机器学习

苏州大学计算机科学与技术学院

自然语言处理实验室

主讲：周夏冰

邮箱：zhouxiabing@suda.edu.cn

感知机

感知机（perceptron）

- 1957年Rosenblatt提出，是神经网络和支持向量机的基础

- 感知准则
  - 由于Rosenblatt企图将其用于脑模型感知器，因此被称为感知准则函数，其特点是随意确定的判别函数初始值，在对样本分类训练过程中逐步修正直至最终确定

感知机（perceptron）

- 二类分类的线性分类模型
  - 输入：特征向量
  - 输出：实例的类别，取+1和-1二值

Dog recognition

+1

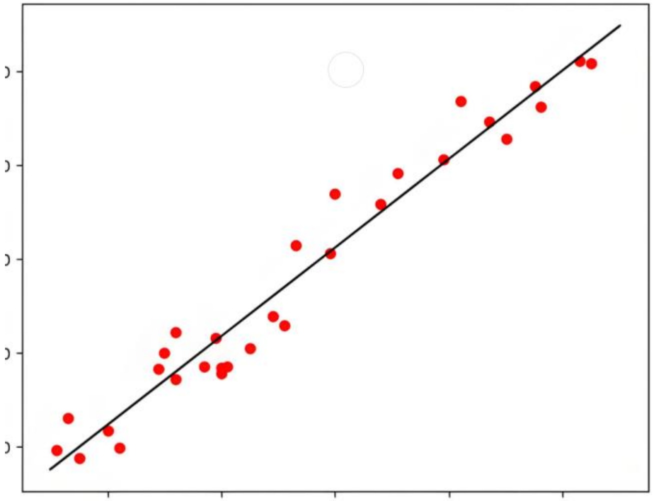
-1

sentiment classification

The weather is great today! +1

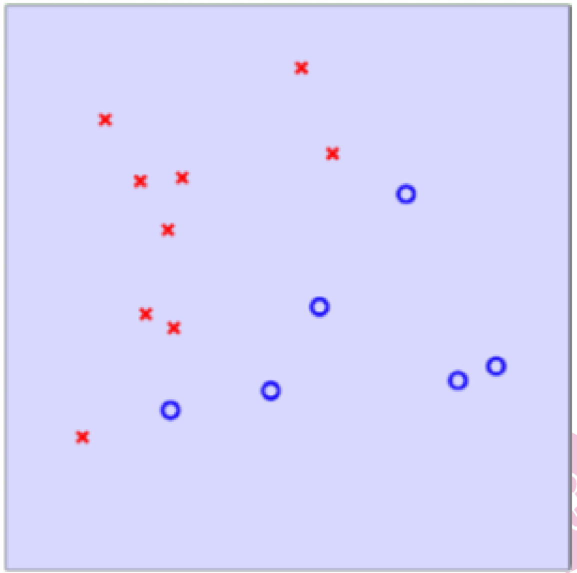
I didn't do well in the exam··· -1

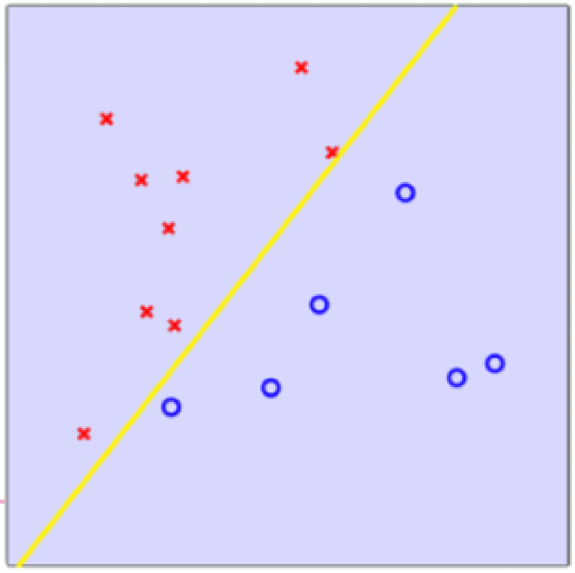
感知机（perceptron）



- 二类分类的线性分类模型
  - 输入：特征向量
  - 输出：实例的类别，取+1和-1二值

- 感知机对应于输入空间中将实例划分为正负两类的分离超平面，属于判别模型





感知机（perceptron）

- 二类分类的线性分类模型
  - 输入：特征向量
  - 输出：实例的类别，取+1和-1二值
- 感知机对应于输入空间中将实例划分为正负两类的分离超平面，属于判别模型
- 思想：导入基于误分类的损失函数，利用梯度下降法对损失函数进行极小化，求得感知机模型

感知机模型

- **定义**：假设输入空间（特征空间）是$\chi \in \mathcal{R}^n$，输出空间是$\mathcal{Y} \in \{-1,+1\}$。输入$x \in \chi$表示实例的特征向量，对应于输入空间的点；输出$y \in \mathcal{Y}$表示实例的类别。由输入空间到输出空间的如下函数：
- $f(x) = sign(w \cdot x + b)$ 称为**感知机**
- $sign(x) = \begin{cases} +1, x \geq 0 \\ -1, x < 0 \end{cases}$ **符号函数**
- **假设空间**：所有线性分类模型或线性分类器

$$\{f|f(x) = w \cdot x + b\}$$

感知机模型

- 几何解释
  - 超平面S $w \cdot x + b = 0$
  - 法向量 w
- 学习目标：
  - 寻找超平面，即求参数w、b的值

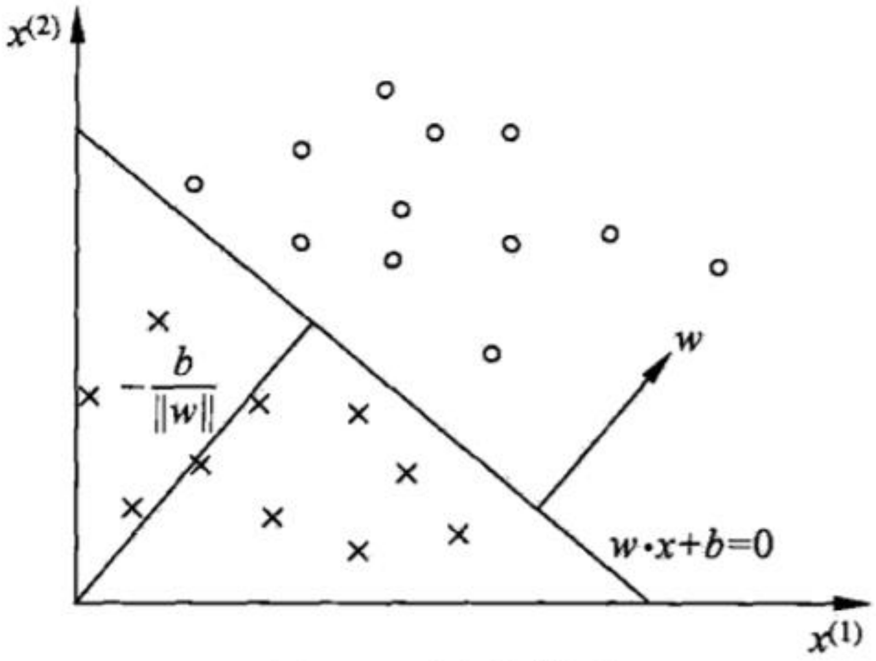
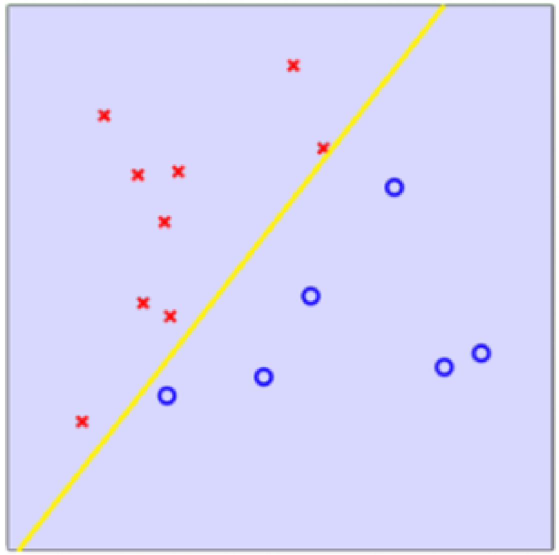
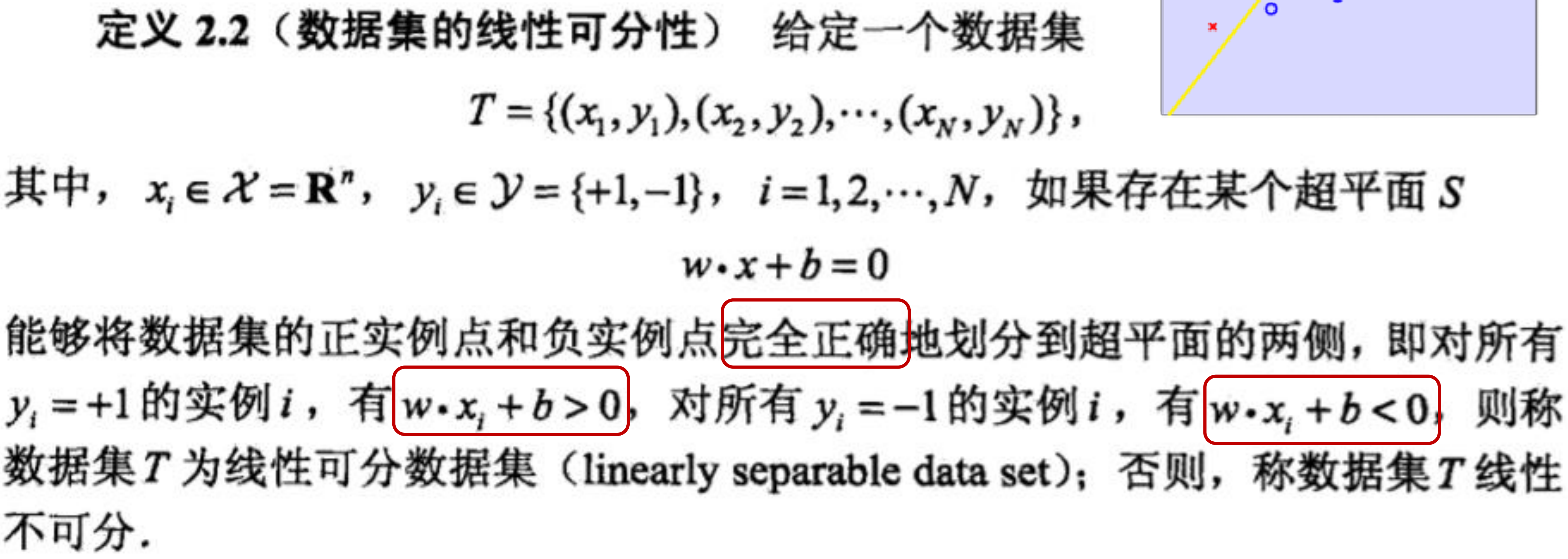


图 2.1　感知机模型

感知机学习策略

- 数据集的线性可分性





定义 2.2（数据集的线性可分性） 给定一个数据集

$$T=\{(x_1,y_1),(x_2,y_2),\cdots,(x_N,y_N)\},$$

其中，$x_i\in\mathcal{X}=\mathbf{R}^n$，$y_i\in\mathcal{Y}=\{+1,-1\}$，$i=1,2,\cdots,N$，如果存在某个超平面 S

$$w\cdot x+b=0$$

能够将数据集的正实例点和负实例点完全正确地划分到超平面的两侧，即对所有 $y_i=+1$ 的实例 i，有 $\boxed{w\cdot x_i+b>0}$，对所有 $y_i=-1$ 的实例 i，有 $\boxed{w\cdot x_i+b<0}$，则称数据集 T 为线性可分数据集（linearly separable data set）；否则，称数据集 T 线性不可分.

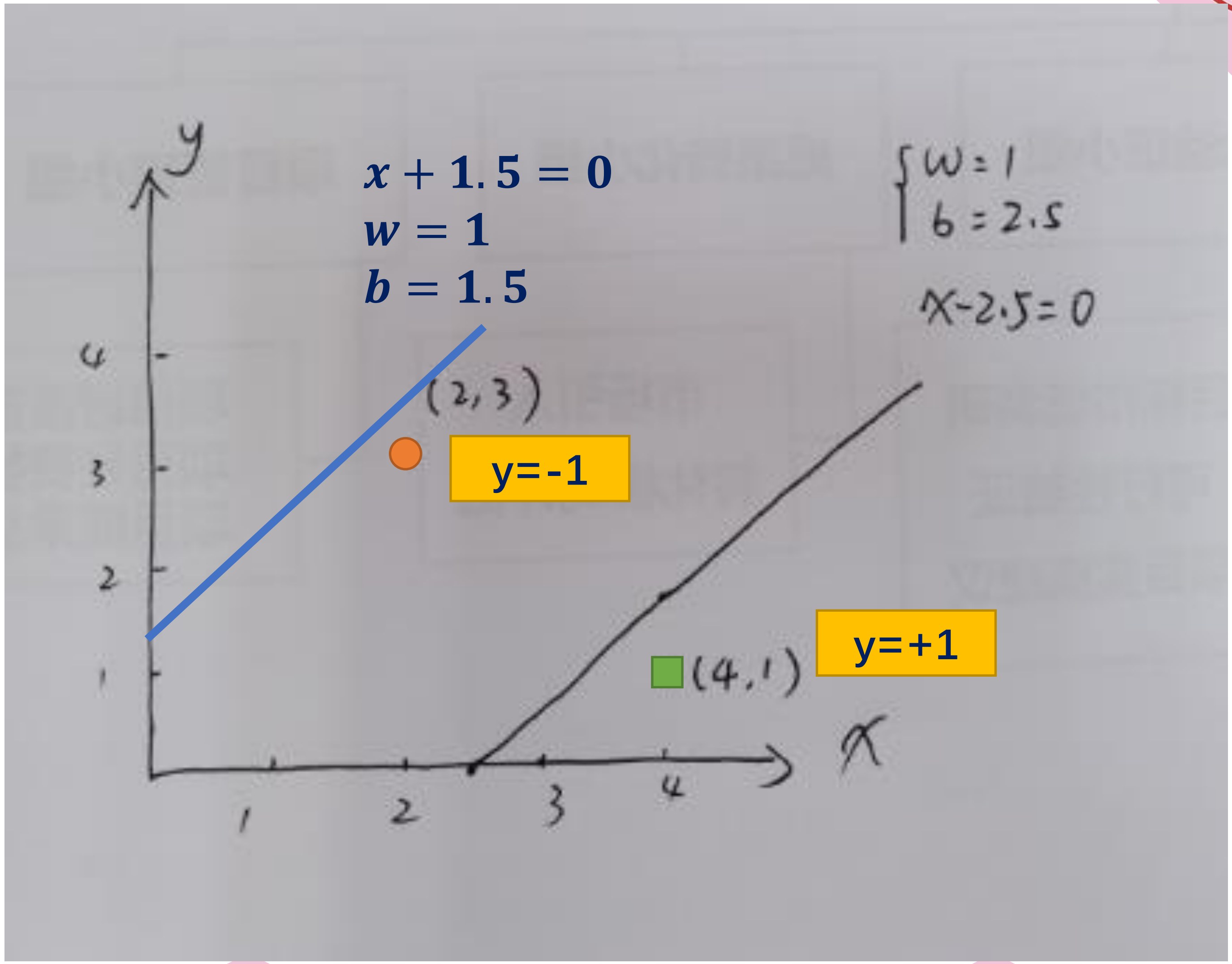
感知机

$$2 - 2.5 = -0.5 < 0$$

$$4 - 2.5 = 1.5 > 0$$

$$2 + 1.5 = 3.5 > 0$$

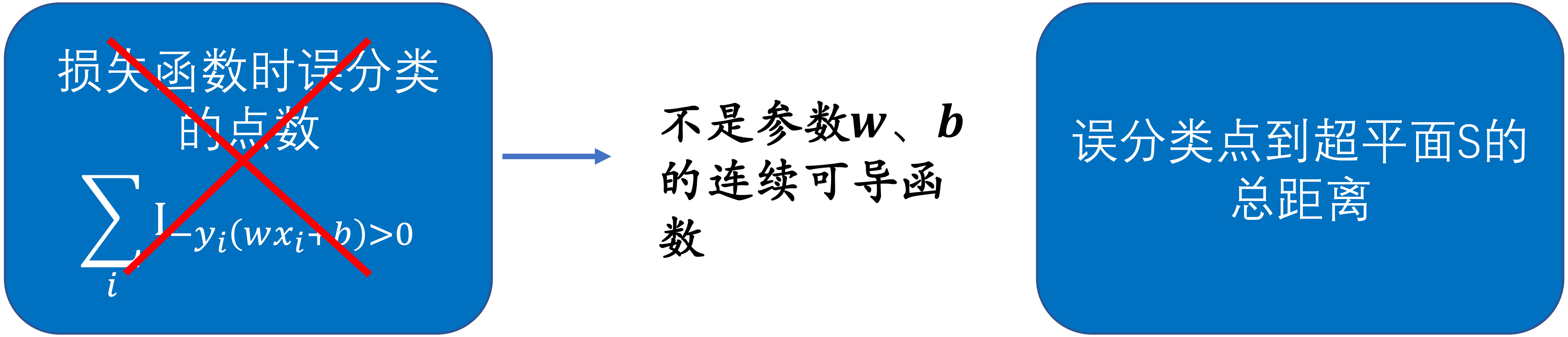
$$4 + 1.5 = 4.5 > 0$$



感知机学习策略

- 学习策略——定义损失函数并将损失函数极小化

- 损失函数的选择
  - 基于误分类点：wx_i+b与y_i异号



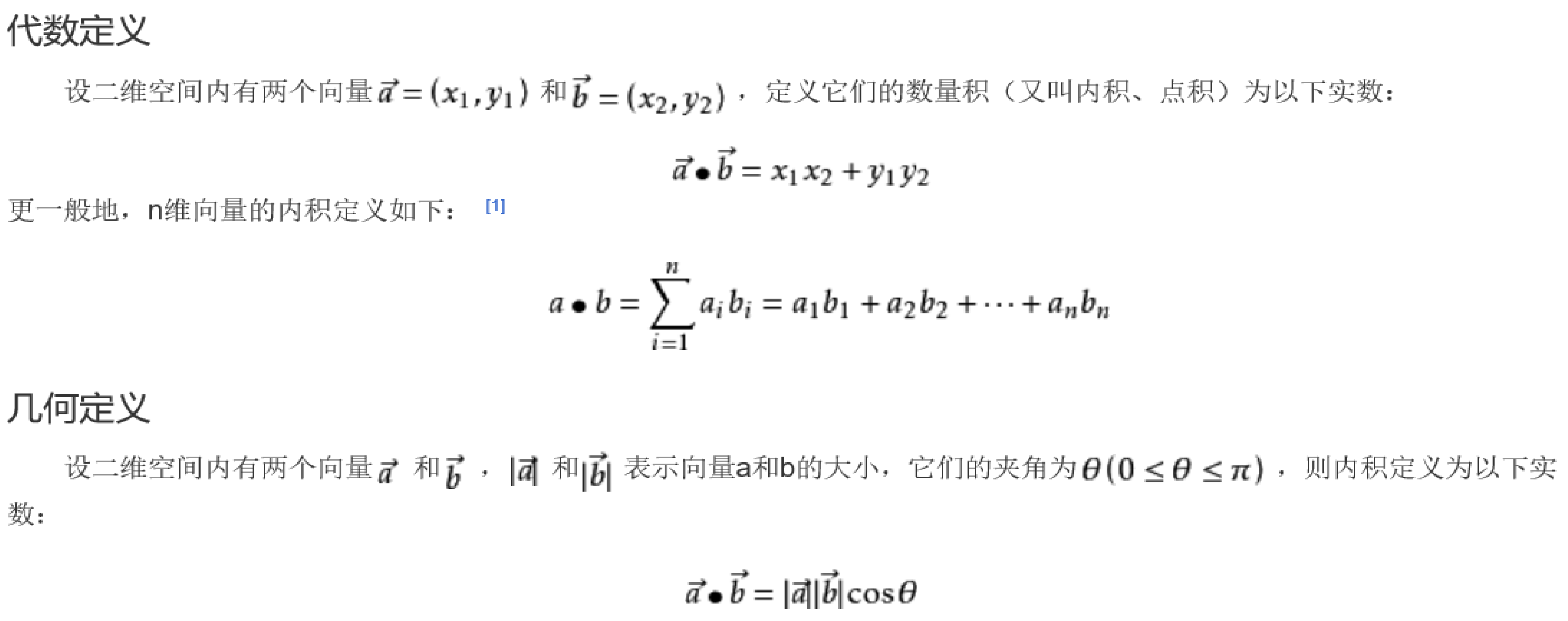
~~损失函数时误分类的点数~~

$$\cancel{\sum_i \mathrm{I}_{-y_i(wx_i+b)>0}}$$

不是参数w、b的连续可导函数

误分类点到超平面S的总距离

感



代数定义

设二维空间内有两个向量$\vec{a}=(x_1,y_1)$和$\vec{b}=(x_2,y_2)$，定义它们的数量积（又叫内积、点积）为以下实数：

$$\vec{a}\bullet\vec{b}=x_1x_2+y_1y_2$$

更一般地，n维向量的内积定义如下：[1]

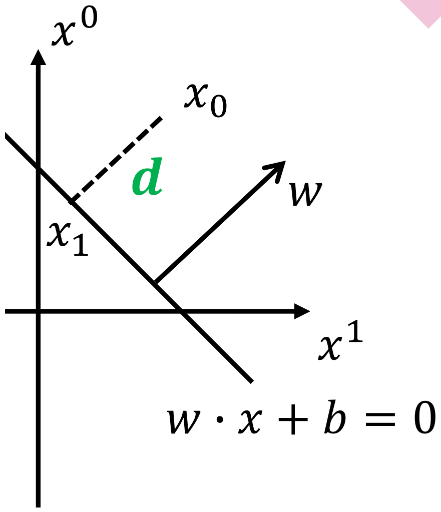
$$a\bullet b=\sum_{i=1}^{n}a_ib_i=a_1b_1+a_2b_2+\cdots+a_nb_n$$

几何定义

设二维空间内有两个向量$\vec{a}$ 和$\vec{b}$，$|\vec{a}|$ 和$|\vec{b}|$ 表示向量a和b的大小，它们的夹角为$\theta(0\le\theta\le\pi)$，则内积定义为以下实数：

$$\vec{a}\bullet\vec{b}=|\vec{a}||\vec{b}|\cos\theta$$

- 输
- 证



- $wx_1+b=0$
- $|w\cdot\overrightarrow{x_0x_1}|=|w||\overrightarrow{x_0x_1}|=\sqrt{(w^0)^2+(w^1)^2+\cdots+(w^{d-1})^2}\cdot d=\|w\|\cdot d$
- $w\cdot\overrightarrow{x_0x_1}=w^0(x_0^0-x_1^0)+w^1(x_0^1-x_1^1)+\cdots+w^{d-1}(x_0^{d-1}-x_1^{d-1})$

  $=w^0x_0^0+\cdots+w^{d-1}x_0^{d-1}-(-\mathrm{b})$
- $|w\cdot\overrightarrow{x_0x_1}|=\|w\|\cdot d=\left|w^0x_0^0+\cdots+w^{d-1}x_0^{d-1}-(-\mathrm{b})\right|=|\mathrm{w}\cdot x_0+b|$

感知机学习策略

- 对于误分类点(x_i, y_i)，$-y_i(wx_i+b)>0$恒成立。因此误分类点到超平面的距离

$$\frac{|w\cdot x_0+b|}{\|w\|} \Rightarrow \frac{-y_i(wx_i+b)}{\|w\|}$$

- 假设超平面S的误分类点集合为M，那么就有误分类点到超平面的总距离为

$$-\frac{1}{\|w\|}\sum_{x_i\in M} y_i(w\cdot x_i+b)$$

- 损失函数

$$L(w,b)=-\sum_{x_i\in M} y_i(w\cdot x_i+b)$$

极小化损失函数

感知机学习算法

- 感知机学习算法采用随机梯度下降法（SGD）
- 损失函数的$L(w,b)$的梯度

$$L(w,b) = -\sum_{x_i \in M} y_i(w \cdot x_i + b)$$

$$\Delta w = \frac{\partial L(w,b)}{\partial w} = -y_i x_i$$

$$\Delta b = \frac{\partial L(w,b)}{\partial b} = -y_i$$

梯度下降小家族

- **批量梯度下降法（Batch Gradient Descent，BGD）**
  - 使用整个数据集去计算损失函数的梯度。每次使用全部数据计算梯度去更新参数，批量梯度下降法会很慢。

$$\frac{\partial L}{\partial w} = \sum_{i=1}^{trainN} 2((wx_i + b) - y_i)(x_i)$$

- **随机梯度下降法（Stochastic Gradient Descent，SGD）**
  - 随机使用整个数据集，在每次迭代仅选择一个训练样本去计算损失函数的梯度，然后更新参数。随机梯度下降法得到结果的准确性可能不会是最好的，但是计算结果的速度很快。

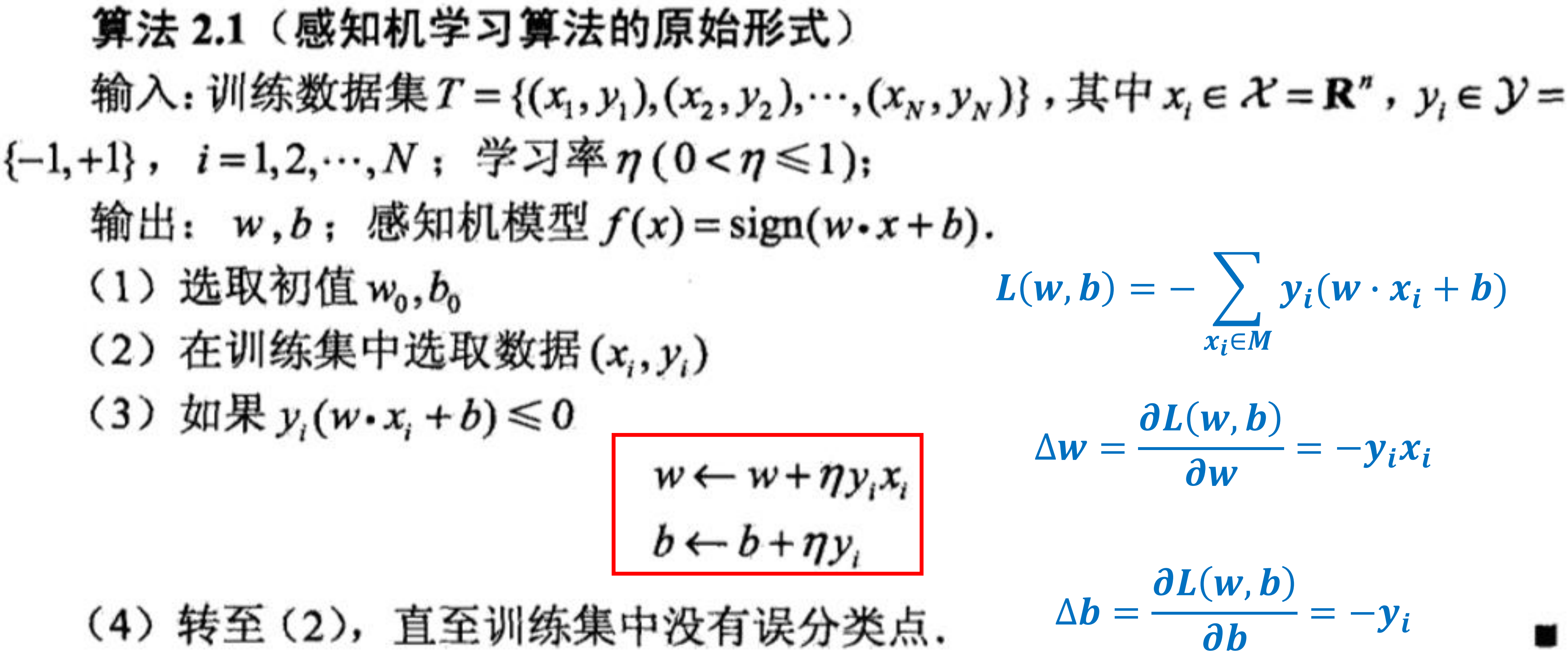
$$\frac{\partial L}{\partial w} = 2((wx_i + b) - y_i)(x_i)$$

- **小批量梯度下降法（Mini-Batch Gradient Descent, MBGD）**
  - 小批量梯度下降法不是使用完整数据集，在每次迭代中仅使用m个训练样本去计算损失函数的梯度。这种方法减少了参数更新时的变化，能够更加稳定地收敛。

$$\frac{\partial L}{\partial w} = \sum_{i=1}^{m} 2((wx_i + b) - y_i)(x_i)$$

感知机学习算法

- 感知机学习算法原始形式：



算法 2.1（感知机学习算法的原始形式）

输入：训练数据集 $T=\{(x_1,y_1),(x_2,y_2),\cdots,(x_N,y_N)\}$，其中 $x_i\in\mathcal{X}=\mathbf{R}^n$，$y_i\in\mathcal{Y}=\{-1,+1\}$，$i=1,2,\cdots,N$；学习率 $\eta\,(0<\eta\leqslant 1)$；

输出：w,b；感知机模型 $f(x)=\operatorname{sign}(w\cdot x+b)$.

（1）选取初值 w_0,b_0

（2）在训练集中选取数据 (x_i,y_i)

（3）如果 $y_i(w\cdot x_i+b)\leqslant 0$

$$\boxed{\begin{aligned}w&\leftarrow w+\eta y_i x_i\\ b&\leftarrow b+\eta y_i\end{aligned}}$$

（4）转至（2），直至训练集中没有误分类点. ■

$$L(w,b)=-\sum_{x_i\in M}y_i(w\cdot x_i+b)$$

$$\Delta w=\frac{\partial L(w,b)}{\partial w}=-y_i x_i$$

$$\Delta b=\frac{\partial L(w,b)}{\partial b}=-y_i$$

感知机学习算法

- 例子1：在训练数据集中
- 正实例点（$y_i = +1$）是$x_1 = (3,3)^T, x_2 = (4,3)^T$
- 负实例点（$y_i = -1$）是$x_3 = (1,1)^T$.
- 试用感知机学习算法的原始形式求感知机模型$\mathrm{f}(x) = \mathrm{sign}(wx + \mathrm{b})$，这里，$w = (w^1, w^2)^T, x \ = (x^1, x^2)^T$

$$(y_i = +1)\ :\ x_1 = (3,3)^T, x_2 = (4,3)^T$$

$$(y_i = -1)\ :\ x_3 = (1,1)^T$$

$\eta = 1$

$y_i(w \cdot x_i + b) \le 0$，
则更新：

$w \leftarrow w + \eta y_i x_i$

$b \leftarrow b + \eta y_i$

迭代次数	误分类点	w	b	w·x+b
0		0	0	0
1	x_1	$(3,3)^T$	1	$3x^{(1)}+3x^{(2)}+1$
2	x_3	$(2,2)^T$	0	$2x^{(1)}+2x^{(2)}$
3	x_3	$(1,1)^T$	-1	$x^{(1)}+x^{(2)}-1$
4	x_3	$(0,0)^T$	-2	-2
5	x_1	$(3,3)^T$	-1	$3x^{(1)}+3x^{(2)}-1$
6	x_3	$(2,2)^T$	-2	$2x^{(1)}+2x^{(2)}-2$
7	x_3	$(1,1)^T$	-3	$x^{(1)}+x^{(2)}-3$
8	0	$(1,1)^T$	-3	$x^{(1)}+x^{(2)}-3$

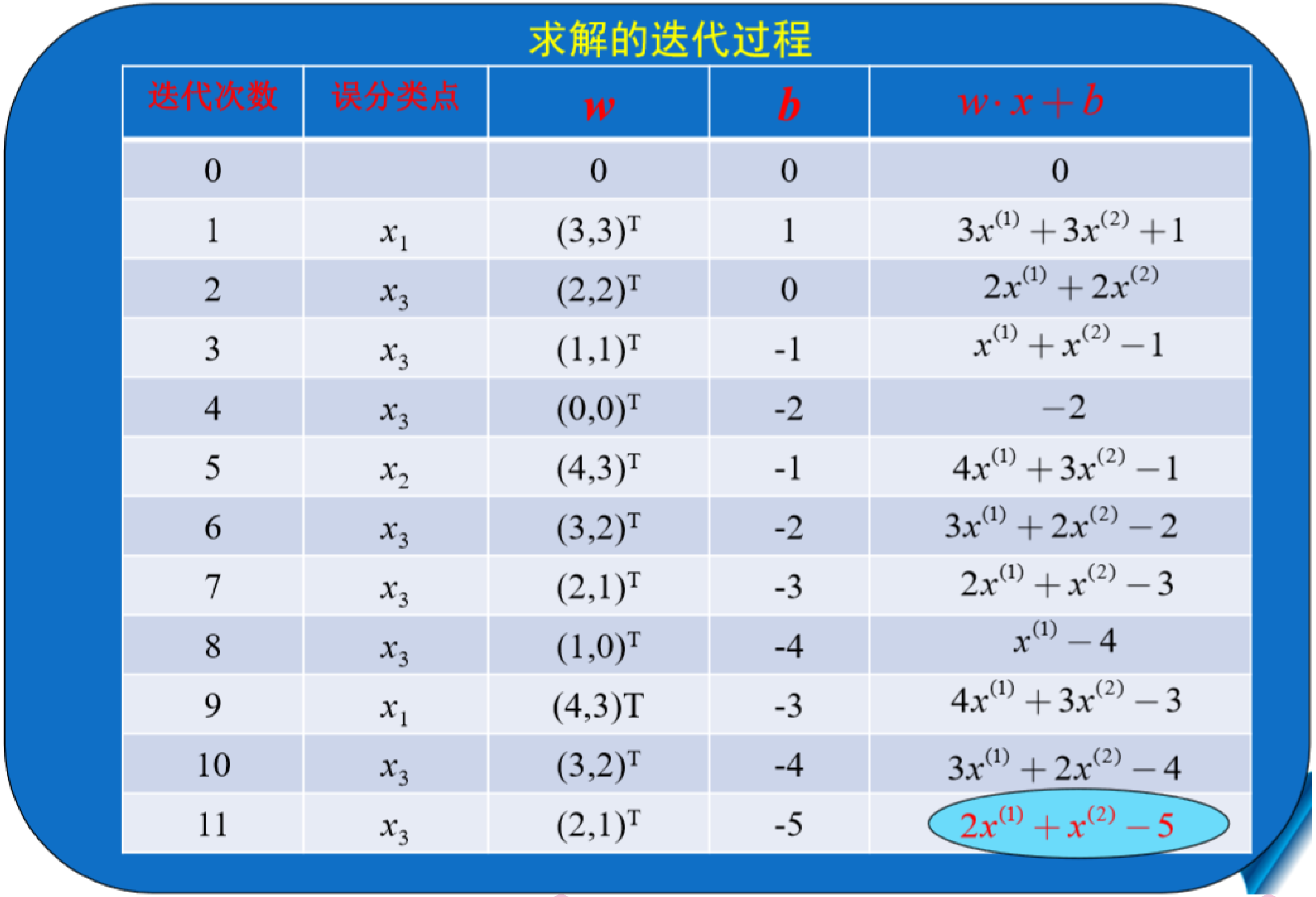
$$(y_i = +1)\ :\ x_1 = (3,3)^T,\ x_2 = (4,3)^T$$
$$(y_i = -1)\ :\ x_3 = (1,1)^T$$

$\eta = 1$

$y_i(w \cdot x_i + b) \leq 0$,
则更新：

$w \leftarrow w + \eta y_i x_i$

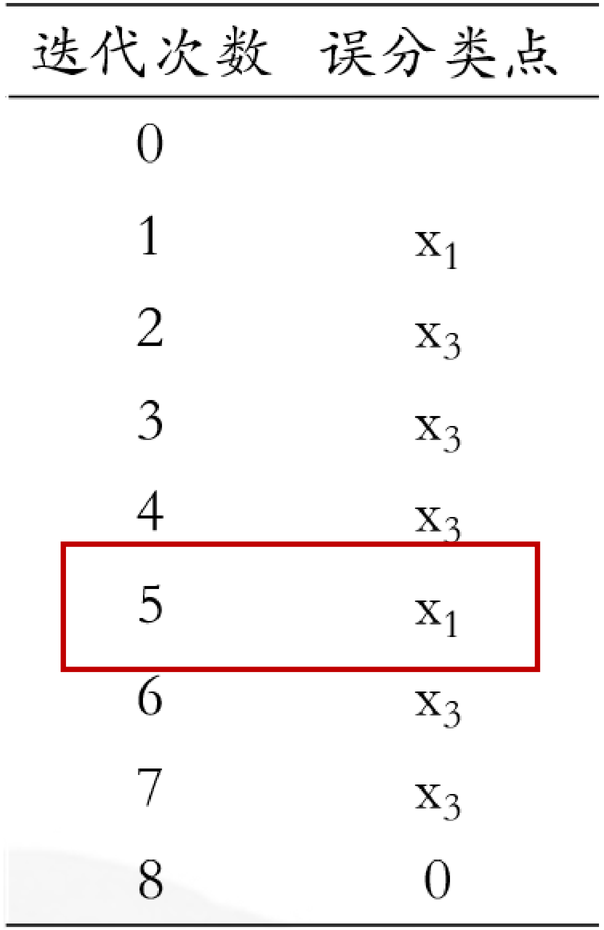
$b \leftarrow b + \eta y_i$



求解的迭代过程

迭代次数	误分类点	w	b	$w \cdot x + b$
0		0	0	0
1	x_1	$(3,3)^T$	1	$3x^{(1)} + 3x^{(2)} + 1$
2	x_3	$(2,2)^T$	0	$2x^{(1)} + 2x^{(2)}$
3	x_3	$(1,1)^T$	-1	$x^{(1)} + x^{(2)} - 1$
4	x_3	$(0,0)^T$	-2	-2
5	x_2	$(4,3)^T$	-1	$4x^{(1)} + 3x^{(2)} - 1$
6	x_3	$(3,2)^T$	-2	$3x^{(1)} + 2x^{(2)} - 2$
7	x_3	$(2,1)^T$	-3	$2x^{(1)} + x^{(2)} - 3$
8	x_3	$(1,0)^T$	-4	$x^{(1)} - 4$
9	x_1	(4,3)T	-3	$4x^{(1)} + 3x^{(2)} - 3$
10	x_3	$(3,2)^T$	-4	$3x^{(1)} + 2x^{(2)} - 4$
11	x_3	$(2,1)^T$	-5	$2x^{(1)} + x^{(2)} - 5$

$$(y_i = +1) \ : \ x_1 = (3,3)^T, x_2 = (4,3)^T$$
$$(y_i = -1) \ : \ x_3 = (1,1)^T$$



迭代次数	误分类点
0	
1	x_1
2	x_3
3	x_3
4	x_3
5	x_1
6	x_3
7	x_3
8	0

求解的迭代过程

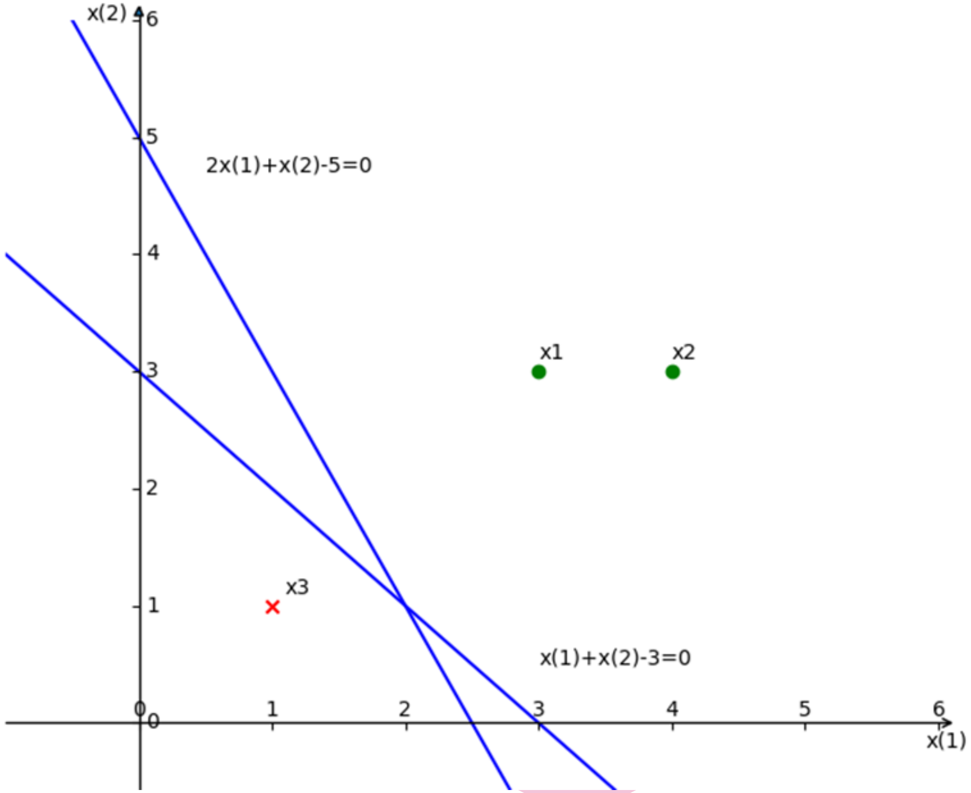
迭代次数	误分类点
0	
1	x_1
2	x_3
3	x_3
4	x_3
5	x_2
6	x_3
7	x_3
8	x_3
9	x_1
10	x_3
11	x_3

感知机学习算法存在许多解，既依赖初值的选择，也依赖迭代过程中误分类点的选择顺序

感知机学习算法

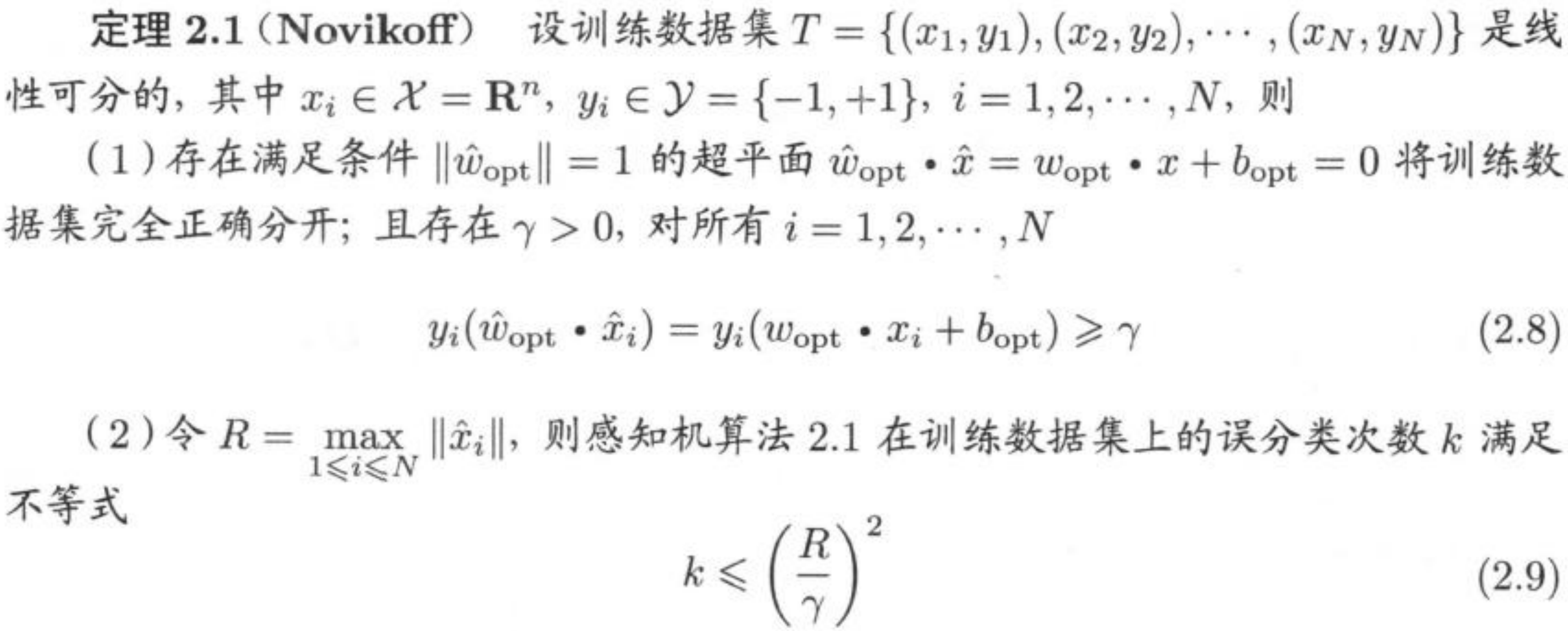
- 例子1：在训练数据集中
- 正实例点（$y_i = +1$）是$x_1 = (3,3)^T, x_2 = (4,3)^T$
- 负实例点（$y_i = -1$）是$x_3 = (1,1)^T$.
- 试用感知机学习算法的原始形式求感知机模

  里，$w = (w^1, w^2)^T, x = (x^1, x^2)^T$



感知机学习算法

- **收敛性**：$\hat{w} = (w^T, b)^T, \hat{x} = (x^T, 1)^T$



定理 2.1（Novikoff） 设训练数据集 $T = \{(x_1, y_1), (x_2, y_2), \cdots, (x_N, y_N)\}$ 是线性可分的，其中 $x_i \in \mathcal{X} = \mathbf{R}^n$，$y_i \in \mathcal{Y} = \{-1, +1\}$，$i = 1, 2, \cdots, N$，则

（1）存在满足条件 $\|\hat{w}_{\text{opt}}\| = 1$ 的超平面 $\hat{w}_{\text{opt}} \cdot \hat{x} = w_{\text{opt}} \cdot x + b_{\text{opt}} = 0$ 将训练数据集完全正确分开；且存在 $\gamma > 0$，对所有 $i = 1, 2, \cdots, N$

$$y_i(\hat{w}_{\text{opt}} \cdot \hat{x}_i) = y_i(w_{\text{opt}} \cdot x_i + b_{\text{opt}}) \geqslant \gamma \tag{2.8}$$

（2）令 $R = \max\limits_{1 \leqslant i \leqslant N} \|\hat{x}_i\|$，则感知机算法 2.1 在训练数据集上的误分类次数 k 满足不等式

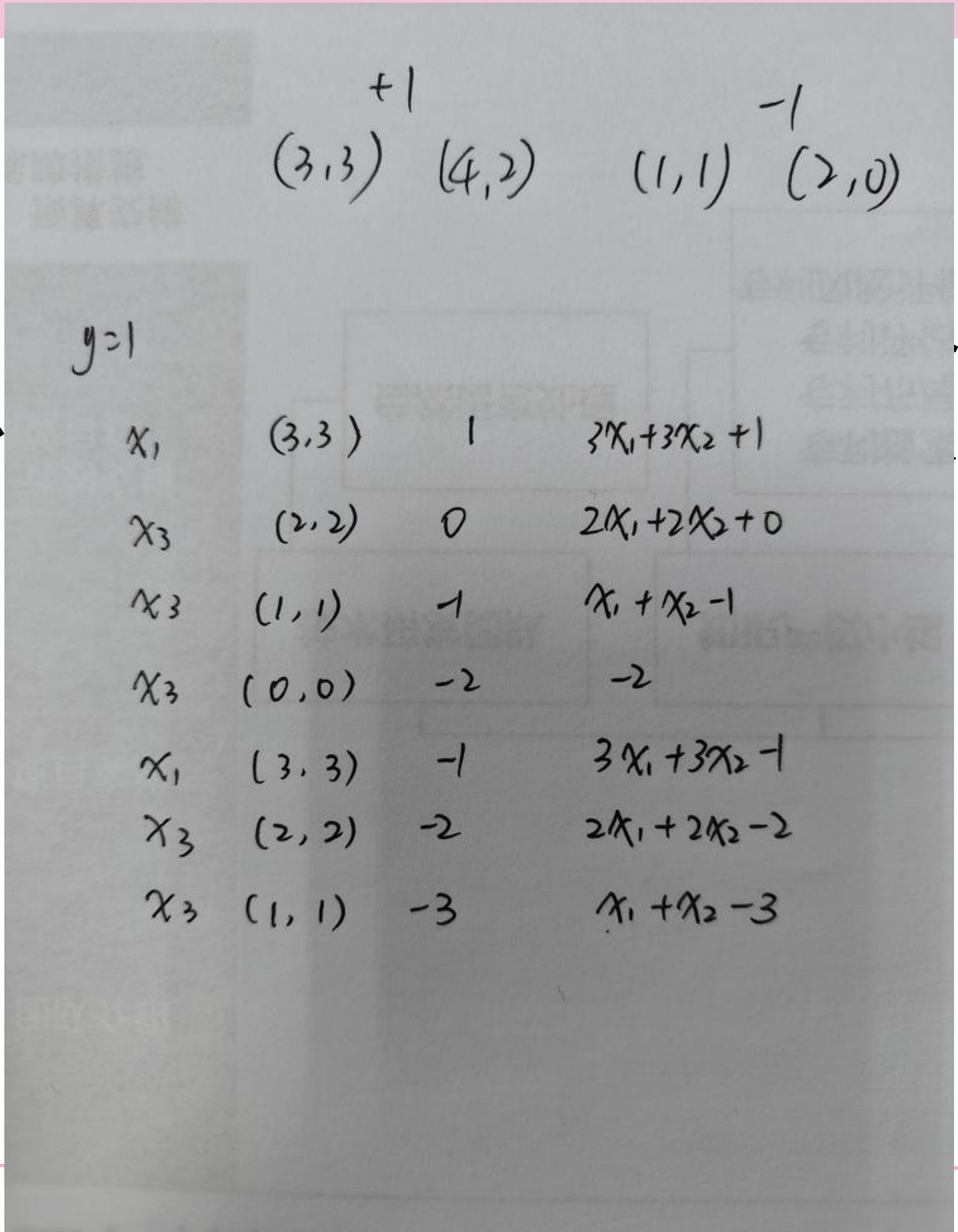
$$k \leqslant \left(\frac{R}{\gamma}\right)^2 \tag{2.9}$$

课堂练习

- 已知训练数据集，正实例点是$x_1=(3,3)^T$, $x_2=(4,2)^T$, 负实例点是$x_3=(1,1)^T$, $x_4=(2,0)^T$，用原始形式求出感知机模型决策函数f=sign(wx+b)，按照例题形式给出过程

课堂练习

- 已知训练数据集，实例点是$x_3=(1,1)^T$, $x_4=(2,0)^T$，用原始gn(wx+b)，按照例题形式给出过程



$+1$ $(3,3)$ $(4,2)$ -1 $(1,1)$ $(2,0)$

$y=1$

x_1	$(3,3)$	1	$3x_1+3x_2+1$
x_3	$(2,2)$	0	$2x_1+2x_2+0$
x_3	$(1,1)$	-1	x_1+x_2-1
x_3	$(0,0)$	-2	-2
x_1	$(3,3)$	-1	$3x_1+3x_2-1$
x_3	$(2,2)$	-2	$2x_1+2x_2-2$
x_3	$(1,1)$	-3	x_1+x_2-3

逻辑回归

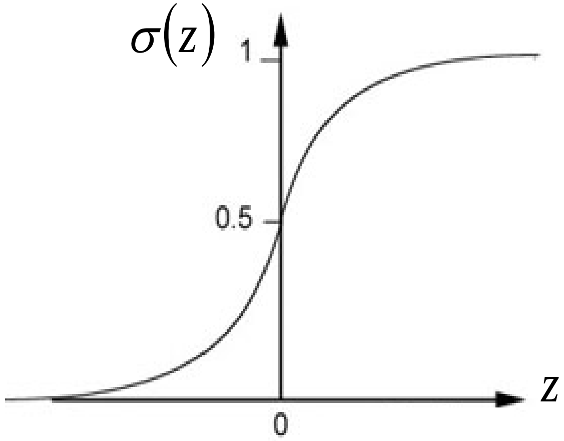
Logistic回归

- 为了解决连续的线性函数不适合进行分类的问题，引入非线性函数g来预测类别标签的条件概率$p(y = c|x)$
- 二分类：$\boldsymbol{p}(\boldsymbol{y} = \mathbf{1}|\boldsymbol{x}) = \boldsymbol{g}\big(\boldsymbol{f}(\boldsymbol{x}; \boldsymbol{w})\big)$
  - 函数f：线性函数
  - 函数g：把线性函数的值域从实数区间“挤压”到了(0,1)之间，可以用来表示概率。

Logistic regression

$$\begin{cases} \sigma(z) \geq 0.5 & \text{class 1} \\ \sigma(z) < 0.5 & \text{class 2} \end{cases}$$

- 逻辑斯蒂分布



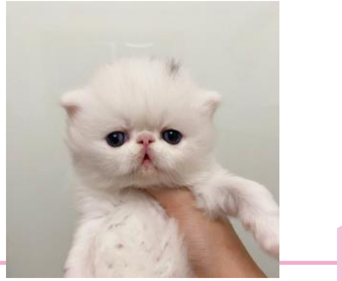
$$\sigma(z) = \frac{1}{1 + exp(-z)}$$

逻辑斯蒂回归

$$p(y = 1|x) = \frac{1}{1 + exp(-wx + b)}$$

Dog recognition

1



0

$y = 0.9$

$y = 0.65$

$y = 0.4$

logistic

Training Data	x^1	x^2	x^3	……	x^N
	C_1	C_1	C_2		C_1

假设x满足：$f_{w,b}(x) = P_{w,b}(C_1|x)$

求出w,b，根据最大似然估计的思路，生成这批训练数据的最大概率为：

$$L(w,b) = f_{w,b}(x^1) f_{w,b}(x^2) \left(1 - f_{w,b}(x^3)\right) \cdots f_{w,b}(x^N)$$

w和b：

$$w^*, b^* = arg \max_{w,b} L(w,b)$$

logistic

$$\boxed{\begin{array}{cccc} x^1 & x^2 & x^3 & \\ & & & \dots\dots \\ C_1 & C_1 & C_2 & \end{array}} \Rightarrow \boxed{\begin{array}{cccc} x^1 & x^2 & x^3 & \\ & & & \dots\dots \\ \hat{y}^1 = 1 & \hat{y}^2 = 1 & \hat{y}^3 = 0 & \end{array}}$$

$\hat{y}^n$: 1 for class 1, 0 for class 2

$$L(w,b) = f_{w,b}(x^1) f_{w,b}(x^2)\left(1 - f_{w,b}(x^3)\right) \cdots$$

$$\boxed{w^*, b^* = arg \max_{w,b} L(w,b)} = \boxed{w^*, b^* = arg \min_{w,b} -lnL(w,b)}$$

$$-lnL(w,b)$$

$$= -lnf_{w,b}(x^1) \Rightarrow -[1\ lnf(x^1) \cancel{+\ 0\ ln(1 - f(x^1))}]$$

$$-lnf_{w,b}(x^2) \Rightarrow -[1\ lnf(x^2) \cancel{+\ 0\ ln(1 - f(x^2))}]$$

$$-ln\left(1 - f_{w,b}(x^3)\right) \Rightarrow -[\cancel{0\ lnf(x^3) +}\ 1\ ln(1 - f(x^3))]$$

$$\vdots$$

logistic

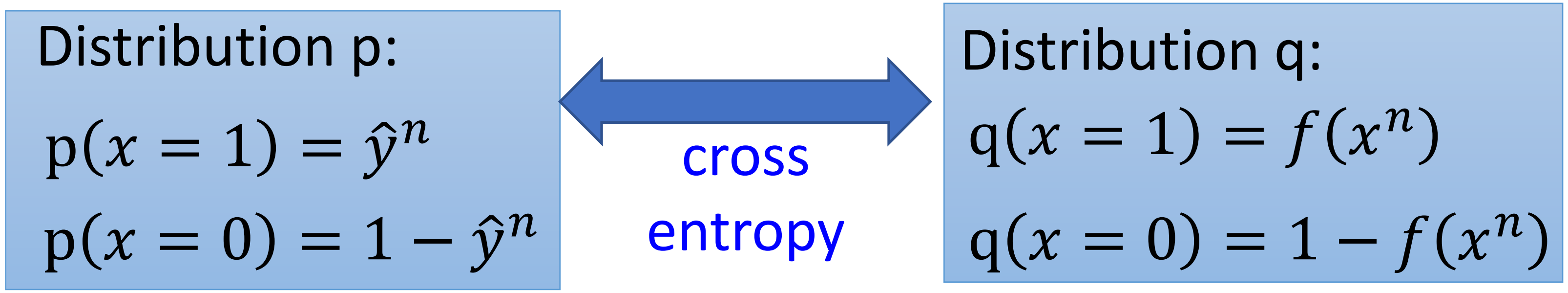
$$L(w,b) = f_{w,b}(x^1)f_{w,b}(x^2)\left(1 - f_{w,b}(x^3)\right)\cdots f_{w,b}(x^N)$$

$$-lnL(w,b) = -\left[lnf_{w,b}(x^1) + lnf_{w,b}(x^2) + ln\left(1 - f_{w,b}(x^3)\right)\cdots\right]$$

$\hat{y}^n$: 1 for class 1, 0 for class 2

$$= \sum_n \underline{-\left[\hat{y}^n lnf_{w,b}(x^n) + (1-\hat{y}^n)ln\left(1 - f_{w,b}(x^n)\right)\right]}$$

Cross entropy between two Bernoulli distribution



Distribution p:

$p(x = 1) = \hat{y}^n$

$p(x = 0) = 1 - \hat{y}^n$

cross entropy

Distribution q:

$q(x = 1) = f(x^n)$

$q(x = 0) = 1 - f(x^n)$

$$H(p,q) = -\sum_x p(x)ln\left(q(x)\right)$$

附

- **相对熵——KL散度**
  - 衡量两个分布的距离
  - 假设p(x)为真实分布，q(x)为非真实
    - $D_{KL}(p||q) = \sum_{i=1}^{n} p(x_i) \log \frac{p(x_i)}{q(x_i)}$
  - **KL散度大于等于0**
    - 最小化KL散度，使得P(x)和q(x)尽可能相同
    - $\sum_{i=1}^{n} p(x_i) \log p(x_i) - \sum_{i=1}^{n} p(x_i) \log q(x_i)$

Logistic

Step 1: $f_{w,b}(x) = \sigma(wx + b) = \frac{1}{1 + exp(-wx + b)}$ Output: between 0 and 1

Step 2: Cross entropy: $l(f(x^n), \hat{y}^n) = -\left[\hat{y}^n lnf(x^n) + (1 - \hat{y}^n) ln\left(1 - f(x^n)\right)\right]$

Logistic

$$\frac{-\ln L(w,b)}{\partial w_i} = \sum_n -\left[\hat{y}^n \overset{\left(1-f_{w,b}(x^n)\right)x_i^n}{\boxed{\frac{\ln f_{w,b}(x^n)}{\partial w_i}}} + (1-\hat{y}^n)\frac{\ln\left(1-f_{w,b}(x^n)\right)}{\partial w_i}\right]$$

$$\frac{\partial \ln f_{w,b}(x)}{\partial w_i} = \frac{\partial \ln f_{w,b}(x)}{\partial z}\frac{\partial z}{\partial w_i} \qquad \frac{\partial z}{\partial w_i} = x_i$$

$$\frac{\partial \ln \sigma(z)}{\partial z} = \frac{1}{\sigma(z)}\frac{\partial \sigma(z)}{\partial z} = \frac{1}{\cancel{\sigma(z)}}\cancel{\sigma(z)}\left(1-\sigma(z)\right)$$

$$f_{w,b}(x) = \sigma(z)$$
$$= 1/(1+exp(-z))$$

$$z = w \cdot x + b = \sum_i w_i x_i + b$$

Logistic

$$\frac{-lnL(w,b)}{\partial w_i} = \sum_n -\left[\hat{y}^n \overset{\left(1-f_{w,b}(x^n)\right)x_i^n}{\boxed{\frac{lnf_{w,b}(x^n)}{\partial w_i}}} + (1-\hat{y}^n)\overset{-f_{w,b}(x^n)x_i^n}{\boxed{\frac{ln\left(1-f_{w,b}(x^n)\right)}{\partial w_i}}}\right]$$

$$\frac{\partial ln\left(1-f_{w,b}(x)\right)}{\partial w_i} = \frac{\partial ln\left(1-f_{w,b}(x)\right)}{\partial z}\frac{\partial z}{\partial w_i} \qquad \frac{\partial z}{\partial w_i} = x_i$$

$$\frac{\partial ln\left(1-\sigma(z)\right)}{\partial z} = -\frac{1}{1-\sigma(z)}\frac{\partial\sigma(z)}{\partial z} = -\frac{1}{\cancel{1-\sigma(z)}}\sigma(z)\cancel{\left(1-\sigma(z)\right)}$$

$$f_{w,b}(x) = \sigma(z)$$
$$= 1/1+exp(-z)$$

$$z = w\cdot x + b = \sum_i w_i x_i + b$$

Logistic

$$\frac{-\ln L(w,b)}{\partial w_i} = \sum_n -\left[\hat{y}^n \overset{\left(1-f_{w,b}(x^n)\right)x_i^n}{\boxed{\frac{\ln f_{w,b}(x^n)}{\partial w_i}}} + (1-\hat{y}^n)\overset{-f_{w,b}(x^n)x_i^n}{\boxed{\frac{\ln\left(1-f_{w,b}(x^n)\right)}{\partial w_i}}}\right]$$

$$= \sum_n -\left[\hat{y}^n \underline{\left(1-f_{w,b}(x^n)\right)x_i^n} - (1-\hat{y}^n)\underline{f_{w,b}(x^n)x_i^n}\right]$$

$$= \sum_n -\left[\hat{y}^n \cancel{-\hat{y}^n f_{w,b}(x^n)} - f_{w,b}(x^n) + \cancel{\hat{y}^n f_{w,b}(x^n)}\right] \underline{x_i^n}$$

$$= \sum_n -\left(\hat{y}^n - f_{w,b}(x^n)\right) x_i^n$$

Larger difference, larger update

$$w_i \leftarrow w_i - \eta \sum_n -\left(\underline{\hat{y}^n - f_{w,b}(x^n)}\right) x_i^n$$

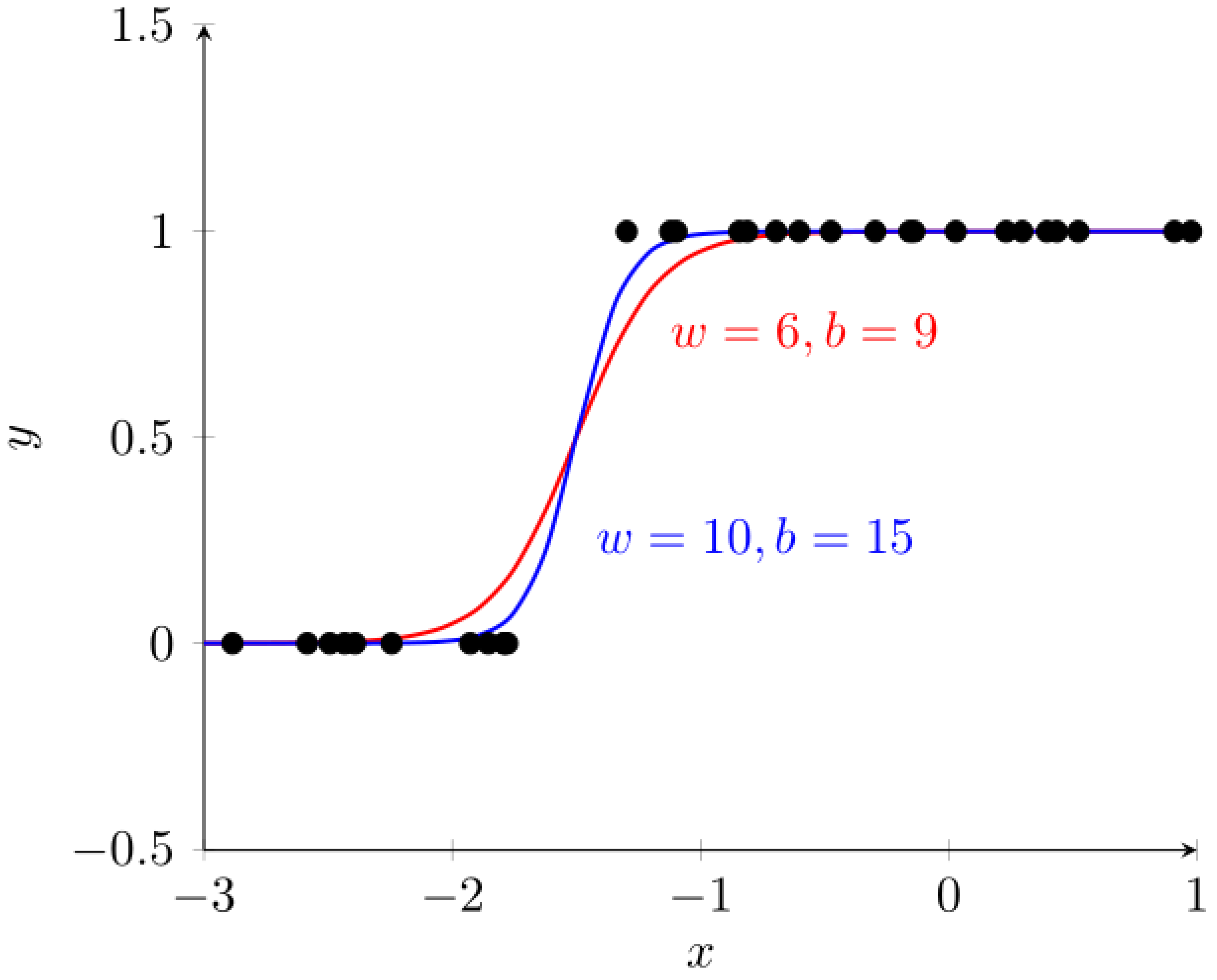
Logistic

Step 1: $f_{w,b}(x) = \sigma(wx + b) = \frac{1}{1 + exp(-wx + b)}$ Output: between 0 and 1

Step 2: Cross entropy: $l(f(x^n), \hat{y}^n) = -\left[\hat{y}^n ln f(x^n) + (1 - \hat{y}^n) ln\left(1 - f(x^n)\right)\right]$

Step 3: Logistic regression: $w_i \leftarrow w_i - \eta \sum_n -\left(\hat{y}^n - f_{w,b}(x^n)\right) x_i^n$

Logistic回归



Logistic

	Logistic Regression	***Linear Regression***
Step 1:	$f_{w,b}(x) = \sigma\left(\sum_i w_i x_i + b\right)$ Output: between 0 and 1	$f_{w,b}(x) = \sum_i w_i x_i + b$ Output: any value
Step 2:	Training data: $(x^n, \hat{y}^n)$ $\hat{y}^n$: 1 for class 1, 0 for class 2 $L(f) = \sum_n l(f(x^n), \hat{y}^n)$	Training data: $(x^n, \hat{y}^n)$ $\hat{y}^n$: a real number $L(f) = \frac{1}{2}\sum_n (f(x^n) - \hat{y}^n)^2$

Step 3:

Logistic regression: $w_i \leftarrow w_i - \eta \sum_n -\left(\underline{\hat{y}^n - f_{w,b}(x^n)}\right) x_i^n$

Linear regression: $w_i \leftarrow w_i - \eta \sum_n -\left(\underline{\hat{y}^n - f_{w,b}(x^n)}\right) x_i^n$

Logistic Regression + Square Error

Step 1: $f_{w,b}(x) = \sigma\left(\sum_i w_i x_i + b\right)$

Step 2: Training data: $(x^n, \hat{y}^n)$, $\hat{y}^n$: 1 for class 1, 0 for class 2

$$L(f) = \frac{1}{2}\sum_n \left(f_{w,b}(x^n) - \hat{y}^n\right)^2$$

Step 3: $\dfrac{\partial\,(f_{w,b}(x)-\hat{y})^2}{\partial w_i} = \left(f_{w,b}(x) - \hat{y}\right)\dfrac{\partial f_{w,b}(x)}{\partial z}\dfrac{\partial z}{\partial w_i}$

$$= \left(f_{w,b}(x) - \hat{y}\right) f_{w,b}(x)\left(1 - f_{w,b}(x)\right) x_i$$

$\hat{y}^n = 1$

If $f_{w,b}(x^n) = 1$ (close to target) ⇨ $\partial L/\partial w_i = 0$

If $f_{w,b}(x^n) = 0$ (far from target) ⇨ $\partial L/\partial w_i = 0$

Logistic Regression + Square Error

$$w_i \leftarrow w_i - \eta \sum_n -\left(\hat{y}^n - f_{w,b}(x^n)\right) x_i^n$$

Step 1: $f_{w,b}(x) = \sigma\left(\sum_i w_i x_i + b\right)$

Step 2: Training data: $(x^n, \hat{y}^n)$, $\hat{y}^n$: 1 for class 1, 0 for class 2

$$L(f) = \frac{1}{2}\sum_n \left(f_{w,b}(x^n) - \hat{y}^n\right)^2$$

Step 3: $\frac{\partial\,(f_{w,b}(x)-\hat{y})^2}{\partial w_i} = 2\left(f_{w,b}(x) - \hat{y}\right) \frac{\partial f_{w,b}(x)}{\partial z} \frac{\partial z}{\partial w_i}$

$$= 2\left(f_{w,b}(x) - \hat{y}\right) f_{w,b}(x)\left(1 - f_{w,b}(x)\right) x_i$$

$\hat{y}^n = 0$

If $f_{w,b}(x^n) = 1$ (far from target) → $\partial L/\partial w_i = 0$

If $f_{w,b}(x^n) = 0$ (close to target) → $\partial L/\partial w_i = 0$

逻辑回归—生成模型角度

$$P(C_1|x) = \frac{P(x|C_1)P(C_1)}{P(x|C_1)P(C_1) + P(x|C_2)P(C_2)}$$

$$= \frac{1}{1 + \frac{P(x|C_2)P(C_2)}{P(x|C_1)P(C_1)}} = \frac{1}{1 + exp(-z)} = \sigma(z)$$

Sigmoid function

$$z = ln\frac{P(x|C_1)P(C_1)}{P(x|C_2)P(C_2)}$$

逻辑回归—生成模型角度

$$P(C_1|x) = \sigma(z)$$

sigmoid

$$z = ln\frac{P(x|C_1)P(C_1)}{P(x|C_2)P(C_2)}$$

$$z = ln\frac{P(x|C_1)}{P(x|C_2)} + ln\boxed{\frac{P(C_1)}{P(C_2)}} \Rightarrow \frac{\frac{N_1}{N_1+N_2}}{\frac{N_2}{N_1+N_2}} = \frac{N_1}{N_2}$$

$$P(x|C_1) = \frac{1}{(2\pi)^{D/2}}\frac{1}{|\Sigma^1|^{1/2}}exp\left\{-\frac{1}{2}(x-\mu^1)^T(\Sigma^1)^{-1}(x-\mu^1)\right\}$$

$$P(x|C_2) = \frac{1}{(2\pi)^{D/2}}\frac{1}{|\Sigma^2|^{1/2}}exp\left\{-\frac{1}{2}(x-\mu^2)^T(\Sigma^2)^{-1}(x-\mu^2)\right\}$$

逻辑回归—生成模型角度

$$z = \underset{\boldsymbol{w}^T}{\underline{(\mu^1-\mu^2)^T\Sigma^{-1}}}x \underset{\mathrm{b}}{\underline{-\frac{1}{2}(\mu^1)^T\Sigma^{-1}\mu^1+\frac{1}{2}(\mu^2)^T\Sigma^{-1}\mu^2+ln\frac{N_1}{N_2}}}$$

生成模型

存在概率分布的假设

需要较少的数据就可以训练

对噪声可能更鲁棒

$P(y|x) \approx p(x)p(x|y)$

判别模型

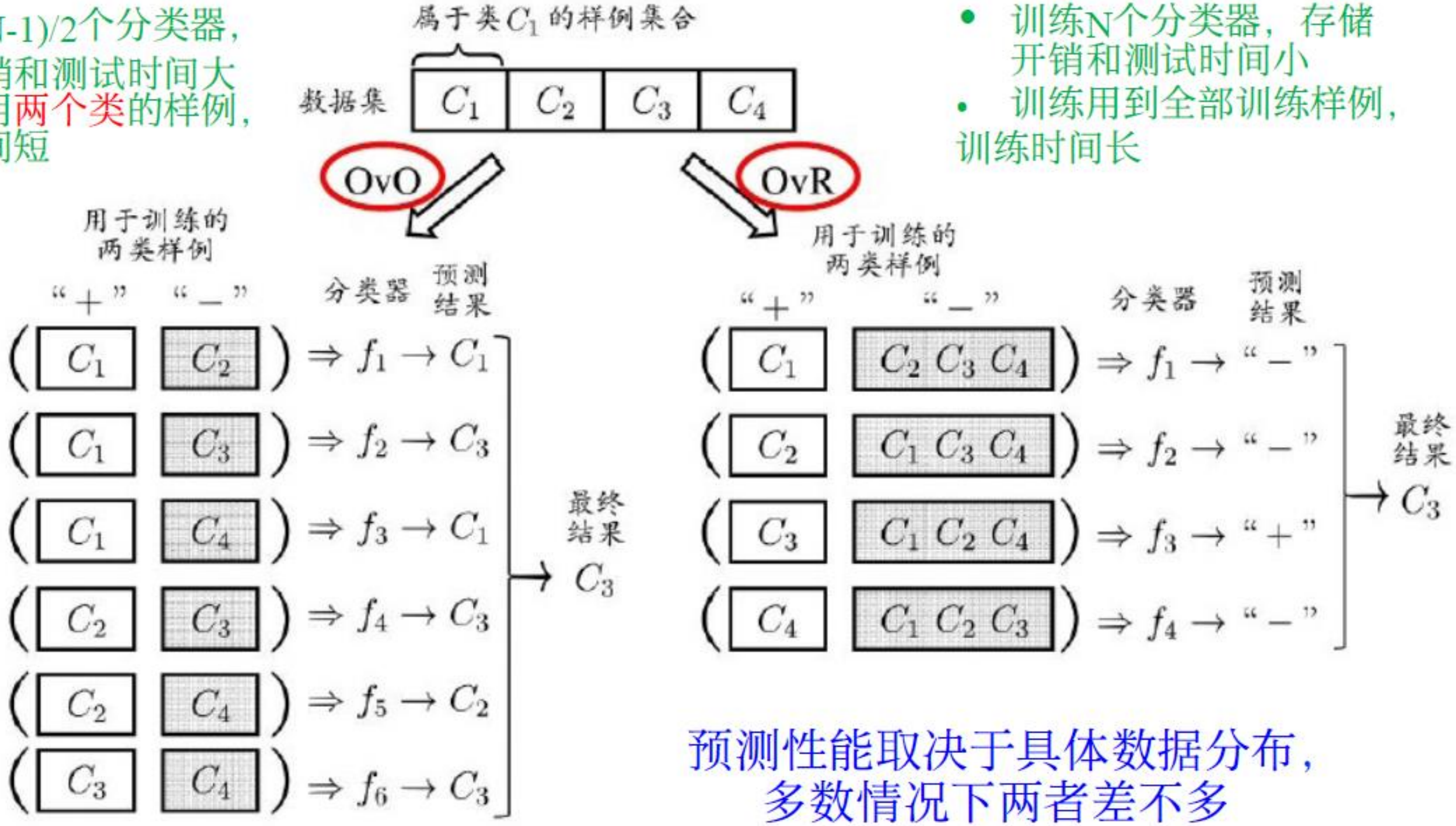
数据比较多时，相对要好

多分类

拆解法：将一个多分类任务拆分为若干个二分类任务求解

- 训练N(N-1)/2个分类器，存储开销和测试时间大
- 训练只用两个类的样例，训练时间短

- 训练N个分类器，存储开销和测试时间小
- 训练用到全部训练样例，训练时间长


属于类C_1的样例集合
数据集
C_1 C_2 C_3 C_4
OvO
OvR
用于训练的两类样例
“+” “−”
分类器
预测结果
$(C_1\ C_2) \Rightarrow f_1 \rightarrow C_1$
$(C_1\ C_3) \Rightarrow f_2 \rightarrow C_3$
$(C_1\ C_4) \Rightarrow f_3 \rightarrow C_1$
$(C_2\ C_3) \Rightarrow f_4 \rightarrow C_3$
$(C_2\ C_4) \Rightarrow f_5 \rightarrow C_2$
$(C_3\ C_4) \Rightarrow f_6 \rightarrow C_3$
最终结果
C_3
用于训练的两类样例
“+” “−”
分类器
预测结果
$(C_1\ C_2\ C_3\ C_4) \Rightarrow f_1 \rightarrow$ “−”
$(C_2\ C_1\ C_3\ C_4) \Rightarrow f_2 \rightarrow$ “−”
$(C_3\ C_1\ C_2\ C_4) \Rightarrow f_3 \rightarrow$ “+”
$(C_4\ C_1\ C_2\ C_3) \Rightarrow f_4 \rightarrow$ “−”
最终结果
C_3

预测性能取决于具体数据分布，多数情况下两者差不多

多分类

- **argmax方式**
  - “一对其余”方式的改进，仍需要**C个判别函数**
    - $\boldsymbol{f_c(x; w_c) = w_c x + b_c}, \quad \boldsymbol{c = [1, \cdots, C]}$
  - 如果存在类别c，对于其他所有类别$\tilde{c}(\tilde{c} \neq c)$都满足$\boldsymbol{f_c(x; w_c) > f_{\tilde{c}}(x; w_{\tilde{c}})}$，那么x属于类别c，即
    - $\boldsymbol{y = argmax_{c=1}^{C} f_c(x; w_c)}$
  - **Softmax函数**

多分类

C$_1$: w^1, b_1 $z_1 = w^1 \cdot x + b_1$

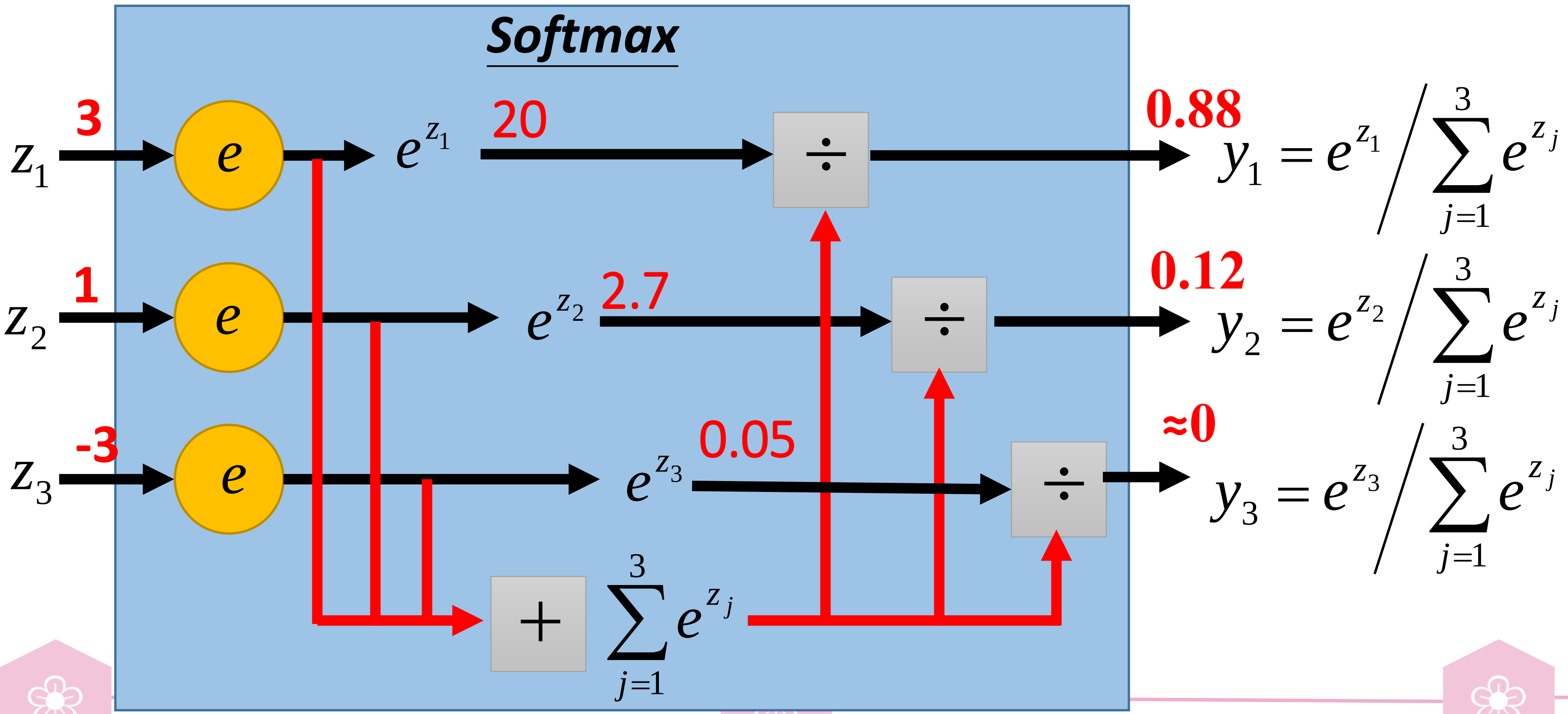
C$_2$: w^2, b_2 $z_2 = w^2 \cdot x + b_2$

C$_3$: w^3, b_3 $z_3 = w^3 \cdot x + b_3$

Probability:

- ■ $1 > y_i > 0$
- ■ $\sum_i y_i = 1$

$$y_i = P(C_i \mid x)$$

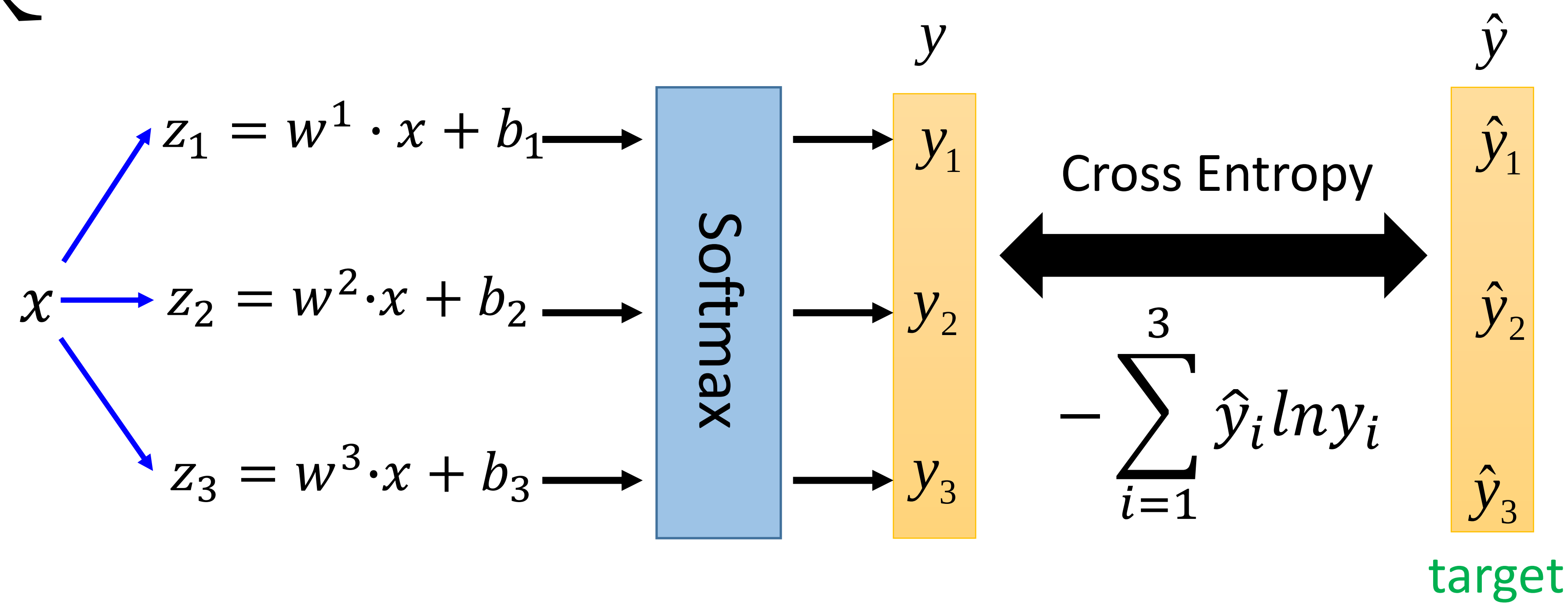


$$y_1 = e^{z_1} \Big/ \sum_{j=1}^{3} e^{z_j}$$

$$y_2 = e^{z_2} \Big/ \sum_{j=1}^{3} e^{z_j}$$

$$y_3 = e^{z_3} \Big/ \sum_{j=1}^{3} e^{z_j}$$

多分类



x

$z_1 = w^1 \cdot x + b_1$

$z_2 = w^2 \cdot x + b_2$

$z_3 = w^3 \cdot x + b_3$

Softmax

y

y_1 y_2 y_3

Cross Entropy

$$-\sum_{i=1}^{3} \hat{y}_i ln y_i$$

$\hat{y}$

$\hat{y}_1$ $\hat{y}_2$ $\hat{y}_3$

target

If x ∈ class 1

$$\hat{y} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

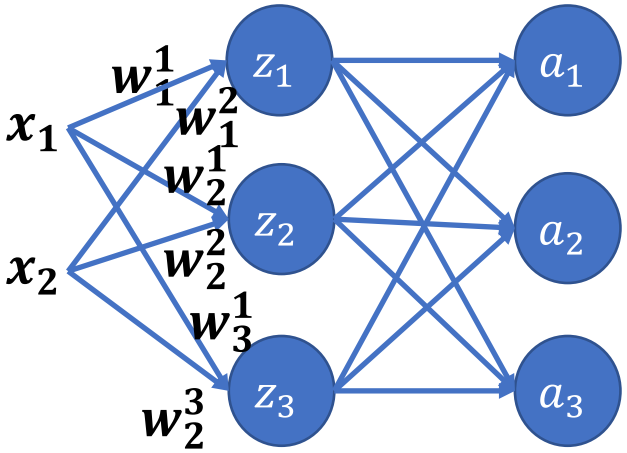
If x ∈ class 2

$$\hat{y} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

If x ∈ class 3

$$\hat{y} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

softmax



$$z_1 = w_1^1 x_1 + w_2^1 x_2$$

$$z_2 = w_1^2 x_1 + w_2^2 x_2$$

$$z_3 = w_1^3 x_1 + w_2^3 x_2$$

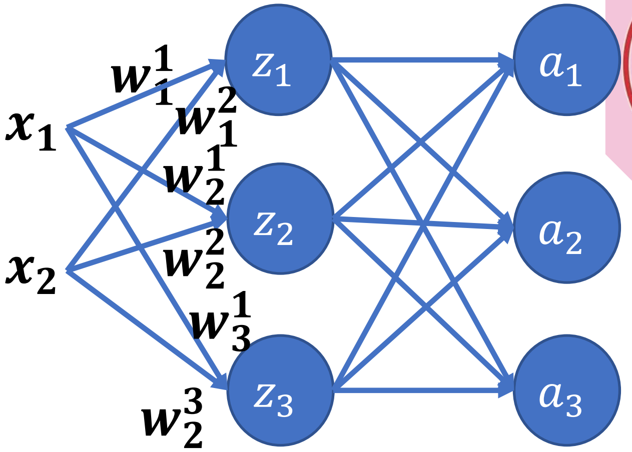
$$a_1 = \frac{e^{z_1}}{e^{z_1} + e^{z_2} + e^{z_3}}$$

$$a_2 = \frac{e^{z_2}}{e^{z_1} + e^{z_2} + e^{z_3}}$$

$$a_3 = \frac{e^{z_3}}{e^{z_1} + e^{z_2} + e^{z_3}}$$

$$\text{Loss} = -y_1 \ln a_1 - y_2 \ln a_2 - y_3 \ln a_3$$

softmax



$$z_1 = \boxed{w_1^1} x_1 + w_2^1 x_2$$

$$z_2 = w_1^2 x_1 + w_2^2 x_2$$

$$z_3 = w_1^3 x_1 + w_2^3 x_2$$

$$a_1 = \frac{e^{z_1}}{e^{z_1} + e^{z_2} + e^{z_3}}$$

$$a_2 = \frac{e^{z_2}}{e^{z_1} + e^{z_2} + e^{z_3}}$$

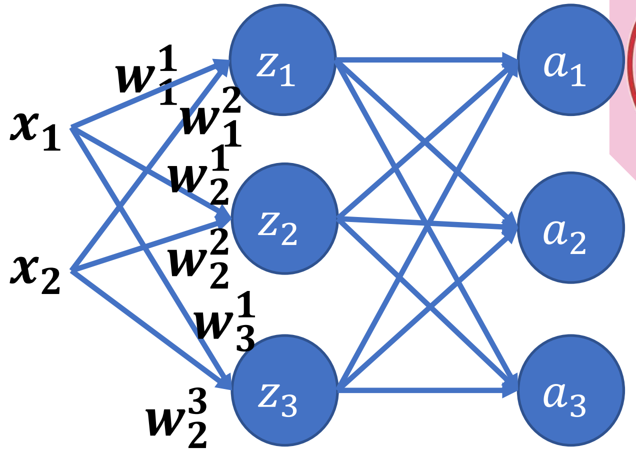
$$a_3 = \frac{e^{z_3}}{e^{z_1} + e^{z_2} + e^{z_3}}$$

$$\text{Loss} = -y_1 \ln a_1 - y_2 \ln a_2 - y_3 \ln a_3$$

$$\frac{\partial L}{\partial w_1^1} = \boxed{\frac{\partial L}{\partial a_1}} \cdot \frac{\partial a_1}{\partial z_1} \cdot \boxed{\frac{\partial z_1}{\partial w_1^1}} + \boxed{\frac{\partial L}{\partial a_2}} \cdot \frac{\partial a_2}{\partial z_1} \cdot \boxed{\frac{\partial z_1}{\partial w_1^1}} + \boxed{\frac{\partial L}{\partial a_3}} \cdot \frac{\partial a_3}{\partial z_1} \cdot \boxed{\frac{\partial z_1}{\partial w_1^1}}$$

$$-\frac{y_i}{a_i} \qquad x_1 \qquad \frac{\partial a_i}{\partial z_1} \begin{cases} \dfrac{\partial a_i}{\partial z_i} = \dfrac{e^{z_i}}{e^{z_1}+e^{z_2}+e^{z_3}} - \dfrac{e^{z_i} \cdot e^{z_i}}{(e^{z_1}+e^{z_2}+e^{z_3})^2} = a_i(1-a_i) \\[2ex] \dfrac{\partial a_i}{\partial z_j} (i \neq j) = -\dfrac{e^{z_i} \cdot e^{z_j}}{(e^{z_1}+e^{z_2}+e^{z_3})^2} = a_i a_j \end{cases}$$

softmax



$$z_1 = w_1^1 x_1 + w_2^1 x_2 \qquad a_1 = \frac{e^{z_1}}{e^{z_1} + e^{z_2} + e^{z_3}}$$

$$z_2 = w_1^2 x_1 + w_2^2 x_2 \qquad a_2 = \frac{e^{z_2}}{e^{z_1} + e^{z_2} + e^{z_3}}$$

$$z_3 = w_1^3 x_1 + w_2^3 x_2 \qquad a_3 = \frac{e^{z_3}}{e^{z_1} + e^{z_2} + e^{z_3}}$$

$$\text{Loss} = -y_1 \ln a_1 - y_2 \ln a_2 - y_3 \ln a_3$$

$$\frac{\partial L}{\partial w_1^1} = \boxed{\frac{\partial L}{\partial a_1} \cdot \frac{\partial a_1}{\partial z_1}} \cdot \frac{\partial z_1}{\partial w_1^1} + \boxed{\frac{\partial L}{\partial a_2} \cdot \frac{\partial a_2}{\partial z_1}} \cdot \frac{\partial z_1}{\partial w_1^1} + \boxed{\frac{\partial L}{\partial a_3} \cdot \frac{\partial a_3}{\partial z_1}} \cdot \frac{\partial z_1}{\partial w_1^1} \qquad = (a_1 - y_1)x_1$$

$$\frac{\partial L}{\partial z_i} = -\frac{y_i}{a_i} a_i(1 - a_i) + \sum_{i \neq j} \frac{y_j}{a_j} a_i a_j = -y_i + y_i a_i + a_i \sum_{i \neq j} y_j$$

$$= -y_i + a_i \sum_j y_j = a_i - y_i$$