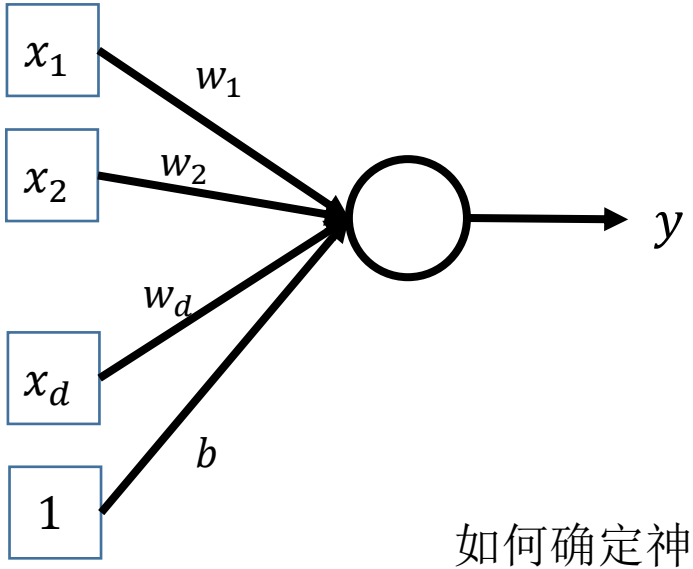
深度学习 第二讲

神经元学习算法

王文中

安徽大学计算机学院

回顾：MCP神经元模型



$$z = \sum_{i=1}^{d} w_i x_i + b = W^T X + b$$

$$y = sign(z)$$

$$y = f(X; W, b) = f(W^T X + b)$$

如何确定神经元模型的参数$\{w_i\}_{i=1}^{d}, b$？

分类与回归

$$y = f(X; \Theta)$$

- 分类(Classification)
  - y是离散值（0，1，2，…）
  - 例如：垃圾邮件分类；图像识别。
- 回归(Regression):
  - $y \in R$是连续值
  - 例如：根据人的身高、体重、年龄估计收缩压；在动物学研究中，根据某种动物的体重估计它的体积。

有监督学习(Supervised Learning)

- 参数化的模型:
  - $y = f(X; \Theta), X \in R^d$, 参数$\Theta$未知
- 训练样本$D = \{X^{(i)}, y^{(i)}\}_{i=1}^{n}, X^{(i)} \in R^d$：特征向量
- 训练/学习阶段(Training/Learning)
  - 用学习算法A根据训练样本D估计最佳模型参数: $\Theta^* = A(D, f)$
- 推断/测试阶段(Inference/Testing)
  - 对新的测试样本$X \notin D$，用模型预测对应的输出y: $\hat{y} = f(X; \Theta^*)$

预备知识

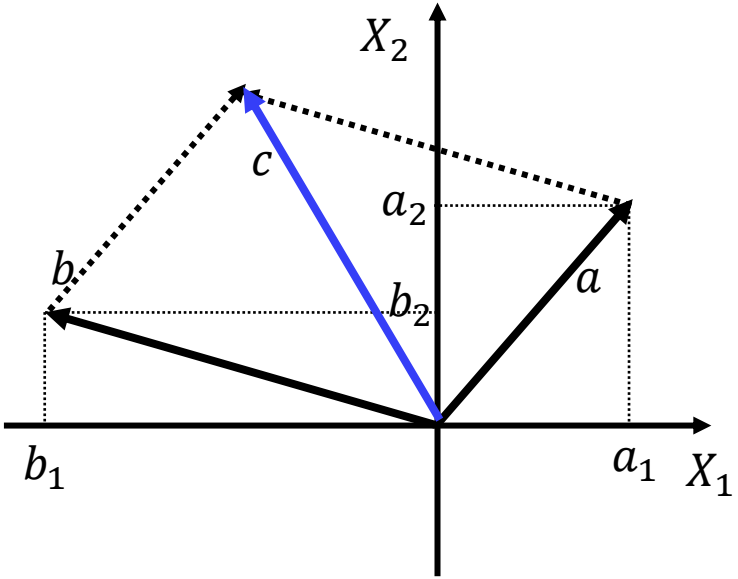
向量

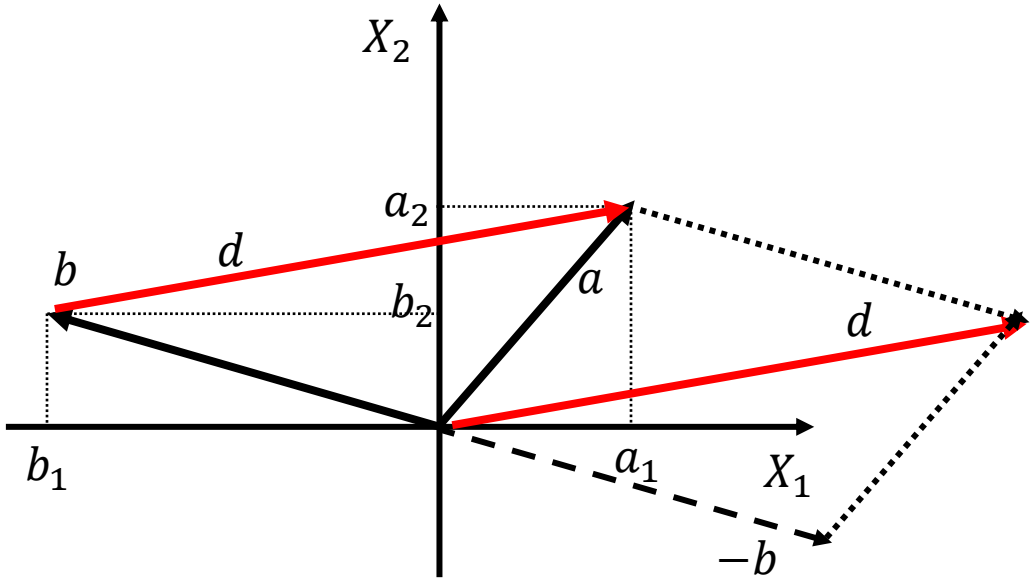
$a = (a_1, a_2, \ldots, a_m) \in R^m$

$-a = (-a_1, -a_2, \ldots, -a_m)$

$c = a + b = (a_1 + b_1, a_2 + b_2, \ldots, a_m + b_m)$

$d = a - b = (a_1 - b_1, a_2 - b_2, \ldots, a_m - b_m)$

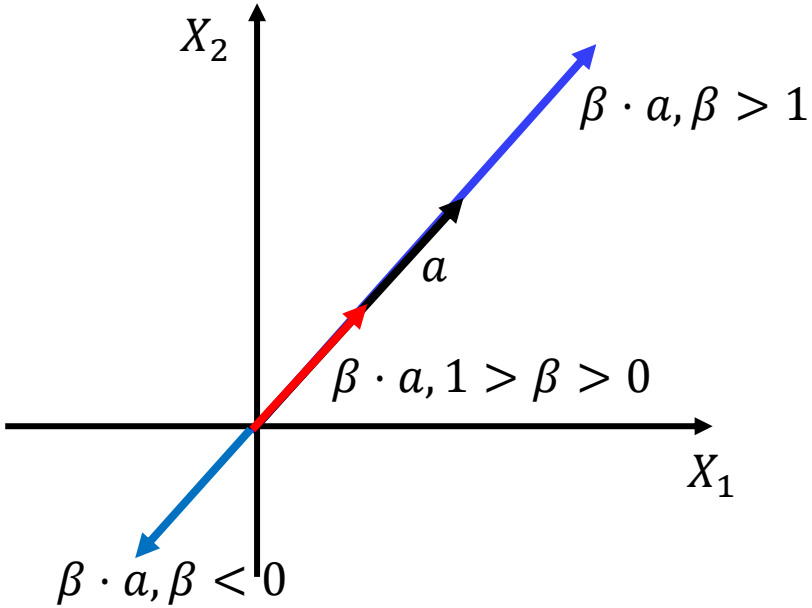




向量

$a = (a_1, a_2, \ldots, a_m) \in R^m$

$\beta \cdot a = (\beta \cdot a_1, \beta \cdot a_2, \ldots, \beta \cdot a_m), \beta \in R$



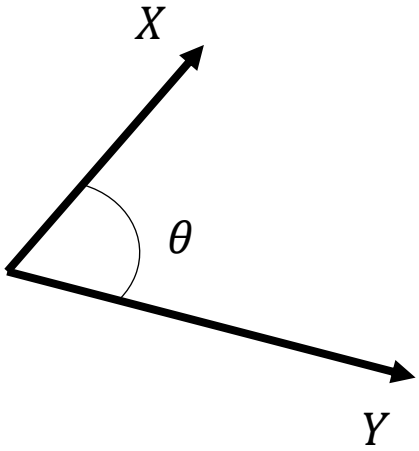
向量

$$X = (x_1, x_2, \dots, x_m) \in R^m$$

$$Y = (y_1, y_2, \dots, y_m) \in R^m$$

内积：$X^T Y = Y^T X = \sum_{i=1}^{m} x_i y_i \in R$

模长：$\|X\| = \sqrt{X^T X} = \sqrt{\sum_{i=1}^{m} x_i^2}$



$$X^T Y = \|X\| \cdot \|Y\| \cdot \cos(\theta)$$
$$X \perp Y \equiv X^T Y = 0$$

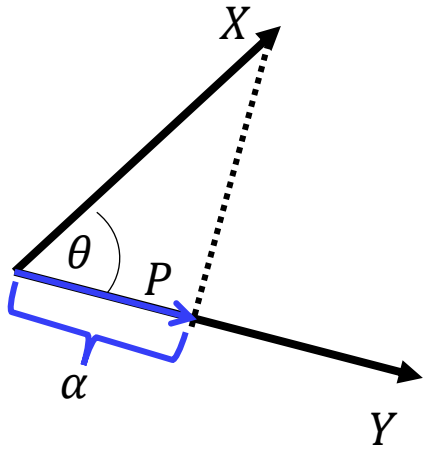
向量

内积：$X^TY = Y^TX = \|X\| \cdot \|Y\| \cdot \cos\theta$

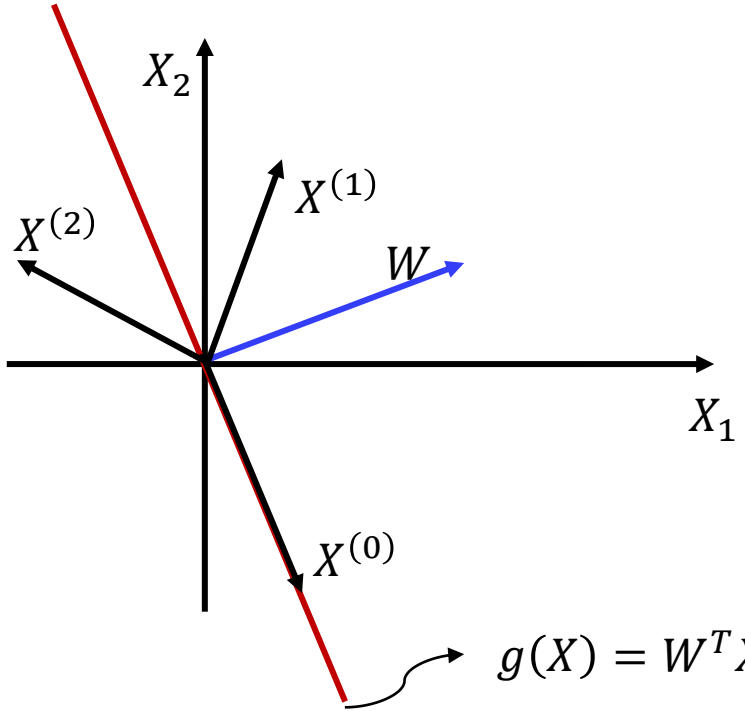
向量X在向量Y上的投影向量P

$$P = \alpha \times \frac{Y}{\|Y\|}, \alpha = \|X\| cos\theta$$

$$P = \frac{\|X\| \cdot Y \cdot cos\theta}{\|Y\|} = \frac{\|X\| \cdot \|Y\| \cdot \cos\theta}{\|Y\|^2} Y = \frac{X^TY}{Y^TY} Y$$



齐次线性方程$W^TX=0$

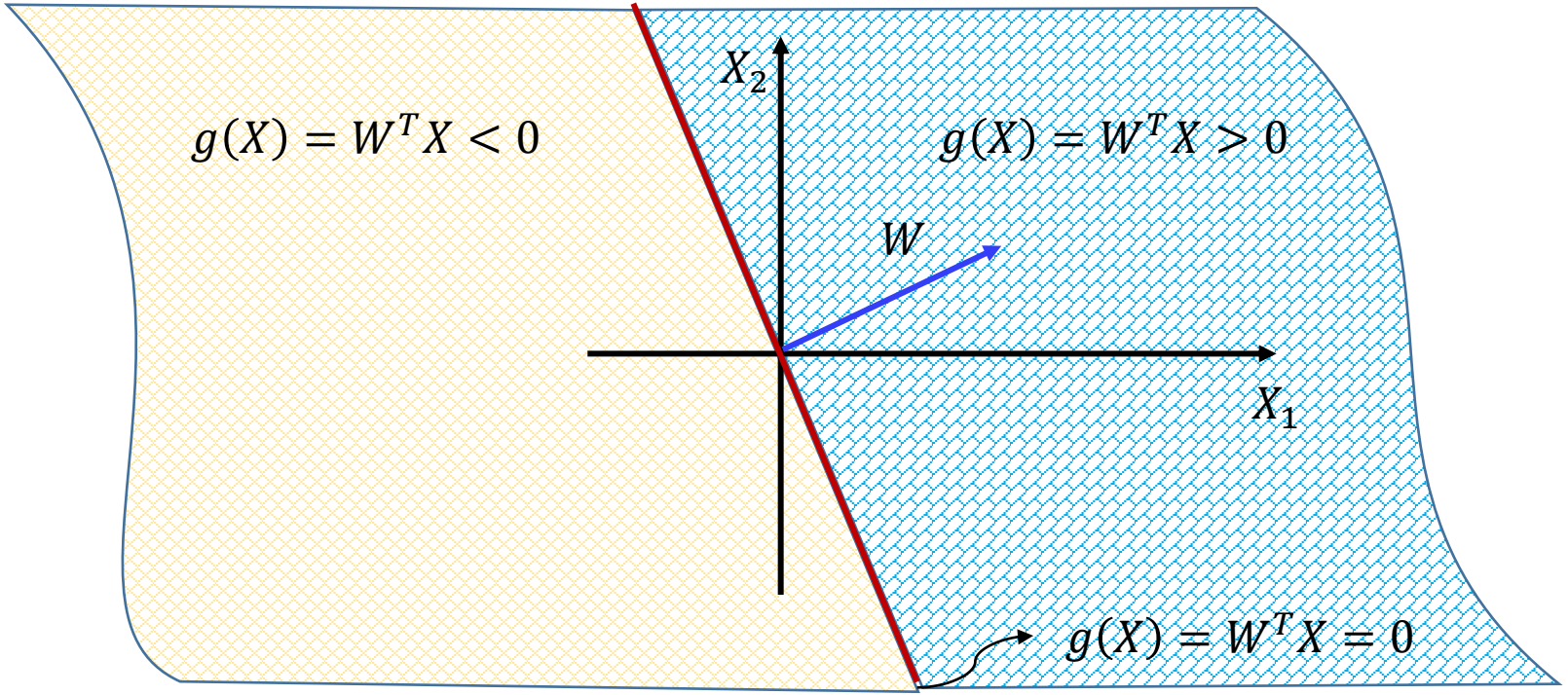


$$W^TX^{(0)}=\|W\|\cdot\left\|X^{(0)}\right\|\cdot cos\theta_0=0$$

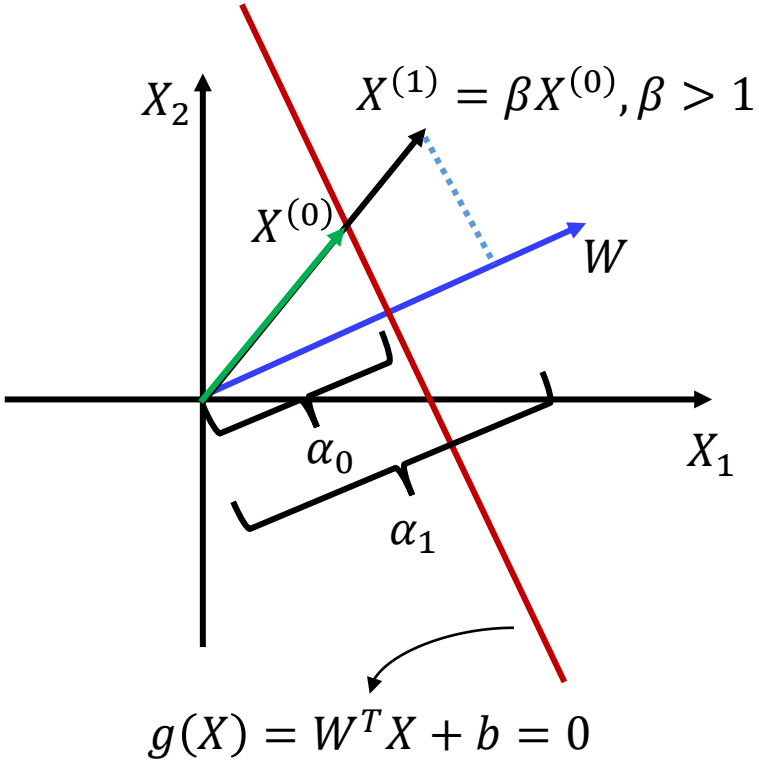
$$W^TX^{(1)}=\|W\|\cdot\left\|X^{(0)}\right\|\cdot cos\theta_1>0$$

$$W^TX^{(2)}=\|W\|\cdot\left\|X^{(0)}\right\|\cdot cos\theta_2<0$$

齐次线性方程$W^TX=0$



非齐次线性方程$g(X) = W^T X + b = 0$



$$\alpha_0 = \left\|X^{(0)}\right\| \cdot cos\theta > 0$$

$$g\left(X^{(0)}\right) = 0 \Rightarrow W^T X^{(0)} + b = 0 \Rightarrow$$

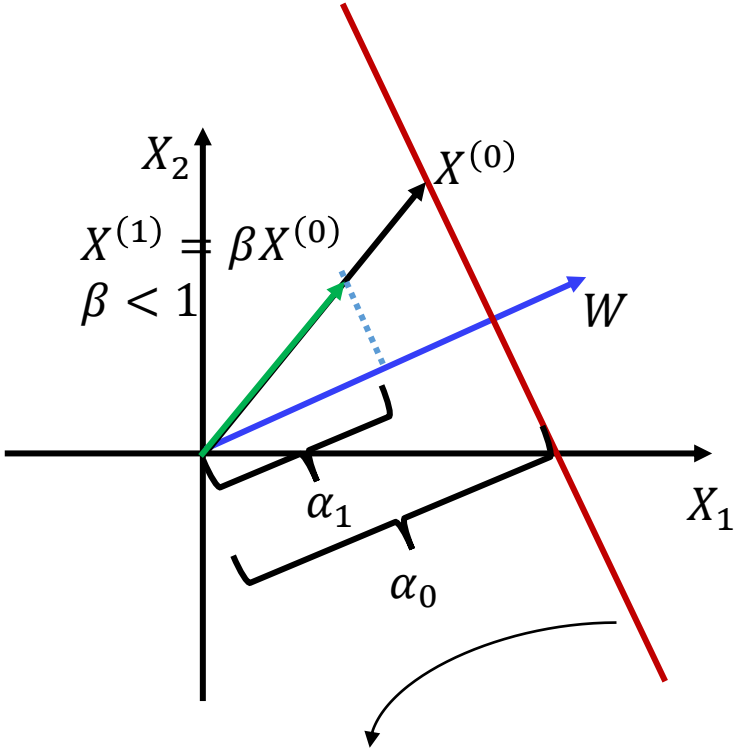
$$\|W\| \cdot \left\|X^{(0)}\right\| \cdot cos\theta + b = \|W\| \cdot \alpha_0 + b = 0$$

$$g\left(X^{(1)}\right) = W^T X^{(1)} + b = \beta \cdot \|W\| \cdot \alpha_0 + b =$$

$$(\beta - 1) \cdot \|W\| \cdot \alpha_0 + \|W\| \cdot \alpha_0 + b =$$

$$(\beta - 1) \cdot \|W\| \cdot \alpha_0 > 0$$

非齐次线性方程$g(X) = W^T X + b = 0$



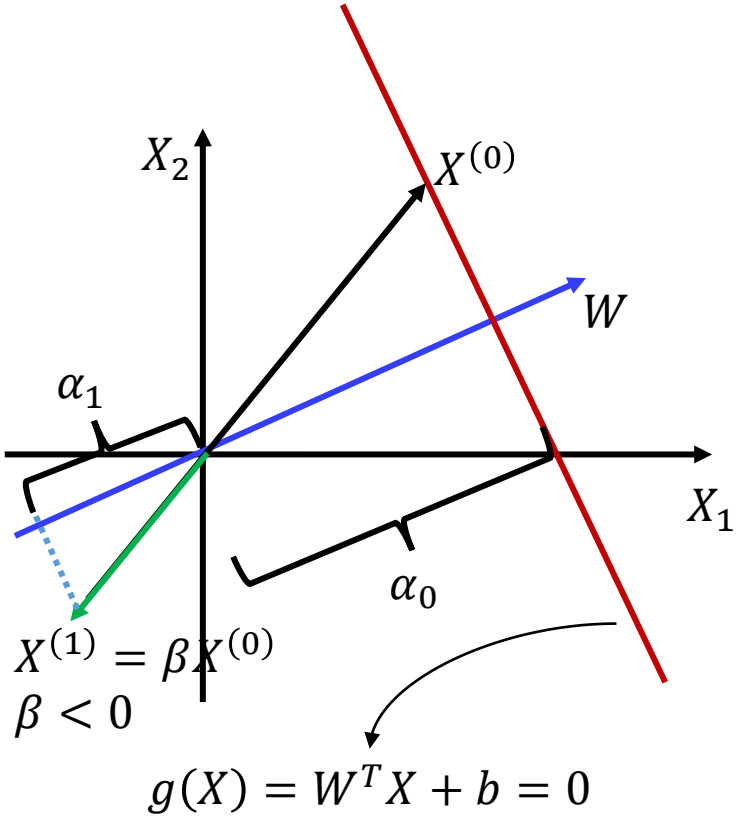
$$\alpha_0 = \left\|X^{(0)}\right\| \cdot cos\theta > 0$$

$$g\left(X^{(0)}\right) = 0 \Rightarrow W^T X^{(0)} + b = 0 \Rightarrow$$

$$\|W\| \cdot \left\|X^{(0)}\right\| \cdot cos\theta + b = \|W\| \cdot \alpha_0 + b = 0$$

$$g\left(X^{(1)}\right) = W^T X^{(1)} + b = (\beta - 1) \cdot \|W\| \cdot \alpha_0 < 0$$

非齐次线性方程$g(X) = W^T X + b = 0$



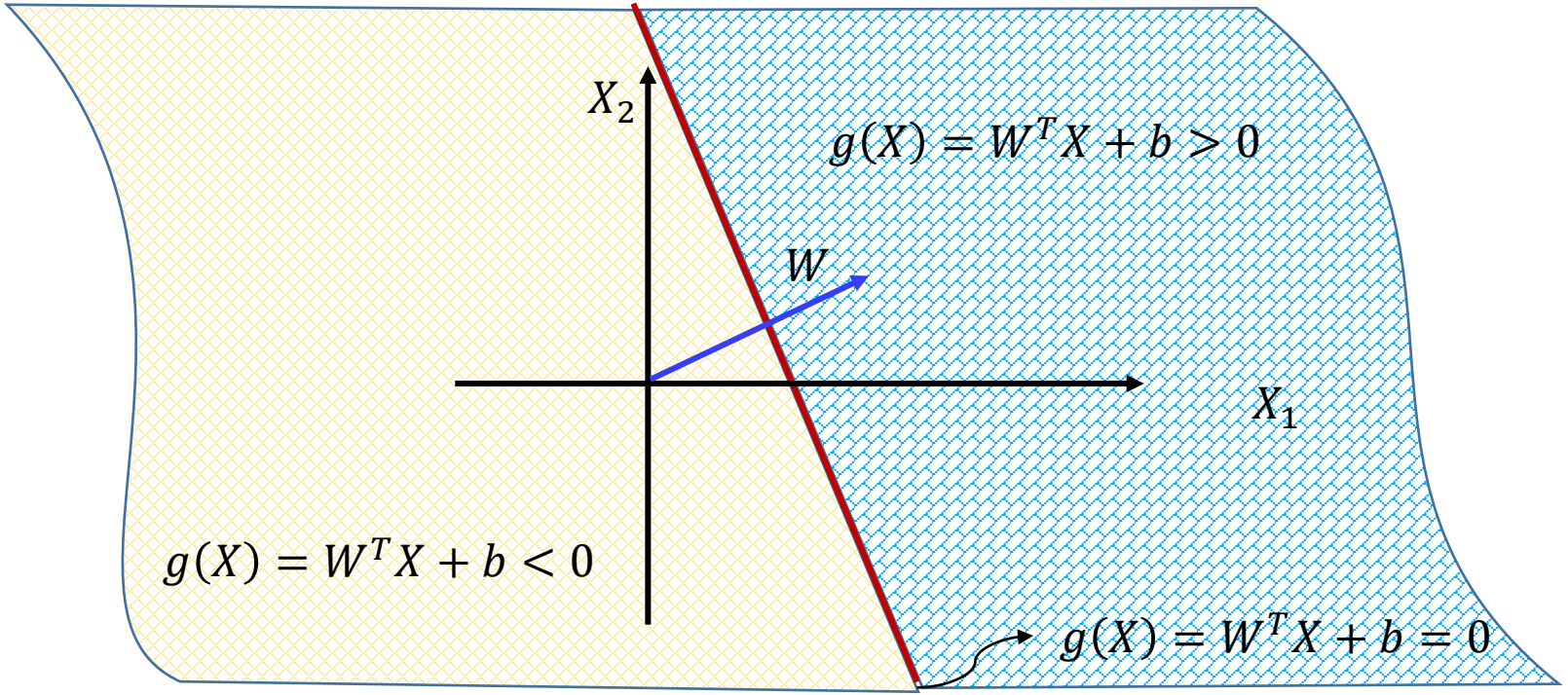
$$\alpha_0 = \left\|X^{(0)}\right\| \cdot cos\theta > 0$$

$$g\left(X^{(0)}\right) = 0 \Rightarrow W^T X^{(0)} + b = 0 \Rightarrow$$

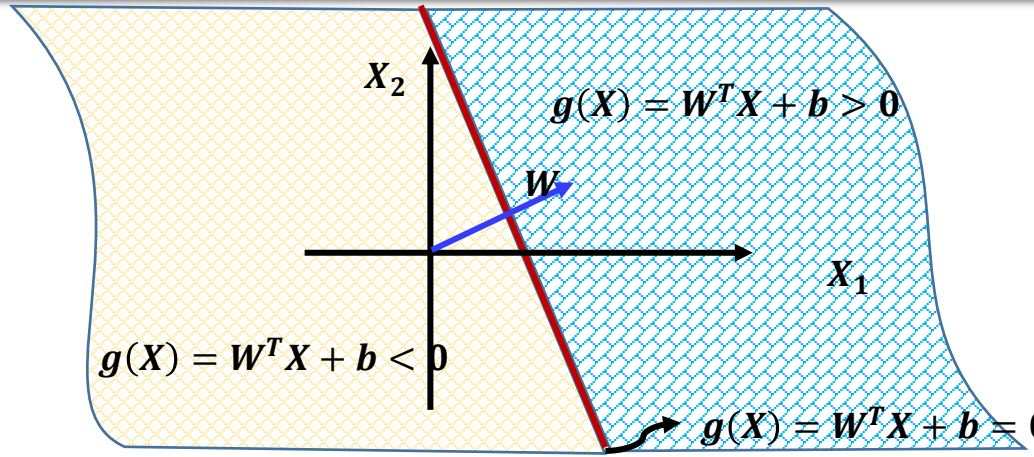
$$\|W\| \cdot \left\|X^{(0)}\right\| \cdot cos\theta + b = \|W\| \cdot \alpha_0 + b = 0$$

$$g\left(X^{(1)}\right) = W^T X^{(1)} + b = (\beta - 1) \cdot \|W\| \cdot \alpha_0 < 0$$

非齐次线性方程$g(X) = W^T X + b = 0$



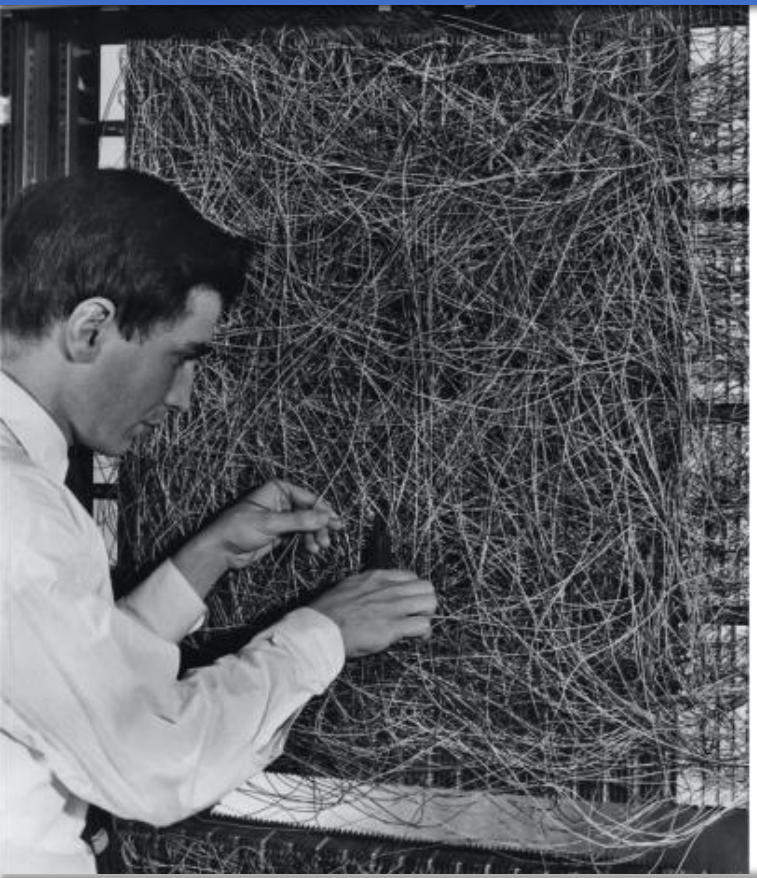
线性方程$g(X)=W^TX+b=0$定义了一个分界面



- $g(X)=W^TX+b=0$是n维空间的一个超平面(Hyper-Plane)
  - 向量W是这个分界面(超平面)的法向（即超平面的朝向）
  - 偏置参数b决定了超平面的位置
- $g(X)=W^TX+b=0$定义了n维空间的一个分界面(Separating-Plane)，把整个空间分为两部分(half-space)
  - 法向W所指向的一侧为正侧，位于该侧的所有点(向量)满足$g(X)>0$
  - 与法向W相反的一侧为负侧，位于该侧的所有点(向量)满足$g(X)<0$

感知机(Perceptron)

感知机模型



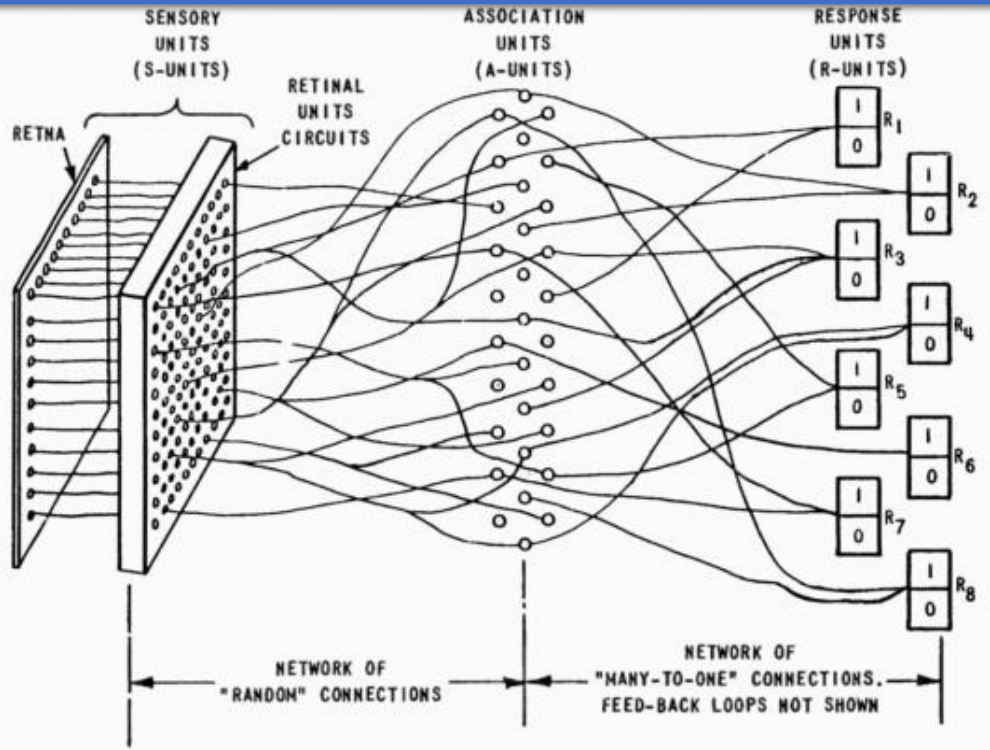
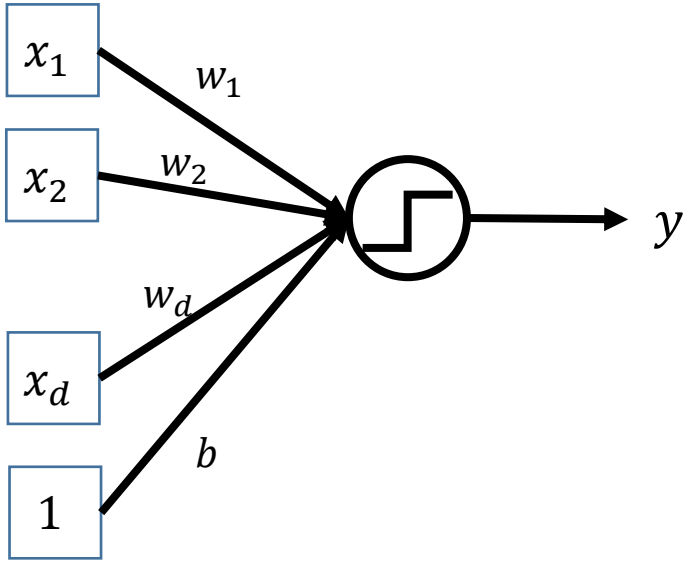


Figure I ORGANIZATION OF THE MARK I PERCEPTRON

感知机模型

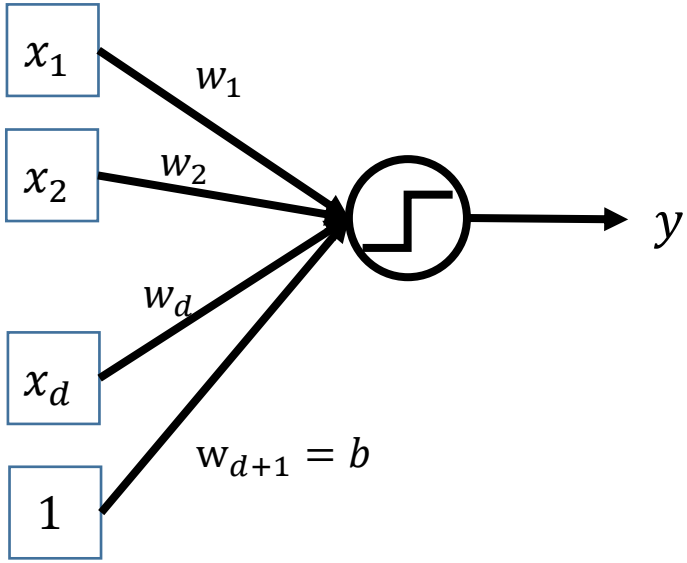


$$z = g(X) = \sum_{i=1}^{d} w_i x_i + b = W^T X + b,$$

$$y = f(X) = sign\big(g(X)\big) = \begin{cases} +1 & g(X) \geq 0 \\ -1 & g(X) < 0 \end{cases}$$

$$X = (x_1, x_2, \ldots, x_d)^T \in R^d$$

感知机模型



$$z = g(X) = \sum_{i=1}^{d+1} w_i x_i = W^T X, x_{d+1} = 1$$

$$y = f(X) = sign\big(g(X)\big) = \begin{cases} +1 & g(X) \geq 0 \\ -1 & g(X) < 0 \end{cases}$$

$$W = (w_1, w_2, \dots, w_d, b)^T \in R^{d+1}$$

$$X = (x_1, x_2, \dots, x_d, 1)^T \in R^{d+1}$$

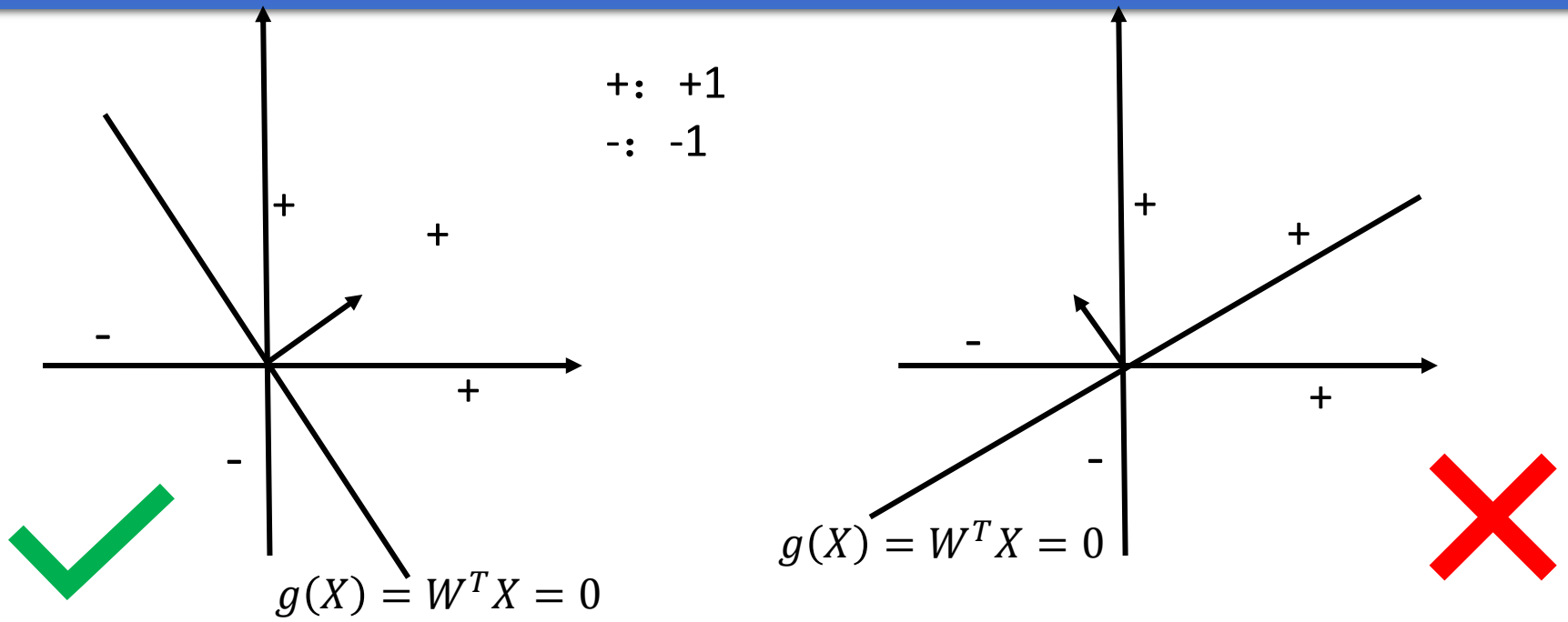
增广向量表示

感知机学习：根据训练样本确定参数W

$$y = f(X;W) = sign(W^T X) = \begin{cases} +1 & W^T X \geq 0 \\ -1 & W^T X < 0 \end{cases}$$

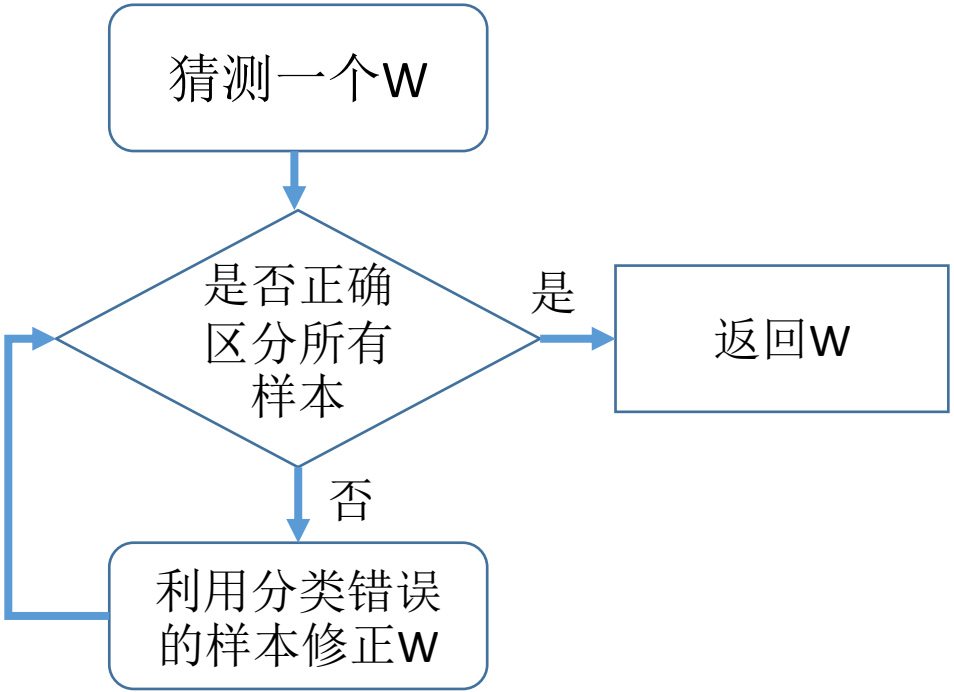
- 什么参数W是最"好"的？
- 如何找到最"好"的W?

感知机学习

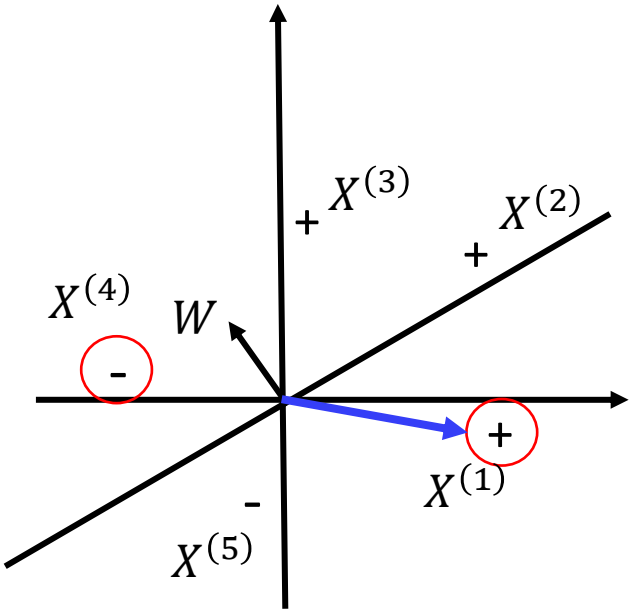


感知机学习目标：寻找合适的W正确区分所有样本

感知机学习算法



感知机学习算法



W对样本$(X^{(1)}, y=+1)$分类错误：

$$g\left(X^{(1)}\right) = W^T X^{(1)} < 0$$

$$\hat{y} = sign\left(g\left(X^{(1)}\right)\right) = -1 \neq y$$

若要正确分类$(X^{(1)}, y=+1) \Rightarrow$
修正W为W'，使得$W'^T X^{(1)}$变大 $\Rightarrow$

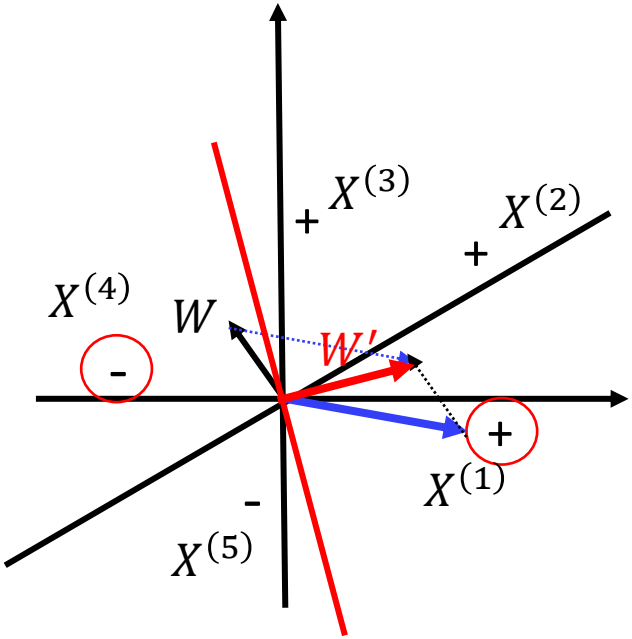
$$W' = W + \Delta, \Delta = ?$$

$$W'^T X^{(1)} = W^T X^{(1)} + \Delta^T X^{(1)} > W^T X^{(1)} \Rightarrow$$

$$\Delta^T X^{(1)} > 0 \Rightarrow$$

取$\Delta = \alpha \cdot X^{(1)}, \alpha > 0, \Delta^T X^{(1)} = \alpha {X^{(1)}}^T X^{(1)} > 0$

感知机学习算法



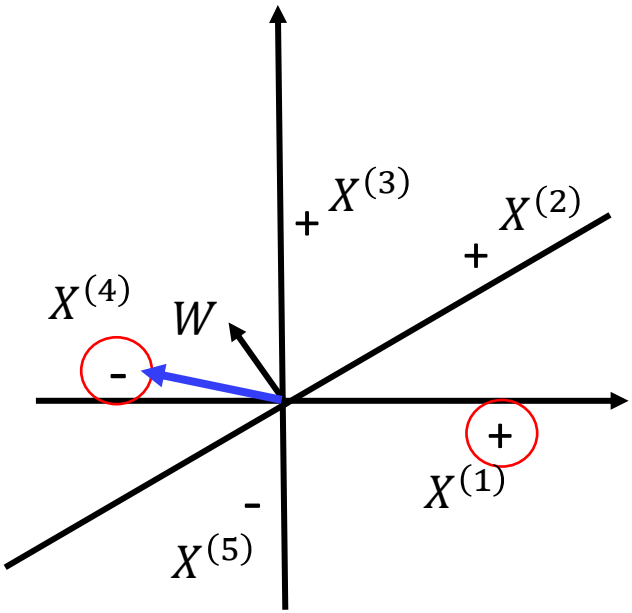
W对样本$(X^{(1)}, y = +1)$分类错误：

$$g\left(X^{(1)}\right) = W^T X^{(1)} < 0$$

$$\hat{y} = sign\left(g\left(X^{(1)}\right)\right) = -1 \neq y$$

$$W' = W + \alpha \cdot X^{(1)}$$

感知机学习算法



W对样本$(X^{(4)}, y^{(4)} = -1)$分类错误：

$$g\left(X^{(4)}\right) = W^T X^{(4)} > 0$$

$$\hat{y}^{(4)} = sign\left(g\left(X^{(4)}\right)\right) = +1 \neq y^{(4)}$$

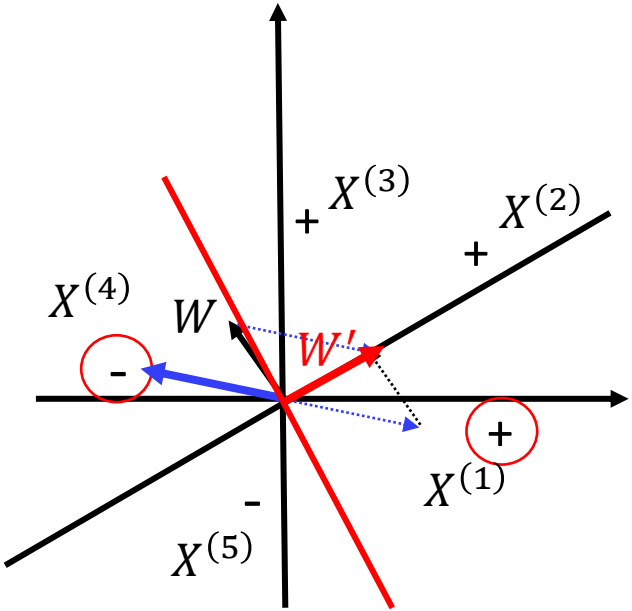
若要正确分类$(X^{(4)}, y^{(4)} = +1) \Rightarrow$
修正W为W'，使得$W'^T X^{(4)}$变小 $\Rightarrow$
$W' = W + \Delta, \Delta = ?$

$$W'^T X^{(4)} = W^T X^{(4)} + \Delta^T X^{(4)} < W^T X^{(4)} \Rightarrow$$

$$\Delta^T X^{(4)} < 0 \Rightarrow$$

取$\Delta = -\alpha \cdot X^{(4)}, \alpha > 0, \Delta^T X^{(4)} = -\alpha {X^{(4)}}^T X^{(4)} < 0$

感知机学习算法



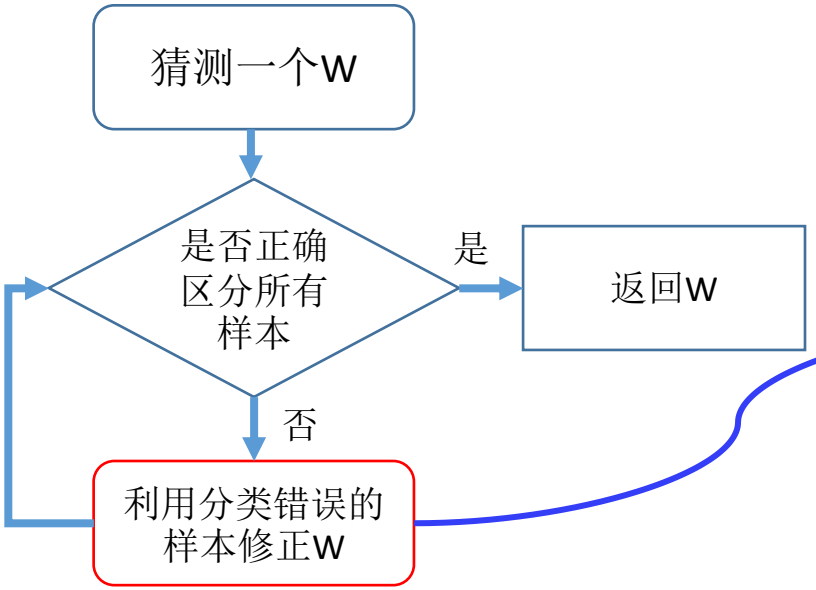
W对样本$(X^{(4)}, y^{(4)} = -1)$分类错误：

$$g\left(X^{(4)}\right) = W^T X^{(4)} > 0$$

$$\hat{y}^{(4)} = sign\left(g\left(X^{(4)}\right)\right) = +1 \neq y^{(4)}$$

$$W' = W - \alpha \cdot \mathrm{X}^{(4)}$$

感知机学习算法



- 若W对某个正样本$(X, y=+1)$分类错误:
  - $g(X)=W^TX<0$
  - $yg(X)=g(X)<0$
  - $W \leftarrow W+\alpha X$
- 若W对某个负样本$(X, y=-1)$分类错误:
  - $g(X)=W^TX>0$
  - $yg(X)=-W^TX<0$
  - $W \leftarrow W-\alpha X$

若W对某个样本(X, y)分类错误，即$yg(X)=yW^TX<0$，则按照如下方式更新W: $W \leftarrow W+\alpha yX$

感知机学习算法

若W对某个样本(X, y)分类错误，即$yg(X; W) = yW^T X < 0$，则按照如下方式更新W: $W' \leftarrow W + \alpha yX$

$$
\begin{aligned}
& yg(X; W') \\
&= yW'^T X = y(W + \alpha yX)^T X \\
&= yW^T X + \alpha y^2 X^T X \\
&= yW^T X + \alpha \|X\|^2 \\
&= yg(X; W) + \alpha \|X\|^2 \\
&> yg(X; W)
\end{aligned}
$$

感知机学习算法(PLA:Perceptron Learning Algorithm)

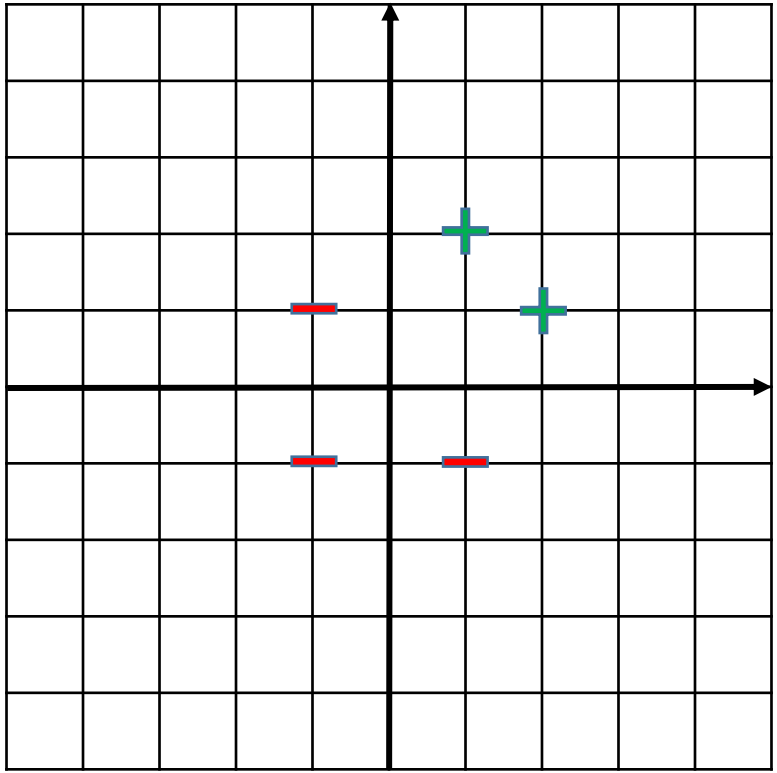
- 输入：训练样本$D = \left\{\left(X^{(i)} \in R^{d+1}, y^{(i)} \in \{-1, +1\}\right)\right\}_{i=1}^{n}$
- 输出：$f(X; W) = sign(W^T X)$
- 1. 初始化$W = \mathbf{0}$
- 2. While True:
  - 2.1 随机选取一个样本X, y，如果$yW^T X < 0$，该样本为错分样本，按2.2更新权重；如果不存在错分样本，退出循环；
  - 2.2 更新W: $W = W + \alpha yX$;
- 3. 返回W

假设X是增广向量,$W = (W, b)^T$

感知机学习算法(PLA:Perceptron Learning Algorithm)

- 输入：训练样本$D = \{(X^{(i)} \in R^d, y^{(i)} \in \{-1, +1\})\}_{i=1}^{n}$
- 输出：$f(X; W, b) = sign(W^T X + b)$
- 1. 初始化$W = \mathbf{0}$
- 2. While True:
  - 2.1 随机选取一个样本X, y，如果$yg(X) < 0$，该样本为错分样本，按2.2更新权重；如果不存在错分样本，退出循环;
  - 2.2 更新W: $W = W + \alpha y X$;
  - 2.3 更新b: $b = b + \alpha y$;
- 3. 返回W, b

PLA



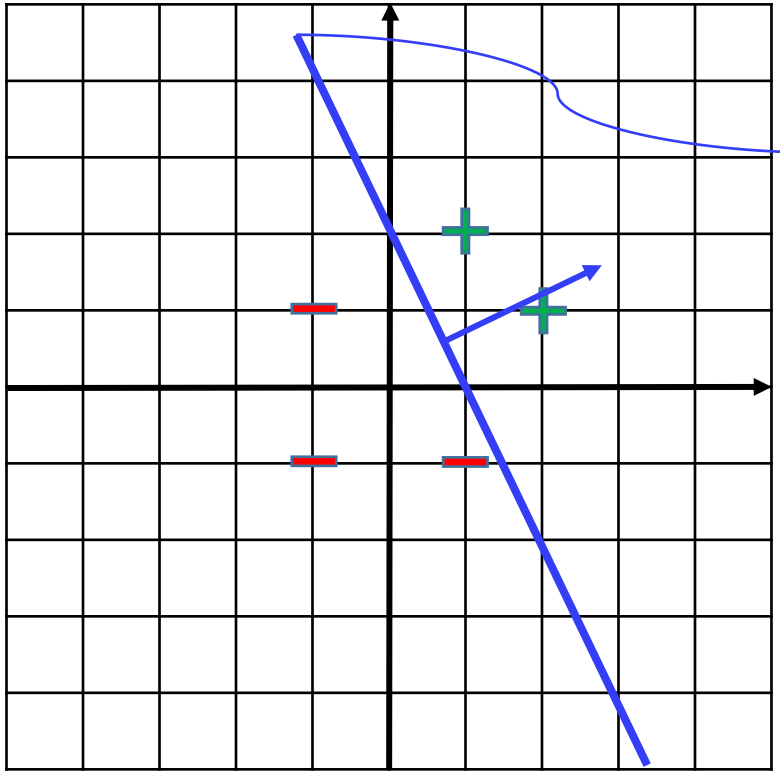
$$D=\left\{\begin{array}{c}\left(X^{(1)}=(1,2),y^{(1)}=+1\right),\\ \left(X^{(2)}=(2,1),y^{(1)}=+1\right),\\ \left(X^{(3)}=(1,-1),y^{(1)}=-1\right),\\ \left(X^{(4)}=(-1,-1),y^{(1)}=-1\right),\\ \left(X^{(5)}=(1,-1),y^{(1)}=-1\right)\end{array}\right\}$$

PLA

$$D=\left\{\begin{array}{c}\left(X^{(1)}=(1,2),y^{(1)}=+1\right),\\ \left(X^{(2)}=(2,1),y^{(1)}=+1\right),\\ \left(X^{(3)}=(1,-1),y^{(1)}=-1\right),\\ \left(X^{(4)}=(-1,-1),y^{(1)}=-1\right),\\ \left(X^{(5)}=(-1,1),y^{(1)}=-1\right)\end{array}\right\}$$

迭代次数	W	b	错误样本	参数更新
0	(0,0)	0	1,2,3,4,5	$W=W+(1,2),b=b+1$
1	(1,2)	1	3,4	$W=W-(1,-1),b=b-1$
2	(0,3)	0	5	$W=W-(-1,1),b=b-1$
3	(1,2)	-1	5	$W=W-(-1,1),b=b-1$
4	$(2,1)$	-2	无	无

PLA



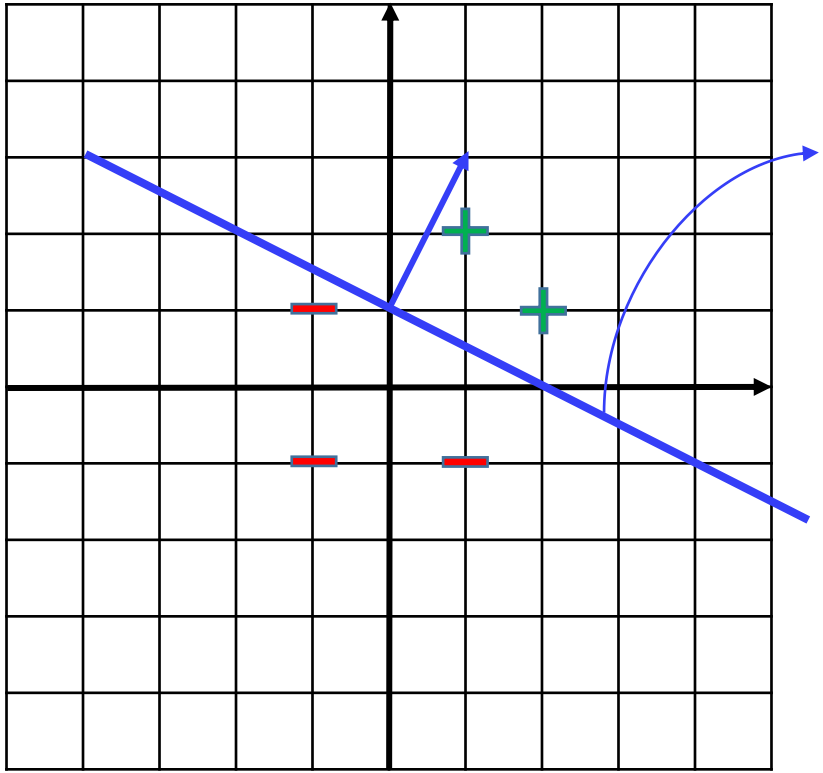
$$g(X;W,b) = 2x_1 + x_2 - 2 = 0$$

PLA

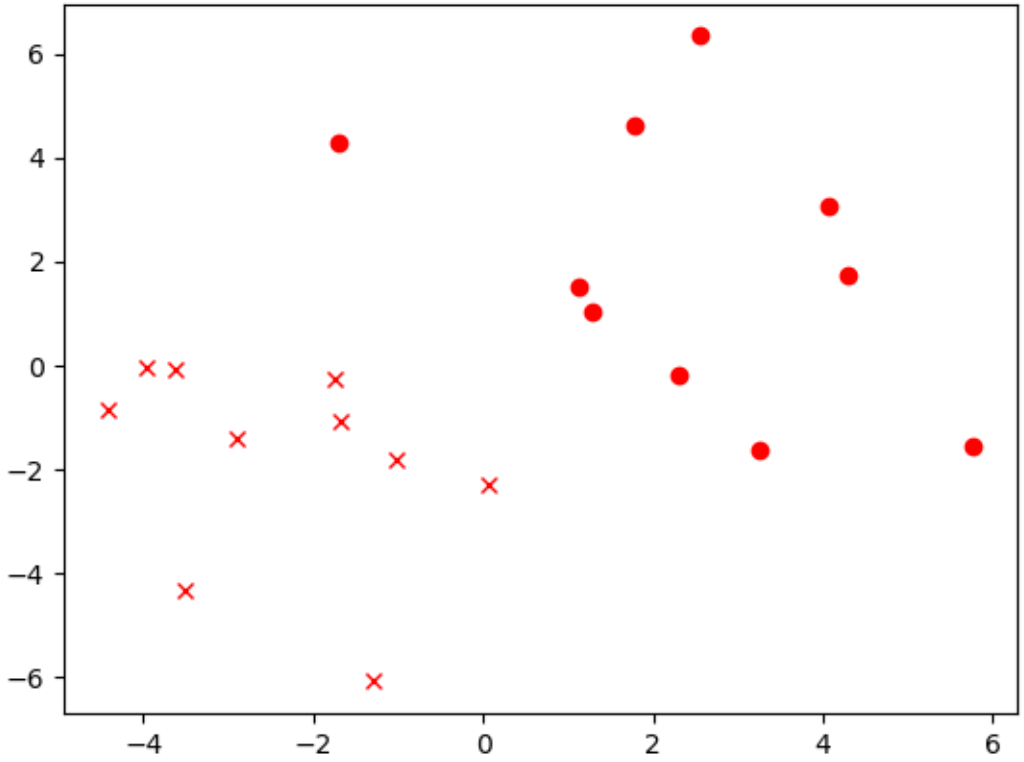
$$D=\left\{\begin{array}{c}\left(X^{(1)}=(1,2),y^{(1)}=+1\right),\\ \left(X^{(2)}=(2,1),y^{(1)}=+1\right),\\ \left(X^{(3)}=(1,-1),y^{(1)}=-1\right),\\ \left(X^{(4)}=(-1,-1),y^{(1)}=-1\right),\\ \left(X^{(5)}=(-1,1),y^{(1)}=-1\right)\end{array}\right\}$$

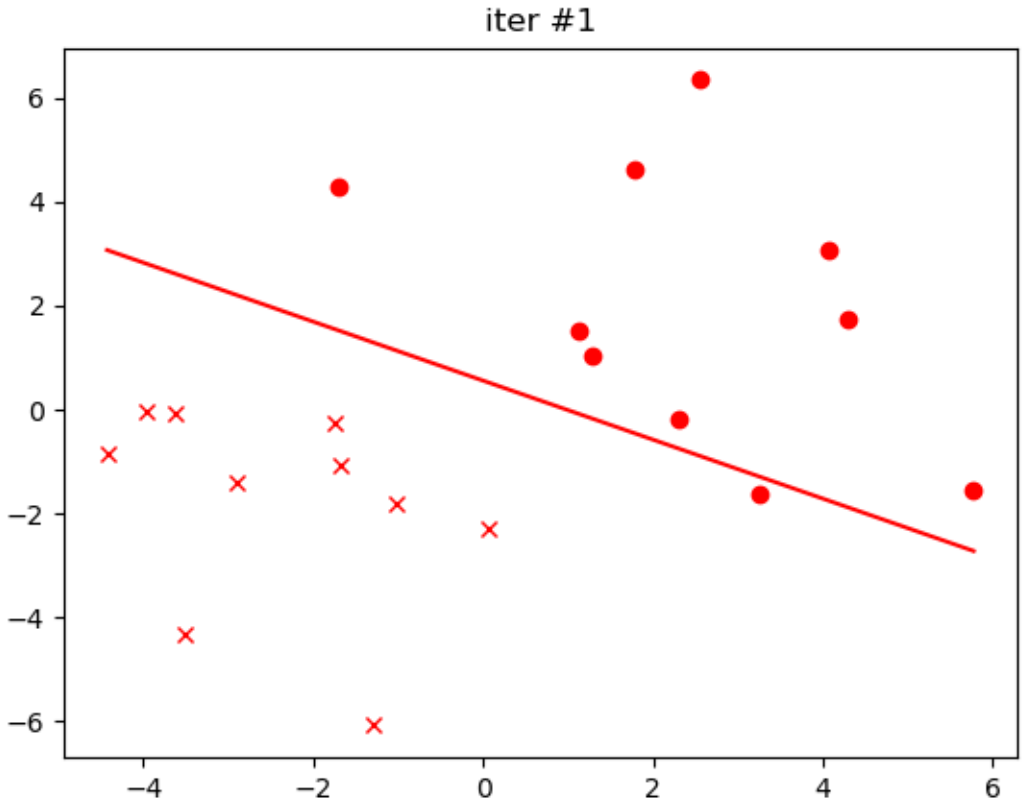
迭代次数	W	b	错误样本	参数更新
0	$(0,0)$	0	1,2,3,4,5	$W=W-(1,-1),b=b-1$
1	$(-1,1)$	-1	1,2,5	$W=W+(2,1),b=b+1$
2	$(1,2)$	0	5	$W=W-(-1,1),b=b-1$
3	$(2,1)$	-1	3	$W=W-(1,-1),b=b-1$
4	$(1,2)$	-2	无	无

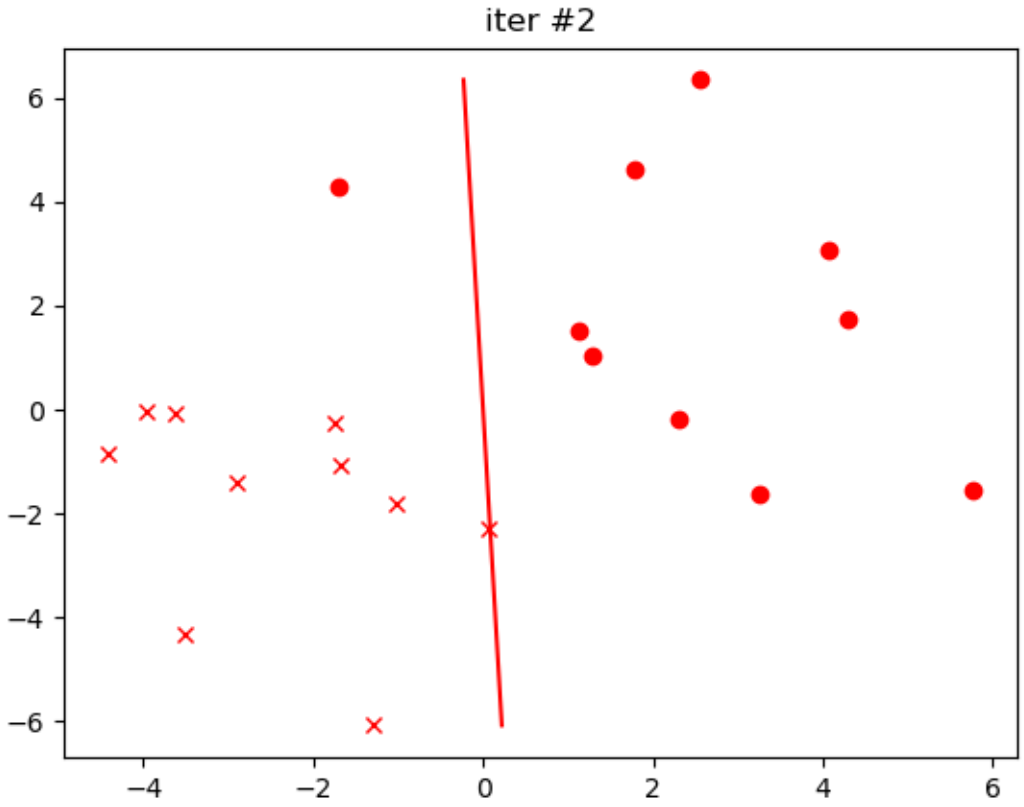
PLA

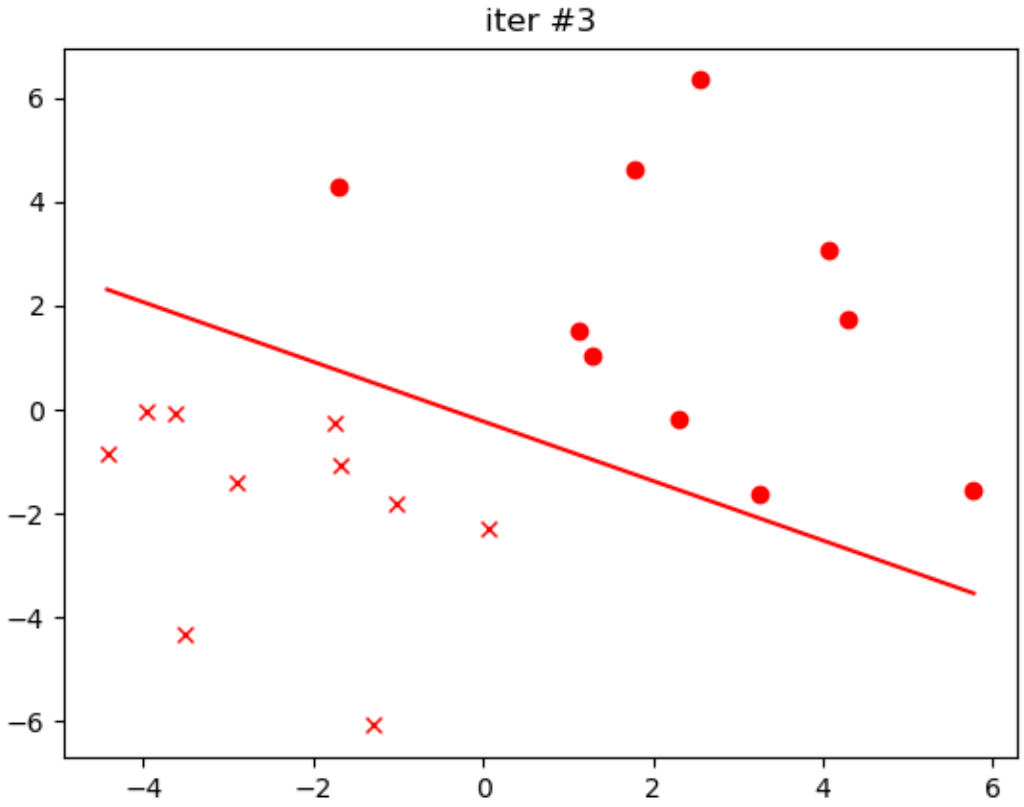


$$g(X;W,b) = x_1 + 2x_2 - 2 = 0$$


6
4
2
0
−2
−4
−6
−4
−2
0
2
4
6


iter #1
6
4
2
0
−2
−4
−6
−4
−2
0
2
4
6


iter #2
6
4
2
0
−2
−4
−6
−4
−2
0
2
4
6


iter #3
6
4
2
0
−2
−4
−6
−4
−2
0
2
4
6

PLA收敛定理

- 定理：假设样本集$\{X_i\}_{i=1}^n$线性可分，即存在一个$W^*: y_i {W^*}^T X_i > 0 \ \forall i$，并假设$\|W^*\| = 1$，那么PLA算法最多需要$\frac{R^2}{\gamma^2}$次更新即可收敛。其中，$R = \max_{i=1}^n \|X_i\|, \gamma = \min_{i=1}^n y_i {W^*}^T X_i$。

PLA收敛定理

- 初始值$W^{(1)} = \vec{0}$
- 第k次更新后权值为$W^{(k)}$，假设它在样本t上分类错误：$y_t {W^{(k)}}^T X_t < 0$
- 第k+1次更新后的权值为$W^{(k+1)} = W^{(k)} + y_t X_t$

PLA收敛定理

- $W^{(k+1)} = W^{(k)} + y_t X_t \Rightarrow$
- $\left\|W^{(k+1)}\right\|^2 = W^{(k+1)^T} W^{(k+1)} = \left(W^{(k)} + y_t X_t\right)^T \left(W^{(k)} + y_t X_t\right) = \left\|W^{(k)}\right\|^2 + 2y_t W^{(k)^T} X_t + y_t^2 \|X_t\|^2 < \left\|W^{(k)}\right\|^2 + \|X_t\|^2 \leq \left\|W^{(k)}\right\|^2 + R^2$
- $\Rightarrow \left\|W^{(k+1)}\right\|^2 \leq \left\|W^{(k)}\right\|^2 + R^2 \leq \left\|W^{(k-1)}\right\|^2 + 2R^2 \leq \cdots \leq kR^2$

PLA收敛定理

- $W^{(k+1)} = W^{(k)} + y_t X_t \Rightarrow$
- $W^{(k+1)^T} W^* = \left(W^{(k)} + y_t X_t\right)^T W^* = W^{(k)} W^* + y_t W^{*T} X_t \geq W^{(k)} W^* + \gamma$
- $\Rightarrow W^{(k+1)^T} W^* \geq W^{(k)} W^* + \gamma \geq \cdots \geq W^{(1)} W^* + k\gamma = k\gamma$
- $\Rightarrow k\gamma \leq W^{(k+1)^T} W^* \leq \left\|W^{(k+1)}\right\| \cdot \|W^*\| = \left\|W^{(k+1)}\right\|$

- $\left\|W^{(k+1)}\right\|^2 \leq kR^2, k\gamma \leq \left\|W^{(k+1)}\right\| \Rightarrow k^2\gamma^2 \leq kR^2 \Rightarrow k \leq \frac{R^2}{\gamma^2}$

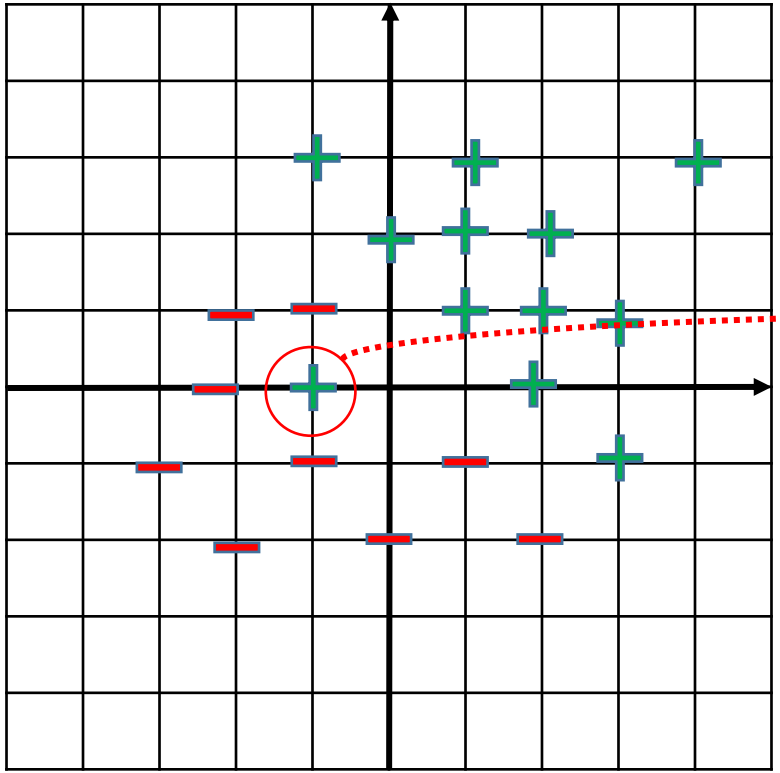
感知机学习算法

- 输入：训练样本$D = \left\{\left(X^{(i)} \in R^{d+1}, y^{(i)} \in \{-1, +1\}\right)\right\}_{i=1}^{n}$
- 输出：$f(X; W) = sign(W^T X)$
- 1. 初始化$W = \mathbf{0}$
- 2. While True:
  - 2.1 随机选取一个样本X, y，如果$yW^T X < 0$，该样本为错分样本，按2.2更新权重；如果不存在错分样本，退出循环；
  - 2.2 更新$W: W = W + \alpha yX$；
- 3. 返回W

要求样本集D线性可分，否则算法无法终止

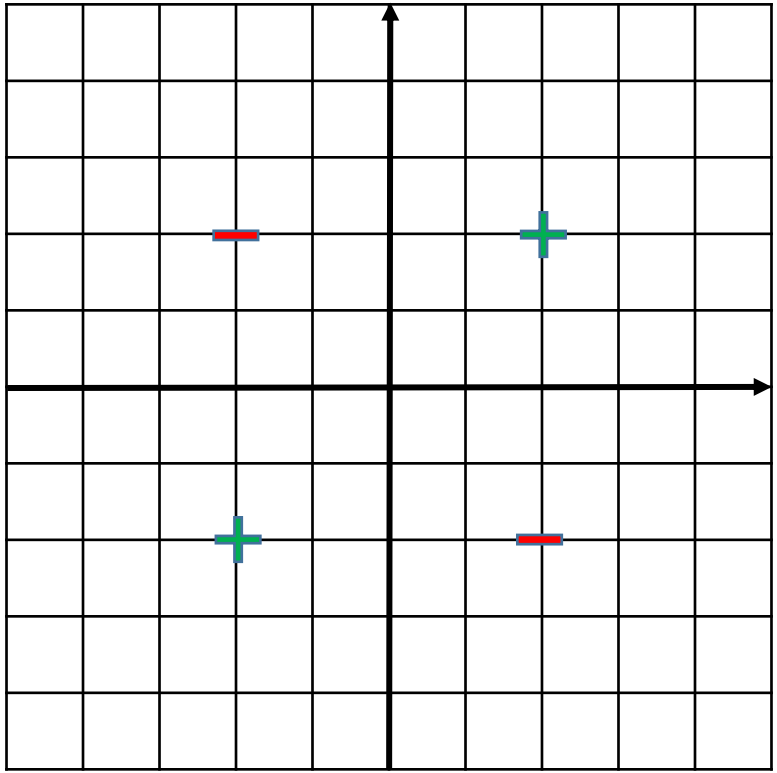
假设X是增广向量,$W = (W, b)^T$

非线性可分1:训练样本中存在噪声/错误样本



噪声/错误样本

非线性可分2:样本本质上是非线性的



$$D=\left\{\begin{array}{l}\left(X^{(1)}=(+2,+2), y^{(1)}=+1\right)\\ \left(X^{(2)}=(+2,-2), y^{(2)}=-1\right)\\ \left(X^{(3)}=(-2,-2), y^{(3)}=+1\right)\\ \left(X^{(4)}=(-2,+2), y^{(4)}=-1\right)\end{array}\right\}$$

异或问题，非线性可分

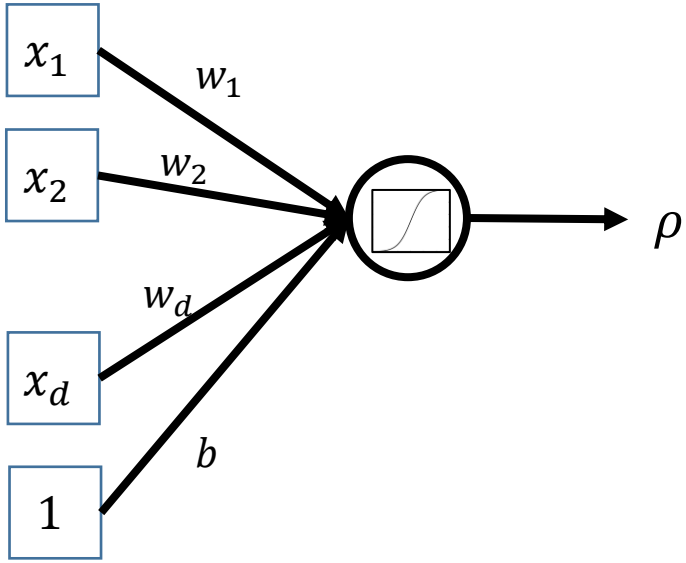
感知机学习算法

- 1. 在线性可分的情况下，一定可以收敛 
  - 可以找到一个能正确分类所有样本的分界面
- 2. 同一个样本集，有可能得到不同的解 
  - 不同初始值、不同的样本处理次序产生不同的结果
  - 不能得到全局最优解
- 3. 线性不可分情况，算法失败 
  - 不能处理线性不可分样本

对数几率回归

Logistic Regression

对率回归



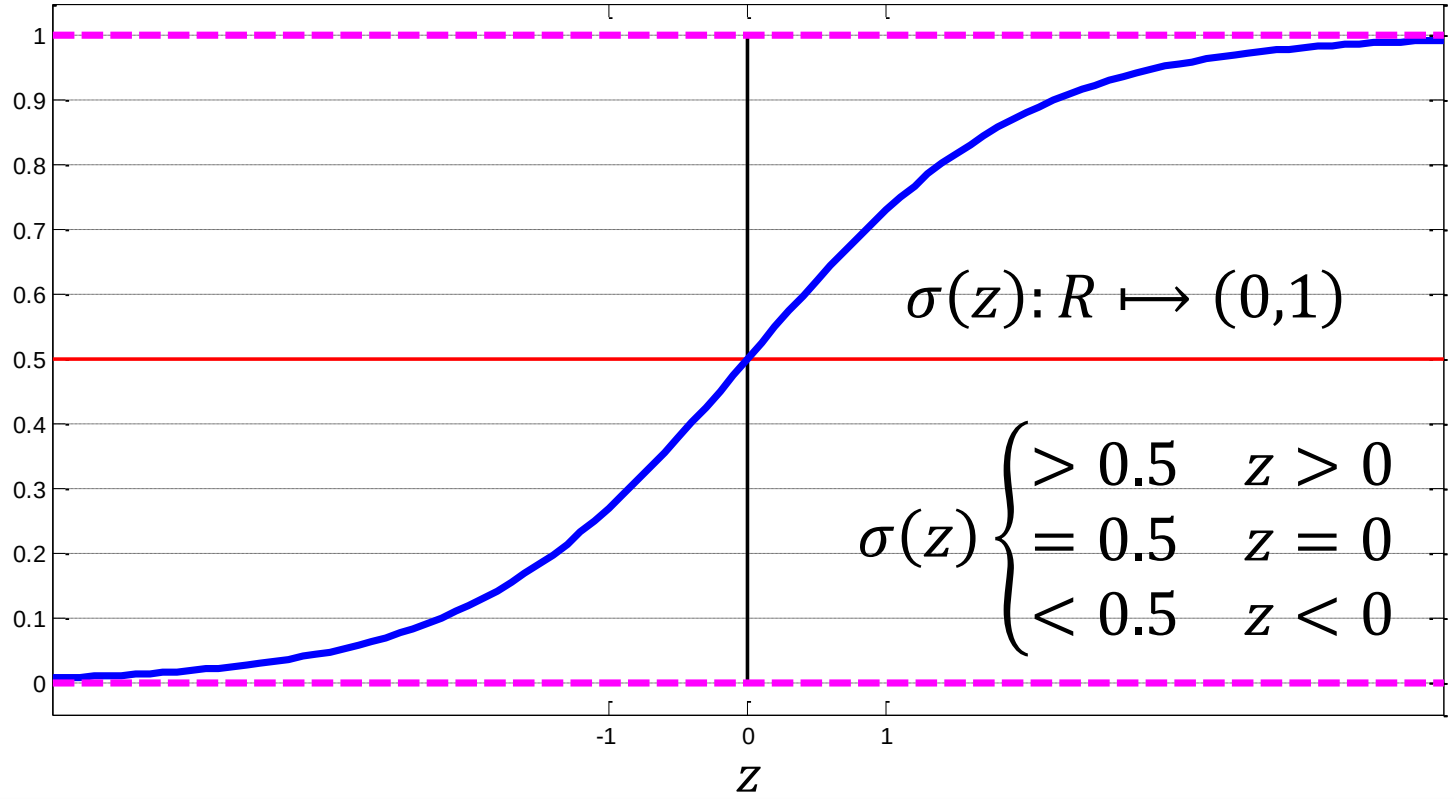
$$z = g(X) = \sum_{i=1}^{d} w_i x_i + b = W^T X + b,$$

$$\rho(X) = \sigma\big(g(X)\big)$$

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

$$X = (x_1, x_2, \dots, x_d)^T \in R^d$$

Sigmoid函数


1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
-1
0
1
z
$\sigma(z): R \longmapsto (0,1)$
$\sigma(z)\begin{cases} > 0.5 & z > 0 \\ = 0.5 & z = 0 \\ < 0.5 & z < 0 \end{cases}$

使用对率回归模型实现分类

$$\rho(X;W,b)=\sigma(W^TX+b)\in(0,1)$$

$$P(y=1|X)\equiv\rho(X;W,b),\qquad P(y=0|X)\equiv 1-\rho(X;W,b)$$

$$y=\begin{cases}1 & P(y=1|X)>P(y=0|X)\\ 0 & P(y=0|X)>P(y=1|X)\end{cases}\Rightarrow$$

$$y=\begin{cases}1 & P(y=1|X)>0.5\\ 0 & P(y=0|X)>0.5\end{cases}\Rightarrow y=\begin{cases}1 & W^TX+b>0\\ 0 & W^TX+b<0\end{cases}$$

使用对率回归模型实现分类

$$\rho(X;W,b)=\sigma(z)=\frac{1}{1+e^{-z}},z=W^TX+b$$

$$P(y=1|X)\equiv\rho(X;W,b)=\frac{1}{1+e^{-z}};P(y=0|X)\equiv1-\rho(X;W,b)=\frac{e^{-z}}{1+e^{-z}}$$

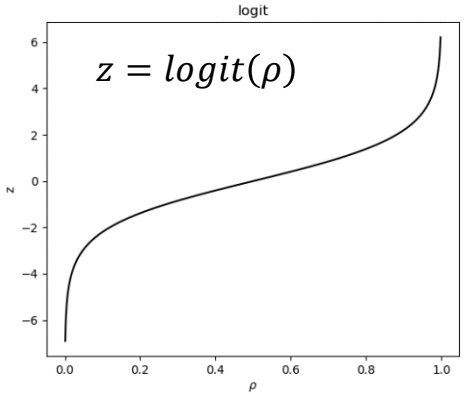
几率:$odds(y=1)=\frac{P(y=1|X)}{P(y=0|X)}=\frac{\rho}{1-\rho}=e^z$

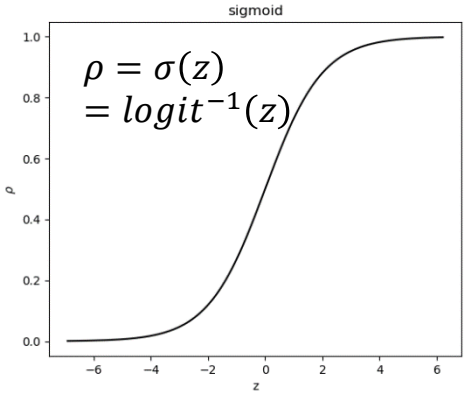
对数几率/对率:

$$logodds(y=1)=\log\big(odds(y=1)\big)$$

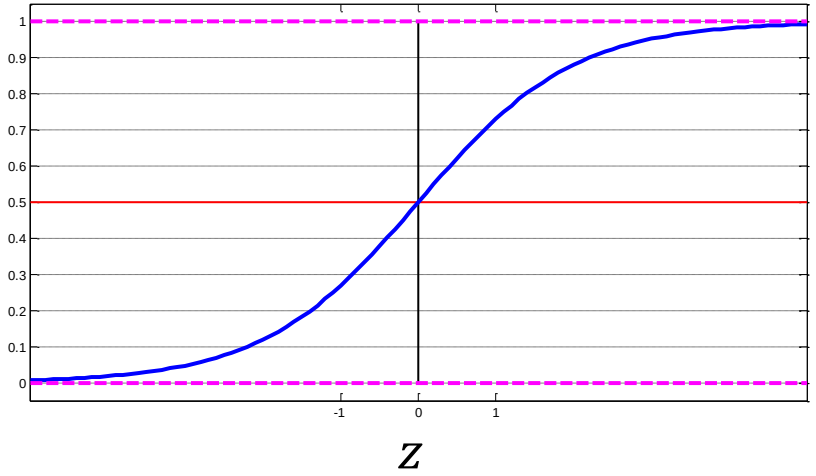
$$=log\frac{\rho}{1-\rho}=z=W^TX+b$$

$$z=logit(\rho)=logodds=log\frac{\rho}{1-\rho}$$





使用对率回归模型实现分类



$$z = g(X) = W^T X + b$$

$$\rho(X) = \sigma(z)$$

在分界面上:

$$z = g(X) = 0 \rightarrow \rho(X) = 0.5$$

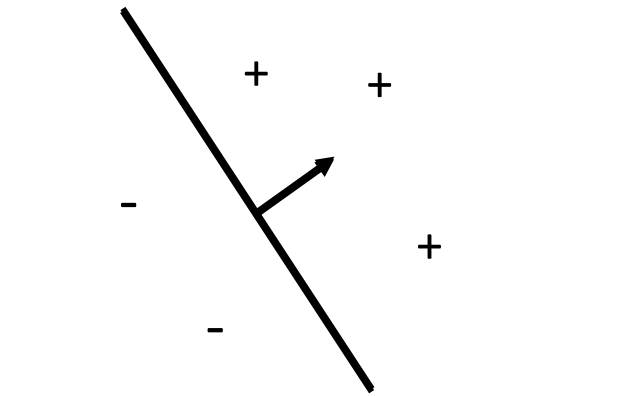
在分界面正向一侧:

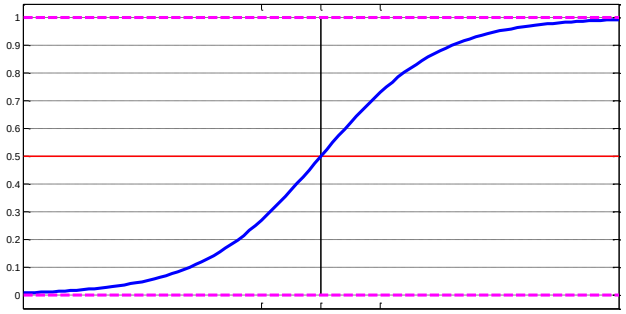
$$z = g(X) > 0 \rightarrow \rho(X) > 0.5$$

在分界面负向一侧:

$$z = g(X) < 0 \rightarrow \rho(X) < 0.5$$

使用对率回归模型实现分类





$z = g(X)$

给定参数W, b，样本X到分界面$g(X) = 0$的距离：

$$d(X) = \frac{|g(X)|}{\|W\|} \Rightarrow |g(X)| = c \cdot d(X)$$

当X位于分界面正向一侧：

$$g(X) > 0: \ g(X) = c \cdot d(X)$$

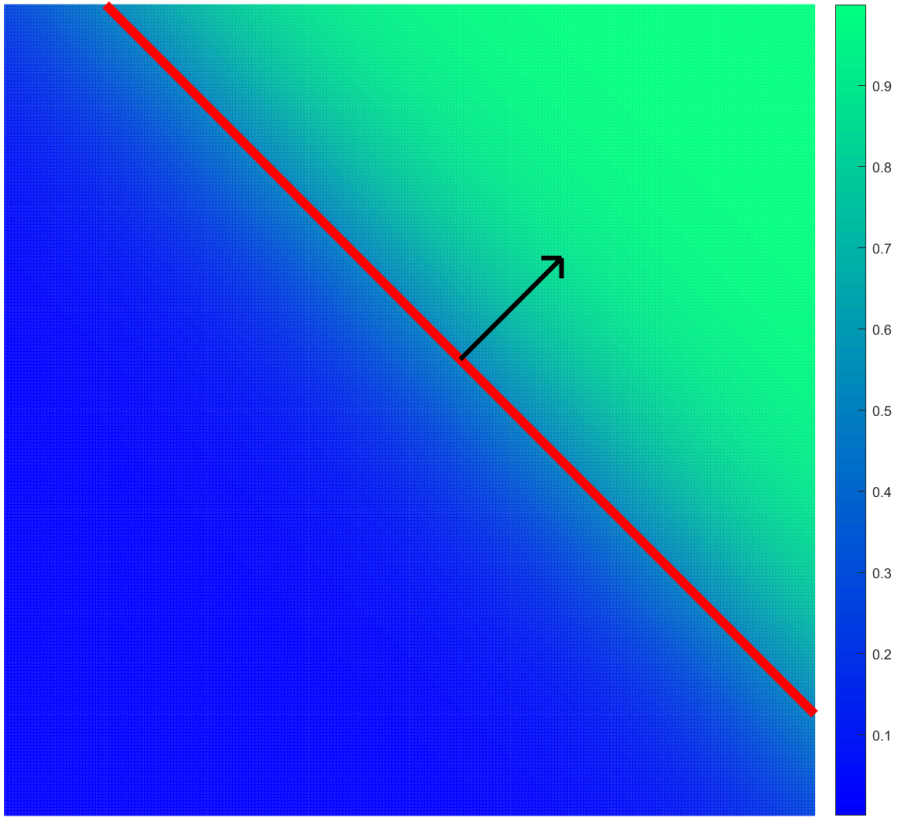
$$d(X) \uparrow \Rightarrow g(X) \uparrow \Rightarrow \rho(X) \uparrow$$

当X位于分界面负向一侧：

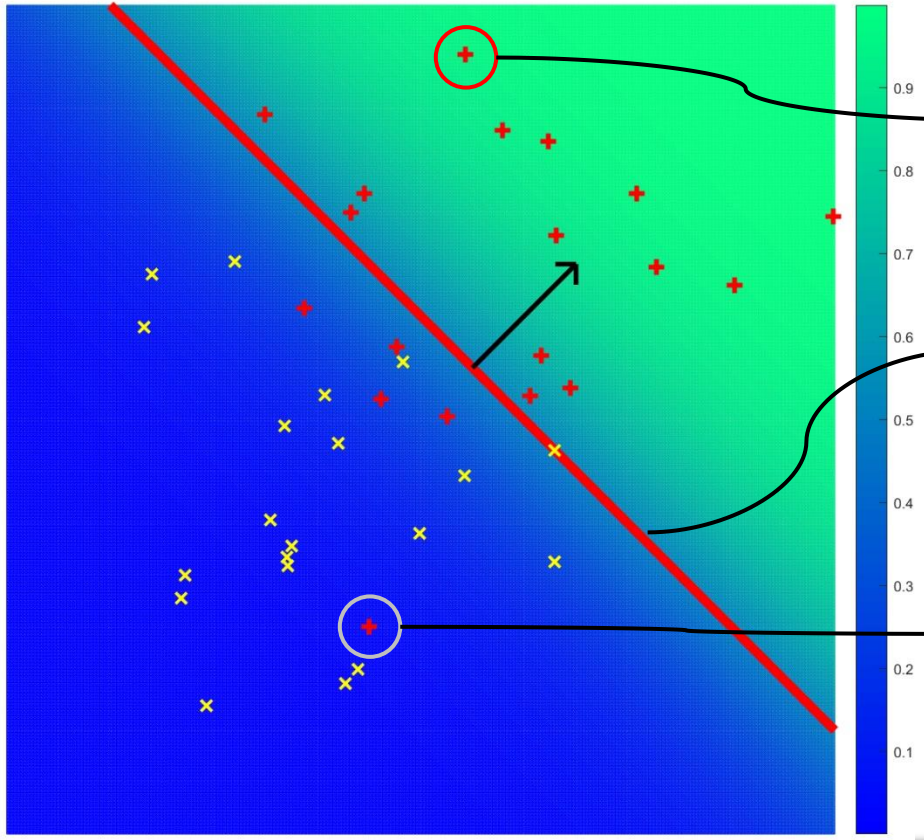
$$g(X) < 0: g(X) = -cd(X)$$

$$d(X) \uparrow \Rightarrow g(X) \downarrow \Rightarrow \rho(X) \downarrow$$

使用对率回归模型实现分类



使用对率回归模型实现分类



分类正确：

$$P(y=1|X)=\rho(X)\gg 0.5$$

$$g(X)=W^TX+b>0$$

$$yg(X)>0$$

$$g(X)=W^TX+b=0$$

分类错误：

$$P(y=1|X)=\rho(X)\ll 0.5$$

$$g(X)=W^TX+b<0$$

$$yg(X)<0$$

损失函数(loss function)

- 损失函数l是一个关于模型参数的实函数:$l(\Theta) \in R, l(\Theta) \geq 0$
- $l(\Theta)$用于评价参数Θ的"好坏"
  - 损失越小，参数越好
  - 损失越大，参数越差
- $l(\Theta)$通过比较模型对样本X的预测结果与样本的真实类别y之间的差异，计算Θ的损失
  - 差异越大，损失越大
  - 差异越小，损失越小
  - 无差异，无损失
- 学习算法通过找到损失函数的极小值点，确定"好的"参数Θ
  - $\Theta^* = argmin_\Theta l(\Theta)$

对率回归的损失函数(loss function)

- $(X, y = 1)$:
  - 预测X为正样本的概率为$P(y = 1|X; W, b) = \rho(X)$
  - $\rho(X)$越高越好➔
  - $\rho(X)$越低,W, b在该样本上的损失l越大
  - $\rho(X)$越高,W, b在该样本上的损失l越小，当$\rho(X) \approx 1$时，$l \approx 0$
- $(X, y = 0)$:
  - 预测X为负样本的概率为$P(y = 0|X; W, b) = 1 - \rho(X)$
  - $1 - \rho(X)$越高越好➔
  - $1 - \rho(X)$越低,W, b在该样本上的损失l越大
  - $1 - \rho(X)$越高,W, b在该样本上的损失l越小,当$1 - \rho(X) \approx 1$时，$l \approx 0$

对数损失函数(Log-loss)
交叉熵损失函数(Cross-Entropy Loss)

- W, b在样本(X, y)上的损失$l(W, b; X, y) \geq 0$
- $(X, y = 1)$:
  - $l(W, b; X, y = 1) = -ln\rho(X) \geq 0$
  - $P(y = 1|X; W, b) = \rho(X) \approx 1 \Rightarrow l(W, b; X, y = 1) \approx 0$
- $(X, y = 0)$:
  - $l(W, b; X, y = 0) = -\ln\big(1 - \rho(X)\big) \geq 0$
  - $P(y = 1|X; W, b) = \rho(X) \approx 0 \Rightarrow l(W, b; X, y = 0) \approx 0$

对数损失函数(Log-loss)
交叉熵损失函数(Cross-Entropy Loss)

- 在一个样本上的损失
  - $l(W,b;X,y) = -y ln\rho(X) - (1-y)\ln\big(1-\rho(X)\big) =$

$$\begin{cases} -ln\rho(X) & y=1 \\ -\ln\big(1-\rho(X)\big) & y=0 \end{cases}$$

- 在整个训练集D上的损失
  - $l(W,b;D) = \frac{1}{n}\sum_{i=1}^{n} l\big(W,b;X^{(i)},y^{(i)}\big)$

梯度下降法

训练对率回归模型:极小化对数损失函数

$$D = \left\{\left(X^{(i)} \in R^d, y^{(i)} \in \{1,0\}\right)\right\}_{i=1}^{n}$$

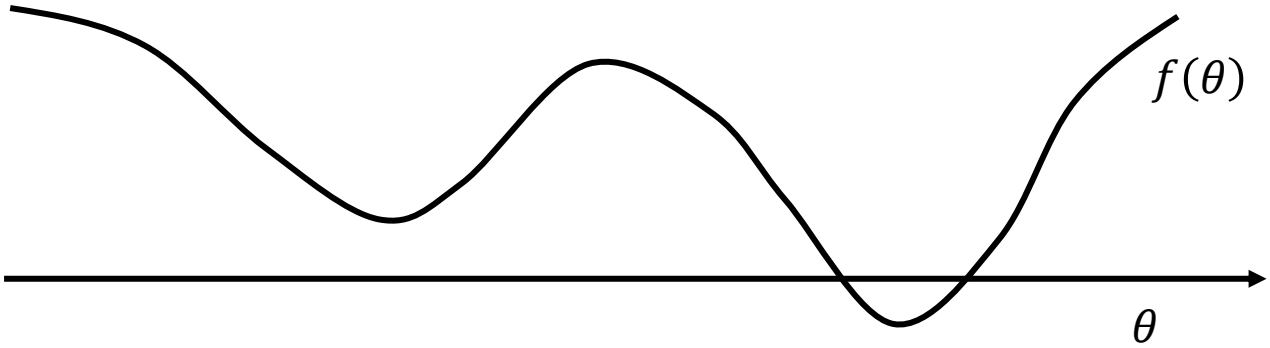
$$l(W, b; D) = \frac{1}{n}\sum_{i=1}^{n} l\left(W, b; X^{(i)}, y^{(i)}\right)$$

$$l(W, b; X, y) = -y ln\rho(X) - (1-y)\, ln\left(1-\rho(X)\right)$$

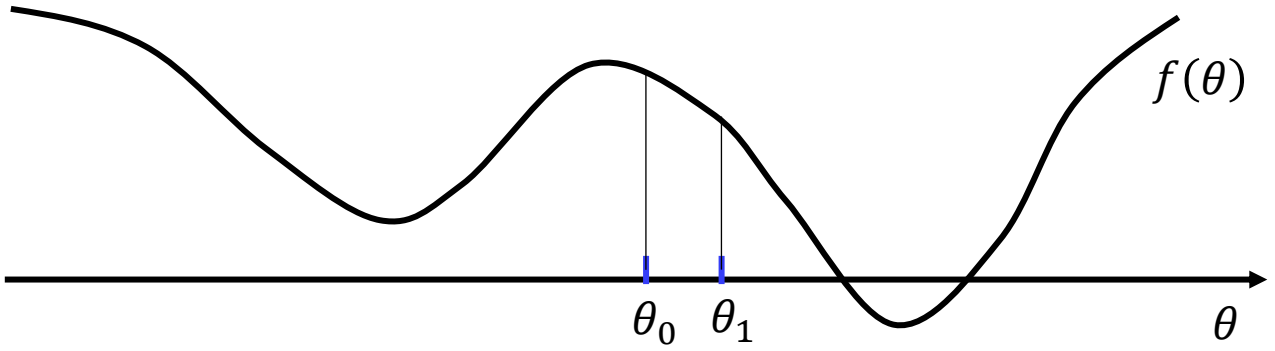
$$W^*, b^* = argmin_{W,b} l(W, b; D)$$

函数极小化问题

- $f(\theta)$是一个关于θ的连续函数
- 求$f(\theta)$的极小值点θ^*
  - $\exists \epsilon > 0: f(\theta^*) < f(\theta), \forall \theta \in (\theta^* - \epsilon, \theta^* + \epsilon)$

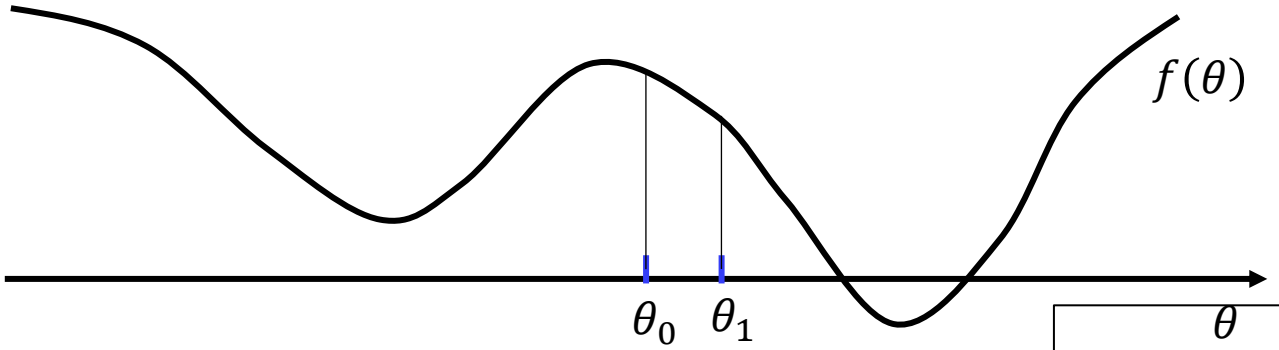


梯度下降法



- 0：猜测一个初始值θ_0
- 1：如果θ_0是局部极小值，退出
- 2：更新θ_0为θ_1，使得$f(\theta_1) < f(\theta_0)$
- 3：$\theta_0 = \theta_1$，转1

梯度下降法



$f(\theta)$在θ_0处的梯度$\dfrac{df}{d\theta_0}$

➢ 2：更新θ_0为θ_1，使得$f(\theta_1) < f(\theta_0)$

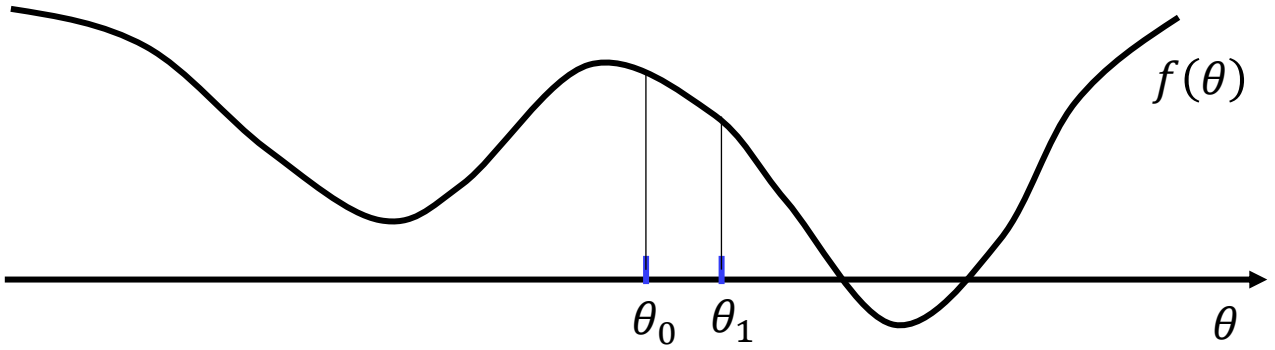
在θ_0的一个很小的局部邻域搜索θ_1: $\theta_1 = \theta_0 + \delta$，要求$f(\theta_1) < f(\theta_0)$

在θ_0的附近对$f(\theta)$做一阶Taylor展开：$f(\theta_1) = f(\theta_0 + \delta) \approx f(\theta_0) + \delta \cdot f'(\theta_0)$

$f(\theta_1) < f(\theta_0) \Rightarrow \delta f'(\theta_0) < 0 \Rightarrow \delta = ?$　　$\delta = -\alpha f'(\theta_0) \Rightarrow \delta f'(\theta_0) = -\alpha[f'(\theta_0)]^2 < 0, \alpha > 0$

$\theta_1 = \theta_0 + \delta = \theta_0 - \alpha f'(\theta_0)$　梯度下降法，$\alpha > 0$：学习速率

梯度下降法



- 0：猜测一个初始值θ_0
- 1：如果θ_0是局部极小值，退出
- 2：更新θ_0为θ_1:$\theta_1 = \theta_0 - \alpha \cdot \frac{df}{d\theta_0}$
- 3：$\theta_0 = \theta_1$，转1

用梯度下降法极小化对数损失函数

$$l(W,b;D)=\frac{1}{n}\sum_{i=1}^{n}l\left(W,b;X^{(i)},y^{(i)}\right)$$

$$l(W,b;X,y)=-ylnp(X)-(1-y)\ln\left(1-p(X)\right)$$

$$W^{*},b^{*}=argmin_{W,b}l(W,b;D)$$

$$\frac{\partial l(W,b;D)}{\partial W}=\frac{1}{n}\sum_{i=1}^{n}\frac{\partial l\left(W,b;X^{(i)},y^{(i)}\right)}{\partial W}$$

$$\frac{\partial l(W,b;D)}{\partial b}=\frac{1}{n}\sum_{i=1}^{n}\frac{\partial l\left(W,b;X^{(i)},y^{(i)}\right)}{\partial b}$$

用梯度下降法极小化对数损失函数

$$l(W,b;X,y)=-ylnp(X)-(1-y)\ln\big(1-\rho(X)\big),\rho(X)=\frac{1}{1+e^{-g(X)}},g(X)=W^TX+b$$

$$\frac{\partial l(W,b;X,y)}{\partial W}=\frac{\partial l}{\partial \rho}\times\frac{\partial \rho}{\partial g}\times\frac{\partial g}{\partial W},\qquad \frac{\partial l(W,b;X,y)}{\partial b}=\frac{\partial l}{\partial \rho}\times\frac{\partial \rho}{\partial g}\times\frac{\partial g}{\partial b}$$

$$\begin{cases}\dfrac{\partial l}{\partial \rho}=-\dfrac{y}{\rho}+\dfrac{1-y}{1-\rho}=\dfrac{-y+y\rho+\rho-y\rho}{\rho(1-\rho)}=\dfrac{\rho-y}{\rho(1-\rho)}\\ \dfrac{\partial \rho}{\partial g}=\dfrac{e^{-g}}{(1+e^{-g})^2}=\dfrac{1}{1+e^{-g}}\times\dfrac{e^{-g}}{1+e^{-g}}=\rho(1-\rho)\end{cases}\Rightarrow\frac{\partial l}{\partial g}=\frac{\partial l}{\partial a}\times\frac{\partial a}{\partial g}=\rho-y$$

$$\frac{\partial g}{\partial W}=X,\frac{\partial g}{\partial b}=1\Rightarrow\frac{\partial l}{\partial W}=(\rho-y)X,\frac{\partial l}{\partial b}=(\rho-y)$$

用梯度下降法极小化对数损失函数

$$\frac{\partial l}{\partial W} = (\rho - y)X, \frac{\partial l}{\partial b} = (\rho - y)$$

$$\rho = P(y = 1|X; W, b) \in (0,1)$$

$e \equiv \rho - y$ ← 误差，error

$\forall X$:

如果$y = 1$：希望预测的概率值$\rho \to 1 \Rightarrow e = \rho - 1 \to 0$

如果$y = 0$：希望预测的概率值$\rho \to 0 \Rightarrow e = \rho - 0 \to 0$

用梯度下降法极小化对数损失函数

$$\frac{\partial l(W,b;X^{(i)},y^{(i)})}{\partial W}=(\rho^{(i)}-y^{(i)})X^{(i)}=e^{(i)}X^{(i)}$$

$$\frac{\partial l(W,b;X^{(i)},y^{(i)})}{\partial W}=(\rho^{(i)}-y^{(i)})=e^{(i)}$$

$$e^{(i)}=\rho^{(i)}-y^{(i)}$$

$$\Delta W=\frac{\partial l(W,b;D)}{\partial W}=\frac{1}{n}\sum_{i=1}^{n}e^{(i)}X^{(i)}$$

$$\Delta b=\frac{\partial l(W,b;D)}{\partial b}=\frac{1}{n}\sum_{i=1}^{n}e^{(i)}$$

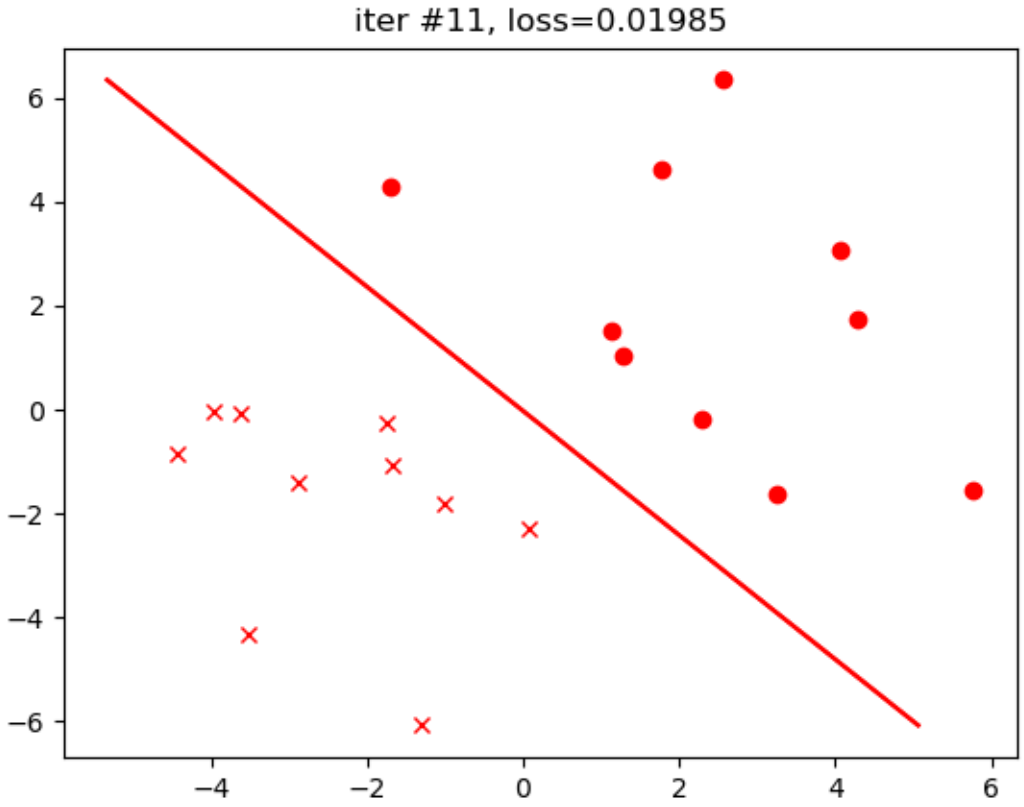
用梯度下降法极小化对数损失函数

$$W^{new} = W^{old} - \alpha \times \frac{\partial l(W, b; D)}{\partial W^{old}} = W^{old} - \alpha \times \frac{1}{n} \sum_{i=1}^{n} e^{(i)} X^{(i)}$$

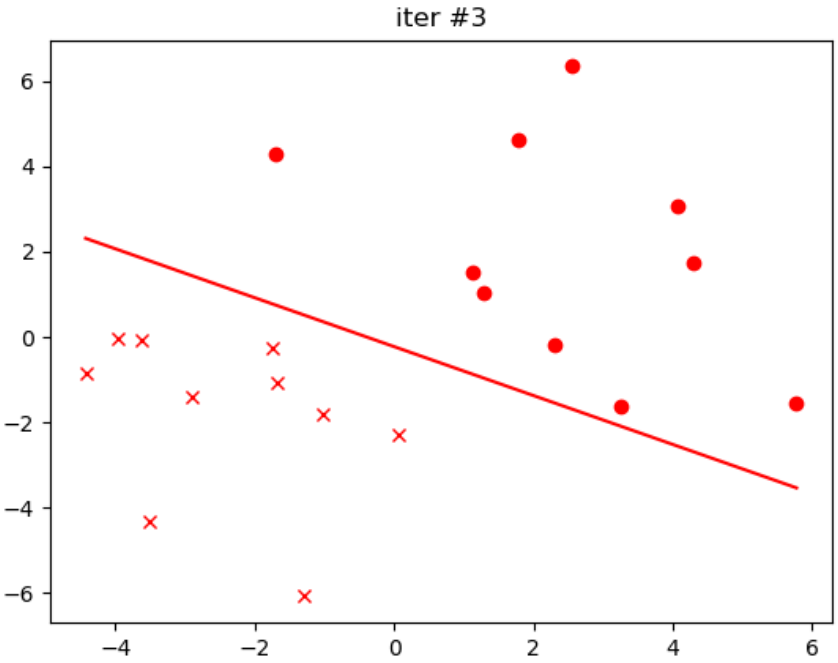
$$b^{new} = b^{old} - \alpha \times \frac{\partial l(W, b; D)}{\partial b^{old}} = b^{old} - \alpha \times \frac{1}{n} \sum_{i=1}^{n} e^{(i)}$$

用梯度下降法训练对率回归模型

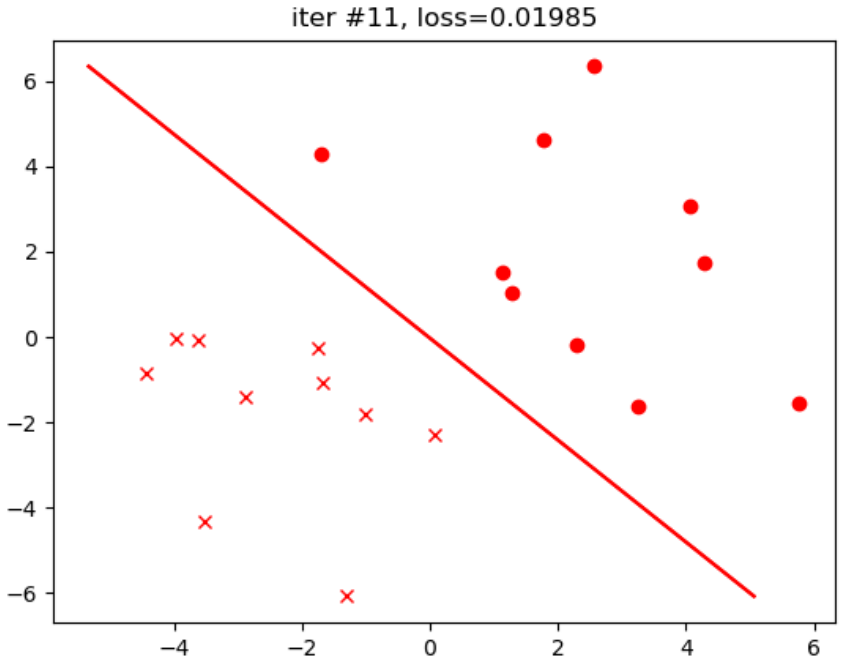
- 输入：训练样本$D=\{(X^{(i)}\in R^d, y^{(i)}\in\{0,1\})\}_{i=1}^{n}$，学习速率$\alpha$，收敛条件$\epsilon$
- 输出：W, b
- 1. 初始化$W=\mathbf{0}, b=0, l_0=Inf$
- 2. While True:
  - 2.1 $for\ i=1{:}n\quad \rho^{(i)}=\sigma(W^TX^{(i)}+b)$
  - 2.2 $l_1=\frac{1}{n}\sum_{i=1}^{n}[-y^{(i)}ln\rho^{(i)}-(1-y^{(i)})\ln(1-\rho^{(i)})], if\ |l_1-l_0|<\epsilon, break; else\ l_0=l_1$
  - 2.3 $for\ i=1{:}n\quad e^{(i)}=\rho^{(i)}-y^{(i)}$
  - 2.4 $\Delta W=\frac{1}{n}\sum_{i=1}^{n}e^{(i)}X^{(i)}, \Delta b=\frac{1}{n}\sum_{i=1}^{n}e^{(i)}$
  - 2.5 $W=W-\alpha\Delta W, b=b-\alpha\Delta b;$
- 3. 返回W, b


iter #11, loss=0.01985
6
4
2
0
−2
−4
−6
−4
−2
0
2
4
6

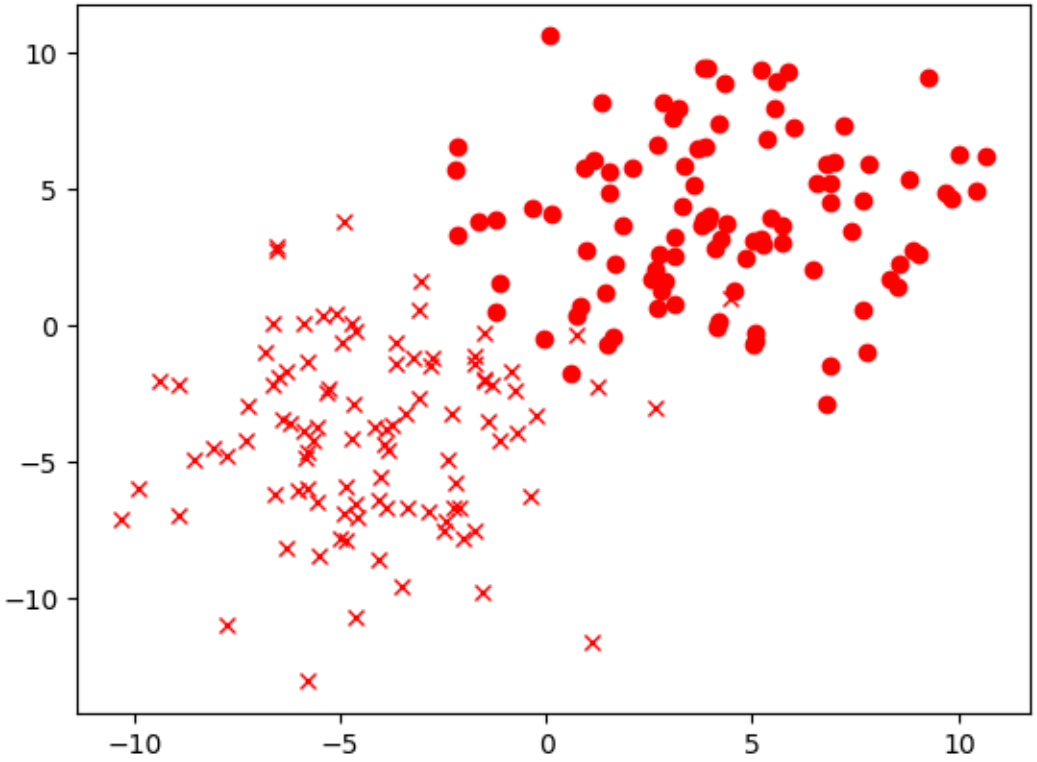
Perceptron-PLA



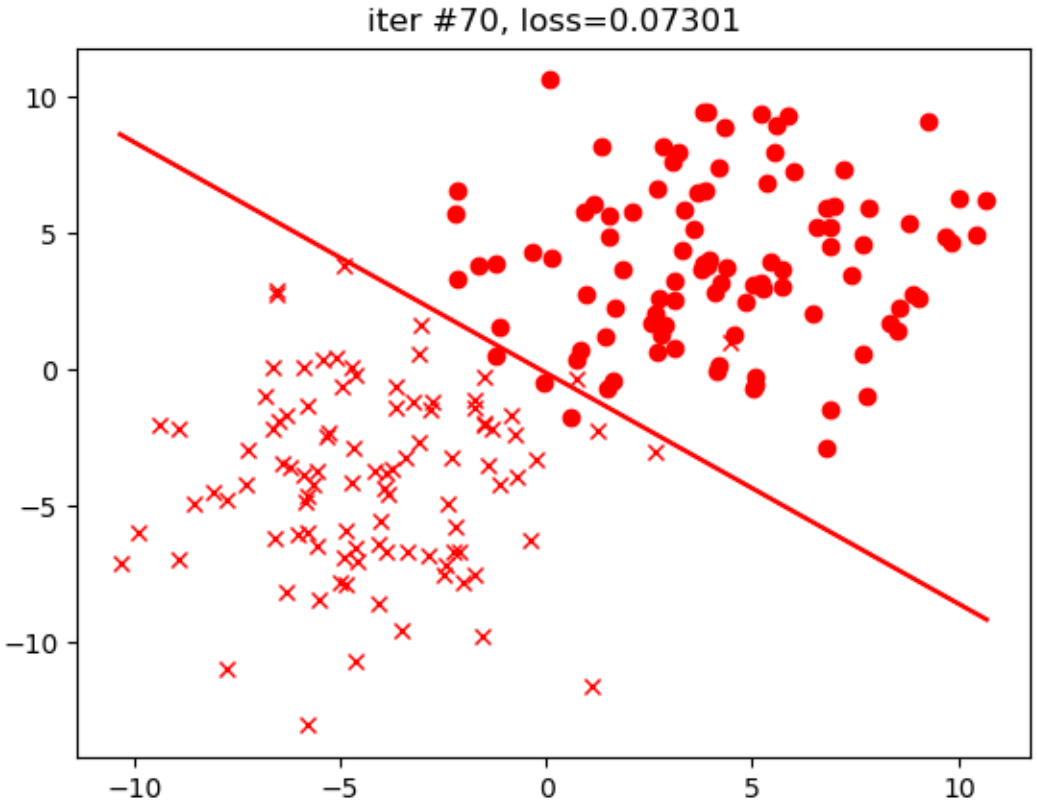
LogisticRegression-GD



噪声引起的线性不可分情况

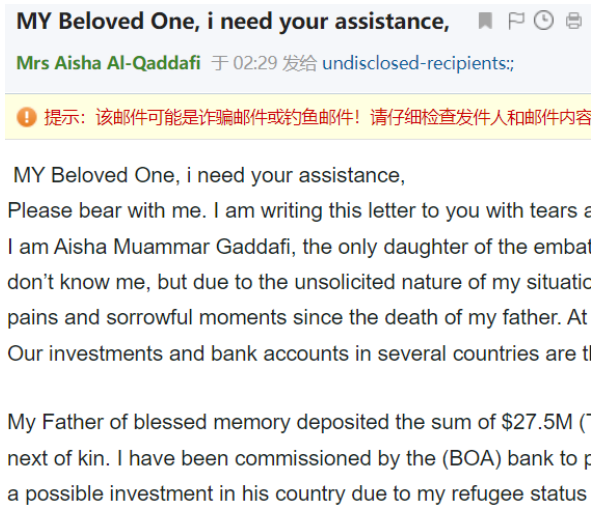


噪声引起的线性不可分情况



线性神经元的应用

垃圾邮件过滤(Spam Filtering)



MY Beloved One, i need your assistance,

Mrs Aisha Al-Qaddafi 于 02:29 发给 undisclosed-recipients:;

提示：该邮件可能是诈骗邮件或钓鱼邮件！请仔细检查发件人和邮件内容

MY Beloved One, i need your assistance,
Please bear with me. I am writing this letter to you with tears a
I am Aisha Muammar Gaddafi, the only daughter of the embat
don't know me, but due to the unsolicited nature of my situatic
pains and sorrowful moments since the death of my father. At
Our investments and bank accounts in several countries are tl

My Father of blessed memory deposited the sum of $27.5M (
next of kin. I have been commissioned by the (BOA) bank to p
a possible investment in his country due to my refugee status

Spam

Congratulations! Your article has been published in the Early Access area on IEEE Xplore

wmsprod 于 2021-11-20 00:36 发给 wenzhong、wenzhong

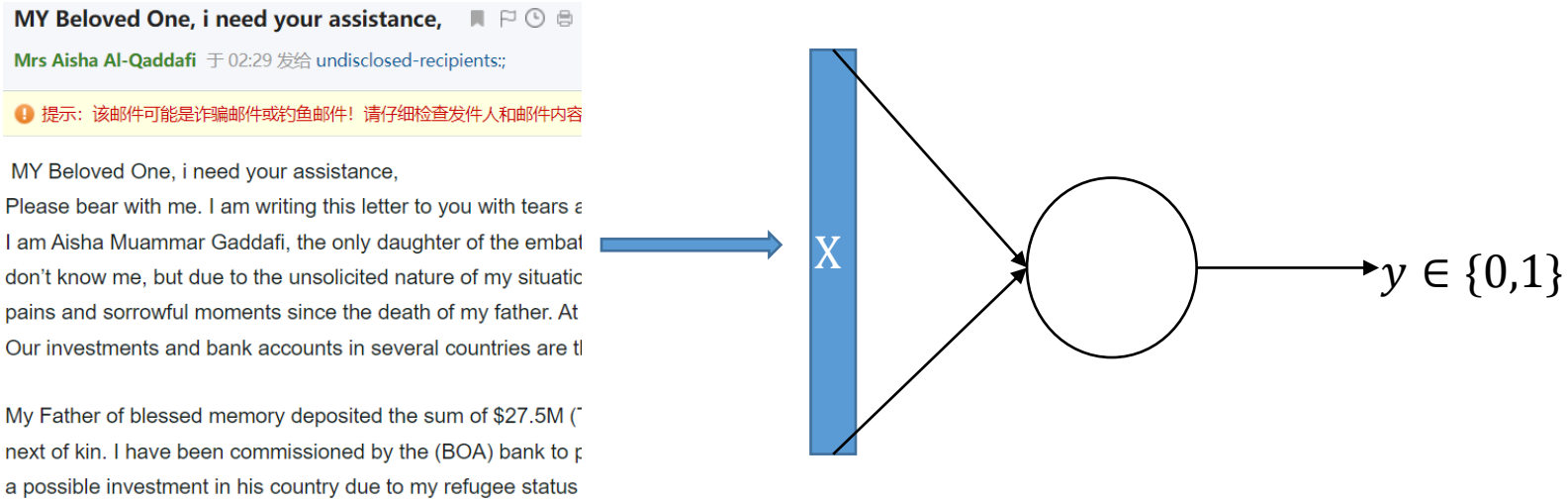
Dear Dr. Wenzhong Wang

This is to notify you that the following article, "Remote Sensing Scene Classification v
under the "Early Access" area on IEEE Xplore. This article has been accepted for publica
been edited and content may change prior to final publication. It may be cited as an art
Identifier. To view this article, please visit the journal homepage (listed below) on IE

This paper appears in: IEEE Transactions on Image Processing
On page(s): 1-11
Print ISSN: 1057-7149
Online ISSN: 1941-0042
Digital Object Identifier: 10.1109/TIP.2021.3127851

Ham

如何用向量表示一个文档?



词袋模型(Bag of Words,BoW)

D: 我喜欢计算机，我也喜欢程序设计。

词典：

$w_1 =$ 我, $w_2 =$ 喜欢, $w_3 =$ 课程, $w_4 =$ 学习,
$w_5 =$ 机器, $w_6 =$ 计算机, $w_7 =$ 视觉, $w_8 =$ 程序,
$w_9 =$ 设计

$D = \{w_1, w_2, w_6, w_8, w_9\}$ ← D:单词集合

$D = (w_1, w_2, w_6, w_1, w_2, w_8, w_9)$ ← D:单词序列

单词的独热向量(One-hot Vector)表示

词典：我，喜欢，课程，学习，机器，计算机，视觉，程序，设计

$w_1 =$ 我 $v_1 = [1,0,0,0,0,0,0,0,0]$
$w_2 =$ 喜欢 $v_2 = [0,1,0,0,0,0,0,0,0]$
$w_3 =$ 课程 $v_3 = [0,0,1,0,0,0,0,0,0]$
$w_4 =$ 学习 $v_4 = [0,0,0,1,0,0,0,0,0]$
$w_5 =$ 机器 $v_5 = [0,0,0,0,1,0,0,0,0]$
$w_6 =$ 计算机 $v_6 = [0,0,0,0,0,1,0,0,0]$
$w_7 =$ 视觉 $v_7 = [0,0,0,0,0,0,1,0,0]$
$w_8 =$ 程序 $v_8 = [0,0,0,0,0,0,0,1,0]$
$w_9 =$ 设计 $v_9 = [0,0,0,0,0,0,0,0,1]$

D1: 我喜欢计算机，我也喜欢程序设计。

$D1 = \{w_1, w_2, w_6, w_8, w_9\}$ $BoW(D1) = v_1 + v_2 + v_6 + v_8 + v_9 = [1,1,0,0,0,1,0,1,1]$

词袋模型(Bag of Words)

D1: 我喜欢计算机，我也喜欢程序设计。 → $[1, 1, 0, 0, 0, 1, 0, 1, 1]$

$D1 = \{w_1, w_2, w_6, w_8, w_9\}$

$BoW(D1) = v_1 + v_2 + v_6 + v_8 + v_9 = [1, 1, 0, 0, 0, 1, 0, 1, 1]$

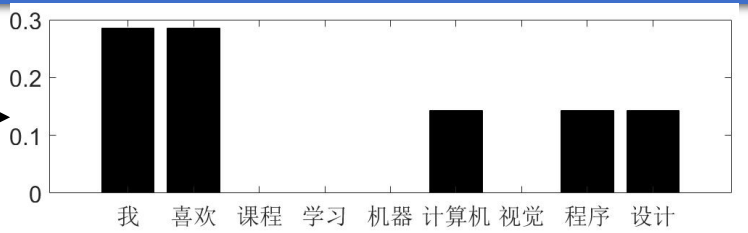
D2: 我喜欢计算机视觉课程。 → $[1, 1, 1, 0, 0, 1, 1, 0, 0]$

D3: 我喜欢机器学习课程。 → $[1, 1, 1, 1, 1, 0, 0, 0, 0]$

BoW: 每一篇文章看作是单词的无序集合。一篇文章通过BoW模型转换为一个向量（向量维度=词典大小），向量的不同维度表示不同单词在该文章中出现与否 (0,1)。

词袋模型(Bag of Words)

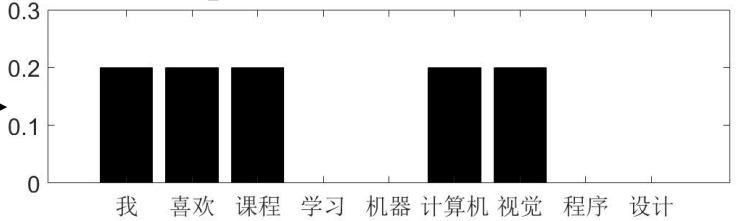
D1: 我喜欢计算机，我也喜欢程序设计。



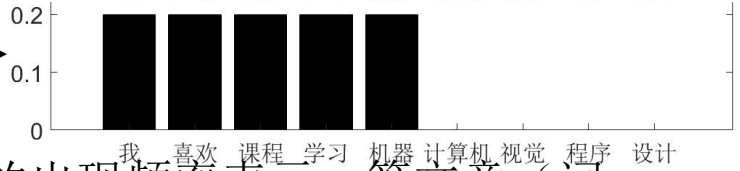
$$D1 = (w_1, w_2, w_6, w_1, w_2, w_8, w_9)$$

$$BoW(D1) = \frac{v_1 + v_2 + v_6 + v_1 + v_2 + v_8 + v_9}{|D1|} = \left[\frac{2}{7}, \frac{2}{7}, 0, 0, 0, \frac{1}{7}, 0, \frac{1}{7}, \frac{1}{7}\right]$$

D2: 我喜欢计算机视觉课程。

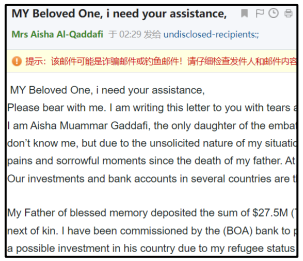


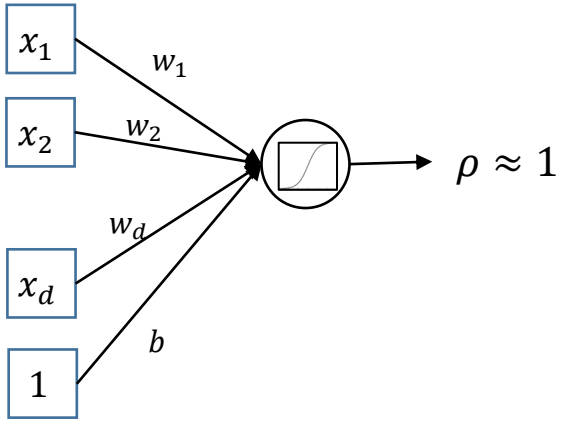
D3: 我喜欢机器学习课程。

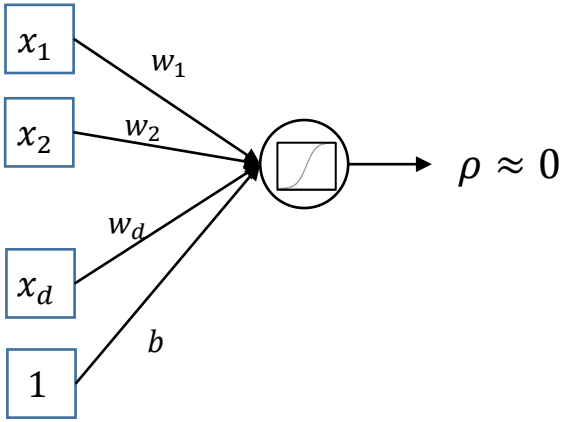


BoW: 每一篇文章看作是单词的序列，用不同单词的出现频率表示一篇文章（词频向量）。向量的不同维度表示不同单词在该文章中的出现频率(0~1)。

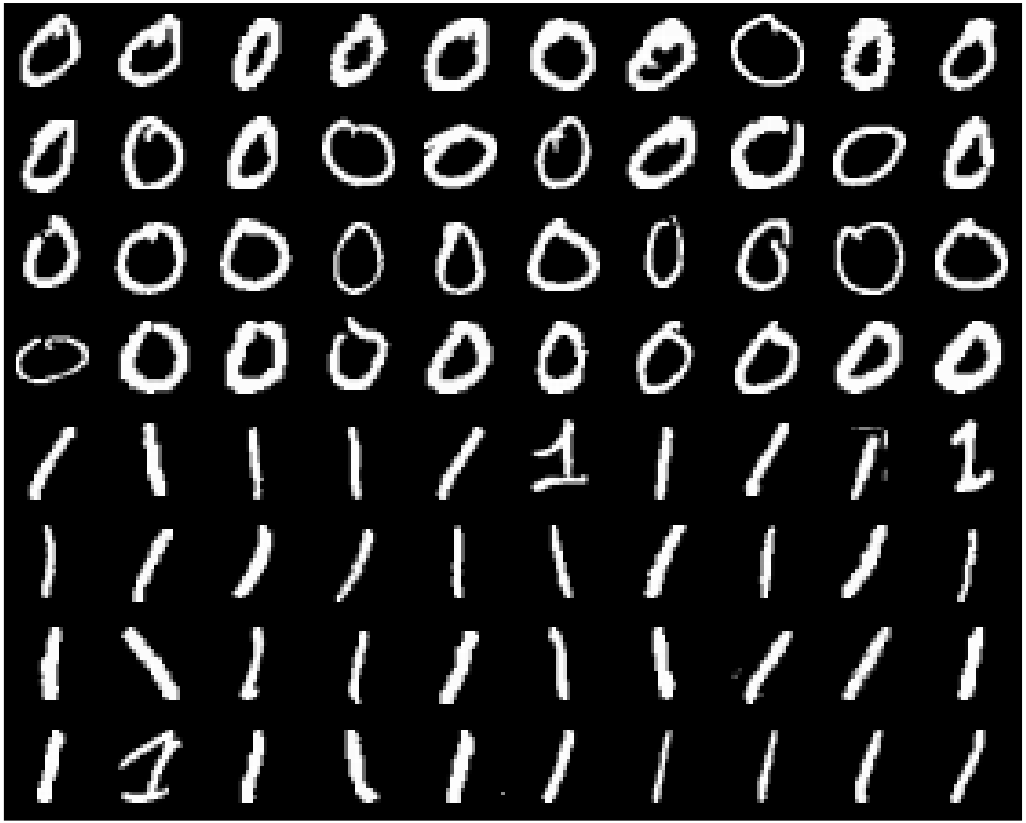
用对率回归识别垃圾邮件



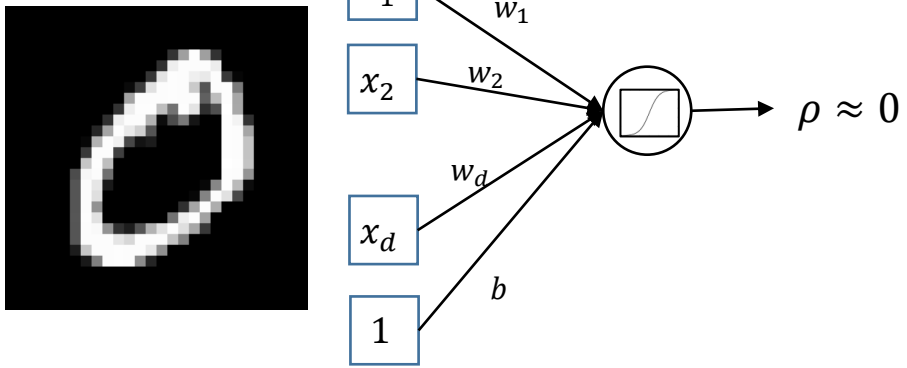


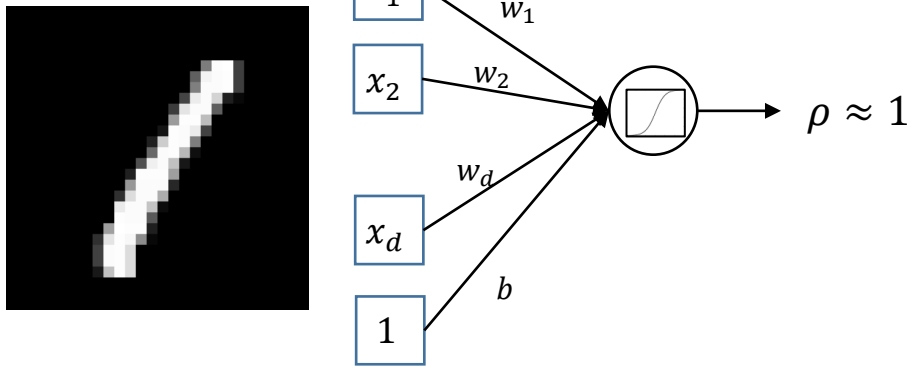


识别手写体数字0和1

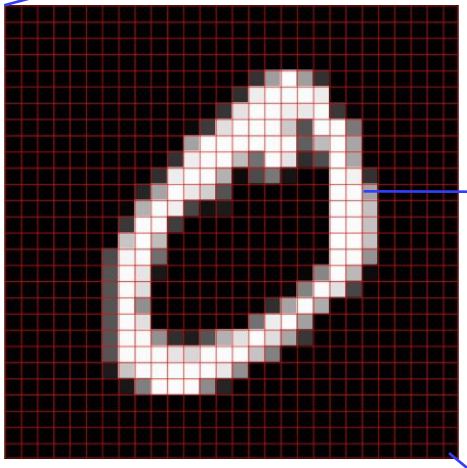


用对率回归识别数字0和1





用对率回归识别数字0和1



特征提取

x_1 w_1

x_2 w_2

x_2 ρ

w_d

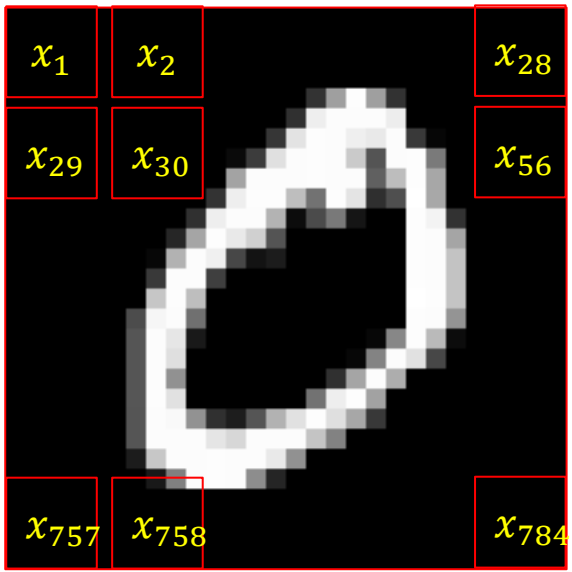
x_d

b

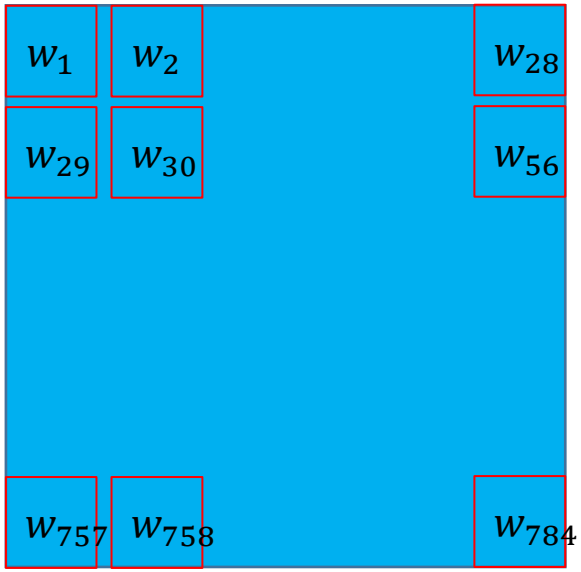
1

$d = 28 \times 28 = 784$

用对率回归识别数字0和1



X

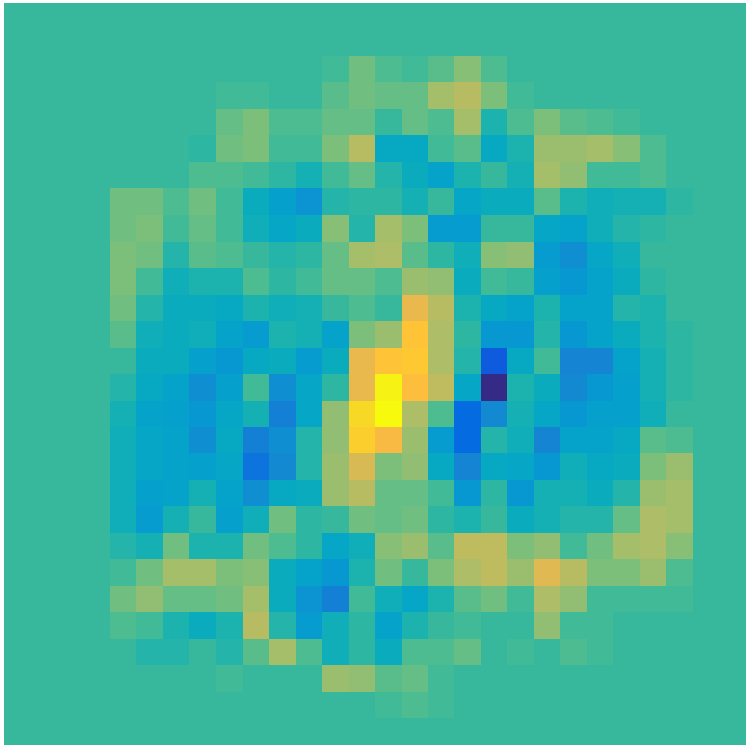


W

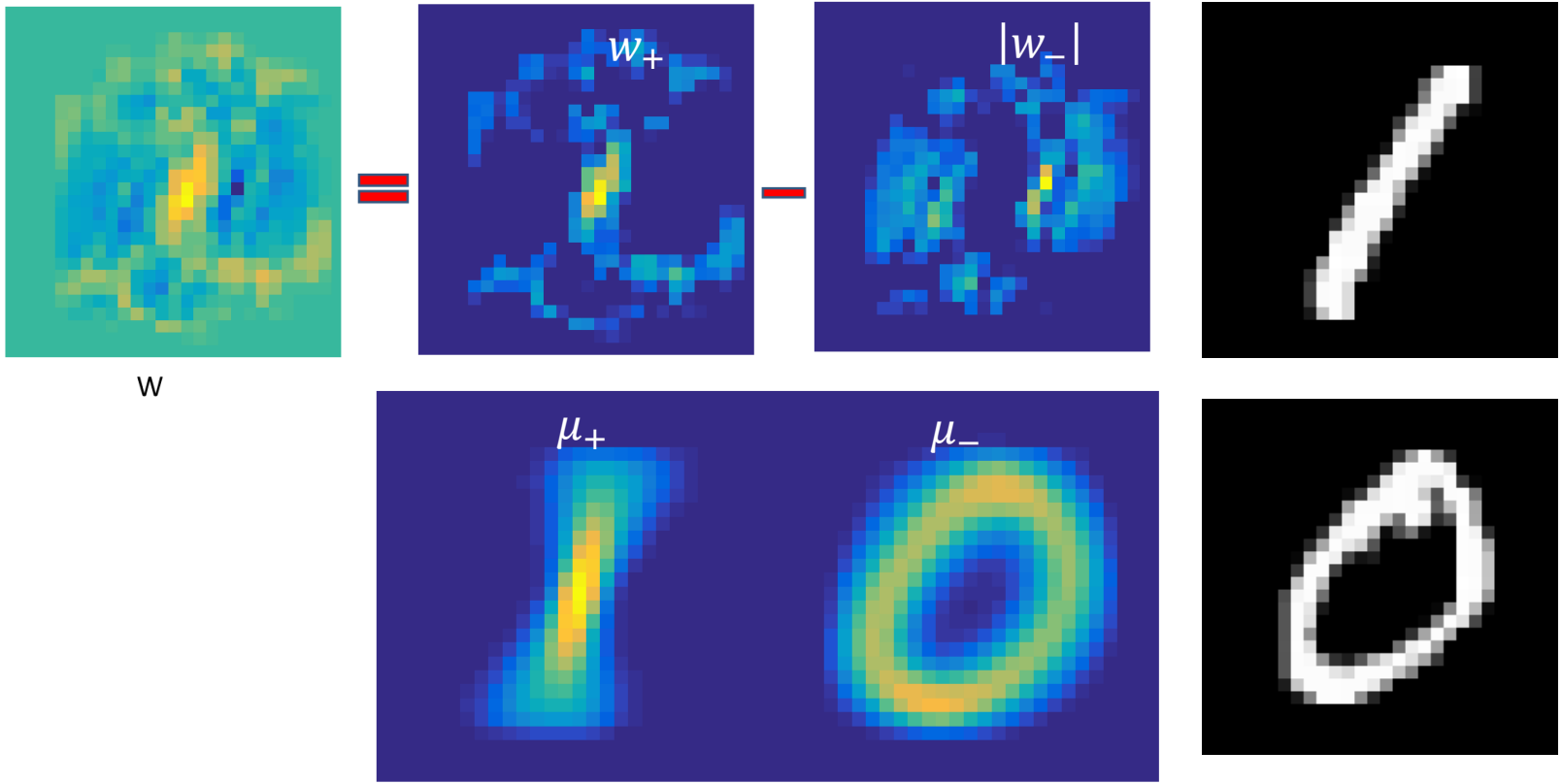
$$z = w_1x_1 + w_2x_2 + \cdots + w_{784}x_{784} + b$$

用对率回归识别数字0和1

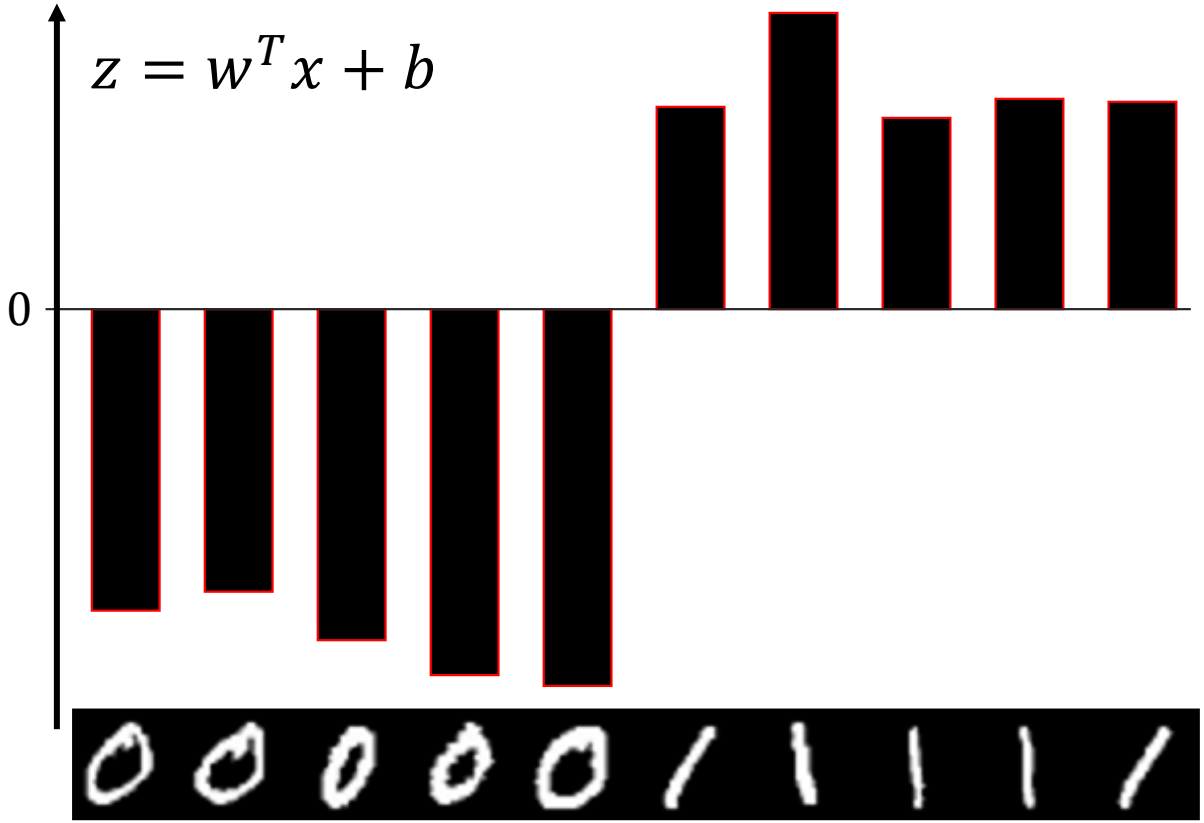
$$W =$$



用对率回归识别数字0和1



用对率回归识别数字0和1



小结

- 线性模型
- 感知机模型及其学习算法
  - 学习目标：正确分类全体训练样本
  - 不能处理不可分样本
  - 不一定得到最佳解
- 对率回归模型
  - Sigmoid响应函数
- 损失函数
  - 用损失函数评价模型在训练样本上的优劣
- 梯度下降法

课后作业

- 1. 假设有如下样本集：$D=\{(X^{(1)}=(1,2),y^{(1)}=+1),(X^{(2)}=(2,1),y^{(1)}=-1),(X^{(3)}=(-1,1),y^{(1)}=+1),(X^{(4)}=(-1,-1),y^{(1)}=-1)\}$。请用PLA算法计算可以区分该样本集的感知器。请参照本讲义P33页列表写出权重与偏置的更新过程。并写出你得到的感知器模型所确定的分界面方程。假设权值初始值为0向量，偏置初始值为0。
- 2. 对第一题的样本集，采用对数几率回归模型拟合一个分类器。请计算出权值 W 的第一步更新公式。假设权值初始值为0向量，学习率为1。
- 3. 用梯度下降法计算函数$f(x_1,x_2)=2x_1^2-3x_2+1$的极小值点，请写出该函数的导函数。假设算法采用$x_1=0,x_2=0$作为初始值，请写出第一步更新后x_1,x_2的值。假设学习率为1。