《神经网络与深度学习》

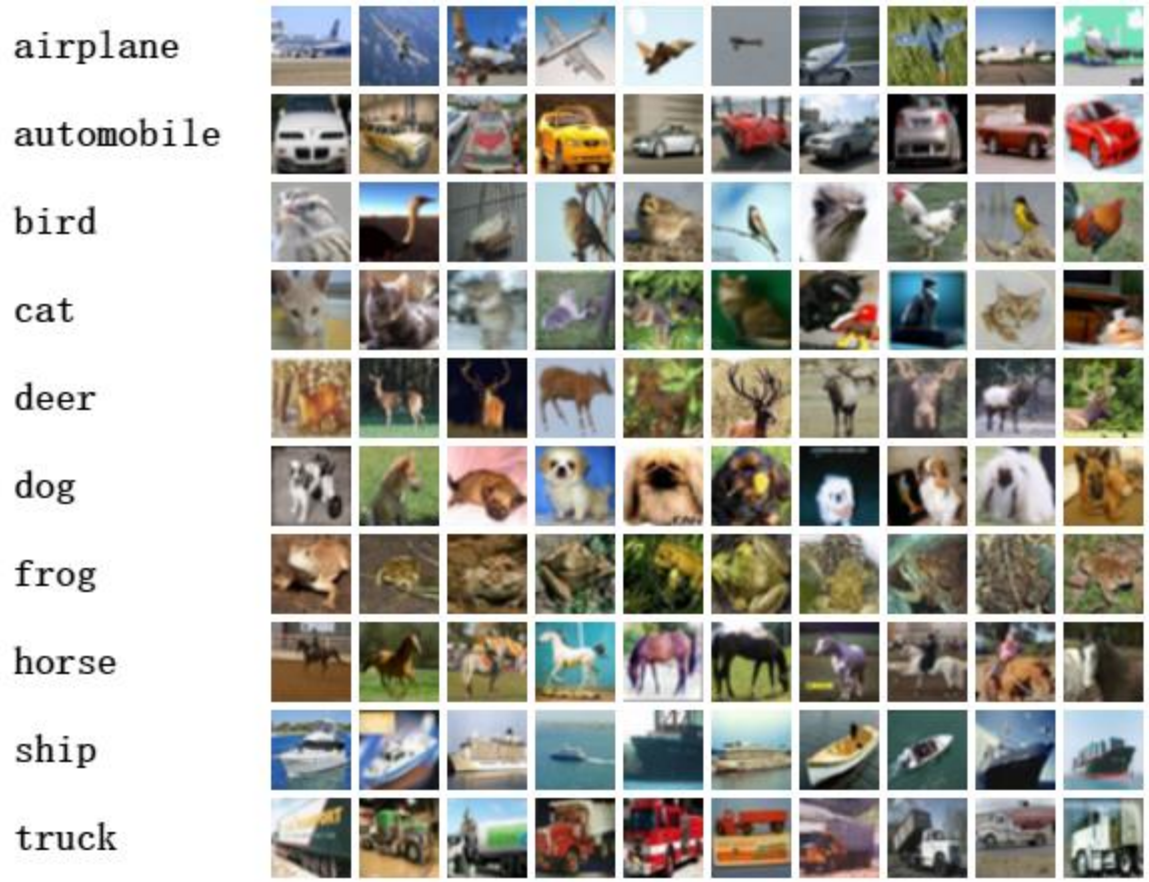
线性模型

大纲

- 线性分类模型
  - Logistic Regression
  - Softmax Regression
  - Perceptron
  - SVM

示例：图像分类

- 数据集：CIFAR-10
  - 60000张32x32色彩图像，共10类
  - 每类6000张图像



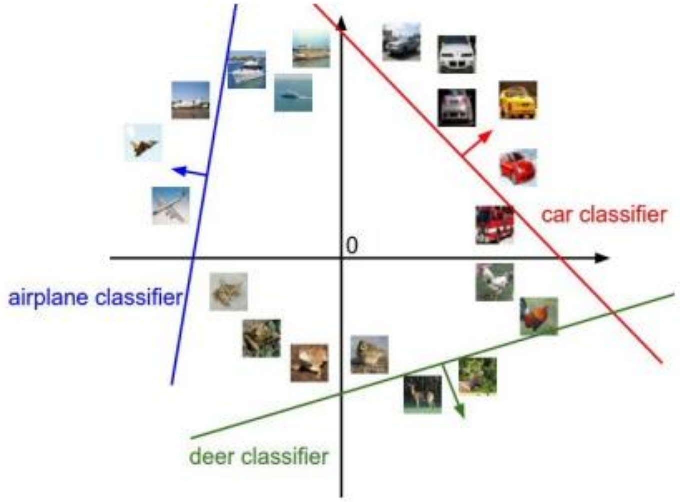
示例：图像分类

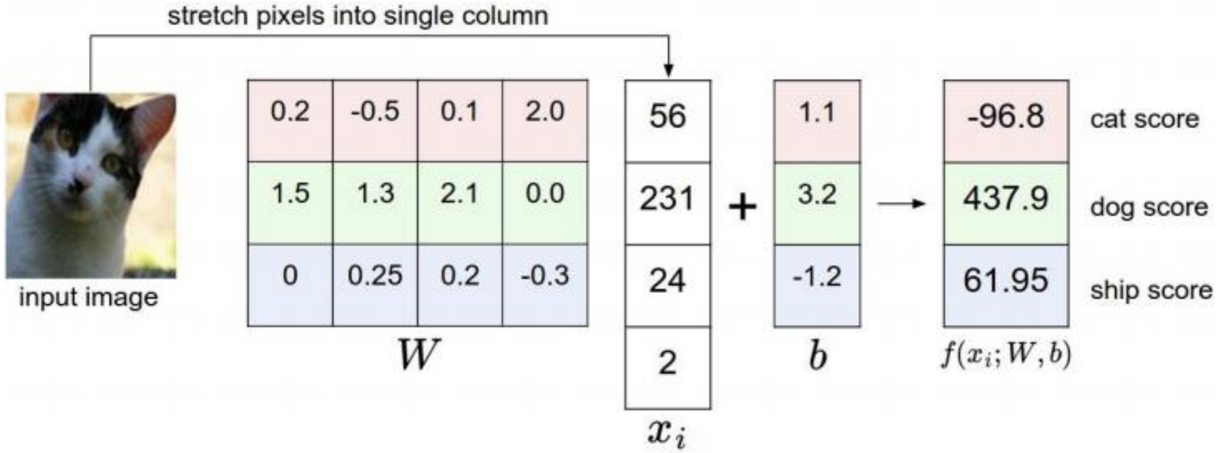
[32x32x3]
array of numbers 0...1
(3072 numbers total)

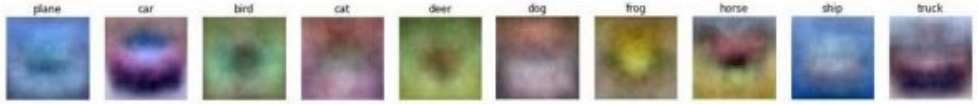
image parameters

$$f(\mathbf{x},\mathbf{W})$$

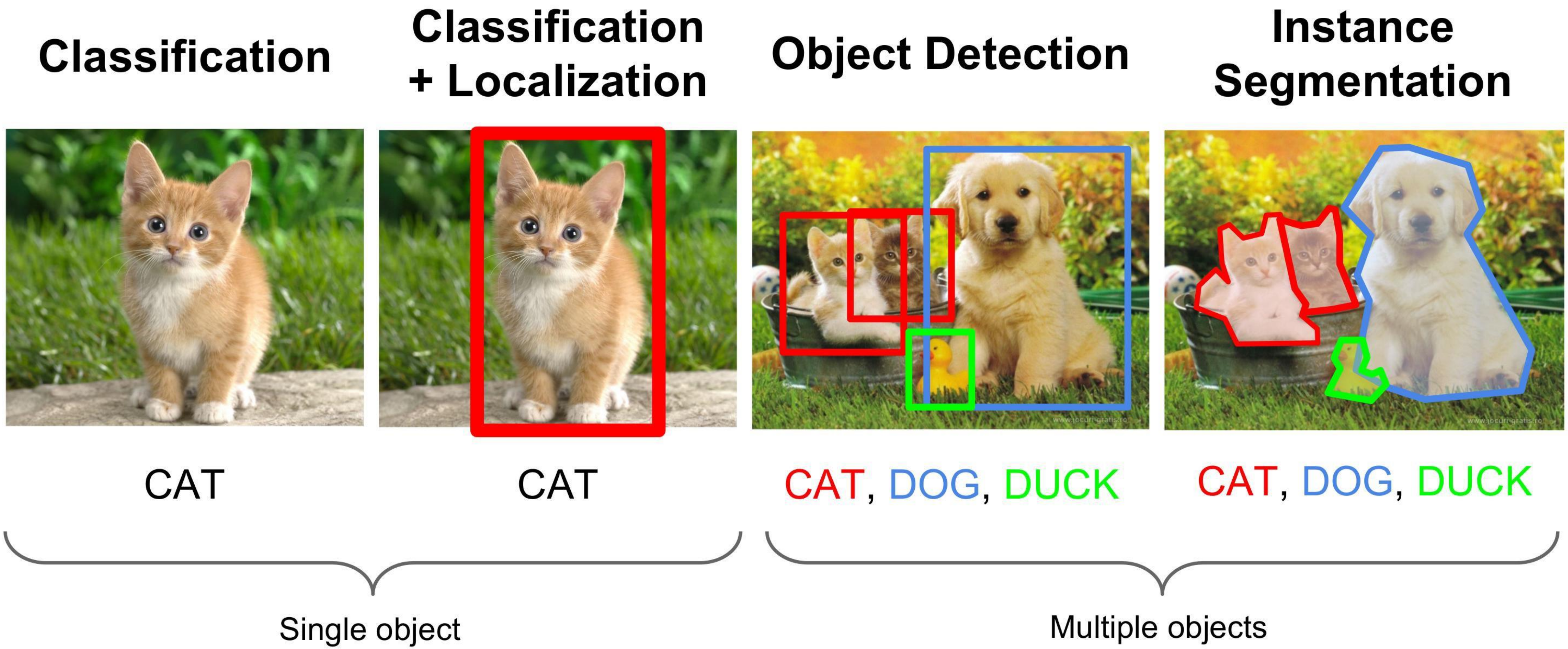
10 numbers, indicating class scores





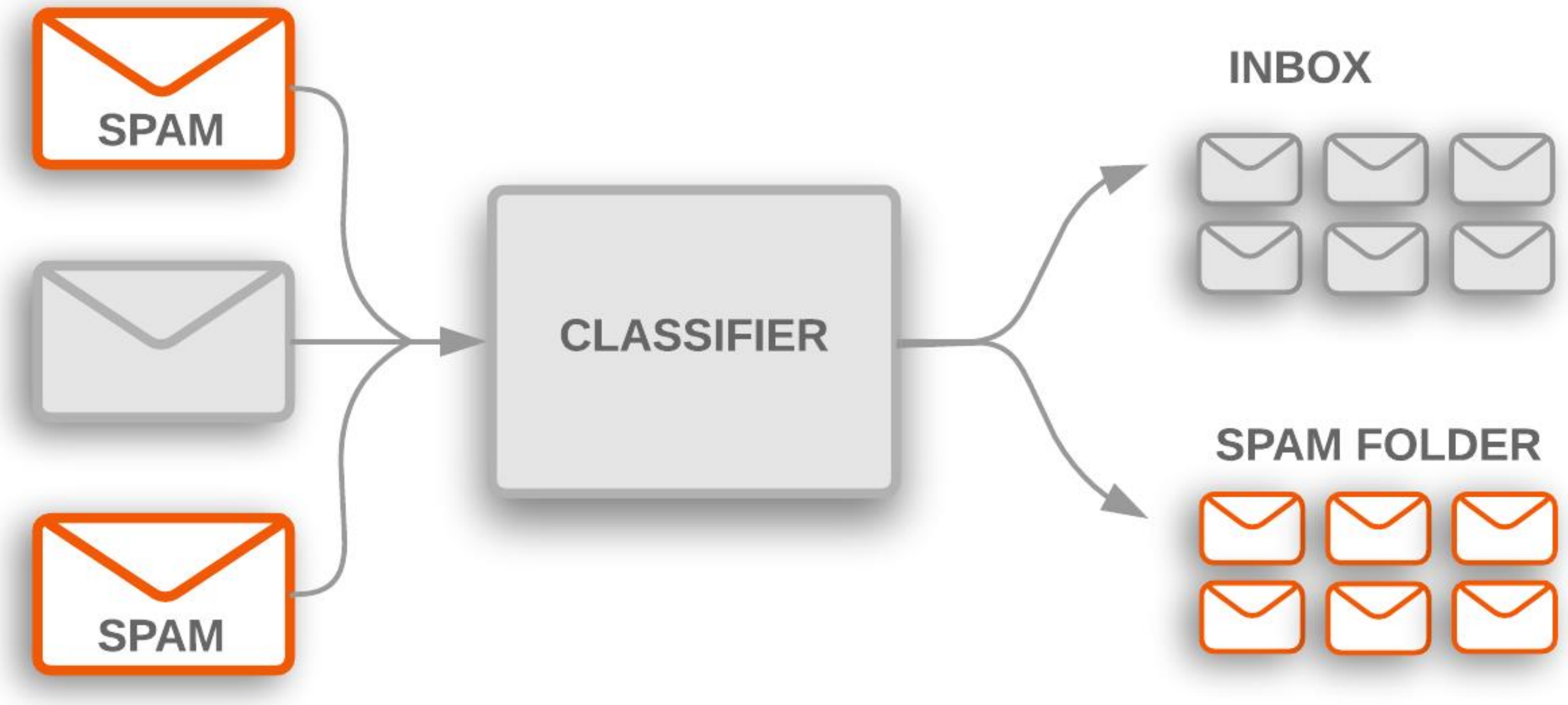


示例：图像分类、目标检测、实例分割



https://medium.com/zylapp/review-of-deep-learning-algorithms-for-object-detection-c1f3d437b852

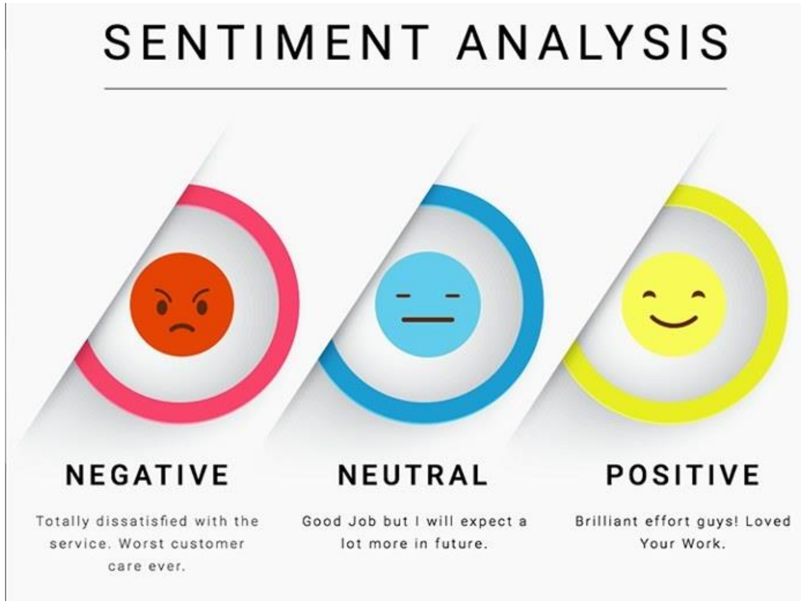
示例：垃圾邮件过滤



示例：文档归类

https://towardsdatascience.com/automated-text-classification-using-machine-learning-3df4f4f9570b

示例：情感分类



Review (X)	Rating (Y)
"This movie is fantastic! I really like it because it is so good!"	★★★★☆
"Not to my taste, will skip and watch another movie"	★★☆☆☆
"This movie really sucks! Can I get my money back please?"	★☆☆☆☆

示例：文本分类

- 将样本x从文本形式转为向量形式
  - 词袋模型（Bag-of-Words，BoW）模型

比如两个文本"我 喜欢 读书"和"我 讨厌 读书"中共有"我"、"喜欢"、"讨厌"、"读书"四个词，它们的BoW表示分别为

$$\mathbf{v}_1 = [1\ 1\ 0\ 1\ ]^{\mathrm{T}},$$
$$\mathbf{v}_2 = [1\ 0\ 1\ 1\ ]^{\mathrm{T}}.$$

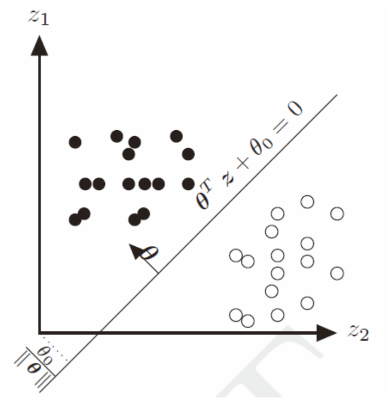
示例：文本情感分类

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



+

-

线性分类模型

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.
  - 线性判别函数：

$$
\begin{aligned}
f(\boldsymbol{x};\boldsymbol{w}) &= w_1x_1 + w_2x_2 + \cdots + w_Dx_D + b \\
&= \boldsymbol{w}^\top\boldsymbol{x} + b,
\end{aligned}
$$

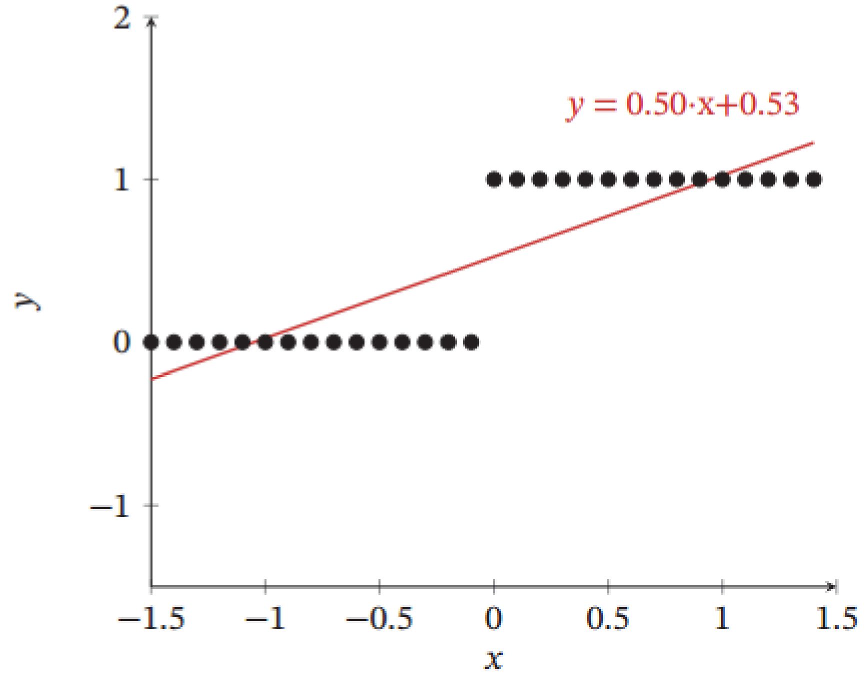
  - 线性的决策函数：

$$
y = f(\boldsymbol{x};\boldsymbol{w})
$$

线性决策函数

- 线性的决策函数预测分类问题的输出效果差，因为输出目标y是一些离散的标签

$$y = f(\boldsymbol{x}; \boldsymbol{w})$$



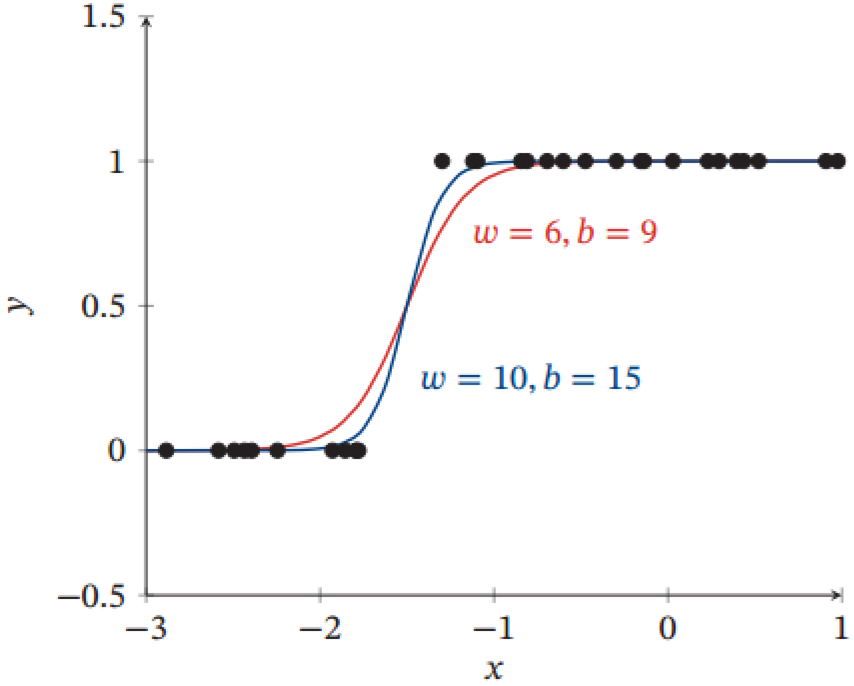
(a) 线性回归

非线性决策函数

- 引入一个非线性的决策函数来更好的进行拟合

$$y = g(f(\boldsymbol{x};\boldsymbol{w}))$$

- 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。



线性模型

- Logistic回归
- Softmax回归
- 感知器
- 支持向量机

二分类与Logistic 回归

二分类

- 二分类问题的类别标签只有两种取值
- 通常可以设为 $\{+1, -1\}$ 或 $\{0, 1\}$.
- 决策函数

$$g(f(\boldsymbol{x};\boldsymbol{w})) = \begin{cases} 1 & \text{if} \quad f(\boldsymbol{x};\boldsymbol{w}) > 0 \\ 0 & \text{if} \quad f(\boldsymbol{x};\boldsymbol{w}) < 0 \end{cases}$$

- 或

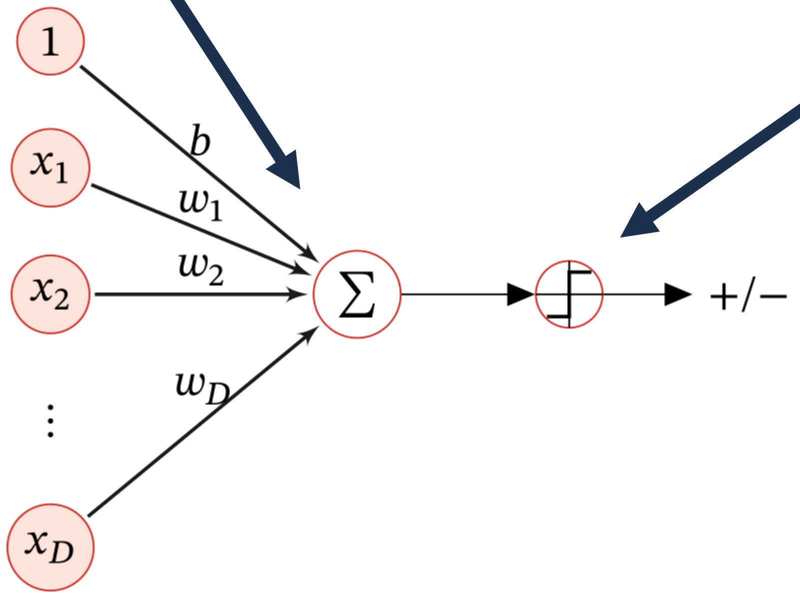
$$\begin{aligned} g(f(\boldsymbol{x};\boldsymbol{w})) &= \operatorname{sgn}\left(f(\boldsymbol{x};\boldsymbol{w})\right) \\ &\triangleq \begin{cases} +1 & \text{if} \quad f(\boldsymbol{x};\boldsymbol{w}) > 0, \\ -1 & \text{if} \quad f(\boldsymbol{x};\boldsymbol{w}) < 0. \end{cases} \end{aligned}$$

二分类模型的基本框架

- 线性判别函数 + 二分类决策函数

$$f(\boldsymbol{x};\boldsymbol{w}) = \boldsymbol{w}^{\top}\boldsymbol{x} + b$$

$$g(f(\boldsymbol{x};\boldsymbol{w})) = \operatorname{sgn}(f(\boldsymbol{x};\boldsymbol{w}))$$



二分类模型的基本框架

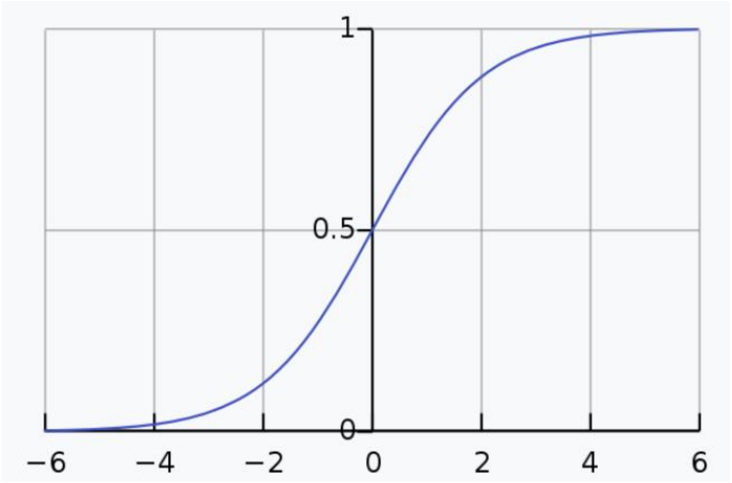
- 逻辑回归的总体思路就是，先用逻辑函数把线性回归的结果(-∞,∞)映射到(0,1)，再通过决策边界建立与分类的概率联系。

- 存在问题
  - 函数g非连续，无法求导 $g(f(\boldsymbol{x};\boldsymbol{w})) = \operatorname{sgn}\left(f(\boldsymbol{x};\boldsymbol{w})\right)$
  - 学习速率较慢

Logistic函数与回归

- Logistic函数(Sigmoid函数)

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



- 优点
  - 将任意实数映射到 [0,1] 范围内
  - 任意阶可导
  - 在0 附近几乎是线性的，但在两端趋于平缓
  - 将异常值压向 0 或 1

其他常用的激活函数

Hyperbolic tangent

$$f(x) = \tanh x = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

Arctangent function

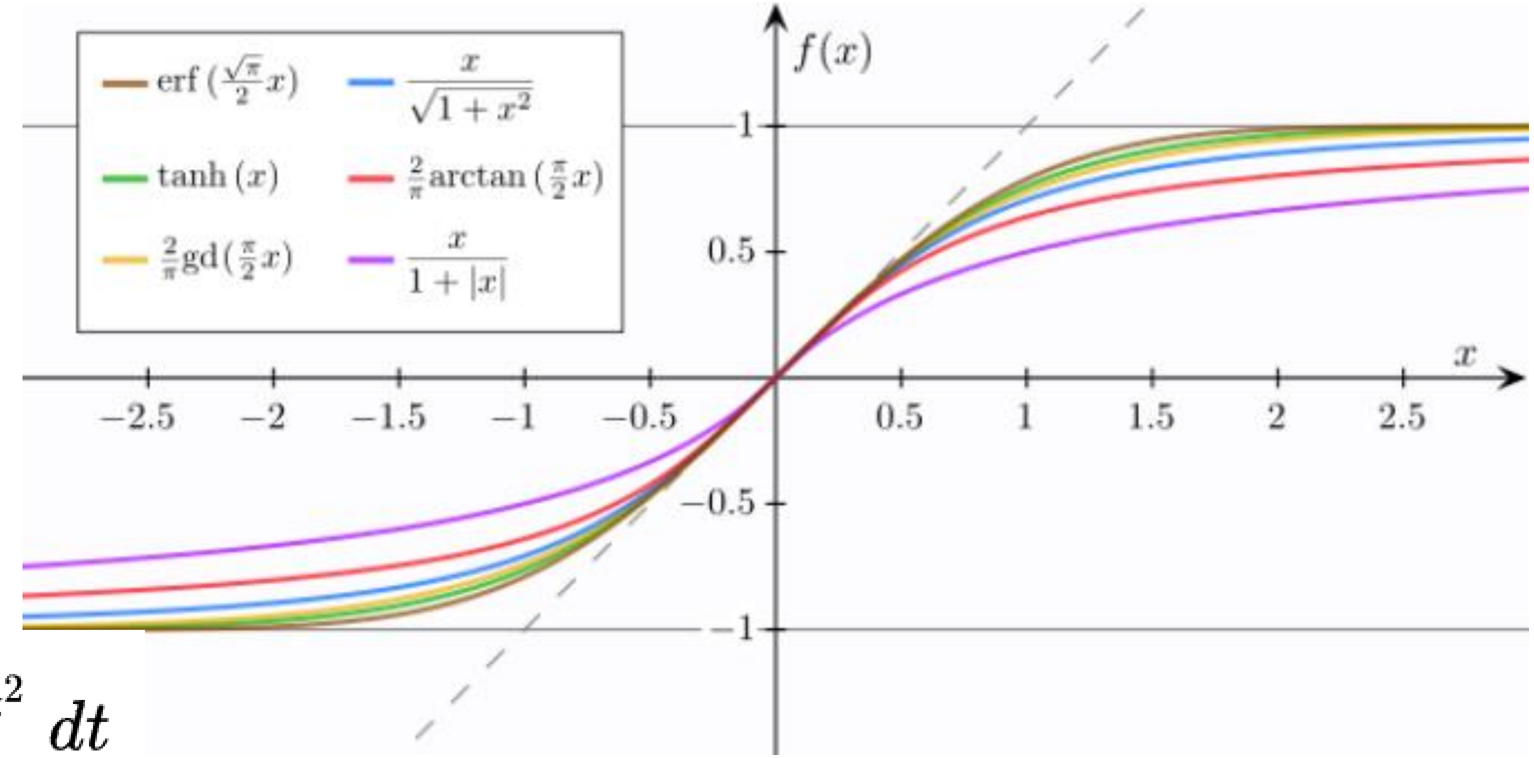
$$f(x) = \arctan x$$

Gudermannian function

$$f(x) = \mathrm{gd}(x) = \int_0^x \frac{dt}{\cosh t}$$

Error function

$$f(x) = \mathrm{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2}\, dt$$



Logistic函数与回归

- 对于一个二元逻辑回归，我们只需要确保

$$P(y = 1) + P(y = 0) = 1$$

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

$$P(y=0) = 1 - \sigma(z) = 1 - \frac{1}{1+\exp(-z)} = \frac{\exp(-z)}{1+\exp(-z)}$$

- 根据 Sigmoid 函数的性质

$$1 - \sigma(x) = \sigma(-x)$$

推导过程

- 将分类问题看作条件概率估计问题

  - 基于已有数据以及学习权重类别标签的条件概率 $p(y=c|x)$ 最大化。
  - 以二分类为例，

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

  - 标签 y = 1的后验概率为

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

推导过程

- 标签 y = 0的后验概率为

$$p(y=0|\boldsymbol{x}) = 1 - p(y=1|\boldsymbol{x}) \\ = \frac{\exp(-\boldsymbol{w}^{\mathsf{T}}\boldsymbol{x})}{1+\exp(-\boldsymbol{w}^{\mathsf{T}}\boldsymbol{x})}.$$

- 在二分类模型中，事件发生与不发生的概率之比p/(1-p) 称为事件的几率. 我们有

$$\boldsymbol{w}^{\mathsf{T}}\boldsymbol{x} = \log \frac{p(y=1|\boldsymbol{x})}{1-p(y=1|\boldsymbol{x})}$$

风险函数

- 我们使用极大似然估计法来求解. 我们把单个样本看做一个事件，那么这个事件发生的概率就是：

$$P(y|\boldsymbol{x}) = \begin{cases} p, y = 1 \\ 1 - p, y = 0 \end{cases}$$

- 它等价于:

$$P(y_i|\boldsymbol{x}_i) = p^{y_i}(1-p)^{1-y_i}$$

- 则组样本的概率：

$$\begin{aligned} P &= P(y_1|\boldsymbol{x}_1)P(y_2|\boldsymbol{x}_2)P(y_3|\boldsymbol{x}_3)\dots.P(y_m|\boldsymbol{x}_m) \\ &= \prod_{i=1}^{m} p^{y_i}(1-p)^{1-y_i} \end{aligned}$$

风险函数

- 则组样本的概率：

$$
\begin{aligned}
P &= P(y^1|\boldsymbol{x}^1)P(y^2|\boldsymbol{x}^2)P(y^3|\boldsymbol{x}^3)....P(y^m|\boldsymbol{x}^m) \\
&= \prod_{i=1}^{m} p^{y^{(i)}}(1-p)^{1-y^{(i)}}
\end{aligned}
$$

- 则似然函数为

$$
L(w)=\prod_{i=1}^{m} P\left(y^{(i)} \mid \boldsymbol{w}^{\top} x^{(i)}\right)=\prod_{i=1}^{m}\left(\sigma\left(\boldsymbol{w}^{\top} x^{(i)}\right)\right)^{y^{(i)}}\left(1-\sigma\left(\boldsymbol{w}^{\top} x^{(i)}\right)\right)^{1-y^{(i)}}
$$

梯度下降

- 等式两边同时取对数：

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

- 交叉熵损失函数，模型在训练集的风险函数为

$$l(w) = \log L(w) = \sum_{i=1}^{m}\left(y^{(i)}\log\left(\sigma\left(\boldsymbol{w}^{\mathsf{T}}x^{(i)}\right)\right) + \left(1-y^{(i)}\right)\log\left(1-\sigma\left(\boldsymbol{w}^{\mathsf{T}}x^{(i)}\right)\right)\right)$$

$$= -\frac{1}{N}\sum_{n=1}^{N}\left(y^{(n)}\log\hat{y}^{(n)} + (1-y^{(n)})\log(1-\hat{y}^{(n)})\right) \qquad \hat{y}^{(n)} = \sigma(\boldsymbol{w}^{\mathsf{T}}\boldsymbol{x}^{(n)})$$

推导过程

- 梯度为：

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

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

$$
\begin{aligned}
\sigma'(z) = \left(\frac{1}{1+e^{-z}}\right)' &= \frac{e^{-z}}{(1+e^{-z})^2} \\
&= \frac{1+e^{-z}-1}{(1+e^{-z})^2} \\
&= \frac{1}{(1+e^{-z})}\left(1-\frac{1}{(1+e^{-z})}\right) \\
&= \sigma(z)(1-\sigma(z))
\end{aligned}
$$

推导过程

- 采用梯度下降法，Logistic 回归的训练过程为:
  - 初始化 $w_0 \leftarrow 0$，然后通过下式来迭代更新参数:

$$\boldsymbol{w}_{t+1} \leftarrow \boldsymbol{w}_t + \alpha \frac{1}{N} \sum_{n=1}^{N} \boldsymbol{x}^{(n)} \left(y^{(n)} - \hat{y}_{\boldsymbol{w}_t}^{(n)} \right)$$

多分类与 Softmax 回归

多分类（Multi-class Classification）

- 分类的类别数C大于2，一般需要多个线性判别函数
  - “argmax” 方式：共需要 C个判别函数

$$f_c(\boldsymbol{x};\boldsymbol{w}_c) = \boldsymbol{w}_c^{\mathrm{T}}\boldsymbol{x} + b_c, \qquad c \in \{1, \cdots, C\}$$

  - “argmax”方式的预测函数定义为

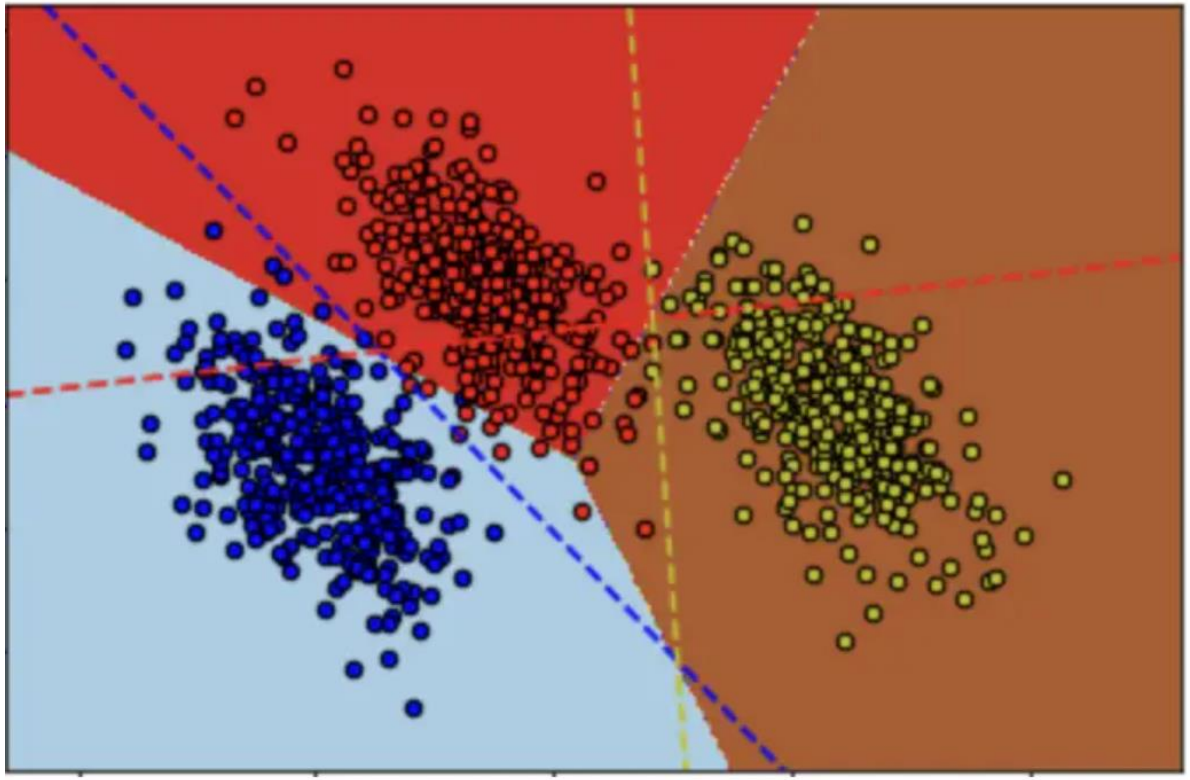
$$y = \arg\max_{c=1}^{C} f_c(\boldsymbol{x};\boldsymbol{w}_c)$$

多分类（Multi-class Classification）

- 相邻两类i和j的决策边界实际上是由

$$f_i(x; w_i) - f_j(x; w_j) = 0$$

- 决定，其法向量为$w_i - w_j$.



Softmax回归

- Softmax函数

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

- 给定一个样本x，Softmax回归预测的属于类别c的条件概率为

$$\begin{aligned} p(y = c|\boldsymbol{x}) &= \mathrm{softmax}(\boldsymbol{w}_c^{\mathrm{T}}\boldsymbol{x}) \\ &= \frac{\exp(\boldsymbol{w}_c^{\mathrm{T}}\boldsymbol{x})}{\sum_{c'=1}^{C} \exp(\boldsymbol{w}_{c'}^{\mathrm{T}}\boldsymbol{x})}, \end{aligned}$$

Softmax回归

- 多分类

$$y = \arg\max_{c=1}^{C} f_c(\boldsymbol{x}; \boldsymbol{w}_c)$$

- Softmax回归的决策函数可以表示为

$$\begin{aligned}\hat{y} &= \arg\max_{c=1}^{C} p(y = c|\boldsymbol{x}) \\ &= \arg\max_{c=1}^{C} \frac{\exp(\boldsymbol{w}_c^{\top}\boldsymbol{x})}{\sum_{c'=1}^{C}\exp(\boldsymbol{w}_{c'}^{\top}\boldsymbol{x})}\end{aligned}$$

Softmax回归

- 向量形式可以写为

$$\hat{\boldsymbol{y}} = \text{softmax}(\boldsymbol{W}^{\top}\boldsymbol{x})$$

$$= \frac{\exp(\boldsymbol{W}^{\top}\boldsymbol{x})}{\mathbf{1}_C^{\top}\exp(\boldsymbol{W}^{\top}\boldsymbol{x})},$$

  - 其中 $\boldsymbol{W} = [\boldsymbol{w}_1, \cdots, \boldsymbol{w}_C]$ 是由C个类的权重向量组成的矩阵，
  - $\hat{\boldsymbol{y}} \in \mathbb{R}^C$ 为所有类别的预测条件概率组成的向量

参数学习

- 采用交叉熵损失函数，Softmax回归模型的风险函数为

$$\mathcal{R}(\boldsymbol{W}) = -\frac{1}{N}\sum_{n=1}^{N}\sum_{c=1}^{C} \boldsymbol{y}_c^{(n)} \log \hat{\boldsymbol{y}}_c^{(n)}$$

$$= -\frac{1}{N}\sum_{n=1}^{N} (\boldsymbol{y}^{(n)})^{\top} \log \hat{\boldsymbol{y}}^{(n)}, \qquad \hat{\boldsymbol{y}}^{(n)} = \mathrm{softmax}(\boldsymbol{W}^{\top}\boldsymbol{x}^{(n)})$$

- 风险函数 $\mathcal{R}(W)$关于W的梯度为

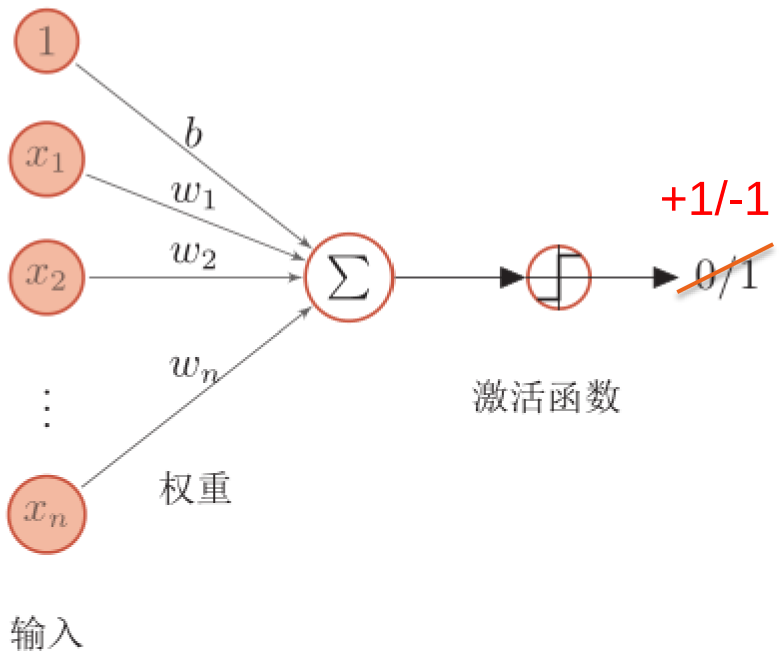
$$\frac{\partial \mathcal{R}(\boldsymbol{W})}{\partial \boldsymbol{W}} = -\frac{1}{N}\sum_{n=1}^{N} \boldsymbol{x}^{(n)} \left(\boldsymbol{y}^{(n)} - \hat{\boldsymbol{y}}^{(n)}\right)^{\top}$$

感知器

感知器

- 由 Frank Roseblatt 于 1957 年提出， 是一种广泛使用的线性分类器. 感知器可谓是最简单的人工神经网络，只有一个神经元
- 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。

$$\hat{y} = \mathrm{sgn}(\boldsymbol{w}^{\top}\boldsymbol{x}) = \begin{cases} +1 & 当\mathbf{w}^{\mathrm{T}}\mathbf{x} > 0 \\ -1 & 当\mathbf{w}^{\mathrm{T}}\mathbf{x} \leq 0 \end{cases},$$



感知器

- 学习算法
  - 感知器的学习算法是一种错误驱动的在线学习算法:
  - 先初始化一个权重向量$\boldsymbol{w} \leftarrow 0$（通常是全零向量）；
  - 每次分错一个样本$(\boldsymbol{x}, y)$时，即

$$y\boldsymbol{w}^T\boldsymbol{x} < \mathbf{0}$$

  - 用这个样本来更新权重

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

  - 根据感知器的学习策略，可以反推出感知器的损失函数为

$$\mathcal{L}(\boldsymbol{w}; \boldsymbol{x}, y) = \max(0, -y\boldsymbol{w}^\top\boldsymbol{x})$$

感知器

- 采用随机梯度下降，其每次更新的梯度为

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

感知器的学习过程

算法 3.1　两类感知器的参数学习算法

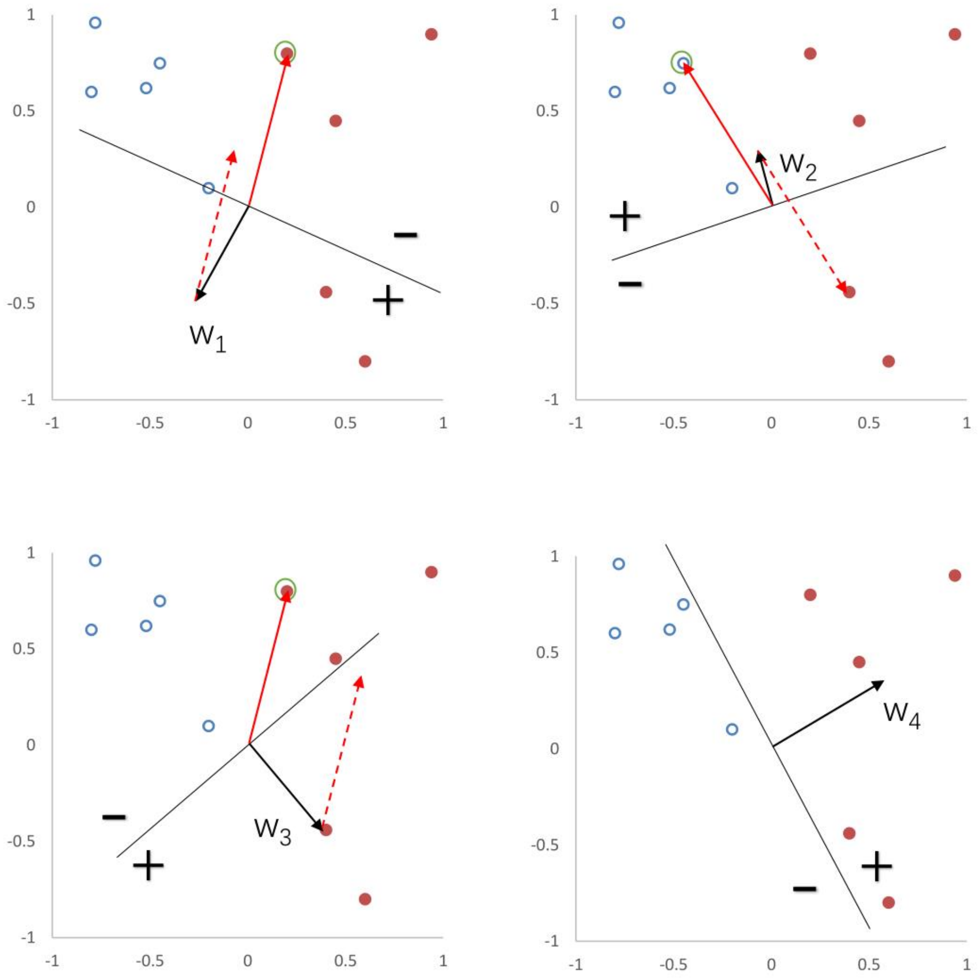
输入: 训练集 $\mathcal{D} = \{(\boldsymbol{x}^{(n)}, y^{(n)})\}_{n=1}^{N}$，最大迭代次数 T

初始化：$\boldsymbol{w}_0 \leftarrow 0, k \leftarrow 0, t \leftarrow 0$；
repeat
　对训练集 $\mathcal{D}$ 中的样本随机排序;
　for $n = 1 \cdots N$ **do**
　　选取一个样本 $(\boldsymbol{x}^{(n)}, y^{(n)})$;
　　if $\boldsymbol{w}_k^{\mathrm{T}}(y^{(n)}\boldsymbol{x}^{(n)}) \leq 0$ **then**　　表示分错
　　　$\boldsymbol{w}_{k+1} \leftarrow \boldsymbol{w}_k + y^{(n)}\boldsymbol{x}^{(n)}$;
　　　$k \leftarrow k + 1$;
　　end
　　$t \leftarrow t + 1$;
　　if $t = T$ **then** break;　　// 达到最大迭代次数
　end
until $t = T$;
输出: $\boldsymbol{w}_k$

对比Logistic回归的更新方式:

$$\boldsymbol{w}_{t+1} \leftarrow \boldsymbol{w}_t + \alpha \frac{1}{N} \sum_{n=1}^{N} \boldsymbol{x}^{(n)} \left(y^{(n)} - \hat{y}_{\boldsymbol{w}_t}^{(n)} \right)$$

感知器参数学习的更新过程



支持向量机

支持向量机

- 是在分类与回归分析中分析数据的监督式学习模型与相关的学习算法。
- 支持向量机的分类方法，是在一组分布中找出一个超平面作为决策边界，使模型在数据上的分类误差尽量接近于零
- 支持向量机在线性和非线性分类中，效果都非常好。

支持向量机

- 给定一个二分类器数据集

$$\mathcal{D} = \{(\boldsymbol{x}^{(n)}, y^{(n)})\}_{n=1}^{N} \qquad y_n \in \{+1, -1\}$$

- 如果两类样本是线性可分的，即存在一个超平面

$$\boldsymbol{w}^{\top}\boldsymbol{x} + b = 0$$

- 将两类样本分开，那么对于每个样本都有

$$y^{(n)}(\boldsymbol{w}^{\top}\boldsymbol{x}^{(n)} + b) > 0$$

- 每个样本 $x^{(n)}$ 到分割超平面的距离为

$$\gamma^{(n)} = \frac{|\boldsymbol{w}^{\top}\boldsymbol{x}^{(n)} + b|}{\|\boldsymbol{w}\|} = \frac{y^{(n)}(\boldsymbol{w}^{\top}\boldsymbol{x}^{(n)} + b)}{\|\boldsymbol{w}\|}$$

支持向量机

- 支持向量机的目标是寻找一个超平面$(\boldsymbol{w}^*, b^*)$使得γ最大，即

$$\max_{\boldsymbol{w},b} \quad \gamma$$
$$\text{s.t.} \quad \frac{y^{(n)}(\boldsymbol{w}^\top \boldsymbol{x}^{(n)} + b)}{\|\boldsymbol{w}\|} \geq \gamma, \forall n \in \{1, \cdots, N\}.$$

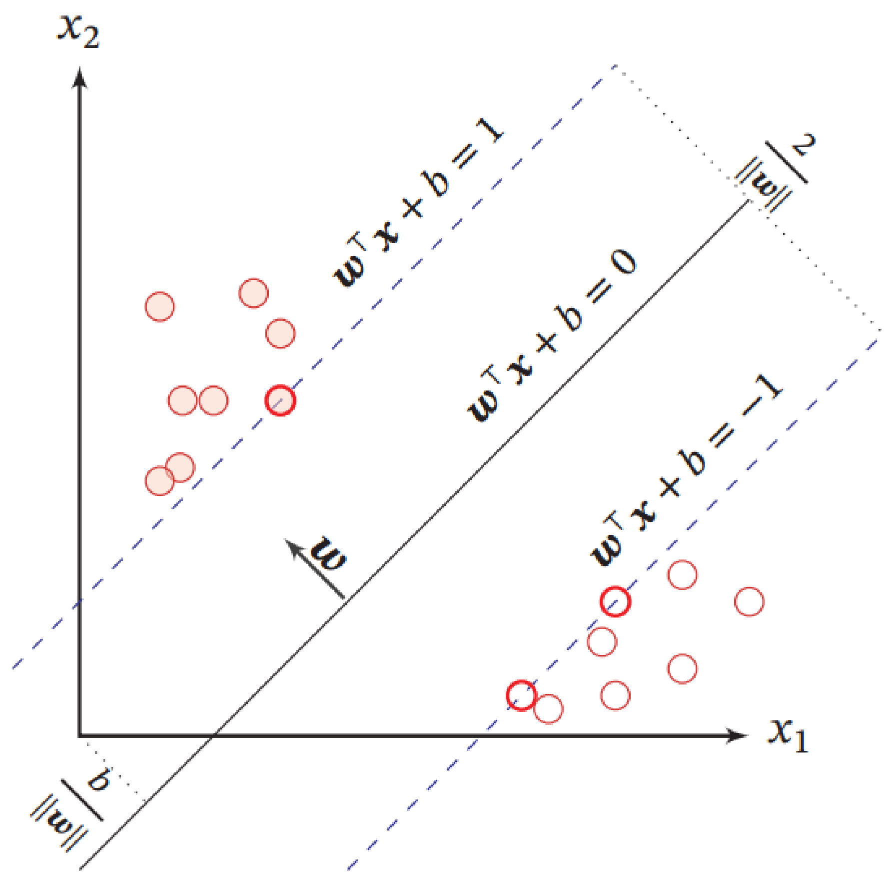
- 等价于

$$\max_{\boldsymbol{w},b} \quad \frac{1}{\|\boldsymbol{w}\|^2}$$
$$\text{s.t.} \quad y^{(n)}(\boldsymbol{w}^\top \boldsymbol{x}^{(n)} + b) \geq 1, \forall n \in \{1, \cdots, N\}.$$

- 数据集中所有满足$y^{(n)}(\boldsymbol{w}^\top \boldsymbol{x}^{(n)} + b) = 1$的样本点，都称为**支持向量**

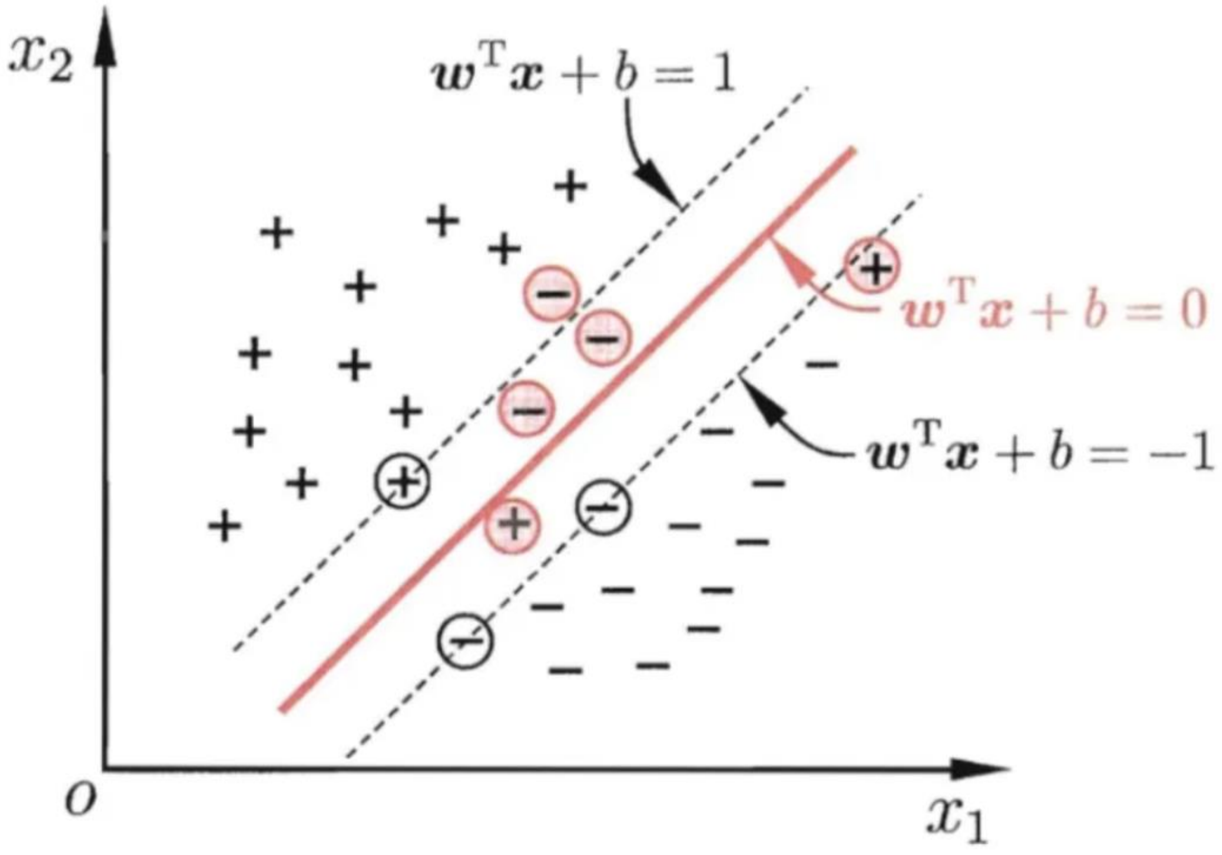
支持向量机

- 其分割超平面有很多个，但是间隔最大的超平面是唯一的

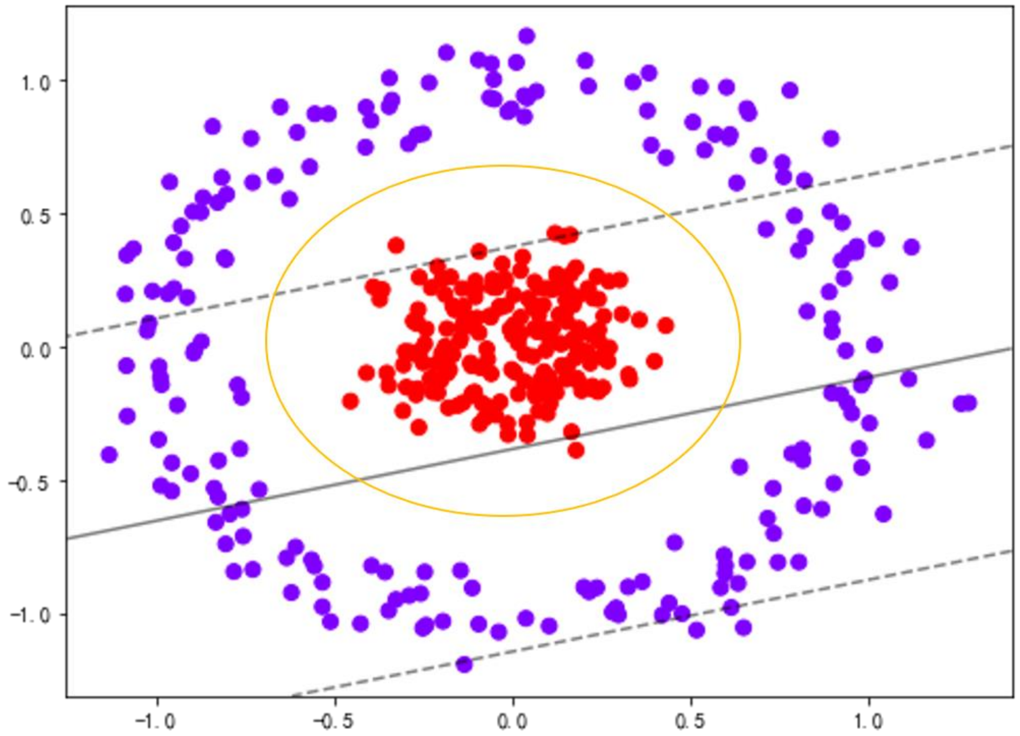


支持向量机

- 在最大化间隔的同时，不满足约束的样本应尽可能地少

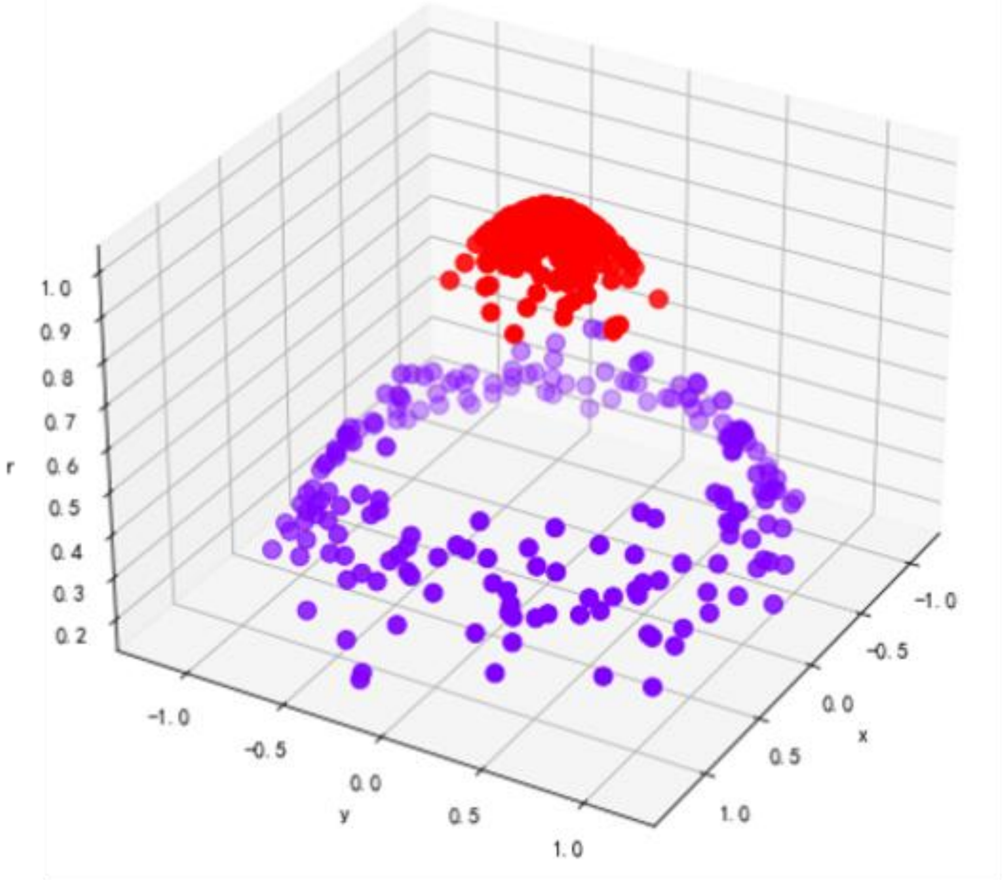


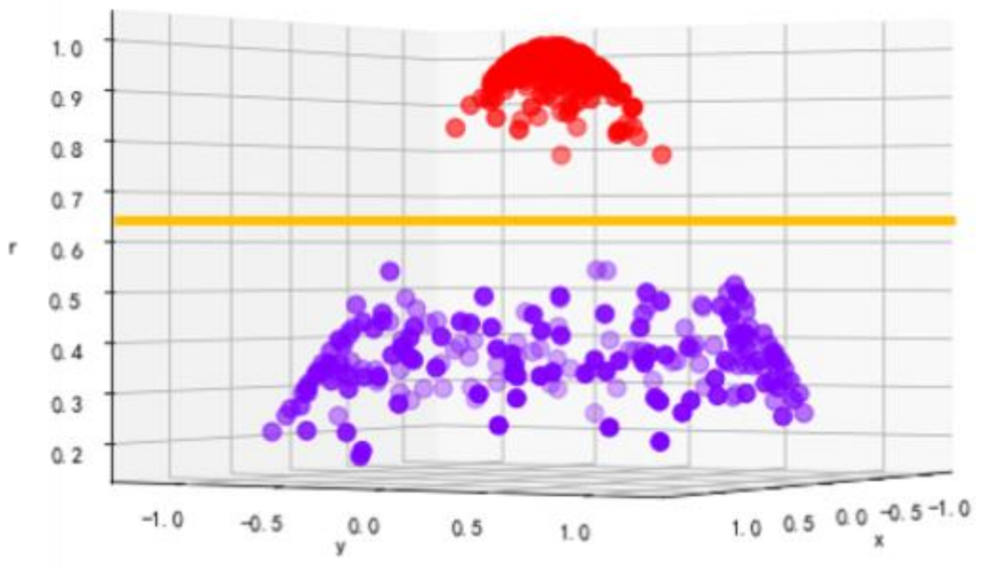
非线性支持向量机



非线性支持向量机

- 原理利用核技巧，将数据从原始的空间投射到新空间中
- 这种非线性变换，将非线性问题变换成线性问题





总结

线性分类模型小结

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