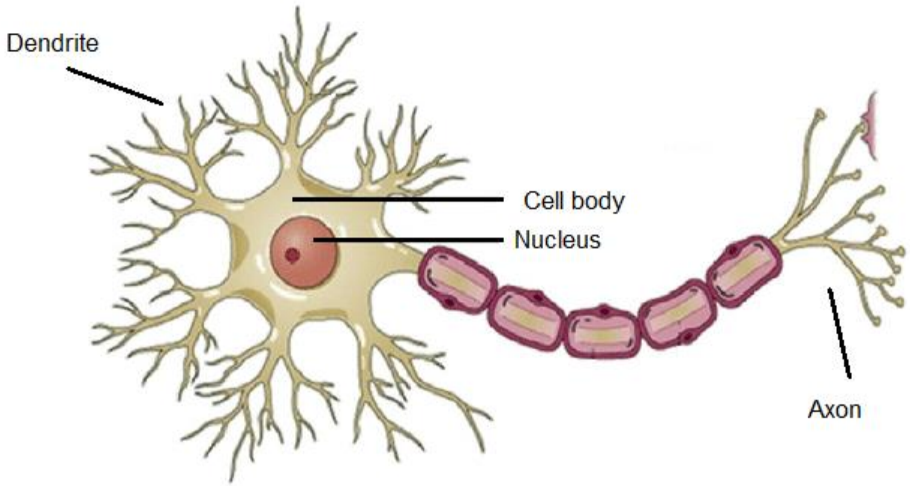
فصل سوم
شناسایی الگو
طبقه‌بندی کننده‌های خطی
Linear Classifiers

محمدجواد فدائی‌اسلام

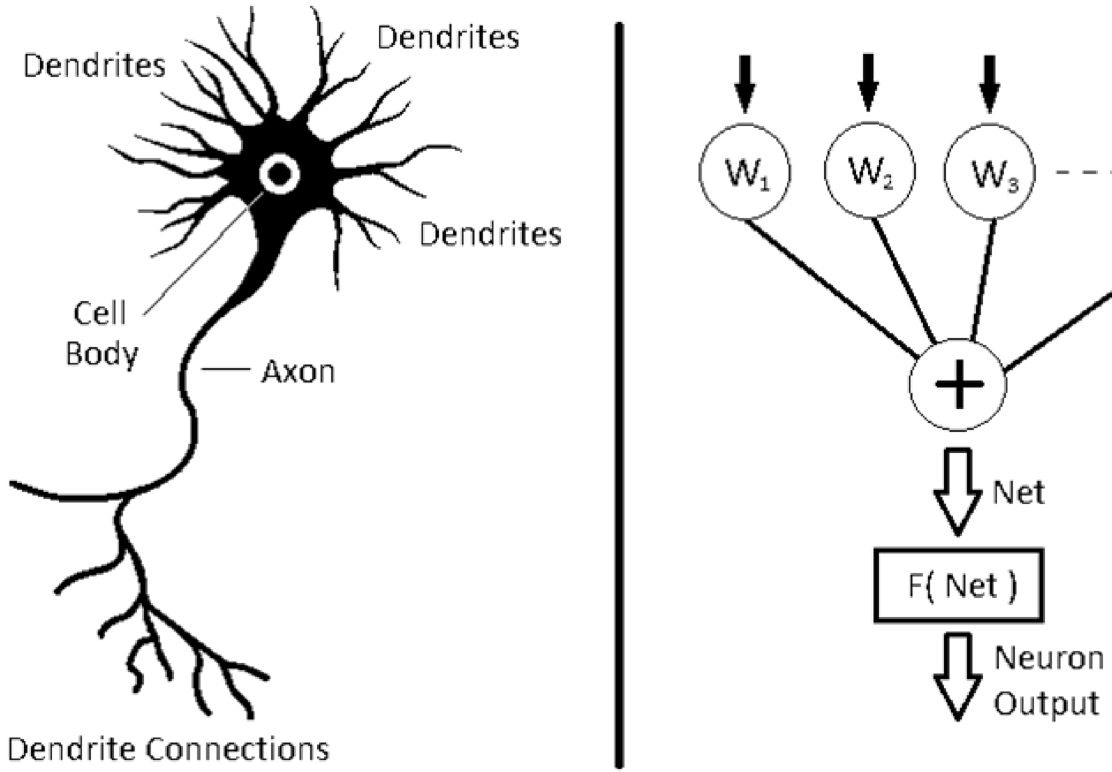
Simple Biological Neuron

- نورون دارای دندریت‌هایی است که سیگنال‌های نورون‌های دیگر را دریافت می‌کنند.
- بدنه سلول، فعال‌سازی را کنترل می‌کند.
- آکسون یک پالس الکتریکی را به دندریت نورون‌های دیگر حمل می‌کند.



ARTIFICIAL NEURAL NETWORKS

- نورون مصنوعی دارای یک سری ورودی وزن‌دار و یک جمع‌کننده است (سمت راست).
- یک تابع فعال‌سازی (آتش)، که در صورت گذر جمع مقدار ورودی از آستانه، سیگنال را به نرون بعدی ارسال می‌کند.



F(Net) =
firing mechanism

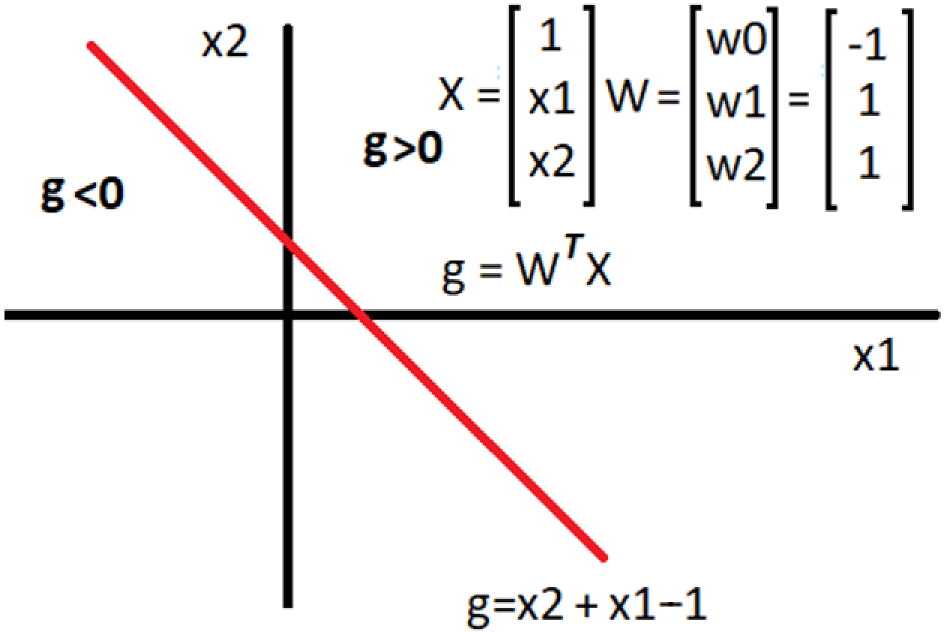
LINEAR DISCRIMINANT FUNCTIONS AND DECISION HYPERPLANES

- اگر جداسازی داده‌ها در یک مساله بخواهد با استفاده از یک جداکننده خطی انجام شود. مرز تصمیم یک ابرصفحه به صورت زیر خواهد بود:

$$g(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x} + \boldsymbol{w}_0 = 0$$

در رابطه بالا $\boldsymbol{w} = [w_1, w_2, w_3 \dots w_l]^T$ بردار وزن و w_0 عرض از مبدا یا یک آستانه است.

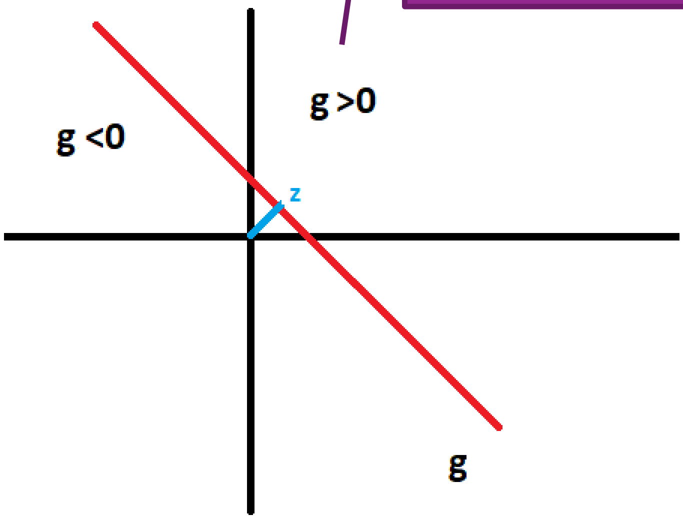
معادله خط


x2
g>0
g<0
$X = \begin{bmatrix} 1 \\ x1 \\ x2 \end{bmatrix} \quad W = \begin{bmatrix} w0 \\ w1 \\ w2 \end{bmatrix} = \begin{bmatrix} -1 \\ 1 \\ 1 \end{bmatrix}$
$g = W^T X$
x1
g=x2 + x1−1

فاصله از خط راست

با فرض مثبت بودن ضریب y

$$= \frac{|g(\boldsymbol{x})|}{\sqrt{w_1^2 + w_2^2}}$$



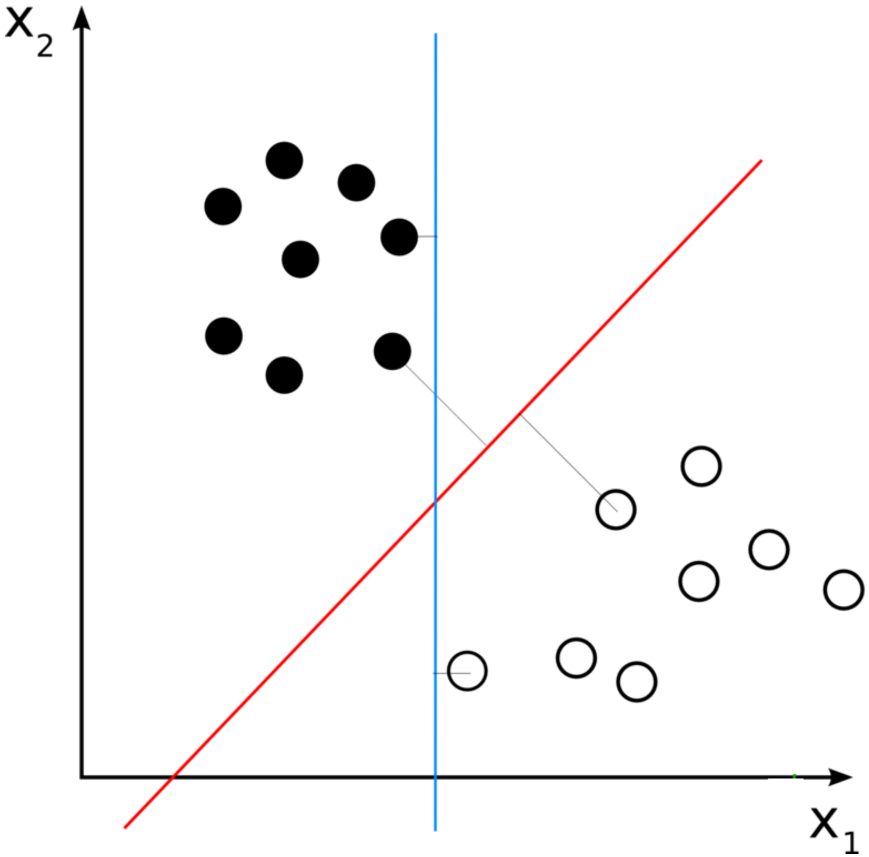
معادله خطوط مختلف برای یک خط

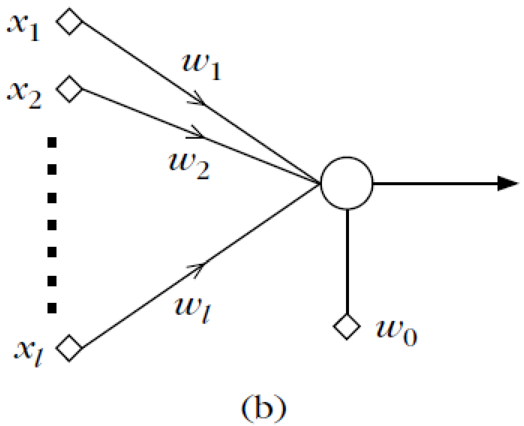
$$g = y + x - 1$$
$$g = 2y + 2x - 2$$
$$g = -y - x + 1$$

$$g([0,0]) = 0 + 0 - 1 = -1 < 0$$

$$z = \frac{|-1|}{\sqrt{2}} = \frac{\sqrt{2}}{2}$$

طبقه‌بند خطی، مدل پرسپترون



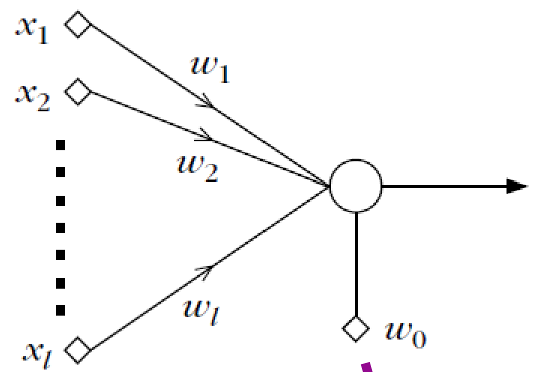


The basic perceptron model

x_i ویژگی است.
خط قرمز و آبی جداکننده داده‌های دو کلاس هستند.
فاصله نمونه‌ها از خط قرمز بیشتر است.

تابع جداکننده خطی و ابرصفحه تصمیم‌گیری

$$g = x_1 w_1 + x_2 w_2 + x_3 w_3 + \cdots + w_0$$



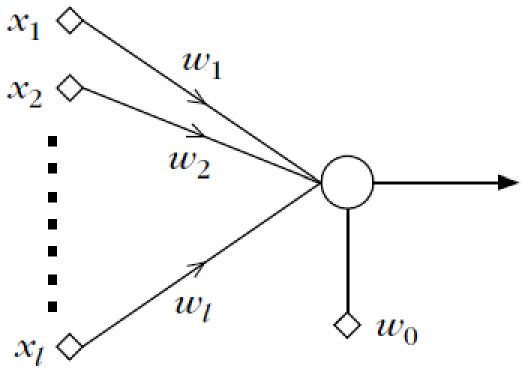
عرض از مبدا

$$\boldsymbol{X} = \begin{bmatrix} 1 \\ x_1 \\ x_2 \\ \vdots \\ x_{l-1} \\ x_l \end{bmatrix}, \boldsymbol{W} = \begin{bmatrix} w_0 \\ w_1 \\ w_2 \\ \vdots \\ w_{l-1} \\ w_l \end{bmatrix}$$

$$g = \boldsymbol{W}^T \boldsymbol{X}$$

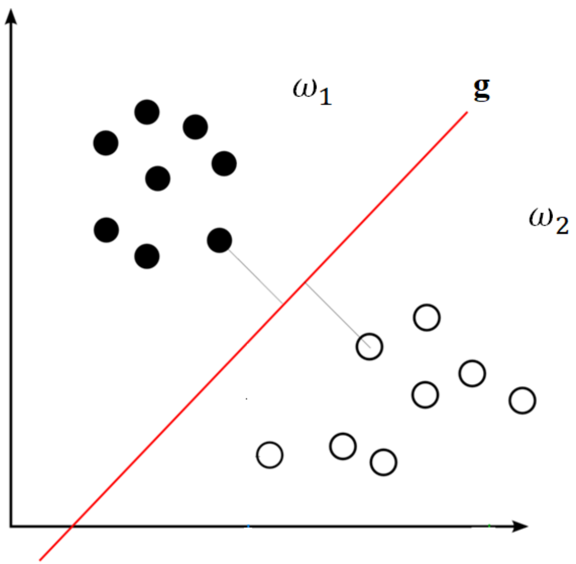
T = transpose=ترانهاده

آموزش پرسپترون



- هدف از آموزش پرسپترون یافتن وزن‌ها (معادله خط) است به صورتی که داده‌ها درست کلاس‌بندی شوند.

$$g = x_1 w_1 + x_2 w_2 + x_3 w_3 + \cdots + w_0$$



$$\boldsymbol{W} = \begin{bmatrix} w_0 \\ w_1 \\ w_2 \\ \vdots \\ w_{l-1} \\ w_l \end{bmatrix} = ?$$

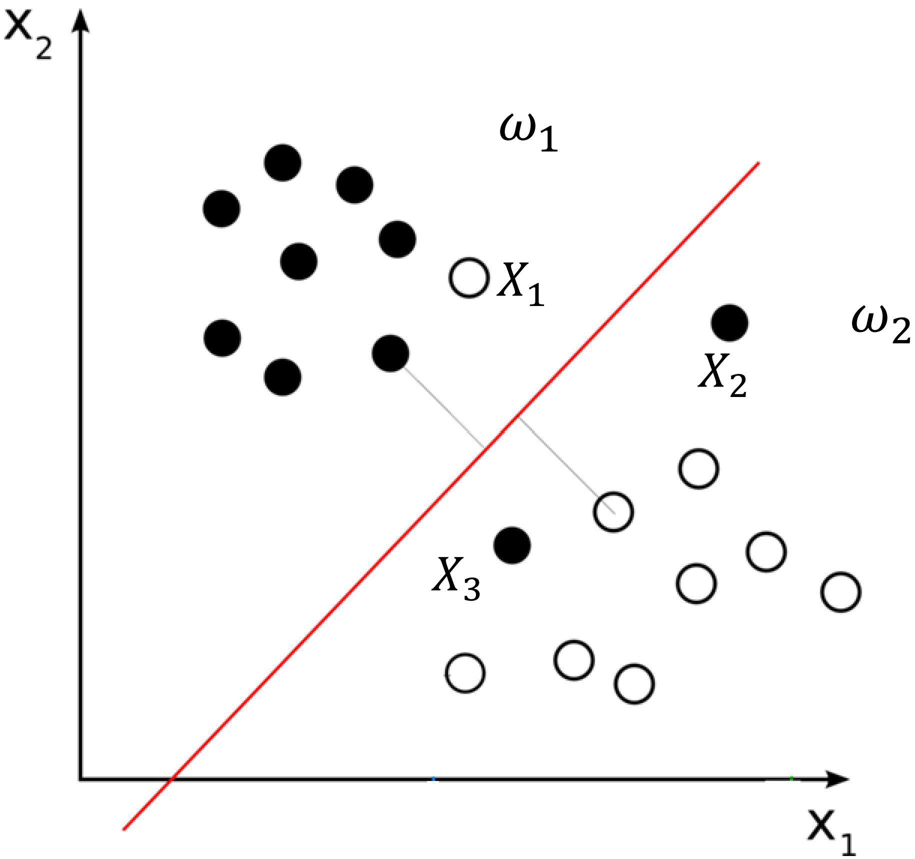
$$g = \boldsymbol{W}^T \boldsymbol{X}$$

THE PERCEPTRON COST FUNCTION

$$J(W) = \sum_{x \in Y} (\delta_x W^T x)$$

- $Y = the\ set\ of\ misclassified\ points$
- $\delta_x = \begin{cases} -1, x \in \omega_1 \\ +1, x \in \omega_2 \end{cases}$
- $J \geq 0$
- $if\ J = 0, solution\ has\ been\ obtained$
- $for\ each\ misclassified\ point: \delta_x W^T x > 0$

الگوریتم آموزش پرسپترون (دسته‌ای)



$\delta_x = -1, if\ x \in \omega_1$ (Black circle)
$\delta_x = +1, if\ x \in \omega_2$ (white circle)

$Y = \{X_1, X_2, X_3\}$

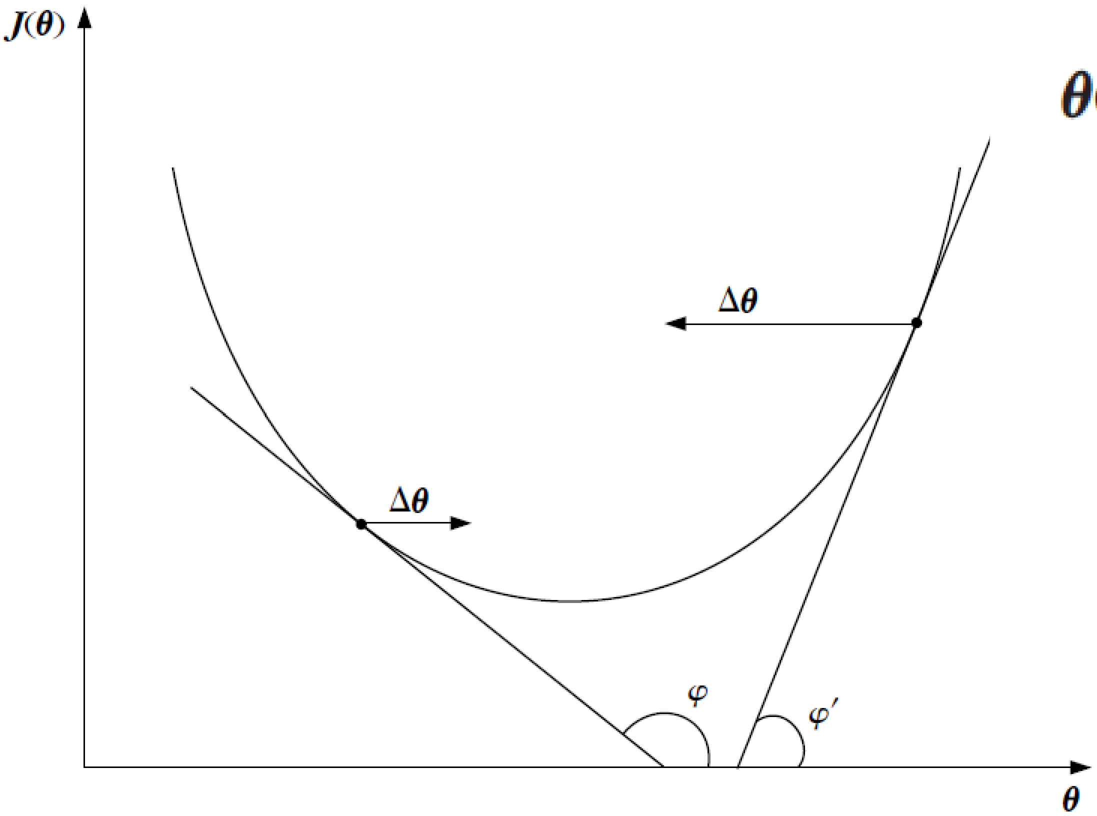
Gradient descent method to minimize the cost function

$$\boldsymbol{w}(t+1) = \boldsymbol{w}(t) - \rho_t \frac{\partial J(\boldsymbol{w})}{\partial \boldsymbol{w}}\bigg|_{w=w(t)}$$

$$\frac{\partial J(\boldsymbol{w})}{\partial \boldsymbol{w}} = \sum_{\boldsymbol{x} \in Y} \delta_x \boldsymbol{x}$$

$$\boldsymbol{w}(t+1) = \boldsymbol{w}(t) - \rho_t \sum_{\boldsymbol{x} \in Y} \delta_x \boldsymbol{x}$$

THE GRADIENT DESCENT METHOD



$$\theta(new) = \theta(old) + \Delta\theta$$

$$\Delta\theta = -\mu \frac{\partial J(\theta)}{\partial \theta}\bigg|_{\theta=\theta(old)}$$

حالت دسته‌ای

BATCH MODE

The Perceptron Algorithm

$$\rho_0 = 1$$

$$\rho_t = \frac{1}{t}$$

به مرور زمان ρ کاهش می یابد.

- Choose $\boldsymbol{w}(0)$ randomly
- Choose ρ_0
- $t = 0$
- Repeat
  - $Y = \emptyset$

  مجموعه نقاط با کلاس بندی نادرست
  - For $i = 1$ to N
    - If $\delta_{x_i} \boldsymbol{w}(t)^T \boldsymbol{x}_i \geq 0$ then $Y = Y \cup \{\boldsymbol{x}_i\}$
  - End {For}

  حلقه برای یافتن مجموعه نقاط با کلاس بندی نادرست
  - $\boldsymbol{w}(t+1) = \boldsymbol{w}(t) - \rho_t \sum_{x \in Y} \delta_x \boldsymbol{x}$

  به روز رسانی وزن‌ها
  - $t = t + 1$
- Until $Y = \emptyset$

EXAMPLE

$\boldsymbol{w}(0): x_1 + x_2 - 0.5 = 0 \qquad \rho = 0.7$

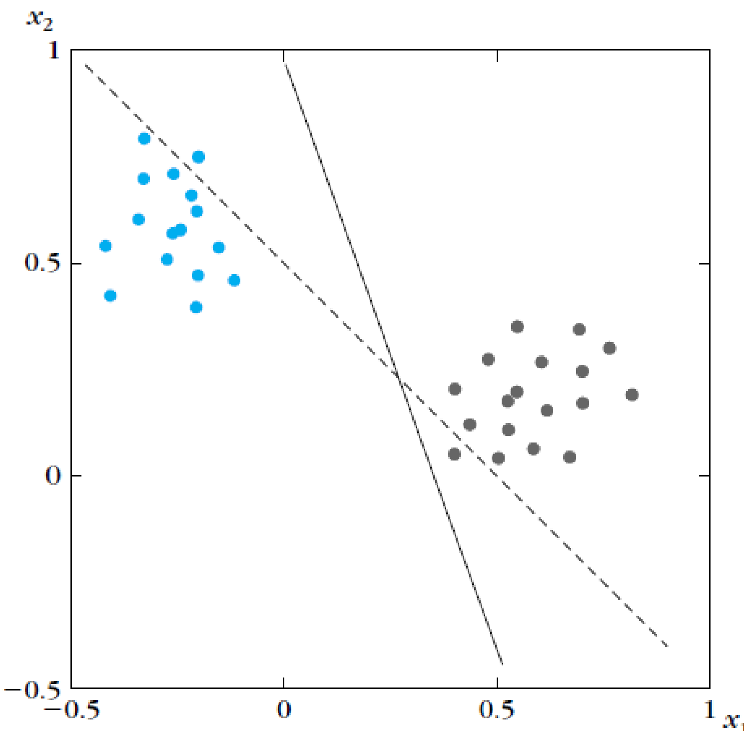
The line classifies correctly
all the vectors except $[0.4, 0.05]^T$ and $[-0.20, 0.75]^T$.
the next weight vector will be

$$\boldsymbol{w}(t+1) = \begin{bmatrix} 1 \\ 1 \\ -0.5 \end{bmatrix} - 0.7(-1)\begin{bmatrix} 0.4 \\ 0.05 \\ 1 \end{bmatrix} - 0.7(+1)\begin{bmatrix} -0.2 \\ 0.75 \\ 1 \end{bmatrix}$$

or

$$\boldsymbol{w}(t+1) = \begin{bmatrix} 1.42 \\ 0.51 \\ -0.5 \end{bmatrix}$$

The resulting new (solid) line $1.42x_1 + 0.51x_2 - 0.5 = 0$
classifies all vectors correctly, and
the algorithm is terminated.



الگوریتم آموزش پرسپترون (دسته‌ای)

CONVERGENCE OF PERCEPTRON ALGORITHM

- الگوریتم پرسپترون ممکن است جواب‌های متعددی داشته باشد.
- اگر ρ دارای شرایط زیر باشد حتما الگوریتم پرسپترون همگرا می‌شود.(خط جدا کننده را می‌یابد)

ρ_t must satisfy the following two conditions:

$$\lim_{t\to\infty}\sum_{k=0}^{t}\rho_k = \infty$$

$$\lim_{t\to\infty}\sum_{k=0}^{t}\rho_k^2 < \infty$$

مثال

$$\rho_k = \frac{1}{k}$$

$$\sum_{k=1}^{\infty}\frac{1}{k} = +\infty$$

$$\sum_{k=1}^{\infty}\frac{1}{k^2} = \frac{\pi^2}{6}$$

Rosenblatt algorithm, sample mode, pattern mode:

$$\boldsymbol{w}(t+1) = \boldsymbol{w}(t) + \rho \boldsymbol{x}_{(t)} \quad \text{if } \boldsymbol{x}_{(t)} \in \omega_1 \text{ and } \boldsymbol{w}^T(t)\boldsymbol{x}_{(t)} \leq 0$$

$$\boldsymbol{w}(t+1) = \boldsymbol{w}(t) - \rho \boldsymbol{x}_{(t)} \quad \text{if } \boldsymbol{x}_{(t)} \in \omega_2 \text{ and } \boldsymbol{w}^T(t)\boldsymbol{x}_{(t)} \geq 0$$

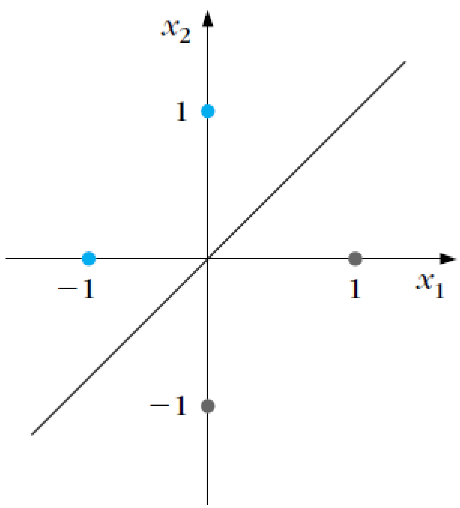
$$\boldsymbol{w}(t+1) = \boldsymbol{w}(t) \quad \text{otherwise}$$

الگوریتم رزنبلات - مثال

Points $(-1,0),(0,1)$ belong to class ω_1,
and points $(0,-1),(1,0)$ belong to class ω_2.
The parameter ρ is set equal to one, and the initial weight vector is chosen as $\boldsymbol{w}(0)=[0,0,0]^T$ in the extended three-dimensional space.

Step 1. $$\boldsymbol{w}^T(0)\begin{bmatrix}-1\\0\\1\end{bmatrix}=0,\quad \boldsymbol{w}(1)=\boldsymbol{w}(0)+\begin{bmatrix}-1\\0\\1\end{bmatrix}=\begin{bmatrix}-1\\0\\1\end{bmatrix}$$

Step 2. $$\boldsymbol{w}^T(1)\begin{bmatrix}0\\1\\1\end{bmatrix}=1>0,\quad \boldsymbol{w}(2)=\boldsymbol{w}(1)$$



Example 2 (cont)

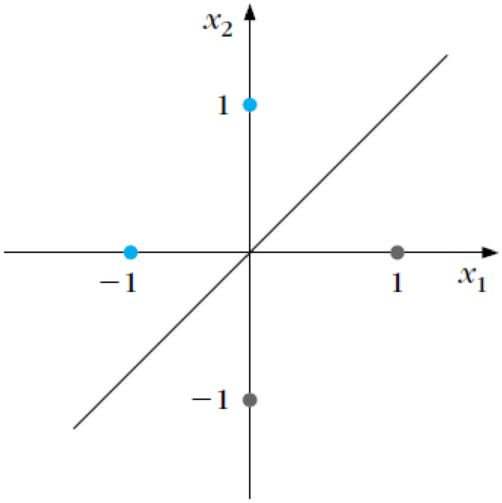
Step 3. $\boldsymbol{w}^T(2)\begin{bmatrix} 0 \\ -1 \\ 1 \end{bmatrix} = 1 > 0, \quad \boldsymbol{w}(3) = \boldsymbol{w}(2) - \begin{bmatrix} 0 \\ -1 \\ 1 \end{bmatrix} = \begin{bmatrix} -1 \\ 1 \\ 0 \end{bmatrix}$

Step 4. $\boldsymbol{w}^T(3)\begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} = -1 < 0, \quad \boldsymbol{w}(4) = \boldsymbol{w}(3)$

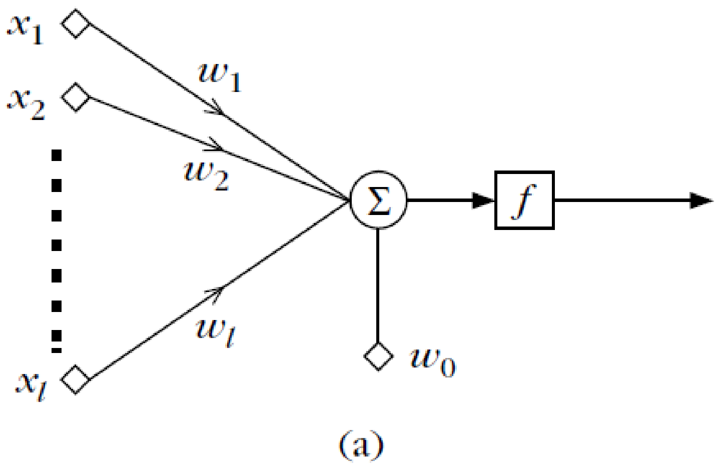
Step 5. $\boldsymbol{w}^T(4)\begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix} = 1 > 0, \quad \boldsymbol{w}(5) = \boldsymbol{w}(4)$

Step 6. $\boldsymbol{w}^T(5)\begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} = 1 > 0, \quad \boldsymbol{w}(6) = \boldsymbol{w}(5)$

Step 7. $\boldsymbol{w}^T(6)\begin{bmatrix} 0 \\ -1 \\ 1 \end{bmatrix} = -1 < 0, \quad \boldsymbol{w}(7) = \boldsymbol{w}(6)$



THE BASIC PERCEPTRON MODEL



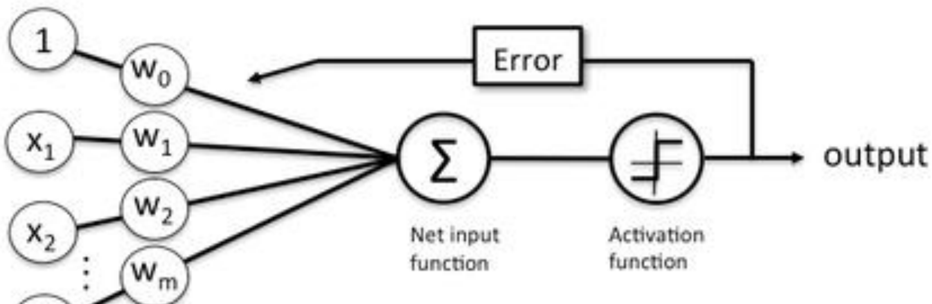
$f(\cdot)$ is the step function

$f(x) = -1$ if $x < 0$

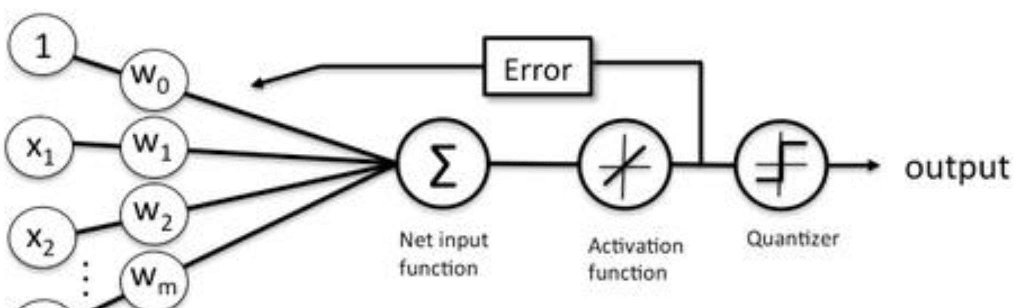
$f(x) = 1$ if $x > 0$

The basic perceptron model.
A linear combiner is followed by the activation function.

PERCEPTRON VS ADALINE



Perceptron rule.



Adaline.

The Perceptron uses the class labels to learn model coefficients

Adaline uses continuous predicted values to learn the model coefficients, which is more “powerful” since it tells us by “how much” we were right or wrong

ADALINE (ADAPTIVE LINEAR ELEMENT)

The weight vector will be computed so as to minimize the mean square error (MSE) between the desired and true outputs:

$$J(\boldsymbol{w}) = E[|y - \boldsymbol{x}^T \boldsymbol{w}|^2]$$

$$\frac{\partial J(\boldsymbol{w})}{\partial \boldsymbol{w}} = 2E[\boldsymbol{x}(y - \boldsymbol{x}^T \boldsymbol{w})] = \mathbf{0}$$

$$\hat{\boldsymbol{w}}(k) = \hat{\boldsymbol{w}}(k-1) + \rho_k \boldsymbol{x}_k \left(y_k - \boldsymbol{x}_k^T \hat{\boldsymbol{w}}(k-1)\right)$$

The algorithm is known as Widrow–Hoff algorithm.

Rosenblatt

$$\boldsymbol{w}(t+1) = \boldsymbol{w}(t) + \rho \boldsymbol{x}_{(t)} \quad \text{if } \boldsymbol{x}_{(t)} \in \omega_1 \text{ and } \boldsymbol{w}^T(t)\boldsymbol{x}_{(t)} \leq 0$$

$$\boldsymbol{w}(t+1) = \boldsymbol{w}(t) - \rho \boldsymbol{x}_{(t)} \quad \text{if } \boldsymbol{x}_{(t)} \in \omega_2 \text{ and } \boldsymbol{w}^T(t)\boldsymbol{x}_{(t)} \geq 0$$

$$\boldsymbol{w}(t+1) = \boldsymbol{w}(t) \quad \text{otherwise}$$

Adaline

$$\hat{\boldsymbol{w}}(k) = \hat{\boldsymbol{w}}(k-1) + \rho_k \boldsymbol{x}_k \left(y_k - \boldsymbol{x}_k^T \hat{\boldsymbol{w}}(k-1)\right) \quad y_k = \pm 1$$

Sum of Error Squares Estimation

$$J(\boldsymbol{w}) = \sum_{i=1}^{N} \left(y_i - \boldsymbol{x}_i^T \boldsymbol{w}\right)^2 \equiv \sum_{i=1}^{N} e_i^2$$

$$\sum_{i=1}^{N} \boldsymbol{x}_i \left(y_i - \boldsymbol{x}_i^T \boldsymbol{w}\right) = 0 \Rightarrow \left(\sum_{i=1}^{N} \boldsymbol{x}_i \boldsymbol{x}_i^T\right) \boldsymbol{w} = \sum_{i=1}^{N} (\boldsymbol{x}_i y_i)$$

That is, X is an $N \times l$ matrix whose rows are the available training feature vectors, and $\boldsymbol{y}$ is a vector consisting of the corresponding desired responses. Then $\sum_{i=1}^{N} \boldsymbol{x}_i \boldsymbol{x}_i^T = X^T X$ and also $\sum_{i=1}^{N} \boldsymbol{x}_i y_i = X^T \boldsymbol{y}$. Hence

$$(X^T X)\hat{\boldsymbol{w}} = X^T \boldsymbol{y} \Rightarrow \hat{\boldsymbol{w}} = (X^T X)^{-1} X^T \boldsymbol{y}$$

Example 3.4

Class ω_1 consists of the two-dimensional vectors $[0.2, 0.7]^T$, $[0.3, 0.3]^T$, $[0.4, 0.5]^T$, $[0.6, 0.5]^T$, $[0.1, 0.4]^T$ and class ω_2 of $[0.4, 0.6]^T$, $[0.6, 0.2]^T$, $[0.7, 0.4]^T$, $[0.8, 0.6]^T$, $[0.7, 0.5]^T$. Design the sum of error squares optimal linear classifier $w_1x_1 + w_2x_2 + w_0 = 0$.

We first extend the given vectors by using 1 as their third dimension and form the 10×3 matrix X, which has as rows the transposes of these vectors. The resulting sample correlation 3×3 matrix X^TX is equal to

$$X = \begin{bmatrix} 0.2 & 0.7 & 1 \\ \vdots & \vdots & \vdots \\ 0.7 & 0.5 & 1 \end{bmatrix}$$

$$X^TX = \begin{bmatrix} 2.8 & 2.24 & 4.8 \\ 2.24 & 2.41 & 4.7 \\ 4.8 & 4.7 & 10 \end{bmatrix}$$

The corresponding $\boldsymbol{y}$ consists of five 1's and then five -1's and

$$X^T\boldsymbol{y} = \begin{bmatrix} -1.6 \\ 0.1 \\ 0.0 \end{bmatrix}$$

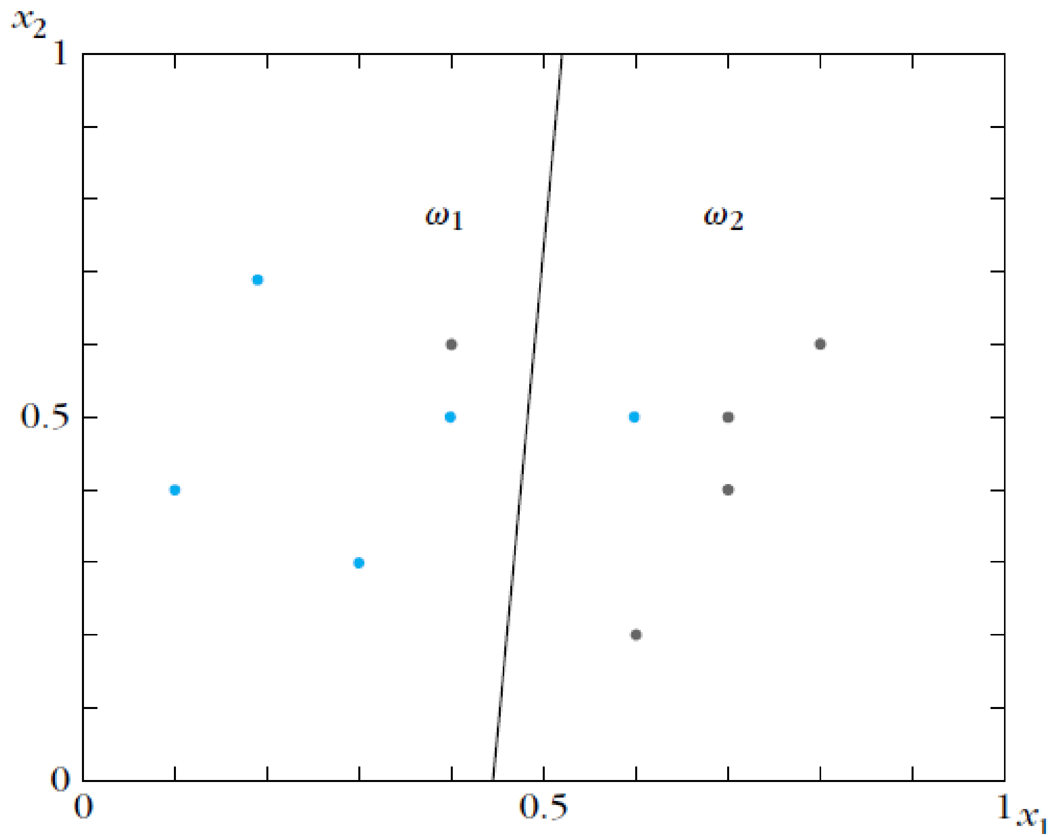


FIGURE 3.7

Least sum of error squares linear classifier. The task is not linearly separable. The linear LS classifier classifies some of the points in the wrong class. However, the resulting sum of error squares is minimum.