



杭州电子科技大学
HANGZHOU DIANZI UNIVERSITY

篆 隶 行 草 隶 书 新



深度学习

线性模型
Linear Model

高飞 Fei Gao

gaofei@hdu.edu.cn

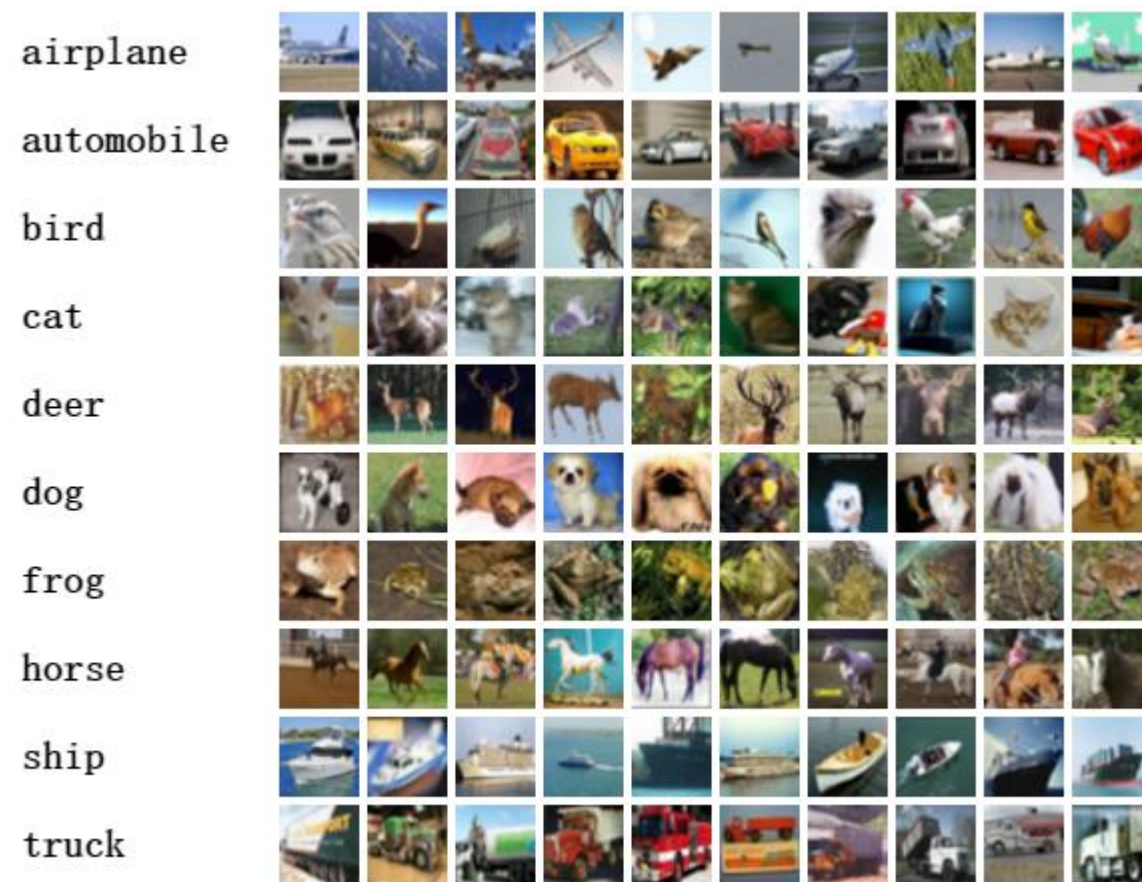
目 录

- 线性判别函数和决策边界
- Logistic Regression
- Softmax Regression
- Perceptron 感知器

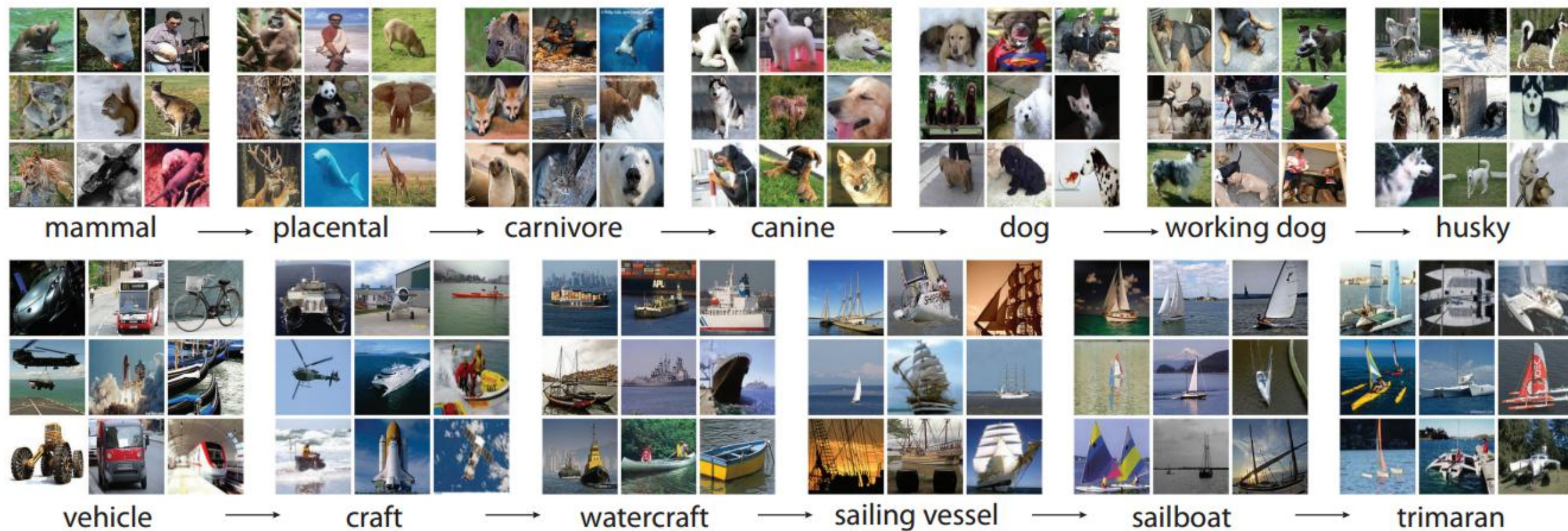
分类示例

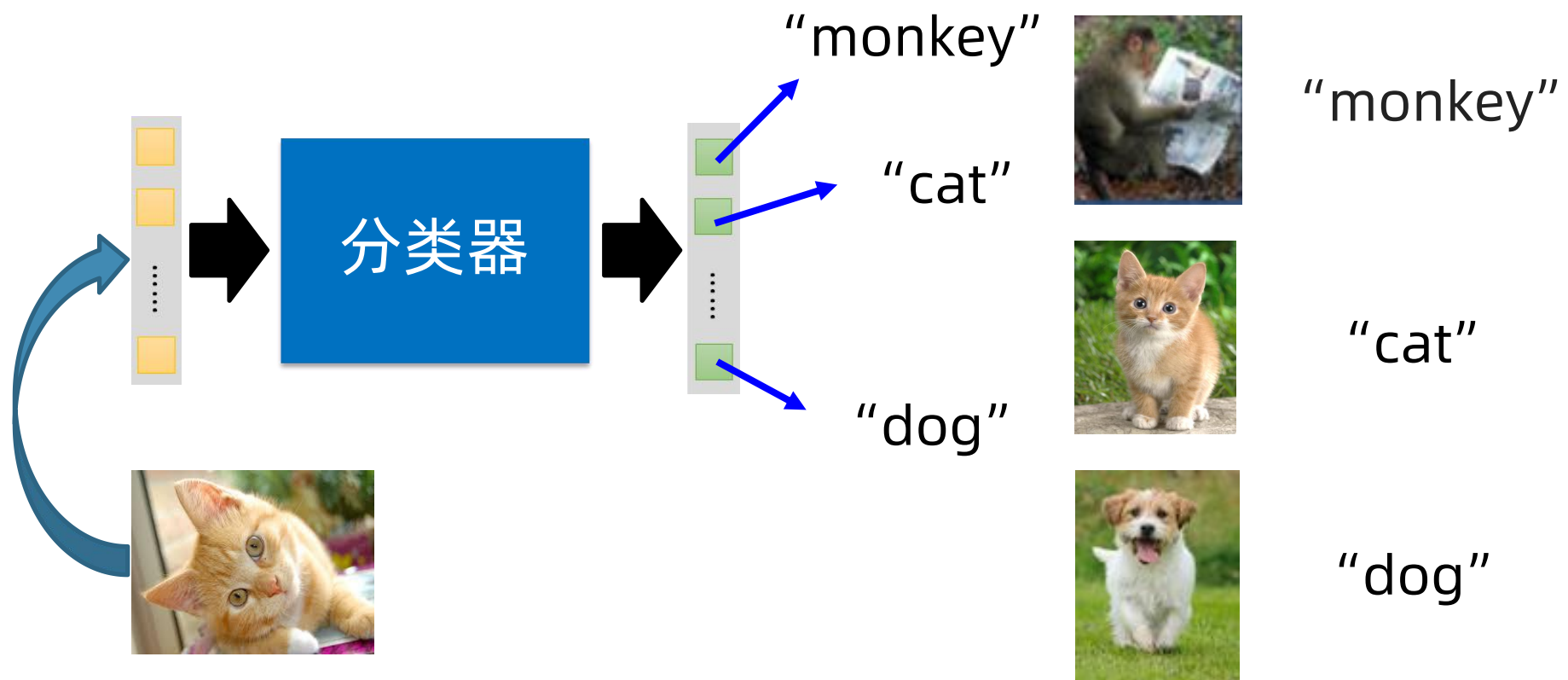
数据集：CIFAR-10

- 60000张32x32色彩图像，共10类，每类6000张图像。



- 14,197,122 images, 21841 synsets





根据文本内容来判断文本的相应类别

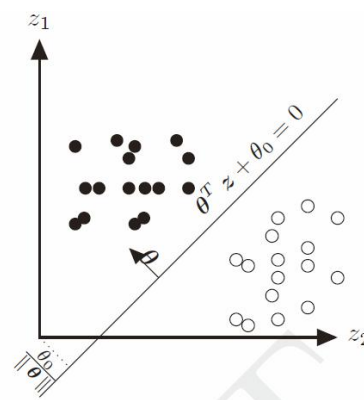
D_1 : “我喜欢读书”

D_2 : “我讨厌读书”

	我	喜欢	讨厌	读书
D_1	1	1	0	1
D_2	1	0	1	1

+

-





(<http://spam-filter-review.toptenreviews.com/>)



线性判别函数和决策边界

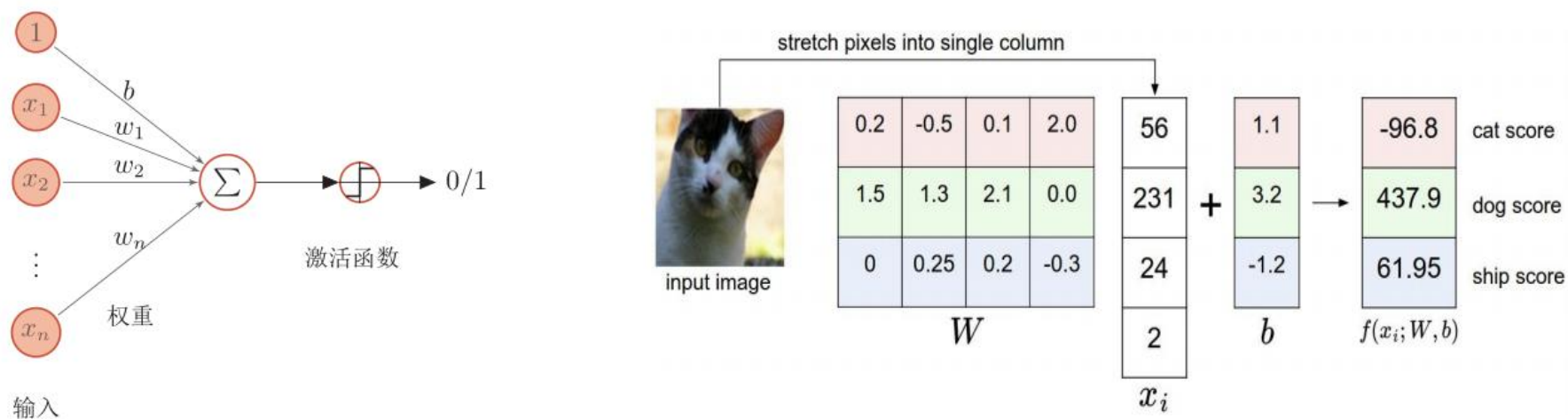
线性模型

线性模型 (Linear Model) 是机器学习中应用最广泛的模型，指通过样本特征的线性组合来进行预测的模型。给定一个 **d 维样本** $[x_1, \dots, x_d]^T$ ，其线性组合函数为

$$f(\mathbf{x}, \mathbf{w}) = w_1x_1 + w_2x_2 + \dots + w_dx_d + b \quad (3.1)$$

$$= \mathbf{w}^T \mathbf{x} + b, \quad (3.2)$$

其中 $\mathbf{w} = [w_1, \dots, w_d]^T$ 为 d 维的**权重向量**， b 为**偏置**。



线性模型

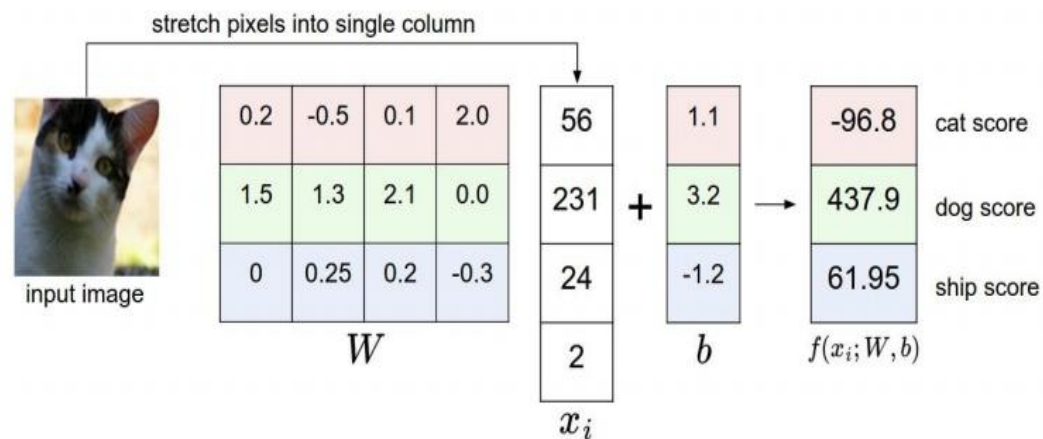
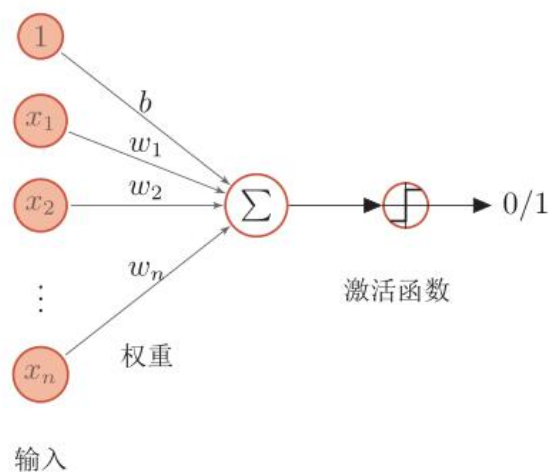
线性模型 (Linear Model) 是机器学习中应用最广泛的模型，指通过样本特征的线性组合来进行预测的模型。给定一个 **d 维样本** $[x_1, \dots, x_d]^T$ ，其线性组合函数为

$$f(\mathbf{x}, \mathbf{w}) = w_1x_1 + w_2x_2 + \dots + w_dx_d + b \quad (3.1)$$

$$= \mathbf{w}^T \mathbf{x} + b, \quad (3.2)$$

其中 $\mathbf{w} = [w_1, \dots, w_d]^T$ 为 d 维的**权重向量**， b 为**偏置**。

分类结果需要是离散值，怎么办？



在分类问题中，由于输出目标 y 是一些离散的标签，而 $f(\mathbf{x}, \mathbf{w})$ 的值域为实数，因此无法直接用 $f(\mathbf{x}, \mathbf{w})$ 来进行预测，需要引入一个非线性的决策函数 (decision function) $g(\cdot)$ 来预测输出目标

$$y = g(f(\mathbf{x}, \mathbf{w})), \quad (3.3)$$

其中 $f(\mathbf{x}, \mathbf{w})$ 也称为判别函数 (discriminant function)。

对于两类分类问题， $g(\cdot)$ 可以是符号函数 (sign function)

$$g(f(\mathbf{x}, \mathbf{w})) = \text{sgn}(f(\mathbf{x}, \mathbf{w})) \quad (3.4)$$

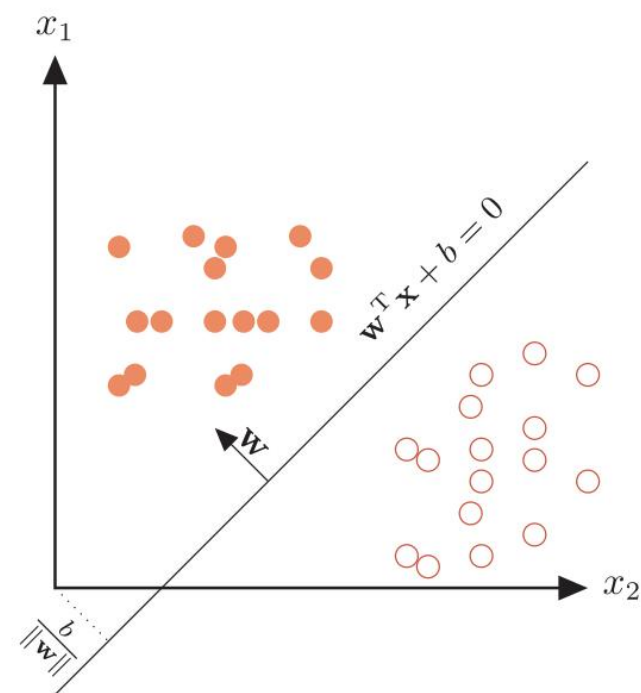
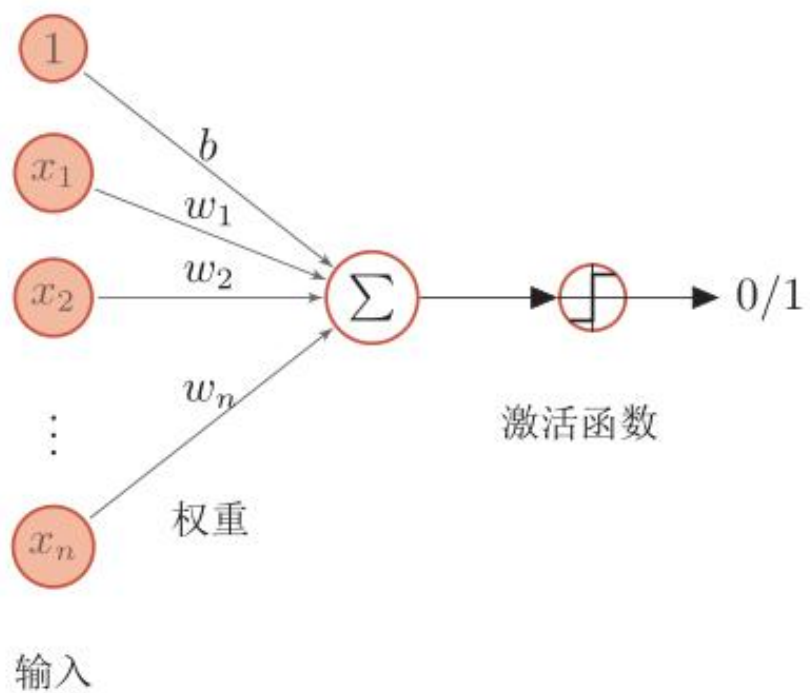
$$\triangleq \begin{cases} +1 & \text{if } f(\mathbf{x}, \mathbf{w}) > 0, \\ -1 & \text{if } f(\mathbf{x}, \mathbf{w}) < 0. \end{cases} \quad (3.5)$$

当 $f(\mathbf{x}, \mathbf{w}) = 0$ 时不进行预测。公式 (3.5) 定义了一个典型的两类分类问题的决策函数，其结构如图3.1所示。

两类线性分类模型（符号函数）

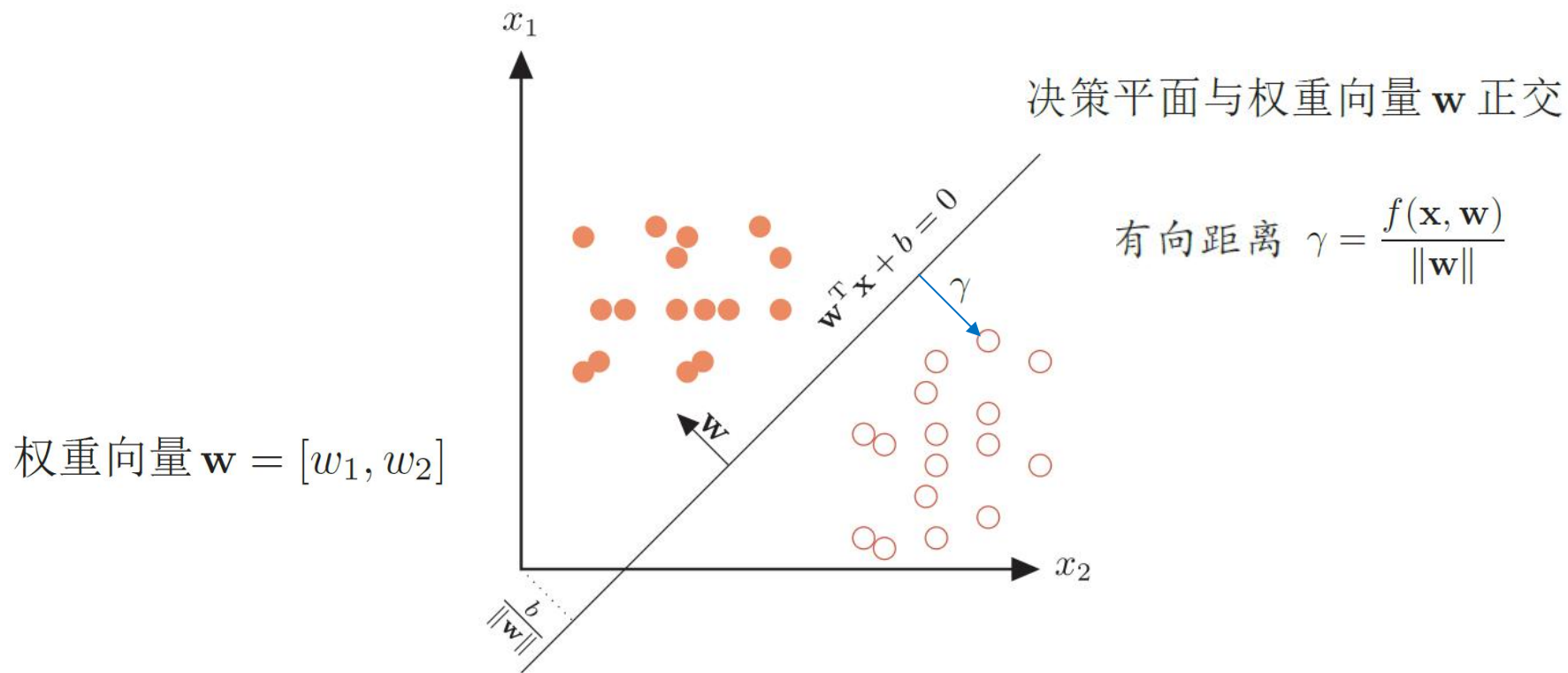
$$g(f(\mathbf{x}, \mathbf{w})) = \text{sgn}(f(\mathbf{x}, \mathbf{w})) \triangleq \begin{cases} +1 & \text{if } f(\mathbf{x}, \mathbf{w}) > 0, \\ -1 & \text{if } f(\mathbf{x}, \mathbf{w}) < 0. \end{cases}$$

$$f(\mathbf{x}, \mathbf{w}) = \mathbf{w}^T \mathbf{x} + b.$$

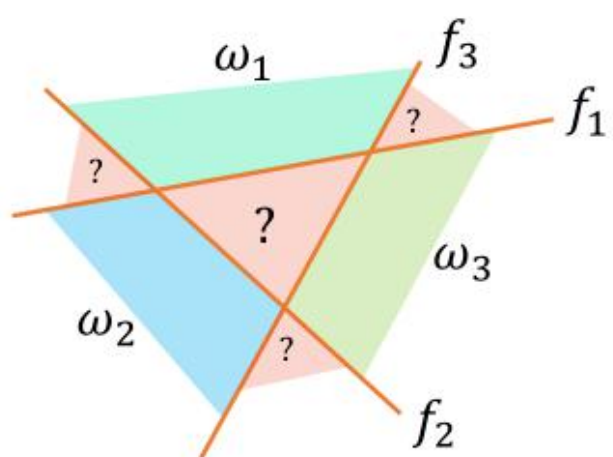


决策边界/决策平面

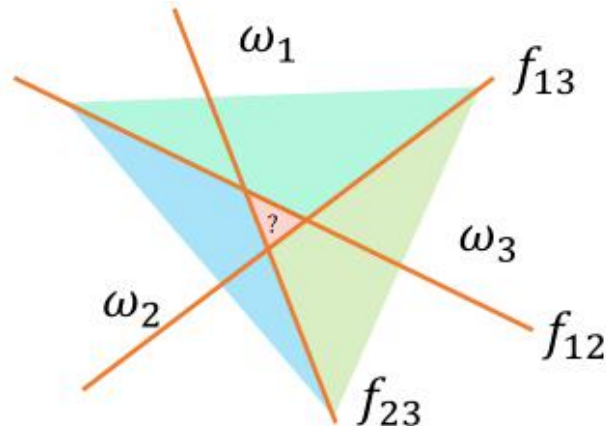
在两个分类中，我们只需要一个线性判别函数 $f(\mathbf{x}, \mathbf{w}) = \mathbf{w}^T \mathbf{x} + b$ 。特征空间 \mathbb{R}^d 中所有满足 $f(\mathbf{x}, \mathbf{w}) = 0$ 的点组成用一个分割超平面（hyperplane），称为决策边界（decision boundary）或决策平面（decision surface）。决策边界将特征空间一分为二，划分成两个区域，每个区域对应一个类别。



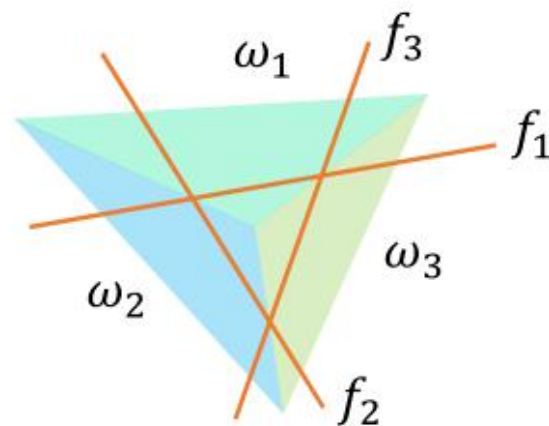
多类分类 (Multi-class Classification)



(a) “一对其余” 方式



(b) “一对一” 方式



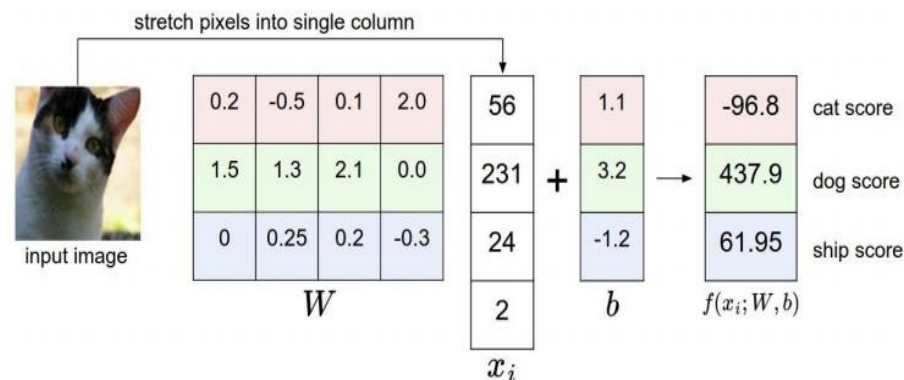
(c) “argmax” 方式

“argmax” 方式：这是一种改进的“一对其余”方式，共需要 C 个判别函数

$$f_c(\mathbf{x}; \mathbf{w}_c) = \mathbf{w}_c^T \mathbf{x} + b_c, \quad c = [1, \dots, C] \quad (3.10)$$

如果存在类别 c , 对于所有的其他类别 $\tilde{c} (\tilde{c} \neq c)$ 都满足 $f_c(\mathbf{x}; \mathbf{w}_c) > f_{\tilde{c}}(\mathbf{x}, \mathbf{w}_{\tilde{c}})$, 那么 \mathbf{x} 属于类别 c 。即

$$y = \arg \max_{c=1}^C f_c(\mathbf{x}; \mathbf{w}_c). \quad (3.11)$$



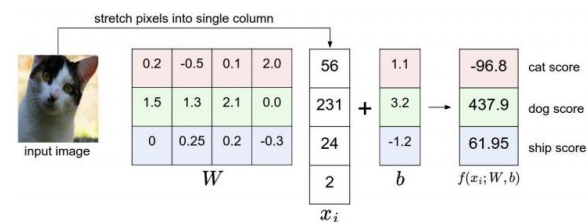
- 将分类问题看作条件概率估计问题

- 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 来预测类别标签的后验概率 $p(y = c|x)$ 。
- 以两类分类为例，

$$p(y = 1|x) = g(f(\mathbf{x}; \mathbf{w}))$$

- 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。

如何构建函数 g ?

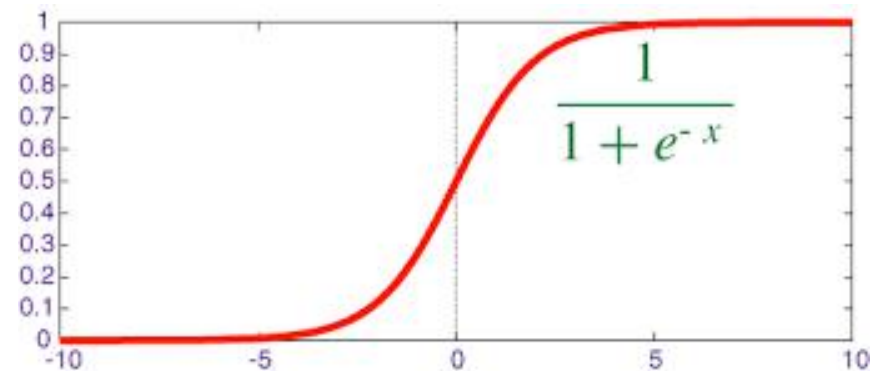


逻辑回归

Logistic Regression

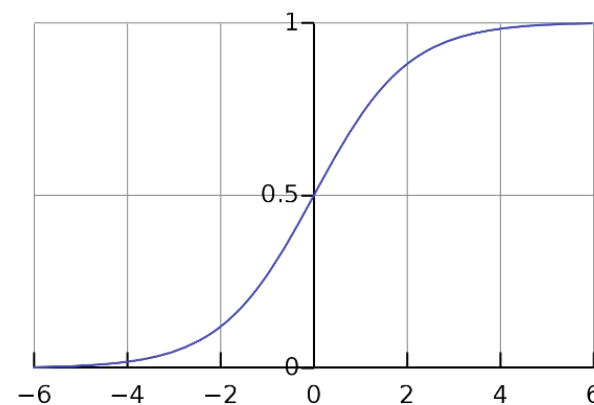
- Logistic函数

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



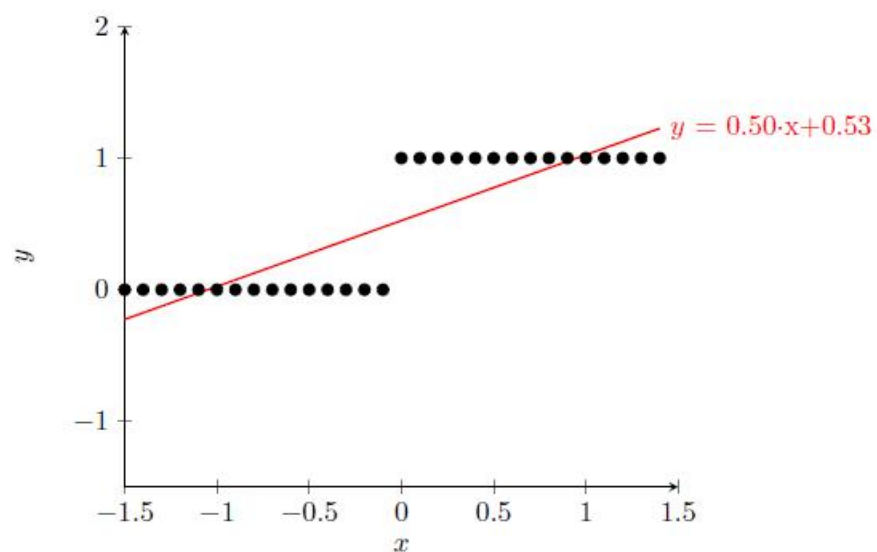
- Logistic回归

$$p(y = 1|\mathbf{x}) = \sigma(\mathbf{w}^T \mathbf{x}) \triangleq \frac{1}{1 + \exp(-\mathbf{w}^T \mathbf{x})}$$



线性分类器

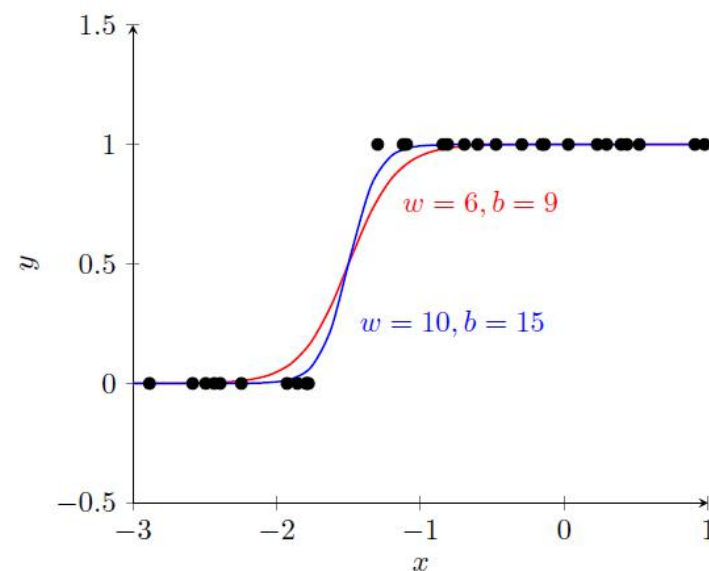
$$\hat{y} = \begin{cases} 1 & \text{if } \mathbf{w}^T \mathbf{x} > 0 \\ 0 & \text{if } \mathbf{w}^T \mathbf{x} \leq 0 \end{cases}$$



(a) 线性回归

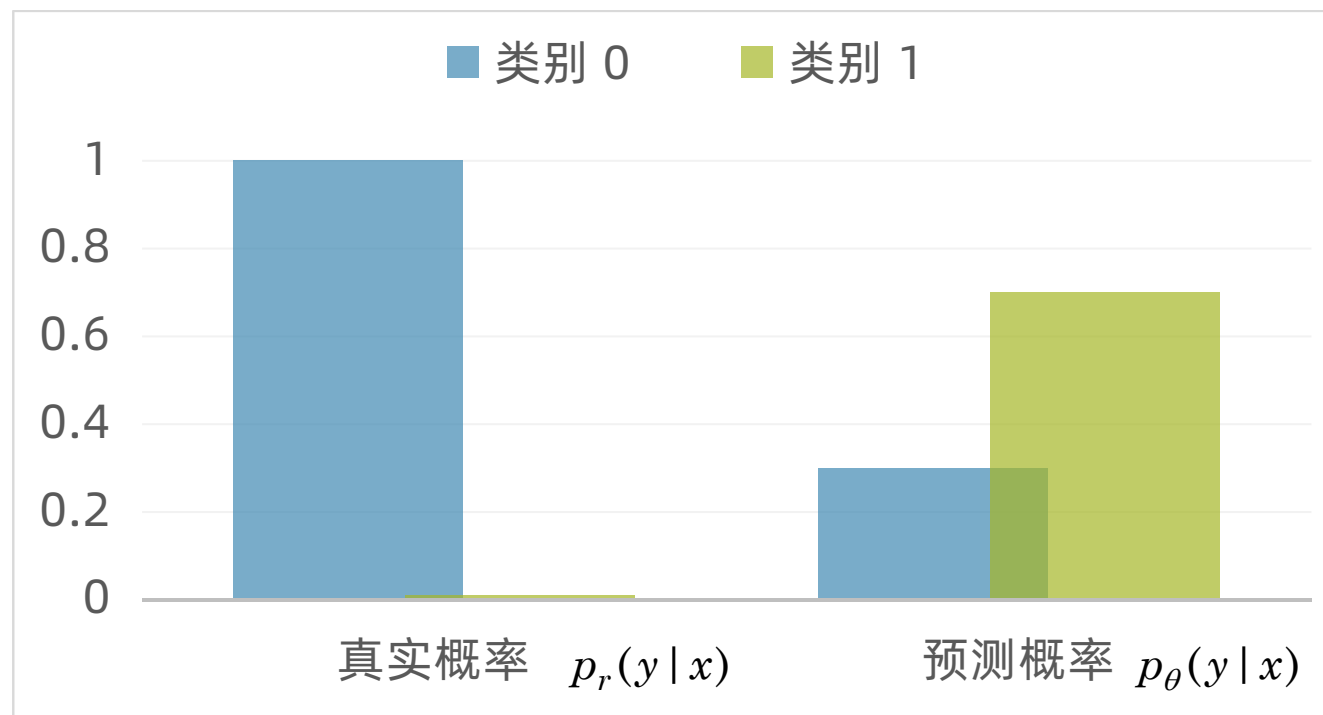
Logistic分类器

$$P(y = 1 | \mathbf{x}) = \sigma(\mathbf{w}^T \mathbf{x})$$
$$\triangleq \frac{1}{1 + \exp(-\mathbf{w}^T \mathbf{x})},$$



(b) Logistic 回归

- 分类任务：真实概率 vs. 预测概率



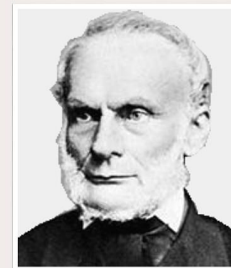
如何衡量两个条件分布的差异？

- 在信息论中，熵用来衡量一个随机事件的不确定性
- 自信息 (Self Information)

$$I(x) = -\log(p(x))$$

- 熵

$$\begin{aligned} H(X) &= \mathbb{E}_X[I(x)] \\ &= \mathbb{E}_X[-\log(p(x))] \\ &= -\sum_{x \in \mathcal{X}} p(x) \log p(x) \end{aligned}$$



Rudolf Julius
Emanuel Clausius

热力学第二定律

Entropy

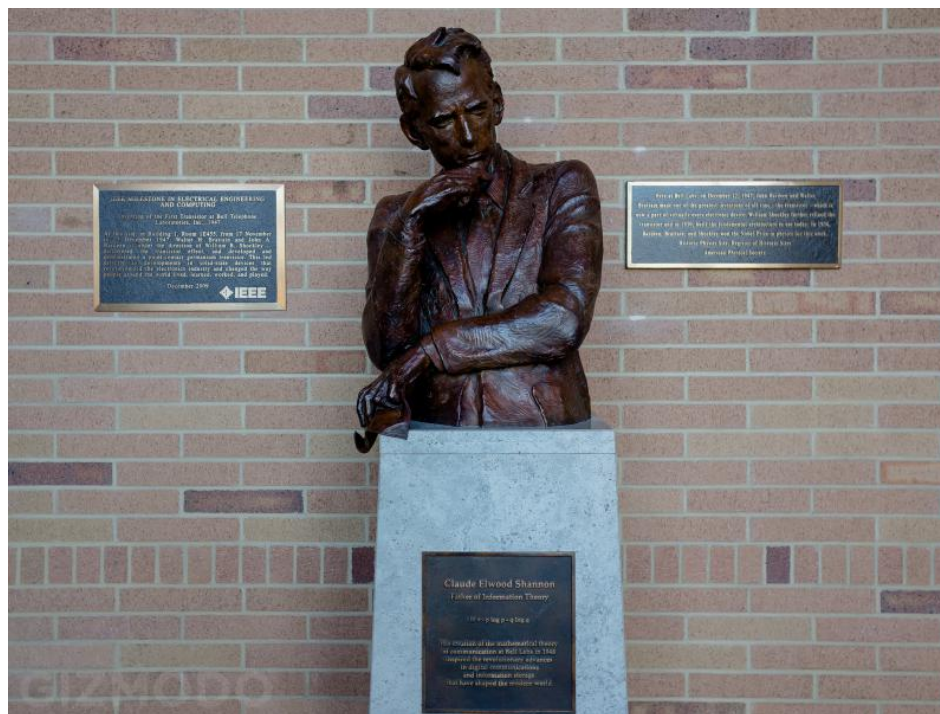
熵

香农创造性的引入
“信息熵”
解决了对信息的量化问题

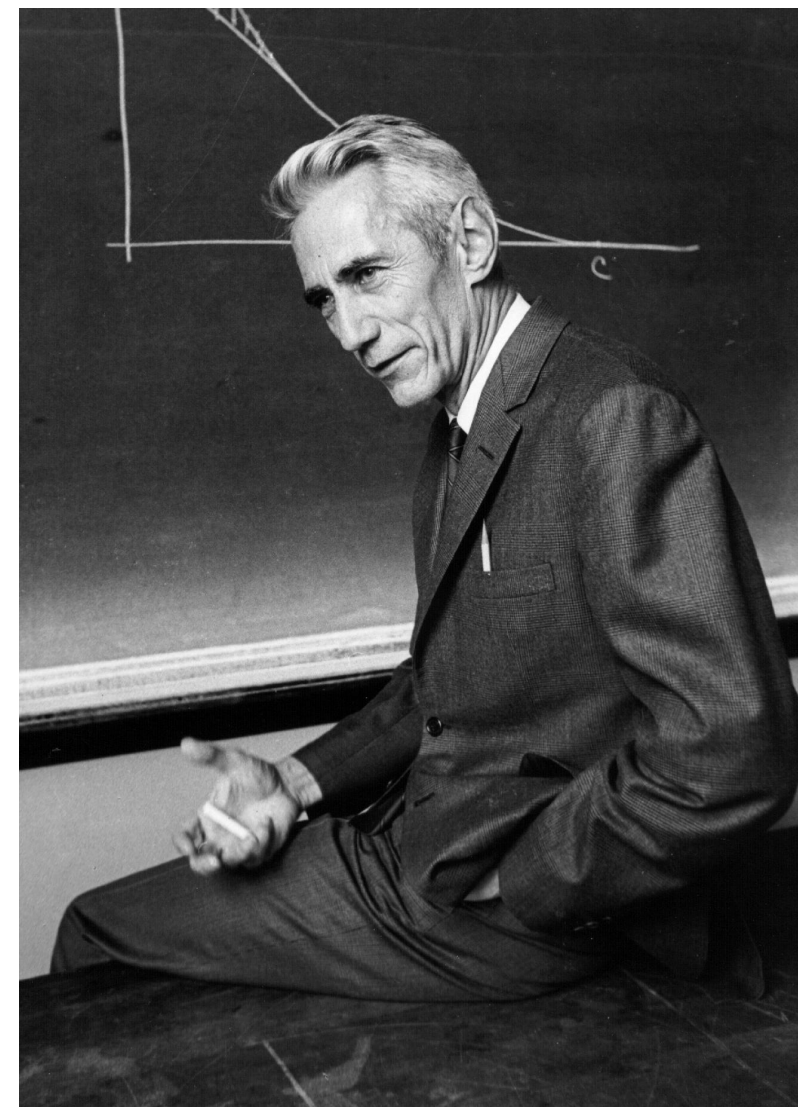
IEEE INFORMATION THEORY SOCIETY
SCIE.CQUPT

- 克劳德·香农 Claude E. Shannon

- 生卒年：1916-2001
- 信息论之父



贝尔实验室雕像



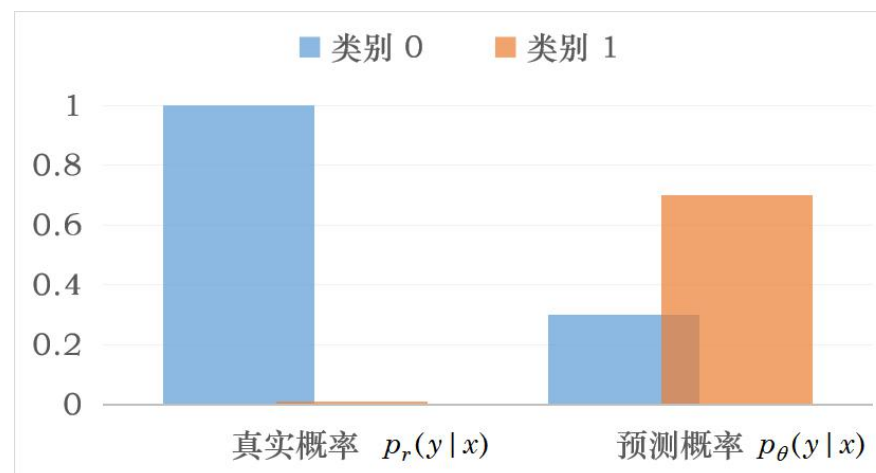
交叉熵 (Cross Entropy)

- 交叉熵是按照概率分布 q 的最优编码对真实分布为 p 的信息进行编码的长度。

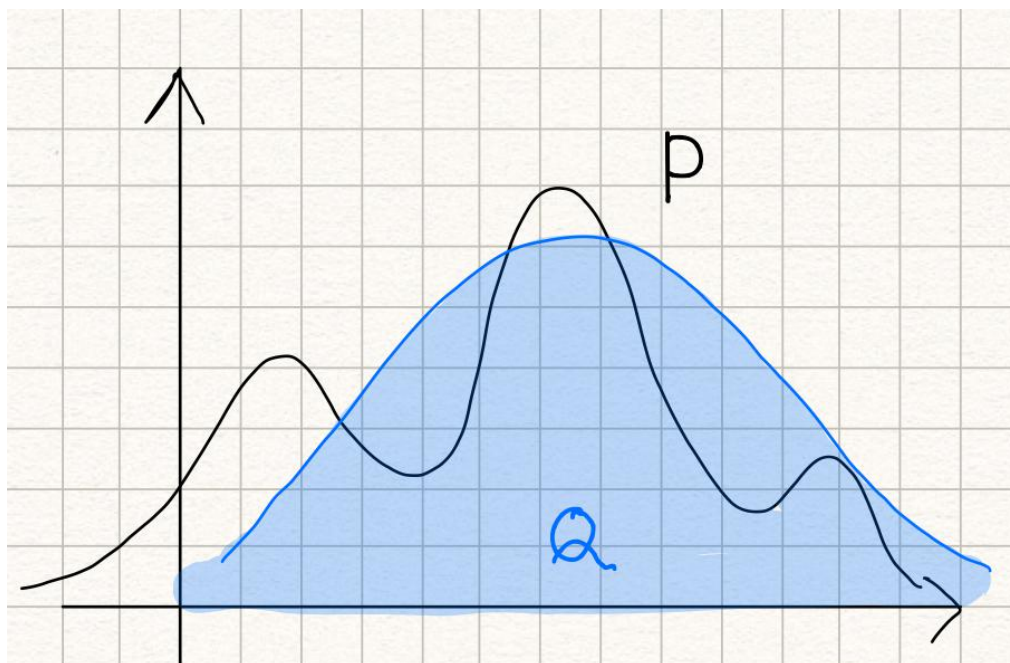
$$\begin{aligned} H(p, q) &= \mathbb{E}_p[-\log q(x)] \\ &= -\sum_x p(x) \log q(x) \end{aligned}$$

- 二分类问题中

$$H(p, q) = \begin{cases} ? & \text{if } p = q \\ ? & \text{if } p = [1, 0], q = [0.3, 0.7] \end{cases}$$



- KL散度是用概率分布q来近似p时所造成的信息损失量。
 - KL散度是按照概率分布q的最优编码对真实分布为p的信息进行编码，其平均编码长度（即交叉熵） $H(p,q)$ 和p的最优平均编码长度（即熵） $H(p)$ 之间的差异。



$$D_{KL}(P||Q) = \sum_i P(i) \log \frac{P(i)}{Q(i)}$$
$$D_{KL}(P||Q) = \int P(x) \log \frac{P(x)}{Q(x)} dx$$

- 交叉熵损失 → 负对数似然

$$D_{\text{kl}}(p_r(y|x) || p_\theta(y|x)) = \sum_{y=0}^1 p_r(y|x) \log \frac{p_r(y|x)}{p_\theta(y|x)} \quad \text{KL散度}$$

$$\propto - \sum_{y=0}^1 p_r(y|x) \log p_\theta(y|x) \quad \text{交叉熵损失}$$

y^* 为 x 的真实标签

$$= -I(y^* = 1) \log p_\theta(y = 1|x) - I(y^* = 0) \log p_\theta(y = 0|x)$$

$$= -y^* \log p_\theta(y = 1|x) - (1 - y^*) \log p_\theta(y = 0|x)$$

$$= -\log p_\theta(y^*|x) \quad \text{负对数似然}$$

- 负对数似然损失函数

$$\mathcal{L}(\mathbf{y}, f(\mathbf{x}, \theta)) = - \sum_{c=1}^C y_c \log f_c(\mathbf{x}, \theta)$$

- 对于一个三类分类问题，类别为[0,0,1]，预测的类别概率为[0.3,0.3,0.4]，则

Ex:

Computed (\hat{y})	Targets (y)
[0.3, 0.3, 0.4]	[0, 0, 1]

$$\mathcal{L}(\theta) =$$

- 负对数似然损失函数

$$\mathcal{L}(\mathbf{y}, f(\mathbf{x}, \theta)) = - \sum_{c=1}^C y_c \log f_c(\mathbf{x}, \theta)$$

- 对于一个三类分类问题，类别为[0,0,1]，预测的类别概率为[0.3,0.3,0.4]，则

Ex:

Computed (\hat{y})	Targets (y)
[0.3, 0.3, 0.4]	[0, 0, 1]

$$\begin{aligned}\mathcal{L}(\theta) &= -(0 \times \log(0.3) + 0 \times \log(0.3) + 1 \times \log(0.4)) \\ &= -\log(0.4).\end{aligned}$$

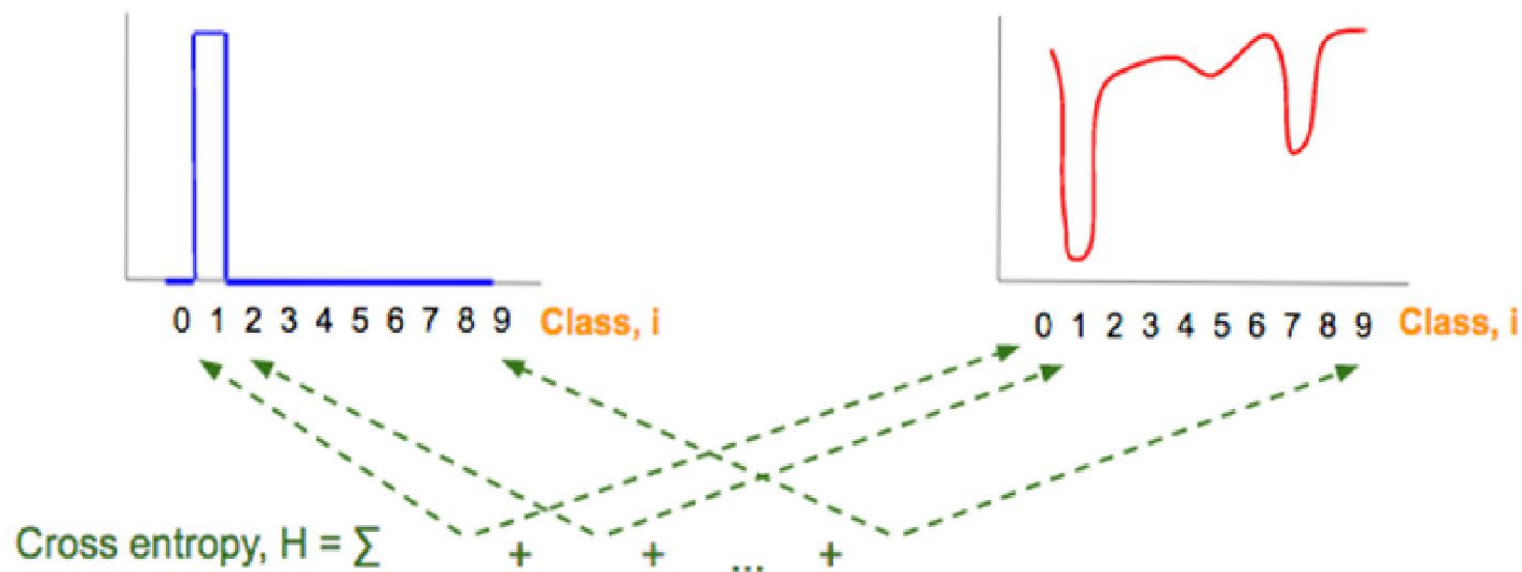
交叉熵损失

- 交叉熵损失 → 负对数似然

$$-\sum_{y=1}^c p_r(y|x) \log p_{\theta}(y|x)$$

真实概率 $p_r(y|x)$

预测概率的负对数 $-\log p_{\theta}(y|x)$



- 交叉熵损失函数，模型在训练集的风险函数为

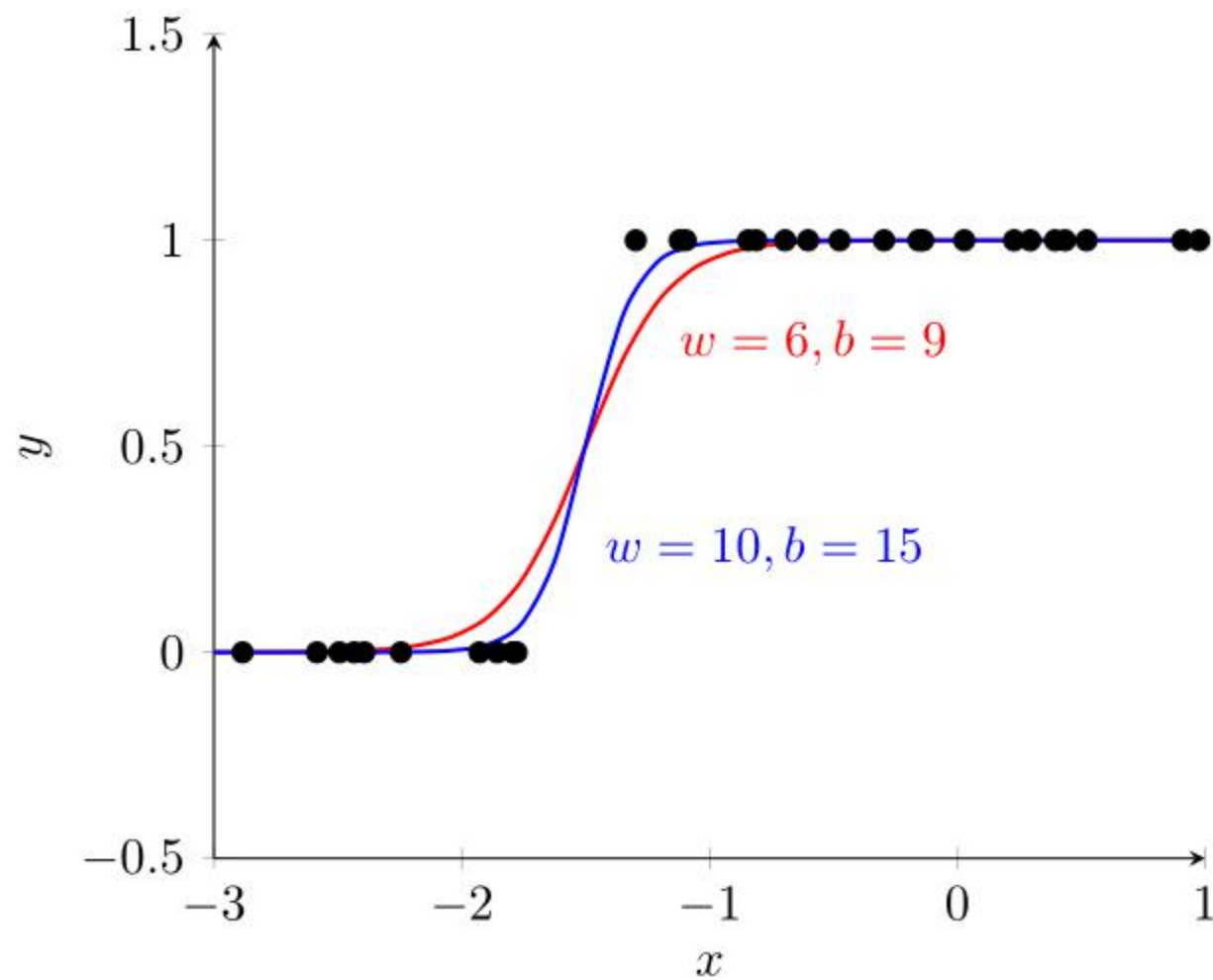
$$\mathcal{R}(\mathbf{w}) = -\frac{1}{N} \sum_{i=1}^N \left(y^{(i)} \log \left(\sigma(\mathbf{w}^T \mathbf{x}^{(i)}) \right) + (1 - y^{(i)}) \log \left(1 - \sigma(\mathbf{w}^T \mathbf{x}^{(i)}) \right) \right).$$

- 梯度为

$$\frac{\partial \mathcal{R}(\mathbf{w})}{\partial \mathbf{w}} = \frac{1}{N} \sum_{i=1}^N \left(\mathbf{x}^{(i)} \cdot \left(\sigma(\mathbf{w}^T \mathbf{x}^{(i)}) - y^{(i)} \right) \right)$$

- 推导

$$\begin{aligned} \frac{\partial \mathcal{R}(\mathbf{w})}{\partial \mathbf{w}} &= -\frac{1}{N} \sum_{n=1}^N \left(y^{(n)} \frac{\hat{y}^{(n)}(1 - \hat{y}^{(n)})}{\hat{y}^{(n)}} \mathbf{x}^{(n)} - (1 - y^{(n)}) \frac{\hat{y}^{(n)}(1 - \hat{y}^{(n)})}{1 - \hat{y}^{(n)}} \mathbf{x}^{(n)} \right) \\ &= -\frac{1}{N} \sum_{n=1}^N \left(y^{(n)} (1 - \hat{y}^{(n)}) \mathbf{x}^{(n)} - (1 - y^{(n)}) \hat{y}^{(n)} \mathbf{x}^{(n)} \right) \\ &= -\frac{1}{N} \sum_{n=1}^N \mathbf{x}^{(n)} (y^{(n)} - \hat{y}^{(n)}). \end{aligned}$$



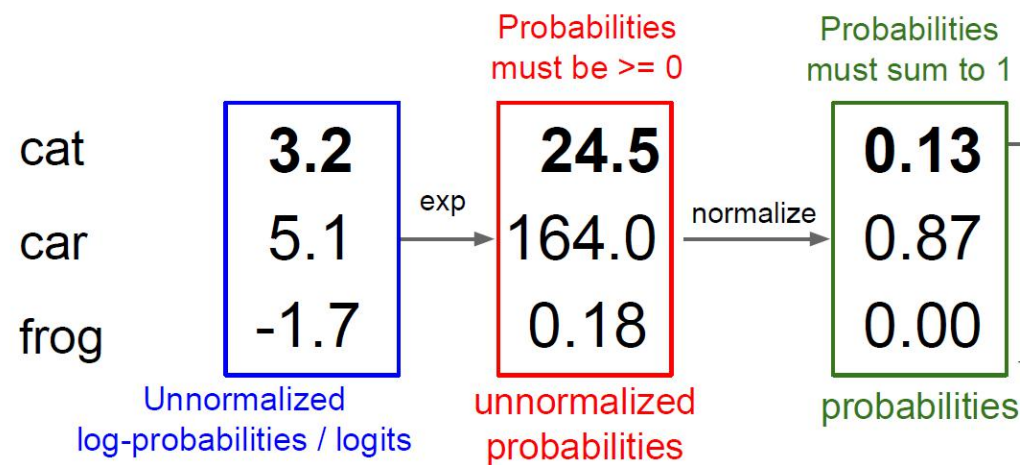
Softmax Regression

Softmax函数

$$\text{softmax}(x_k) = \frac{\exp(x_k)}{\sum_{i=1}^K \exp(x_i)}$$

Softmax回归

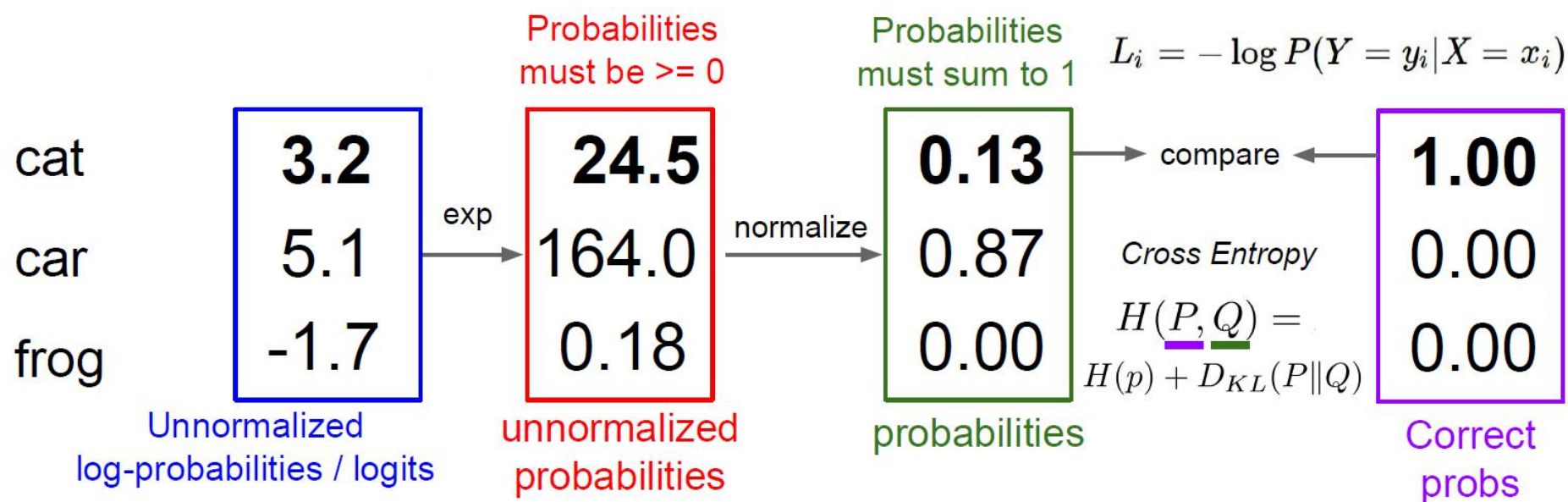
$$P(y = c|\mathbf{x}) = \text{softmax}(\mathbf{w}_c^T \mathbf{x}) \\ = \frac{\exp(\mathbf{w}_c^T \mathbf{x})}{\sum_{i=1}^C \exp(\mathbf{w}_i^T \mathbf{x})}$$



- Softmax回归是logistic回归的多类推广: $\hat{y} = \arg \max_{c=1}^C \mathbf{w}_c^T \mathbf{x}$

- 交叉熵损失 Cross Entropy Loss

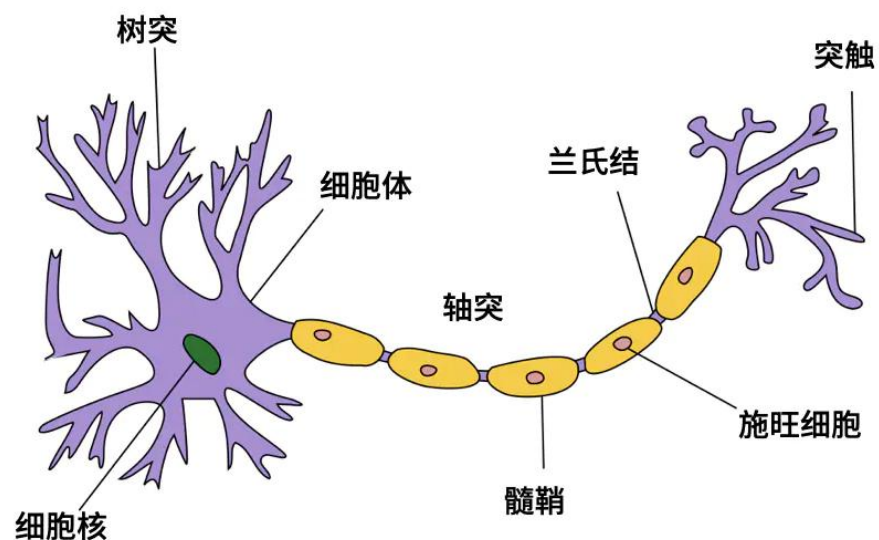
$$\mathcal{R}(W) = -\frac{1}{N} \sum_{n=1}^N (\mathbf{y}^{(n)})^T \log \hat{\mathbf{y}}^{(n)}$$



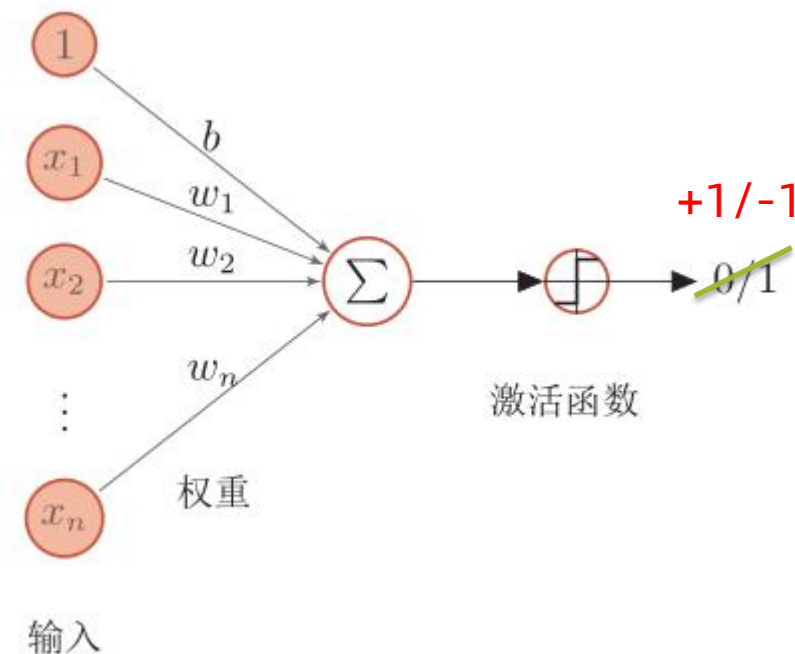
Perceptron 感知器

- 模拟生物神经元行为的机器，有与生物神经元相对应的部件，如权重（突触）、偏置（阈值）及激活函数（细胞体），输出为+1或-1。

标准神经元结构



$$\hat{y} = \begin{cases} +1 & \text{当 } \mathbf{w}^T \mathbf{x} > 0 \\ -1 & \text{当 } \mathbf{w}^T \mathbf{x} \leq 0 \end{cases},$$



- 一种错误驱动的在线学习方法

1. 初始化权重向量

2. 每分错一个样本时更新权重

$$\mathbf{w} \leftarrow \mathbf{w} + y\mathbf{x}$$

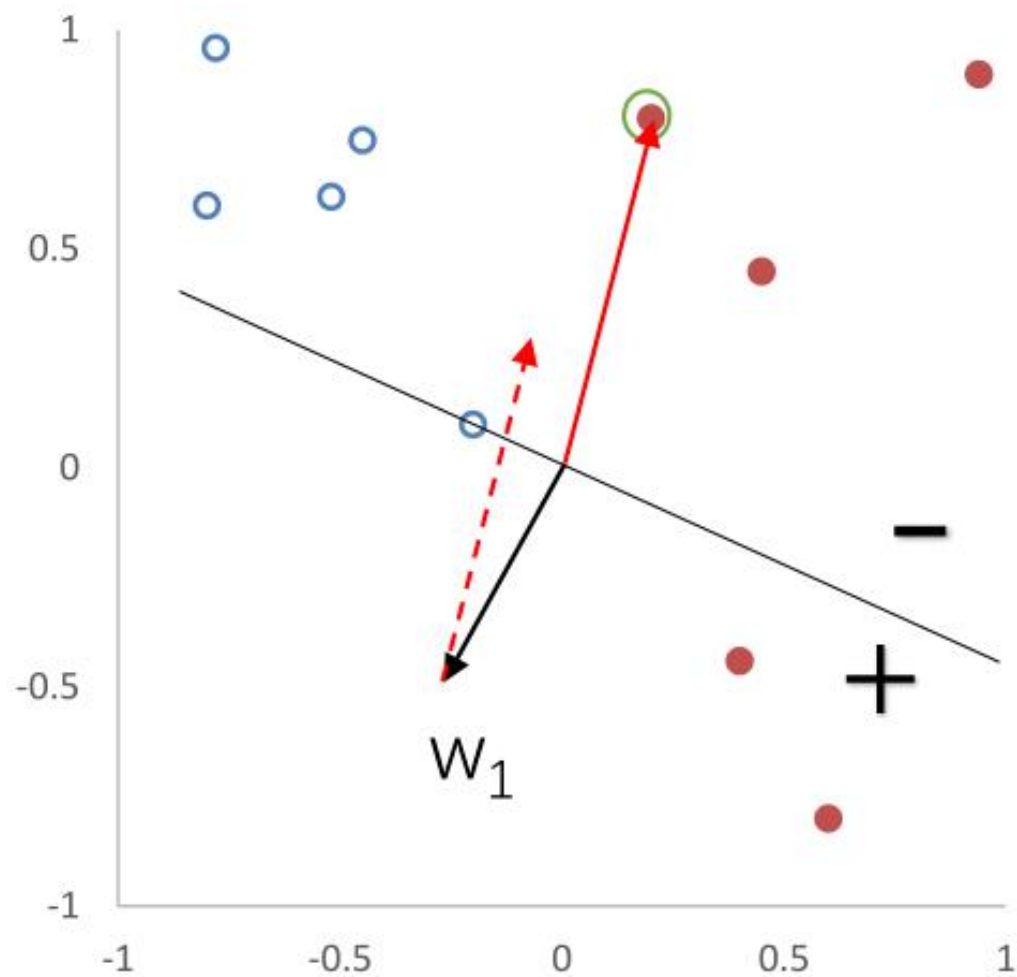
输入: 训练集: $(\mathbf{x}_i, y_i), i = 1, \dots, N$, 迭代次数: T

```
1 初始化:  $\mathbf{w}_0 = 0$  ;  
2  $k = 0$  ;  
3 for  $t = 1 \dots T$  do  
4   for  $i = 1 \dots N$  do  
5     选取一个样本  $(\mathbf{x}_i, y_i)$ , if  $\mathbf{w}^T(y_i\mathbf{x}_i) < 0$  then  
6        $\mathbf{w}_{k+1} = \mathbf{w}_k + y_i\mathbf{x}_i$  ;  
7        $k = k + 1$ ;  
8     end  
9   end  
10 end  
    输出:  $\mathbf{w}_k$ 
```

表示分错

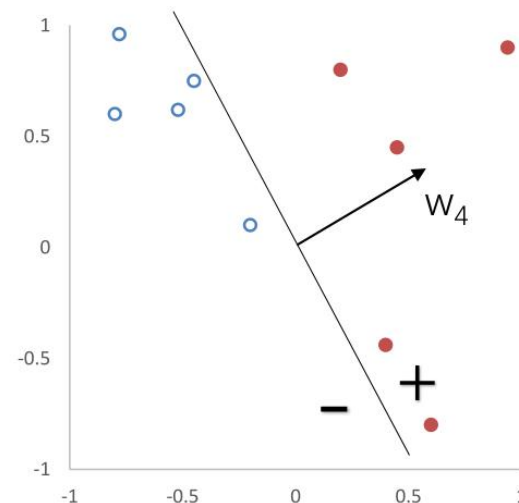
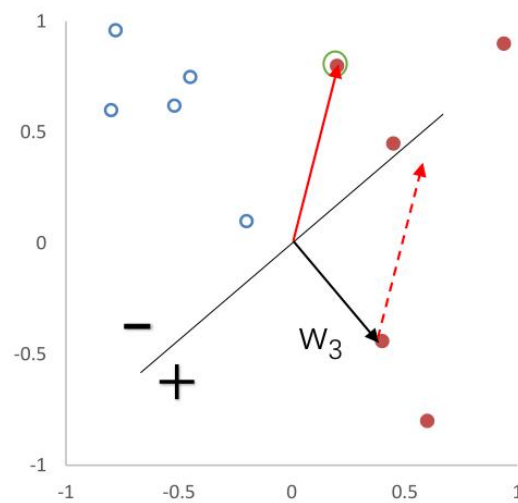
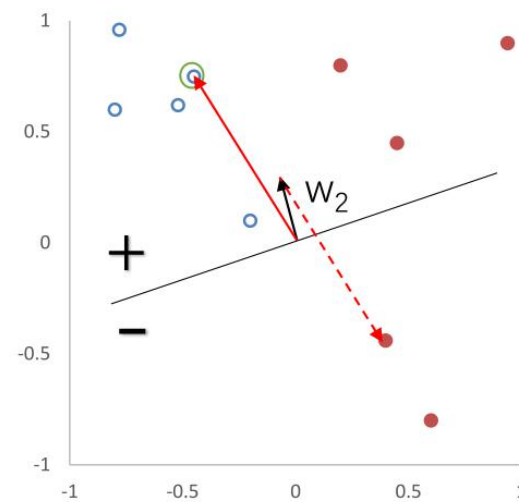
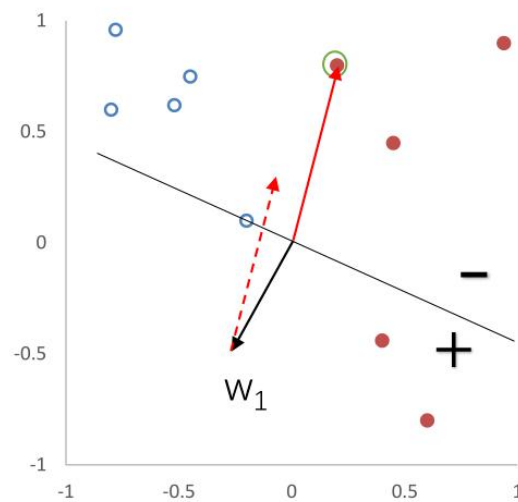
$$\hat{y} = \begin{cases} +1 & \text{当 } \mathbf{w}^T \mathbf{x} > 0 \\ -1 & \text{当 } \mathbf{w}^T \mathbf{x} \leq 0 \end{cases},$$

- 红色实心点为正例
- 蓝色空心点为负例
- 黑色箭头表示权重向量
- 红色虚线箭头表示权重的更新方向



感知器参数学习的更新过程

- 红色实心点为正例
- 蓝色空心点为负例
- 黑色箭头表示权重向量
- 红色虚线箭头表示权重的更新方向



- 一种错误驱动的在线学习方法

根据感知器的学习策略，可以反推出感知器的损失函数为：

$$\mathcal{L}(\mathbf{w}; \mathbf{x}, y) = \max(0, -y\mathbf{w}^T \mathbf{x}).$$

采用随机梯度下降，其每次更新的梯度为

$$\frac{\partial \mathcal{L}(\mathbf{w}; \mathbf{x}, y)}{\partial \mathbf{w}} = \begin{cases} 0 & \text{if } y\mathbf{w}^T \mathbf{x} > 0, \\ -y\mathbf{x} & \text{if } y\mathbf{w}^T \mathbf{x} < 0. \end{cases}$$

每分错一个样本时更新权重 $\mathbf{w} \leftarrow \mathbf{w} + y\mathbf{x}$

感知器的学习过程

- 一种错误驱动的在线学习方法

输入: 训练集: $(\mathbf{x}_i, y_i), i = 1, \dots, N$, 迭代次数: T

1. 初始化权重向量

1 初始化: $\mathbf{w}_0 = 0$;

2 $k = 0$;

3 for $t = 1 \dots T$ do

4 for $i = 1 \dots N$ do

5 选取一个样本 (\mathbf{x}_i, y_i) , if $\mathbf{w}^T(y_i \mathbf{x}_i) < 0$ then

6 $\mathbf{w}_{k+1} = \mathbf{w}_k + y_i \mathbf{x}_i$;

7 $k = k + 1$;

8 end

9 end

10 end

输出: \mathbf{w}_k

2. 每分错一个样本时更新权重

$$\mathbf{w} \leftarrow \mathbf{w} + y\mathbf{x}$$

表示分错

对比Logistic回归的更新方式:

$$\frac{\partial \mathcal{R}(\mathbf{w})}{\partial \mathbf{w}} = \frac{1}{N} \sum_{i=1}^N \left(\mathbf{x}^{(i)} \cdot \left(\sigma(\mathbf{w}^T \mathbf{x}^{(i)}) - y^{(i)} \right) \right)$$

定义 3.1 – 两类线性可分：对于训练集 $\mathcal{D} = \{(\mathbf{x}^{(n)}, y^{(n)})\}_{n=1}^N$ ，如果存在权重向量 \mathbf{w}^* ，对所有样本都满足 $y f(\mathbf{x}; \mathbf{w}^*) > 0$ ，那么训练集 \mathcal{D} 是线性可分的。

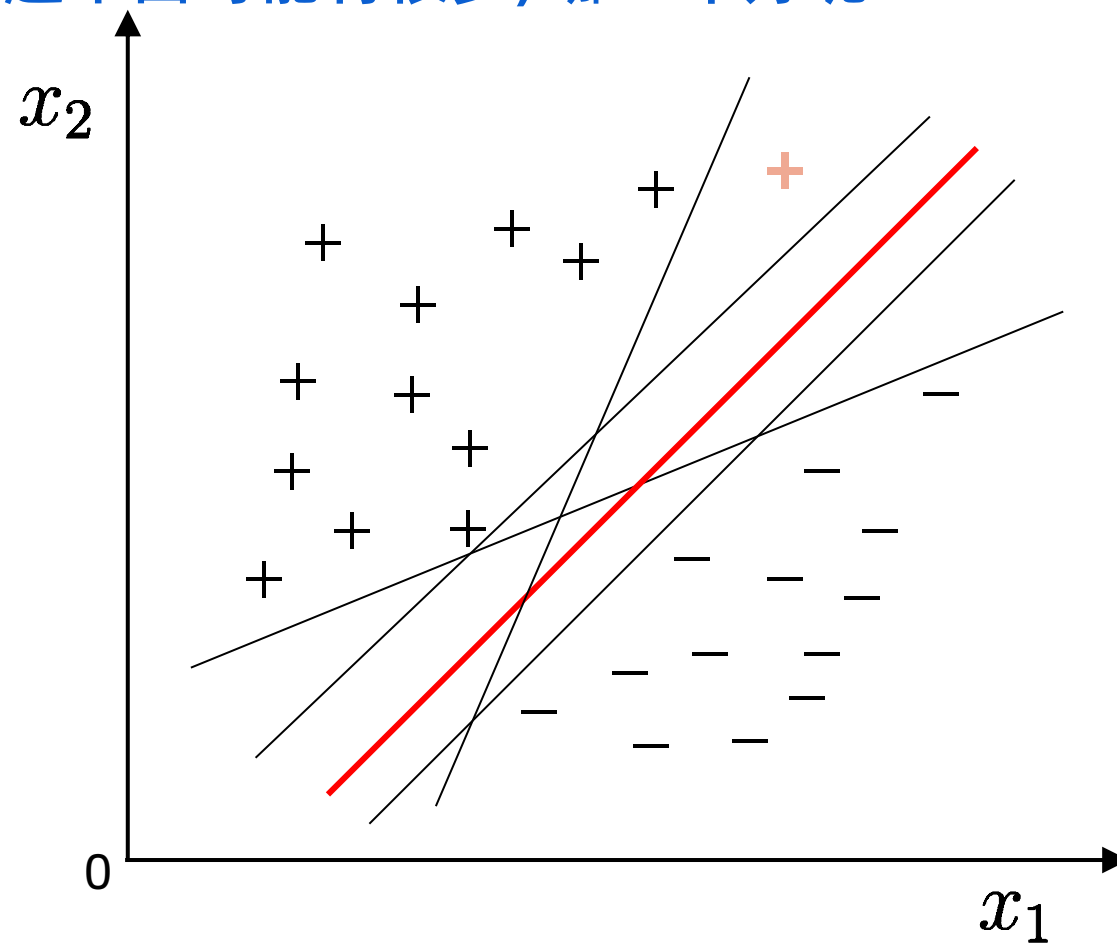
定理 3.1 – 感知器收敛性：给定一个训练集 $\mathcal{D} = \{(\mathbf{x}^{(n)}, y^{(n)})\}_{n=1}^N$ ，假设 R 是训练集中最大的特征向量的模，

$$R = \max_n \|\mathbf{x}^{(n)}\|.$$

如果训练集 \mathcal{D} 线性可分，感知器学习算法3.1的权重更新次数不超过 $\frac{R^2}{\gamma^2}$ 。

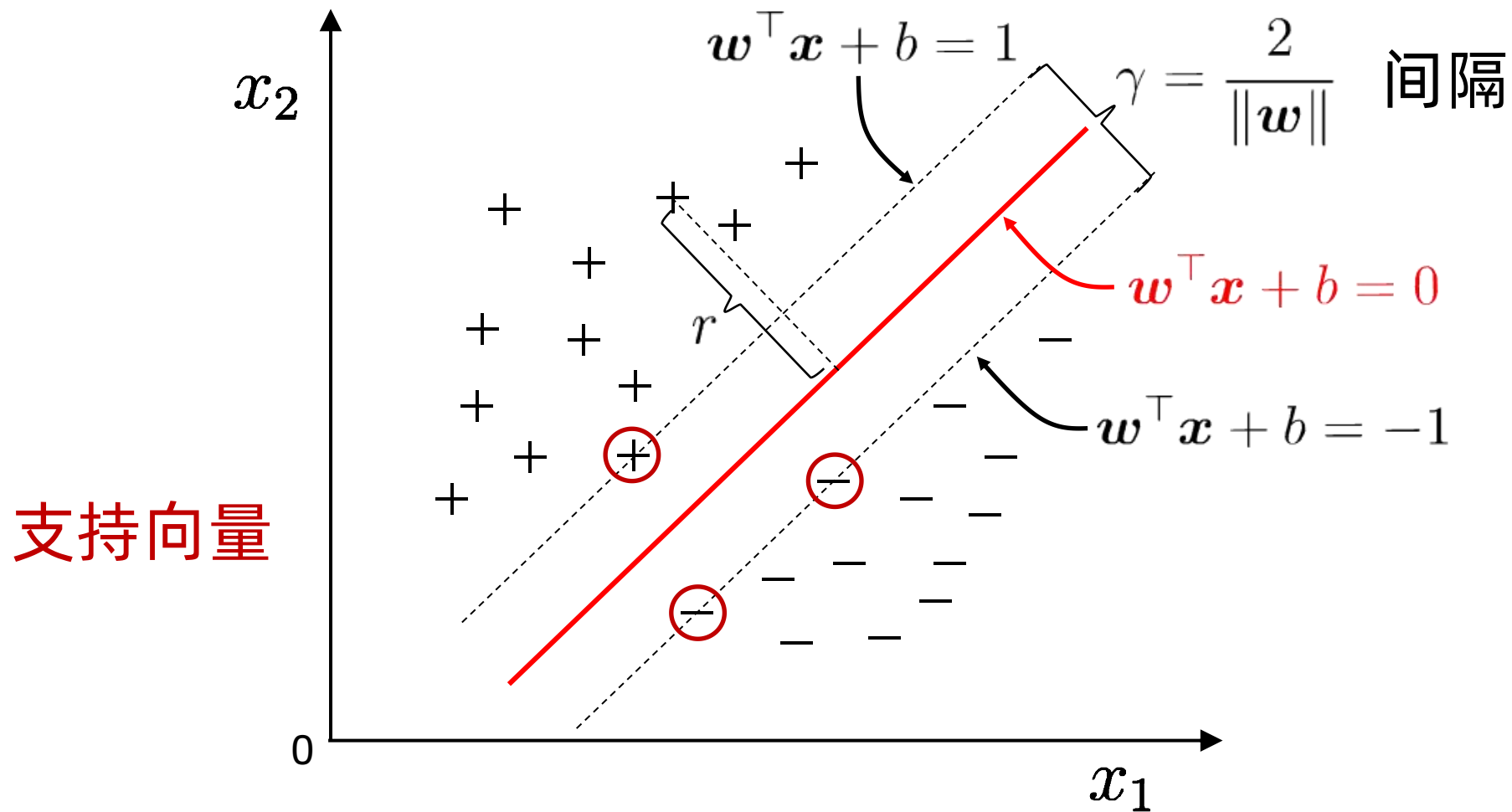
支持向量机 SVM

-Q: 将训练样本分开的超平面可能有很多, 哪一个好呢?



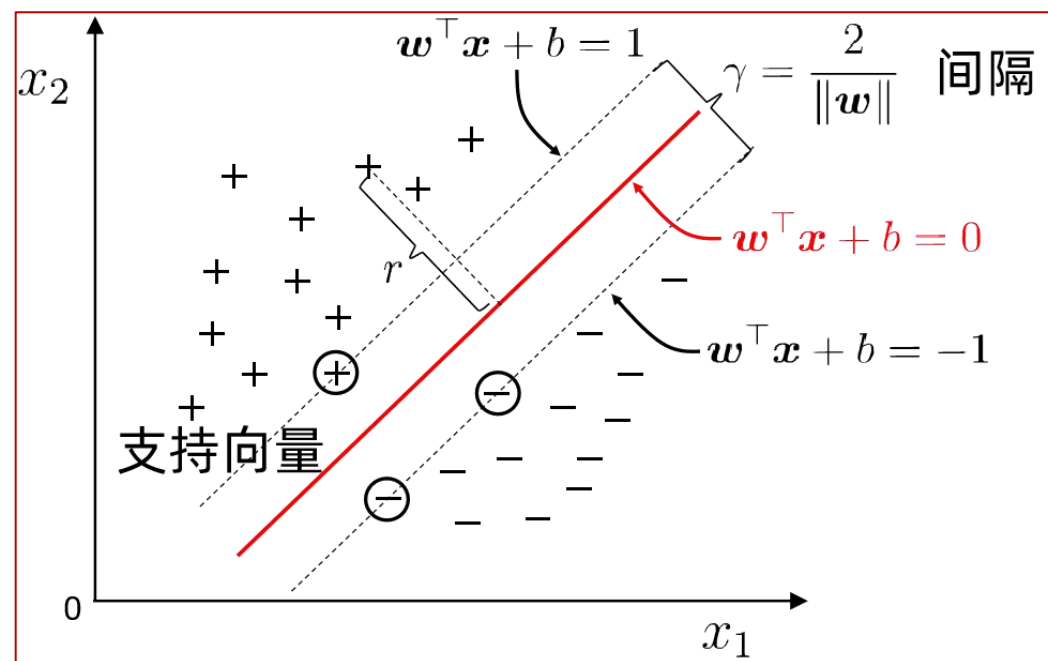
- A: 应选择“正中间”, 容忍性好, 鲁棒性高, 泛化能力最强.

超平面方程: $w^\top x + b = 0$



- 最大间隔: 寻找参数 w 和 b , 使得 γ 最大.

$$\begin{aligned} \arg \max_{w, b} \quad & \frac{2}{\|w\|} \\ \text{s.t.} \quad & y_i(w^\top x_i + b) \geq 1, \quad i = 1, 2, \dots, m. \end{aligned}$$



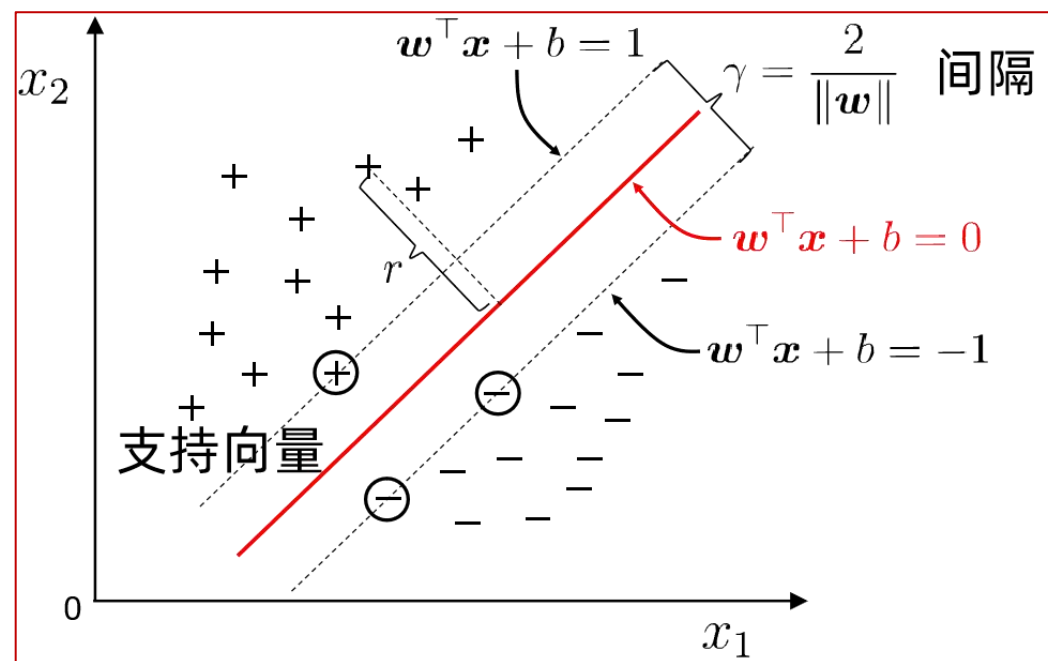
支持向量机基本型

- 最大间隔: 寻找参数 w 和 b , 使得 γ 最大.

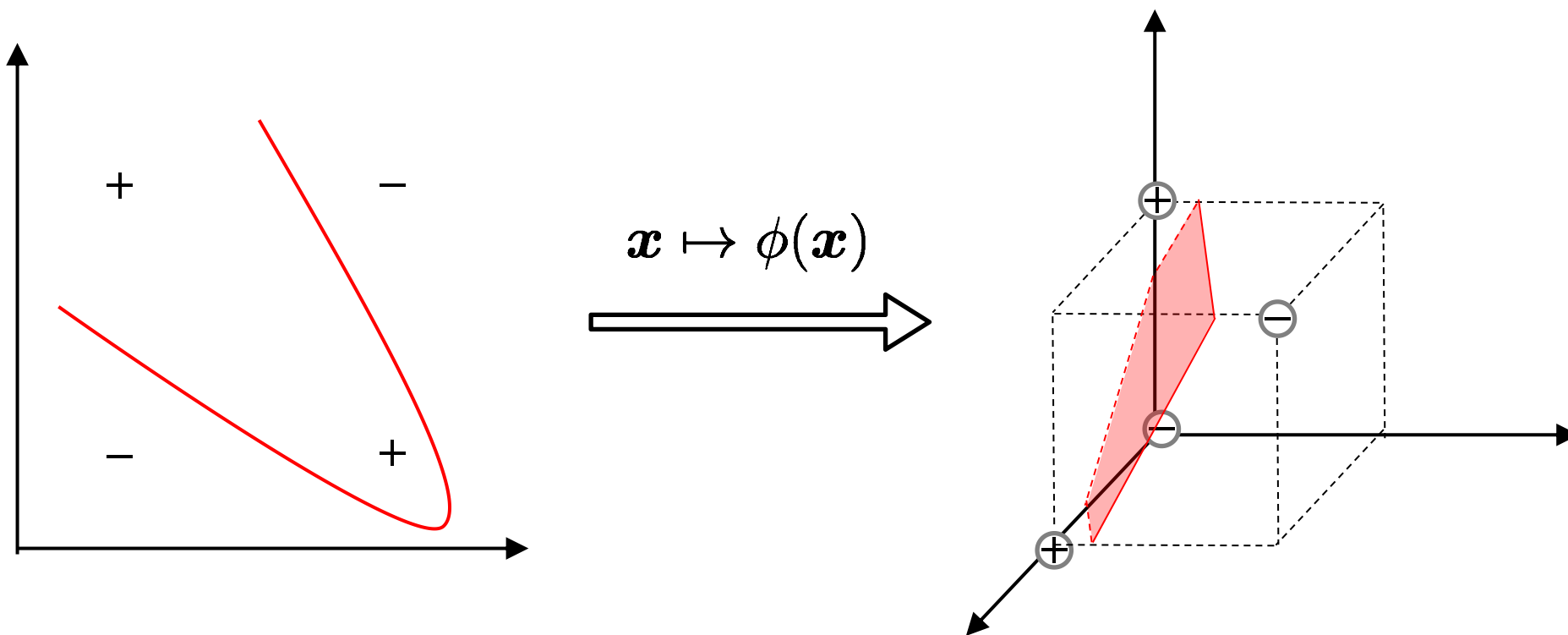
$$\begin{aligned} \arg \max_{w,b} \quad & \frac{2}{\|w\|} \\ \text{s.t.} \quad & y_i(w^\top x_i + b) \geq 1, \quad i = 1, 2, \dots, m. \end{aligned}$$



$$\begin{aligned} \arg \min_{w,b} \quad & \frac{1}{2} \|w\|^2 \\ \text{s.t.} \quad & y_i(w^\top x_i + b) \geq 1, \quad i = 1, 2, \dots, m. \end{aligned}$$



- Q: 若不存在一个能正确划分两类样本的超平面, 怎么办?
- A: 将样本从原始空间映射到一个更高维的特征空间, 使得样本在这个特征空间内线性可分.



- 设样本 \mathbf{x} 映射后的向量为 $\phi(\mathbf{x})$, 划分超平面为 $f(\mathbf{x}) = \mathbf{w}^\top \phi(\mathbf{x}) + b$

原始问题

$$\begin{aligned} \min_{\mathbf{w}, b} \quad & \frac{1}{2} \|\mathbf{w}\|^2 \\ \text{s.t.} \quad & y_i(\mathbf{w}^\top \phi(\mathbf{x}_i) + b) \geq 1, \quad i = 1, 2, \dots, m. \end{aligned}$$

对偶问题

$$\begin{aligned} \min_{\alpha} \quad & \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^m \alpha_i \alpha_j y_i y_j \boxed{\phi(\mathbf{x}_i)^\top \phi(\mathbf{x}_j)} - \sum_{i=1}^m \alpha_i \\ \text{s.t.} \quad & \sum_{i=1}^m \alpha_i y_i = 0, \quad \alpha_i \geq 0, \quad i = 1, 2, \dots, m. \end{aligned}$$

只以内积的形式出现

预测

$$f(\mathbf{x}) = \mathbf{w}^\top \phi(\mathbf{x}) + b = \sum_{i=1}^m \alpha_i y_i \boxed{\phi(\mathbf{x}_i)^\top \phi(\mathbf{x})} + b$$

- 基本想法：不显式地设计核映射, 而是设计核函数.

$$\kappa(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^\top \phi(\mathbf{x}_j)$$

- 基本想法：不显式地设计核映射, 而是设计核函数.

$$\kappa(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^\top \phi(\mathbf{x}_j)$$

- Mercer定理(充分非必要): 只要一个对称函数所对应的核矩阵半正定, 则它就能作为核函数来使用.

核函数

- 基本想法：不显式地设计核映射, 而是设计核函数.

$$\kappa(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^\top \phi(\mathbf{x}_j)$$

- Mercer定理(充分非必要): 只要一个对称函数所对应的核矩阵半正定, 则它就能作为核函数来使用.
- 常用核函数:

名称	表达式	参数
线性核	$\kappa(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^\top \mathbf{x}_j$	
多项式核	$\kappa(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i^\top \mathbf{x}_j)^d$	$d \geq 1$ 为多项式的次数
高斯核	$\kappa(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\ \mathbf{x}_i - \mathbf{x}_j\ ^2}{2\delta^2}\right)$	$\delta > 0$ 为高斯核的带宽(width)
拉普拉斯核	$\kappa(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\ \mathbf{x}_i - \mathbf{x}_j\ }{\delta}\right)$	$\delta > 0$
Sigmoid核	$\kappa(\mathbf{x}_i, \mathbf{x}_j) = \tanh(\beta \mathbf{x}_i^\top \mathbf{x}_j + \theta)$	\tanh 为双曲正切函数, $\beta > 0, \theta < 0$

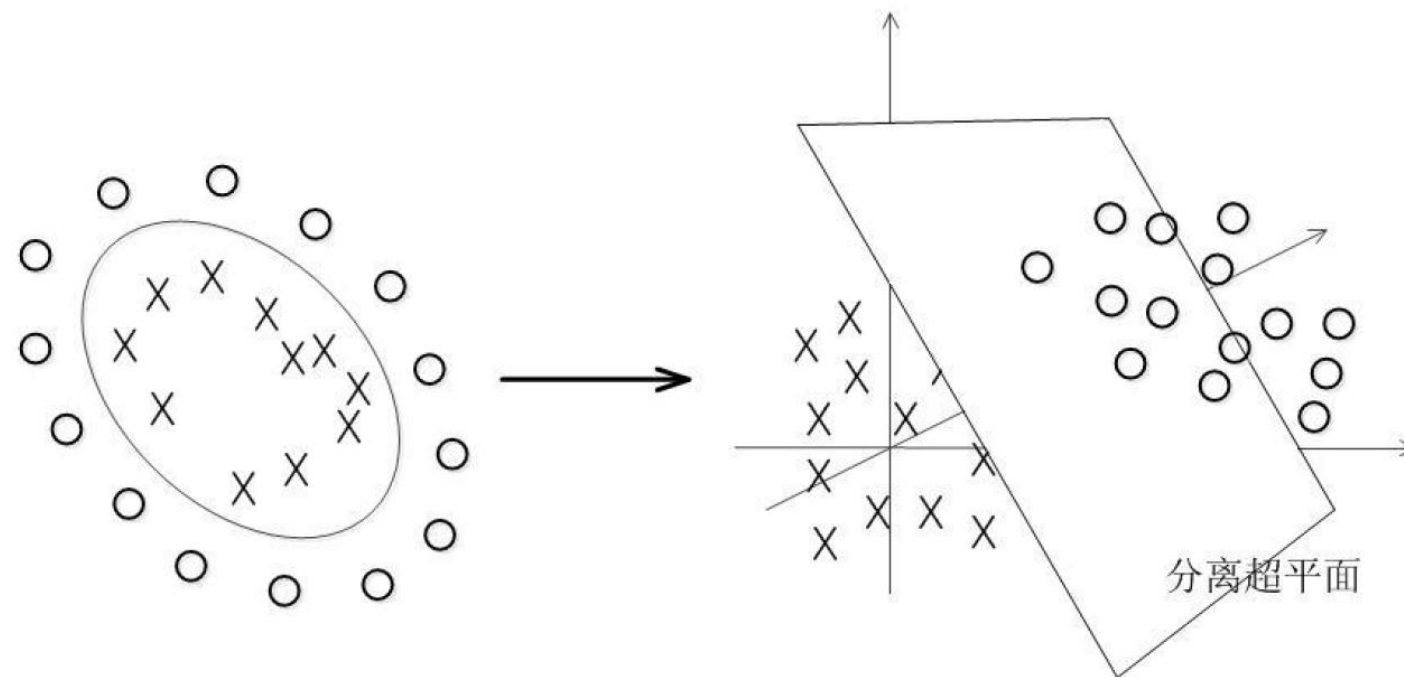
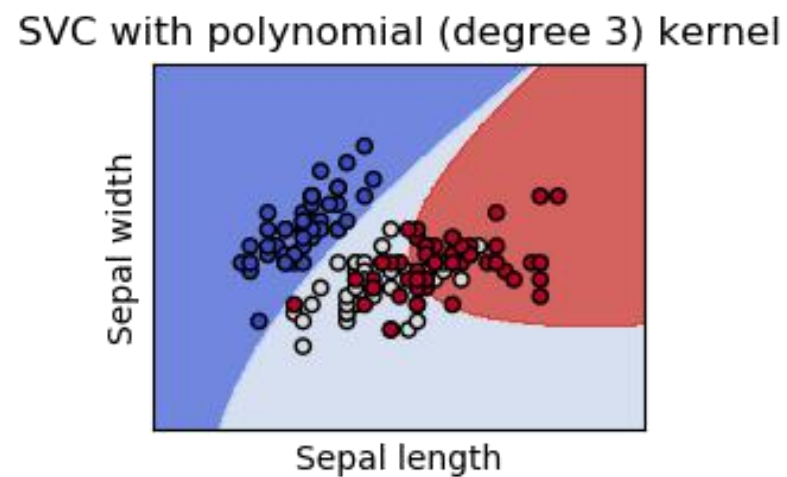
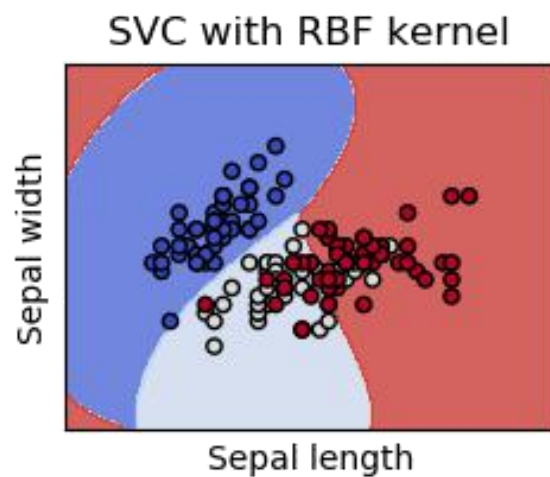
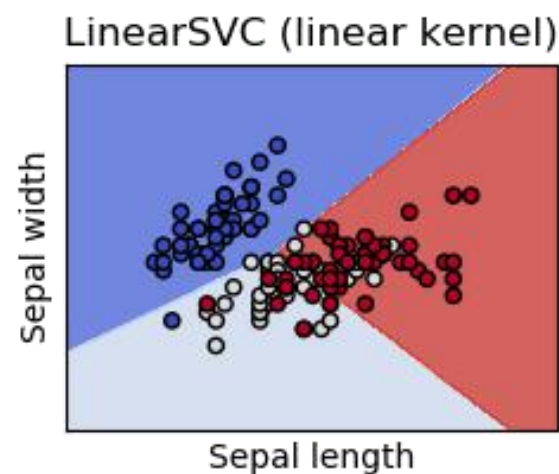
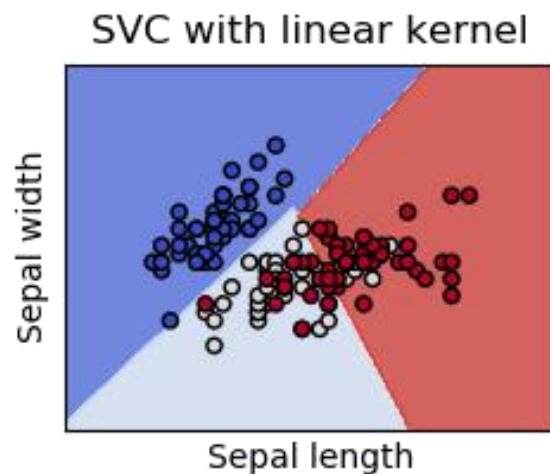


图 7: 核函数映射示意

《支持向量机通俗导论：理解SVM 的三层境界》



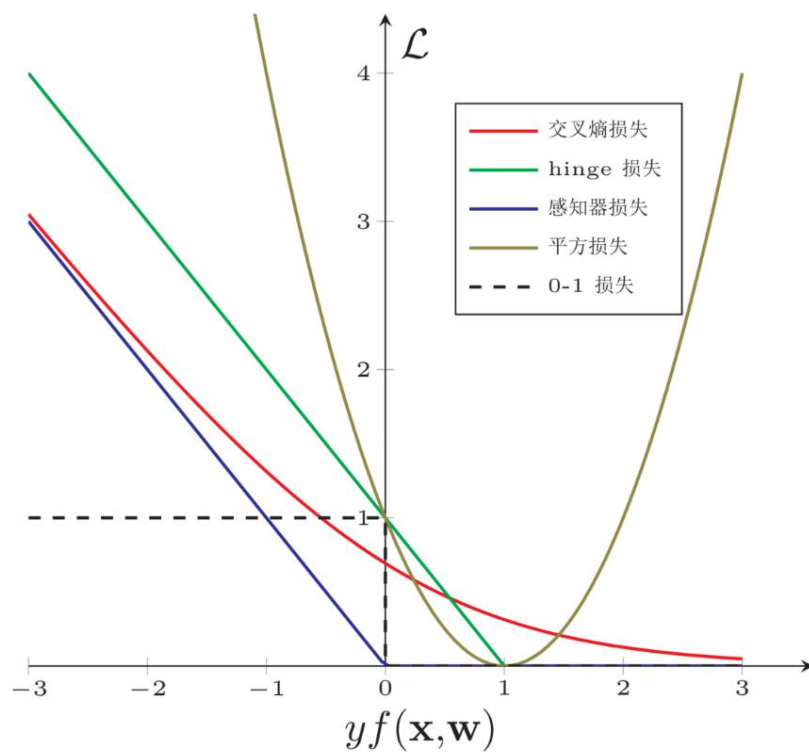
<http://scikit-learn.org/stable/modules/svm.html>

小结

	激活函数	损失函数	优化方法
线性回归	-	$(y - \mathbf{w}^T \mathbf{x})^2$	最小二乘、梯度下降
Logistic 回归	$\sigma(\mathbf{w}^T \mathbf{x})$	$y \log \sigma(\mathbf{w}^T \mathbf{x})$	梯度下降
Softmax 回归	$\text{softmax}(W^T \mathbf{x})$	$y \log \text{softmax}(W^T \mathbf{x})$	梯度下降
感知器	$\text{sgn}(\mathbf{w}^T \mathbf{x})$	$\max(0, -y\mathbf{w}^T \mathbf{x})$	随机梯度下降
支持向量机	$\text{sgn}(\mathbf{w}^T \mathbf{x})$	$\max(0, 1 - y\mathbf{w}^T \mathbf{x})$	二次规划、SMO 等

表 3.1 几种不同的线性模型对比

为了比较这些损失函数，我们统一定义类别标签 $y \in \{+1, -1\}$ ，并定义 $f(\mathbf{x}; \mathbf{w}) = \mathbf{w}^T \mathbf{x} + b$ 。这样对于样本 (\mathbf{x}, y) ，若 $yf(\mathbf{x}; \mathbf{w}) > 0$ ，则分类正确，相反则分类错误。这样为了方便比较这些模型，我们可以将它们的损失函数都表述为定义在 $yf(\mathbf{x}; \mathbf{w})$ 上的函数。



$$\mathcal{L}_{LR} = \log(1 + \exp(-yf(\mathbf{x}; \mathbf{w})))$$

$$\mathcal{L}_{hinge} = \max(0, 1 - yf(\mathbf{x}; \mathbf{w}))$$

$$\mathcal{L}_p = \max(0, -yf(\mathbf{x}; \mathbf{w}))$$

$$\mathcal{L}_{squared} = (1 - yf(\mathbf{x}; \mathbf{w}))^2$$

- 编程练习: [chap3_softmax_regression](#)

	激活函数	损失函数	优化方法
线性回归	-	$(y - \mathbf{w}^T \mathbf{x})^2$	最小二乘、梯度下降
Logistic 回归	$\sigma(\mathbf{w}^T \mathbf{x})$	$y \log \sigma(\mathbf{w}^T \mathbf{x})$	梯度下降
Softmax 回归	$\text{softmax}(W^T \mathbf{x})$	$y \log \text{softmax}(W^T \mathbf{x})$	梯度下降
感知器	$\text{sgn}(\mathbf{w}^T \mathbf{x})$	$\max(0, -y \mathbf{w}^T \mathbf{x})$	随机梯度下降
支持向量机	$\text{sgn}(\mathbf{w}^T \mathbf{x})$	$\max(0, 1 - y \mathbf{w}^T \mathbf{x})$	二次规划、SMO 等

表 3.1 几种不同的线性模型对比



深度学习



杭州电子科技大学
HANGZHOU DIANZI UNIVERSITY

新禾屯育 非学历教育

附录：概率基本概念

关于概率的一些基本概念

- **概率 (Probability)**

- 一个随机事件发生的可能性大小，为0到1之间的实数。

- **随机变量 (Random Variable)**

- 比如随机掷一个骰子，得到的点数就可以看成一个随机变量X，其取值为{1,2,3,4,5,6}。

- **概率分布 (Probability Distribution)**

- 一个随机变量X取每种可能值的概率

- 并满足

$$P(X = x_i) = p(x_i), \quad \forall i \in \{1, \dots, n\}.$$

$$\sum_{i=1}^n p(x_i) = 1,$$

$$p(x_i) \geq 0, \quad \forall i \in \{1, \dots, n\}.$$

- 伯努利分布 (Bernoulli Distribution)

- 在一次试验中, 事件A出现的概率为 μ , 不出现的概率为 $1 - \mu$ 。若用变量X 表示事件A出现的次数, 则X 的取值为0和1, 其相应的分布为

$$p(x) = \mu^x (1 - \mu)^{(1-x)}$$

- 二项分布 (Binomial Distribution)

- 在n次伯努利分布中, 若以变量X 表示事件A出现的次数, 则X 的取值为 $\{0, \dots, n\}$, 其相应的分布

$$P(X = k) = \binom{n}{k} \mu^k (1 - \mu)^{n-k}, \quad k = 1 \cdots, n$$

二项式系数, 表示从 n 个元素中取出 k 个元素而不考虑其顺序的組合的总数。

- 条件概率 (Conditional Probability)

- 对于离散随机向量 (X, Y) , 已知 $X = x$ 的条件下, 随机变量 $Y = y$ 的条件概率为:

$$p(y|x) = P(Y = y|X = x) = \frac{p(x, y)}{p(x)}$$

- 贝叶斯公式

- 两个条件概率 $p(y|x)$ 和 $p(x|y)$ 之间的关系

$$p(y|x) = \frac{p(x|y)p(y)}{p(x)}$$