人工智能

目录

CONTENTS

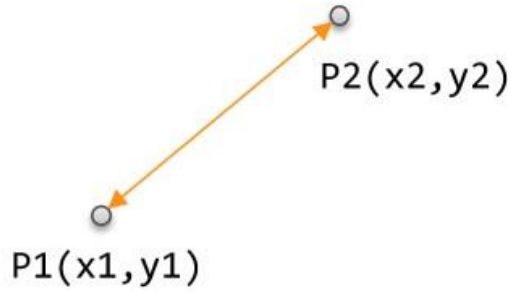
3.1 聚类分析

- 聚类：根据数据的“相似性”将数据归纳为多类的过程
- 良好的聚类效果需满足：
  - 同一类中，样本之间保证高相似性
  - 类与类之间，样本之间要高差异性或不相似
- 相似性衡量标准的选择，对于聚类(clustering)十分重要
- 如何评估样本之间相似性？相似性的衡量标准？

（1）相似性

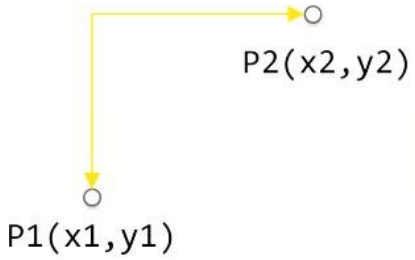
（2）相似性衡量方法

（1）欧氏距离



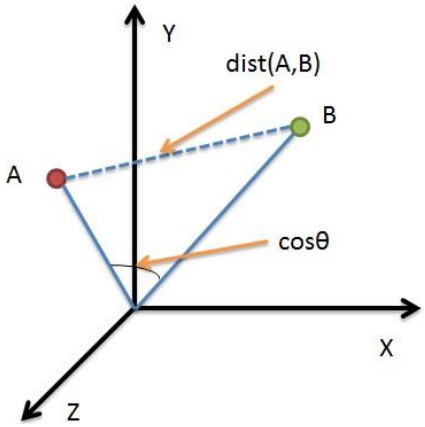
$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

（2）曼哈顿距离



$$d = |x_1 - x_2| + |y_1 - y_2|$$

（3）余弦相似度



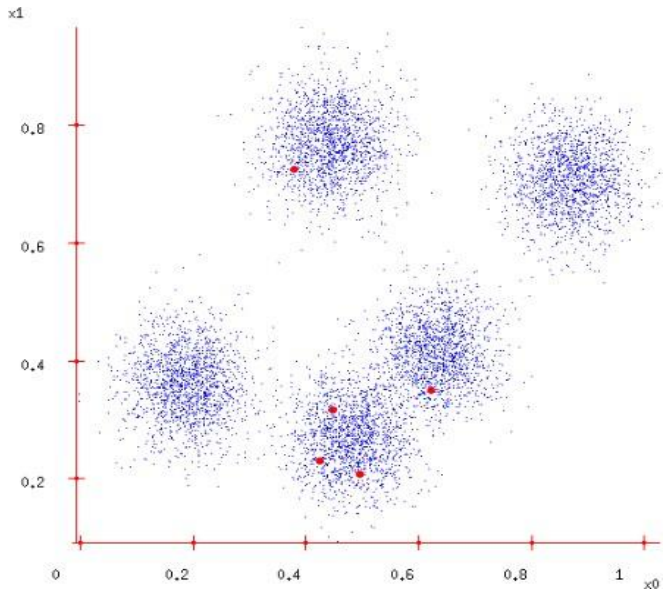
（3）典型聚类算法

- K-means：建立数据的不同分割，并用欧氏距离等评价聚类结果
- GMM：对于每个类假定一个分布模型，试图找到每个类最好的模型
- Aprior：从数据背后发现事物之间可能存在的关联或者联系

3.2 K-means

- k-means算法也就是k均值算法
- k-means算法以k为参数，把n个对象分成k个簇（类）
- 处理过程1：
  - 选择k个点作为初始的聚类中心；

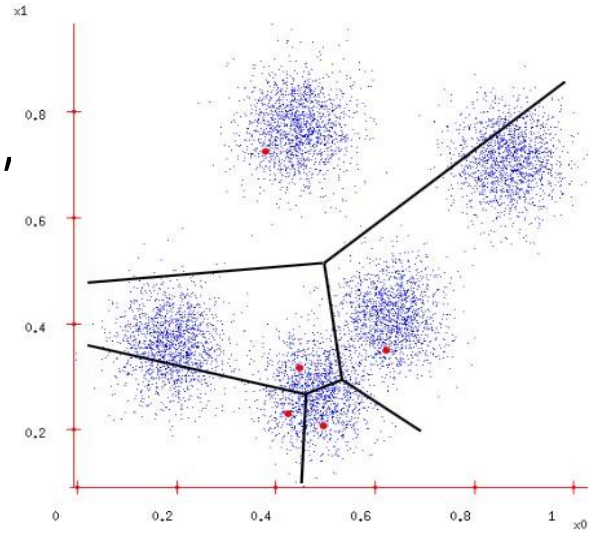
$$\mu^{(0)} = \mu_1^{(0)}, \ldots, \mu_k^{(0)}$$



3.2 K-means

- k-means算法也就是k均值算法
- k-means算法以k为参数，把n个对象分成k个簇（类）
- 处理过程2：
  - 剩下的点，根据其与聚类中心的欧式距离，将其归入最近的簇

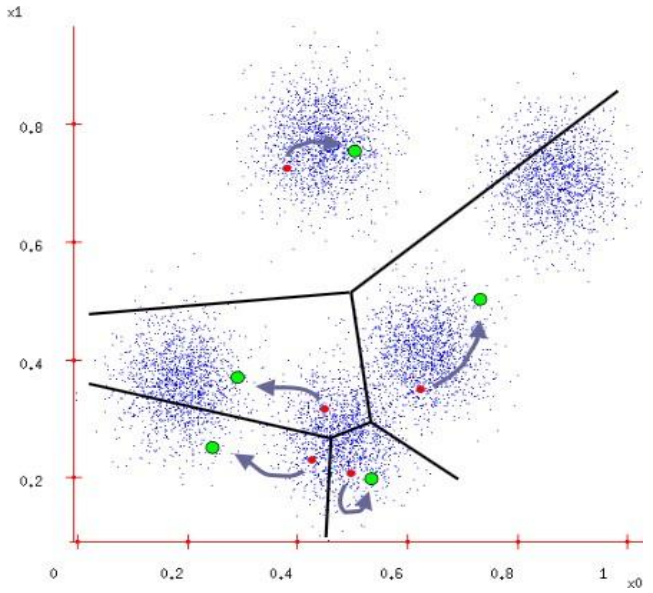
$$C^{(t)}(j) \leftarrow \arg\min_i ||\mu_i - x_j||^2$$



3.2 K-means

- k-means算法也就是k均值算法
- k-means算法以k为参数，把n个对象分成k个簇（类）
- 处理过程3：
  - 对每个簇，计算所有点的均值作为新的聚类中心

$$\mu_i^{(t+1)} \leftarrow \arg\min_{\mu} \sum_{j:C(j)=i} ||\mu - x_j||^2$$

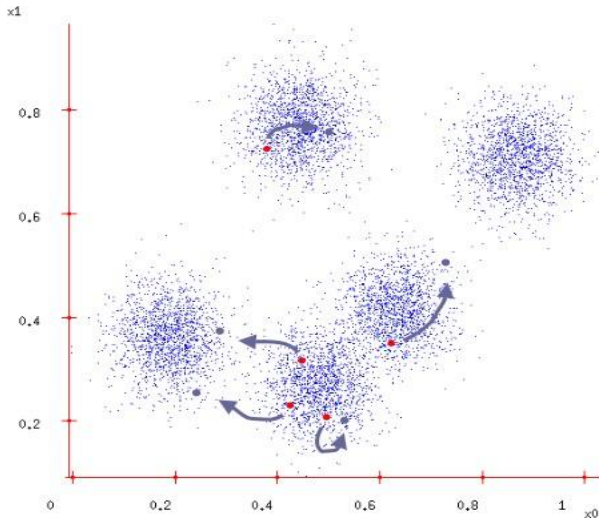


3.2 K-means

- k-means算法也就是k均值算法
- k-means算法以k为参数，把n个对象分成k个簇（类）
- 处理过程4：
  - 重复(2)，(3)步骤，
    直到聚类中心不再发生改变

$$F(\mu, C) = \sum_{j=1}^{m} ||\mu_{C(j)} - x_j||^2$$

$\min_{\mu}\min_{C} F(\mu, C)$



3.2 K-means---关键问题导读

(1)K值怎么确定?

- 解决方案：根据实际的业务需求，人工来指定。

(2)关于初始质心的选择,会对分类结果产生很大影响，可能偏离全局最优解或者增加计算量。

- 解决方案：随机多次选择不同的初始聚类中心，反复多次进行实验。

(3)如何判断算法是否该停止?

- 解决方法：随机选择质心，迭代计算每个数据到新质心的距离，直到新质心和原质心相等，算法结束。

3.2 K-means---实例

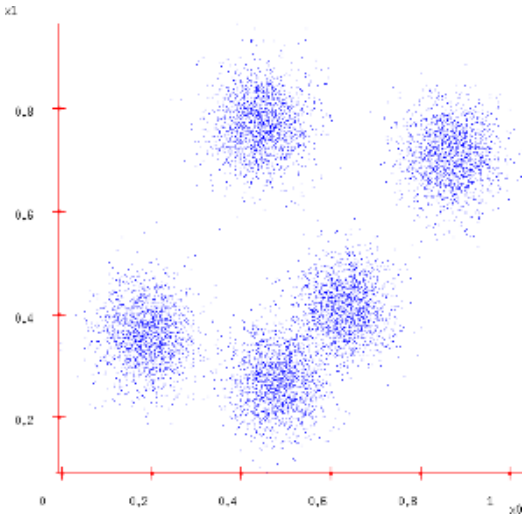
- *Kmeans_user_age. clustering*

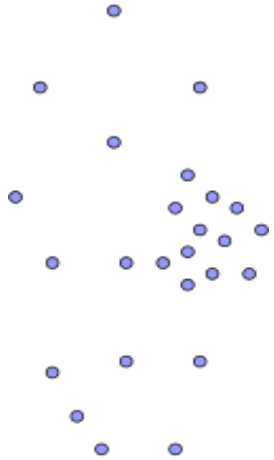
3.2 K-means---局限性

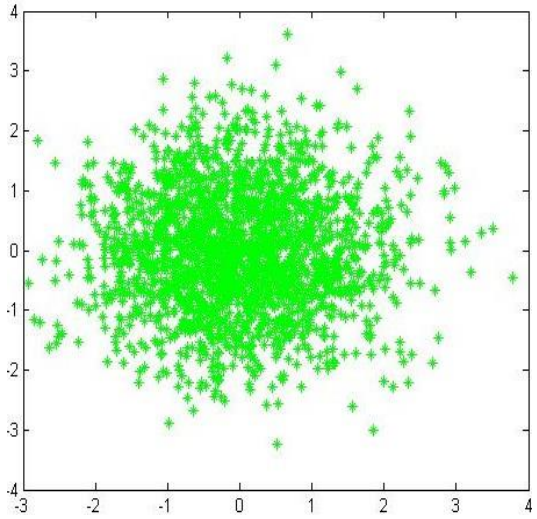
- 属于“硬聚类”，每个样本只能属于一个类别。
- K-means对异常点的“免疫力”差，异常值对其聚类中心影响比较大（改进：中心不直接取均值，而是找均值最近的样本点代替 -- k-medoids算法）。
- 对于团状的数据点集区分度好，对于带状(环绕)等“非凸”形状不太好。

3.3 GMM（高斯混合模型）

GMM的产生解决了K-means的局限性







3.3 GMM（高斯混合模型）

1 GMM是如何解决上述问题：

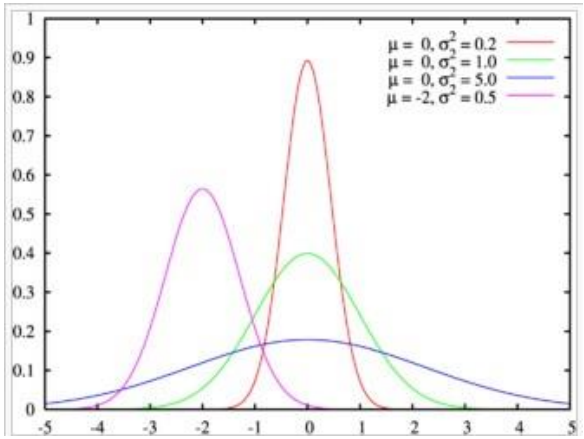
- 求解每个测试数据属于某个类别的概率（软指标）

2 GSM（高斯模型）

- 给定均值和方差，将一个事物分解为基于高斯概率密度函数（正态分布曲线）形成的模型，表示随机变量每个取值有多大的可能性

$$P(x) = \varphi(x;\mu,\sigma) = \frac{1}{\sigma\sqrt{2\pi}} \exp(-\frac{(x-\mu)^2}{2\sigma^2})$$

$\theta=(\mu,\sigma^2)$



3.3 GMM（高斯混合模型）

3　　GMM（高斯混合模型）

- K个GSM混合成一个GMM，每个GSM称为GMM的一个component，也就是分为K个类。

$$P(y \mid \theta) = \sum_{k=1}^{K} \alpha_k \phi(y \mid \theta_k)$$

  - 求和式的各项的结果就分别代表样本y属于各个类的概率
  - ak：样本y属于第k个类的概率

3.3 GMM（高斯混合模型）

- 属于假设有K个类，样本数量分别为$N_1,N_2,\ldots,N_k$且$N_1+N_2+\ldots+N_k=N$,即有观测数据$y_1,y_2,\ldots,y_k$ ，第k个分类的样本集合表示为S(k)，上式中的三个参数可表示为：

$$\gamma_{jk} = P(z=k \mid y_j,\theta) = \frac{P(z=k,y_j \mid \theta)}{\sum_{k=1}^{K} P(z=k,y_j \mid \theta)} = \frac{P(y_j \mid z=k,\theta)P(z=k \mid \theta)}{\sum_{k=1}^{K} P(y_j \mid z=k,\theta)P(z=k \mid \theta)} = \frac{\alpha_k \phi(y_j \mid \theta_k)}{\sum_{k=1}^{K} \alpha_k \phi(y_j \mid \theta_k)}$$

$$\alpha_k = N_k/N \qquad \alpha_k = \frac{\sum_{j=1}^{N} \gamma_{jk}}{N} \qquad (10)$$

$$\mu_k = \frac{1}{N_k}\sum_{y \in S(k)} y \qquad \mu_k = \frac{\sum_{j=1}^{N} \gamma_{jk} y_j}{\sum_{j=1}^{N} \gamma_{jk}} \qquad (11)$$

$$\sigma_k^2 = \frac{1}{N_k}\sum_{y \in S(k)} (y-\mu_k)^2 \qquad \sigma_k^2 = \frac{\sum_{j=1}^{N} \gamma_{jk}(y_j-\mu_k)^2}{\sum_{j=1}^{N} \gamma_{jk}} \qquad (12)$$

ak指的是第**k**个**component**被选中的概率，**rjk**需要对所有的数据 **j** 进行累加

3.3 GMM（高斯混合模型）

选取初始值θ^0初始化θ，

repeat{

(1)估计每个数据的每个component生成概率，即γ_{jk}：

$$\gamma_{jk} = \frac{\alpha_k \phi(y_j \mid \theta_k)}{\sum_{k=1}^{K} \alpha_k \phi(y_j \mid \theta_k)}$$

(2)根据γ_{jk}，估计每个component的参数，得：

公式(10),(11),(12)

}直到收敛

3.3 GMM—GMM与K-means

4 GMM与K-means相同点

- 需要指定K值
- 需要指定初始值，K-means的中心点，GMM的参数
- 都是含有EM算法思想

5 GMM与K-means不同点

- 优化目标函数不同，K-means：最短距离(硬指标)；GMM：最大化log似然估计，求解每个观测数据属于每个component的概率(软指标)

3.4 Aprori算法

- 关联分析是一种在大规模数据集中寻找有趣关系的任务
- 这些任务有两种形式：频繁项集和关联规则
  - 频繁项集：经常出现在一块的物品的集合；
  - 关联规则：两种物品之间可能存在很强的关系；
- 关联分析典型方法： Apriori算法

3.4 Aprori算法

1　　使用Apriori算法来发现频繁项集

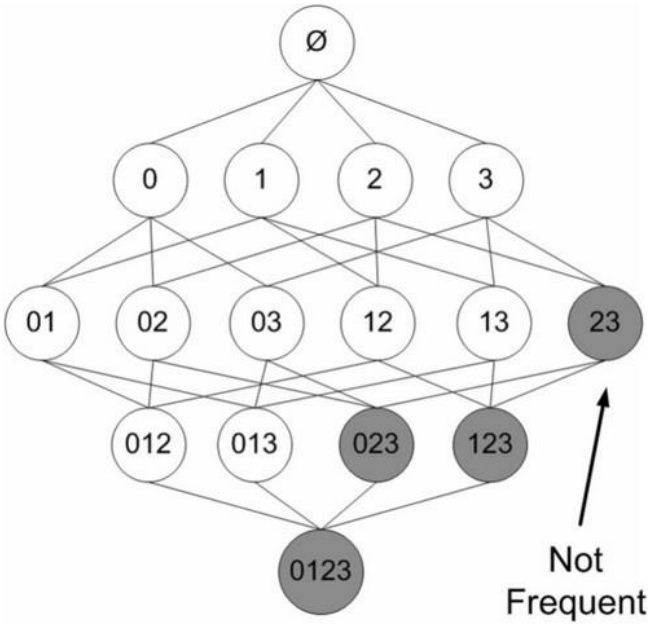
- 两个输入参数分别是最小支持度和数据集，根据最小支持度确实频繁项集。

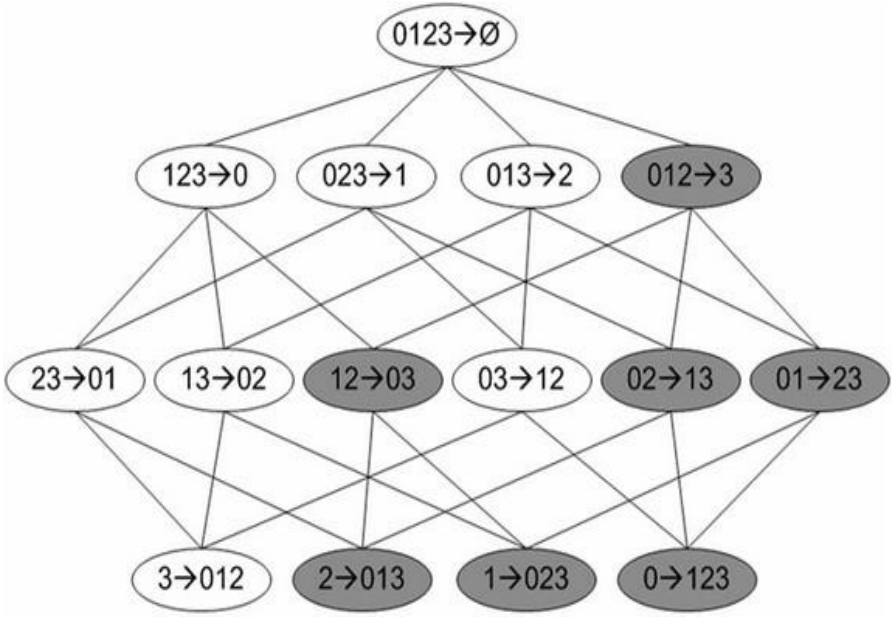
2　　从频繁项集中挖掘关联规则

- 从一个频繁项集开始，创建一个规则列表，首先将规则的右边限定为一个元素，对这些规则进行测试，接下来合并剩下的规则来创建一个新的规则列表，规则的右边限定为两个元素，项集中挖掘关联规则。

3Apriori原理是说如果某个项集是频繁的，那么它的所 有子集也是频繁的。

支持度与可信度





支持度

可信度

3.4 Aprori算法

交易号码	商品
0	豆奶，莴苣
1	莴苣，尿布，啤酒，甜菜
2	豆奶，尿布，啤酒，橙汁
3	莴苣，豆奶，尿布，啤酒
4	莴苣，豆奶，尿布，橙汁

- 频繁项集：例{尿布，啤酒}
  - 支持度：数据集中包含指定项集的记录所占的比例
- 从频繁项集到关联规则
  - 可信度：support(P | H)/support(P)

类别	包括的主要算法
划分方法	K-Means算法、K-MEDOIDS算法、CLARANS算法
层次分析法	BIRCH算法、CURE算法、CHAMELEON算法
基于密度的方法	DBCSCAN算法、DENCLUE算法、OPTICS算法
基于网格的方法	STING算法、CLIOUE算法、WAVE——CLUSTER算法
基于模型的方法	统计学方法、神经网络方法

算法名称	算法描述
K-Means	K-均值聚类也称为快速聚类法，在最小化误差函数的基础上将数据划分为预定的类数K。该算法原理简单并便于处理大量数据
K-中心点	K-均值算法对孤立点的敏感性，K-中心点算法不采用簇中对象的平均值作为簇中心，而选用簇中离平均值最近的对象作为簇中心
系统聚类	系统聚类也称为多层次聚类，分类的单位由高到低呈树形结构，且所处的位置越低，其包含的对象就越少，但这些对象间的共同特征越多。该聚类方法只适用在小数据量的时候使用，数据量大的时候速度会非常慢

目录

CONTENTS

4.1 降维

1 降维的过程

- 降维是指在某些限定条件下，降低随机变量个数，得到一组“不相关”主变量的过程。

2 降维的作用：特征选择和特征提取

- 特征选择：假定数据中包含大量冗余或无关变量（或称特征、属性等），旨在从原有变量中找出主要变量。
- 特征提取：将高维数据转化为低维数据的过程，可能舍弃原数据、构造新变量，其代表方法为主成分分析（PCA）。

4.1 降维

学生编号	语文	数学	物理	化学
1	90	140	99	100
2	90	97	88	92
3	90	110	79	83
. . .	. . .	. . .	. . .	. . .

如果根据成绩判断学习的情况，直观上，哪些科目成绩对判断结果可能没有影响？？

学生编号	数学	物理	化学	语文	历史	英语
1	65	61	72	84	81	79
2	77	77	76	64	70	55
3	67	63	49	65	67	57
4	80	69	75	74	74	63
5	74	70	80	84	82	74
6	78	84	75	62	72	64
7	66	71	67	52	65	57
8	77	71	57	72	86	71
9	83	100	79	41	67	50
. . .	. . .	. . .	. . .	. . .	. . .	. . .

当科目更多，无法直接观察呢？？

4.1 降维

3　　降维后，欲达到的目标

- 减少冗余信息造成的误差，可提高识别精度或分类效果
- 寻找数据内部的本质结构特征
- 加速后续计算的速度
- 在很多算法中，降维算法成为了数据预处理的一部分，如主成分分析(PCA)。事实上，有一些算法如果没有降维预处理，其实是很难得到很好的效果的。

4.2 PCA

1 PCA降维

- Principal Component Analysis(PCA)是最常用的线性降维方法。
- 它的目标是通过某种线性投影，将高维的数据映射到低维的空间中表示，并期望在所投影的维度上数据的方差最大，以此使用较少的数据维度，同时保留住较多的原数据点的特性。

4.2 PCA

2　降维的过程（设有m条n维数据）

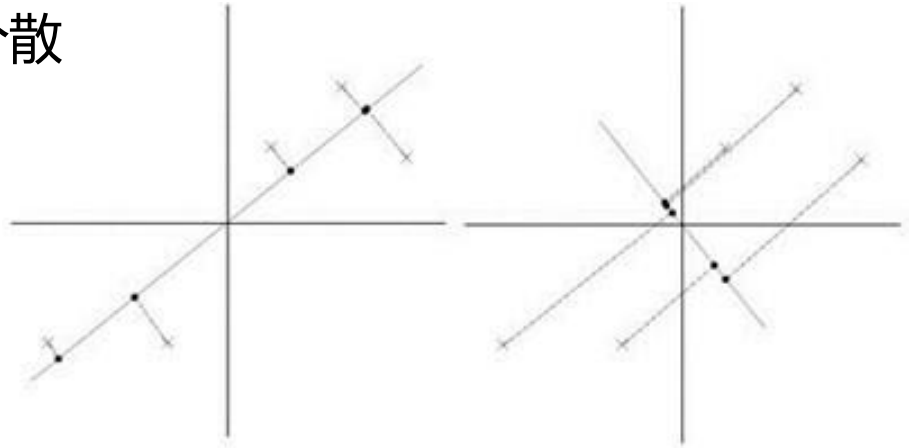
- 将原始数据按列组成n行m列矩阵X
- 数据预处理：将X的每一行（代表一个属性字段）进行零均值化，即减去这一行的均值
- 求出协方差矩阵 $C = \frac{1}{m} X X^{\mathrm{T}}$
- 求出协方差矩阵的特征值及对应的特征向量
- 将特征向量按对应特征值大小从上到下按行排列成矩阵，取前k行组成矩阵P
- $Y = PX$ 即为从n维降维到k维后的数据

4.2 PCA — 关键问题导读

如何选择这个投影方向，才能尽量保留最多的原始信息呢?

解决方案:

- 一种直观的方法是观察，
- 投影后的投影值尽可能分散



4.2 PCA – 实例

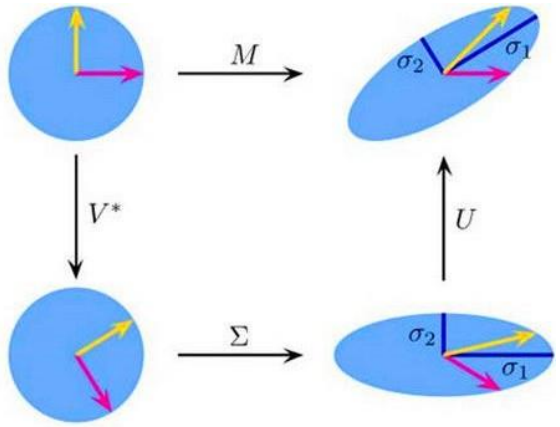
- *PCA.example*

4.3 SVD

1 SVD（奇异值分解）与 PCA：

- PCA的实现一般有两种，一种是用特征值分解去实现的，一种是用奇异值分解去实现的。

2 SVD实现的原理：



$$M = U \cdot \Sigma \cdot V^*$$

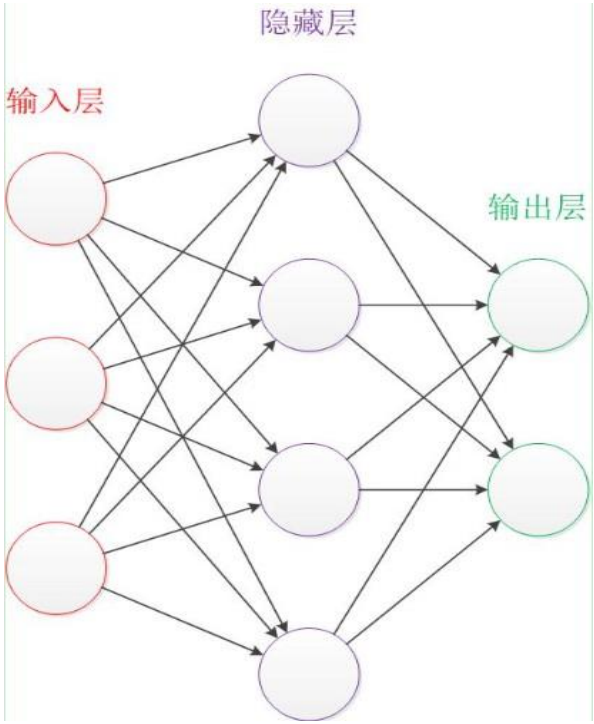
目录 CONTENTS

5.1 神经网络与深度学习

- 神经网络，是将许多个单一“神经元”联结在一起，一个“神经元”的输出就可以是另一个“神经元”的输入。
- 神经网络中，神经元处理单元可表示不同的对象，例如特征、字母、概念，或者一些有意义的抽象模式。
- 网络中处理单元的类型分为三类：输入单元、输出单元和隐单元。
  - 输入单元接受外部的信号与数据；
  - 输出单元实现系统处理结果的输出；
  - 隐单元是处在输入和输出单元之间，
  - 神经元间的连接权值反映了单元间的连接强度，信息的表示和处理体现在网络处理单元的连接关系中。

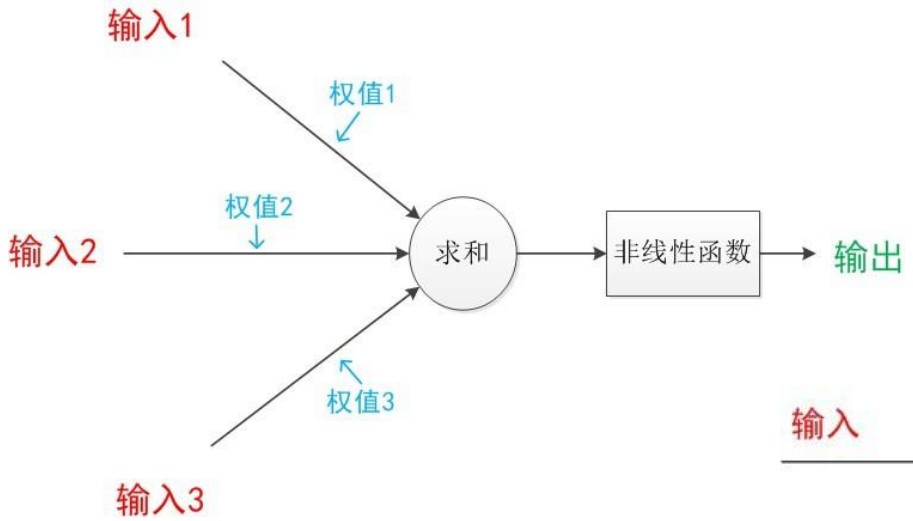
5.1 神经网络与深度学习

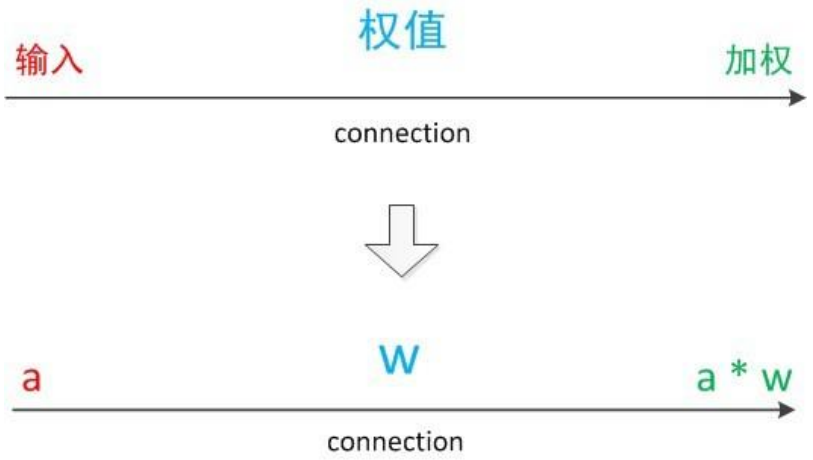
- 下图是一个包含三个层次的神经网络。红色的是输入层，绿色的是输出层，紫色的是中间层（也叫隐藏层）。输入层有3个输入单元，隐藏层有4个单元，输出层有2个单元。



5.1神经网络与深度学习

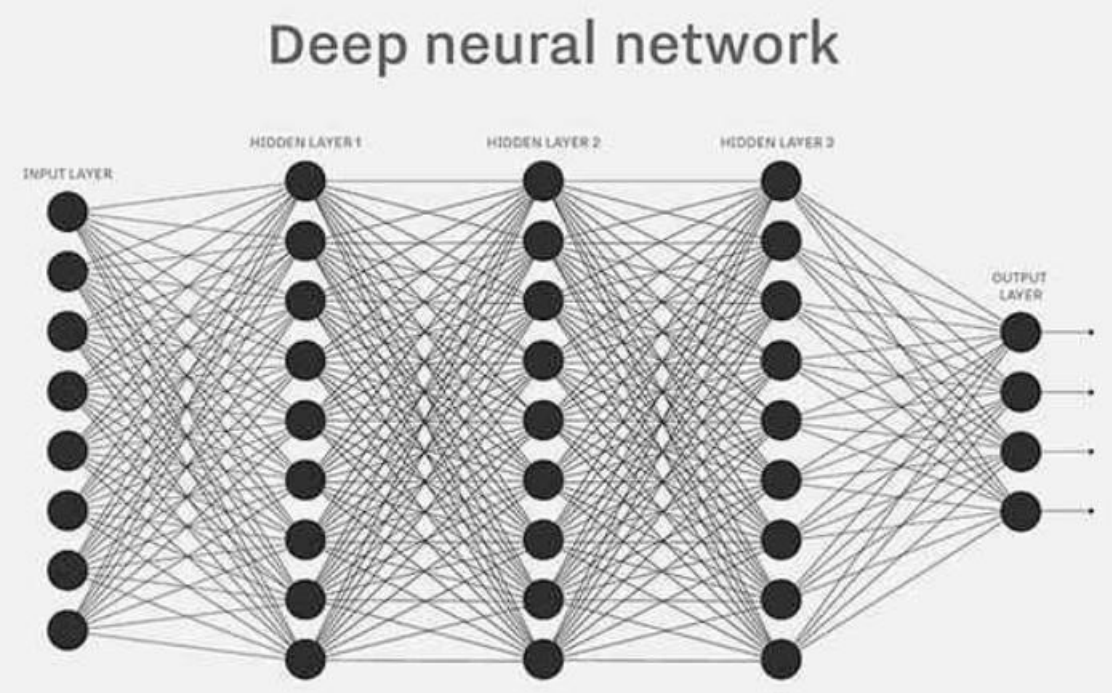
- 神经网络抽象为数学模型





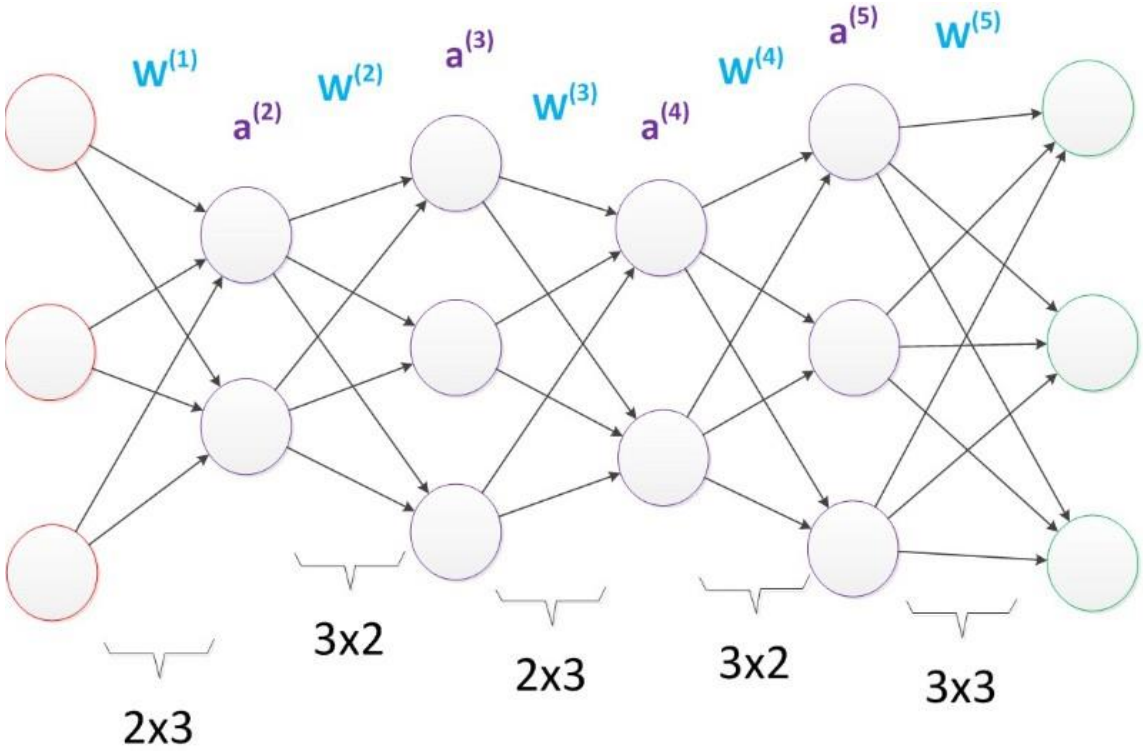
5.1神经网络与深度学习

- 深度学习是基于人工神经网络的研究，含多个隐层的多层感知器就是一种深度学习结构。



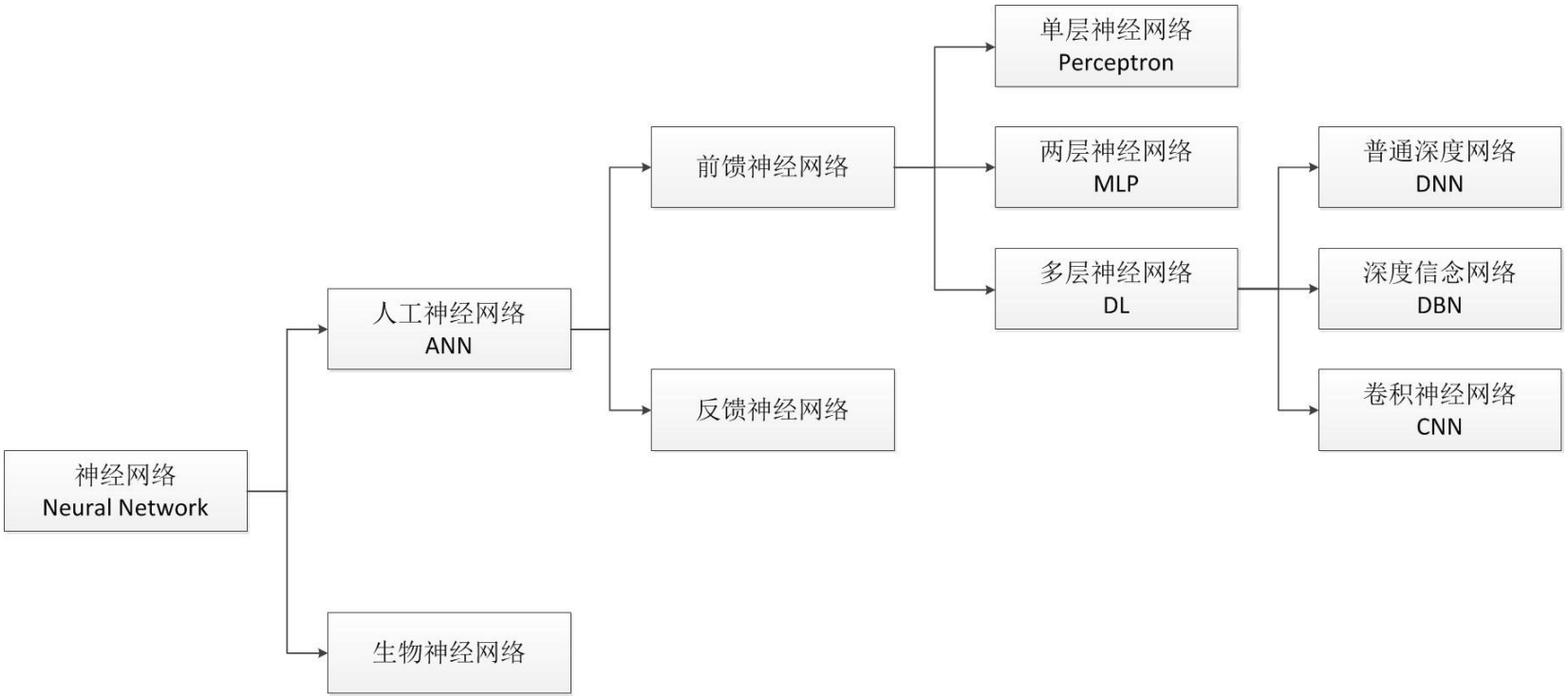
5.1神经网络与深度学习

- 深度学习分解为多个简单的网络



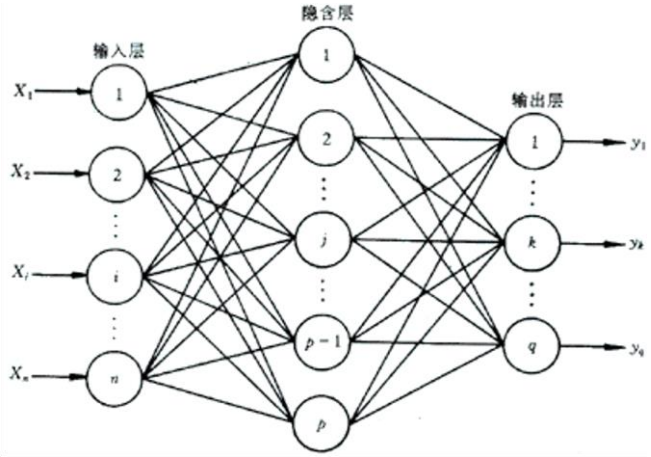
5.1神经网络与深度学习

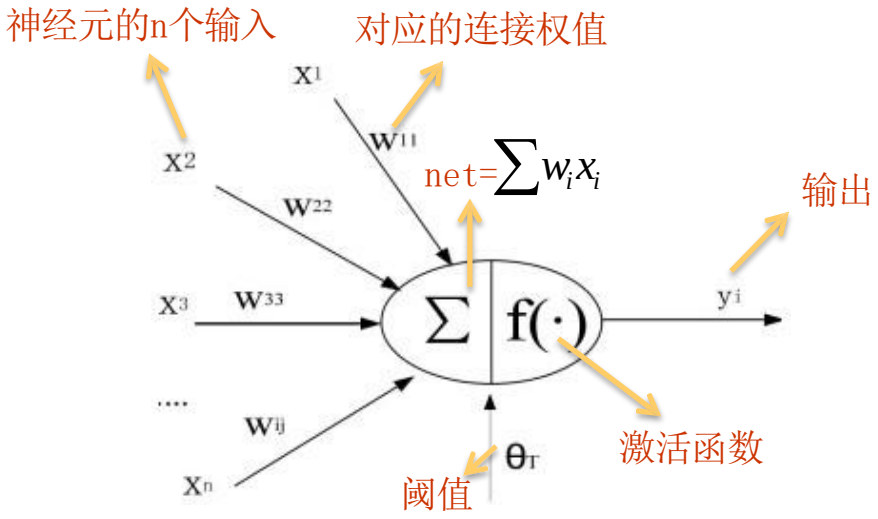
- 神经网络与深度学习的关系与分类



5.2 ANN

- 人工神经网络（Artifical Neural Network，ANN）结构





5.2 ANN

- 数学建模:

$$o_j = f\left(\text{net}_j\right) = f\left(\sum_{i=1}^{n} w_{ji} x_i - \theta_j\right) = f\left(\sum_{i=0}^{n} w_{ji} x_i\right)$$

其中， θ_j是阈值；$w_{j0} = -\theta_j$；$x_0 = 1$；

5.2 ANN

- 训练（学习）过程
  - Step1　设置连接权W的初值。对权系数W=（w_{ji}）的各个元素置一个较小的随机值。
  - Step2　输入样本X=（ x_1，x_2 ，…，x_n ），以及它的期望输出Y=（ y_1，y_2 ，…，y_n ）。
  - Step3 计算感知器的实际输出值

$$o_j = f\left(\sum_{i=0}^{n} w_{ji} x_i\right)$$

  - Step4 根据实际输出求误差

$$e_j = y_j - o_j$$

5.2 ANN

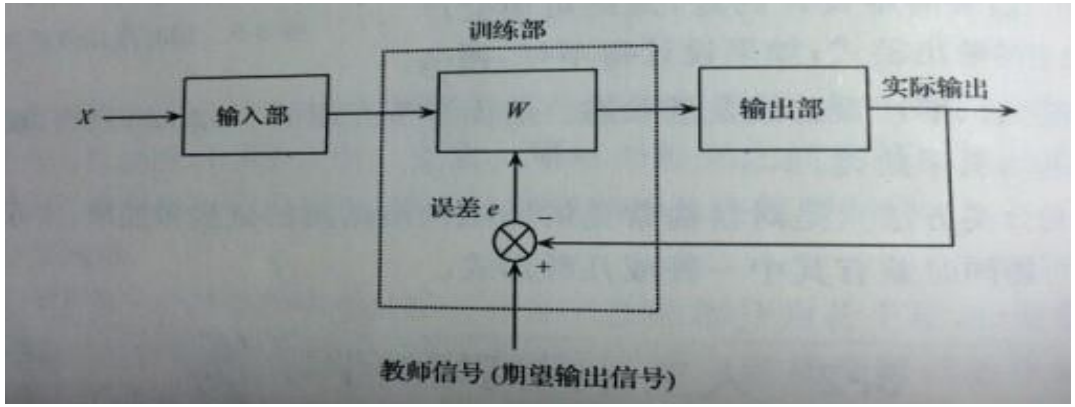
- 训练（学习）过程
  - Step5 用误差e_j去调整权值

$$w_{ji}(n+1)=w_{ji}(n)+\eta \cdot e_j \cdot x_i \quad ，i=0,1，\ldots，n$$

    - W_{ji}（n）是第n次调整连接权值；
    - η称为学习效率，且0＜η≤1，用于调整权值的调整速度。通常，η的取值不能太大，如果η的取值太大，则会影响$W_{ji}(n)$的稳定，η的取值太小则会使$W_{ji}(n)$得收敛速度太慢。当实际输出和期望值y相同时，有 $W_{ji}(n+1)=W_{ji}(n)$。
  - Step6 转到step2，一直执行到一切样本均稳定为止。

5.3 BP神经网络

- 经典网络模型—BP神经网络
  - BP神经网络（Back Propagation Neural Network），即误差后向传播神经网络，是一种按误差逆向传播算法训练的多层前馈网络，是目前应用最广泛的网络模型之一。



5.3 BP神经网络

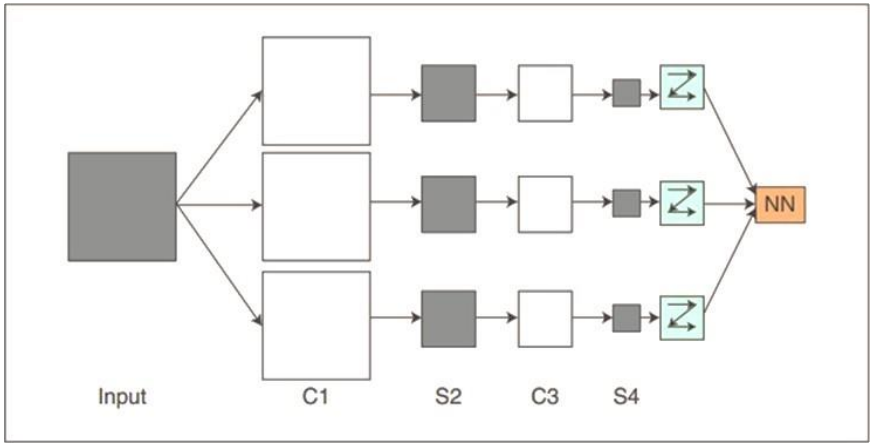
- BP神经网络训练过程
  - 初始化连接权值v_{ki}和w_{jk}；
  - 初始化精度控制系数ε；
  - E= ε+1；
  - while E > ε do
  - 1. E=0
  - 2. 对S中的每一个样本（X^p，Y^p）
  - 1. 计算出X^p，对应的实际输出o_p；
  - 2. 计算出E_p；
  - E.2.3 E=E+ E_p；
  - E.2.4 根据$\Delta w_{jk} = \eta \delta_j z_k$调整输出层的权值$w_{jk}$（n）；
  - E.2.4 根据$\Delta v_{ki} = \eta \delta_k x_i$调整输出层的权值$v_{ki}$（n）；
  - E.3 E=E/2.0

5.4 CNN

- 卷积神经网络（CNN）
  - 是神经网络的一种，专门用来处理矩阵输入的任务，能够将矩阵形式的输入编码为较低维度的一维向量，而保留大多数有用信息。
- 应用领域：
  - 图像分类，目标检测，目标识别，目标跟踪，文本检测和识别以及位置估计
  - 很少应用于数据分类领域

5.4 CNN

- CNN模型结构



  - C层为特征提取层；
  - S层是特征映射层，
  - 特征映射结构采用影响函数核小的sigmoid函数作为卷积网络的激活函数。

5.4 CNN

- 给出训练样本集 $\{X_i, Y_i\}$
- 根据X_i以及CNN的模型结构，前向传播(forward propagation)
- 前向传播结果计算出模型输出 y_i，并计算出损失函数 $J(\theta) = L(y, y')$
- 根据损失函数进行反向传播(back propagation)，计算出所以参数梯度
- 根据参数梯度进行梯度下降算法，求取最后模型参数

5.4 CNN

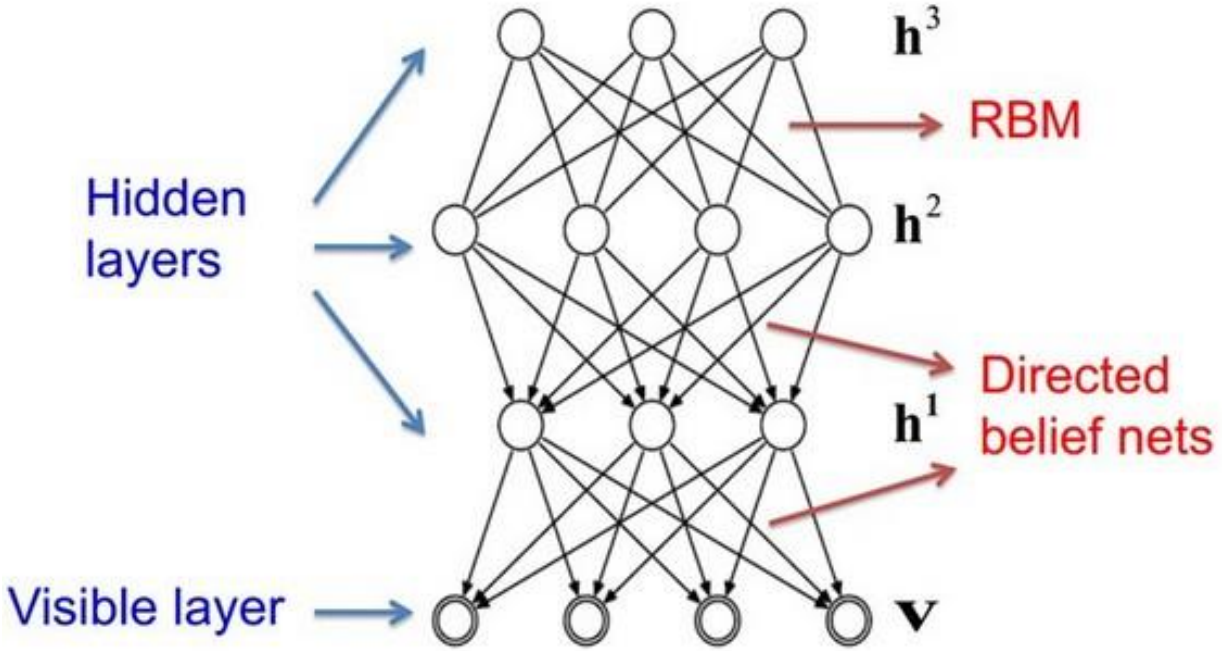
- CNN优点：
  - 避免了显式的特征抽取，而隐式地从训练数据中进行学习
  - 同一特征映射面上的神经元权值相同，所以网络可以并行学习
  - 布局更接近于实际的生物神经网络，权值共享降低了网络的复杂性，避免了特征提取和分类过程中数据重建的复杂度

5.5 RNN

- 递归神经网络
  - 作用跟卷积神经网络是一样的，将矩阵形式的输入编码为较低维度的一维向量，而保留大多数有用信息。
  - 跟卷积神经网络的区别在于，卷积神经网络更注重全局的模糊感知，而RNNs则是注重邻近位置的重构
- 应用领域：
  - 自然语言处理

5.6 DBN

DBN结构


h3
RBM
Hidden layers
h2
Directed belief nets
h1
Visible layer
v

5.6 DBN

DBN网络中存在的问题：

- 需要为训练提供一个有标签的样本集；
- 学习过程较慢；
- 不适当的参数选择会导致学习收敛于局部最优解

目录 CONTENTS

6 关于模型的评价

- 回归模型评估：
  - 模型的拟合度
  - 偏差平方和
  - 局部最优值与全局最优
- 分类模型评估：
  - 准确率、精确率、召回率、F1值
  - ROC曲线和AUC

6.1 方差与协方差

- 定义：
  - 方差：用于两个及两个以上样本均数差别的显著性检验，即判断总体的真实情况与原假设是否有显著性差异

$$s^2 = \frac{\sum_{i=1}^{n}(X_i - \bar{X})^2}{n-1}$$

  - 协方差：表示的是两个变量总体误差的期望

$$\begin{aligned} Cov(X,Y) &= E[(X - E[X])(Y - E[Y])] \\ &= E[XY] - 2E[Y]E[X] + E[X]E[Y] \\ &= E[XY] - E[X]E[Y] \end{aligned}$$

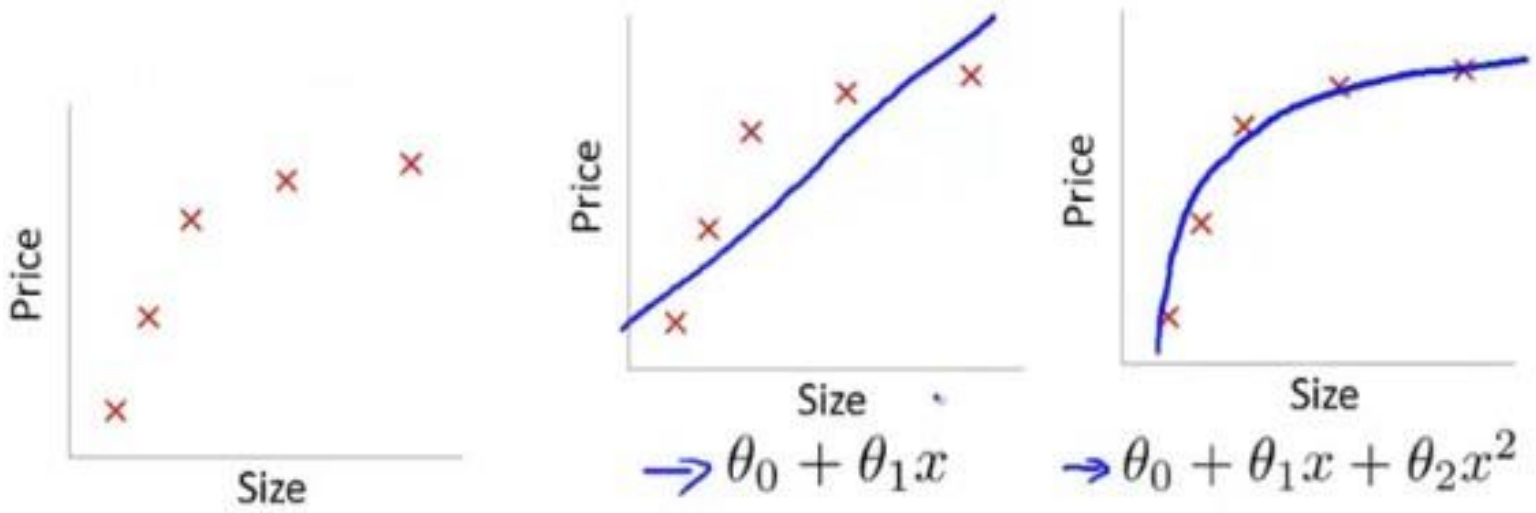
- 应用：
  - 当多个变量独立时，用方差来评估这种影响的差异。
  - 当多个变量相关时，用协方差来评估这种影响的差异，用于评估它们因相关而产生的对应变量的影响

6.2 回归模型评估标准

- 拟合程度
  - 拟合度检验是对已制作好的预测模型进行检验，比较它们的预测结果与实际发生情况的吻合程度。
  - 由测量的数据，估计一个假定的模型/函数。拟合的模型是否合适？可分为以下三类：
    - 欠拟合
    - 合适拟合
    - 过拟合

6.2 回归模型评估标准

- 欠拟合：选定的模型没有很好地捕捉到数据特征，不能够很好地拟合数据

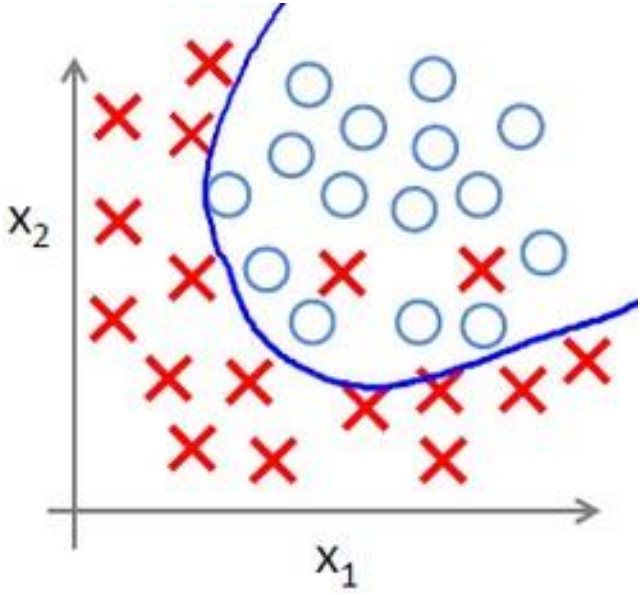


6.2 回归模型评估标准

- 解决方案：
  - 添加其他特征项，有时候模型出现欠拟合的时候是因为特征项不够导致的。
  - 添加多项式特征，这个在机器学习算法里面用的很普遍，可以将线性模型添加二次项或者三次项使模型泛化能力更强。
  - 减少正则化参数，正则化的目的是用来防止过拟合的，模型出现了欠拟合，则需要减少正则化参数。

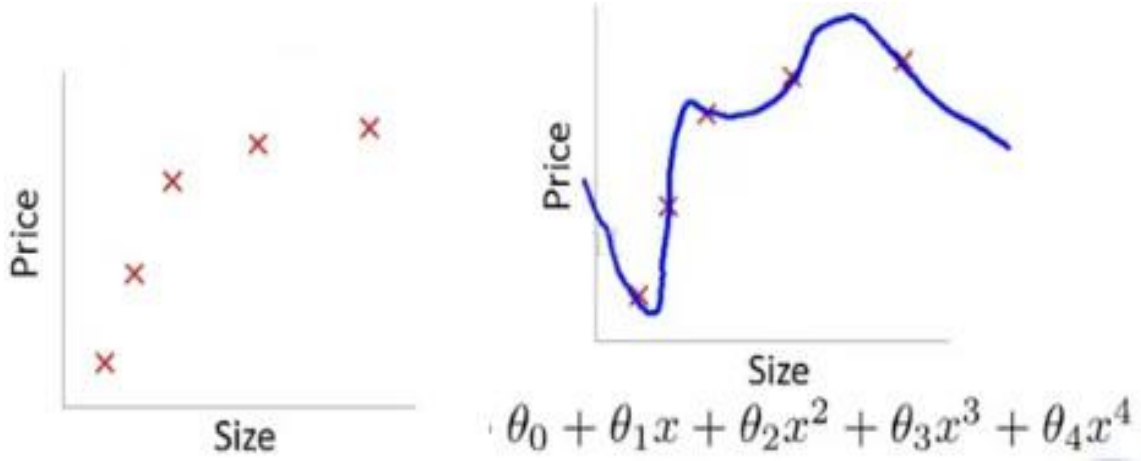
6.2 回归模型评估标准

- 合适拟合：选定的模型很好地捕捉到数据特征，能够很好地拟合数据，允许个别失误



6.2 回归模型评估标准

- 过拟合：模型将数据学得太彻底，把噪声数据的特征也学习到了，导致在后期测试时候不能够很好地识别训练集外的数据，模型泛化能力太差，只适用于训练和测试的数据。



6.2 回归模型评估标准

- 解决方案：
  - 重新清洗数据，通过人工选择，或者采用模型选择算法，减少特征的数量。
  - 增大数据的训练量，还有一个原因就是我们用于训练的数据量太小导致的，训练数据占总数据的比例过小。采用正则化方法。如：为防止过拟合的模型出现，在损失函数里增加一个每个特征的惩罚因子。这个就是正则化。如正则化的线性回归的损失函数：

$$J(\theta) = \frac{1}{2m}\left[\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2 + \lambda \sum_{j=1}^{n}\theta_j^2\right]$$

6.2 回归模型评估标准

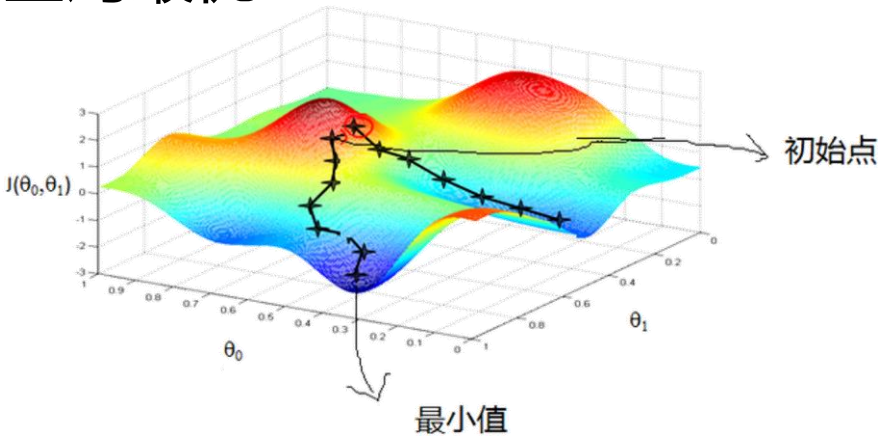
- 偏差平方和

$$\sum_{i=1}^{N}\left[y_i-\left(a+bx_i\right)\right]^2$$

  - 评估标准：要使得回归系数最优，则需要保证所有数据偏差的平方和最小

6.2 回归模型评估标准

- 局部最优值与全局最优



  - 对函数进行凸优化时，如果使用导数方法（如：梯度下降法）来寻找最优解，有可能陷入到局部最优解而非全局最优解。为了防止得到局部最优，可以对梯度下降法进行一些改进，防止陷入局部最优。

6.3 分类模型评估参数

- 准确率
  - Precision =提取出的正确信息条数 / 提取出的信息条数
- 召回率
  - Recall = 提取出的正确信息条数 / 样本中的信息条数
- F1值（综合评估）
  - F1 = 正确率 * 召回率 * 2 / (正确率 + 召回率)

TN: 预测为负，实际为负　　TP: 预测为正，实际为正　　准确率： TP/ (TP+FP)

FN: 预测为负，实际为正　　FP: 预测为正，实际为负　　召回率： TP/ (TP +FN)

6.3 分类模型评估参数

一个数据库有500个文档，其中有50个文档符合定义的问题。系统提取筛选到75个文档，但是只有45个符合定义的问题。

- TP = 45； FP： 30； TN： 425； FN： 5；
- 准确率P=TP/(TP+FP)=45/75=60%
- 召回率R=TP/(TP+FN)=45/50=90%
- F1:正确率 * 召回率 * 2 / (正确率 + 召回率) =72%

TP: 预测为正，实际为正

FP: 预测为正，实际为负

TN: 预测为负，实际为负

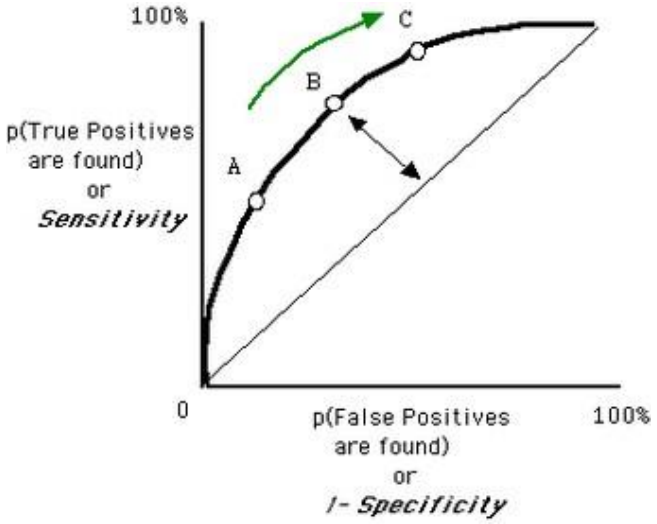
FN: 预测为负，实际为正

准确率： TP/ (TP+FP)

召回率： TP/ (TP +FN)

6.4 ROC曲线与AUC

- ROC (Receiver Operating Characteristic)曲线
  - 由两个变量1-specificity 和 Sensitivity绘制。1-specificity=FPR，即负正类率。Sensitivity即是真正类率，TPR(True positive rate)，反映了正类覆盖程度，曲线距离左上角越近,证明分类器效果越好。



6.4 ROC曲线与AUC

- AUC曲线
  - AUC曲线是ROC曲线所覆盖的区域面积，AUC越大，分类器分类效果越好。假设分类器的输出是样本属于正类的socre（置信度），则AUC的物理意义为，任取一对（正、负）样本，正样本的score大于负样本的score的概率，且概率为：

$$\frac{\sum_{\text{所有正样本}} \text{rank} - M(M+1)/2}{M * N}$$

    - 取N*M(N为正样本数，M为负样本数) rank=n(n=N+M)