



大数据成就未来



从BP网络到深度学习 & TensorFlow

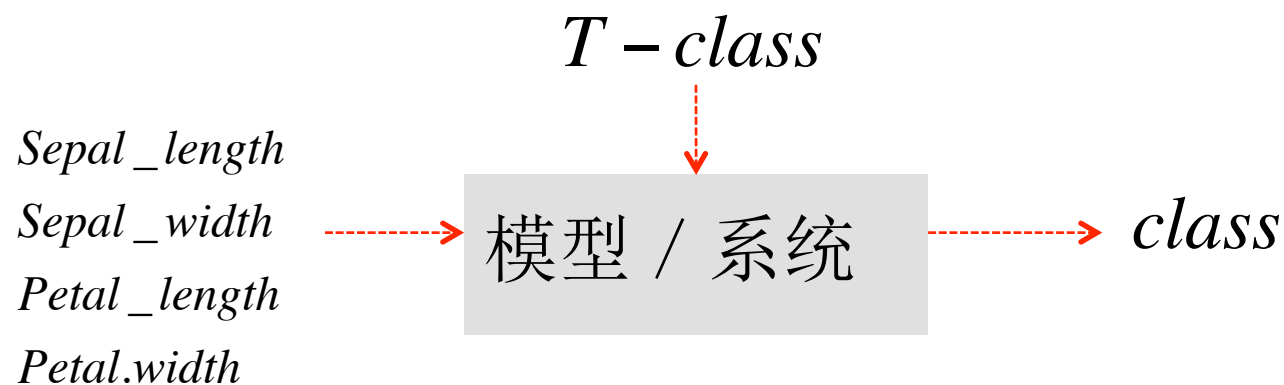
张敏

17/10/26

分类与聚类

- 分类：学习 / 训练过程有监督，训练样本有明确标签

Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	T-class
5.1	3.5	1.4	0.2	setosa
4.9	3	1.4	0.2	setosa
7	3.2	4.7	1.4	versicolor
6.4	3.2	4.5	1.5	versicolor
6.3	3.3	6	2.5	virginica
5.8	2.7	5.1	1.9	virginica
6.5	3	5.8	2.2	?
6.2	2.9	4.3	1.3	?



分类与聚类

- 聚类：学习 / 训练过程无监督，样本无明确标签

Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	
5.1	3.5	1.4	0.2	
4.9	3	1.4	0.2	
7	3.2	4.7	1.4	
6.4	3.2	4.5	1.5	
6.3	3.3	6	2.5	
5.8	2.7	5.1	1.9	
6.5	3	5.8	2.2	
6.2	2.9	4.3	1.3	

Sepal_length

Sepal_width

Petal_length

Petal.width



模型 / 系统

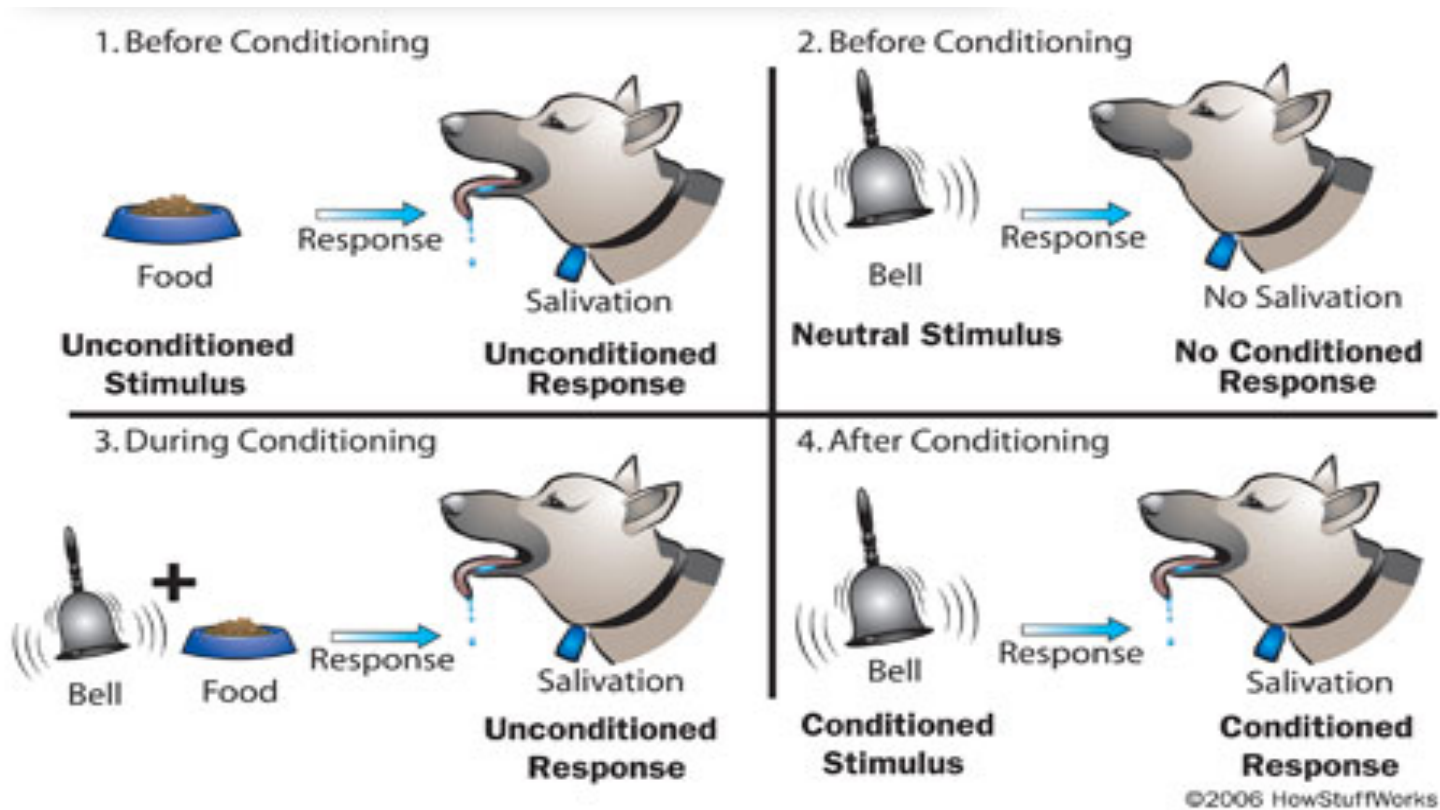


class



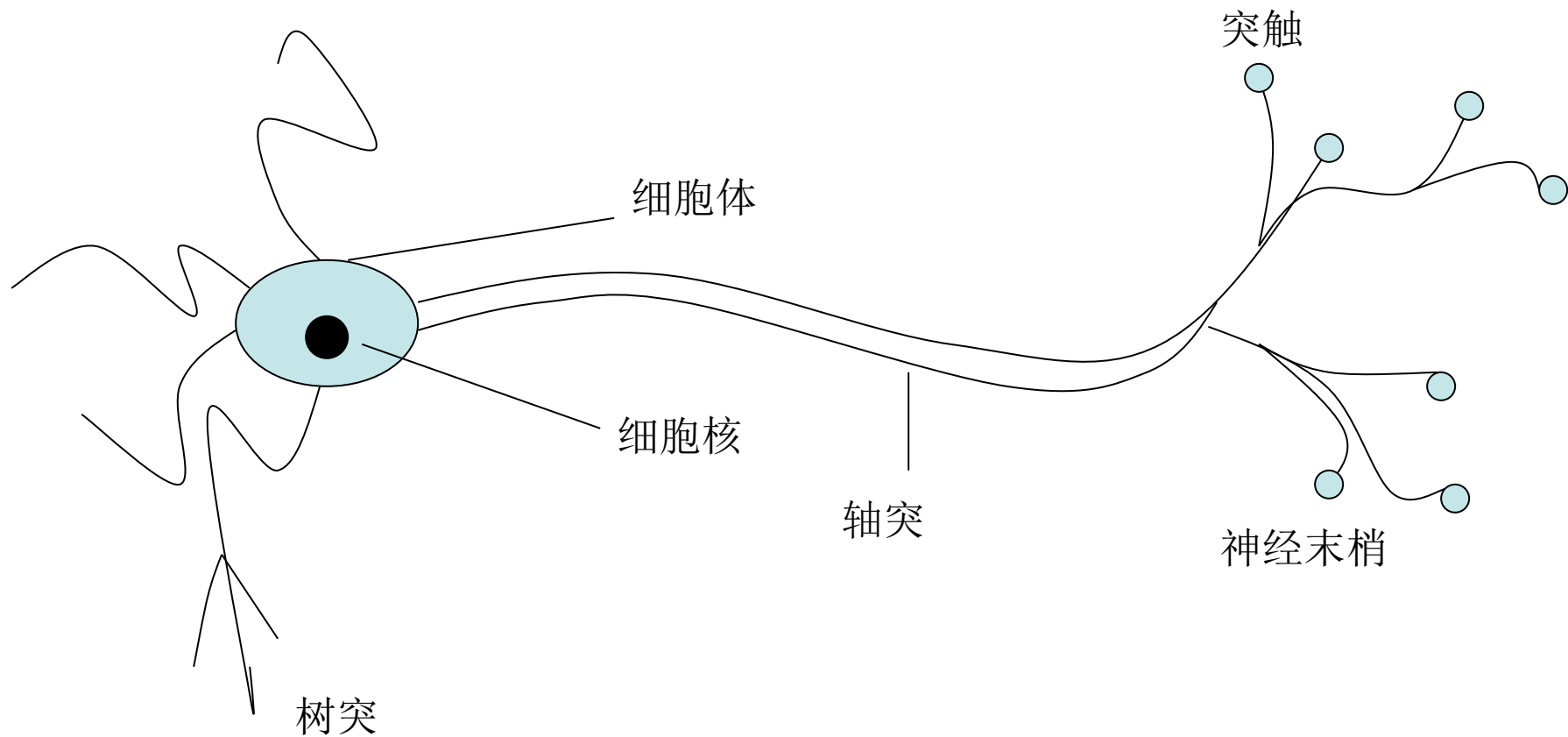
神经网络

巴普洛夫关于神经反射的实验



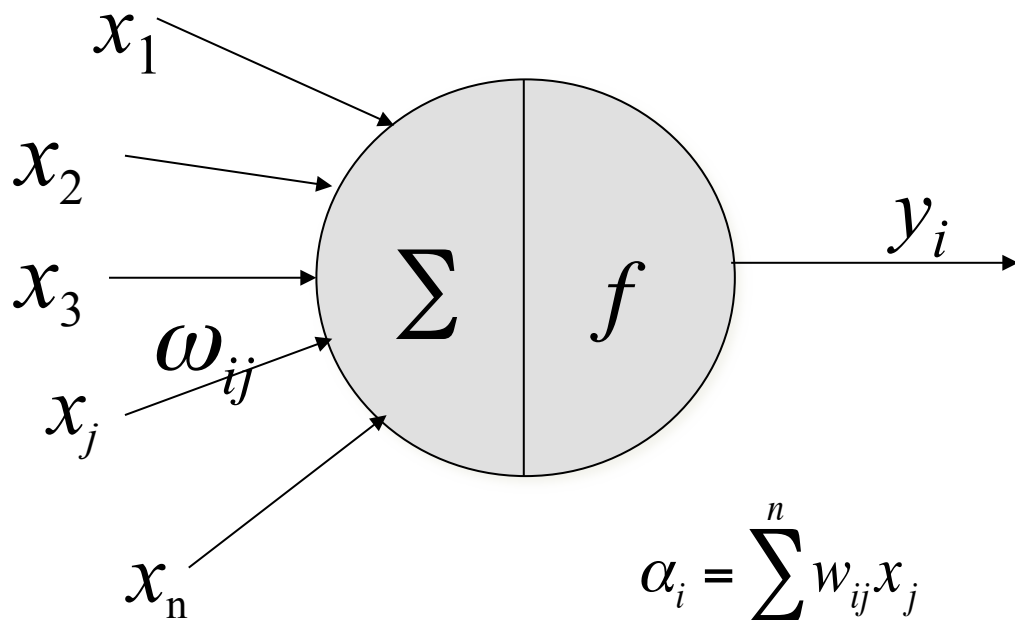
神经网络

生物神经元结构



神经网络

数学神经元结构

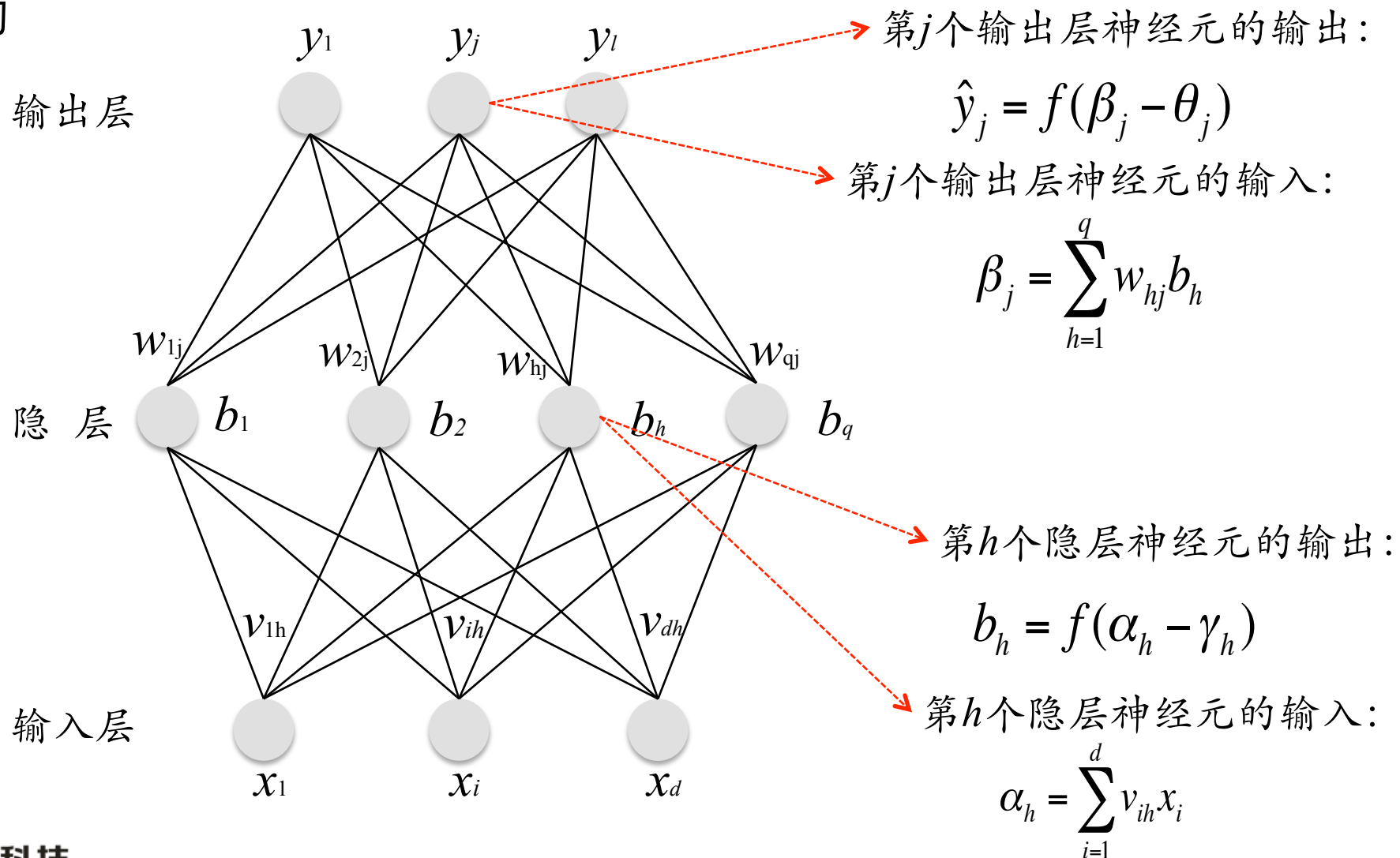


$$\alpha_i = \sum_{j=1}^n w_{ij} x_j \quad y_i = f\left(\sum_{j=1}^n w_{ij} x_j - \theta\right)$$

x_j 为输入信号， f 为传递函数， $w_{i,j}$ 表示与神经元 x_j 连接的权值， y_i 表示输出值， θ 表示阈值

神经网络

BP网络结构

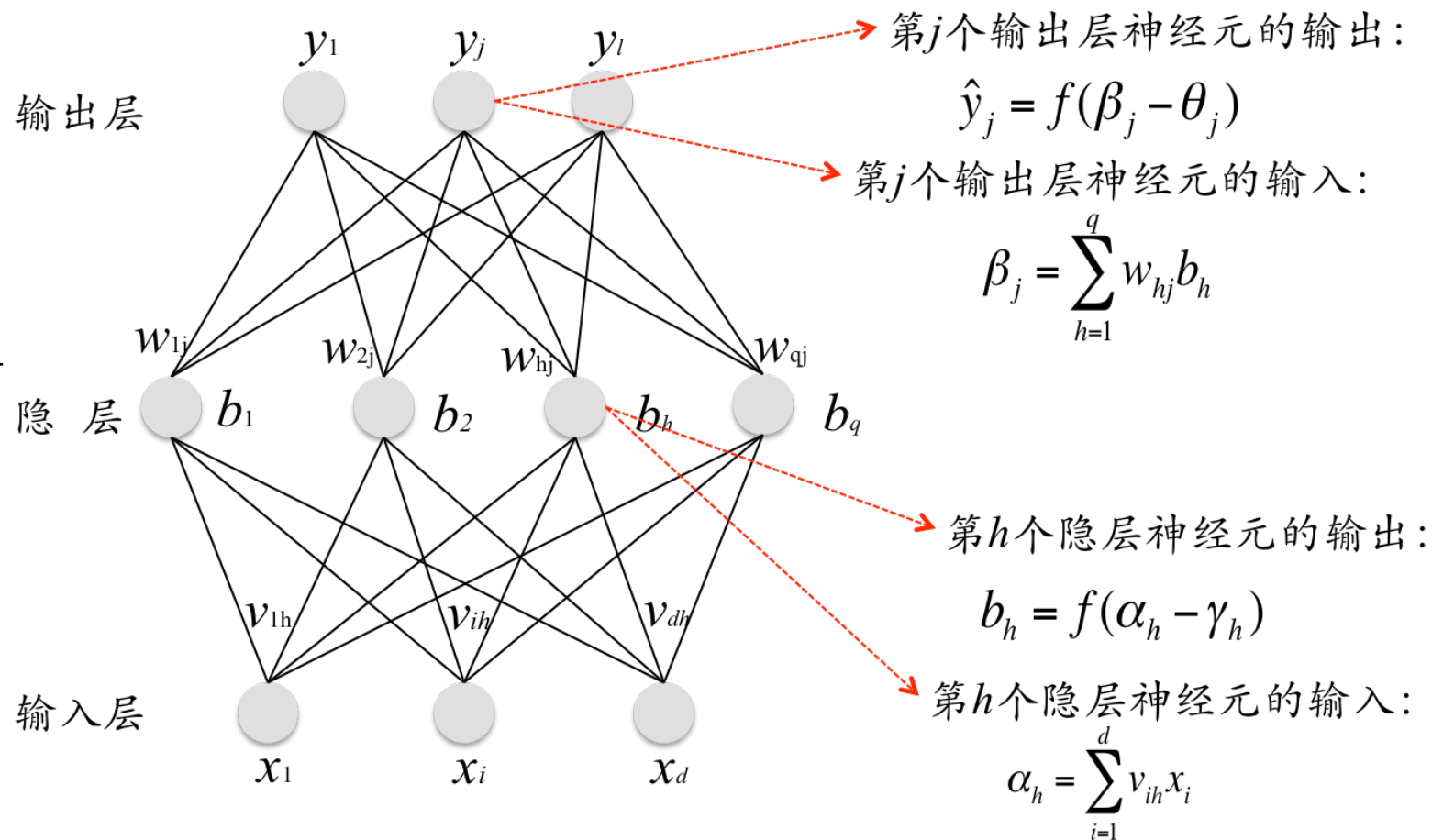
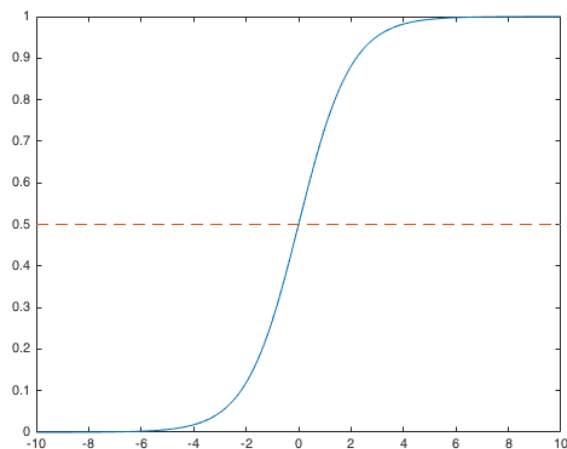


神经网络

BP网络结构

$$E = \frac{1}{2} \sum_{j=1}^l (\hat{y}_j - y_j)^2$$

$$f(x) = \text{sigmoid}(x) = \frac{1}{1 + e^{-x}}$$



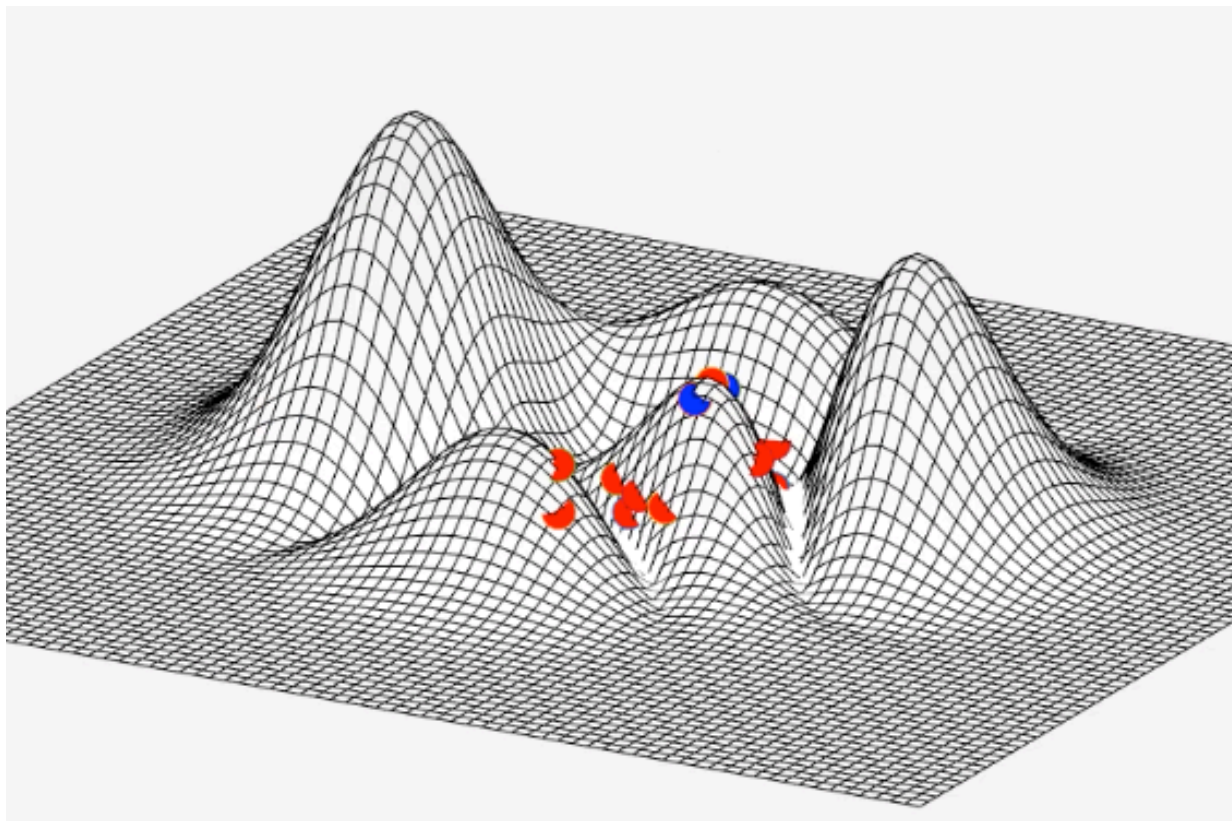
神经网络

BP网络结构

$$E = \frac{1}{2} \sum_{j=1}^l (\hat{y}_j - y_j)^2$$

网络训练目标：

- 找出合适的权值和阈值，使得误差 E 最小



神经网络

BP网络结构

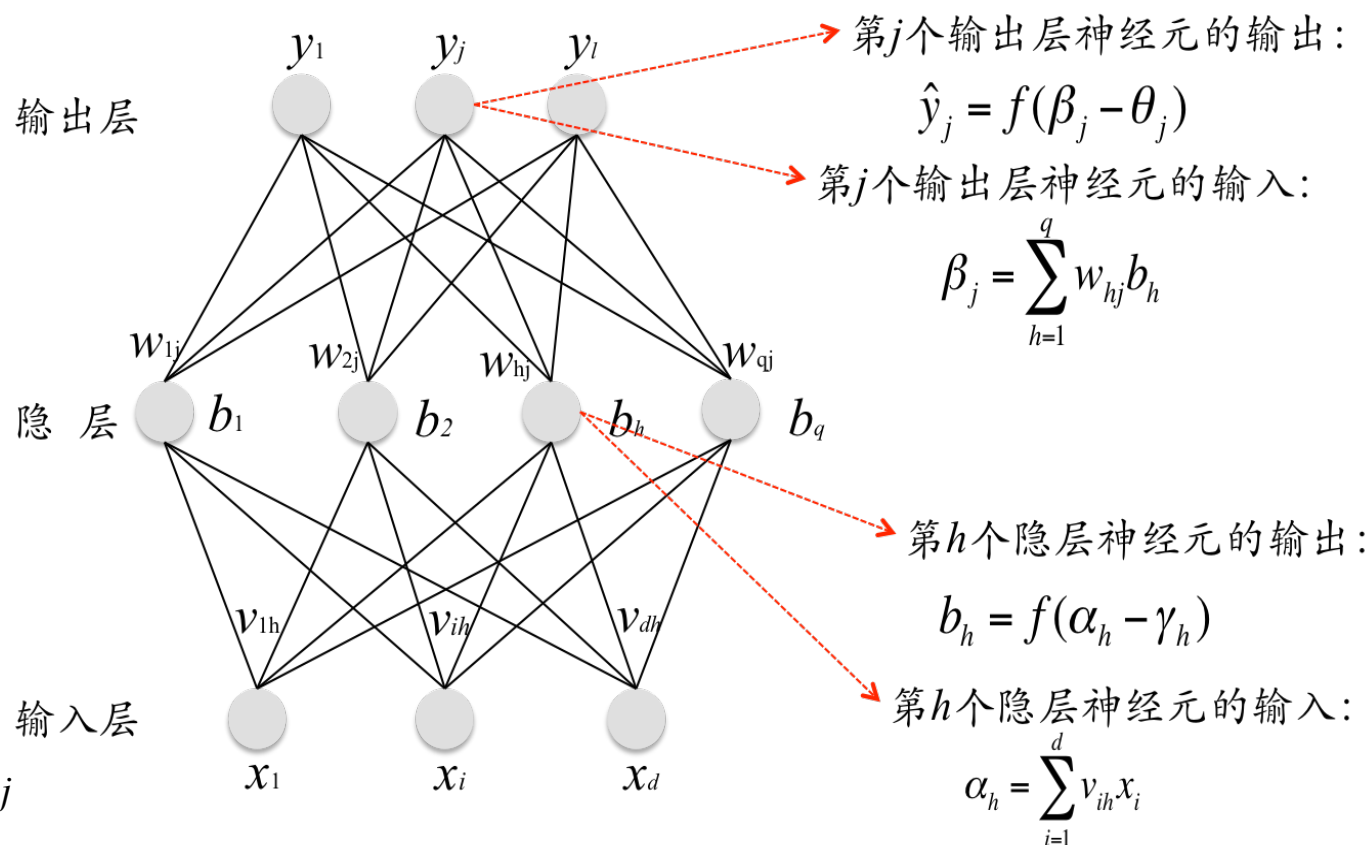
$$f(x) = \text{sigmoid}(x) = \frac{1}{1 + e^{-x}}$$

$$f'(x) = f(x)(1 - f(x))$$

$$\hat{y}_j = f(\beta_j - \theta_j)$$

$$E = \frac{1}{2} \sum_{j=1}^l (\hat{y}_j - y_j)^2 \rightarrow \frac{\partial E}{\partial \hat{y}_j} = \hat{y}_j - y_j$$

$$\Delta w_{hj} = -\eta \frac{\partial E}{\partial w_{hj}} \quad \frac{\partial E}{\partial w_{hj}} = \frac{\partial E}{\partial \hat{y}_j} \cdot \frac{\partial \hat{y}_j}{\partial \beta_j} \cdot \frac{\partial \beta_j}{\partial w_{hj}}$$



神经网络

BP网络结构

$$\frac{\partial E}{\partial w_{hj}} = \frac{\partial E}{\partial \hat{y}_j} \cdot \frac{\partial \hat{y}_j}{\partial \beta_j} \cdot \frac{\partial \beta_j}{\partial w_{hj}}$$

$$\frac{\partial \beta_j}{\partial w_{hj}} = b_h \quad \frac{\partial E}{\partial \hat{y}_j} = \hat{y}_j - y_j$$

$$\frac{\partial \hat{y}_j}{\partial \beta_j} = f'(\beta_j - \theta_j) \quad f'(x) = f(x)(1 - f(x))$$

$$= f(\beta_j - \theta_j)(1 - f(\beta_j - \theta_j))$$

$$= \hat{y}_j(1 - \hat{y}_j)$$

$$g_j = -\frac{\partial E}{\partial \hat{y}_j} \cdot \frac{\partial \hat{y}_j}{\partial \beta_j}$$

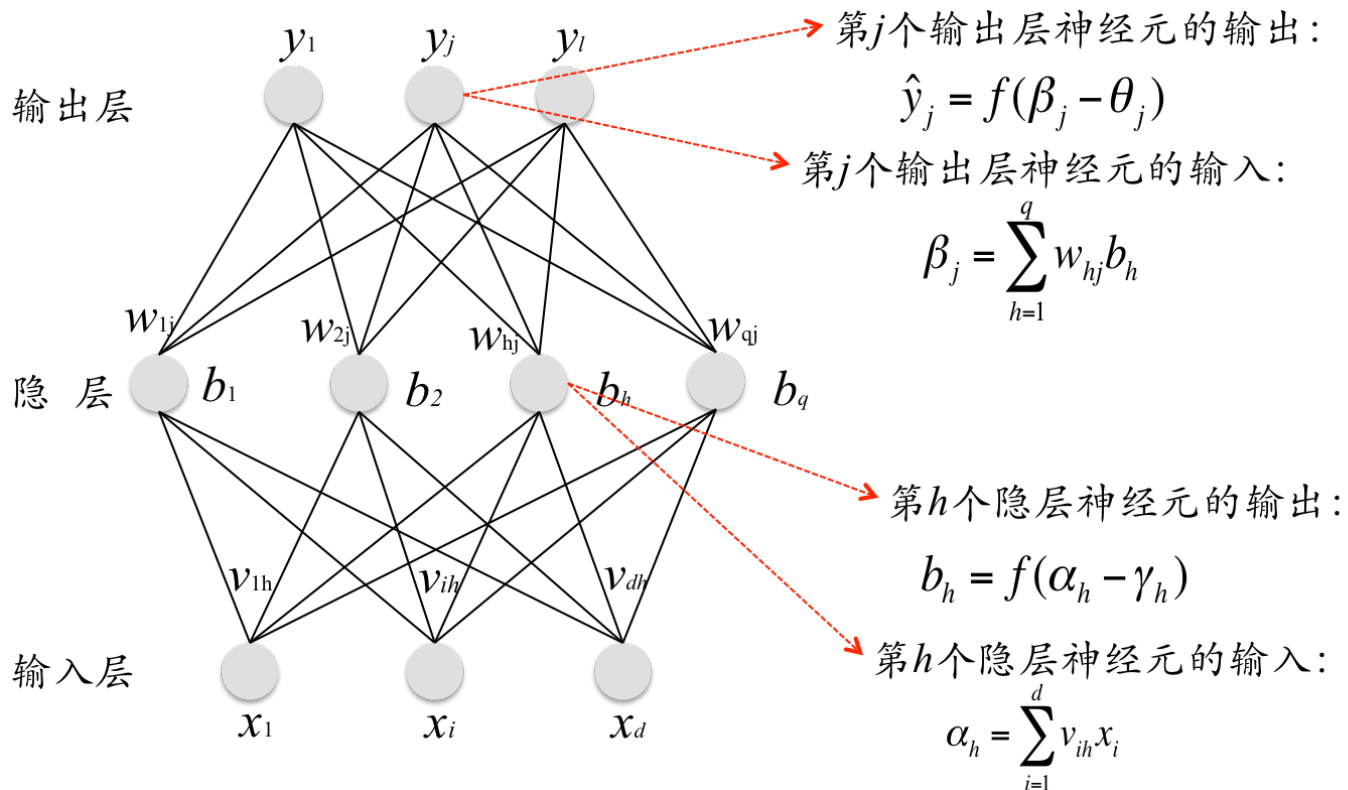
$$= -(\hat{y}_j - y_j)\hat{y}_j(1 - \hat{y}_j)$$

$$= \hat{y}_j(1 - \hat{y}_j)(y_j - \hat{y}_j)$$

$$\Delta w_{hj} = -\eta \frac{\partial E}{\partial w_{hj}}$$

$$= -\eta \frac{\partial E}{\partial \hat{y}_j} \cdot \frac{\partial \hat{y}_j}{\partial \beta_j} \cdot \frac{\partial \beta_j}{\partial w_{hj}}$$

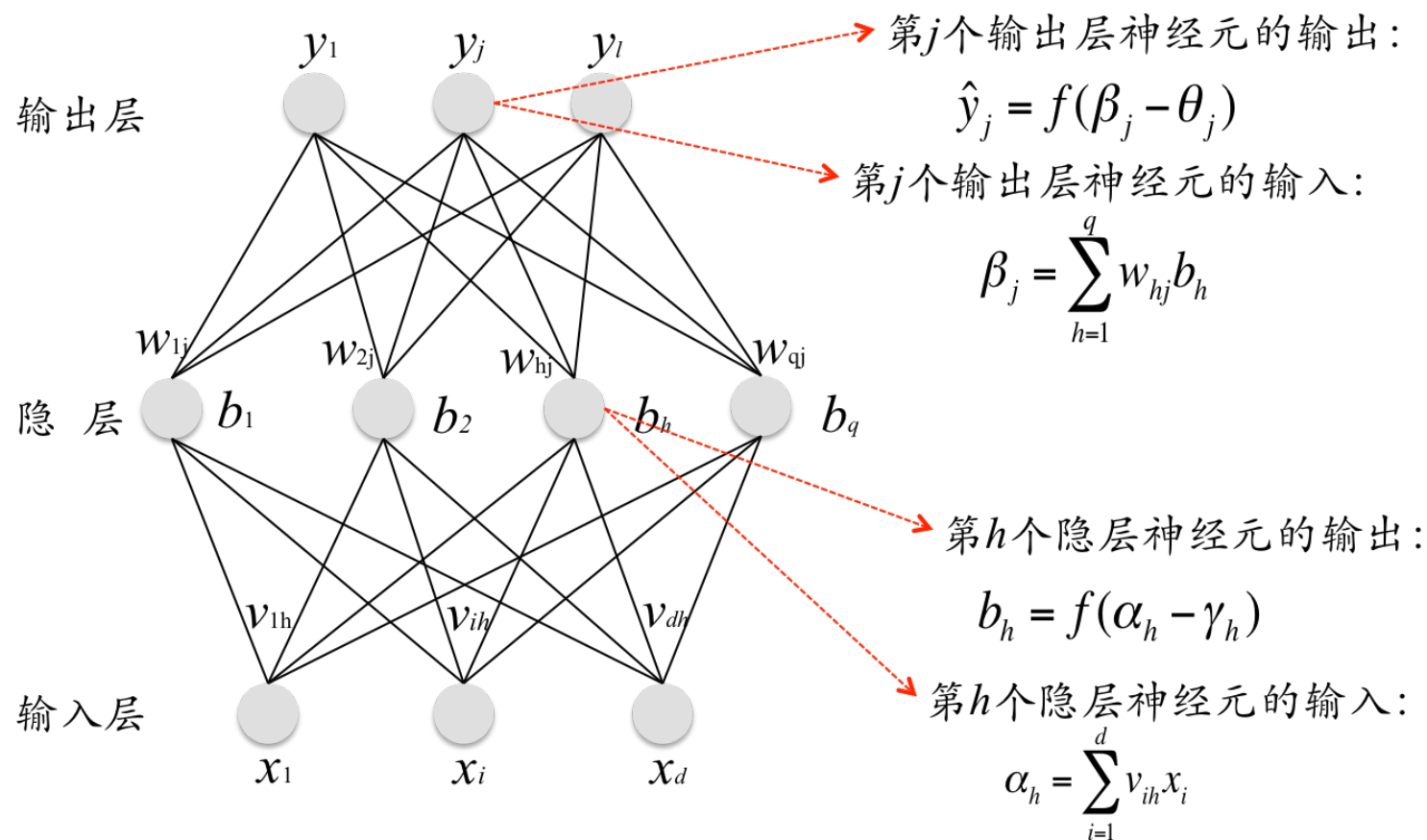
$$= \eta g_j b_h$$



神经网络

BP网络结构

$$\begin{aligned}\Delta w_{hj} &= -\eta \frac{\partial E}{\partial w_{hj}} \\ &= -\eta \frac{\partial E}{\partial \hat{y}_j} \cdot \frac{\partial \hat{y}_j}{\partial \beta_j} \cdot \frac{\partial \beta_j}{\partial w_{hj}} \\ &= \eta g_j b_h \\ &= \eta \hat{y}_j (1 - \hat{y}_j) (y_j - \hat{y}_j) b_h \\ \Delta \theta_j &= -\eta g_j \\ &= -\eta \hat{y}_j (1 - \hat{y}_j) (y_j - \hat{y}_j)\end{aligned}$$

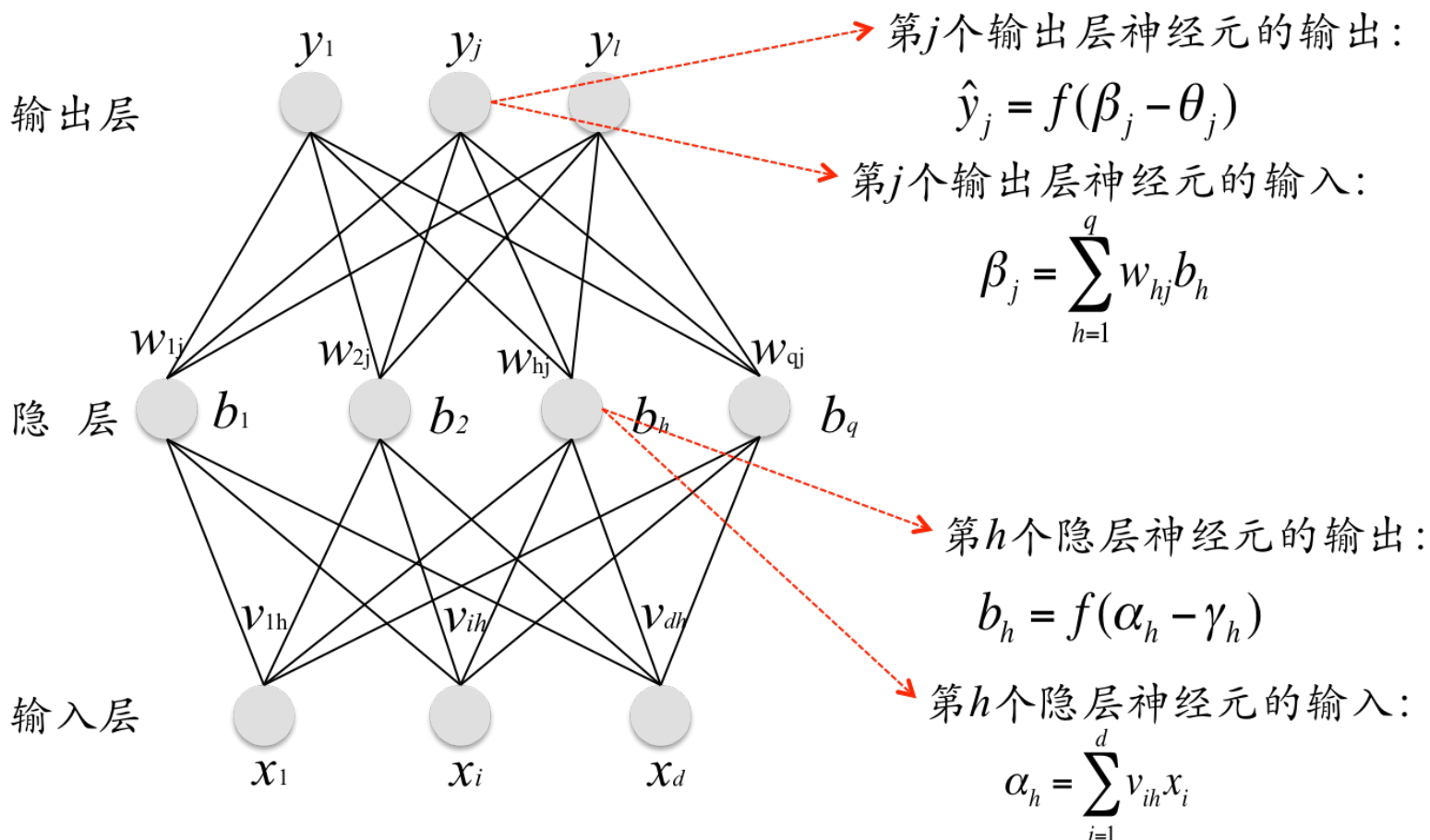


神经网络

BP网络结构

$$\begin{aligned}\Delta v_{ih} &= \eta e_h x_i \\ &= -\eta \frac{\partial E}{\partial b_h} \cdot \frac{\partial b_h}{\partial \alpha_h} x_i \\ &= \eta b_h (1 - b_h) \sum_{j=1}^l w_{hj} g_j x_i\end{aligned}$$

$$\begin{aligned}\Delta \gamma_h &= -\eta e_h \\ &= \eta \frac{\partial E}{\partial b_h} \cdot \frac{\partial b_h}{\partial \alpha_h} \\ &= -\eta b_h (1 - b_h) \sum_{j=1}^l w_{hj} g_j\end{aligned}$$



神经网络

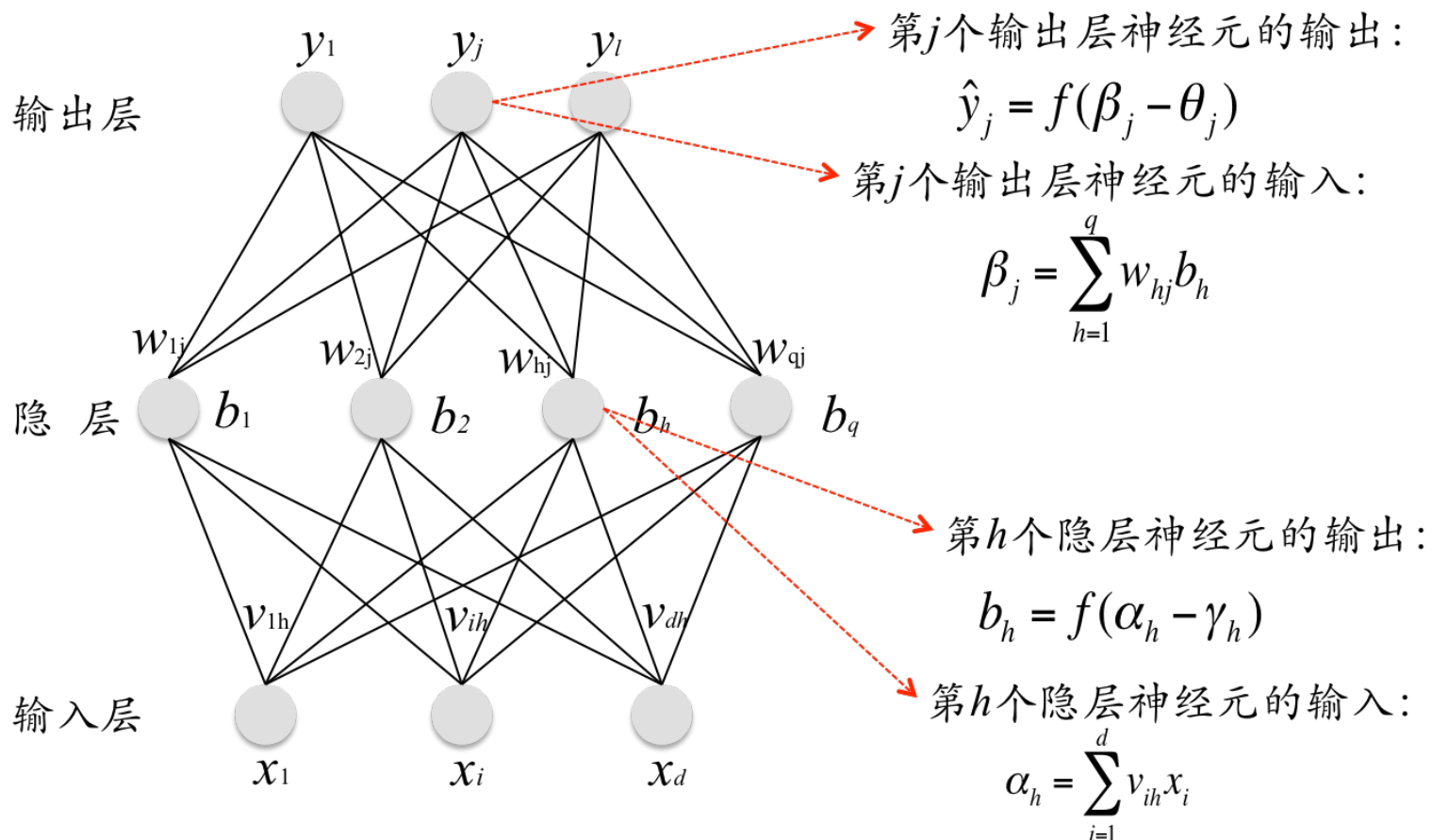
BP网络结构

$$\Delta w_{hj} = \eta \hat{y}_j (1 - \hat{y}_j) (y_j - \hat{y}_j) b_h$$

$$\Delta \theta_j = -\eta \hat{y}_j (1 - \hat{y}_j) (y_j - \hat{y}_j)$$

$$\Delta v_{ih} = \eta b_h (1 - b_h) \sum_{j=1}^l w_{hj} g_j x_i$$

$$\Delta \gamma_h = -\eta b_h (1 - b_h) \sum_{j=1}^l w_{hj} g_j$$



神经网络

网络训练过程

输入：训练集数据、学习速率 η

过程：

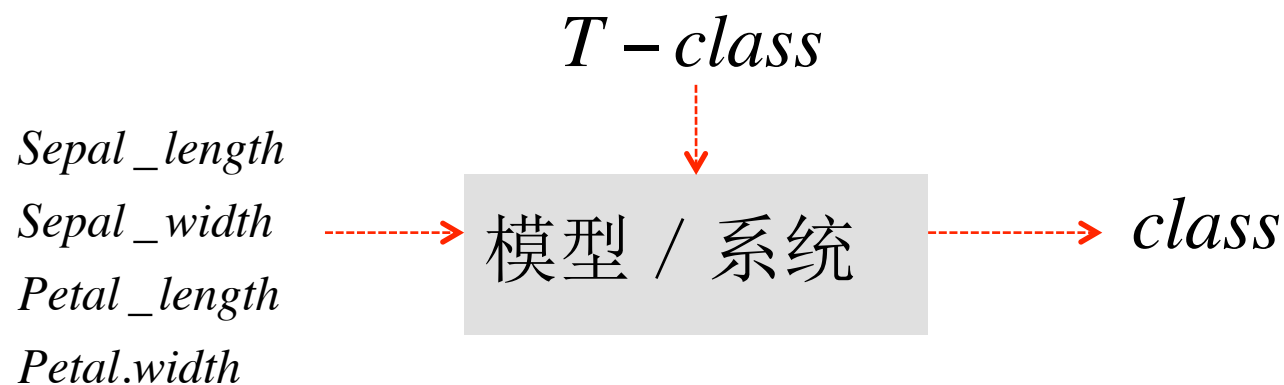
- 在(0,1)范围内随机初始化网络中所有连接权和阈值
- repeat
 - 根据网络输入和当前参数计算网络输出值 y
 - 计算输出层神经元梯度项 g_j
 - 计算隐层神经元梯度项 e_h
 - 更新连接权值和阈值
- until达到停止条件
- 输出：连接权值和阈值



神经网络

- 分类：学习 / 训练过程有监督，训练样本有明确标签

Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	T-class
5.1	3.5	1.4	0.2	setosa
4.9	3	1.4	0.2	setosa
7	3.2	4.7	1.4	versicolor
6.4	3.2	4.5	1.5	versicolor
6.3	3.3	6	2.5	virginica
5.8	2.7	5.1	1.9	virginica
6.5	3	5.8	2.2	?
6.2	2.9	4.3	1.3	?



神经网络

代码实现

Python (sklearn)

- `Net = MLPClassifier(hidden_layer_sizes=10,max_iter=1000).fit(tr_data.ix[:,0:6],tr_data.ix[:,6])`
- `res = Net.predict(te_data.ix[:,0:6])`

R (nnet)

- `nnet(x, y, size, softmax = FALSE, maxit = 100)`



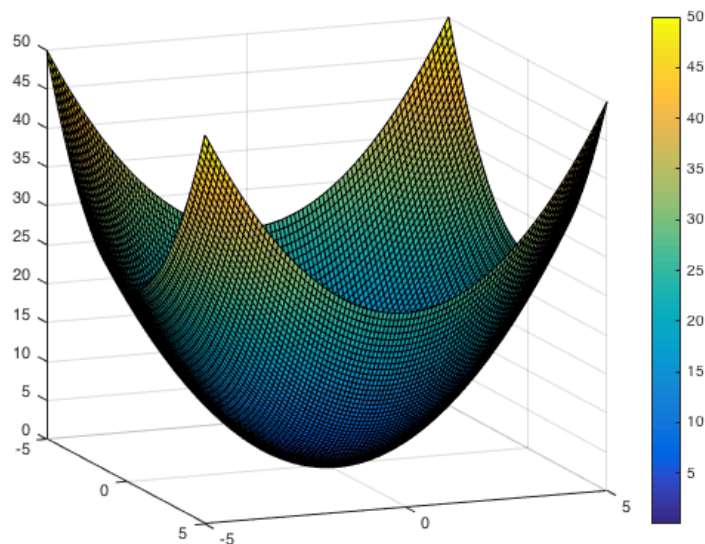
神经网络

附录：BP神经网络自编代码

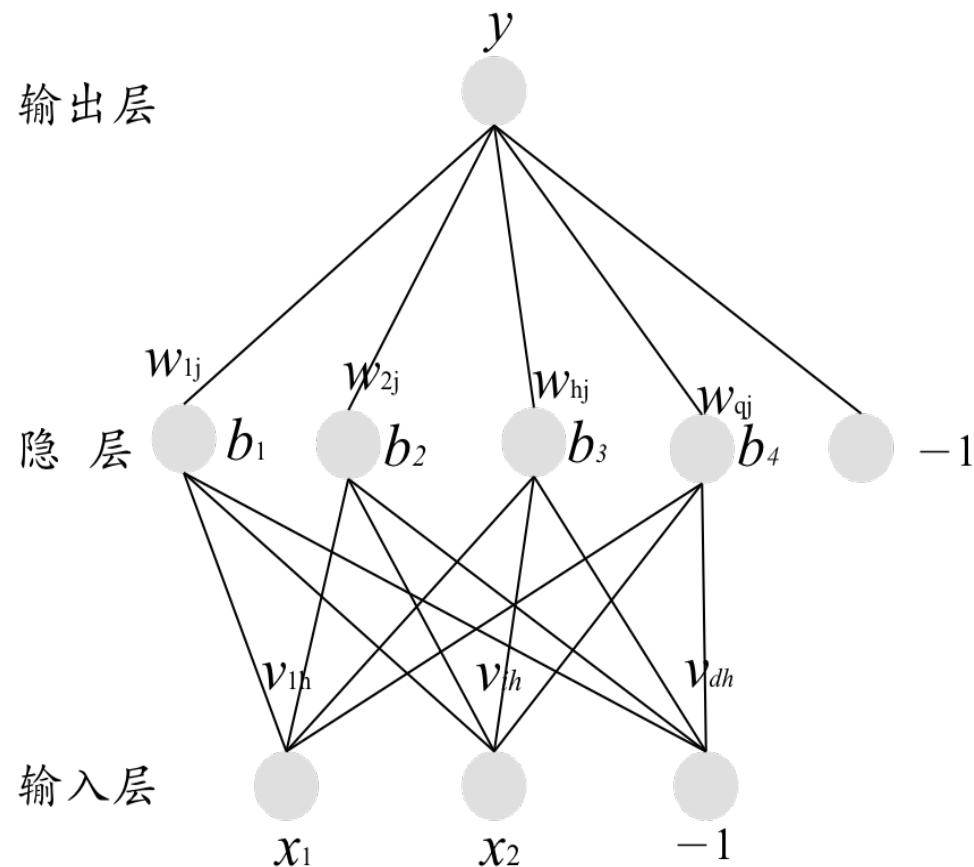
$$y = x_1^2 + x_2^2$$

训练集数据：BPdata_tr.txt

测试集数据：BPdata_te.txt

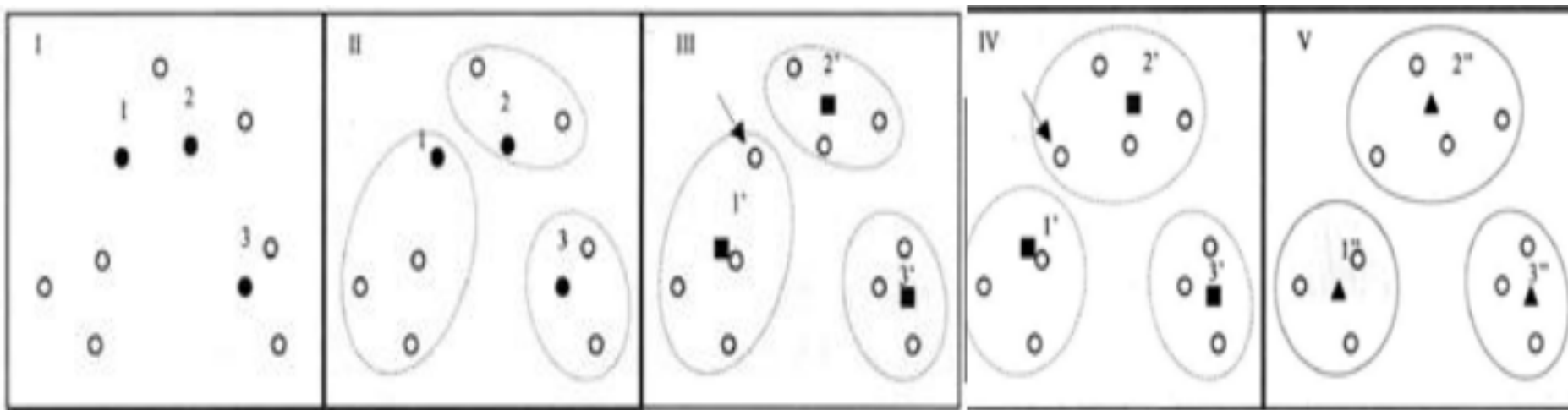


	x_1	x_2	y
0	0.29	0.23	0.14
1	0.50	0.62	0.64
2	0.00	0.53	0.28
3	0.21	0.53	0.33
4	0.10	0.33	0.12
5	0.06	0.15	0.03
6	0.13	0.03	0.02
7	0.24	0.23	0.11
8	0.28	0.03	0.08
9	0.38	0.49	?
10	0.29	0.47	?



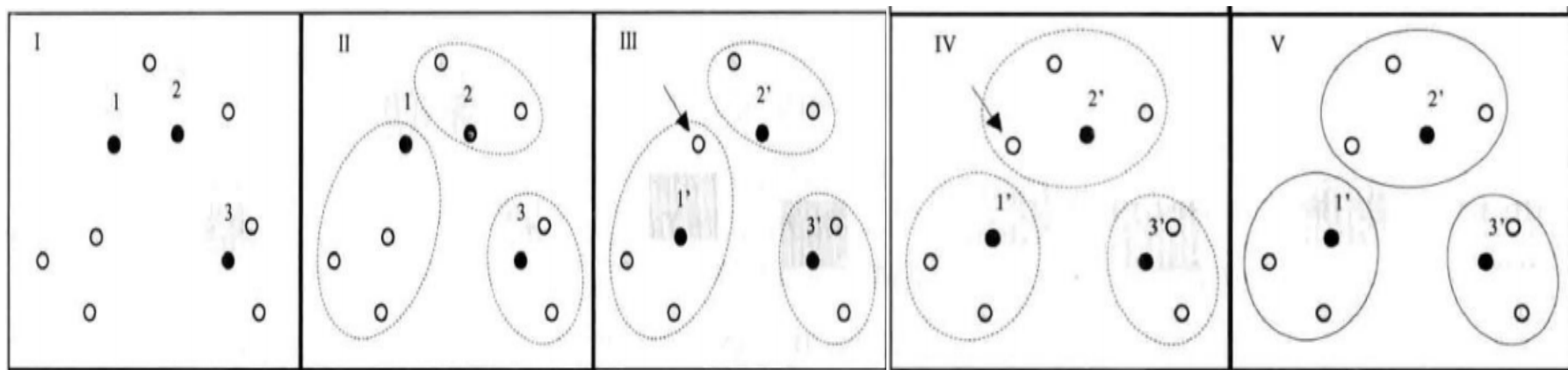
K-Means算法步骤

- 1、随机选取K个样本作为类中心；
- 2、计算各样本与各类中心的距离；
- 3、将各样本归于最近的类中心点；
- 4、求各类的样本的均值，作为新的类中心；
- 5、判定：若类中心不再发生变动或达到迭代次数，算法结束，否则回到第2步。



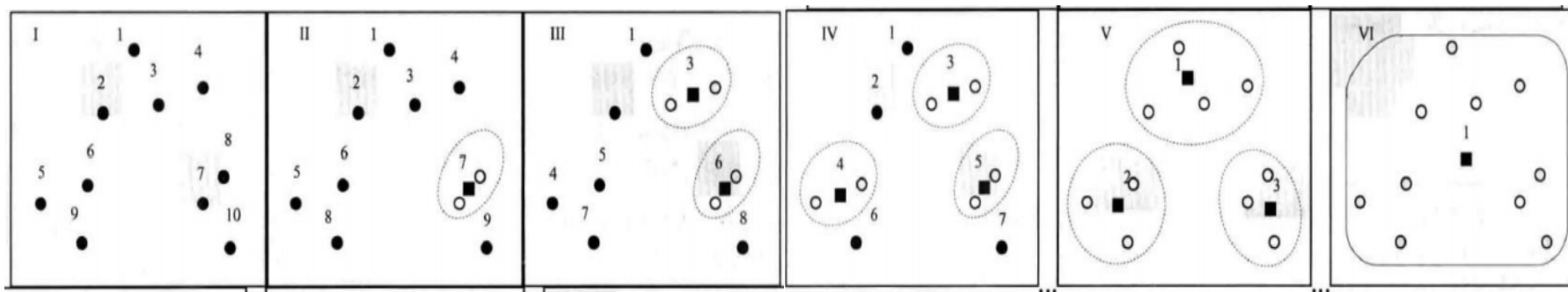
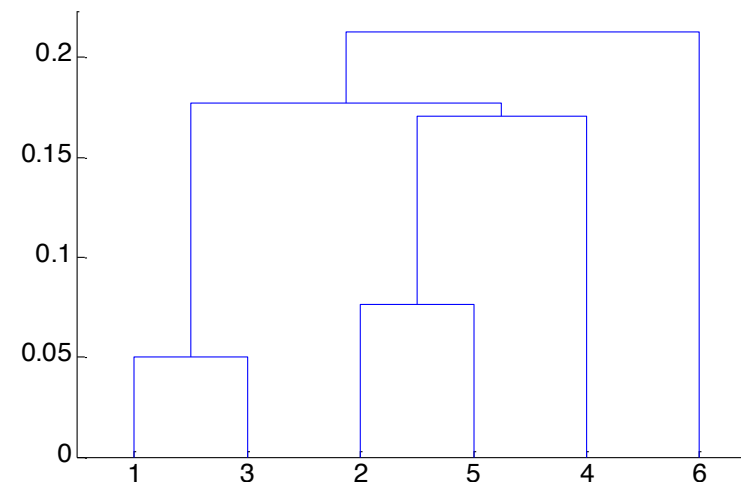
K-Medoids算法步骤

- 1、随机选取K个样本作为类中心；
- 2、计算各样本与各类中心的距离；
- 3、将各样本归于最近的类中心点；
- 4、在各类别内选取到其余样本距离之和最小的样本作为新的类中心；
- 5、判定：若类中心不再发生变动或达到迭代次数，算法结束，否则回到第2步。



层次聚类(系谱聚类 Hierarchical Clustering, HC)

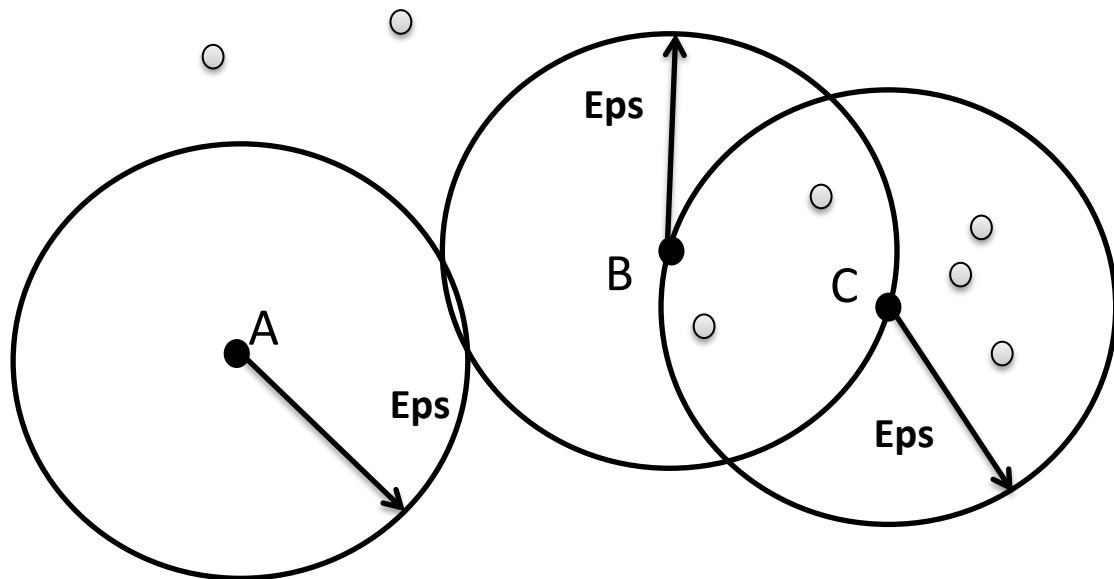
- 不需事先设定类别数 k
- 每次迭代过程仅将距离最近的两个样本/簇聚为一类
- 得到 $k=n$ 至 $k=1$ (n 为待分类样本总数)个类别的聚类结果



DBSCAN (密度聚类)

样本 (对象点) 的区域半径Eps和区域内点的个数的阈值MinPts

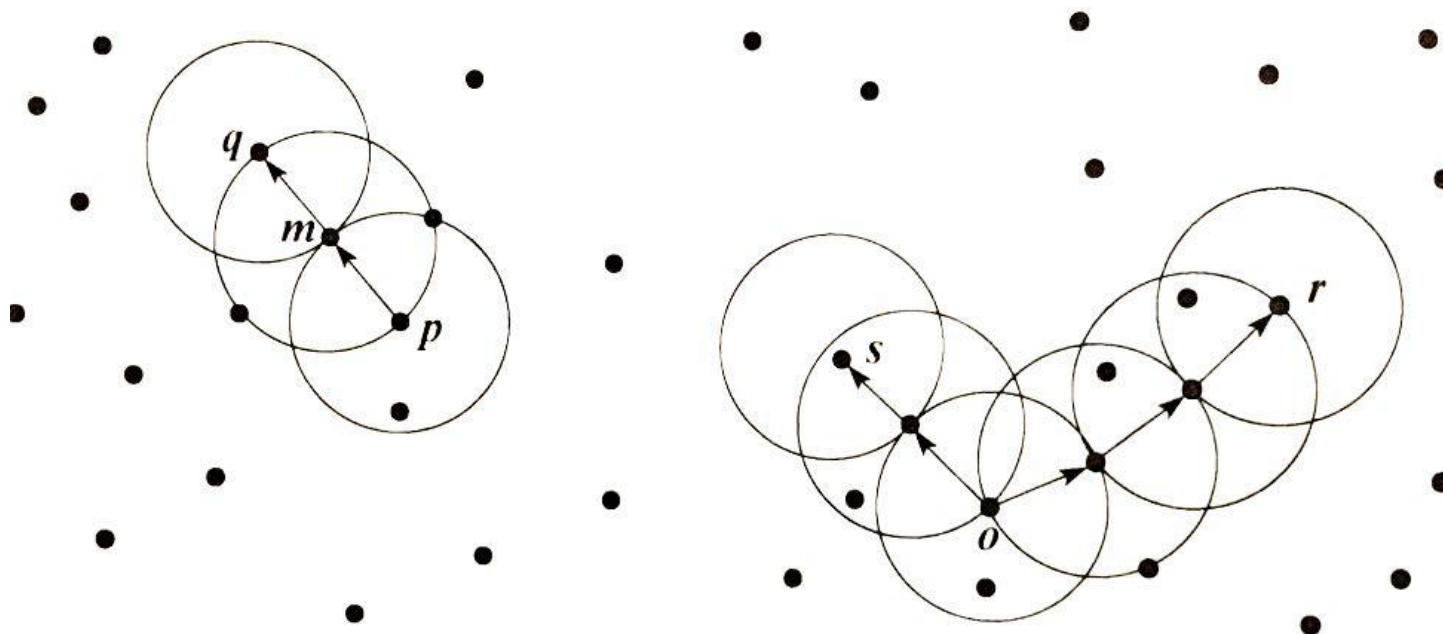
- 核心点：如果点p的密度等于或者大于阈值MinPts，则P为核心点。
- 边界点：如果点p不是核心点，但落在其他核心点的区域内，那么p点为边界点。
- 噪声点：如果点p既不是核心点，也不是边界点，则p点为噪声点。



A为噪声点
B为边界点
C为核心点

DBSCAN (密度聚类)

DBSCAN算法原理直观图



- 1.密度直达：点 q 距核心点 m 距离小于等于 ϵ ，从 m 到 q 密度直达。不对称
- 2.密度可达：若从 p 到 m 密度直达，从 m 到 q 密度直达，则从 p 到 q 密度可达。不对称
- 3.密度相连：若从 o 到 s 密度可达，且从 o 到 r 密度可达的，所有 o, r 和 s 都是密度相连的。对称

DBSCAN聚类步骤

定义半径和MinPts

从对象集合D中抽取未被访问过的样本点 q

检验该样本点是否为核心对象，如果是则进入下一步，否则返回上一步

找出该样本点所有从该点密度可达的对象，构成聚类 $C \downarrow q$

如果全部样本点都已被访问，则结束算法，否则返回第2步骤



EM步骤与流程

最大期望算法经过两个步骤交替进行计算：

- 第一步是计算期望（E），利用概率模型参数的现有估计值，计算隐藏变量的期望；
- 第二步是最大化（M），利用E步上求得的隐藏变量的期望，对参数模型进行最大似然估计。
- M步上找到的参数估计值被用于下一个E步计算中，这个过程不断交替进行。

总体来说，EM的算法流程如下：

- 1.初始化分布参数
- 2.重复直到收敛：
 - E步骤：估计未知参数的期望值，给出当前的参数估计。
 - M步骤：重新估计分布参数，以使得数据的似然性最大，给出未知变量的期望估计。



双硬币问题

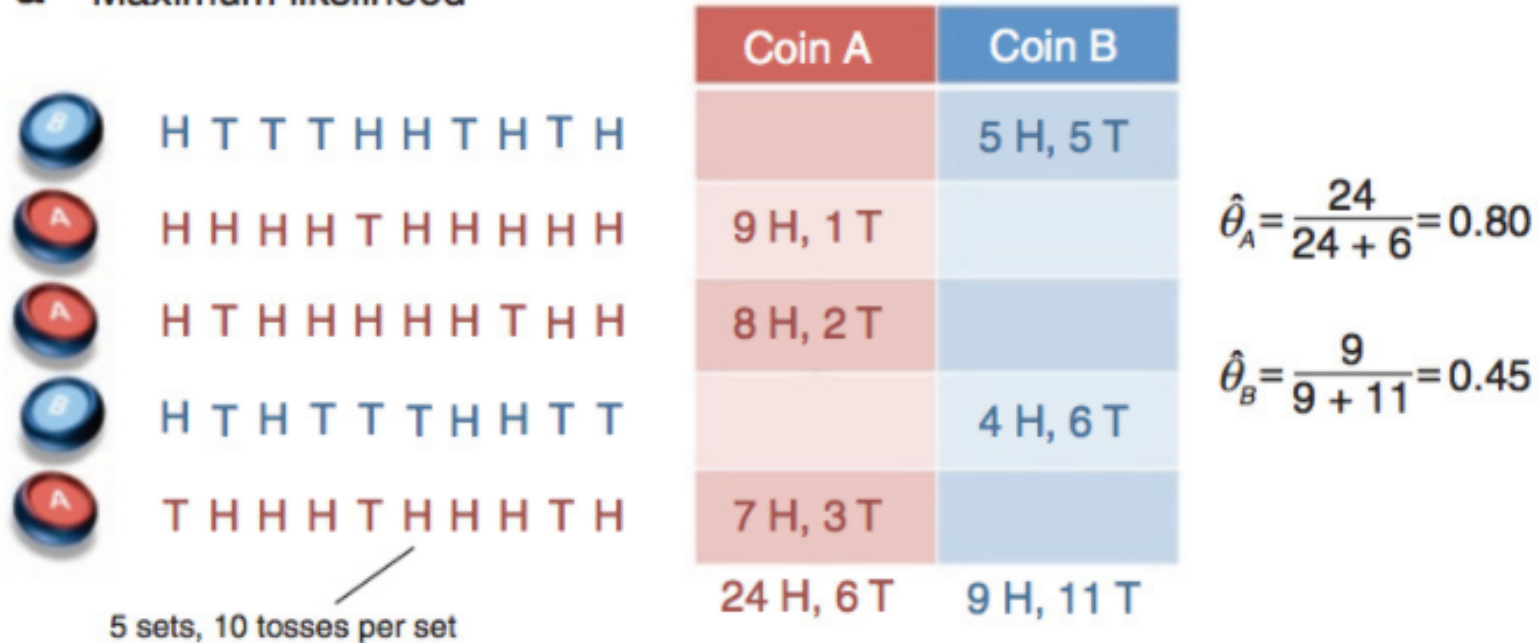
- 假设有两枚硬币A、B，以相同的概率随机选择一个硬币，进行如下的抛硬币实验：共做5次实验，每次实验独立的抛十次，结果如图中a所示，例如某次实验产生了H、T、T、T、H、H、T、H、T、H，H代表正面朝上。
- 假设试验数据记录员可能是实习生，业务不一定熟悉，造成a和b两种情况
- a表示实习生记录了详细的试验数据，我们可以观测到试验数据中每次选择的是A还是B
- b表示实习生忘了记录每次试验选择的是A还是B，我们无法观测实验数据中选择的硬币是哪一个
- 问在两种情况下分别如何估计两个硬币正面出现的概率？



双硬币问题

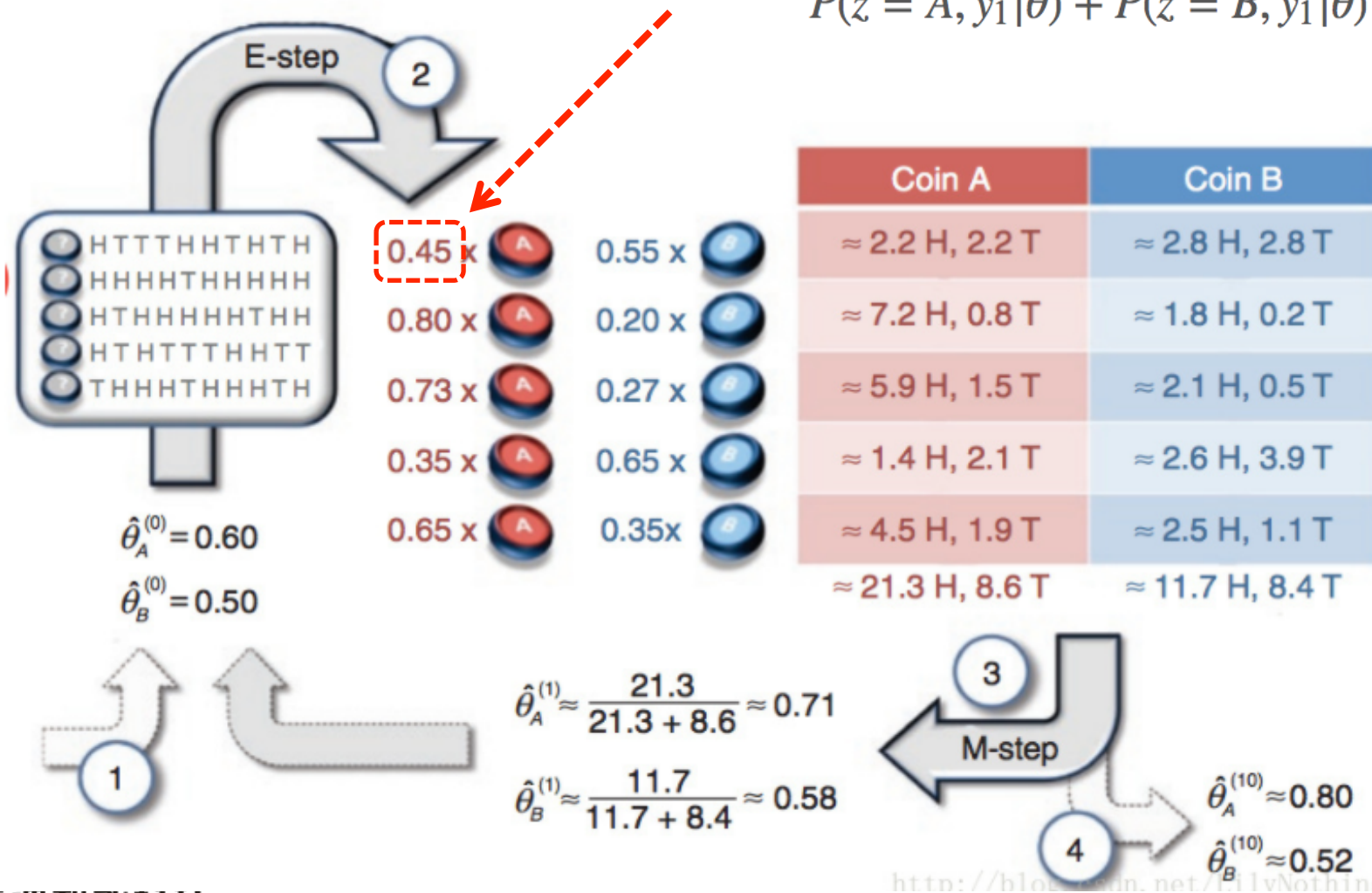
- 已知是A硬币还是B硬币抛出的结果的时候，可以直接采用概率的求法来进行求解
- 对于含有隐变量的情况，也就是不知道到底是A还是B就需要采用EM算法进行求解了

a Maximum likelihood



双硬币问题

b Expectation maximization $P(z = A|y_1, \theta) = \frac{P(z = A, y_1|\theta)}{P(z = A, y_1|\theta) + P(z = B, y_1|\theta)} = \frac{(0.6)^5 * (0.4)^5}{(0.6)^5 * (0.4)^5 + (0.5)^{10}} = 0.45$





大数据成就未来



Thank you!

泰迪科技 : www.tipdm.com
热线电话 : 40068-40020

