



# 信息熵

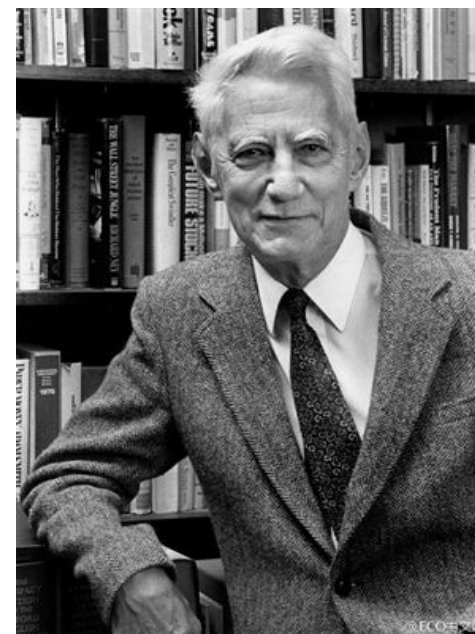
(Information Entropy)

刘远超

哈尔滨工业大学  
计算机科学与技术学院

# 信息论中的熵(entropy)

- 热力学中的熵: 是表示分子状态混乱程度的物理量
- 信息论中的熵: 用来描述信源的不确定性的**大小**
- 经常使用的熵概念有下列几种:
  - 信息熵
  - 交叉熵
  - 相对熵
  - 条件熵
  - 互信息



克劳德·艾尔伍德·香农（Claude Elwood Shannon，1916年4月30日—2001年2月24日）是美国数学家、信息论的创始人。1936年获得密歇根大学学士学位。1940年在麻省理工学院获得硕士和博士学位，1941年进入贝尔实验室工作。香农提出了信息熵的概念，为信息论和数字通信奠定了基础。

# 信息熵

- 信源信息的不确定性函数  $f$  通常满足两个条件：

1)是概率  $p$  的单调递减函数。

2)两个独立符号所产生的不确定性应等于各自不确定性之和，即  $f(p_1, p_2) = f(p_1) + f(p_2)$ 。

- 对数函数同时满足这两个条件： $f(p) = \log \frac{1}{p} = -\log p$

- **信息熵**：要考虑信源所有可能发生情况的平均不确定性。若信源符号有  $n$  种取值： $U_1, \dots, U_i, \dots, U_n$ ，对应概率为  $p_1, \dots, p_i, \dots, p_n$ ，且各种出现彼此独立。此时信源的平均不确定性应当为单个符号不确定性 $-\log p_i$ 的统计平均值(E)，称为信息熵，即

$$H(U) = E[-\log p_i] = -\sum_{i=1}^n p_i \log p_i = \sum_{i=1}^n p_i \log \left( \frac{1}{p_i} \right)$$


# 交叉熵(cross entropy)

- **定义：**交叉熵是信息论中一个重要的概念,用于表征两个变量概率分布  $P, Q$ （假设  $P$  表示真实分布,  $Q$  为模型预测的分布）的差异性。交叉熵越大,两个变量差异程度越大。
- **交叉熵公式：**

$$H(P, Q) = - \sum_{x \in X} P(x) \log Q(x) = \sum_{x \in X} P(x) \log \frac{1}{Q(x)}$$

# 相对熵(relative entropy)

- 也称为KL散度(Kullback–Leibler divergence, 简称KLD)、信息散度(information divergence)、信息增益(information gain)。
- **相对熵的定义：**是交叉熵与信息熵的差值。表示用分布Q模拟真实分布P，所需的额外信息。
- 计算公式为

$$D_{KL}(P||Q) = \sum_{x \in X} P(x) \log \frac{1}{Q(x)} - \sum_{x \in X} P(x) \log \frac{1}{P(x)} = \sum_{x \in X} P(x) \log \frac{P(x)}{Q(x)}$$


交叉熵

信息熵

# 相对熵(relative entropy)举例

- **举例：**假设某字符发射器随机发出0和1两种字符。且其真实发出概率分布为A。现在有两人的观察概率分布B与C。各个分布如下：

$$A(0)=1/2, A(1)=1/2$$

$$B(0)=1/4, B(1)=3/4$$

$$C(0)=1/8, C(1)=7/8$$

则B和C哪个更接近实际分布A？

- **求解过程：**

用公式  $D_{KL}(P||Q) = \sum_{x \in X} P(x) \log \frac{P(x)}{Q(x)}$ ，则

$$\blacksquare D_{KL}(A||B) = \frac{1}{2} \log \left( \frac{1/2}{1/4} \right) + \frac{1}{2} \log \left( \frac{1/2}{3/4} \right)$$

$$\blacksquare D_{KL}(A||C) = \frac{1}{2} \log \left( \frac{1/2}{1/8} \right) + \frac{1}{2} \log \left( \frac{1/2}{7/8} \right)$$

结果：

$$\blacksquare D_{KL}(A||B) = 0.14,$$

$$\blacksquare D_{KL}(A||C) = 0.41$$

# 相对熵的性质

- 相对熵（KL散度）有两个主要的性质：

- 相对熵（KL散度）**不具有对称性**，即 $D_{KL}(P||Q) \neq D_{KL}(Q||P)$ 。

例如  $D_{KL}(A||B) = \frac{1}{2} \log\left(\frac{1/2}{1/4}\right) + \frac{1}{2} \log\left(\frac{1/2}{3/4}\right) = \mathbf{0.1438},$

$$D_{KL}(B||A) = \frac{1}{4} \log\left(\frac{1/4}{1/2}\right) + \frac{3}{4} \log\left(\frac{3/4}{1/2}\right) = \mathbf{0.1308}$$

**即 $D_{KL}(A||B) \neq D_{KL}(B||A)$**

- 相对熵**具有非负性**。即 $D_{KL}(P||Q) \geq 0$

# JS散度

- **JS散度(Jensen–Shannon divergence)**具有对称性:

由于KL散度不具对称性，因此JS散度在KL散度的基础上进行了改进。

现有两个分布p1和p2，其JS散度公式为：

$$JS(P_1||P_2) = \frac{1}{2}KL(P_1||\frac{P_1+P_2}{2}) + \frac{1}{2}KL(P_2||\frac{P_1+P_2}{2})$$



# 联合熵

- 联合熵 (复合熵, Joint Entropy):
  - 用 $H(X, Y)$ 表示
  - 两个随机变量 $X$ ,  $Y$ 的联合分布的熵, 形成联合熵

# 条件熵

- 条件熵（ the conditional entropy ）： $H(X|Y)$ 表示在已知随机变量Y的条件下随机变量X的不确定性。

■  $H(X|Y) = H(X, Y) - H(Y)$ ，表示(X, Y)的联合熵，减去Y单独发生包含的熵。

推导过程：

① 假设已知 $y = y_j$ ，则  $H(x|y_j) = -\sum_{i=1}^n p(x_i|y_j) \log p(x_i|y_j)$

② 对于y的各种可能值，需要根据出现概率做加权平均。即

$$\begin{aligned} H(x|y) &= -\sum_{i=1}^n \sum_{j=1}^m p(y_j) p(x_i|y_j) \log p(x_i|y_j) \\ &= -\sum_{i=1}^n \sum_{j=1}^m p(x_i, y_j) \log \frac{p(x_i, y_j)}{p(y_j)} \\ &= H(x, y) - H(y) \end{aligned}$$

# 互信息

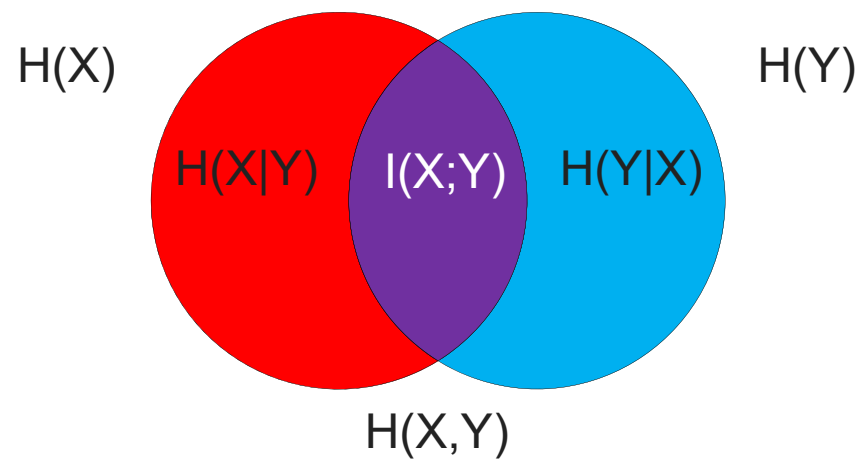
- **互信息(Mutual Information)**可以被看成是一个随机变量中包含的关于另一个随机变量的信息量，或者说是一个随机变量由于已知另一个随机变量而减少的不确定性。

$$\begin{aligned} I(X; Y) &= H(X) - H(X|Y) \\ &= H(X) + H(Y) - H(X, Y) \end{aligned}$$

$$\begin{aligned} &= \sum_x p(x) \log \frac{1}{p(x)} + \sum_y p(y) \log \frac{1}{p(y)} - \sum_{x,y} p(x, y) \log \frac{1}{p(x, y)} \\ &= \sum_{x,y} p(x, y) \log \frac{p(x, y)}{p(x)p(y)} \end{aligned}$$

即互信息 $I(X; Y)$ 是联合分布 $p(x, y)$ 与乘积分布 $p(x)p(y)$ 的相对熵

# 文氏图图解



# Thanks!





# 反向传播中的梯度

(Gradient in Backpropagation)

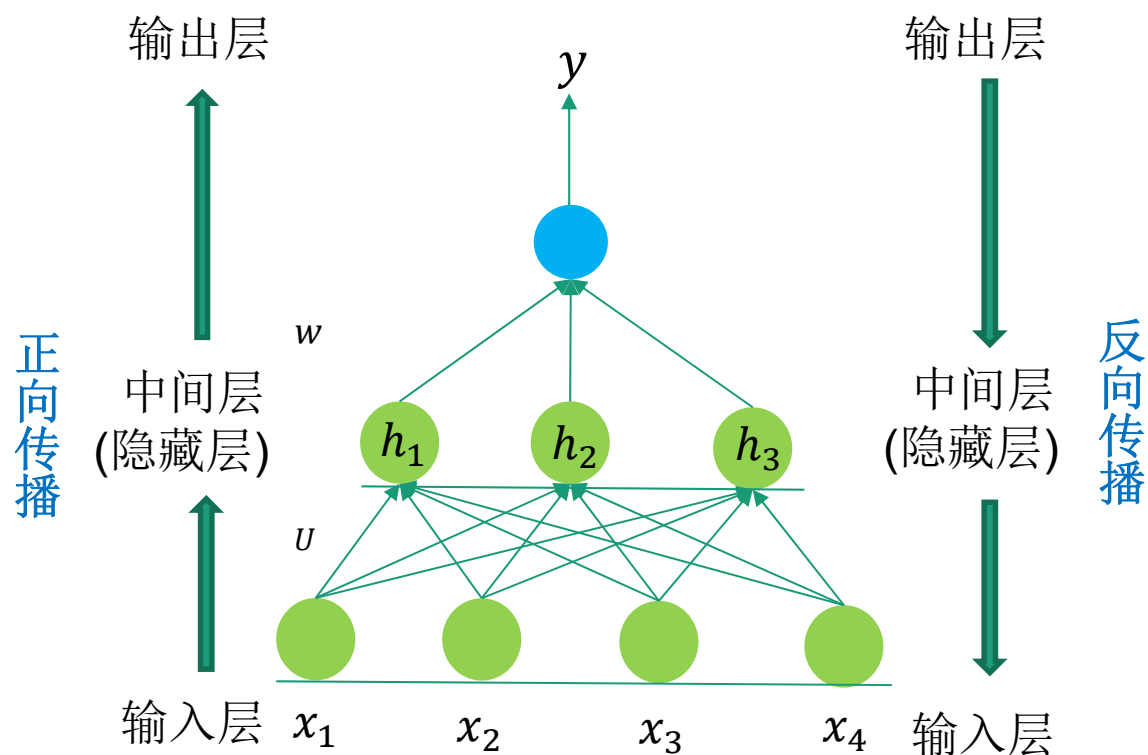
刘远超

哈尔滨工业大学

计算机科学与技术学院

# 反向传播 (backpropagation) 中的梯度

反向传播 (BP) 算法的学习过程由正向传播过程和反向传播过程组成。



反向传播需要通过递归调用链规则(chain rule)计算表达式的梯度。

# 梯度的简单解释

● 一般形式:  $\frac{df(x)}{dx} = \lim_{h \rightarrow 0} \frac{f(x+h) - f(x)}{h}$

■ **举例1:** 对于函数  $f(x, y) = xy$ ,  $\frac{\partial f}{\partial x} = y$ ,  $\frac{\partial f}{\partial y} = x$ , 由于梯度  $\nabla f$  实际上是偏导向量, 因此我们有

$$\nabla f = \left[ \frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right] = [y, x]$$

■ **举例2:** 对于函数  $\max(x, y)$ , 有

$$\frac{\partial f}{\partial x} = 1(x \geq y), \quad \frac{\partial f}{\partial y} = 1(y \geq x)$$



## 使用链规则对复合表达式(Compound expressions)求导

例：假设复合表达式为 $f(u, v, w) = (u + v)w$ 。令 $q = u + v$ ，则 $f = qw$ ，

则有 $\frac{\partial f}{\partial q} = w$ ,  $\frac{\partial f}{\partial w} = q$ ,  $\frac{\partial q}{\partial u} = 1$ ,  $\frac{\partial q}{\partial v} = 1$ ,

使用链规则，则有 $\frac{\partial f}{\partial u} = \frac{\partial f}{\partial q} \cdot \frac{\partial q}{\partial u} = w * 1$ ,  $\frac{\partial f}{\partial v} = \frac{\partial f}{\partial q} \cdot \frac{\partial q}{\partial v} = w * 1$ ,  $\frac{\partial f}{\partial w} = q$

➤ **u = 1; v = 2; w = -3**

➤ # 正向传播

➤ **q = u + v**

➤ **f = q \* w**

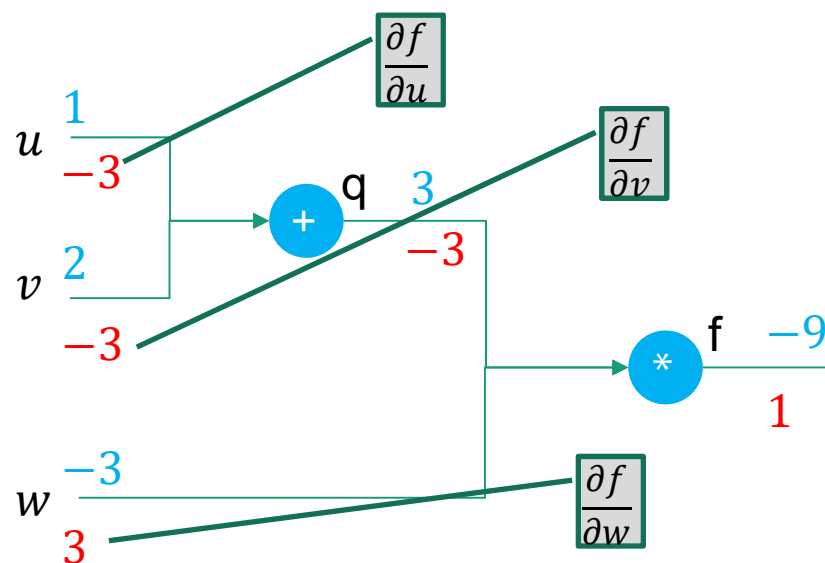
➤ # 反向传播

➤ **dfdq = w**

➤ **dfdu = 1.0 \* dfdq**

➤ **dfdv = 1.0 \* dfdq**

➤ **dfdw = q**



# Sigmoid 门的反向传播

- sigmoid函数 $\sigma(x) = \frac{1}{1+e^{-x}}$ , 则  $\frac{d\sigma(z)}{dz} = \frac{e^{-z}}{(1+e^{-z})^2} = \left(\frac{1+e^{-z}-1}{1+e^{-z}}\right) \left(\frac{1}{1+e^{-z}}\right) = (1 - \sigma(z))\sigma(z)$
- 假设有 $f(w, x) = \frac{1}{1+e^{-(w_0x_0+w_1x_1+w_2)}}$ , 则其正、反向传播过程为:

➤import math

➤w = [3, 4, 5]

➤x = [6, 7]

➤# 正向传播

➤dot = w[0]\*x[0] + w[1]\*x[1] + w[2]

➤f = 1.0 / (1 + math.exp(-dot))

➤# 反向传播

➤dfddot = (1 - f) \* f

➤dfdx = [w[0] \* dfddot, w[1] \* dfddot]

➤dfdw = [x[0] \* dfddot, x[1] \* dfddot, 1.0 \* dfddot]

$$\triangleright df/dx = \frac{df}{ddot} \frac{ddot}{dx} = (1 - f)f(w_0, w_1)$$

# 矩阵-矩阵相乘的梯度

矩阵-矩阵相乘的梯度需要注意维度和转置操作。例如,  $D = W \cdot X$

➤ `import numpy as np`

➤ `# 正向传播`

➤ `W = np.random.randn(5, 10)`

➤ `X = np.random.randn(10, 2)`

➤ `D = W.dot(X)`

➤ `# 反向传播`

➤ `dD = np.random.randn(*D.shape)`

➤ `dDdW = dD.dot(X.T)`

➤ `dDdX = W.T.dot(dD)`

# Thanks!





# 感知机

(Perceptron)

刘远超

哈尔滨工业大学

计算机科学与技术学院

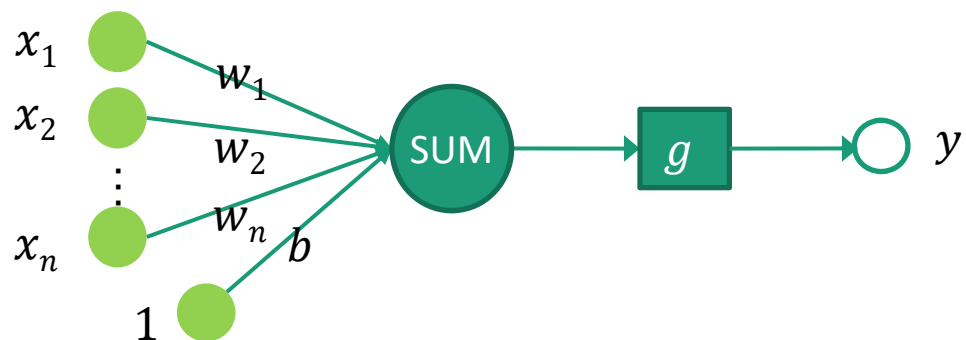
# 什么是感知机 (perceptron)

- 感知机由Rosenblatt提出，是神经网络的基础。
- 感知机是两类分类的线性分类模型。假设输入为实例样本的特征向量 $x$ ，输出为实例样本的类别 $y$ 。则由输入空间到输出空间的如下函数称之为感知机。

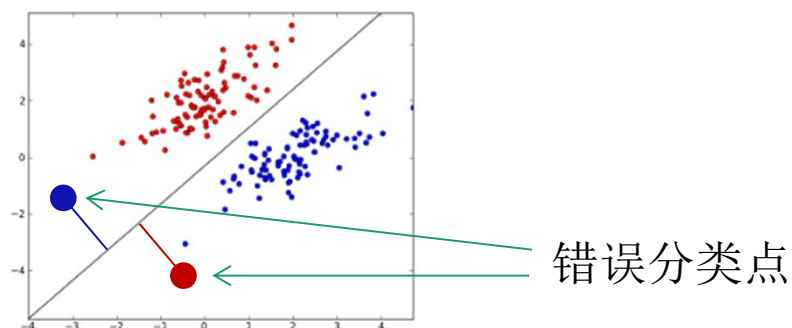
$$y = g(\sum_{i=1}^n w_i x_i + b) \text{ 或 } y = g(w \cdot x + b)$$

$g$ 为激励函数，以达到对样本分类的目的。Rosenblatt的感知器用阶跃函数作为激励函数，其函数公式如下：

$$g(z) = \begin{cases} +1, & z \geq 0 \\ -1, & z < 0 \end{cases}, \text{ 其中 } z = w \cdot x + b$$



# 感知机模型的损失函数



- 损失函数的选择采用误分类点到超平面的距离（这里采用点到直线的距离）： $\frac{1}{\|w\|} |w * x_i + b|$ ，其中 $\|w\|$ 是L2范数。
- 对于误分类点 $(x_i, y_i)$ 来说： $-y_i(w * x_i + b) > 0$ 总是成立。则误分类点到超平面的距离可以写为 $-\frac{1}{\|w\|} y_i(w * x_i + b)$ ，从而所有误分类点到超平面的总距离为 $-\frac{1}{\|w\|} \sum_{x_i \in M} y_i(w * x_i + b)$ 。
- 不考虑 $\frac{1}{\|w\|}$ ，就得到极小化损失函数 $L(w, b) = -\sum_{x_i \in M} y_i(w * x_i + b)$ 。其中M为误分类点的集合。

# 感知机模型的优化-随机梯度下降法

- 优化目的：找到使损失函数 $L(\theta) = L(\mathbf{w}, \mathbf{b}) = -\sum_{x_i \in M} y_i(\mathbf{w} * \mathbf{x}_i + \mathbf{b})$ 变小的参数 $\theta$ 。
- 模型参数 $\theta$ ，即 $\mathbf{w}$ 和 $\mathbf{b}$ ，的更新公式为

$$\theta := \theta - \eta \nabla_{\theta} L(\theta), \quad (\text{其中}\eta\text{是步长, } 0 < \eta \leq 1, \nabla_{\theta} L(\theta)\text{是梯度})$$

对 $\mathbf{w}$ 和 $\mathbf{b}$ 求的两个偏导分别为

$$\frac{\partial L(\theta)}{\partial \mathbf{w}} = -\sum_{x_i \in M} y_i \mathbf{x}_i, \quad \frac{\partial L(\theta)}{\partial \mathbf{b}} = -\sum_{x_i \in M} y_i$$

感知机算法的损失函数极小化过程是每次随机选择一个误分类点使其梯度下降。因此，随机选择一个误分类点 $(\mathbf{x}_i, y_i)$ ，对 $\mathbf{w}$ 和 $\mathbf{b}$ 进行更新。即

$$\mathbf{w} := \mathbf{w} + \eta y_i \mathbf{x}_i, \quad \mathbf{b} := \mathbf{b} + \eta y_i$$



# 感知机学习算法

**输入：** 训练数据集  $T = \{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}$ , 其中  $x_i \in R^n, y_i \in \{-1, +1\}, i = 1, 2, \dots, N$ . 学习速率  $\eta$  ( $0 < \eta \leq 1$ )

**输出：** 感知机  $y = g(w \cdot x + b)$  的参数  $(w, b)$

**训练学习过程：**

- (1). 选取初始值  $w_0, b_0$
- (2). 在训练集中选取数据  $(x_i, y_i)$
- (3). 如果  $y_i(w \cdot x_i + b) \leq 0$ , 更新参数  
 $w := w + \eta y_i x_i, b := b + \eta y_i$
- (4). 转至 (2), 直到训练集中没有误分类点。

# 感知机迭代实例--学习笔记

● 已知：

正样本点：  $x_1 = (3, 3)^T$ ,  $x_2 = (4, 3)^T$ ,

负样本点：  $x_3 = (1, 1)^T$ ,

(正例的输出 $y$ 为1，负例的输出 $y$ 为-1)

求感知机模型  $f(x) = g(w \cdot x + b)$ ，其中  $w = (w^{(1)}, w^{(2)})^T$ ， $x = (x^{(1)}, x^{(2)})^T$ ，

## 感知机迭代实例--学习笔记 (续)

解：构建最优化问题 $\min_{w,b} L(w,b) = -\sum_{x_i \in M} y_i(w * x_i + b)$ 。其中M为误分类点的集合。求解 $w, b$ ,  $\eta = 1$

(1) 取初始值 $w_0 = (0,0)^T$ ,  $b_0 = 0$

(2) 对样本点 $x_1 = (3,3)^T$ ,  $y_1(w_0 * x_1 + b_0) = 0$ , 未能正确分类, 更新 $w, b$

$$w_1 := w_0 + \eta y_1 x_1 = (3,3)^T, b_1 := b_0 + \eta y_1 = 1$$

得到线性模型 $f_1 = w_1 * x + b_1 = 3x^{(1)} + 3x^{(2)} + 1$

(3) 对样本点 $x_1, x_2$ , 显然,  $y_i(w_1 * x_i + b_1) > 0$ , 被正确分类;

对样本点 $x_3 = (1,1)^T$ ,  $y_3(w_1 * x_3 + b_1) < 0$ , 被误分类, 更新 $w, b$ :

$$w_2 := w_1 + \eta y_3 x_3 = (2,2)^T, b_2 := b_1 + \eta y_3 = 0$$

得到线性模型 $f_2 = w_2 * x + b_2 = 2x^{(1)} + 2x^{(2)} + 0$

(4) 如此继续下去, 直到

$$w_7 = (1,1)^T, b_7 = -3, \text{得到线性模型 } w_7 * x + b_7 = 1x^{(1)} + 1x^{(2)} - 3$$

对所有数据点 $y_i(w_7 * x_i + b_7) > 0$ , 没有误分类点, 损失函数达到极小<sup>27</sup>。

# Thanks!

