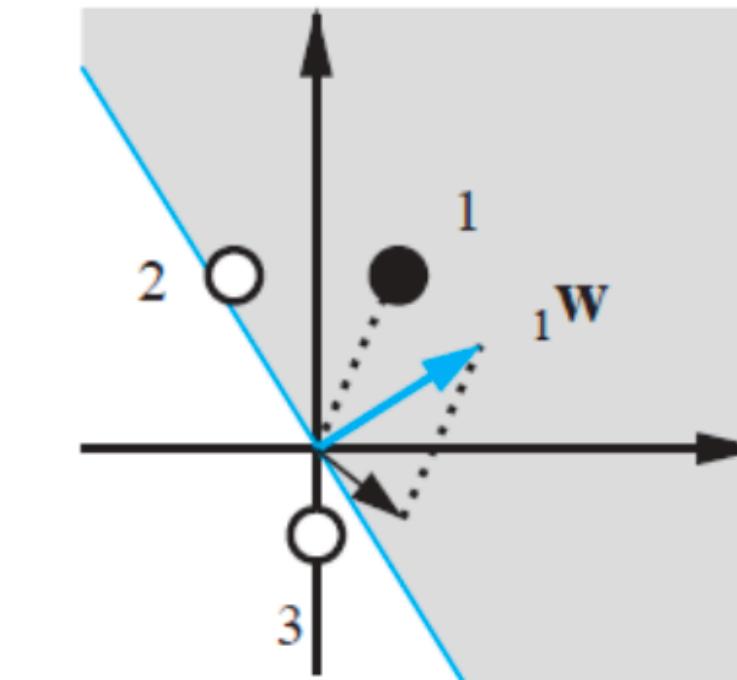
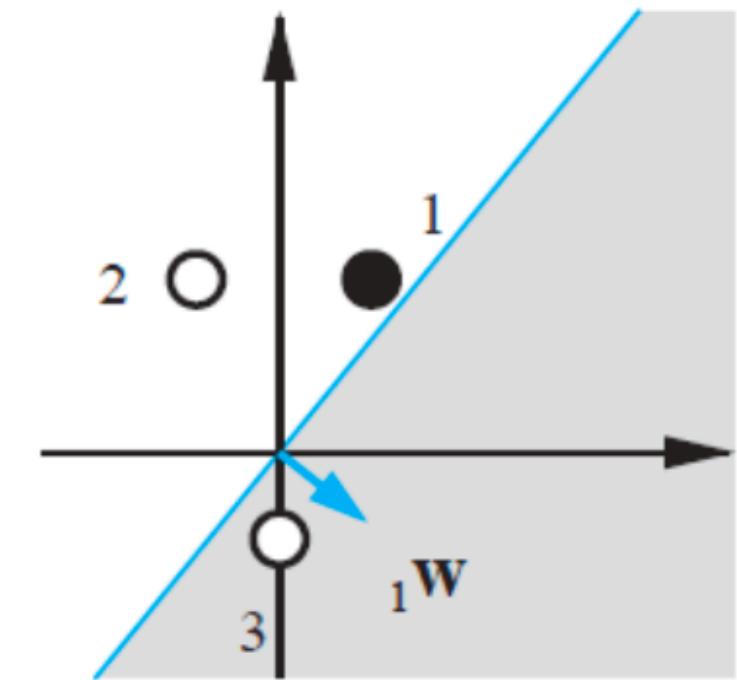
Initialize weights:  ${}_1\mathbf{w}^T = [1.0 \ -0.8]$

$$a = \text{hardlim}({}_1\mathbf{w}^T \mathbf{p}_1) = \text{hardlim}\left([1.0 \ -0.8] \begin{bmatrix} 1 \\ 2 \end{bmatrix}\right)$$

$$a = \text{hardlim}(-0.6) = 0.$$

If  $t = 1$  and  $a = 0$ , then  ${}_1\mathbf{w}^{new} = {}_1\mathbf{w}^{old} + \mathbf{p}$ .

$${}_1\mathbf{w}^{new} = {}_1\mathbf{w}^{old} + \mathbf{p}_1 = \begin{bmatrix} 1.0 \\ -0.8 \end{bmatrix} + \begin{bmatrix} 1 \\ 2 \end{bmatrix} = \begin{bmatrix} 2.0 \\ 1.2 \end{bmatrix}$$



-اگر داده ای از اعداد دیگر فاصله زیادی داشته باشد چه اتفاقی می افتد؟  
در عبارت  $w=w+ep$  وزنها تغییرات زیادی می کنند. بهتر است داده ها نرمال شوند.

-آیا شبکه عصبی پرسپترون در حالتیکه نقاط به صورت خطی جدایزیر باشند. آیا الزاما خط مناسب را می یابد؟  
آیا شبکه همگرا می شود؟  
نکته: در مورد شبکه هب جواب منفی بود.

بله

بعضی از اسلایدها و همچنین توضیحات زیادی از این ارائه، از اسلایدهای درس دکتر پورامینی بزرگوار برداشته شده است. ممنونم از ایشان که اجازه این امر را به بنده دادند.

**pouramini@yahoo.com**

آدرس گیت من، برای دسترسی به کدهای، پرسپیترون و هب

**<https://github.com/mohammadrezaBorjian/learning-files>**