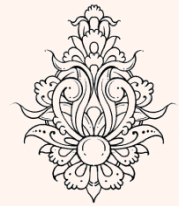
یادگیری عمیق
بخش نخست

Hebb's rule
perceptron
LMS, ADALINE
Gradient decent

دانشگاه شهید بهشتی
پاییز ۱۴۰۰
احمد محمودی ازناوه

http://faculties.sbu.ac.ir/~a_mahmoudi/

فهرست مطالب

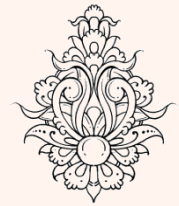


چرا شبکه‌ی عصبی؟

- چگونه می‌توان برنامه‌ای نوشت که هویت یک فرد را از طریق چهره تشخیص دهد یا برنامه‌ای که بتواند اشیاء متفاوت را دسته‌بندی کند؟
- یا برنامه‌ای که با توجه به سابقه‌ی پزشکی و خانوادگی فرد، عمر تقریبی او را حدس بزند!
  - نوشتن چنین برنامه‌هایی بسیار دشوار است، در حالی که مغز انسان ۱۰۰ تا ۲۰۰ میلی‌ثانیه چنین پردازشی را انجام می‌دهد.
  - در این موارد با داده‌های حجیمی روبرو هستیم که ارتباط کاملا دقیق و مشخصی بین آن‌ها برقرار نیست و یا کشف این ارتباط بسیار دشوار است.
  - نمی‌دانیم مغز ما چگونه‌ی چنین کارهایی را انجام می‌دهد.
  - نکته‌ی دیگر این که در مواردی، این ارتباط با مرور زمان تغییر خواهند کرد و این باعث دشواری بیشتر مسأله خواهد شد.

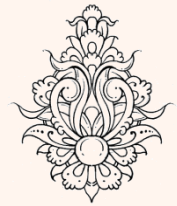
رویکرد یادگیری ماشین

- در الگوریتم‌های «یادگیری ماشین»، تعداد زیادی مثال همراه با پاسخ صحیح دریافت و برنامه‌ای برای حل مسأله تولید می‌کند(رویکرد بانظارت).
  - در صورت انجام درست کار، برنامه برای نمونه‌های جدید هم درست کار خواهد کرد(**تعمیم‌پذیری**).
  - در صورتی که داده‌ها تغییر کنند، برنامه هم توانایی تغییر خواهد داشت(**وفقی بودن**).
- با توجه به افزایش قدرت محاسبات، انجام حجم عظیمی از محاسبات ارزان‌تر از نوشتن یک الگوریتم خاص می‌باشد.



کاربردها

- بازشناسی الگو(دسته‌بندی و رده‌بندی)
  - تشخیص اشیاء، تشخیص کاراکتر، تشخیص چهره یا تشخیص حالات چهره، تشخیص واژه‌ها
  - پیش‌بینی لیست فیلم‌های مورد علاقه‌ی یک شخص
- رگرسیون
  - پیش‌بینی قیمت سهام
- تشخیص ناهنجاری
  - استفاده از کارت اعتباری به صورت نامتعارف



مثال

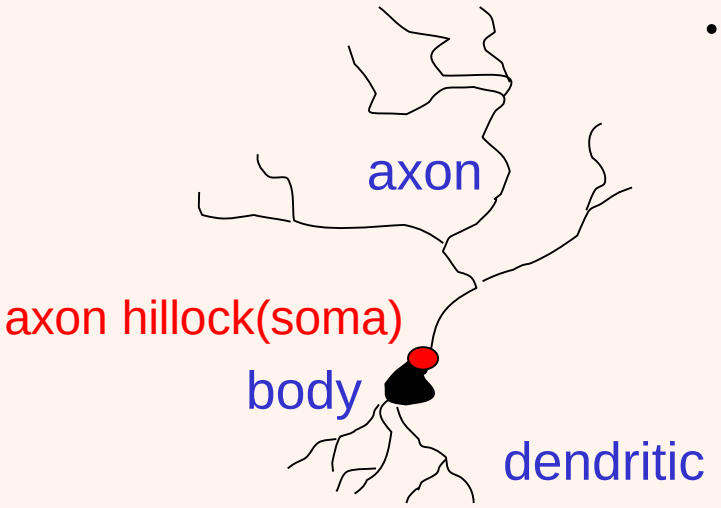
- آموختن یک شیوه‌ی نگارش
  - تشخیص متون Shakespeare
- تشخیص خلوص روغن زیتون
- برچسب‌زدن تصاویر
- تبدیل صوت از یک زبان به زبان دیگر
- یافتن بهترین اهداکننده برای اهدای قلب
- سیستم‌های تحلیل ریسک

شبکه‌های عصبی مصنوعی

- ایده‌ی اصلی این شبکه‌ها مبتنی بر «شبکه‌های عصبی زیستی» است.
- بسیاری از مسائل توسط انسان به سادگی قابل حل می‌باشد.
- مغز به صورت موازی محاسبات را انجام می‌دهد.
- این مدل می‌تواند برای مسائلی که توسط ذهن آدمی به راحتی انجام می‌شود، مفید باشد.
- در واقع شیوه‌ی به کار رفته در ذهن به نوعی الهام بخش ارائه‌ی مدلی برای ایجاد قابلیت‌هایی مشابه با مغز است، هرچند شبکه‌ی عصبی مورد استفاده‌ی ما تفاوت‌های بسیاری با شبکه‌های عصبی زیستی دارد.

ساختار یک نورون طبیعی

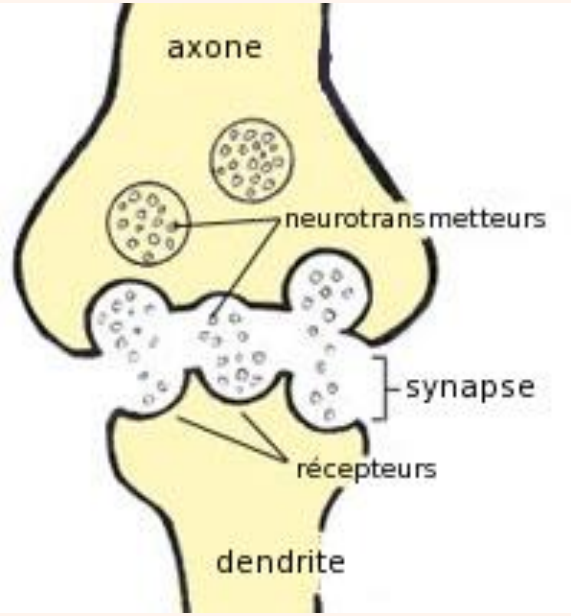
- مغز انسان شامل حدود 10^{11} نورون است که به صورت فوق‌العاده‌ای به هم پیوسته هستند که هر نورون به طور متوسط با 10^{4} نورون دیگر مرتبط است.
- یافته نشان می‌دهند داده‌ها در اتصالات بین نرون‌ها ذخیره می‌شود.
- شامل یک آسه(آکسون) است که شاخه شاخه شده و پیام‌های الکتریکی را به بیرون یاخته هدایت می‌کند.
- یک خوشه از دارینه(دندریت)ها که پیام‌های الکتریکی را از سلول‌های مجاور دریافت می‌کند.



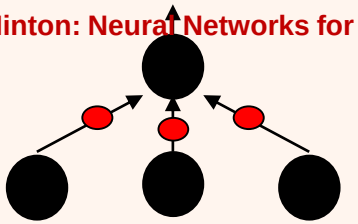
(Hinton: Neural Networks for Machine Learning)

ساختار یک نورون طبیعی(ادامه...)

- همایه(سیناپس) یک ساختار زیستی در پایانه آکسون‌ها است که از راه آن یک سلول عصبی پیام خود را به دندریت یک نورون دیگر یا یاخته ماهیچه‌ای یا یک غده می‌فرستد.
- جسم سلولی مولد این سیگنال‌های ارسالی است. در صورتی که میزان سیگنال دریافتی از طریق دارینه‌ها از یک حد آستانه بیشتر باشد؛ نورون تحریک می‌شود.



(Hinton: Neural Networks for Machine Learning)



مغز چگونه کار می‌کند؟

- هر نورون از نورون‌های دیگری ورودی دریافت می‌کند.
- برخی نورون‌های به سلول‌های گیرنده(receptor) متصل هستند.
- نورون‌ها با ارسال سیگنال‌های الکتریکی با یکدیگر ارتباط برقرار می‌کنند.
- اثر هر ورودی به وزن ارتباط سیناپسی بستگی دارد.
- این وزن‌ها به صورت وفقی تغییر می‌یابند تا کل شبکه محاسبات را به درستی انجام دهد.

مغز چگونه کار می‌کند؟(ادامه...)

- هر بخش قشر مغز وظیفه‌ای خاص دارد.
  - آسیب به هر بخش از مغز یک انسان بالغ، باعث تاثیرات خاصی می‌شود.
  - در صورت انجام فعالیت‌های خاص جریان خون در بخشی از بخش‌ها افزایش می‌یابد.
- بخش‌های مختلف قشر مغز(cortex) بسیار شبیه به هم هستند.
  - در صورتی که در بخشی از آن آسیب ببیند، بخش دیگر می‌تواند عهده‌دار وظیفه‌ی آن بخش شود، در واقع به نظر می‌رسد همه‌ی بخش‌ها از یک شیوه‌ی یادگیری استفاده می‌کنند.
  - در یک آزمایش سیگنال‌های بینایی موش‌خرما به بخش شنوایی مغزش هدایت شد و بخش شنوایی دیدن را آموخت!

Visual Projections Routed to the Auditory Pathway in Ferrets: Receptive Fields of Visual Neurons in Primary Auditory Cortex

Anna W. Roe,[a] Sarah L. Pallas,[b] Young H. Kwon, and Mriganka Sur
Department of Brain and Cognitive Sciences, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139

شبکه‌ی عصبی مصنوعی

- شبکه‌ی عصبی پردازشگری با ساختار توزیع شده و قابلیت بالای موازی‌سازی است که از واحدهای پردازشگر ساده‌ای تشکیل شده است و قابلیت ذخیره کردن تجربیات و به کارگیری آن برای استفاده‌های آتی را دارا می‌باشد.
  - از طریق یادگیری از محیط اطراف کسب دانش می‌کند.
  - برای ذخیره‌سازی دانش از وزن‌های سیناپسی استفاده می‌کند.
- عمده مطالب این درس، در مورد نحوه‌ی تنظیم این وزن‌هاست تا بتواند مسائل خاصی را حل کنند.

ویژگی‌های شبکه‌های عصبی مصنوعی

- پردازش موازی (سرعت بالا)
- تحمیل‌پذیری
- محاسبات غیرخطی
- برقراری ارتباط یک‌سری ورودی و یک‌سری خروجی
  - بازیابی اطلاعات
- توانایی تطبیق (adaptivity)
- پاسخ به داده‌های نویزی
- تحمل‌پذیری خطا
- یادگیری

نیازمندی‌های شبکه‌های عصبی مصنوعی

- جمع‌آوری و آنالیز مناسب داده
- طرح، آموزش و تست شبکه‌ی عصبی
- بهنجار کردن(normalize) ورودی‌ها:
  - تغییرات باید به نحوی باشد که قابل برگشت بوده و هیستوگرام ورودی را تغییر ندهد.

data leakage(target leakage)

نشتی داده زمانی رخ می‌دهد که اطلاعاتی استفاده شود که هنگام پیش‌بینی معمولا در دسترس نیست

Hyperparameter tuning

تاریخچه‌ی مختصر

- ۱۹۴۳، مفهوم نورون McCulloch&Pitts
- ۱۹۴۹، قانون آموزش Hebb
- ۱۹۵۸، مفهوم پرسپترون Rosenblatt
- ۱۹۶۰، Adaline توسط Widrow&Hoff
- ۱۹۶۹، نقد شبکه‌ی عصبی Minsky&Papert
- ۱۹۷۲، شبکه‌های رقابتی و حافظه‌ی تداعی‌گر
- ۱۹۸۰، الگوریتم یادگیری پس‌انتشار خطا
- ۲۰۱۵~، دنیاگیری شبکه‌های عصبی مصنوعی عمیق

McCulloch and Pitts 1943

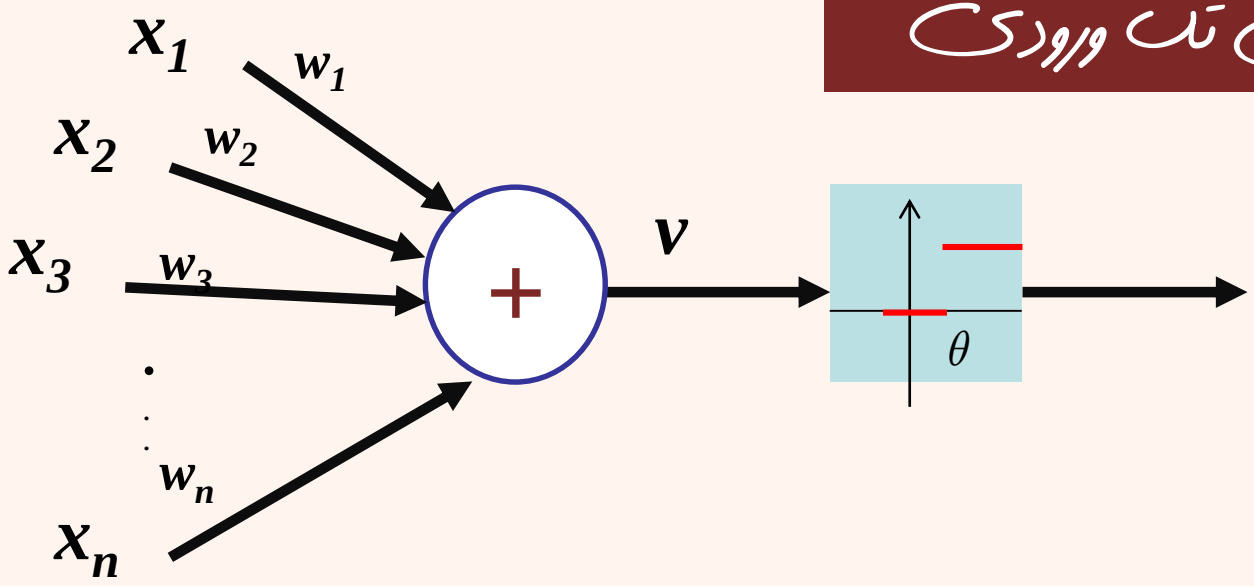
A LOGICAL CALCULUS OF THE IDEAS IMMANENT IN NERVOUS ACTIVITY

WARREN S. MCCULLOCH and WALTER H. PITTS

مدل نورون

- کوچک‌ترین واحد پردازشگر اطلاعات

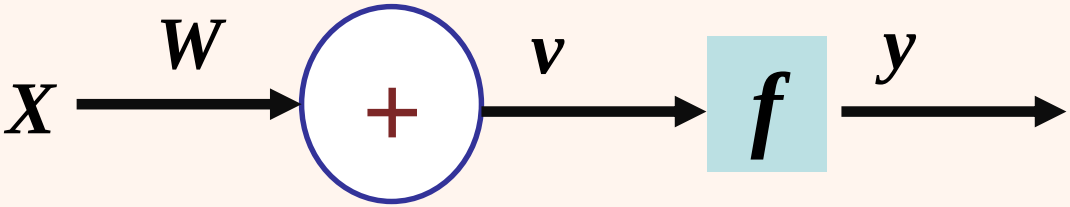
ساختار نورون تک ورودی



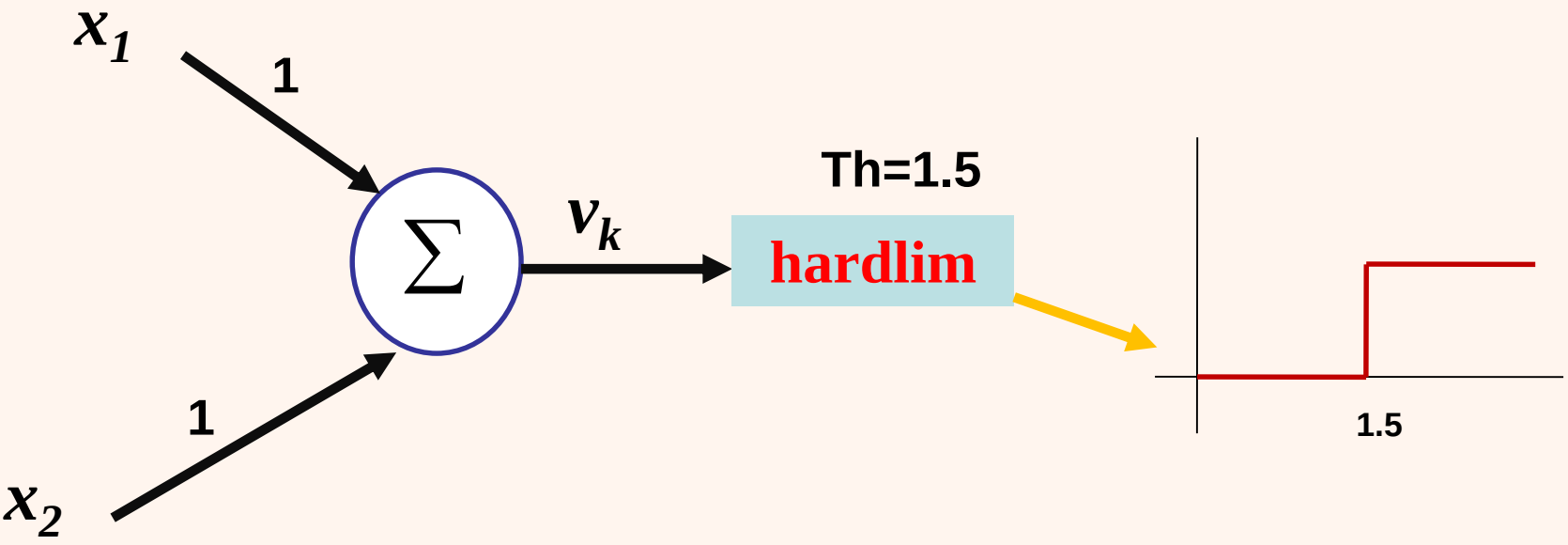
$$v = W.X$$

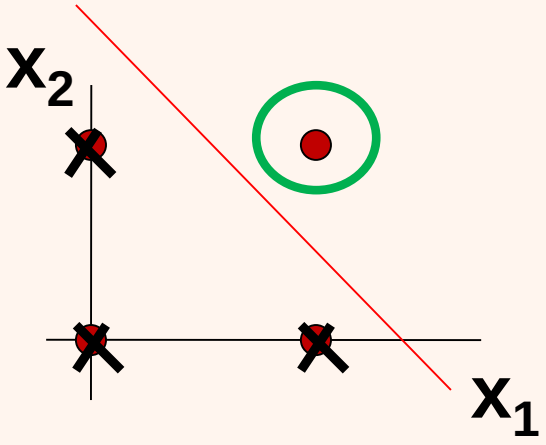
$$y = f(v)$$

$$y = f(W.X)$$

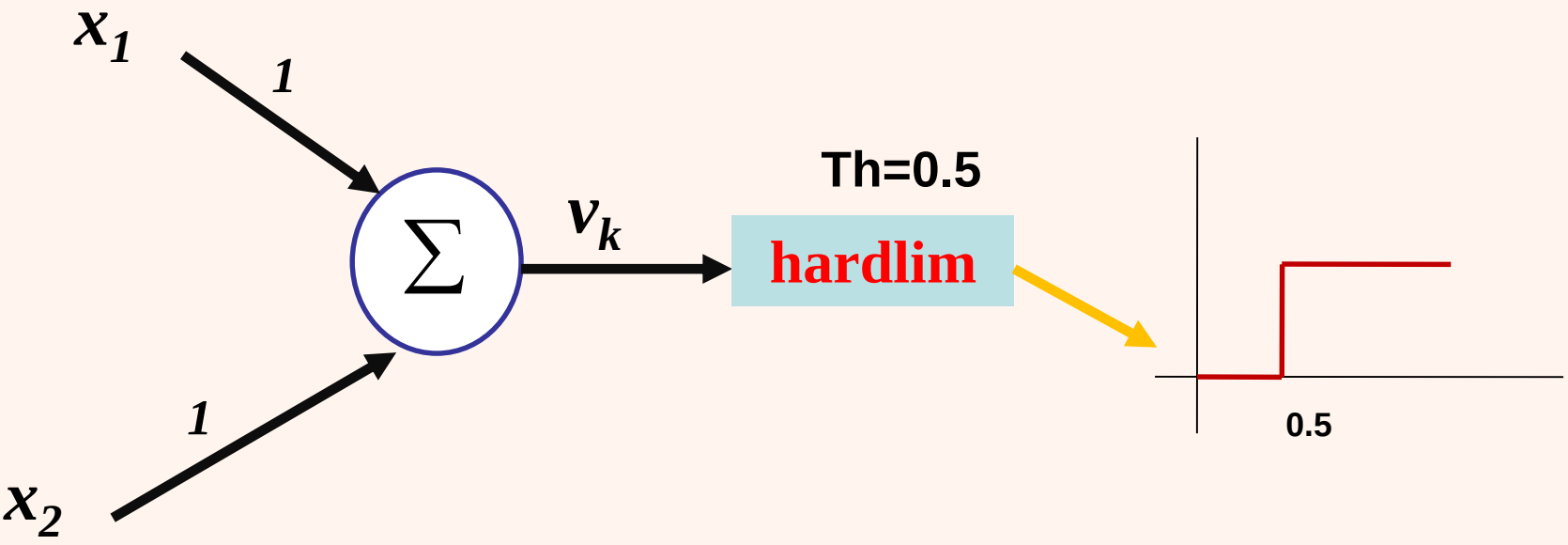


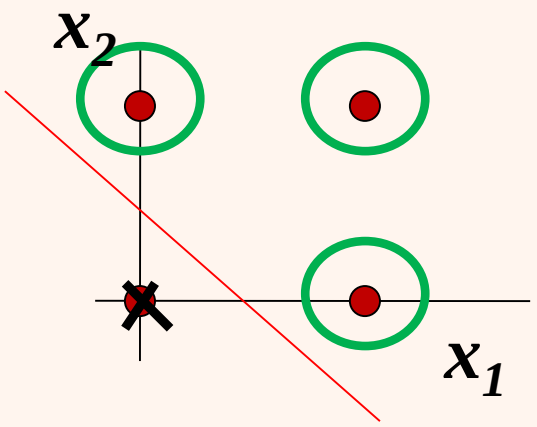
AND Gate





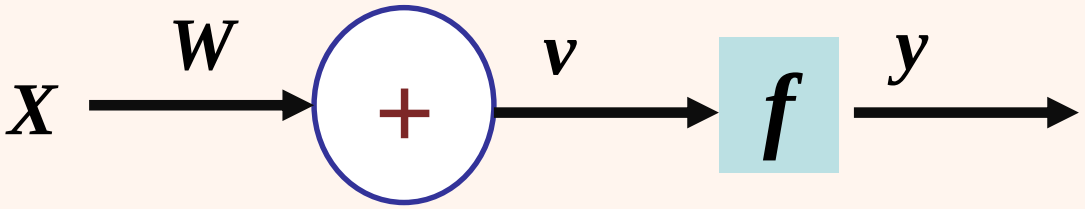
OR Gate





مدل نورون (ادامه...)

- به دو صورت می‌توان چنین نرونی را نمایش داد:



$$v = \sum_i x_i w_i$$

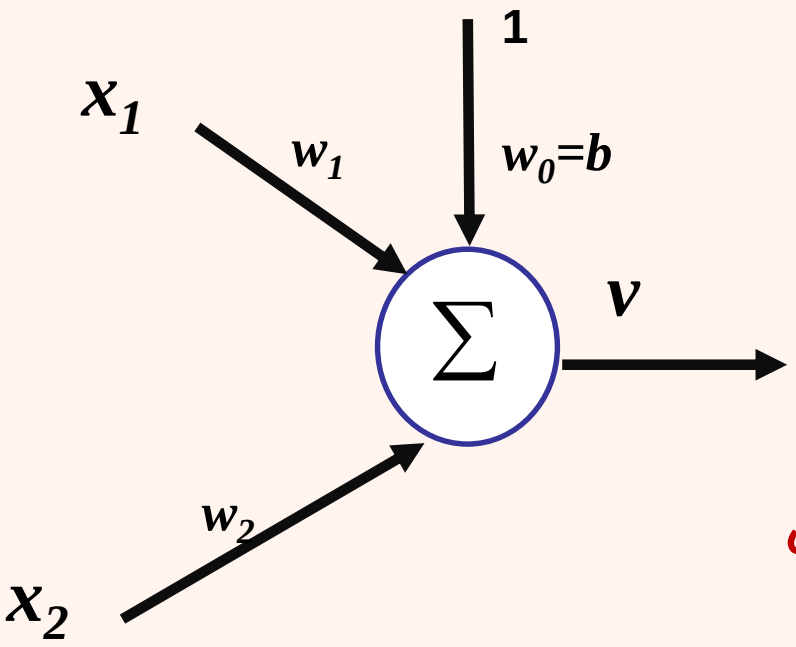
$$y = \begin{cases} 1 \text{ if } v \geq \theta \\ 0 \text{ otherwise} \end{cases}$$

$$v = b + \sum_i x_i w_i$$

$$q = -b$$

$$y = \begin{cases} 1 \text{ if } v \geq 0 \\ 0 \text{ otherwise} \end{cases}$$

بایاس(سوگیری)



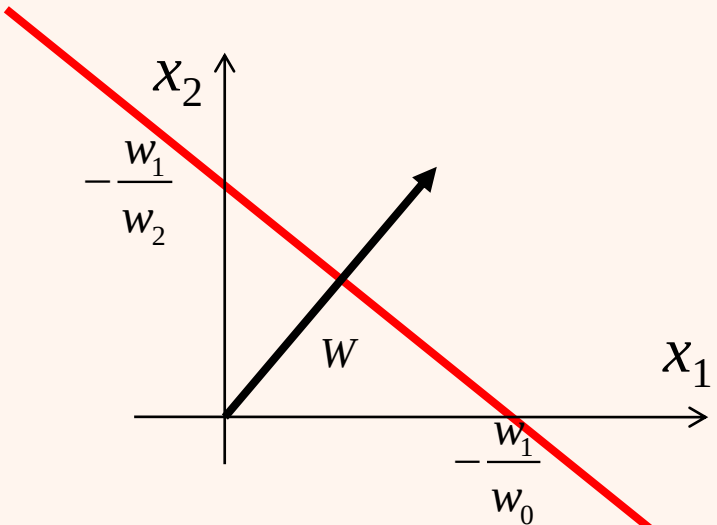
$$v = w_1 x_1 + w_2 x_2 + w_0$$

$$x_2 = -\frac{w_1}{w_2} x_1 - \frac{w_0}{w_2}$$

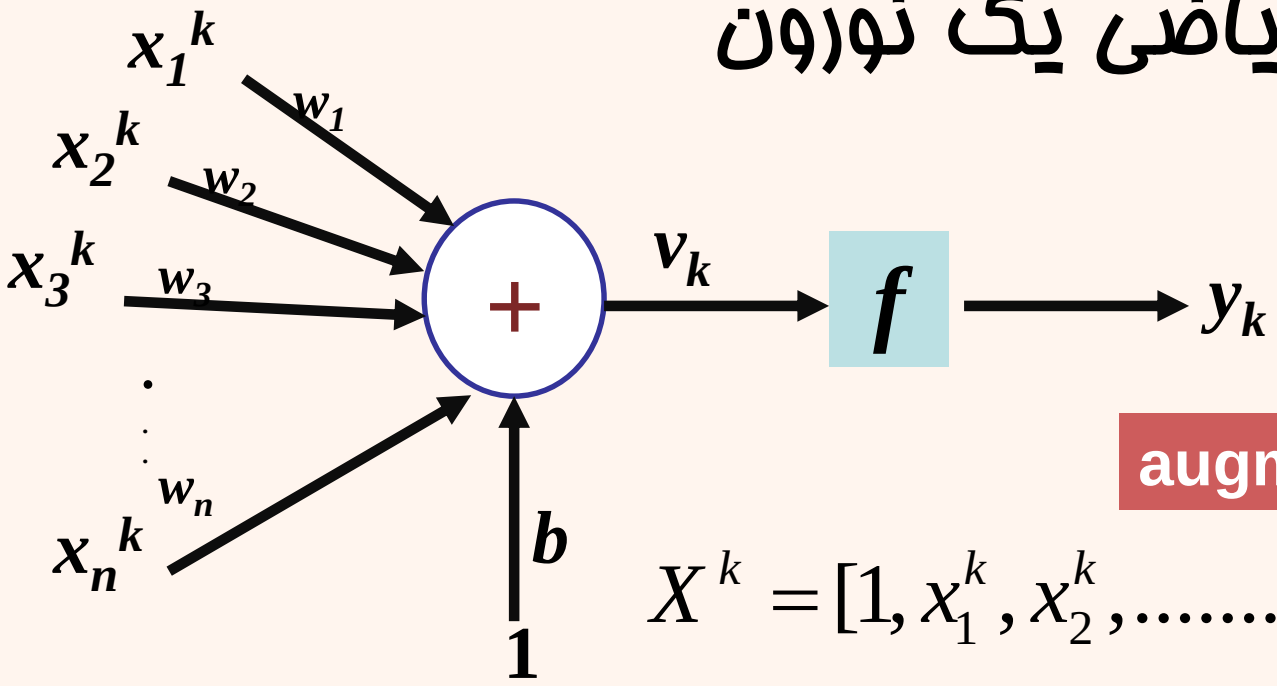
شیب

عرض از مبدا

- به جای تغییر آستانه می‌توان بایاس را تغییر داد.



مدل ریاضی یک نورون



augmented output

$$X^k = [1, x_1^k, x_2^k, \ldots\ldots, x_n^k]$$

$$W = [w_0 = b, w_1, w_2, \ldots\ldots, w_n]$$

$$u_k = \sum_{i=1}^{n} w_i . x_i^k \quad \Rightarrow \quad v_k = u_k + b$$

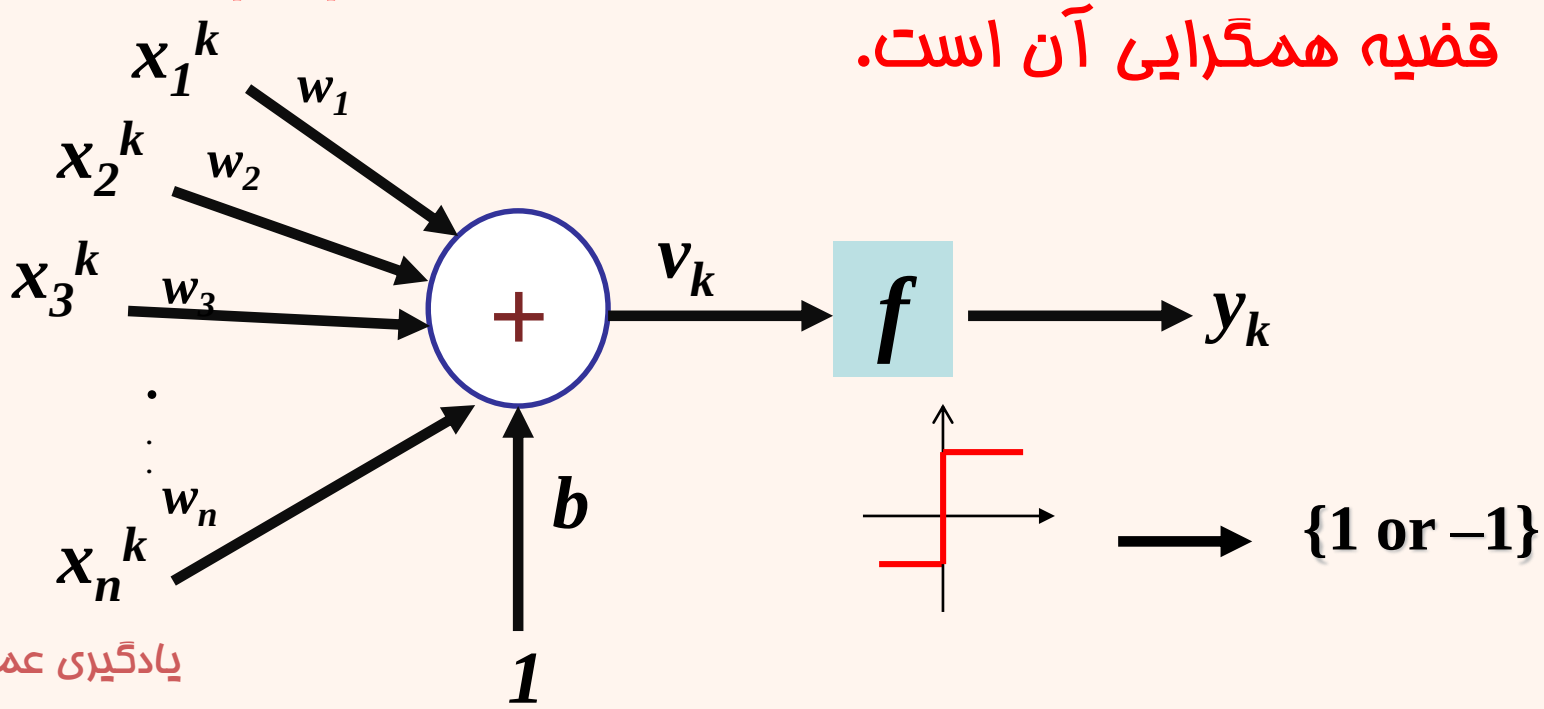
$$v_k = W . X^k$$

$$y_k = f\left(\sum_{i=0}^{n} w_i . x_i^k + b \right)$$

Rosenblat 1957

پرسپترون

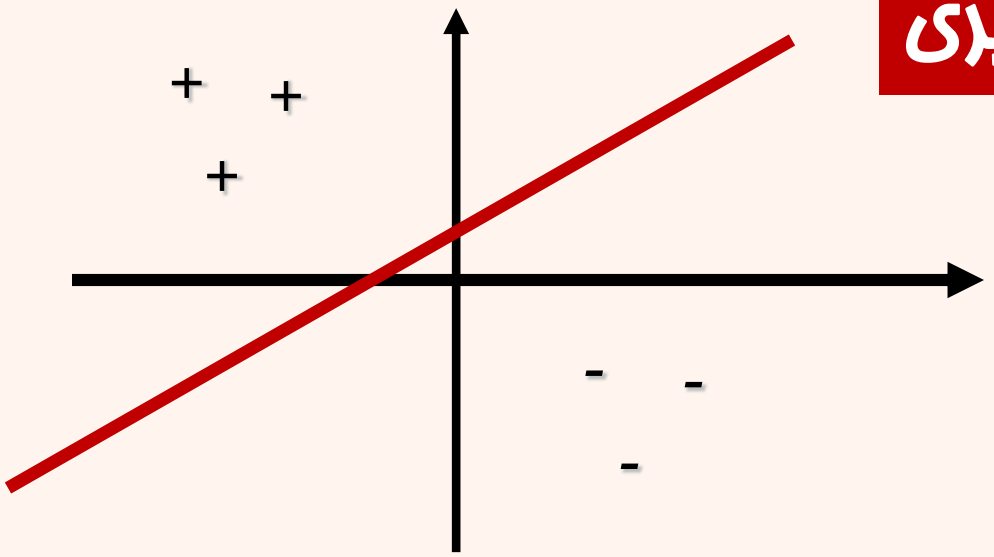
- یک پرسپترون یک بردار ورودی را گرفته، ترکیبی خطی از آن‌ها را محاسبه نموده، خروجی را فراهم می‌آورد.
- اگر خروجی از میزان آستانه‌ای بالاتر بود یک و در غیر این‌صورت صفر(منهای یک) باز می‌گرداند.
- مهمترین مساله مطرح شدن قانون آموزش پرسپترون و قضیه همگرایی آن است.



پرسپترون

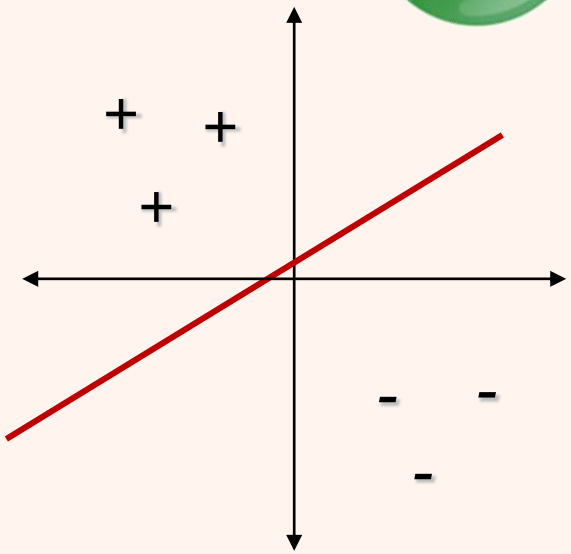
- پرسپترون توانایی جداسازی داده‌های جدایی‌پذیر خطا را داراست.
- می‌توان آن را به صورت یک جداکننده دودویی در نظر گرفت.

مرز تصمیم‌گیری

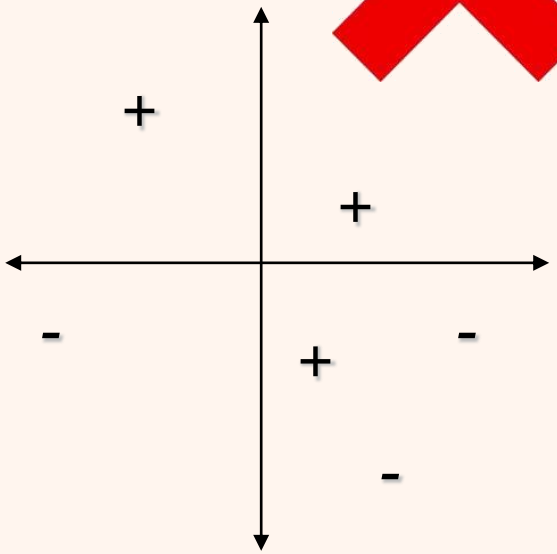


مثال

جدایی‌پذیر خطی



جدایی‌پذیر غیرخطی

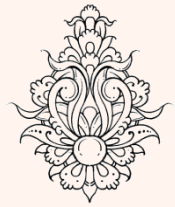


مراحل طراحی یک شبکه‌ی عصبی و الگوریتم‌های آموزش

- انتخاب وزن‌ها به صورت تصادفی
- اعمال مجموعه‌ی آموزشی (training set)

$$M = \left\{ \left(X^1, d^1 \right), \left(X^2, d^2 \right), \cdots \right\}$$

  - اعمال هر ورودی به شبکه و به دست آوردن خروجی
  - مقایسه‌ی خروجی مطلوب و واقعی
  - آموزش شبکه به صورت تغییر وزن‌ها و در جهت نزدیک شدن خروجی مطلوب و واقعی



Hebb 1949

- فرضیه‌ی مطرح شده توسط Hebb در حال حاضر بر روی تحقیقات عصب‌شناسی مؤثر است.
- این فرضیه پیشتر نیز به بیان‌های مختلف مطرح شده بود.

When an axon of cell A is near enough to excite a cell B and repeatedly or persistently takes part in firing it, some growth process or metabolic change takes place in one or both cells such that A's efficiency, as one of the cells firing B, is increased

- در بخش‌های بعدی دوباره با این قانون مواجه خواهیم شد.

قانون آموزش پرسپترون

- مقادیر ورودی ۱ و ۱- هستند.
- تابع فعال‌ساز(انگیزش): تابع علامت

$$y(t) = f\left[\sum_i w_i(t) x_i\right]$$

$y(t)$ is correct $\qquad w_i(t+1) = w_i(t)$

$y(t)$ is not correct

$y(t) = -1 \qquad w_i(t+1) = w_i(t) + x_i$

$y(t) = 1 \qquad w_i(t+1) = w_i(t) - x_i$

اولین قانون آموزش(ادامه...)

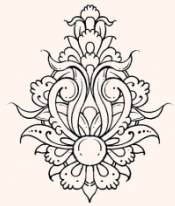
Perceptron learning rule

- و بدین شکل در یک رابطه تجمیع شد:

$$y(t) \text{ is correct} \qquad w_i(t+1) = w_i(t)$$

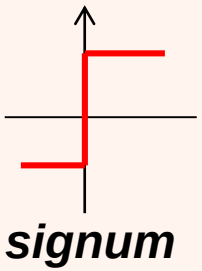
$$y(t) \text{ is not correct} \qquad w_i(t+1) = w_i(t) + d^k x_i^k$$

- در صورتی که تابع انگیزش به صورت غیرنزولی باشد، بدین ترتیب تغییر وزن‌ها باعث کاهش خطا می‌شود.



مثال ۱

$$X^1=[1 \quad -1 \quad -1 \quad -1], \quad d^1=1$$
$$X^2=[1 \quad 1 \quad -1 \quad -1], \quad d^2=-1$$
$$X^3=[1 \quad 1 \quad 1 \quad 1], \quad d^3=1$$



$t=0, \quad W=[0 \quad 0 \quad 0 \quad 0];$ X^1 ✓ $W_{new}=W_{old}$

$t=1, \quad W=[0 \quad 0 \quad 0 \quad 0];$ X^2 ✗ $W_{new}=W_{old}-X^2;$

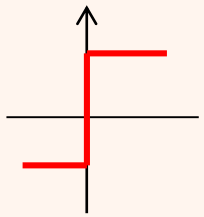
$t=2, \quad W=[-1 \quad -1 \quad 1 \quad 1];$ X^3 ✓ $W_{new}=W_{old}$

$t=3, \quad W[-1 \quad -1 \quad 1 \quad 1];$ X^1 ✗ $W_{new}=W_{old}+X^1;$

$t=4, \quad W=[0 \quad -2 \quad 0 \quad 0];$ X^2 ✓ $W_{new}=W_{old}$

مثال ۲

$$X^1=[1 \quad -1 \quad -1 \quad -1], \quad d^1=1$$
$$X^2=[1 \quad 1 \quad -1 \quad -1], \quad d^2=-1$$
$$X^3=[1 \quad 1 \quad 1 \quad 1], \quad d^3=1$$



signum

$t=5, \quad W=[0 \quad -2 \quad 0 \quad 0];$ X^3 ✗ $W_{new}=W_{old}+X^3;$

$t=6, \quad W=[1 \quad -1 \quad 1 \quad 1];$ X^1 ✓ $W_{new}=W_{old}$

$t=7, \quad W=[1 \quad -1 \quad 1 \quad 1];$ X^2 ✓ $W_{new}=W_{old}$

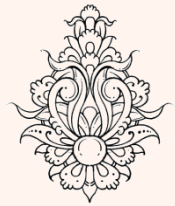
$t=8, \quad W=[1 \quad -1 \quad 1 \quad 1];$ X^3 ✓ $W_{new}=W_{old}$

مثال

```
P=[ 1  1  1;
    -1  1  1;
    -1 -1  1;
    -1 -1  1];
T=[1 -1 1];
W=[0 0 0 0];
flg=0;
nit=0;
```

```
while flg~=length(T)
    disp('itr:');
    disp(nit);
    k=mod(nit, length(T))+1;
    y=hardlims(W*P(:,k));
    if y==T(k)
        flg=flg+1;
    else
        flg=0;
        W=W+T(k)*P(:,k)';
        disp(W);

    end
    nit=nit+1;
end
```



Convergence theorem

قضیه‌ی همگرایی

- در صورت که مجموعه وزن‌های W^* وجود داشته باشد که قابلیت جداسازی یک مجموعه‌ی محدود (جدایی‌پذیر خطی) را داشته باشد، قانون آموزش پرسپترون به یک پاسخ همگرا خواهد شد.
  - این پاسخ الزاما با W^* یکسان نخواهد بود.
  - تمام خروجی‌ها را به گونه‌ای تغییر می‌دهیم که خروجی مطلوب «+۱» شود.
  - وزن اولیه را صفر در نظر می‌گیریم.
  - بردار ورودی n-تایی است.

$$X^k = [1, x_1^k, x_2^k, ..., x_n^k]$$

اثبات قضیه‌ی همگرایی

- هدف محاسبه‌ی حداکثر تعداد مراحلی است که وزن‌ها باید اصلاح شوند. با توجه به مفروضات

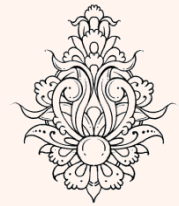
$$\forall k,\ \exists \delta \geq 0 \quad W^{*}.X^{k} \geq \delta,$$

- فرض کنید در مرحله‌ی $t+1$ نیاز به اصلاح وزن‌ها وجود دارد:

$$W_{(t+1)} = W_{(t)} + d^{k} X^{k}$$

$$W^{*}.W_{(t+1)} = W^{*}.W_{(t)} + W^{*} X^{k}$$

$$W^{*}.W_{(t+1)} \geq W^{*}.W_{(t)} + \delta \quad \Rightarrow \quad \boxed{W^{*}.W_{(t)} \geq t\,\delta}$$



اثبات قضیه‌ی همگرایی(ادامه ...)

$$\left\|W_{(t+1)}\right\|^2 = W_{(t+1)}W_{(t+1)}^T = \left[W_{(t)} + d^k X^k\right]\left[W_{(t)} + d^k X^k\right]^T$$

این مقدار منفی است

$$\left\|W_{(t+1)}\right\|^2 = \left\|W_{(t)}\right\|^2 + \left\|X^k\right\|^2 + 2W_{(t)}\left[X^k\right]^T$$

$$\left\|W_{(t+1)}\right\|^2 \le \left\|W_{(t)}\right\|^2 + \left\|X^k\right\|^2$$

(n+1)

$$\left\|W_{(t+1)}\right\|^2 \le \left\|W_{(t)}\right\|^2 + (n+1) \Rightarrow \left\|W_{(t)}\right\|^2 \le t(n+1)$$

اثبات قضیه‌ی همگرایی(ادامه ...)

$$W^{*}.W_{(t)} \geq t\delta$$

$$\cos(\theta) = \frac{W^{*}.W_{(t)}}{\|W^{*}\|\|W_{(t)}\|} \leq 1$$

$$\|W^{*}\|\|W_{(t)}\| \geq t\delta$$

$$\|W_{(t)}\|^2 \leq t(n+1)$$

$$\|W^{*}\|\sqrt{t(n+1)} \geq t\delta$$

$$t \leq \frac{\|W^{*}\|^2 (n+1)}{\delta^2}$$

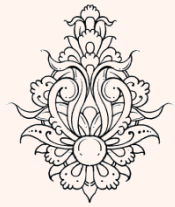
LMS(Least Mean Square)

1960

Widrow and his graduate student Hoff introduced ADALINE network and learning rule which they called the LMS(Least Mean Square) Algorithm.

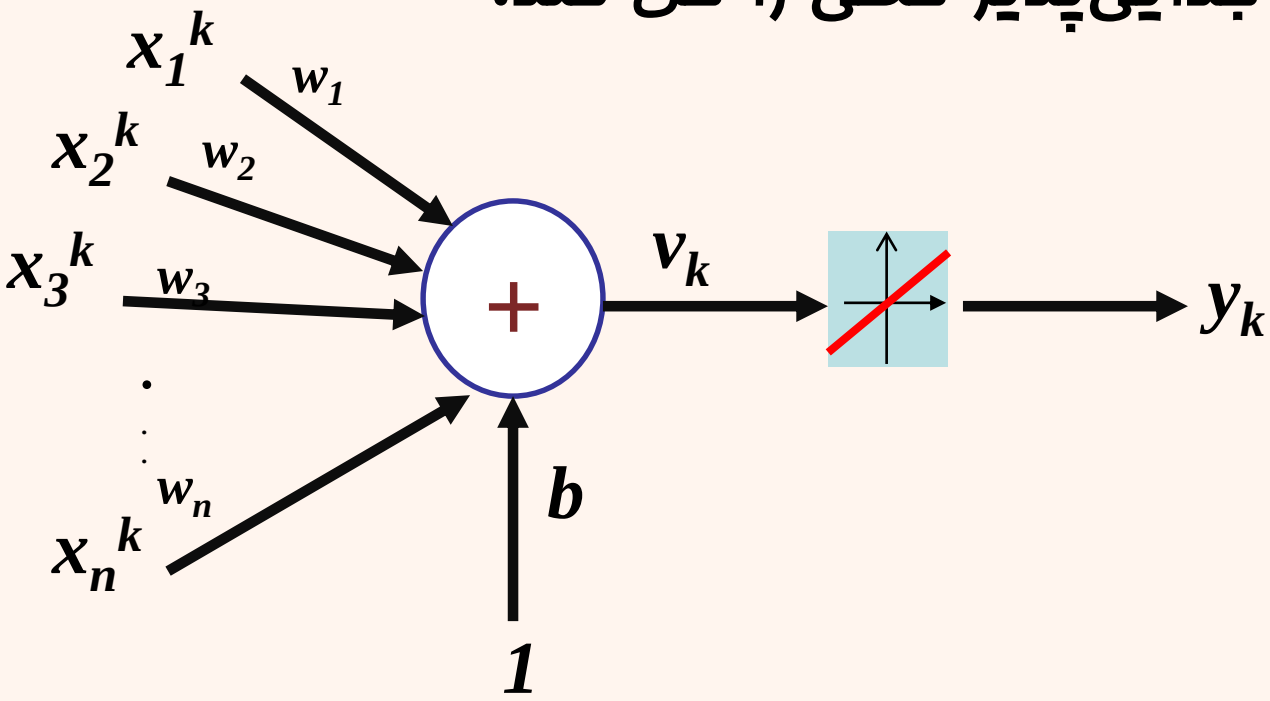
$$w_{new} = w_{old} + \Delta w$$

- برای تولید وزن‌های جدید از تأثیر خطا استفاده می‌شود.
- در این شیوه میزان *به‌روزنمایی* متناسب با میزان خطا خواهد بود و در نتیجه همگرایی سریع‌تر صورت می‌گیرد.



ADALINE

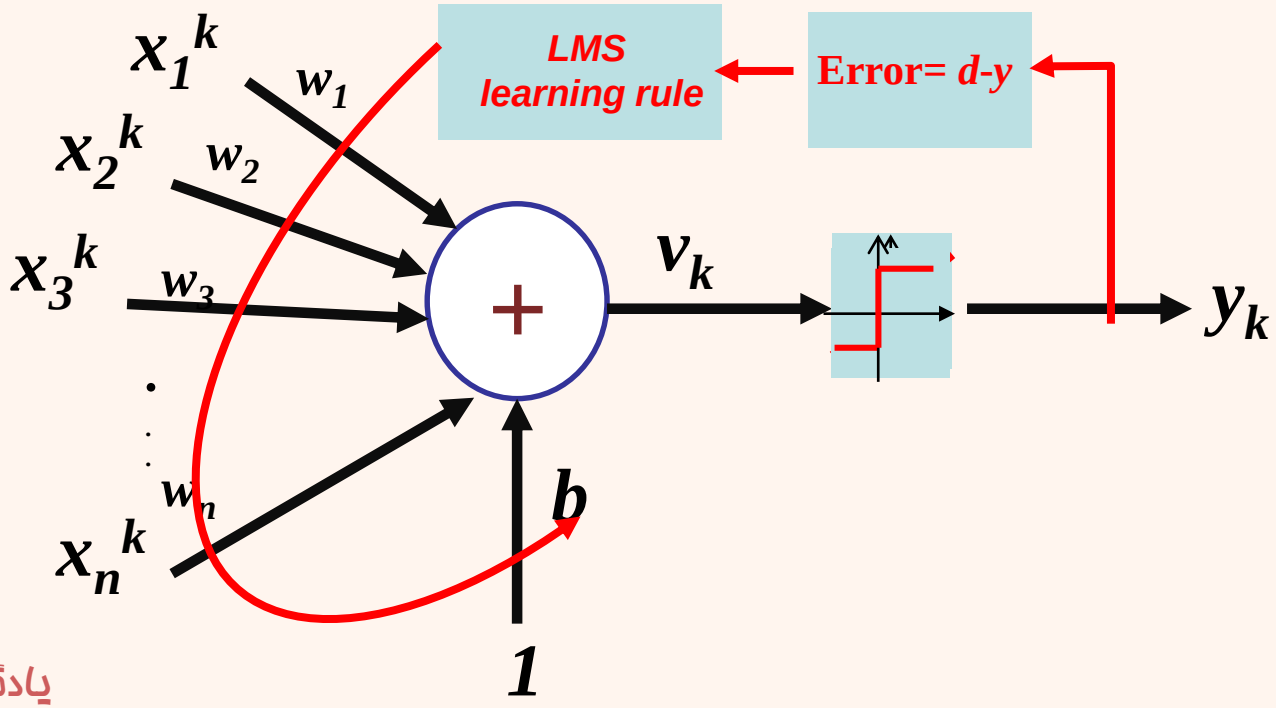
- ADALINE همانند پرسپترون است تنها تابع آن به جای دوسطحی بودن (که مقادیر ۱ و ۱-(0) را به خود اختصاص می‌دهد) تابعی خطی است.
- ADALINE همانند پرسپترون می‌تواند مسائل جدایی‌پذیر خطی را حل کند.



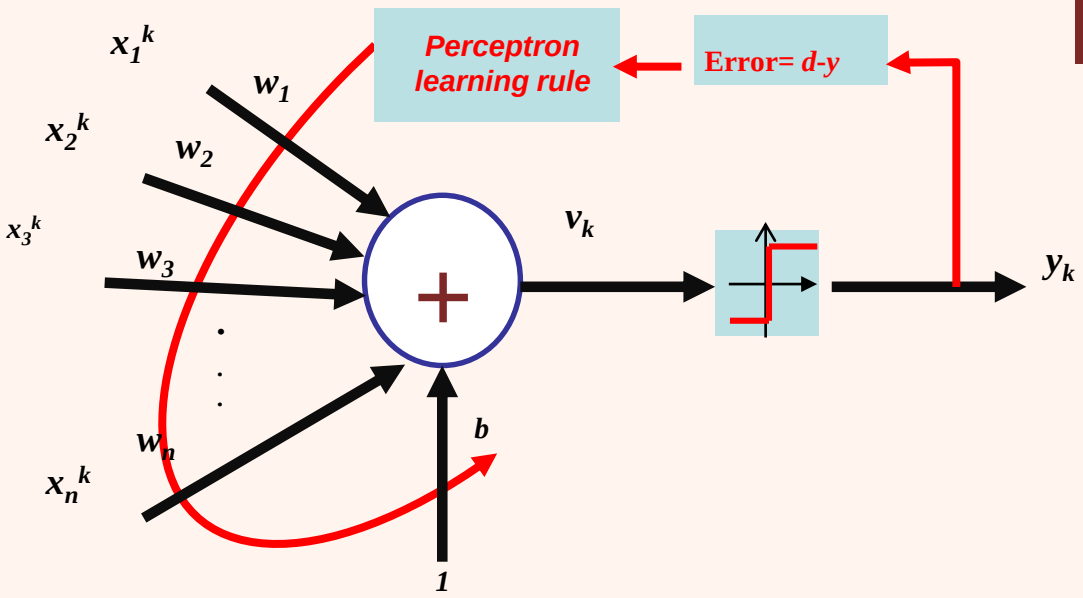
Widrow-Hoff Learning Rule

LMS(Least Mean Square)

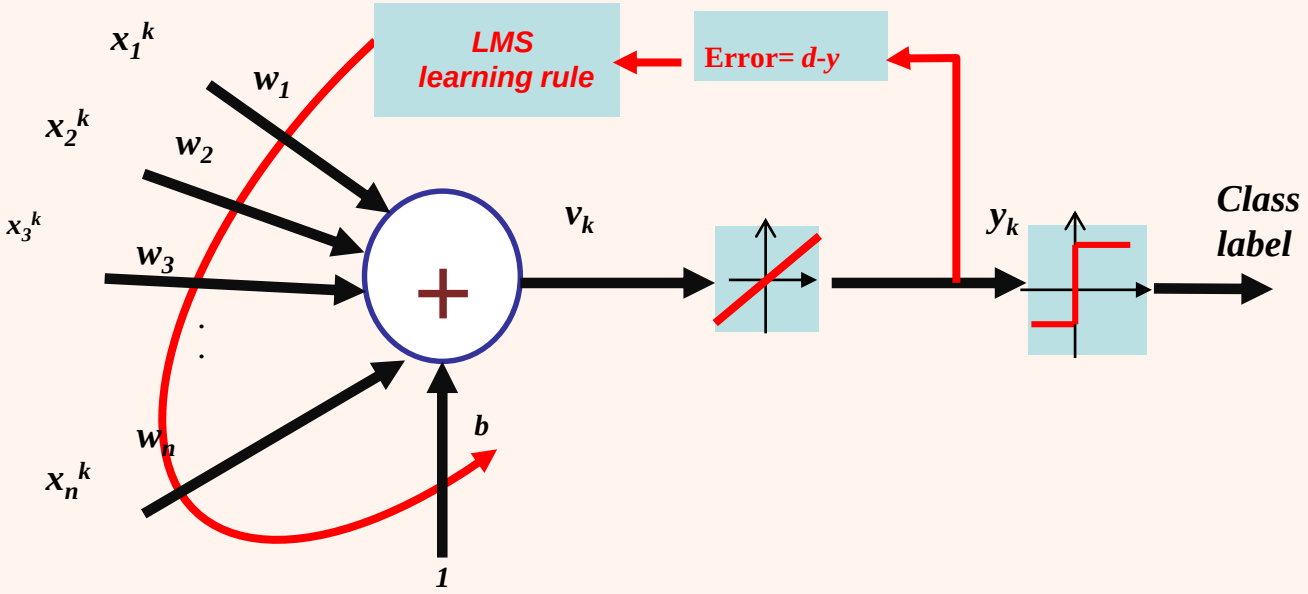
- الگوریتم LMS وزن‌ها و بایاس را به گونه‌ای تغییر می‌دهد که میانگین مربعات خطا (بین خروجی مطلوب و خروجی واقعی) سیستم را به حداقل برساند.

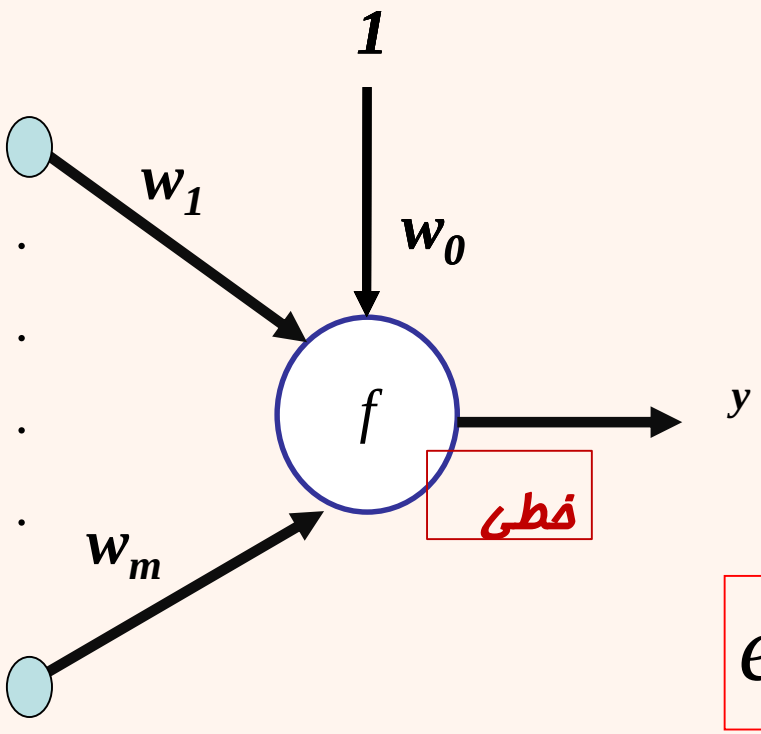


Percptron



Adaline





خطای به‌دست آمده به ازای ورودی X^k

$$e_k(n) = d^k - W(n)X^k$$

$$X^k = [1, x_1^k, \ldots\ldots, x_m^k]^T$$

$$\mathbf{X} = [X^1, X^2, \ldots\ldots, X^N]_{(m+1)\times N}$$

ورودی‌ها

N ورودی m تایی

$$D = [d^1, d^2, \ldots\ldots, d^N]_{1\times N}$$

خروجی‌های مطلوب

$$W = [w_0, w_1, \ldots\ldots, w_m]_{1\times(m+1)}$$

$$X^k = [1, x_1^k, ..., x_m^k]^T$$

$$\mathbf{X} = [X^1, X^2, ..., X^N]_{(m+1)\times N}$$

$$D = [d^1, d^2, ..., d^N]_{1\times N}$$

$$W = [w_0, w_1, ..., w_m]_{1\times(m+1)}$$

$$e_k(n) = d^k - W(n)X^k$$

Batch Mode

$$SSE = E(n) = \sum_{k=1}^{N} (d^k - W(n)X^k)^2$$

Number of epoch

$$E(n) = \| D - W(n)\mathbf{X} \|^2$$

Y(n)

$E(W(n))$ **پارامتر آزاد برای تابع خطا وزن‌ها هستند.**

$$Y(n) = [y^1(n), y^2(n), ..., y^N(n)]$$

کمینه کردن خطا

- باید به گونه‌ای عمل کرد که تابع خطا طی فرآیند آموزش کم‌تر شود:

$$E(n+1) < E(n) \quad \text{یا} \quad E(W(n+1)) < E(W(n))$$

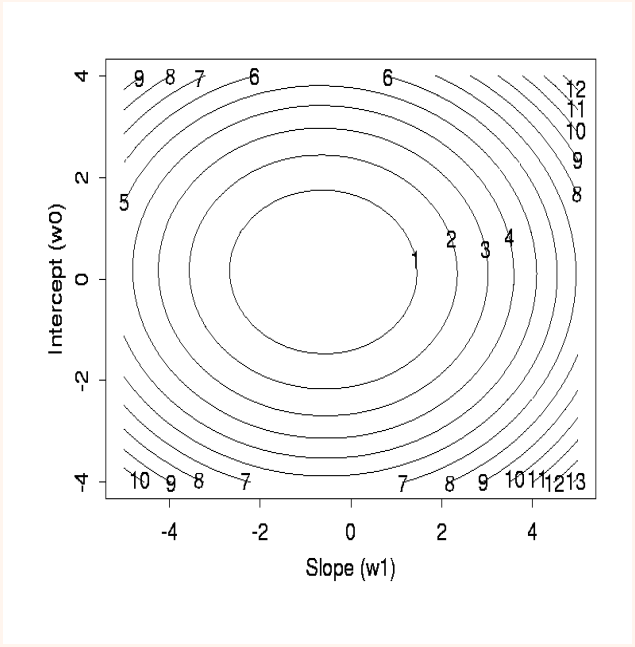
- هدف یافتن وزن بهینه‌ای است که به ازی آن تابع خطا(هزینه) مینیمم شود:

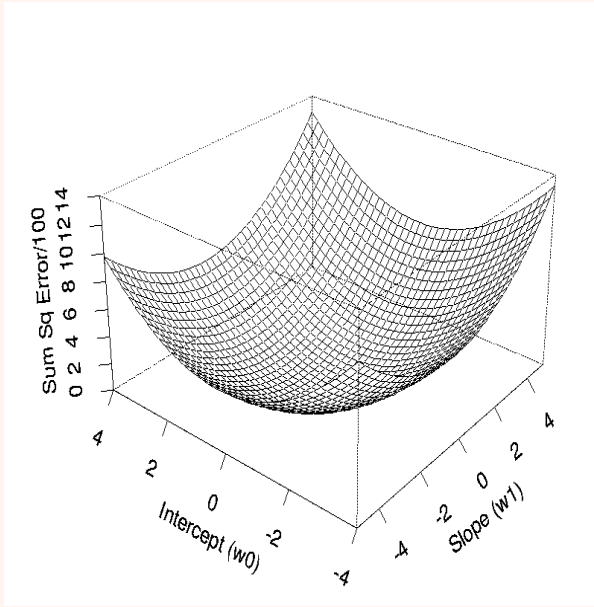
$$E(W^*) \leq E(W)$$

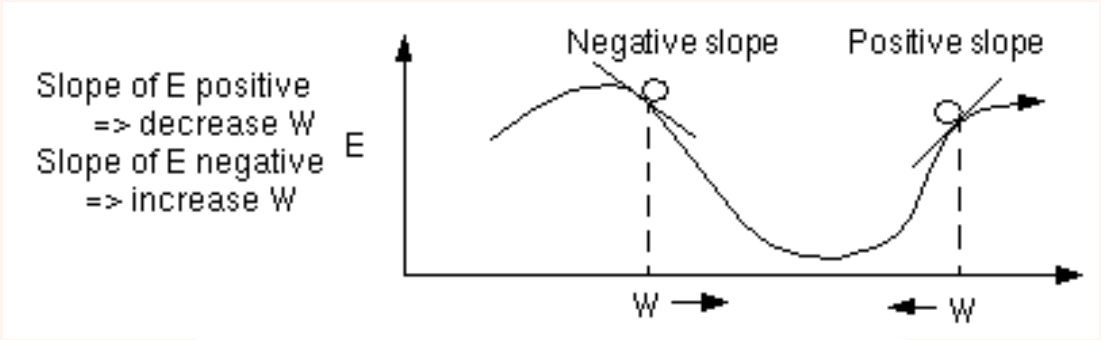
- شرط لازم برای وجود وزن بهینه این است که:

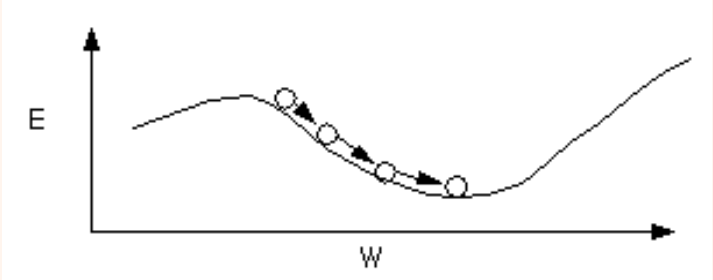
$$\nabla E(W^*) = 0$$

Steepest descent کمینه کردن خطا(ادامه...)









گرادیان

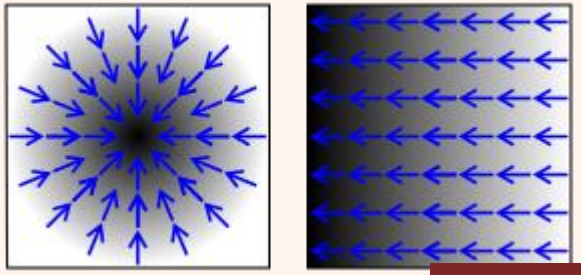
- گرادیان یک تابع اسکالر چند متغیره $f(\mathbf{x})$

$$f(\mathbf{x}):\mathbb{R}^n \to \mathbb{R}, \qquad \mathbf{x}=[x_1, x_2, \cdots, x_n]$$

به صورت زیر تعریف می‌شود:

$$\nabla f = \left[\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \cdots, \frac{\partial f}{\partial x_n}\right]$$

- گرادیان جهت و اندازه‌ی بیشترین تغییرات تابع f را نشان می‌دهد.



wikipedia

در این شکل نقاط تاریک‌تر نقاط با مقادیر بیشتر را نشان می‌دهند

- هدف به حداقل رساندن مقدار E یا S.S.E است:

$$\nabla_w E_{(n)} = [\frac{\partial E(n)}{\partial w_0(n)}, \frac{\partial E(n)}{\partial w_1(n)}, \ldots\ldots\ldots, \frac{\partial E(n)}{\partial w_m(n)}]$$

$$SSE = E(n) = \sum_{k=1}^{N} (d^k - W(n)X^k)^2$$

داشتیم

Batch Mode

$$\frac{\partial E(n)}{\partial w_i(n)} = -2\sum_{k=1}^{N} (d^k - y^k(n)) \frac{\partial y^k(n)}{\partial w_i(n)}$$

$$\frac{\partial E(n)}{\partial w_i(n)} = -2\sum_{k=1}^{N}(d_k - y_k(n))\frac{\partial y_k(n)}{\partial w_i(n)}$$

$$\frac{\partial E(n)}{\partial w_i(n)} = -2\sum_{k=1}^{N}(d_k - y_k(n))x_i^k$$

$$= -2(D - Y(n))[\mathbf{X}_i]^T \qquad \mathbf{X}_i = [x_i^1, x_i^2, ..., x_i^N]$$

- برای انتخاب w مطلوب

$$w_i(n+1) = w_i(n) - \eta\frac{\partial E(n)}{\partial w_i(n)}$$

learning rate

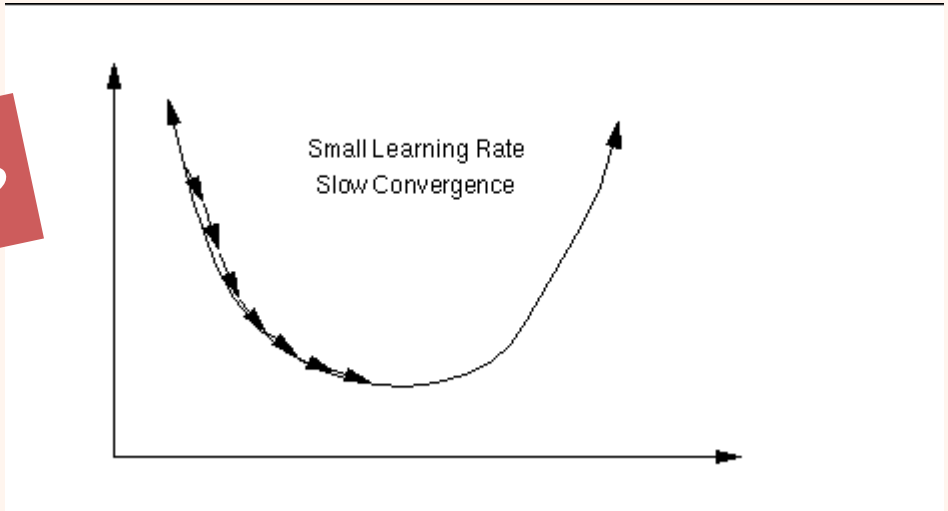
نرخ آموزش

$$w_i(n+1) = w_i(n) + 2\eta(D - y(n))[\mathbf{X}_i]^T$$

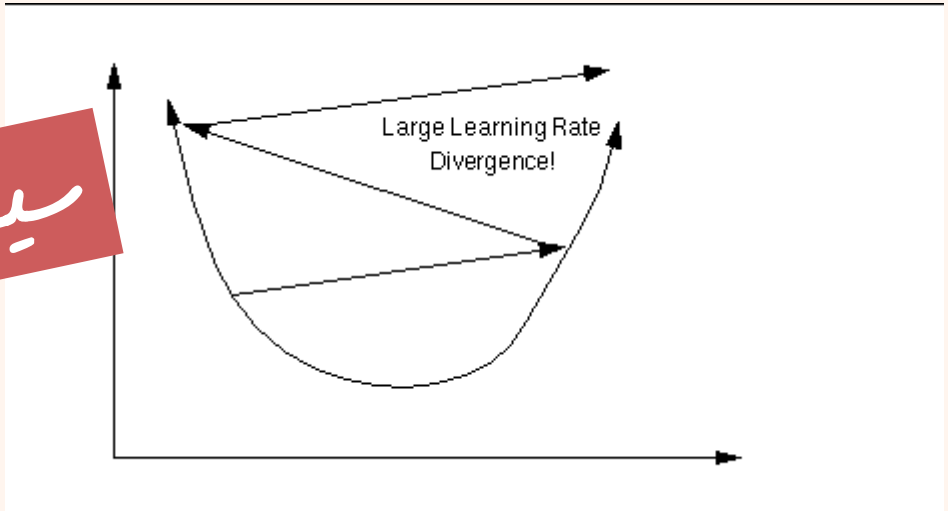
Adaline learning rule

تنظیم نرخ یادگیری

همگرایی کند است.



سیستم ناپایدار است.



به دست آوردن محدوده نرخ آموزش

- نرخ آموزش «پایداری» و «سرعت همگرایی» را مشخص می‌کند.

با فرض تک‌الگو

$$E_{(t+1)} = \left\| D - W_{(t+1)} X \right\|^2$$

$$E_{(t+1)} = \left\| D - \left[W_{(t)} + \eta \left(D - Y_{(t)} \right) X^T \right] X \right\|^2$$

$$E_{(t+1)} = \left\| D - W_{(t)} X - \eta \left(D - Y_{(t)} \right) \| X \|^2 \right\|^2$$

$$E_{(t+1)} = E_{(t)} + \eta^2 \left\| D - Y_{(t)} \right\|^2 \left(\| X \|^2 \right)^2 - 2\eta \left\| \left(D - Y_{(t)} \right) \right\|^2 \| X \|^2$$

به دست آوردن محدوده نرخ آموزش(ادامه...)

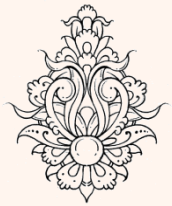
$$E_{(t+1)} = E_{(t)} + \eta^2 \left\| D - Y_{(t)} \right\|^2 \left(\| X \|^2 \right)^2 - 2\eta \left\| \left(D - Y_{(t)} \right) \right\|^2 \| X \|^2$$

$$E_{(t+1)} = E_{(t)} \left[1 + \eta^2 \left(\| X \|^2 \right)^2 - 2\eta \| X \|^2 \right]$$

$$E_{(t+1)} = E_{(t)} \left[1 - \eta \| X \|^2 \right]^2$$

$$\frac{E_{(t+1)}}{E_{(t)}} = \left[1 - \eta \| X \|^2 \right]^2 < 1$$

$$-1 < 1 - \eta \| X \|^2 < 1$$



به دست آوردن محدوده نرخ آموزش(ادامه...)

$$-1 < 1 - \eta \|X\|^2 < 1$$

$$0 < \eta \|X\|^2 < 2$$

$$0 < \eta < \frac{2}{\|X\|^2}$$

$$0 < \eta < \frac{2}{\max_k \|X^k\|^2}$$

اگر N الگو

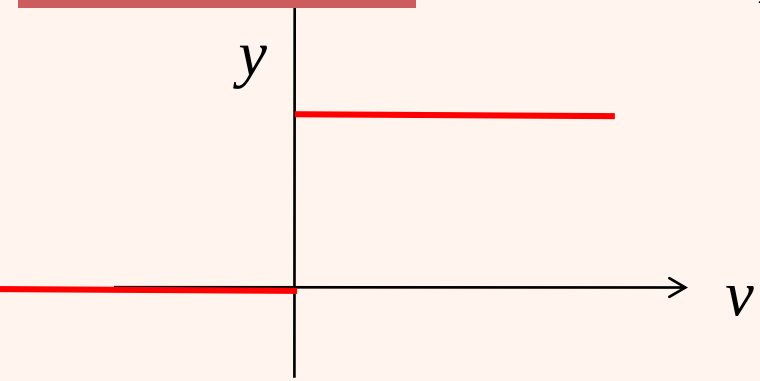
$$\frac{0.1}{\max_k \|X^k\|^2} < \eta < \frac{2}{\max_k \|X^k\|^2}$$

به صورت تجربی

Activation function

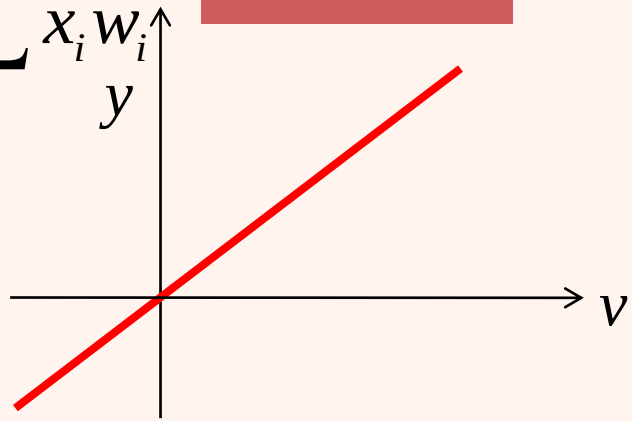
تابع فعال‌سازی

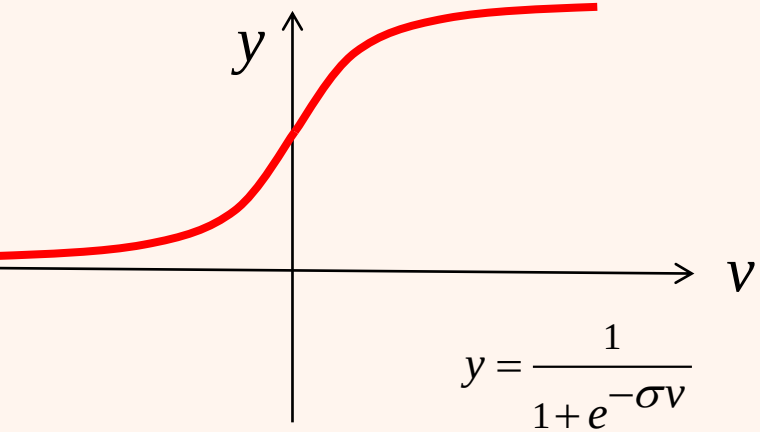
Binary Step



Pure line

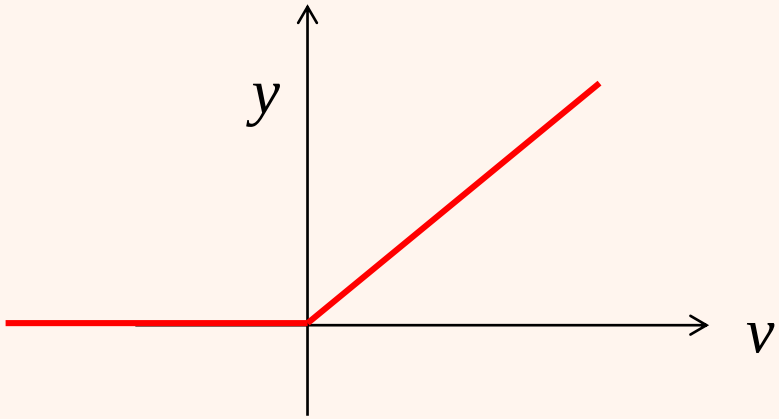
$$v = b + \sum_i x_i w_i$$





$$y = \frac{1}{1+e^{-\sigma v}}$$

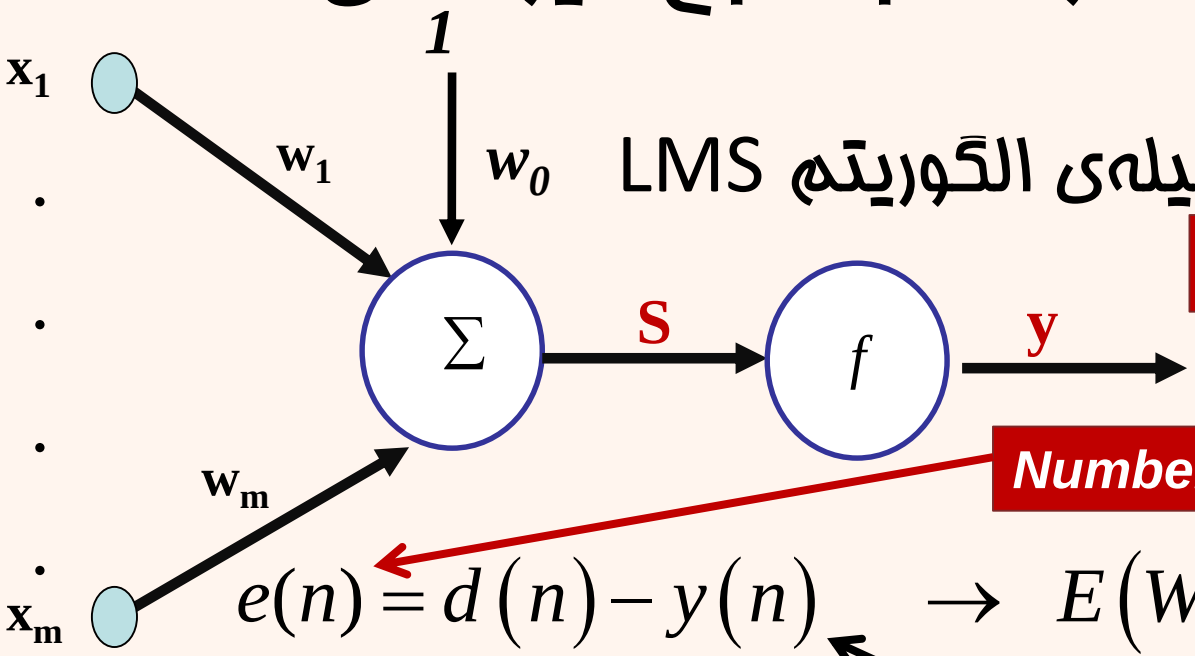
Sigmoid



REctified Linear Unit (ReLU)

یادگیری عمیق

تک‌لایه تک‌واحد با تابع غیر خطی



- حل به وسیله‌ی الگوریتم LMS

Sequential Mode

Number of iteration

$$e(n)=d(n)-y(n) \quad \rightarrow \quad E(W(n))=\frac{1}{2}e^2(n)$$

خروجی به ازای ورودی در تکرار nام

$$w_k(n+1)=w_k(n)-\eta\frac{\partial E}{\partial w_k}$$

$$\frac{\partial E}{\partial w_k}=\frac{1}{2}\frac{\partial e^2}{\partial w_k}=e\frac{\partial e}{\partial w_k}=e\frac{\partial e}{\partial y}\cdot\frac{\partial y}{\partial w_k}$$

$$=e\frac{\partial e}{\partial y}\cdot\frac{\partial y}{\partial s}\cdot\frac{\partial s}{\partial w_k}$$

$$\frac{\partial E}{\partial w_k} = e\frac{\partial e}{\partial y}\frac{\partial y}{\partial s}\frac{\partial s}{\partial w_k}$$

-1

$$y = f(s) \rightarrow \frac{\partial y}{\partial s} = f'(s) \qquad x_k$$

$$= -ef'(s)x_k$$

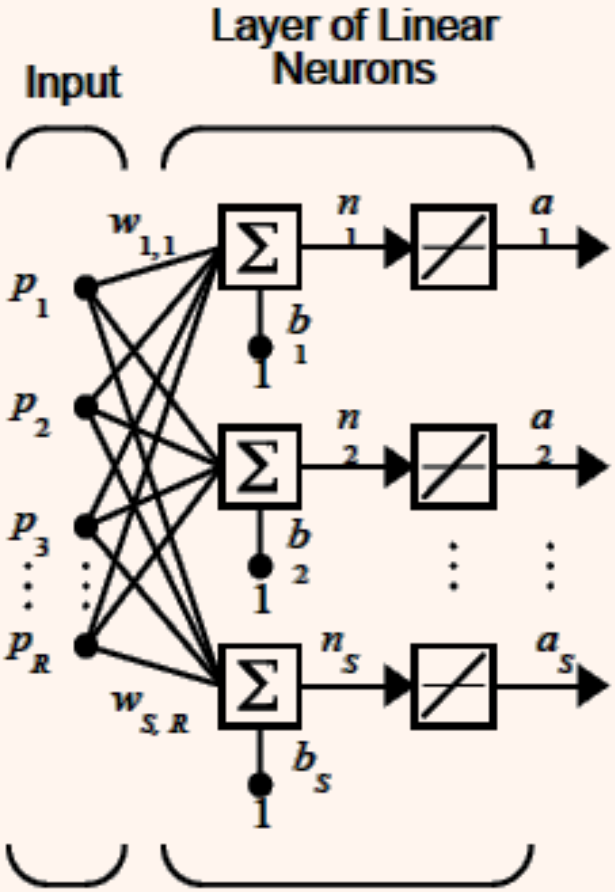
تابع انگیزش باید مشتق پذیر باشد

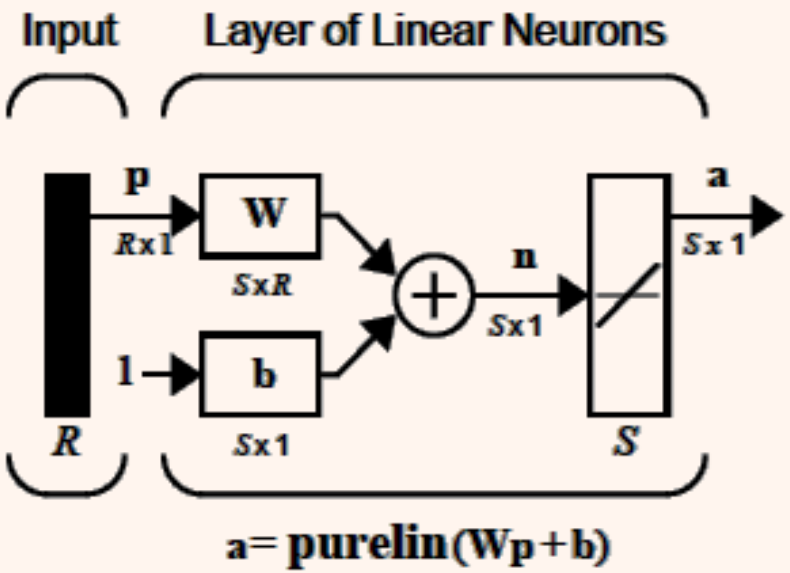
$$w_k(n+1) = w_k(n) + (d - y(n))\, x_k$$

بسته به تابع f متفاوت است

شبکه‌ی تک‌لایه با چند خروجی

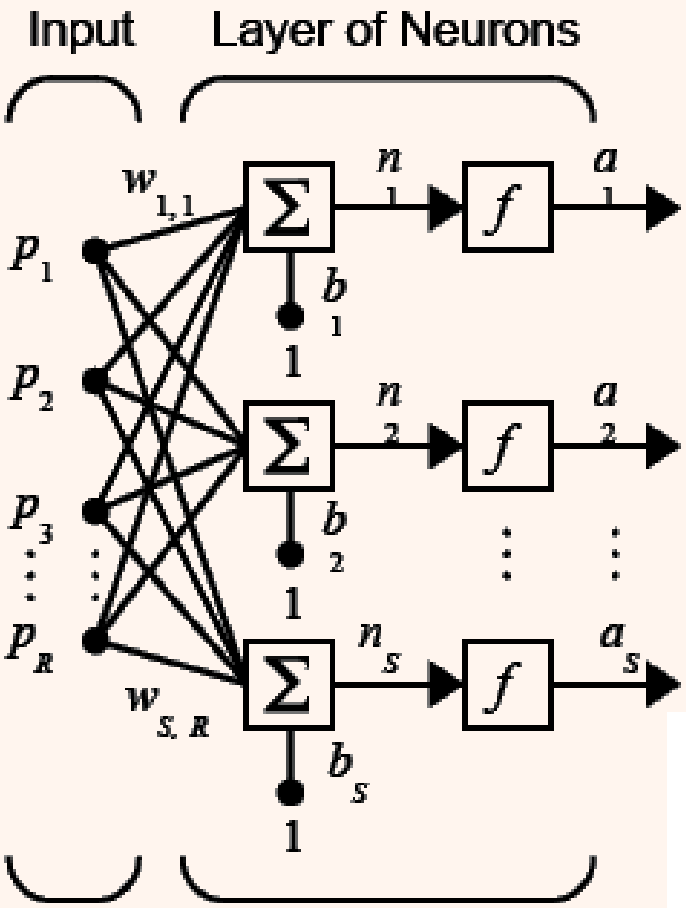
Single-Layer Linear Network





Where... R = numberof elements in input vector

S = numberof neurons in layer



R تعداد المان های ورودی

S تعداد نورون های موجود در یک لایه

$$\mathbf{a} = \mathbf{f}(\mathbf{W}\mathbf{p} + \mathbf{b})$$

$$\mathbf{W} = \begin{bmatrix} w_{1,1} & w_{1,2} & \cdots & w_{1,R} \\ w_{2,1} & w_{2,2} & \cdots & w_{2,R} \\ & & & \\ w_{S,1} & w_{S,2} & \cdots & w_{S,R} \end{bmatrix}$$

MATLAB

مثال

- می‌خواهیم پنج داده‌ی زیر را که خروجی‌های مطلوب آن‌ها نیز مشخص است را در دو کلاس طبقه‌بندی کنیم:

```
P1=[0.7,0.2];
T1=[1];
```

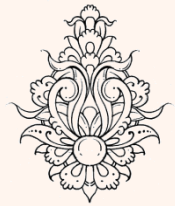
```
P2=[-0.1,0.9];
T2=[1];
```

```
P3=[-0.3,0.3];
T3=[0];
```

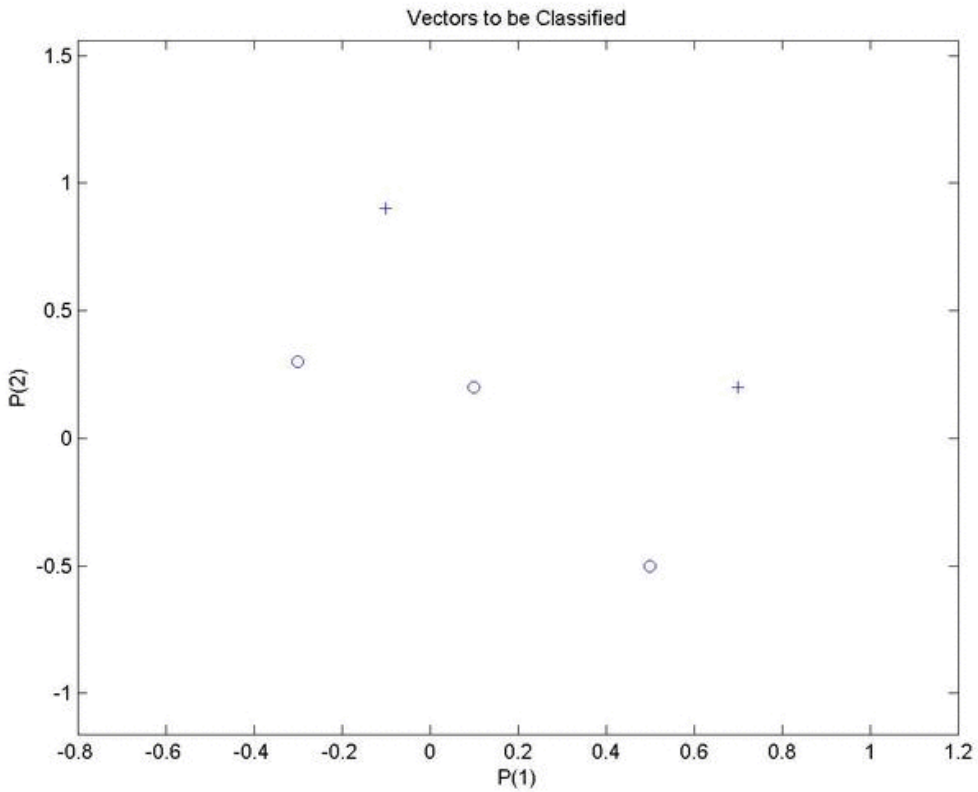
```
P4=[0.1,0.2];
T4=[0];
```

```
P5=[0.5,-0.5];
T5=[0];
```

```
P=[0.7 -0.1 -0.3 0.1  0.5;
   0.2  0.9  0.3 0.2 -0.5];
T=[1 1 0 0 0];
```



```
P=[0.7 -0.1 -0.3 0.1 0.5;
   0.2 0.9 0.3 0.2 -0.5];
T=[1 1 0 0 0];
W=[0 0];
b=-1;
plotpv(P,T);
plotpc(W,b);
nepoc=0
Y=hardlim(W*P+b);
while any(Y~=T)
    Y=hardlim(W*P+b);
    E=T-Y;
    dW=E*P';
    db=sum(E);
    W=W+dW;
    b=b+db;
    nepoc=nepoc+1;
    disp('epochs='),disp(nepoc),
    disp(W), disp(b);
    plotpv(P,T);
    plotpc(W,b);
end
```

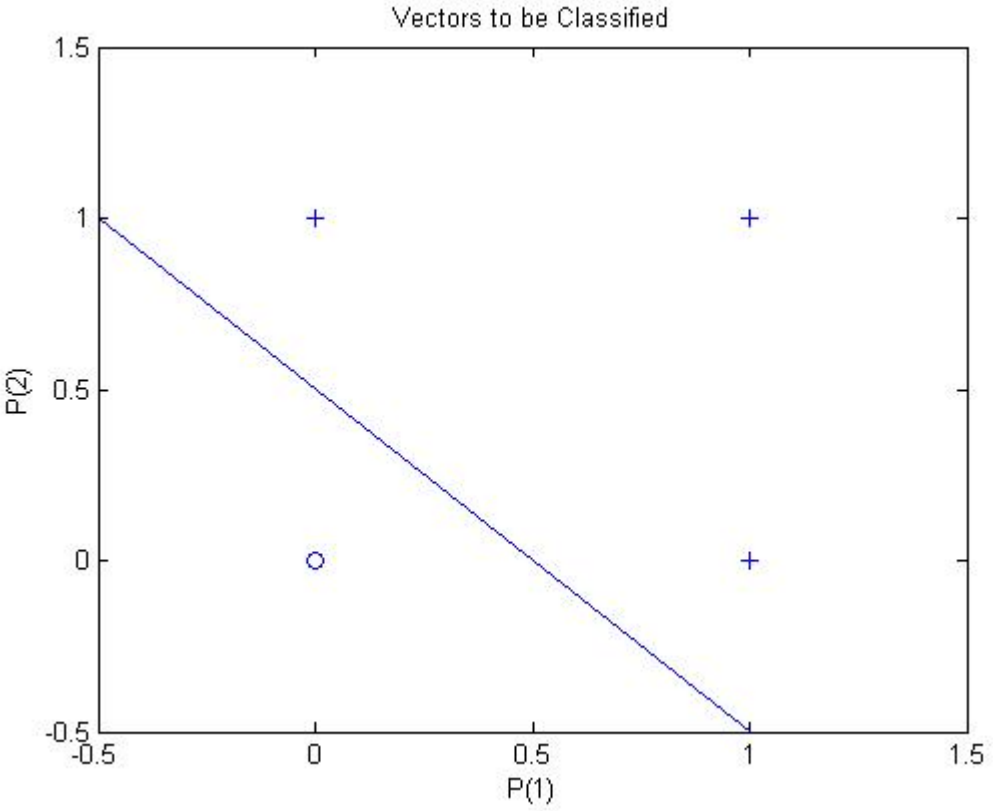



```
[dW,db]= learnp(P,E);
```

Epoch=9

W1=2.7 W2=2.9
B=-2

OR

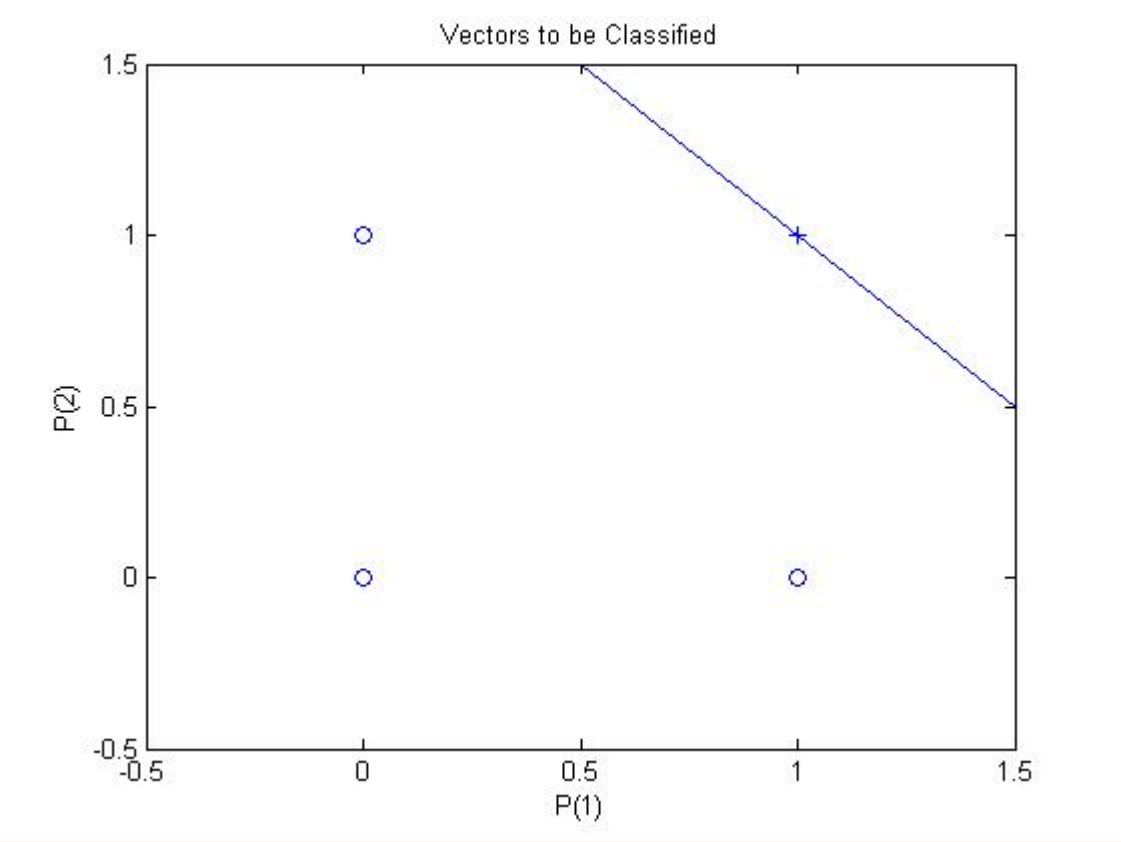


Epochs= 5

W1=2 W2= 2

b= -1

AND

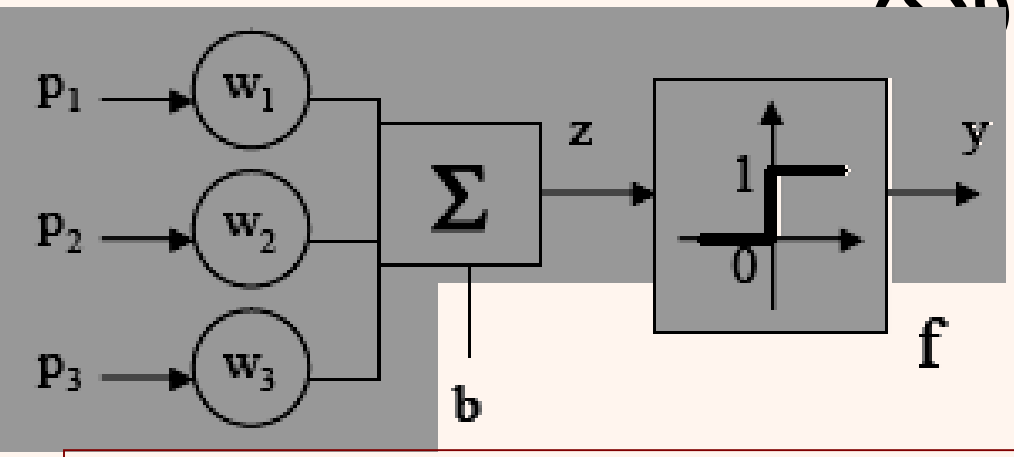


epochs= 4

W1=1 w2= 1

b= -2

پرسپترون سه ورودی

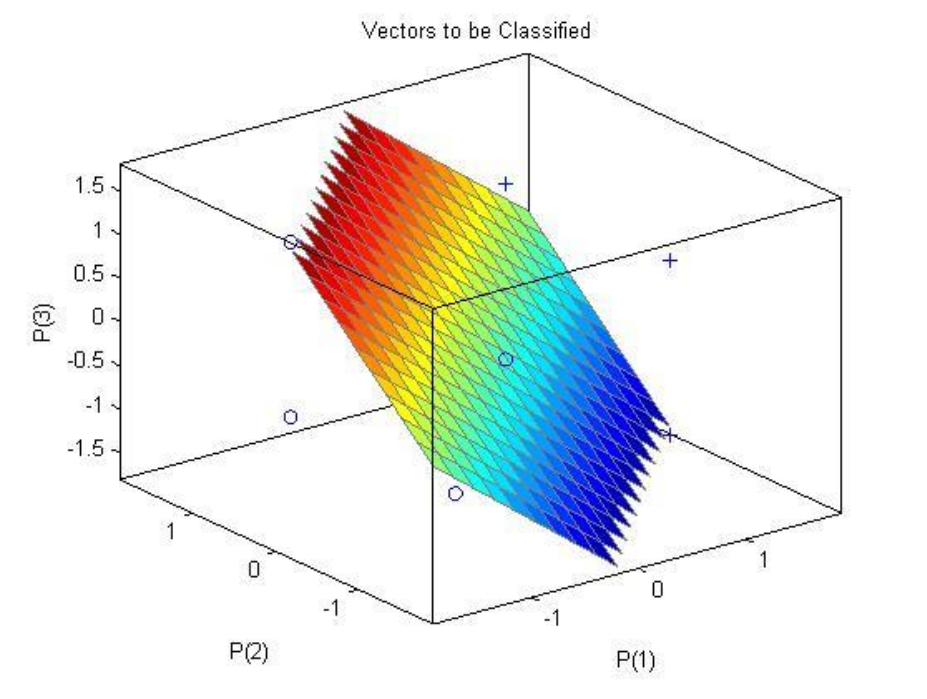


$$y = \text{hardlim}(z) = \text{hardlim}([w_1, w_2, w_3].[p_1, p_2, p_3]^T + b)$$

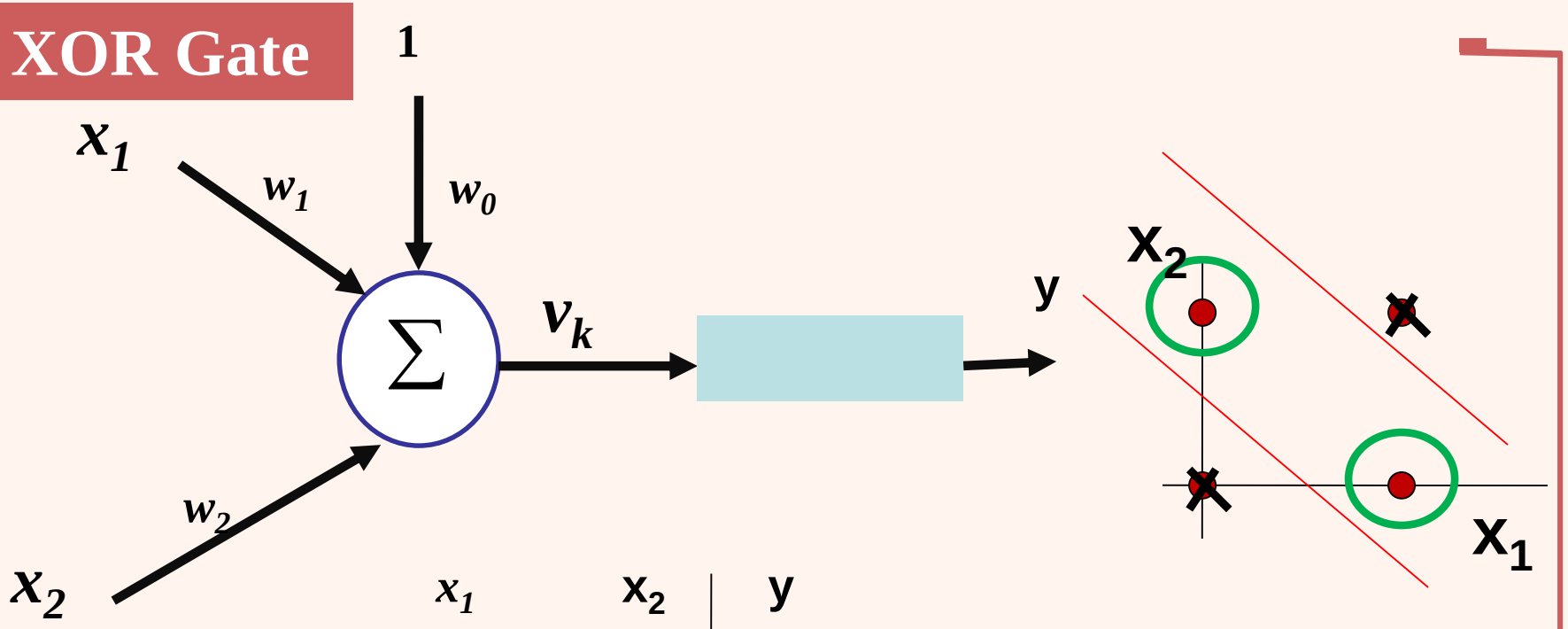
epochs=
3

w1= 3 w2= -3 w3= 3

b=0



XOR Gate



x_1	x_2	y
1	1	0
0	0	0
1	0	1
0	1	1

$W_1+ W_2+W_0<0$

$W_0<0$

$W_1+ W_0>0$

$W2+W_0>0$

یادگیری عمیق

به وسیله‌ی یک خط جدایی‌پذیر نیستند پس قابلیت جداسازی چنین تابعی را ندارد