

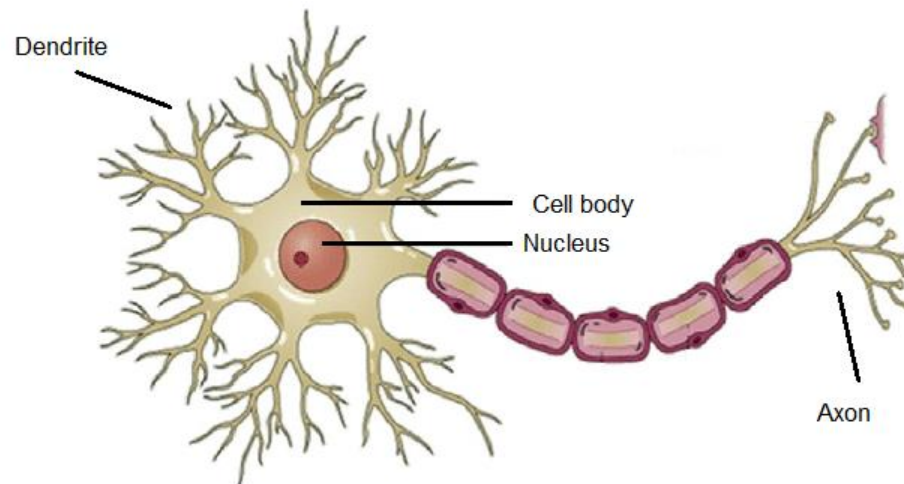


فصل سوم شناسایی الگو طبقه‌بندی کننده‌های خطی LINEAR CLASSIFIERS

محمد جواد فدائی اسلام

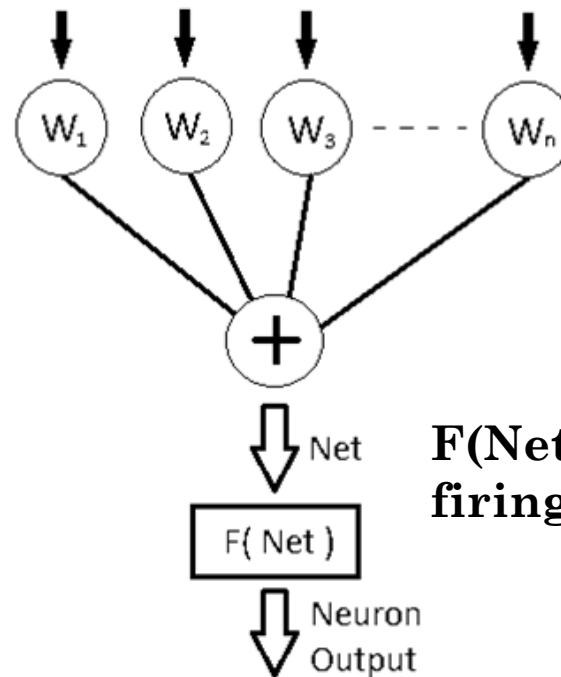
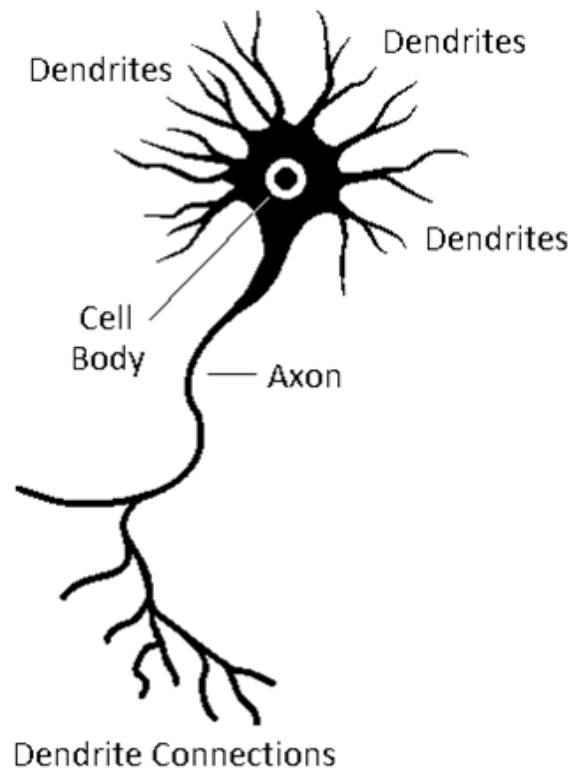
SIMPLE BIOLOGICAL NEURON

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



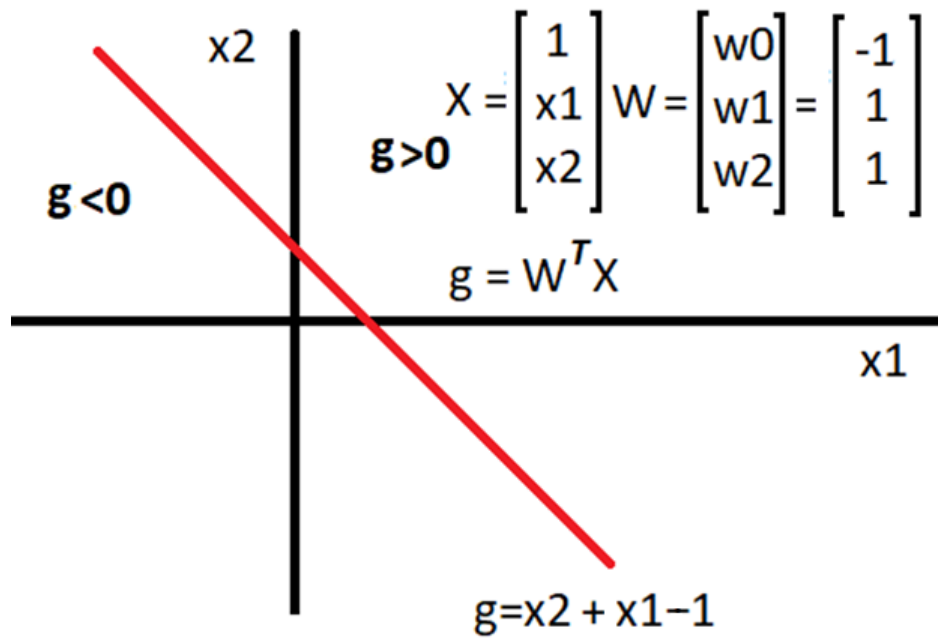
ARTIFICIAL NEURAL NETWORKS

- نورون مصنوعی دارای یک سری ورودی وزن دار و یک جمع کننده است (سمت راست).
- یک تابع فعال سازی (آتش)، که در صورت گذر جمع مقدار ورودی از آستانه، سیگنال را به نرون بعدی ارسال می کند.



**$F(Net) =$
firing mechanism**

معادله خط



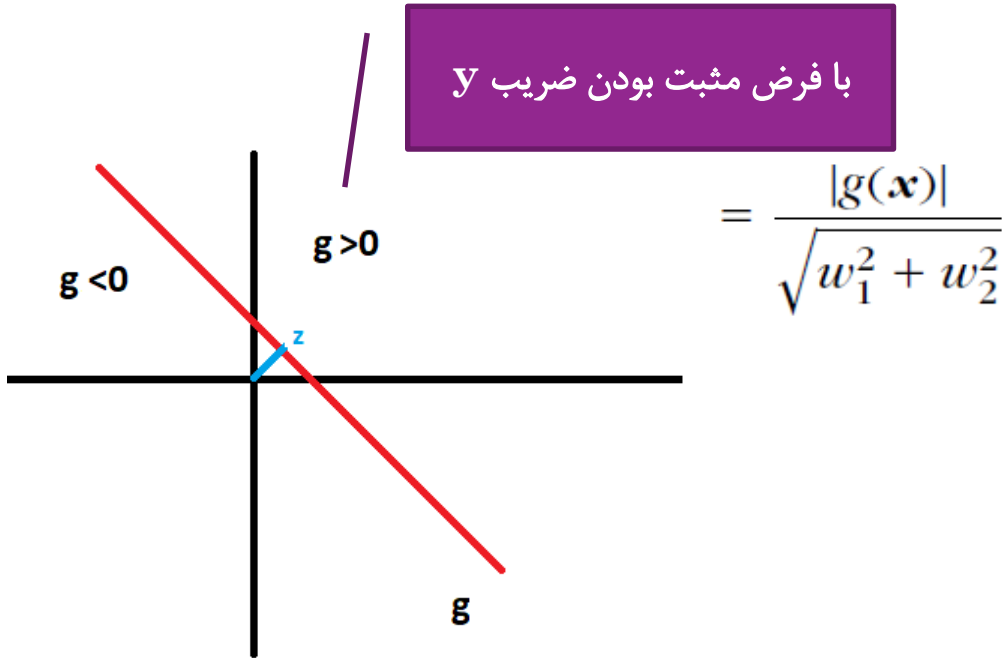
LINEAR DISCRIMINANT FUNCTIONS AND DECISION HYPERPLANES

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

$$g(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + w_0 = 0$$

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

فاصله از خط راست



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

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

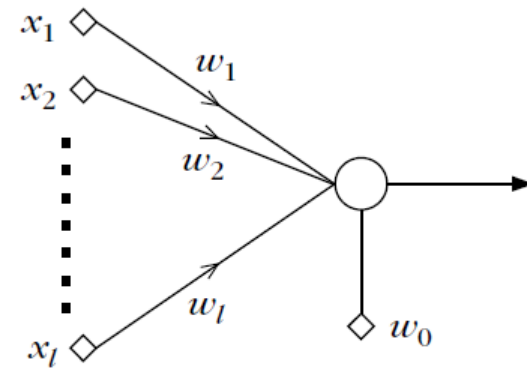
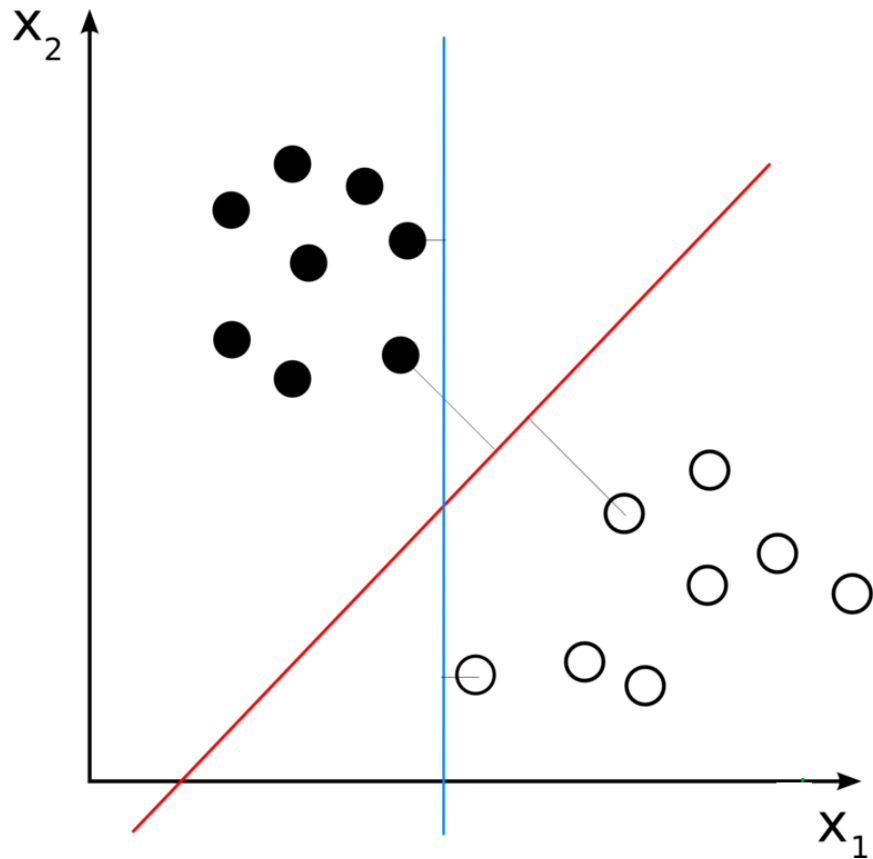
معادله خطوط مختلف برای یک خط

$$g = y + x - 1$$

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

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

طبقه‌بند خطی، مدل پرسپترون



(b)

The basic perceptron model

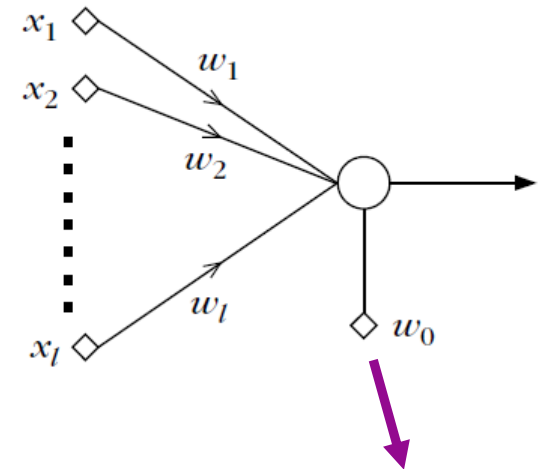
x_i ویژگی است.
خط قرمز و آبی جداکننده داده‌های دو کلاس هستند.
فاصله نمونه‌ها از خط قرمز بیشتر است.

تابع جداکننده خطی و ابرصفحه تصمیم‌گیری

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

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

$$g = W^T X$$



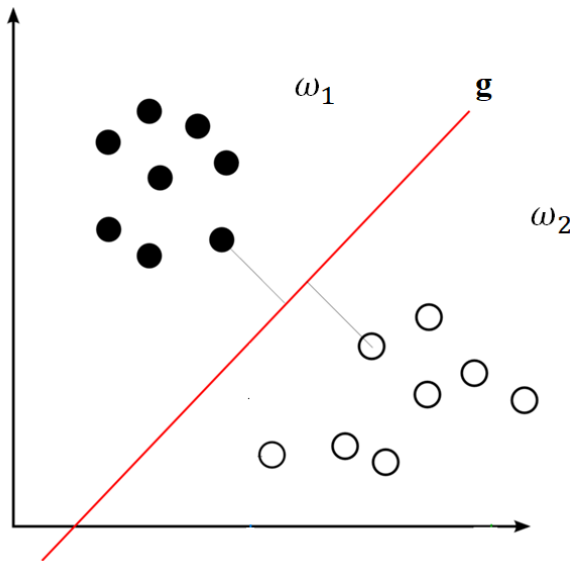
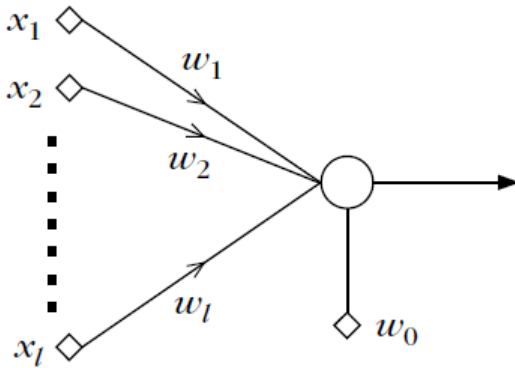
عرض از مبدا

T = transpose = ترانهاده

آموزش پرسپترون

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

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



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

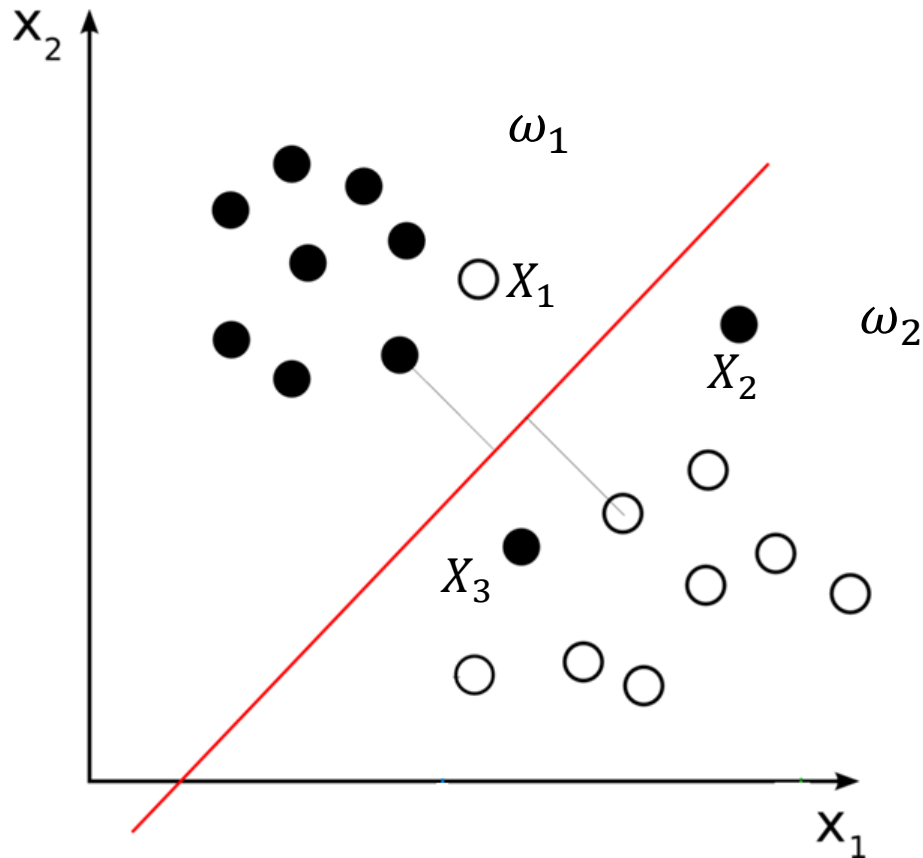
$$g = W^T X$$

THE PERCEPTRON COST FUNCTION

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

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

الگوریتم آموزش پرسپترون (دسته‌ای)



$\delta_x = -1, \text{ if } x \in \omega_1 \text{ (Black circle)}$

$\delta_x = +1, \text{ if } x \in \omega_2 \text{ (white circle)}$

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

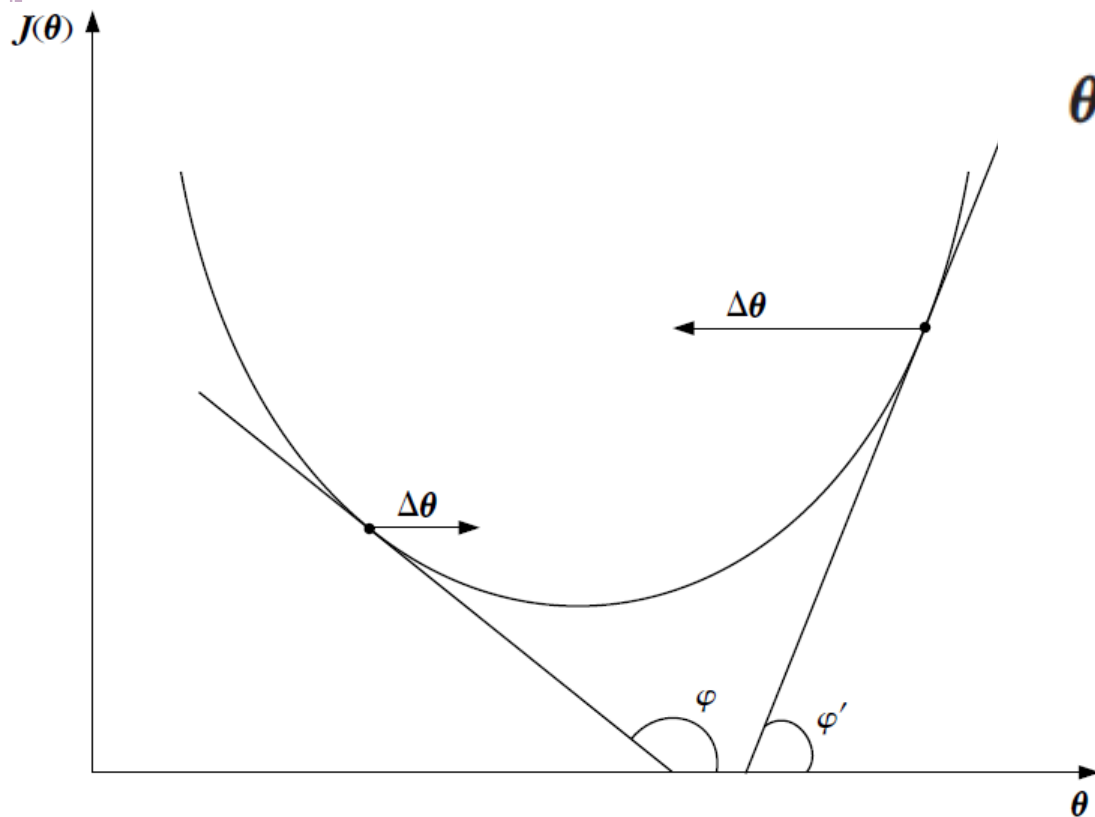
GRADIENT DESCENT METHOD TO MINIMIZE THE COST FUNCTION

$$\mathbf{w}(t + 1) = \mathbf{w}(t) - \rho_t \left. \frac{\partial J(\mathbf{w})}{\partial \mathbf{w}} \right|_{\mathbf{w}=\mathbf{w}(t)}$$

$$\frac{\partial J(\mathbf{w})}{\partial \mathbf{w}} = \sum_{\mathbf{x} \in Y} \delta_{\mathbf{x}} \mathbf{x}$$

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

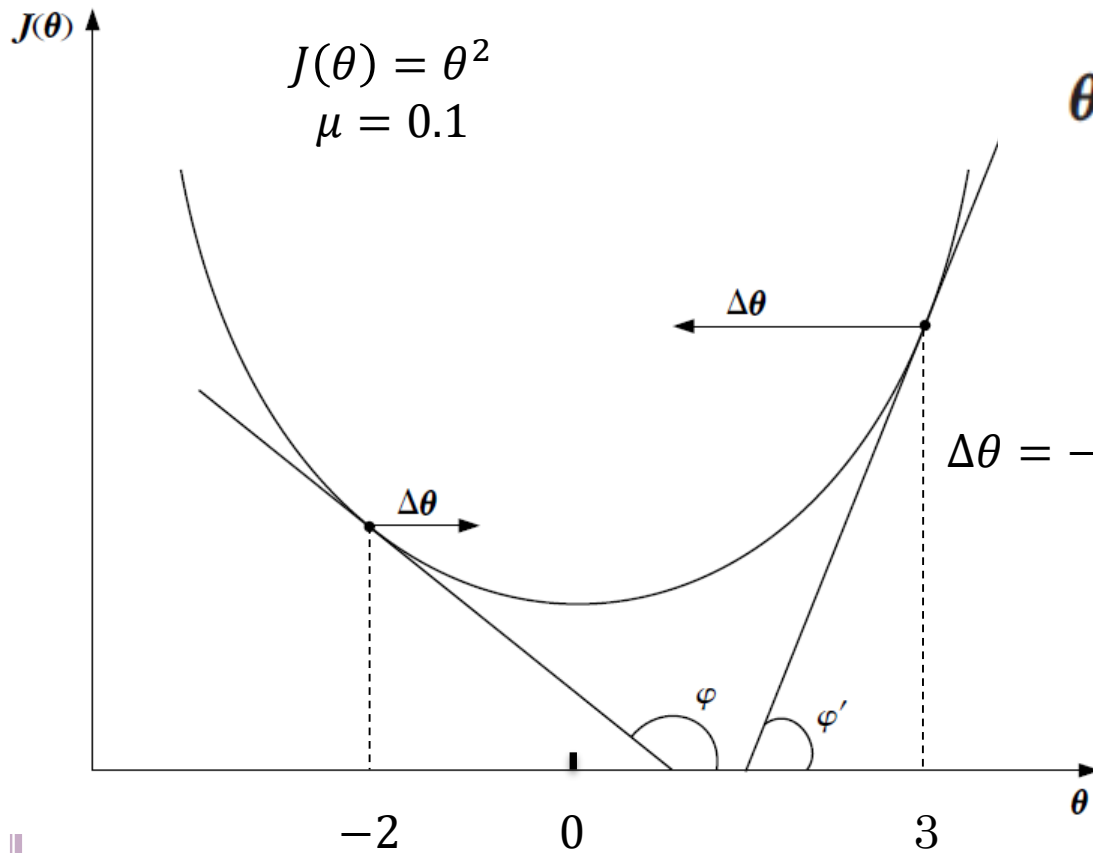
THE GRADIENT DESCENT METHOD



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

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

THE GRADIENT DESCENT METHOD-EXAMPLE



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

$$\Delta\theta = -\mu \left. \frac{\partial J(\theta)}{\partial \theta} \right|_{\theta=\theta(\text{old})}$$

$$\Delta\theta = -0.1 \times 2 \times 3 = -0.6$$

$$\Delta\theta = -0.1 \times 2 \times (-2) = -0.4$$

The Perceptron Algorithm

$$\rho_0 = 1$$

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

به مرور زمان ρ کاهش می یابد.

■ Choose $\mathbf{w}(0)$ randomly

■ Choose ρ_0

■ $t = 0$

■ Repeat

• $Y = \emptyset$

مجموعه نقاط با کلاس بندی نادرست

• For $i = 1$ to N

◦ If $\delta_{x_i} \mathbf{w}(t)^T \mathbf{x}_i \geq 0$ then $Y = Y \cup \{\mathbf{x}_i\}$

حلقه برای یافتن مجموعه نقاط با کلاس بندی نادرست

• End {For}

• $\mathbf{w}(t + 1) = \mathbf{w}(t) - \rho_t \sum_{\mathbf{x} \in Y} \delta_{\mathbf{x}} \mathbf{x}$

به روز رسانی وزن‌ها

• $t = t + 1$

■ Until $Y = \emptyset$

EXAMPLE

$$w(0): x_1 + x_2 - 0.5 = 0 \quad \rho = 0.7$$

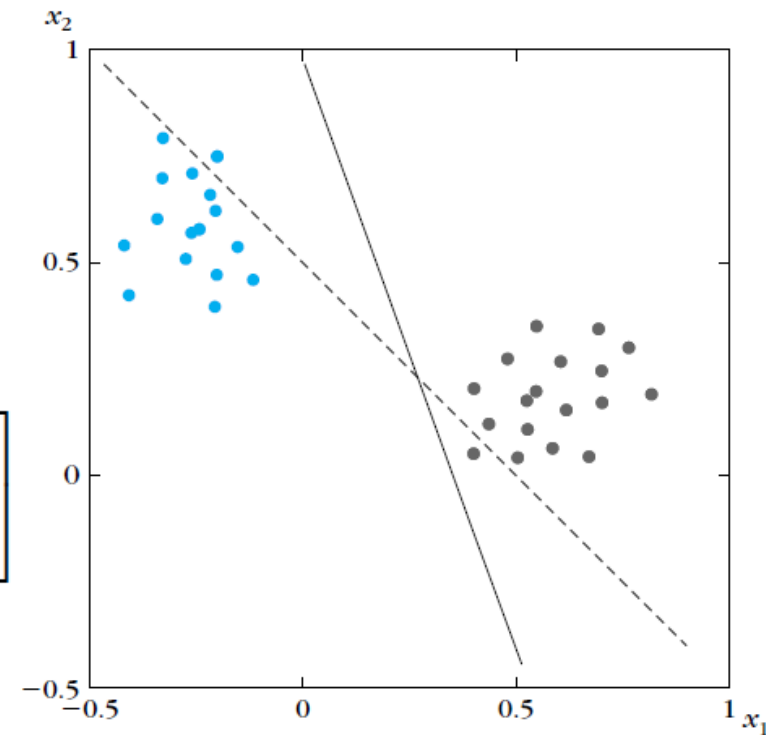
The line classifies correctly
all the vectors except $[0.4, 0.05]^T$ and $[-0.20, 0.75]^T$.
the next weight vector will be

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

or

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

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)

مثال

ρ_t must satisfy the following two conditions:

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

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

$$\begin{aligned} \rho_k &= \frac{1}{k} \\ \sum_{k=1}^{\infty} \frac{1}{k} &= +\infty \\ \sum_{k=1}^{\infty} \frac{1}{k^2} &= \frac{\pi^2}{6} \end{aligned}$$

ROSENBLATT ALGORITHM, SAMPLE MODE, PATTERN MODE:

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

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

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

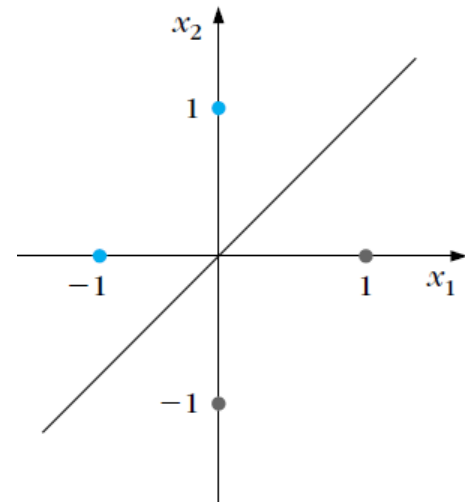
الگوریتم رزنبلات - مثال

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 $\mathbf{w}(0) = [0, 0, 0]^T$ in the extended three-dimensional space.

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

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



EXAMPLE 2 (CONT)

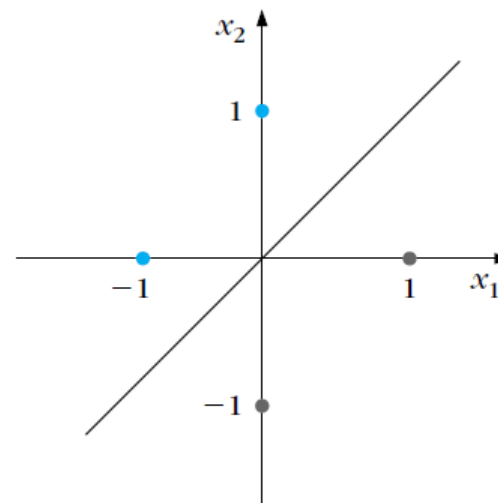
Step 3. $\mathbf{w}^T(2) \begin{bmatrix} 0 \\ -1 \\ 1 \end{bmatrix} = 1 > 0, \quad \mathbf{w}(3) = \mathbf{w}(2) - \begin{bmatrix} 0 \\ -1 \\ 1 \end{bmatrix} = \begin{bmatrix} -1 \\ 1 \\ 0 \end{bmatrix}$

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

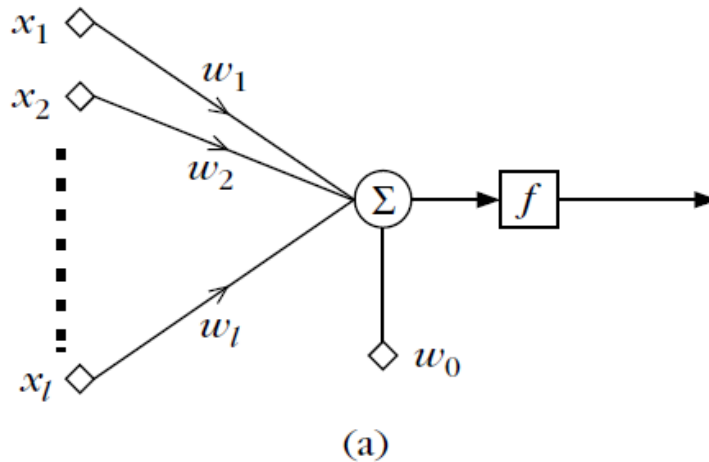
Step 5. $\mathbf{w}^T(4) \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix} = 1 > 0, \quad \mathbf{w}(5) = \mathbf{w}(4)$

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

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



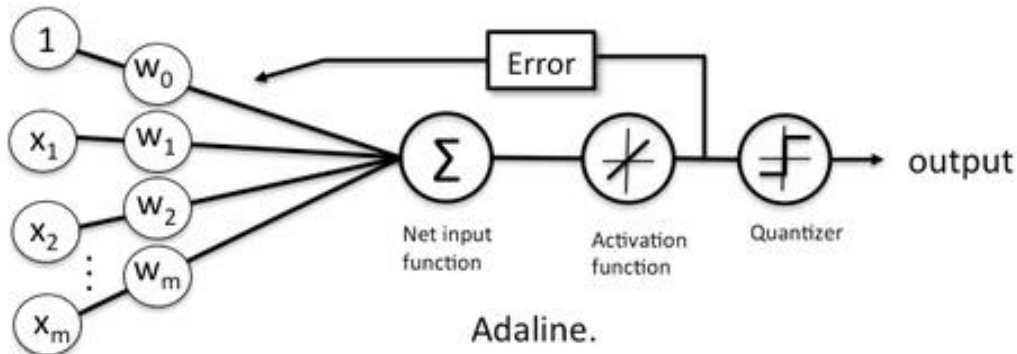
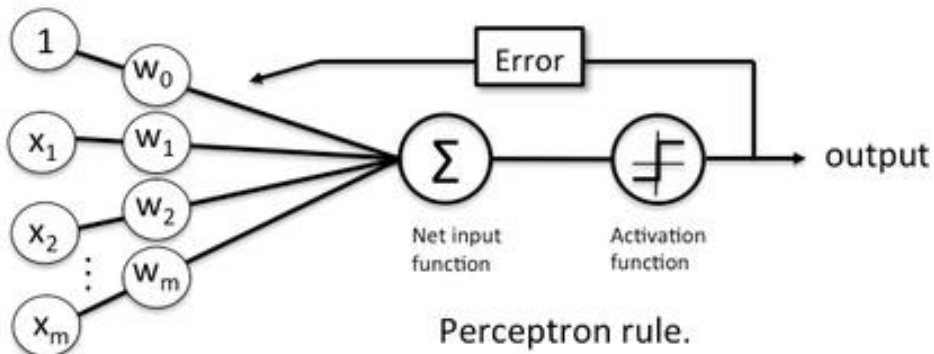
THE BASIC PERCEPTRON MODEL



$f(\cdot)$ 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



- 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.
- The algorithm is known as **Widrow–Hoff** algorithm.

$$w(t + 1) = w(t) + \rho_t x_t (y_t - x_t^T w(t))$$

ROSENBLATT

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

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

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

ADALINE

$$w(t+1) = w(t) + \rho_t x_t (y_t - x_t^T w(t)) \quad y_t = \pm 1$$

SUM OF ERROR SQUARES ESTIMATION

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

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

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

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

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 $X^T X$ is equal to

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

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

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

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

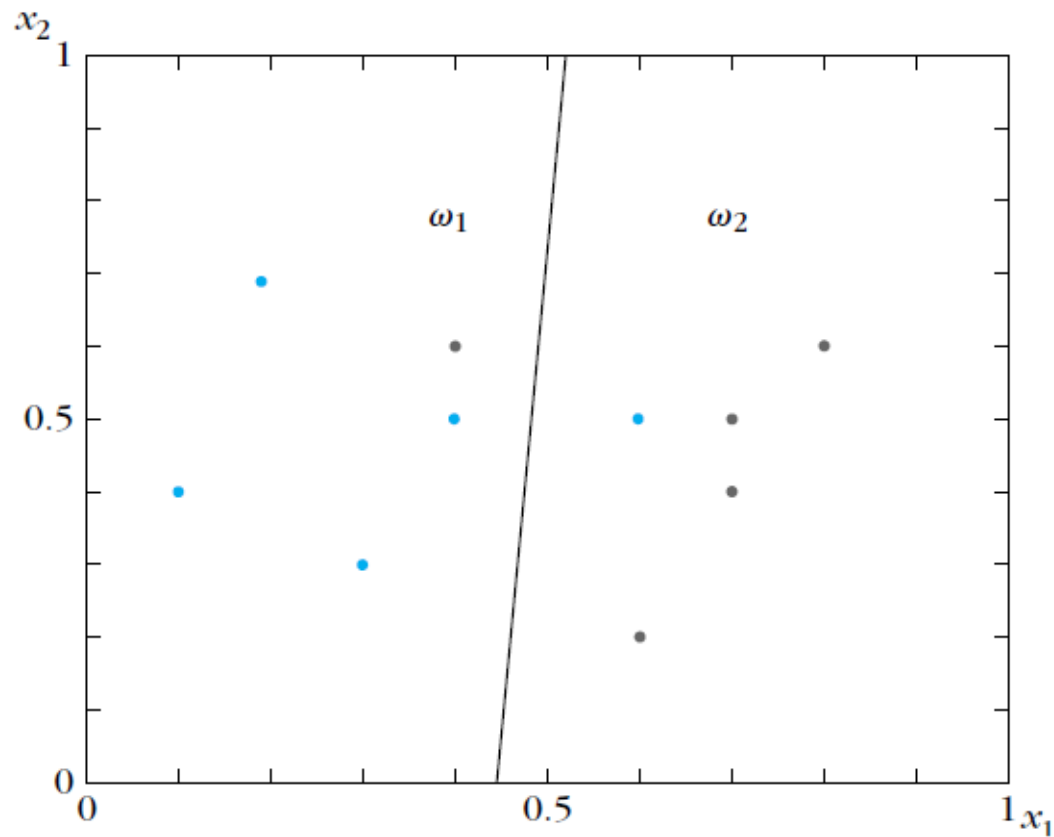


FIGURE 3.7

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.

تمرین

۱- داده‌های دو کلاس زیر داده شده است، روش پرسپترون دسته‌ای را بر روی این داده‌ها اعمال نمایید.

$$W_0 = (1, 1, -5), \rho_0 = 1/2$$

ω_2	ω_1
(6,9)	(0,0)
(5,9)	(0,1)
(9,8)	(2,2)
(9,9)	(3,1)
(9,10)	(3,2)
(8,11)	(3,3)

۲- داده‌های بالا را با استفاده از روش تخمین با استفاده از جمع مربعات خطا (روش ماتریسی - تکرار شونده نیست) کلاس‌بندی نمایید (حل می‌تواند با ابزار باشد - پایتون یا متلب).