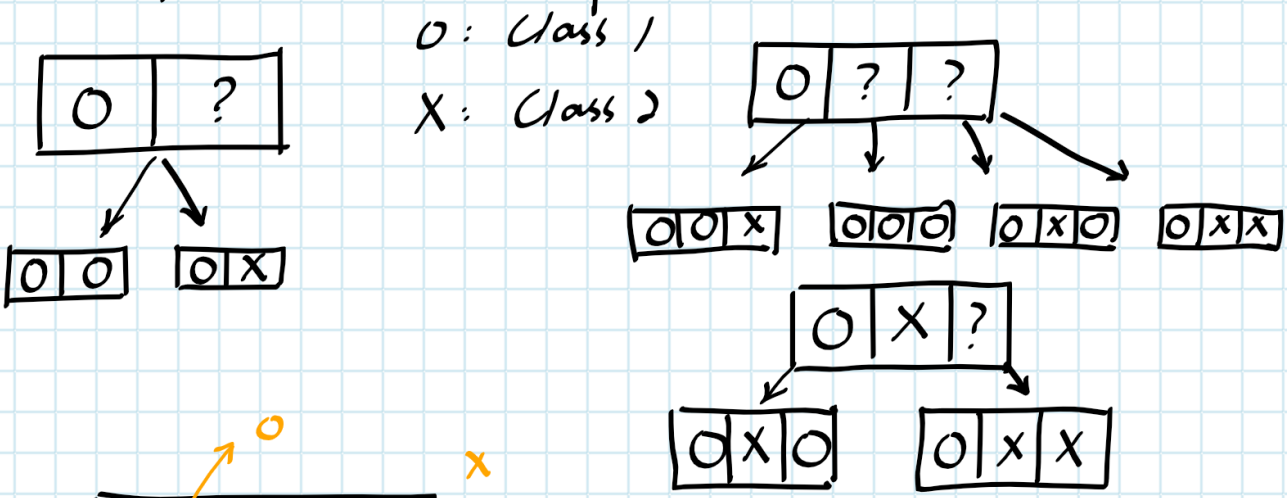
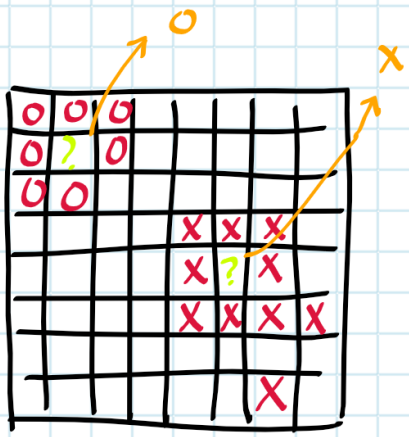
没有免费午餐定理（No Free Lunch Theorem）

↳ 如果我们不对特征空间有先验假设，则所有算法的平均表现是一样的！





特征差距小的样本更有可能是同一类～

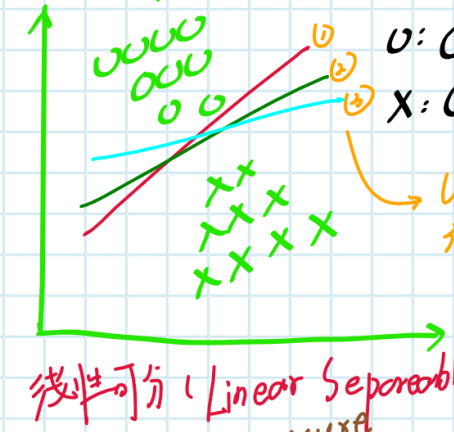
支持向量机（Support Vector Machine）

Vapnik

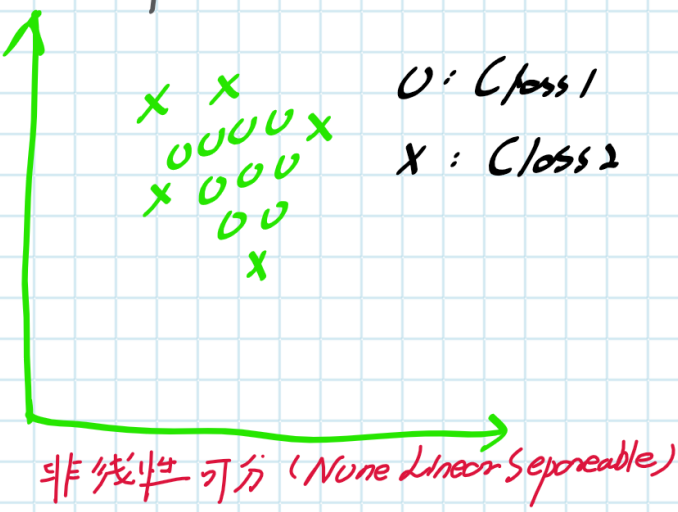
→ 小样本方法

↳ 小样本依旧能获得很好的结果!!!

线性模型



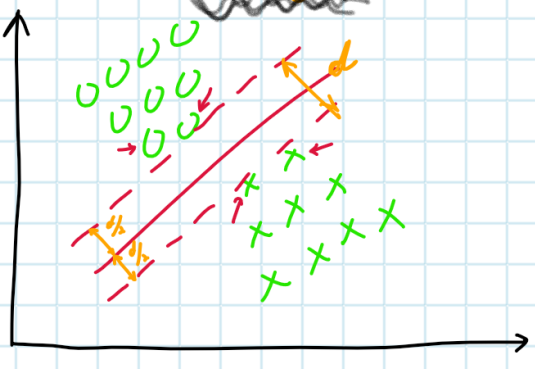
线性可分（Linear Seporable）



非线性可分（None Linear Seporeable）

performance measure 性能指标

Why?



d: Margin

支持向量（Support Vectors）

“↗”所指的数据

定义：

① 训练数据及标签 $(x_1, y_1)(x_2, y_2)\cdots(x_n, y_n)$
x：向量　y：标签 $(y=+1/-1)$

② 线性模型 (W, b)　$W^Tx + b = 0$（超平面 Hyperplane）
W：向量　b：参数　常数

$$X=\begin{bmatrix}x_1\\x_2\\\vdots\\x_n\end{bmatrix}\quad w=\begin{bmatrix}w_1\\w_2\\\vdots\\w_n\end{bmatrix}$$

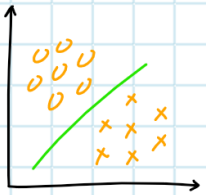
③ 一个训练集线性可分定义：

$\{(x_i, y_i)\}_{i=1\sim N}$

$\exists (w, b)$，使，对 $\forall i=1\sim N$，有：

(a) 若 $y_i=+1$，则 $w^Tx_i+b>0$

(b) 若 $y_i=-1$，则 $w^Tx_i+b<0$



$y_i[w^Tx_i+b]>0$ ---- 公式1

优化问题：（凸优化问题，二次规划问题）

最小化（Minimize）：$\|w\|^2\cdot\frac{1}{2}$ 求导方便

限制条件（Subject to）：$y_i(w^Tx_i+b)\geq 1,\ i=1\sim N$

事实1：$w^Tx+b=0$ 与 $aw^Tx+ab=0$ 是同一个平面　$a\in R^+$

若 (w, b) 满足公式1，则 (aw, ab) 也满足公式1

事实2：点到面的距离公式：

平面：$w_1x+w_2y+b=0$，则点 (x_0, y_0) 到此面的距离 $d=\frac{|w_1x_0+w_2y_0+b|}{\sqrt{w_1^2+w_2^2}}$

向量 x_0 到超平面 $w^Tx+b=0$ 的距离 $d=\frac{|w^Tx_0+b|}{\|w\|}$，$\|w\|=\sqrt{w_1^2+w_2^2+\cdots+w_n^2}$

我们可以对 a 进行缩放 $(w, b)\rightarrow(aw, ab)$

求给定在支持向量 x_0 上有：$|w^Tx_0+b|=1$

此时支持向量与平面距离：$d=\frac{1}{\|w\|}$

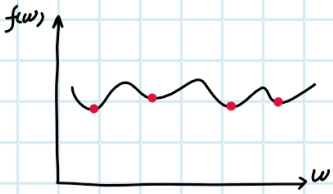
二次规划（Quadratic Programming）：

① 目标函数（Objective Function）是二次的

② 限制条件是一次的

要么无解，要么只有一个极值

⇒ 全局最优解



⇒ Hard to Solve

① SVM 是最大化间隔（Margin）的分类算法

② 优化问题

训练样本 $\{(x_i, y_i)\}_{i=1\sim N}$　x：向量　y：标签

③ 最小化：$\frac{1}{2}\|w\|^2$

限制条件：$y_i[w^Tx_i+b]\geq 1\ (i=1\sim N)$

SVM 处理非线性：

① 最小化：$\frac{1}{2}\|w\|^2+C\sum_{i=1}^{N}\xi_i$　ξ_i（Slack Variable）松弛变量

② 限制条件：$y_i[w^Tx_i+b]\geq 1-\xi_i\ (i=1\sim N)$

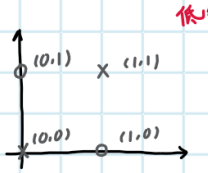
$\xi_i\geq 0$

$C\sum_{i=1}^{N}\xi_i$：正则项（Regulation Term）

C：事先设定的参数

定义一个高维的映射 $\varphi(x)$：

x（低维）$\xrightarrow{\varphi}$ $\varphi(x)$（高维）



⇒ 异或问题

$x_1=\begin{bmatrix}0\\0\end{bmatrix}\in C_1\quad x_2=\begin{bmatrix}1\\1\end{bmatrix}\in C_1\quad x_3=\begin{bmatrix}1\\0\end{bmatrix}\in C_2\quad x_4=\begin{bmatrix}0\\1\end{bmatrix}\in C_2$

$\varphi(x)$：$x=\begin{bmatrix}a\\b\end{bmatrix}\xrightarrow{\varphi}\varphi(x)=\begin{bmatrix}a'\\b'\\a\\b\\ab\end{bmatrix}$

$\varphi(x_1)=\begin{bmatrix}0\\0\\0\\0\\0\end{bmatrix}\quad\varphi(x_2)=\begin{bmatrix}1\\1\\1\\1\\1\end{bmatrix}\in C_1$

$\varphi(x_3)=\begin{bmatrix}1\\0\\1\\0\\0\end{bmatrix}\quad\varphi(x_4)=\begin{bmatrix}0\\1\\0\\1\\0\end{bmatrix}\in C_2$

我们可以不知道无限维映射 $\varphi(x)$ 的显式表达，只需要知道一个核函数（Kernel Function）

$K(x_1, x_2)=\varphi(x_1)^T\varphi(x_2)$

则①式优化式依然可解

核函数

① $K(x_1, x_2)=e^{-\frac{\|x_1-x_2\|^2}{2\sigma^2}}=\varphi(x_1)^T\varphi(x_2)$（高斯核）

② $K(x_1, x_2)=(x_1^Tx_2+1)^d=\varphi(x_1)^T\varphi(x_2)$（多项式核）

$K(x_1, x_2)\Rightarrow\varphi(x_1)^T\varphi(x_2)$ 的充要条件：

① $K(x_1, x_2)=K(x_2, x_1)$（交换性）

② $\forall c_i, x_i\ (i=1\sim N)$，有（半正定性）

$\sum_{i}^{N}\sum_{j}^{N}c_ic_jK(x_i, x_j)\geq 0$

优化理论：

① 《Convex optimization》

② 《Nonlinear Programming》

原问题（Prime Problem） ⟶ 非等式

求小化： $f(w)$

限制条件： $g_i(w) \le 0 \quad (i=1\sim K)$

$h_i(w) = 0 \quad (i=1\sim M)$

对偶问题（Dual Problem）：

① 定义： $L(w,\alpha,\beta)$

$= f(w) + \sum_{i=1}^{K}\alpha_i g_i(w) + \sum_{i=1}^{M}\beta_i h_i(w)$

$= f(w) + \alpha^T g(w) + \beta^T h(w)$

$g(w) = \begin{bmatrix} g_1(w) \\ g_2(w) \\ \vdots \\ g_K(w) \end{bmatrix}$ $h(w) = \begin{bmatrix} h_1(w) \\ h_2(w) \\ \vdots \\ h_M(w) \end{bmatrix}$

② 对偶问题的定义：（$\inf$ → 求最小值）

求大化： $\theta(\alpha,\beta) = \inf_{\text{所有}w}\{L(w,\alpha,\beta)\}$

限制条件： $\alpha_i \ge 0 \quad (i=1\sim K)$

定理：如果 w^* 是原问题的解，而 α^*, β^* 是对偶问题的解，则有：

$f(w^*) \ge \theta(\alpha^*, \beta^*)$

证： $\theta(\alpha^*,\beta^*) = \inf_{\text{所有}w}\{L(w,\alpha^*,\beta^*)\} \le L(w^*,\alpha^*,\beta^*)$

$= f(w^*) + \sum_{i=1}^{K}\alpha_i^* g_i(w^*) + \sum_{i=1}^{M}\beta_i^* h_i(w^*) \le f(w^*)$

（$\alpha_i^* \ge 0$，$g_i(w^*) \le 0$，$h_i(w^*) = 0$）

定义： $G = f(w^*) - \theta(\alpha^*,\beta^*) \ge 0$

G 叫做原问题与对偶问题的间距（Duality Gap）

对于某些特定的优化问题，可以证明： $G = 0$

强对偶定理：

若 $f(w)$ 为凸函数，且 $g(w) = Aw+b$，$h(w) = Cw+b$，

则此优化问题的原问题与对偶问题间距为 0，即 $f(w^*) = \theta(\alpha^*,\beta^*)$

对于 $\forall i = 1\sim K$，或者 $\alpha_i^* = 0$，或者 $g_i(w^*) = 0$

（KKT 条件）

将支持向量机的原问题转化为对偶问题

求小化 $\frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N}\xi_i$

限制条件 ① $y_i[w^T\varphi(x_i)+b] \ge 1-\xi_i$

② $\xi_i \ge 0$

⟶

$\frac{1}{2}\|w\|^2 - C\sum_{i=1}^{N}\xi_i$

① $y_i[w^T\varphi(x_i)+b] \le 1+\xi_i$

② $\xi_i \le 0$

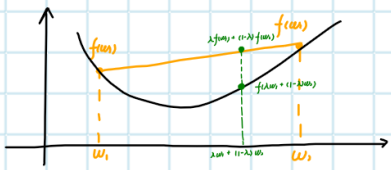
↓

$\frac{1}{2}\|w\|^2 - C\sum_{i=1}^{N}\xi_i$

① $1+\xi_i - y_i w^T\varphi(x_i) - y_i b \le 0$

② $\xi_i \le 0$

关于凸函数：



$\forall w_1, w_2, \forall \lambda \in [0,1]$，

$f(\lambda w_1 + (1-\lambda)w_2) \le \lambda f(w_1) + (1-\lambda) f(w_2)$ } 数学定义

对偶问题：

求大化 $\theta(\alpha,\beta) = \inf_{\text{所有}(w,\xi,b)}\left\{\frac{1}{2}\|w\|^2 - C\sum_{i=1}^{N}\xi_i + \sum_{i=1}^{N}\alpha_i[1+\xi_i - y_i w^T\varphi(x_i) - y_i b]\right\}$

限制条件 $\alpha_i \ge 0 \quad \beta_i \ge 0 \quad (i=1\sim N)$

$\begin{cases} \frac{\partial L}{\partial w} = 0 \Rightarrow w - \sum_{i=1}^{N}\alpha_i y_i \varphi(x_i) = 0 \\ \frac{\partial L}{\partial \xi_i} = 0 \Rightarrow -C + \beta_i + \alpha_i = 0 \\ \frac{\partial L}{\partial b} = 0 \Rightarrow -\sum_{i=1}^{N}\alpha_i y_i = 0 \end{cases}$

$w = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_m \end{bmatrix}$ $f(w) \to$ 数

$\frac{\partial f}{\partial w} = \begin{bmatrix} \frac{\partial f}{\partial w_1} \\ \frac{\partial f}{\partial w_2} \\ \vdots \\ \frac{\partial f}{\partial w_m} \end{bmatrix}$

若 $f(w) = \pm\|w\|^2$ 则 $\frac{\partial f}{\partial w} = w$

若 $f(w) = w^T x$ 则 $\frac{\partial f}{\partial w} = x$

求大化： $\theta(\alpha,\beta) = \sum_{i=1}^{N}\alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j k(x_i,x_j)$

限制条件： ① $0 \le \alpha_i \le C$

② $\sum_{i=1}^{N}\alpha_i y_i = 0$

凸优化问题 ↓ SMO 算法

测试流程：

测试样本 x，

若 $w^T\varphi(x)+b \ge 0$，则 $y = +1$

若 $w^T\varphi(x)+b < 0$，则 $y = -1$

SVM 算法：

① 训练流程

输入训练样本 $\{(x_i,y_i)\}_{i=1\sim N}$

（解优化问题）

求大化： $\theta(\alpha) = \sum_{i=1}^{N}\alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j x_i y_j K(x_i,x_j)$

限制条件： ① $0 \le \alpha_i \le C$

② $\sum_{i=1}^{N}\alpha_i y_i = 0$

算 b，找一个 $0 < \alpha_i < C$，

$b = \frac{1 - y_i\sum_{j=1}^{N}\alpha_j y_j K(x_i,x_j)}{y_i}$

② 测试流程

输入测试样本 X

若 $\sum_{i=1}^{N}\alpha_i y_i K(x_i,X) + b \ge 0$，则 $y = +1$

若 $\sum_{i=1}^{N}\alpha_i y_i K(x_i,X) + b < 0$，则 $y = -1$

SVM 内核函数：

Linear（线性内核）：$K(x,y) = x^T y$

Poly（多项式核）：$K(x,y) = (x^T y + 1)^d$

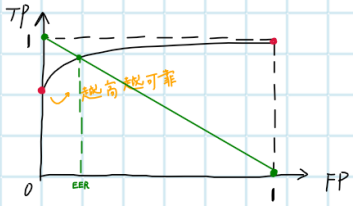
Rdf（高斯径向基函数核）：$K(x,y) = e^{\frac{\|x-y\|^2}{\sigma^2}}$

Tanh（Tanh核）：$K(x,y) = \tanh(\beta x^T y + b)$　$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$

- 混淆矩阵

		预测	
实际		正样本	负样本
	正样本	True Positive（TP）	False Negative（FN）
	负样本	False Positive（FP）	True Negative（TN）

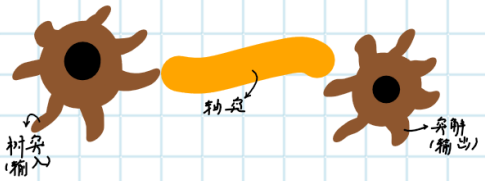
1. TP + FN = FP + TN = 1
2. 对于同一个系统，若TP↗，则FP↗

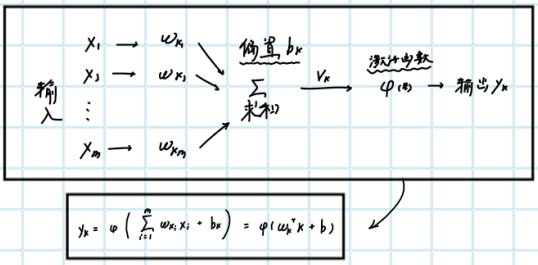


等错误率（Equal Error Rate）：

⇓

用于判别系统好坏





$$y_k = \varphi\left(\sum_{i=1}^{p} w_{ki} x_i + b_k\right) = \varphi(w_k^T x + b)$$

感知器算法 (Perceptron Algorithm):

① 取一个选择 w, b

取一个训练样本 (X, y)

② 若 $w^T x + b > 0$ 且 $y = -1$, 则 (目前的 w, b 对 x 不能正确分类)

$w = w - x \quad b = b - 1$

$w^T x + b$ 新 $= (w - x)^T x + (b - 1)$

$= (w^T x + b) - (\|x\|^2 + 1)$

③ 若 $w^T x + b < 0$ 且 $y = +1$, 则

$w = w + x \quad b = b + 1$

$(w^T x + b) + (\|x\|^2 + 1)$

④ 再取一个 (X, y), 回到 ②

⑤ 终止条件: 直到所有输入输出对都不满足 ② 中的 ③ 和 ④ 之一, 则退出循环。

数学证明感知器算法的有效性:

定义一个增广的量 $\vec{x}$:

② 若 $y = +1$, 则 $\vec{x} = \begin{bmatrix} x \\ 1 \end{bmatrix}$

④ 若 $y = -1$, 则 $\vec{x} = \begin{bmatrix} x \\ -1 \end{bmatrix}$

定义一个增广的 w:

$$w = \begin{bmatrix} w \\ b \end{bmatrix}$$

输入 $\vec{x_i}$

① 随机取 w

② 把一个 $\vec{x_i}$

若 $w^T \vec{x_i} < 0$, 则 $w = w + \vec{x_i}$

④ 跳回 ②, 直到所有的 $\vec{x_i}$ 都不成立

感知器算法收敛定理:

输入 $\{\vec{x_i}\}_{i=1\sim N}$, 若线性可分, 即 $\exists w_{opt}$, 使

$w_{opt}^T \vec{x_i} > 0 \quad (i = 1 \sim N)$

则利用上述感知器算法, 经过有限步后, 得到一个 w, 使

$w^T \vec{x_i} > 0 \quad (i = 1 \sim N)$

证明: 不失一般性, 设 $\|w_{opt}\| = 1$ (即 w_{opt} 与 αw_{opt} 是同一个平面)

假设第 k 步的 w 是 $w(k)$, 且有一个 $\vec{x_i}$, 使

$w(k)^T \vec{x_i} < 0$

根据感知器算法:

$$w(k+1) = w(k) + \vec{x_i} \Rightarrow \|w(k+1) - \alpha w_{opt}\|^2 = \|w(k) + \vec{x_i} - \alpha w_{opt}\|^2$$

$$= \|(w(k) - \alpha w_{opt}) + \vec{x_i}\|^2$$

$$= \|w(k) - \alpha w_{opt}\|^2 + \|\vec{x_i}\|^2 + 2\underbrace{w(k)^T \vec{x_i}}_{<0} - 2\alpha \underbrace{w_{opt}^T \vec{x_i}}_{>0}$$

一定可以取一个很大的 α, 使:

$$\|w(k+1) - \alpha w_{opt}\|^2 < \|w(k) - \alpha w_{opt}\|^2$$

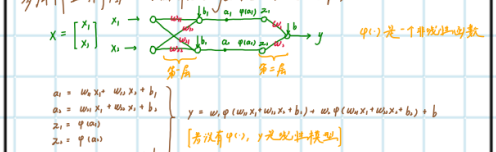
定义 $\beta = \max_{i=1\sim N} \{\|\vec{x_i}\|^2\}$, $\gamma = \min_{i=1\sim N} (w_{opt}^T x_i)$

取 $\alpha = \frac{\beta^2 + 1}{2\gamma}$, 则:

$$\|w(k+1) - \alpha w_{opt}\|^2 < \|w(k) - \alpha w_{opt}\|^2 - 1$$

取 $D = \|w(0) - \alpha w_{opt}\|$, 则至多经过 D^2 步, w 将会收敛到 αw_{opt}

多层神经网络 (Multiple Layer Neural Networks)

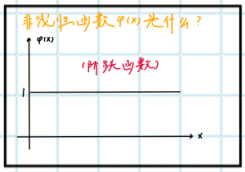


$\varphi(\cdot)$ 是一个非线性函数

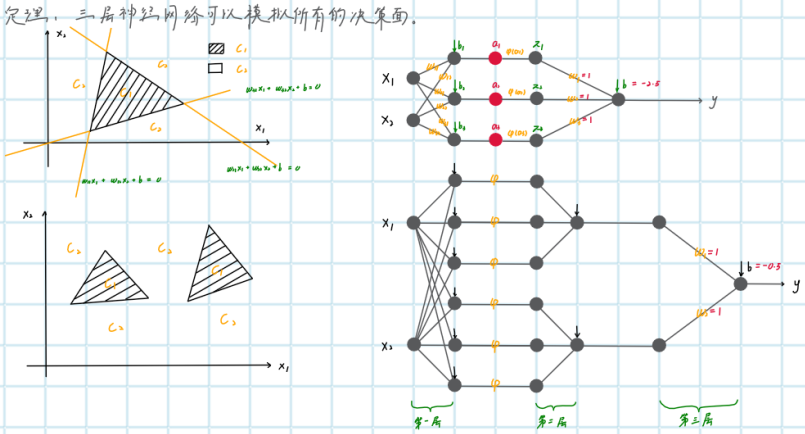
$$\left.\begin{aligned} a_1 &= w_{11} x_1 + w_{12} x_2 + b_1 \\ a_2 &= w_{21} x_1 + w_{22} x_2 + b_2 \\ z_1 &= \varphi(a_1) \\ z_2 &= \varphi(a_2) \\ y &= w_1 z_1 + w_2 z_2 + b \end{aligned}\right\} \quad y = w_1 \varphi(w_{11} x_1 + w_{12} x_2 + b_1) + w_2 \varphi(w_{21} x_1 + w_{22} x_2 + b_2) + b$$

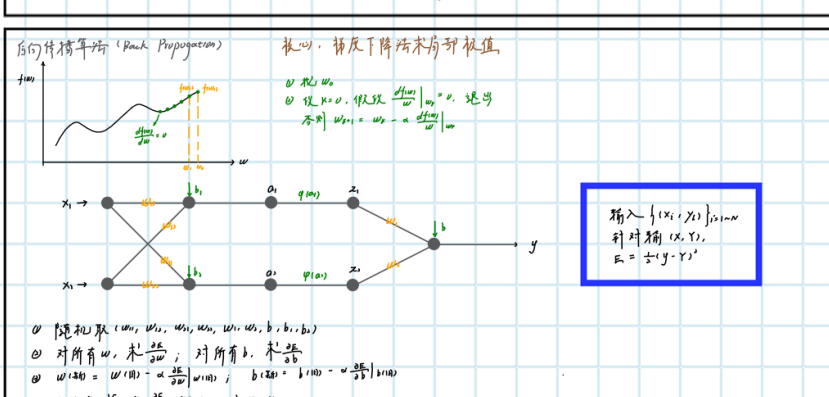
[若没有 $\varphi(\cdot)$, y 是线性模型]

非线性函数 $\varphi(x)$ 是什么?



定理: 三层神经网络可以模拟所有的决策面。





后向传播算法 (Back Propagation)

概述: 梯度下降法求局部极值

① 随机 w_0

② 设 $k=0$, 假设 $\left.\frac{df(w)}{dw}\right|_{w_k} = 0$, 退出

否则 $w_{k+1} = w_k - \alpha \left.\frac{df(w)}{dw}\right|_{w_k}$

输入 $\{(x_i, y_i)\}_{i=1\sim N}$

对对输出 (x, Y),

$E = \frac{1}{2}(y - Y)^2$

① 随机取 $(w_{11}, w_{12}, w_{21}, w_{22}, w_1, w_2, b, b_1, b_2)$

② 对所有 w, 求 $\frac{\partial E}{\partial w}$; 对所有 b, 求 $\frac{\partial E}{\partial b}$

③ $w(\text{新}) = w(\text{旧}) - \alpha \left.\frac{\partial E}{\partial w}\right|_{w(\text{旧})}$; $b(\text{新}) = b(\text{旧}) - \alpha \left.\frac{\partial E}{\partial b}\right|_{b(\text{旧})}$

④ 当所有 $\frac{\partial E}{\partial w}$ 与 $\frac{\partial E}{\partial b}$ 都为 0 时退出.