# 《数据挖掘》考试题型与分数

一、共四道大题：
填空题：每空1分，共20分。
判断题：每题2分，共8题，总计16分。
简答或辨析题：共6题，每题6分，总计36分。
计算题（可以使用计算器）：4道题，共28分。

# 《数据挖掘》填空、判断题复习范围

| 考点 | 来源 | |
|---|---|---|
| 1、OLTP、OLAP、KDD、BI、ETL、CNN、RNN、GAN的缩写（英语） | | |
| 2、数据仓库的特征、数据仓库模型、数据仓库的多层结构、度量（聚集函数）的分类 | | |
| 3、两种学习模型（有监督与无监督） | | |
| 4、数据预处理及其主要内容 | | |
| 5、挖掘的知识类型 | | |
| 6、常见的OLAP操作 | | |
| 7、数据立方的两种表 | | |
| 8、属性离散化 | | |
| 9、P、NP、NP完全与NP难 | | |
| 10、判别与生成模型 | | |
| 11、深度网络训练面临的挑战及贪心有监督和无监督预训练方法 | | |
| 12、常见聚类准则及其合理性 | | |
| 13、过拟合与欠拟合现象 | | |
| 14、训练集、验证集与测试集 | | |
| 15、泛化误差及其偏差与方差分解 | | |
| 16、数据立方体的方体格 | | |
| 17、决策树、贝叶斯分类器及最近邻分类器的时间与空间复杂性比较 | | |

# 《数据挖掘》简答题复习范围

| 考点 | 来源 | |
|---|---|---|
| 1、知识发现过程的主要步骤 | | |
| 2、OLTP与OLAP的主要区别 | | |
| 3、为什么需要构建单独隔离的数据仓库 | | |
| 4、数据预处理内容 | | |
| 5、数据仓库的三种设计模式 | | |
| 6、三种度量函数的定义 | | |
| 7、评估分类器准确率的方法 | | |
| 8、神经网络的结构设计 | | |
| 9、深度学习训练算法及正则化技术 | | |
| 10、常见的深度网络类型 | | |
| 11、自编码器 | | |
| 12、没有免费的午餐定理 | | |
| 13、文本摘要基本概念 | | |
| 14、链接分析算法主要特点 | | |
| 15、传播最大化模型 | | |

# 《数据挖掘》计算题复习范围

| 考点 | 来源 | |
|---|---|---|
| 一、用Apriori算法挖掘强关联规则 | | |
| 二、频繁模式树的构造 | | |
| 三、信息增益的计算 | | |
| 四、朴素贝叶斯分类 | | |
| 五、k-means聚类 | | |
| 六、凝聚层次聚类法（全链接与单链接） | | |
| 七、神经网络的正反向传播计算（sigmoid、ReLU、softmax激活函数） | | |

1、OLTP、OLAP、KDD、BI、ETL、CNN、RNN、GAN的缩写（英语）

联机事物处理 (OLTP ， on-line transaction processing)
联机分析处理 (OLAP，on-line analytical processing)
数据库中的知识发现（Knowledge Discovery in Databases, KDD)
卷积神经网络(Convolutional Neural Network, CNN)
循环神经网络(Recurrent Neural Network, RNN)
生成对抗网络(Generative Adversarial Network, GAN)

商业智能（Business Intelligence, BI），是指通过对数据的收集、管理、分析以及转化，使数据成为可用的信息，从而获得必要的洞察力和理解力，更好地辅助决策和指导行动。

数据抽取、转换、装载的过程 (Extract-Transform-Load, ETL)，是构建数据仓库的重要一环，用户从数据源抽取出所需的数据，经过数据清洗，最终按照预先定义好的数据仓库模型，将数据加载到数据仓库中去。

## 2、数据仓库的特征、数据仓库模型、数据仓库的多层结构、度量（聚集函数）的分类

- 定义
  - 与操作数据库相隔离并单独维护的一个用来支持决策过程的数据库
  - 一个用来对整理过的历史数据进行分析以便支持信息处理的固定平台.
  - 数据仓库是**面向主题的、集成的、时变的、非易失的**数据集合，它用来支持管理部门的决策过程
- 特点
  - 面向主题的
    - 围绕主题组织，如消费者、产品
    - 主要目的是对数据建模和分析，而不是日常操作
    - 排出对决策过程无用的数据
  - 集成的
    - 应用多个、异构数据源
    - 应用到数据清洗和数据集成技术
  - 时变的
    - 数据仓库跨越的时间比操作数据库（只需要当前值数据）要长的多
    - 数据仓库的健值属性：隐式或显式地包含一个时间键（操作数据库可以不包括时间键）
  - 非易失的
    - 与操作数据库分隔存储
    - 操作数据库的数据更新不在数据仓库环境出现
      - 不需要事务处理，数据恢复以及并发控制机制
      - 仅仅需要以下2种操作
        - 数据的初始装载
        - 数据访问

2、数据仓库的特征、**数据仓库模型、数据仓库的多层结构**、度量（聚集函数）的分类

**数据仓库的特征（四种）**
面向对象的、集成的、时变的、非易失的

**数据仓库模型（三种模型）**
**企业仓库**（Enterprise warehouse）
搜集了关于主题的所有信息，跨越整个组织。
**数据集市**（Data Mart）
包含企业范围数据的一个子集，对于特定的用户是有用的，其范围限于选定的主题。
**虚拟仓库**（Virtual warehouse）
操作数据库上视图的一组集合。
为了有效处理查询，只有一些可能的汇总视图被物化。

**数据仓库的多层结构（三层）**
底层是仓库数据库服务器，通常是关系数据库系统。
中间层是OLAP服务器。
顶层是客户，包括查询和报表工具。

# 《数据挖掘》填空、判断题复习范围

2、数据仓库的特征、数据仓库模型、数据仓库的多层结构、**度量（聚集函数）的分类**

- 度量函数的分类
  - 分布式的
    - 将函数用于n个聚集值得到的结果，与将函数用于所有数据得到的结果一样
    - count(), sum(), min(), max()
  - 代数的
    - 由M个分布式函数进行代数计算得到
    - avg(), standard_deviation()
  - 整体的
    - median()（中位数），mode()（众数）

# 《数据挖掘》填空、判断题复习范围

3、两种学习模型（有监督与无监督）：*课程结构如下、会判断、标准为是否有指导信号*

**有监督学习：分类**
①决策树
②K-最近邻分类
③朴素贝叶斯学习
④深度学习
⑤集成学习、没有免费的午餐定理、样本复杂性

**无监督学习：聚类分析**
①基于划分的聚类算法
②层次聚类算法

4、数据预处理及其主要内容

**数据清洗（Data cleaning）**
填充遗失的数据, 平滑噪声数据, 辨识或删除孤立点, 解决不一致性问题
**数据集成（Data integration）**
对多个数据库，数据立方或文件进行集成
**数据变换（Data transformation）**
规范化与聚集（Normalization and aggregation）
**数据约简（Data reduction）**
得到数据集的压缩表示，它小的多，但能够产生同样的（或几乎同样的）分析结果
**数据离散化（Data discretization）**
特别对数字值而言非常重要

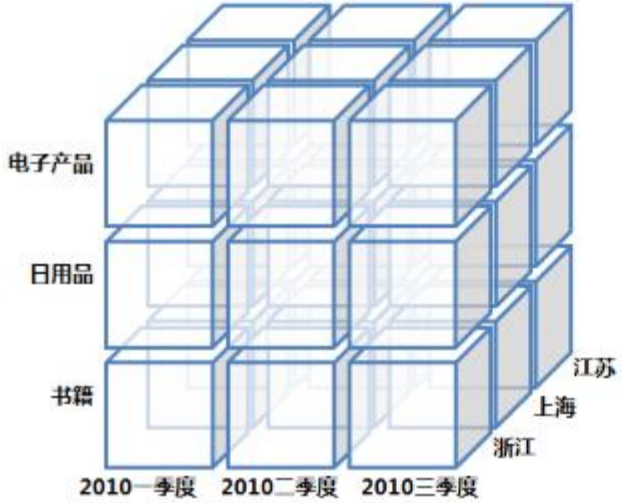
5、挖掘的知识类型

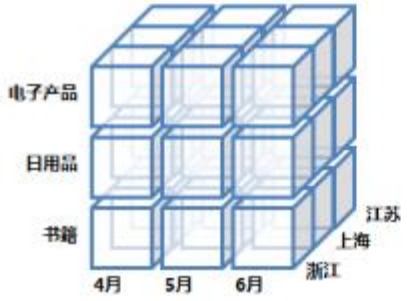
**(1) 概念/类描述：特征化和区分**
**(2) 挖掘频繁模式、关联和相关**
**(3) 分类和预测**
**(4) 聚类分析**
**(5) 离群点分析**
**(6) 演变分析**

# 《数据挖掘》期末复习

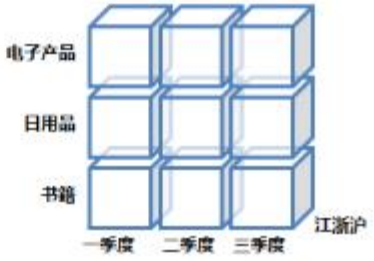
6、常见的OLAP操作

OLAP基本操作
- 上卷roll up/上钻drill up
- 下钻drill down/roll down
- 切片slice
- 切块dice
- 旋转pivot

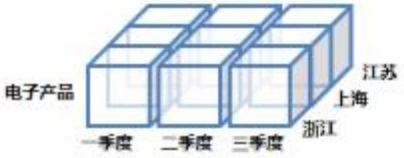


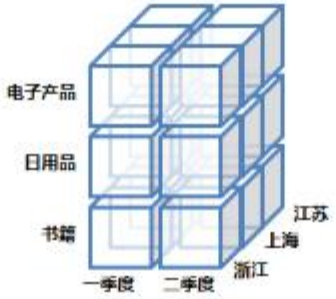

钻取(Drill-down)



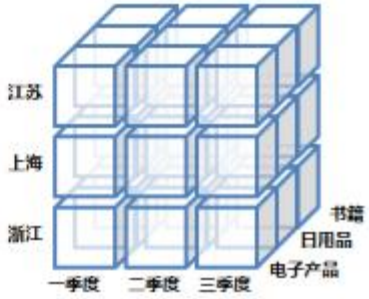

上卷(Roll-up)




切片(Slice)




切块(Dice)




旋转(Pivot)

https://blog.csdn.net/qq_36739040

7、数据立方的两种表

**维表**：如维item (item_name, brand, type)，或维time(day, week, month, quarter, year) 。
**事实表**：包含度量（measures）：如销售额以及每个相关维表的关键字。

下面是两个常见的维度表结构：
**产品维度表**：Prod_id, Product_Name, Category, Color, Size, Price
**时间维度表**：TimeKey, Season, Year, Month, Date

而事实表是数据聚合后依据某个维度生成的结果表。它的结构示例如下：
**销售事实表**：Prod_id(引用产品维度表), TimeKey(引用时间维度表), SalesAmount(销售总量，以货币计), Unit(销售量)

8、属性离散化

属性的三种类型:
**①标称性的** — 取自于无序集合(unordered set)的值
**②有序的**(Ordinal) —取自于有序集合(ordered set)的值
**③连续的** — 实数

离散化:
把连续型属性的取值范围划分成多个小的区间，通过离散化减少数据集大小，为进一步分析做好准备

方法：**①分箱（**等深分箱离散法**）** ②**直方图分析** ③**基于熵的离散化**（使其后划分得到的信息增益最大）

9、P、NP、NP完全与NP难

**P问题：在多项式时间内“能解决”的问题。**
如找n个数中的最大数，最小生成树问题等

**NP问题：在多项式时间内不知道是否“能解决”的问题。**
但是在多项式时间内“能验证”的问题。
如可满足性(SAT)问题

**NP-Complete问题（NP-完全问题）**
所有NP问题能在多项式时间内规约到该问题;
且该问题本身属于NP问题.

**NP-Hard问题（NP-难问题）**
所有NP问题能在多项式时间内规约到该问题.

NPC问题==判定问题，NP难问题==优化问题

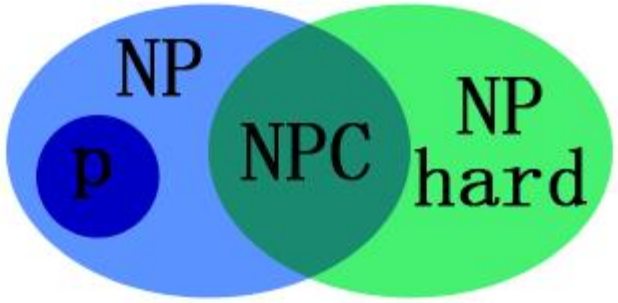


图1 P NP NPC NPhard关系的图形表示

10、判别与生成模型

| 角度 | 判别模型 | 生成模型 |
| --- | --- | --- |
| 贝叶斯 | 判别模型估计的是条件概率分布（后验概率），p(y\|x) | 生成模型估计的是联合概率分布 p(y, x) = p(y\|x) * p(x) |
| 模型建立 | 判别模型是直接用决策模型做类别区分/计算概率。 | 生成模型是首先学习出一个模型，即首先建立样本的联合概率概率密度模型P(X,Y)，再得到后验概率P(Y\|X)，再利用它进行分类。 |
| 数据的关注角度 | 关注数据类别间（inter-class）的特点，寻找不同类别之间的**分类面**（decision boundary），反映的是异类数据之间的差异。 | 生成模型关注数据类别内（intra-class）的特点，从统计的角度表示**数据的分布情况**，能够反映同类数据本身的相似度 |
| 算法举例 | **KNN、感知机（神经网络）、决策树、逻辑斯蒂回归、最大熵模型、SVM** | **朴素贝叶斯、贝叶斯网络** |
| 关系 | 由生成模型可以得到判别模型，但由判别模型得不到生成模型。（生成模型带的信息一般要比判别模型丰富） | |
| 形象类比 | 例如对面一个人走过来，和你说了一句话，你需要识别出她说的到底是汉语、英语还是法语等。 | |
| | 用判别模型的方法是不去学习每种语言，你只学习语言模型之间的差别，然后再分类。 | 用生成模型的方法是学习每一种语言，你花了大量精力把个语言都学会了，就会知每种语音对应什么样的语言。 |

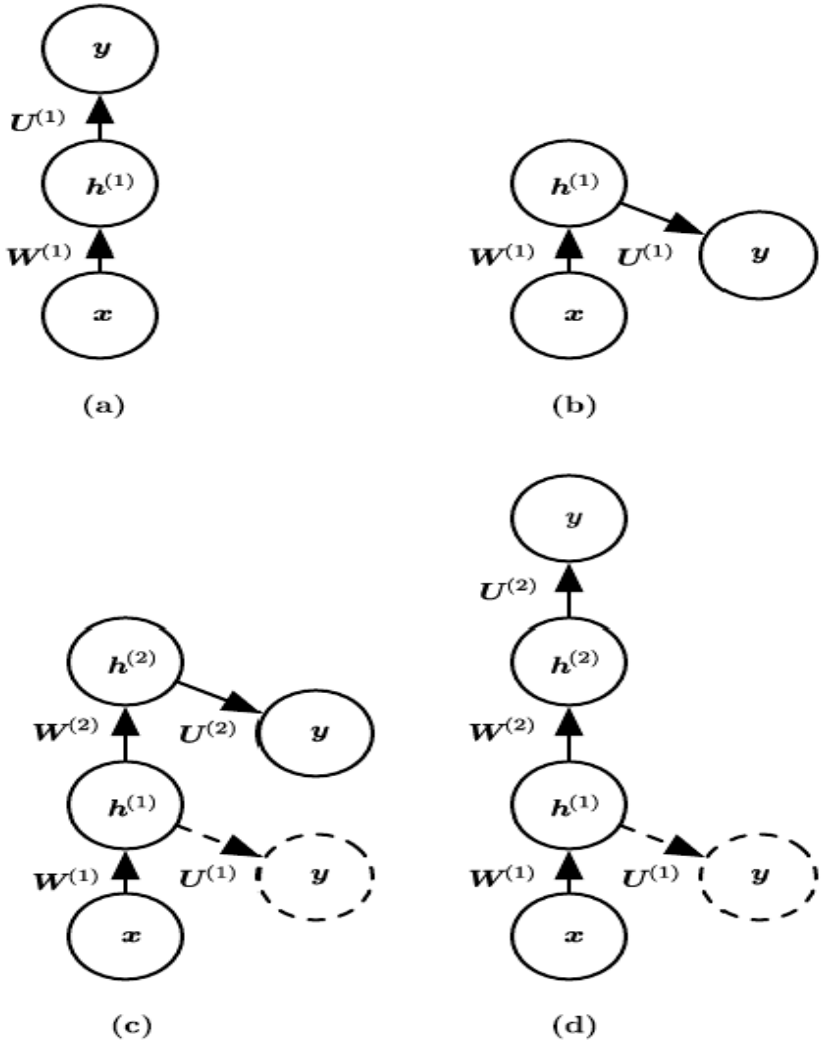
11、深度网络训练面临的挑战及贪心有监督和无监督预训练方法

**BP算法直接用于深度网络面临的挑战**

①数据获取问题：标记数据少,无标记数据多;
②局部最优问题：容易陷入局部最优;
③梯度消失问题：当使用反向传播方法计算导数的时候，随着网络的深度的增加，反向传播的梯度（从输出层到网络的最初几层）的幅度值会急剧地减小;
④如果对所有层同时训练，时间复杂度会太高;

**贪心有监督和无监督预训练方法:**

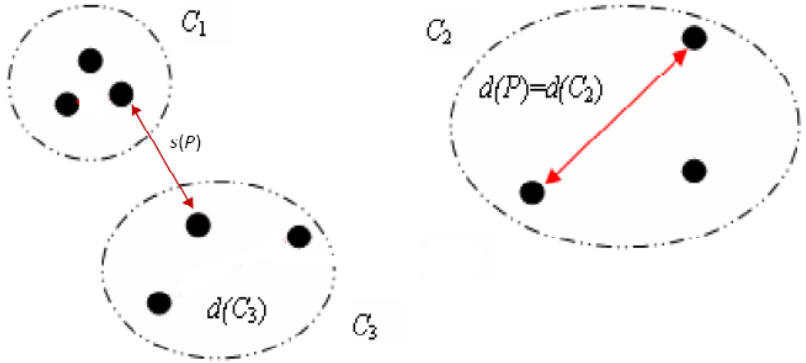
**核心思想：**逐层贪婪预训练(Greedy layer-wise pretraining)
**基本方法：**从底层(输入层)开始，一层一层的往顶层(输出层)训练，原始输入作为训练第一个隐藏层的输入；训练好一层后，参数固定，再在其上堆积（Stacking，又称栈化）一层，然后用前一层的输出作为训练该层的输入
**算法分类：**上述训练可以是无监督的或有监督的
无监督：如使用受限玻尔兹曼机(RBM)，自编码器(auto-encoder)等;
有监督：直接用训练集的标识作为训练一层时的输出（<u>将每一步的分类误差作为目标函数</u>）。



12、常见聚类准则及其合理性

【NPH】K-Center：最大半径最小化；
【NPH】k-Cluster：最大直径的最小化；
【P】最小分离度的最大化
【P/NPH】k-median：聚类内部距离之和的最小化
（若视k为输入变量，则NP-hard；对于固定的k，则是P问题）
【NPH】 k-means：聚类内部距离平方之和的最小化
【NPH】 MRSD准则：分离度s(P)/直径d(P) 最大化

$$\max_{P \in P_n^k} \frac{s(P)}{d(P)}$$



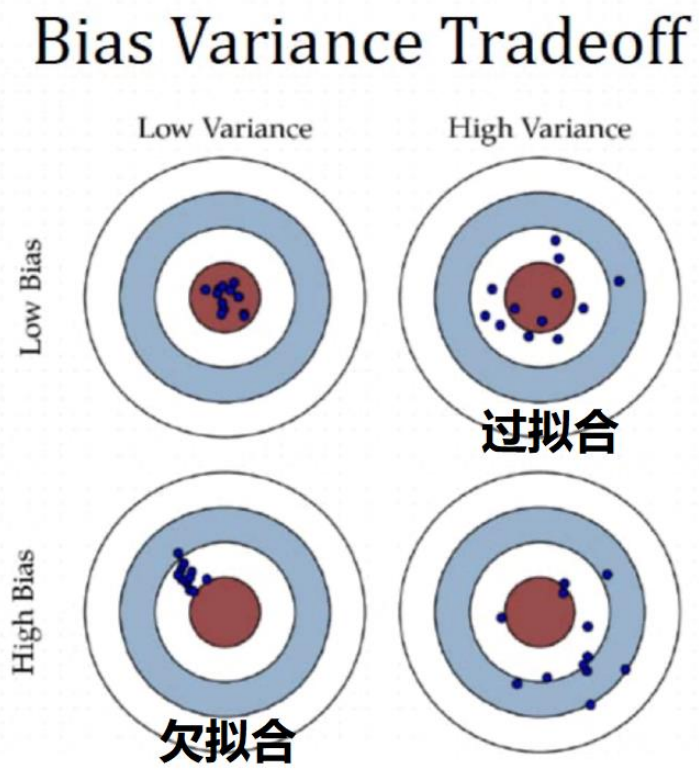

基于图论：
【P】最小割 【NPH】最大割 【NPH】规范割

合理性：

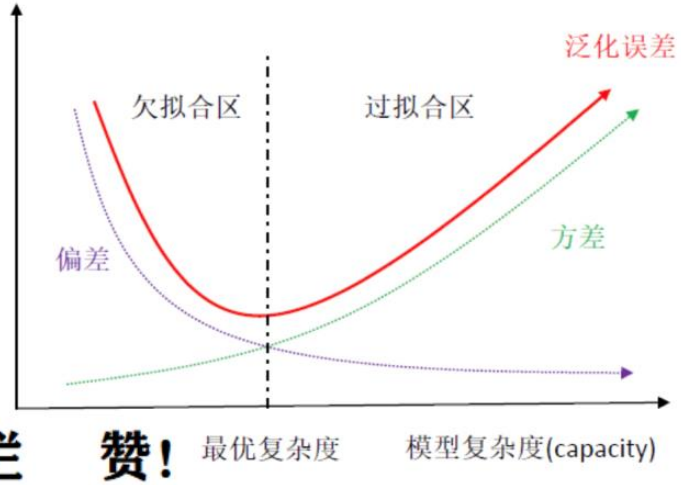
**一个 好的聚类方法 将会产生高质量的聚类结果**
聚类内对象的高度同质性(Homogeneity)
聚类间对象的高度分离性 (Separation)
**聚类算法的质量依赖于**
它所使用的相似性（或距离）的具体定义
优化准则的合理性、计算复杂性及优化算法

**聚类内部的同质性度量：**
聚类直径(Diameter)与半径(Radius)
**聚类间的分离性度量：**
聚类分离度(split)

13、过拟合与欠拟合现象




- 考察两个特殊的学习算法：
  - $h(x) = c$(常数)：即对任意的样本均输出一个常数值。显然模型的方差为0，但显然具有非常大的偏差！
  - $h(x) = y$：即完美拟合训练集。则模型的偏差为0，但模型的方差将为训练集的固有方差！




| | 过拟合 ① | 欠拟合 ② | 烂 ③ | 赞！ ④ |
|---|---|---|---|---|
| Train set(训练集) error | 1% | 15% | 15% | 0.8% |
| Dev set (验证集) error | 11% | 16% | 30% | 1% |

（假设human performance为0%）

14、训练集、验证集与测试集

在**训练集**上训练模型，在**验证集**上评估模型，一旦找到最佳参数，就在**测试集**上最后测试一次。

- 训练集(Training data):用于训练模型(的参数，如神经网络的各层权重和偏置);
- 验证集(Validation data):用于调整超参 数(Hyper-Parameters，如神经网络的宽度和深度等);
- 测试集(Test data):用于评价模型本身的 有效性(准确率等)

# 《数据挖掘》填空、判断题复习范围

15、泛化误差及其偏差与方差分解

$$Err(\boldsymbol{x}) = \mathbb{E}_D\left[\left(f(\boldsymbol{x};D) - \bar{f}(\boldsymbol{x})\right)^2\right] + \left(\bar{f}(\boldsymbol{x}) - y\right)^2 + \mathbb{E}_D\left[\left(y_D - y\right)^2\right]$$

- variance
- bias$^2$
- noise

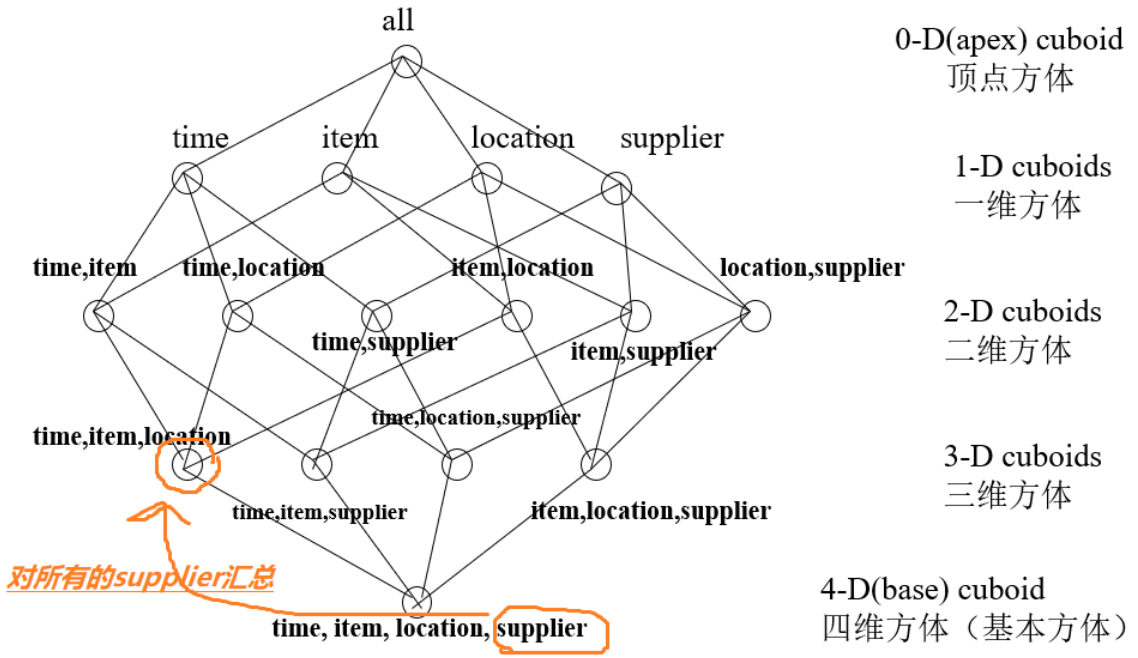
- 期望泛化误差=模型方差+模型偏差$^2$+噪声$^2$.
  - 其中噪声为固有特性，不可优化，任何模型的期望误差都将大于等于$\sigma^2$！
- 我们当然希望同时最小化模型方差与模型偏差，但往往是鱼与熊掌不可兼得！

- 第1项为 $E[(h(x^*)-\bar{h}(x^*))^2]$：模型的方差，它度量了在不同训练集上学习到的不同模型之间的性能差异。
- 第2项为 $(\bar{h}(x^*)-f(x^*))^2$： 模型的期望预测值与真实函数值的偏差的平方，它度量了学习到的平均模型与真实函数之间的差异
- 第3项为$\sigma^2$：为固有观察噪声的方差。

16、数据立方体的方体格

**基本方体**(base cuboid)：一个n维立方体（n-D)称为
**顶点方体**（apex cuboid）：0-D方体存放最高层的汇总
**方体的格**称作数据立方体（data cube）。

***方体格的元素有 顶点方体、一维方体、……n维方体***




***下层对某一维进行聚集/求和运算 得到上层***

***可以把任意n维数据立方体显示成(n-1)维“立方体”的序列***

***上层元素是下层元素的真子集***

青姐批注：
下层对某一维聚合得到上层，

所谓上层元素是下层元素的真子集是说“属性集合”，如{长，宽，高}->{长，宽}

但是自下而上数据项(items)会变多，因为不会对某一属性进行筛选限制了。

17、决策树、贝叶斯分类器及最近邻分类器的时间与空间复杂性比较

## 决策树的复杂度

训练时间复杂度= O(n*log(n)*d)

n=训练集中的样本数

d=数据的维数

运行时复杂度= O(树的最大深度)

注意:当我们有大量低维数据时，我们使用决策树。

## 朴素贝叶斯的复杂度

训练时间复杂度 = O（n * d）

运行时复杂度 = O（c * d）

我们必须为每个类'c'检索特征

## k最近邻算法的复杂度

训练时间复杂度 = O（knd）

遍历每个训练观测值，并计算机器学习训练集观测值和新观测值之间的距离 d。

时间相对于实例数（n）和维数（d）是线性的。

空间复杂度 = O（nd）

K最近邻居存储数据。测试需要较长的时间，因为您必须将每个测试实例与整个训练数据进行比较。

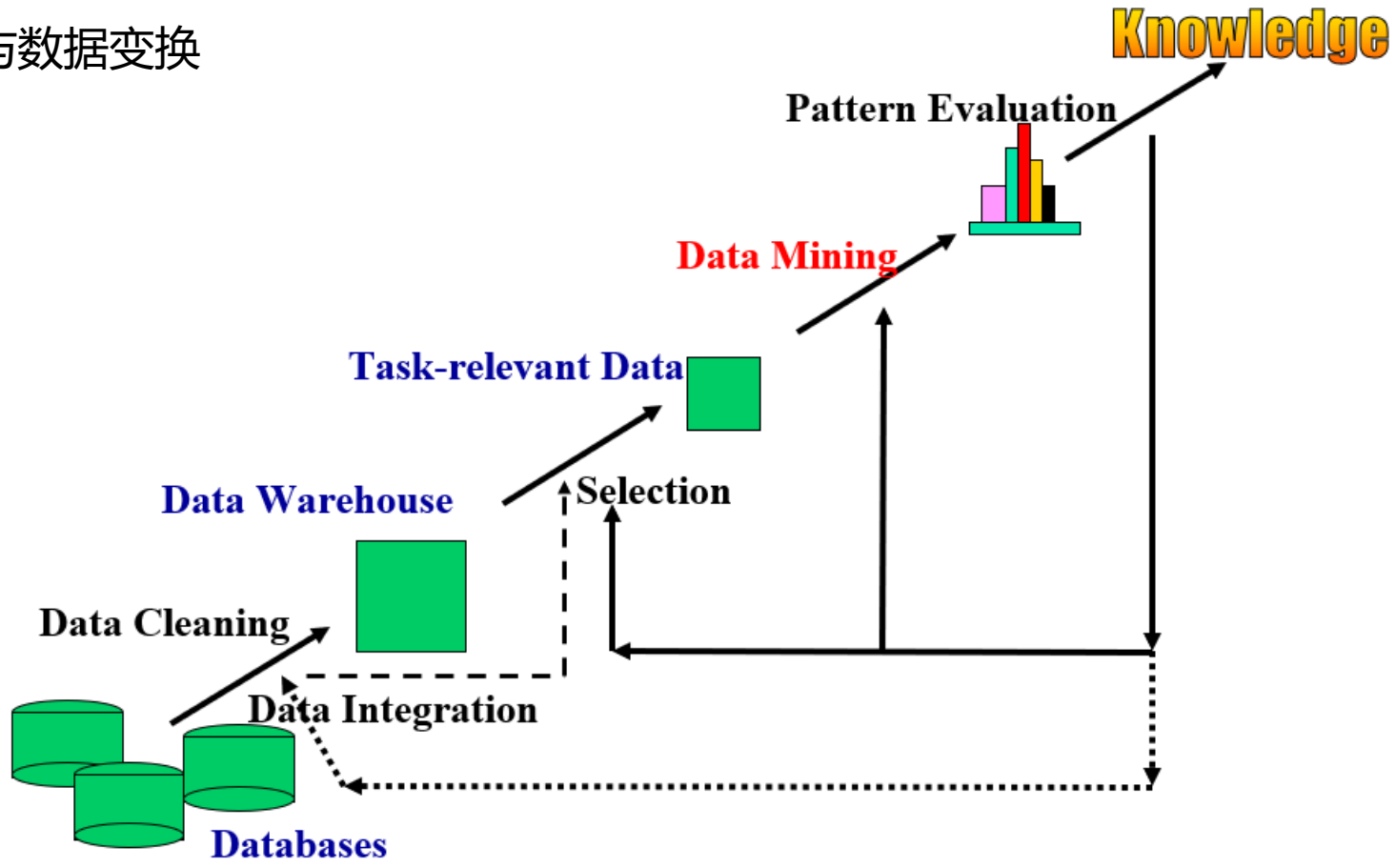
# 《数据挖掘》简答题复习范围

| 考点 | 来源 | |
|---|---|---|
| 1、知识发现过程的主要步骤 | | |
| 2、OLTP与OLAP的主要区别 | | |
| 3、为什么需要构建单独隔离的数据仓库 | | |
| 4、数据预处理内容 | | |
| 5、数据仓库的三种设计模式 | | |
| 6、三种度量函数的定义 | | |
| 7、评估分类器准确率的方法 | | |
| 8、神经网络的结构设计 | | |
| 9、深度学习训练算法及正则化技术 | | |
| 10、常见的深度网络类型 | | |
| 11、自编码器 | | |
| 12、没有免费的午餐定理 | | |
| 13、文本摘要基本概念 | | |
| 14、链接分析算法主要特点 | | |
| 15、传播最大化模型 | | |

# 《数据挖掘》简答题复习范围

1、知识发现过程的主要步骤

**创建目标数据集**：数据选择
**数据预处理**：数据清洗、数据约简与数据变换
**构建数据仓库**（可选）
**数据挖掘（知识发现的核心）**
**模式评价与知识表示**
**知识的使用**




1. 数据清理
2. 数据集成
3. 数据选择
4. 数据交换
5. 数据挖掘
6. 模式评估
7. 知识表示

# 《数据挖掘》简答题复习范围

2、OLTP与OLAP的主要区别

| | OLTP(传统关系数据库) | OLAP（数据仓库） |
|---|---|---|
| 用户 | 员工, IT专业人员 | 知识工作者 |
| 功能 | 每天的日常操作 | 决策支持 |
| DB设计 | 面向应用+ER | 面向主题+Star |
| 数据 | 当前的，详细的数据 | 历史的, 汇总的, 多维的集成的, 整理过的 |
| 使用 | 重复的 | 特定的 |
| 访问 | 读/写、索引 | 多次扫描 |
| 工作单元 | 短的, 简单的事务处理 | 复杂查询 |
| 记录数/查询 | 几十 | 百万 |
| 用户数 | 上千 | 百 |
| DB规模 | 100MB-GB | 100GB-TB |
| metric | transaction throughput | query throughput, response |

2、OLTP与OLAP的主要区别

1.**用户和系统的面向性：**OLTP系统是面向顾客的，用于办事员、客户和信息技术专业人员的事务和查询处理。OLAP系统是面向市场的，用于知识工人（包括经理、主管和分析人员）的数据分析。

2.**数据内容：**OLTP系统管理当前数据。通常，这种数据太琐碎，难以用于决策、OLAP系统管理大量历史数据，提供汇总和聚集机制，并在不同的粒度级别上存储和管理信息。这些特点使得数据更容易用于见多识广的决策。

3.**数据库设计：**通常，OLTP系统采用实体-联系（ER）数据模型和面向应用的数据库设计而OLAP系统通常采用星形或雪花模型和面向主题的数据库设计。

4.**视图：**OLTP系统主要关注企业或部门内部的当前数据，而不涉及历史数据或不同组织的数据。相比之下，由于组织的变化，OLAP系统常常跨越数据库模式的多个版本。OLAP系统还处理来自不同组织的信息，由多个数据存储集成的信息。由于数据量巨大，OLAP数据存放在多个存储介质上。

5.**访问模式：**OLTP系统的访问模式主要由短的原子事务组成。这种系统需要并发控制和恢复机制。然而，对OLAP系统的访问大部分是只读操作（由于大部分数据仓库存放历史数据而不是当前数据）。尽管许多可能是复杂的查询。

3、为什么需要构建单独隔离的数据仓库

**使得操作数据库与数据仓库都获得高性能**
DBMS—OLTP：访问方法，索引，并发控制，数据恢复。
Warehouse—OLAP：复杂OLAP查询，多维视图，整理。

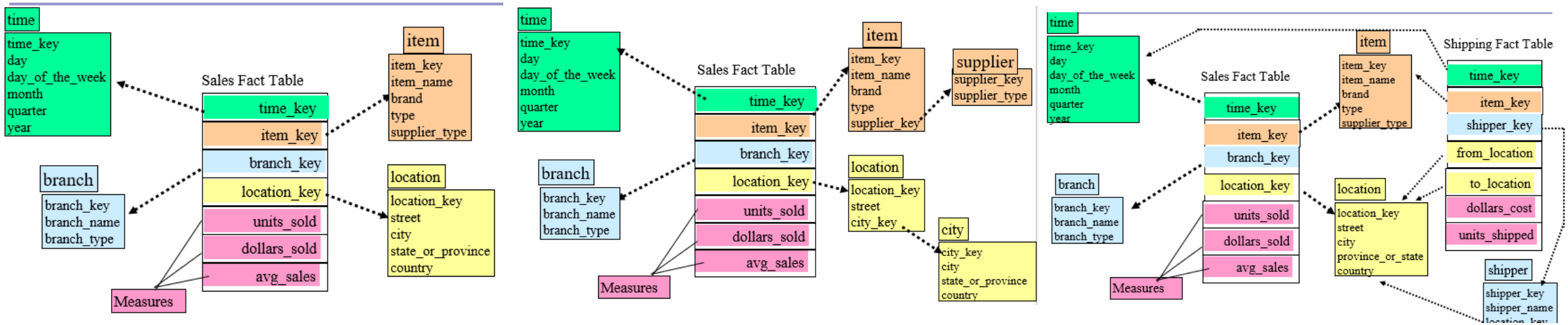
**对数据与功能的要求不同：**
丢失的数据：决策支持需要历史数据，而传统数据库并不一定维护历史数据。
数据整理：　决策支持需要对异构数据源进行数据整理　。
数据质量：不同的数据源常常具有不一致的数据表示，编码结构与格式。

4、数据预处理内容

**数据清洗（Data cleaning）**
填充遗失的数据, 平滑噪声数据, 辨识或删除孤立点, 解决不一致性问题
**数据集成（Data integration）**
对多个数据库，数据立方或文件进行集成
**数据变换（Data transformation）**
规范化与聚集（Normalization and aggregation）
**数据约简（Data reduction）**
得到数据集的压缩表示，它小的多，但能够产生同样的（或几乎同样的）分析结果
**数据离散化（Data discretization）**
特别对数字值而言非常重要

5、数据仓库的三种设计模式

**星型模式**（Star schema）：一个事实表以及一组与事实表连结的维表。
**雪花模式**（Snowflake schema）： 雪花模式是星型模式的变种，其中某些维表是规范化的。（normalized），因而把数据进一步分解到附加的表中。
**事实星座**（Fact constellations）： 多个事实表分享共同的维表，这种模式可以看作星型模式的集合，因此称为星系模式（galaxy schema）或事实星座。



6、三种度量函数的定义

**分布式的（distributive）**：一个聚集函数是分布的，如果它能以以下分布式进行计算:如果将函数用于n个聚集值得到的结果，与将函数用于所有数据得到的结果一样，则该函数可以用分布式计算。如，count()，sum()，min()，max().

**代数的（algebraic）**：一个函数是代数的，如果它能够由一个具有M个参数的代数函数计算（其中M是一个有界整数），而每个参数都可以用一个分布聚集函数得到。 如， avg()，min_N()，standard_deviation().

**整体的（holistic）**：如果描述它的子聚集所需的存储没有一个常数界，即不存在一个具有M个参数的代数函数进行这一计算（其中M是常数）。如，median()，mode()，rank()。

7、评估分类器准确率的方法

训练集（Training data）、验证集（Validation data）与测试集（Test data）

**划分法**：训练集与测试集
把样本划分成2个独立的数据集合，如，训练集 (2/3)，测试集(1/3)。
适用于大规模的数据样本。

**交叉验证**(Cross-validation)
把数据集合划分成k 个子样本；
使用k-1 个子样本作为训练集，另一个作为测试样本—**k-折交叉验证**。
适用于中等规模的数据。

**留一测试**(Leave One Out，k = n)
适用于小规模数据。

$$\lim_{m\to\infty}(1-\frac{1}{m})^m=\frac{1}{e}\approx 0.368$$

**自助法**（bootstrapping）：
采用有放回采样从样本集D中（|D|=m）采m个样本，记为D’。样本在m次采样中始终未采到的概率：
也即初始样本D中有36.8%的样本未出现在D’。于是我们可用D’用作训练集，D\ D’用作测试集。
因此我们仍有m个训练样本，且有约1/3的未出现在训练集中的测试样本。

# 《数据挖掘》简答题复习范围

8、神经网络的结构设计

**输入层：** 神经元个数=feature维度
**输出层：** 神经元个数=分类类别数
**隐层：** 默认只用一个隐层
如果用多个隐层，则每个隐层的神经元数目都一样
隐层神经元个数越多，分类效果越好，但计算量会增大

- **神经网络的结构**是指它具有多少单元以及它们如何连接：多少层？每层多少节点单元？单元之间如何连接？
- 大多数神经网络被组织成称为**层**的单元组。大多数神经网络结构将这些层布置成**链式结构**，其中每一层都是前一层的函数。在这种结构中，第一层由下式给出：

$$\boldsymbol{h}^{(1)} = g^{(1)}\left(\boldsymbol{W}^{(1)\top}\boldsymbol{x} + \boldsymbol{b}^{(1)}\right), \tag{6.40}$$

- 第二层由

$$\boldsymbol{h}^{(2)} = g^{(2)}\left(\boldsymbol{W}^{(2)\top}\boldsymbol{h}^{(1)} + \boldsymbol{b}^{(2)}\right), \tag{6.41}$$

- 给出，以此类推

具有一层隐藏层的前馈神经网络足以表示任何函数，但是**网络层可能非常大**（不可实现）并且可能无法正确地学习和泛化。

但在很多情况下，使用**更深的模型**能够减少表示期望函数所需的单元的数量，并且可以减少泛化误差。

**激活函数：**
分类输出层用 Softmax
隐藏层 优先使用ReLu，其次tanh

# 《数据挖掘》简答题复习范围

9、深度学习训练算法及正则化技术

批量梯度下降（Batch GD，BGD）
随机梯度下降(Stochastic GD, SGD)
小批量梯度下降（Mini Batch GD，MBGD）

- **批量梯度下降法：**在更新参数时使用所有的样本来进行更新，即计算梯度时其损失函数为：

BGD

$$L(w)=\sum_{i=1}^{n}\sum_{j=1}^{C}(T_j^{(i)}-O_j^{(i)})^2$$

- 其中$n$为训练集的样本数目，$T_j^{(i)}$ 为第$i$个样本的输出向量，$O_j^{(i)}$为第$i$个样本的目标向量。
- 在该方法中，每次更新我们都需要在整个数据集上求出所有的偏导数。因此批量梯度下降法的速度会比较慢，甚至对于较大的、内存无法容纳的数据集，该方法无法使用。
- 同时，批量梯度下降法不能以「在线」的形式更新我们的模型，也就是不能在运行中加入新的样本进行运算。

- **随机梯度下降法SGD：**求梯度时没有用所有的$n$个样本的数据，而是仅仅选取一个样本$i$来进行梯度更新。即计算梯度时其损失函数为：

$$L(w)=\sum_{j=1}^{C}(T_j^{(i)}-O_j^{(i)})^2$$

- 相比BGD，SGD单次迭代的运行速度被大大加快，同时也能够「在线」学习，但同时迭代次数可能大大增加。
- 随机梯度下降法更新值的方差很大，在频繁的更新之下，它的目标函数有着剧烈波动。

- **小批量梯度下降法MBGD：**批量梯度下降法和随机梯度下降法的折衷，也就是对于$n$个样本，我们采用$m$ $(1<m<<n)$个样本来迭代。即计算梯度时其损失函数为：

$$L(w)=\sum_{i=1}^{m}\sum_{j=1}^{C}(T_j^{(i)}-O_j^{(i)})^2$$

- $m$一般可以取几十到几千（根据样本的数据量及硬件资源情况进行调整）。
- MBGD具有如下优点：
  - 降低了更新参数的方差（variance），使得收敛过程更为稳定；
  - 能够利用最新的深度学习程序库中高度优化的矩阵运算器，能够高效地求出每小批数据的梯度。
  - 小批量梯度下降法，通常是我们训练神经网络的首选算法。

## 9、深度学习正则化技术

以增大训练误差为代价来减少测试误差。这些策略统称为正则化(Regularization)。

**硬约束：**添加额外的约束，如参数的约束；

**软约束：**有些向目标函数增加额外项，对应于参数值的；

**提前终止策略**；

**参数绑定和共享**；

Dropout

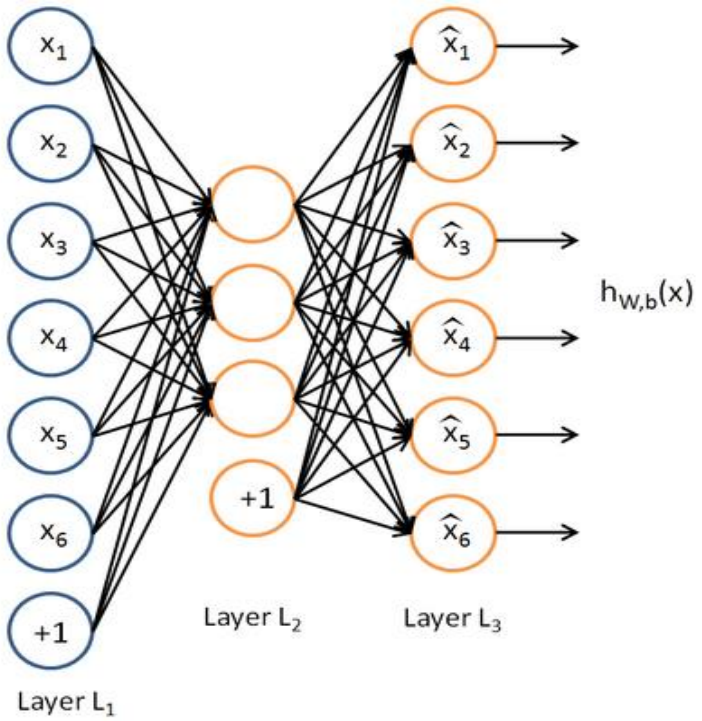
- 深度学习中的正则化
  - 基本定义
    - 正则化是机器学习中抑制过拟合的手段
    - 正则化是一类通过显式设计降低泛化误差，以提升算法通用性的策略的统称
    - 正则化被定义为对学习算法的修改，修改的目的时减小泛化误差
    - 正则化处理可以看作对奥卡姆剃刀原则的应用，正则化得到的是更简单的模型
  - 基于训练数据的正则化
    - 数据集增强
      - 向数据中添加随机的高斯噪声
      - 数据集增强的对象只包括输入层和输出层
      - 通过数据量的增加来减小期望方差和经验方差之间的差别
    - dropout
      - 在训练期间从神经网络随机丢弃神经元及其连接
      - 降低网络的复杂度的同时增强了模型的普适性
      - dropout对训练及容量的要求很高
      - **Dropout可视为集成非常多的大神经网络的实用Bagging方法，但是Dropout是共享参数的**
  - 基于网络架构的正则化
    - 参数共享
      - 强迫某些参数相等或让参数相互接近
      - 常用方法是将一个监督学习模型的参数正则化，令其接近另一个无监督学习的模型，那么这个无监督学习模型就可以匹配监督学习模型的参数
    - 传递函数的正则化　　比如dropout中的maxout单元
  - 基于误差函数/正则化项的正则化
    - 对误差函数进行正则化相当于添加额外的学习任务，即正则项
    - 常用的正则化项：权重衰减项
      正则化项以范数的形式表示
      - 使用L1范数：线性回归->LASSO回归
      - 使用L2范数：线性回归->岭回归
    - L2范数使得在显著减小目标函数方向上的权重保留完好，无益于目标函数减小的方向所对应的分量会因正则化而逐渐被衰减掉
    - L1范数正则化得到的是稀疏的解
  - 基于最优化过程的正则化
    - 对初始化的正则化
    - 对参数更新的正则化
    - 对终止条件的正则化　　早停

10、常见的深度网络类型

**卷积神经网络：**针对网格化数据，如图片
**循环神经网络：**针对序列化数据，如文字、语音
**深度信念网络：**它是一种生成模型，由多个受限玻尔兹曼机组成，采用逐层的方式进行训练
**生成对抗网络：**判别模型的职责是准确区分真实数据和生成数据，而生成模型负责生成符合真实数据概率分布的新数据；通过判别模型和生成模型两个神经网络的对抗训练，生成对抗网络能够有效地生成符合真实数据分布的新数据
**深度强化学习：**与其他机器学习的差异在于，深度强化学习更加注重基于环境的改变而调整自身的行为

11、自编码器

自编码神经网络是一种无监督学习算法，它使用了反向传播算法，并让目标值等于输入值。




**欠完备(under complete)自编码器：**隐藏编码维数小于输入维数。欠完备自编码器可以学习数据分布最显著的特征

**过完备(over-complete)自编码器：**如果隐藏编码的维数允许与输入相等，或隐藏编码维数大于输入的维数。

**栈式自编码算法**

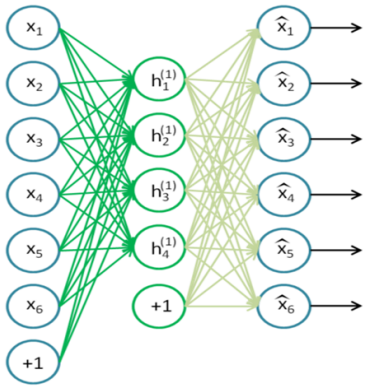
栈式自编码神经网络是一个由多层稀疏自编码器组成的神经网络，其前一层自编码器的输出作为其后一层自编码器的输入。

训练：逐层贪婪训练法依次训练每一层，使用逐层贪婪训练方法将参数训练到快要收敛时再使用反向传播微调

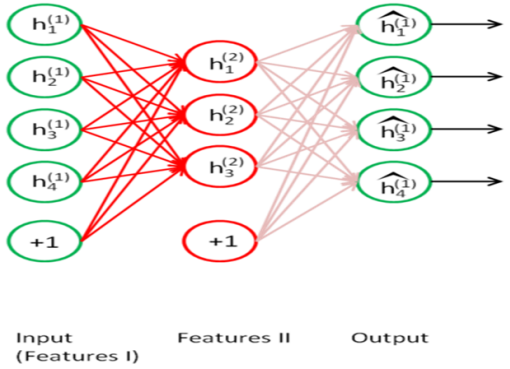
11、自编码器（实例）

训练一个包含两个隐含层的栈式自编码网络，用来进行手写数字分类。

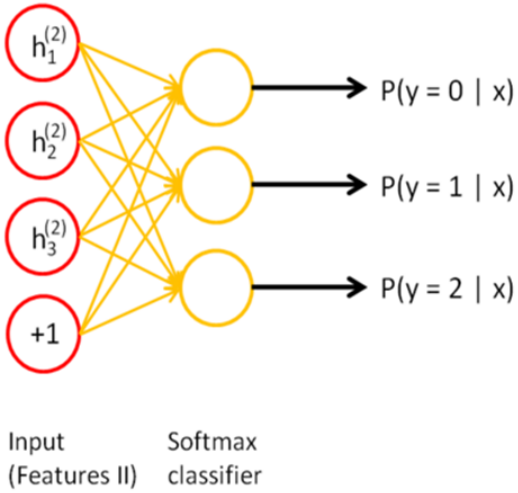
- 首先，用原始输入$x_k$训练第一个自编码器，它能够学习得到原始输入的一阶特征表示$h_k^{(1)}$（如下图所示）。



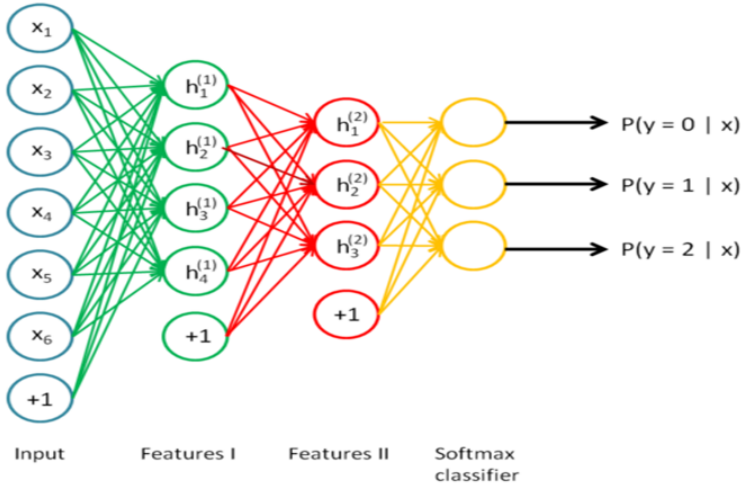

- 接着，把原始数据输入到上述训练好的稀疏自编码器中，对于每一个输入$x_k$，都可以得到它对应的一阶特征表示$h_k^{(1)}$。然后再用这些一阶特征作为另一个稀疏自编码器的输入，使用它们来学习二阶特征$h_k^{(2)}$（如下图所示）。




- 同样，再把一阶特征输入到刚训练好的第二层稀疏自编码器中，得到每个$h_k^{(1)}$对应的二阶特征激活值$h_k^{(2)}$。接下来，你可以把这些二阶特征作为softmax分类器的输入，训练得到一个能将二阶特征映射到数字标签的模型。




- 如下图所示，最终，将这三层结合起来构建一个包含两个隐藏层和一个最终softmax分类器层的栈式自编码网络，这个网络能够进行分类。



12、没有免费的午餐定理

在机器学习中存在一个普适定理—没有免费的午餐(No Free Lunch Theorem，NFL定理)。NFL定理的具体描述为

1）对所有可能的的目标函数求平均，得到的所有学习算法的“非训练集误差”的期望值相同;

2）对任意固定的训练集，对所有的目标函数求平均，得到的所有学习算法的“非训练集误差”的期望值也相同;

3）对所有的先验知识求平均，得到的所有学习算法的“非训练集误差”的期望值也相同;

4）对任意固定的训练集，对所有的先验知识求平均，得到的所有学习算法的的“非训练集误差”的期望值也相同;

NFL定理**表明没有一个学习算法**（包括现在如日中天的深度学习）**可以在任何数据集上总是产生最准确的分类器。**

13、文本摘要基本概念

**概念：**从大量的文本中提取最重要的文本信息

**文本摘要算法的基本要求：**
返回的结果应该具有高度的内容相关性；
返回的结果应该具有低的内容冗余性；
摘要往往具有长度限制。

**从是否需要重写原有语句方面可以进行分类：**
抽取式摘要(Extractive)：从文本中选择最重要的一些句子，无需重写。
生成式摘要(Abstractive)：采取文本重写技术，对一些语句进行压缩、合并等处理。
**从涉及的文档数量方面进行分类：**
单文本摘要(Single Document)：针对单个文本
多文本摘要(Multi-Document)：涉及多个文本
**从涉及查询方面进行分类：**
通用性摘要(Generic)：无查询词
基于查询的摘要(Query-focused)：提供查询词

14、链接分析算法主要特点

两个影响最大的基于链接分析的搜索算法PageRank与HITS

PageRank
优点：防欺骗
网页所有者难以设置其它重要网页指向他自己的网页.
PageRank 值独立于查询，是一种全局度量.
PageRank 值是通过所有网页计算得到并加以存储，而不是提交查询时才计算.
缺点：不能区分全局重要性网页和查询主题重要性网页.

HITS
**优点**：根据查询进行排序，可能会返回更相关的权威性和汇集性网页.
**弱点：**
容易被欺骗. 一个网站开发者很容易在自己的网页中加入许多出边.
主题漂移（Topic drift）. 许多扩展网页可能与主题并不相关.
查询效率：动态抓取网页进行扩展、特征值计算

15、传播最大化模型

将社会网络抽象为一个**有向(或无向)图**G=(V,E),
**节点集合**V代表个人、组织或其他对象
**边集合**E代表对象之间的关系(粉丝、朋友、敌对等).

每个节点有两种状态:
活跃状态(active，如购买了某产品或接受了某观念等)
非活跃状态(inactive，还未购买或未接受).

整个传播过程是不可逆的，即：**一个节点可以从非活跃节点变为活跃节点**，反之则不可.
处于活跃状态的节点对处于非活跃状态的节点存在影响，如果这个影响导致了某个节点非活跃状态变为活跃状态则这个过程称为**激活**

目标：给定一个带权网络G=(V, E)，从V中选取最少的节点集S，使其影响到网络中所有节点；或给定一个带权网络G=(V, E)，一个正整数k，从V中选取最多k个节点，使得影响的节点数最大化；

**独立级联模型**（Independent Cascade Model，简称IC模型）
**线性阈值模型**（Linear Threshold Model，简称LT模型）。

15、传播最大化模型

**独立级联模型**（Independent Cascade Model，简称IC模型）是一种概率模型，当一个节点v被激活时，它会以概率p[v,w]对它未激活的出边邻居节点w尝试激活，这种尝试仅仅进行一次，而且这些尝试之间是互相独立的，即v对w的激活不会受到其他节点的影响。独立级联模型的信息传播过程为：

（1）给定初始的活跃节点集合S，当在时刻t节点v被激活后，它就获得了一次对它的邻居节点w产生影响的机会，成功的概率为p[v,w]，是随机赋予的系统参数，其自身独立不受其他节点的影响，该值越大，节点w越有可能被影响。

（2）若w有多个邻居节点都是新近被激活的节点，那么这些节点将以任意顺序尝试激活节点w。如果节点v成功激活节点w，那么在t+1时刻，节点w转为活跃状态。

（3）在t+1时刻，节点w将对其他节点产生影响，重复上述过程。

需要注意的是，在上述传播过程中，在t时刻无论节点v是否能成功激活它的邻居节点，在以后的时刻，v本身虽然仍保持活跃状态，但它已经不再具备影响力，即在t时刻被激活的节点，已经尝试激活它自身的邻居节点后，在t+1时刻仍然处于活跃状态，但它本身已经不能再去激活其它任何节点，这一类节点称为无影响力的活跃节点。当网络中不存在有影响力的活跃节点时，传播过程结束。

15、传播最大化模型

**线性阈值模型**（Linear Threshold Model，简称LT模型)是一种价值积累模型，它对每个节点v都有一个激活阈值θ[v]∈[0,1]。线性阈值模型的信息传播过程如下：

（1）给定集合中的任意节点v随机分配阈值θ[v]∈[0,1]，该阈值表示这个节点受影响的难易程度，θ[v]越小，表示节点v越容易被影响，θ[v]越大，表示该节点v越难被影响。只有当节点v的新处于激活状态的邻居节点对它的影响力大于该阈值时，节点v才能被激活。
（2）用权值b[w.v]表示节点v被它的邻居节点w的影响，∑w∈in(v) b[w,v]≤1表示节点v的处于活跃状态的邻居节点对它的影响力之和。这里in(v)是v的入边邻居节点集合。
（3）给定初始的活跃节点集合A（网络中其余所有节点均处于非活跃状态），给网络中每个节点任意分配一个阈值，在t时刻，所有在t-1时刻处于活跃状态的节点仍保持活跃，并且当这一时刻节点v的邻居节点的影响力之和大于节点b的阈值时，节点v被激活，即节点v被激活的条件是v已激活的入边邻居对b的积累影响大于v的激活阈值，如下是所述：
∑w∈in(v) ,active(w)≠0 b[w,v]≥θ[v]
（4）节点v被激活后，下一时刻将对它的邻居节点产生影响，重复上述过程。
在LT传播模型中，当网络中已存在的所有活跃节点中任意活跃节点的影响力之和都不能激活他们的处于非活跃状态的邻居节点时，传播过程结束。
它的激活过程时确定的，当我们对一个图用同样的种子节点来激活时，最后的传播范围是完全一样的。

# 《数据挖掘》计算题复习范围

| 考点 | 来源 | |
|---|---|---|
| 一、用Apriori算法挖掘强关联规则 | | |
| 二、频繁模式树的构造 | | |
| 三、信息增益的计算 | | |
| 四、朴素贝叶斯分类 | | |
| 五、k-means聚类 | | |
| 六、凝聚层次聚类法（全链接与单链接） | | |
| 七、神经网络的正反向传播计算（sigmoid、ReLU、softmax激活函数） | | |

# 《数据挖掘》计算题复习范围

## 一、用Apriori算法挖掘强关联规则

基本概念：

**项集**：在关联分析中，包含0个或多个项的集合被称为项集（itemset）。如果一个项集包含k个项，则称它为k-项集。例如:{啤酒，尿布，牛奶，花生} 是一个4-项集。空集是指不包含任何项的项集。

**关联规则**（association rule）：是形如 X → Y 的蕴含表达式，其中X和Y是不相交的项集，即：X∩Y=∅。关联规则的强度可以用它的支持度（support）和置信度（confidence）来度量。

**支持度**：简单地说，X → Y的支持度就是指物品集X和物品集Y同时出现的概率。
规则X → Y的支持度：support(X → Y) = σ(X∪Y) / N

**置信度：**简单地说，可信度就是指在出现了物品集X 的事务T 中，物品集Y 也同时出现的概率有多大。
规则X → Y的置信度：confidence(X → Y) = σ(X∪Y)/σ(X)

**提升度（lift）：**

$$Lift(X->Y)=\frac{P(X,Y)}{P(X)P(Y)}=\frac{P(Y/X)}{P(Y)}=\frac{\text{同时购买}\{X,Y\}\text{的订单数}\times\text{总订单数}}{\text{购买}X\text{的订单数}\times\text{购买}Y\text{的订单数}}$$

提升度>1且越高表明正相关性越高；提升度<1且越低代表负相关性越高；提升度=1表示没有相关性。
通俗的说，提升度反映了“物品集A的出现”对物品集B的出现概率发生了多大的变化。

# 《数据挖掘》计算题复习范围

## 一、用Apriori算法挖掘强关联规则

**主要算法流程：**

1. 扫描数据库，生成候选1项集和频繁1项集。

2. 从2-项集开始循环，由频繁k-1项集生成频繁频繁k项集。
   2.1 频繁k-1项集**两两组合**，判定是否可以连接，若能则连接生成k项集。(连接)
   2.2 对k项集中的每个项集检测其子集是否频繁，舍弃掉子集不是频繁项集即:不在频繁k-1项集中的项集。（先验性质：频繁项集的所有非空子集也一定是频繁的。）
   2.3 扫描数据库，计算2.3步中过滤后的k项集的支持度，舍弃 掉支持度小于阈值的项集，生成频繁k项集。

3. 若当前k项集中只有一个项集时循环结束。

# 《数据挖掘》计算题复习范围

## 一、用Apriori算法挖掘强关联规则（例题1-课堂例题）

事务数据库TDB

| Tid | Items |
|---|---|
| 10 | A, C, D |
| 20 | B, C, E |
| 30 | A, B, C, E |
| 40 | B, E |

1st scan →

$C_1$

| Itemset | sup |
|---|---|
| {A} | 2 |
| {B} | 3 |
| {C} | 3 |
| {D} | 1 |
| {E} | 3 |

→ $L_1$

| Itemset | sup |
|---|---|
| {A} | 2 |
| {B} | 3 |
| {C} | 3 |
| {E} | 3 |

$C_2$

| Itemset |
|---|
| {A, B} |
| {A, C} |
| {A, E} |
| {B, C} |
| {B, E} |
| {C, E} |

2nd scan →

$C_2$

| Itemset | sup |
|---|---|
| {A, B} | 1 |
| {A, C} | 2 |
| {A, E} | 1 |
| {B, C} | 2 |
| {B, E} | 3 |
| {C, E} |  |

→ $L_2$

| Itemset | sup |
|---|---|
| {A, C} | 2 |
| {B, C} | 2 |
| {B, E} | 3 |
| {C, E} | 2 |

$C_3$

| Itemset |
|---|
| {B, C, E} |

3rd scan →

$L_3$

| Itemset | sup |
|---|---|
| {B, C, E} | 2 |

注：关于算法中的连接操作：

- 对于两个$k$-项集能够连接产生一个$k$+1一项集，当且仅当前k项都相等，只有第k+1项不相等。
- 如{A, C, D}与{A, C, E}连接产生{A, C, D, E}；而{A, C, D}与{A, B, D}不能连接。

# 《数据挖掘》计算题复习范围

## 一、用Apriori算法挖掘强关联规则（例2-取自《数据挖掘概念与技术》 ）

TID表示购物车的编号，每行表示购物车中对应的商品列表，商品为i1,i2,i3,i4,i5，D代表整个数据表

| *TID* | 商品 *ID* 的列表 | *TID* | 商品 *ID* 的列表 |
|---|---|---|---|
| T100 | I1，I2，I5 | T600 | I2，I3 |
| T200 | I2，I4 | T700 | I1，I3 |
| T300 | I2，I3 | T800 | I1，I2，I3，I5 |
| T400 | I1，I2，I4 | T900 | I1，I2，I3 |
| T500 | I1，I3 | | |

http://blog.csdn.net/qq1010885678

解：（1）首先扫描整个数据表D，计算每个商品的支持度（出现的次数），得到候选C1表。这里将每个独立的商品都看成一个频繁模式来处理，计算它的支持度

扫描D，对每个候选计数 →

$C_1$

| 项集 | 支持度计数 |
|---|---|
| {I1} | 6 |
| {I2} | 7 |
| {I3} | 6 |
| {I4} | 2 |
| {I5} | 2 |

候选的支持度计数与最小支持度计数比较 →

$L_1$

| 项集 | 支持度计数 |
|---|---|
| {I1} | 6 |
| {I2} | 7 |
| {I3} | 6 |
| {I4} | 2 |
| {I5} | 2 |

（2）将每个商品的支持度和最小支持度作对比（最小支持度为2），小于2的商品将被过滤，得到L1。这里每个商品的支持度都大于2，所以全部保留

# 《数据挖掘》计算题复习范围

## 一、用Apriori算法挖掘强关联规则（例2-取自《数据挖掘概念与技术》 ）

TID表示购物车的编号，每行表示购物车中对应的商品列表，商品为i1,i2,i3,i4,i5，D代表整个数据表

| *TID* | 商品 *ID* 的列表 | *TID* | 商品 *ID* 的列表 |
|---|---|---|---|
| T100 | I1，I2，I5 | T600 | I2，I3 |
| T200 | I2，I4 | T700 | I1，I3 |
| T300 | I2，I3 | T800 | I1，I2，I3，I5 |
| T400 | I1，I2，I4 | T900 | I1，I2，I3 |
| T500 | I1，I3 | | |

http://blog.csdn.net/qq1010885678

由$L_1$产生候选$C_2$ →

$C_2$

| 项集 |
|---|
| {I1, I2} |
| {I1, I3} |
| {I1,I4} |
| {I1, I5} |
| {I2, I3} |
| {I2, I4} |
| {I2, I5} |
| {I3, I4} |
| {I3, I5} |
| {I4, I5} |

扫描$D$，对每个候选计数 →

$C_2$

| 项集 | 支持度计数 |
|---|---|
| {I1, I2} | 4 |
| {I1, I3} | 4 |
| {I1, I4} | 1 |
| {I1, I5} | 2 |
| {I2, I3} | 4 |
| {I2, I4} | 2 |
| {I2, I5} | 2 |
| {I3, I4} | 0 |
| {I3, I5} | 1 |
| {I4, I5} | 0 |

http://blog.csdn.net/qq1010885678

候选的支持度计数与最小支持度计数比较 →

$L_2$

| 项集 | 支持度计数 |
|---|---|
| {I1, I2} | 4 |
| {I1, I3} | 4 |
| {I1, I5} | 2 |
| {I2, I3} | 4 |
| {I2, I4} | 2 |
| {I2, I5} | 2 |

# 《数据挖掘》计算题复习范围

## 一、用Apriori算法挖掘强关联规则（例2-取自《数据挖掘概念与技术》 ）

TID表示购物车的编号，每行表示购物车中对应的商品列表，商品为i1,i2,i3,i4,i5，D代表整个数据表

| *TID* | 商品 *ID* 的列表 | *TID* | 商品 *ID* 的列表 |
|---|---|---|---|
| T100 | I1，I2，I5 | T600 | I2，I3 |
| T200 | I2，I4 | T700 | I1，I3 |
| T300 | I2，I3 | T800 | I1，I2，I3，I5 |
| T400 | I1，I2，I4 | T900 | I1，I2，I3 |
| T500 | I1，I3 | | |

http://blog.csdn.net/qq1010885678

由$L_2$产生候选$C_3$ →

$C_3$

| 项集 |
|---|
| {I1, I2, I3} |
| {I1, I2, I5} |

扫描$D$，对每个候选计数 →

$C_3$

| 项集 | 支持度计数 |
|---|---|
| {I1, I2, I3} | 2 |
| {I1, I2, I5} | 2 |

候选的支持度计数与最小支持度计数比较 →

$L_3$

| 项集 | 支持度计数 |
|---|---|
| {I1, I2, I3} | 2 |
| {I1, I2, I5} | 2 |

图 6.2　候选项集和频繁项集的产生，最小支持计数为 2

# 《数据挖掘》计算题复习范围

## 一、用Apriori算法挖掘强关联规则（例2-取自《数据挖掘概念与技术》 ）

TID表示购物车的编号，每行表示购物车中对应的商品列表，商品为i1,i2,i3,i4,i5，D代表整个数据表

| *TID* | 商品 *ID* 的列表 | *TID* | 商品 *ID* 的列表 |
|---|---|---|---|
| T100 | I1，I2，I5 | T600 | I2，I3 |
| T200 | I2，I4 | T700 | I1，I3 |
| T300 | I2，I3 | T800 | I1，I2，I3，I5 |
| T400 | I1，I2，I4 | T900 | I1，I2，I3 |
| T500 | I1，I3 | | |

http://blog.csdn.net/qq1010885678

我们得到一个频繁模式{i1,i2,i3}，能够提取哪些关联规则？

{i1,i2}=>i3，表示购买了i1，i2的用户中还购买了i3的用户所占的比例。{i1,i2,i3}的出现次数为2，{i1,i2}的出现次数为4，故置信度为2/4=50%

类似的可以算出
{i1,i3}=>i2，confidence=50%
{i2,i3}=>i1，confidence=50%
i1=>{i2,i3}，confidence=33%
i2=>{i1,i3}，confidence=28%
i3=>{i1,i2}，confidence=33%

# 《数据挖掘》计算题复习范围

## 二、频繁模式树的构造（只需要掌握构造，无需用树找出频繁模式）

设项集$I = \{a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p\}$，交易数据库如下表：

| TID | Items Bought |
|---|---|
| 100 | $f, a, c, d, g, i, m, p$ |
| 200 | $a, b, c, f, l, m, o$ |
| 300 | $b, f, h, j, o$ |
| 400 | $b, c, k, s, p$ |
| 500 | $a, f, c, e, l, p, m, n$ |

最小支持度阈值$\xi = 3$。 构造**频繁模式树**只需要扫描（scan）交易数据库**两**次。

每一个交易的**排序频繁项**

| TID | Items Bought | (Ordered) Frequent Items |
|---|---|---|
| 100 | $f, a, c, d, g, i, m, p$ | $f, c, a, m, p$ |
| 200 | $a, b, c, f, l, m, o$ | $f, c, a, b, m$ |
| 300 | $b, f, h, j, o$ | $f, b$ |
| 400 | $b, c, k, s, p$ | $c, b, p$ |
| 500 | $a, f, c, e, l, p, m, n$ | $f, c, a, m, p$ |

**第一次扫描：**

扫描数据库，对其中的每一个项进行计数，得到一个list of frequent items(频繁项的列表），例如，f项出现了4次，依次类推我们对其中的每一项进行计数，因为最小支持度阈值为3，，我们下面只给出出现次数大于3的项：

<(f:4),(c:4),(a:3),(b:3),(m:3),(p:3)）>

**第二次扫描：**

扫描数据库的每一交易，得到每一个交易的**排序频繁项**，构造频繁模式树：

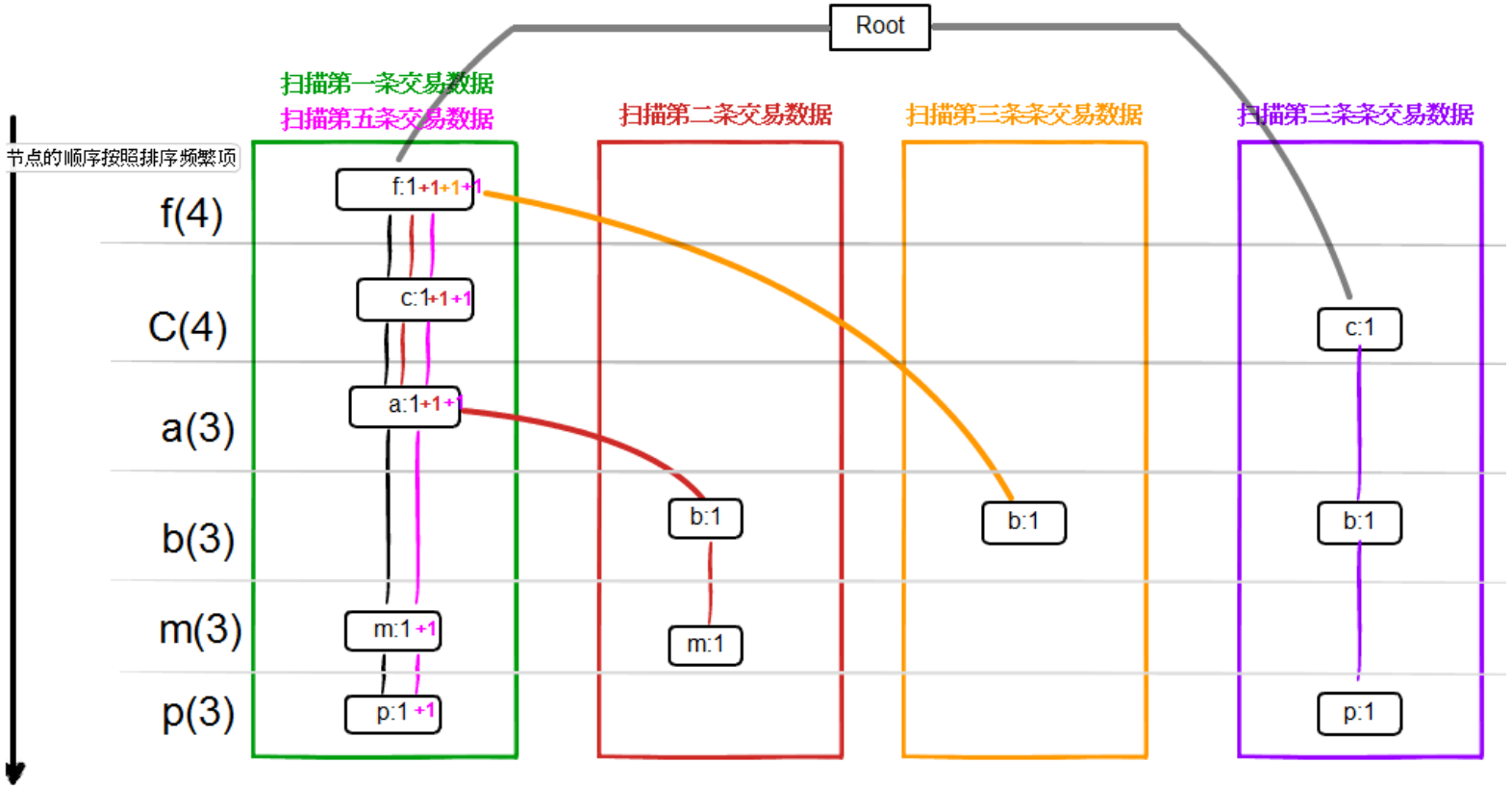
我们对每一个交易，只保留大于3的项，并排序，然后我们得出下表，多出了一列就是**排序频繁项**(Ordered Frequent Items)

# 《数据挖掘》计算题复习范围

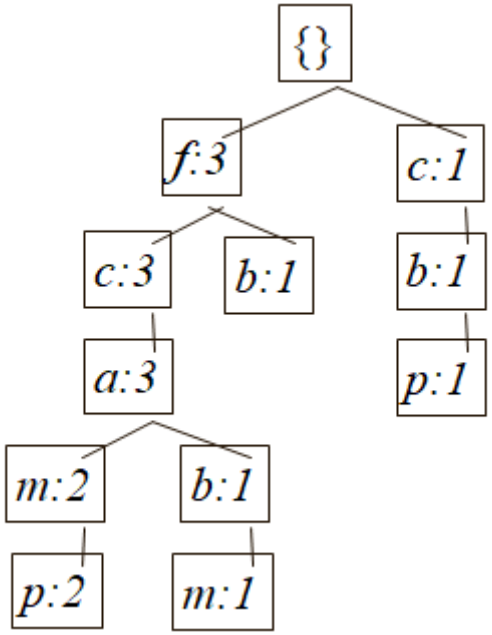
| TID | Items Bought | (Ordered) Frequent Items |
|---|---|---|
| 100 | $f, a, c, d, g, i, m, p$ | $f, c, a, m, p$ |
| 200 | $a, b, c, f, l, m, o$ | $f, c, a, b, m$ |
| 300 | $b, f, h, j, o$ | $f, b$ |
| 400 | $b, c, k, s, p$ | $c, b, p$ |
| 500 | $a, f, c, e, l, p, m, n$ | $f, c, a, m, p$ |

## 二、频繁模式树的构造

根据每一次交易的**排序频繁项**，可以构造出**频繁模式树**




也就对应课件中的**频繁模式树**



# 《数据挖掘》计算题复习范围

## 二、频繁模式树的构造

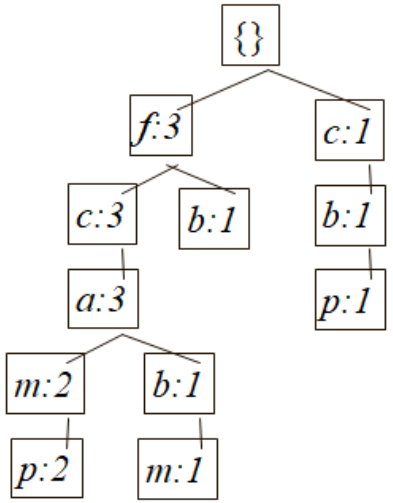
根据上面的两步，我们已经构造出了频繁模式树，怎么样通过频繁模式树，找到频繁模式。

其中，我们拿和项$p$有关的频繁模式举例，其他依次类推：

首先，我们找到所有$p$的节点，并沿着树枝路径向上直到根节点(root)，我们发现有两条路径：

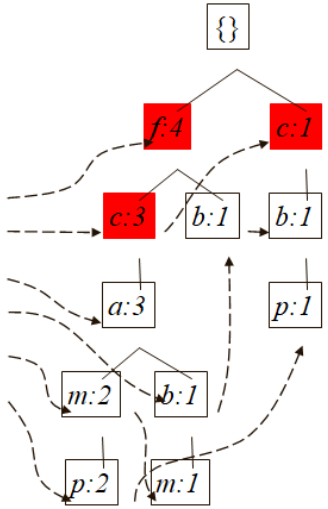
$$<(f:4),(c:3),(a:3),(m:2),(p:2)>\text{和}<(c:1),(b:1),(p:1)>,$$

然后，我们可以得出出现的3次$p$，$c$同时出现了3次，是同时和$p$出现次数最多的项，而且次数大于最小支持度阈值。所以$A=\{c,p\}$就是一个频繁模式，依次类推得出其他项的频繁模式：



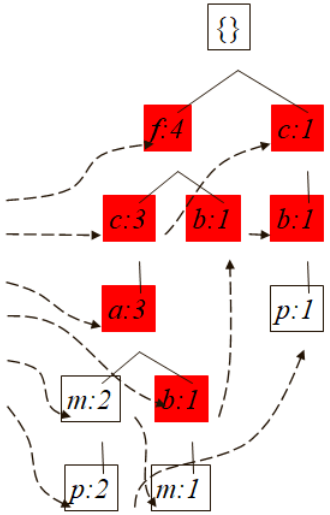

**Header Table**

| Item | frequency | head |
|---|---|---|
| f | 4 | |
| c | 4 | |
| a | 3 | |
| b | 3 | |
| m | 3 | |
| p | 3 | |




**Header Table**

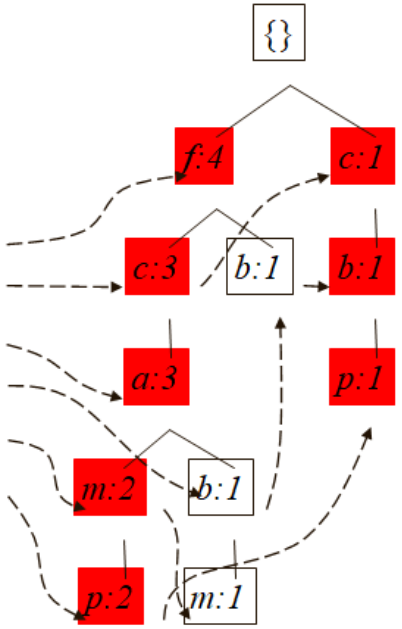
| Item | frequency | head |
|---|---|---|
| f | 4 | |
| c | 4 | |
| a | 3 | |
| b | 3 | |
| m | 3 | |
| p | 3 | |




包含$b$的频繁模式为空

**Header Table**

| Item | frequency | head |
|---|---|---|
| f | 4 | |
| c | 4 | |
| a | 3 | |
| b | 3 | |
| m | 3 | |
| p | 3 | |

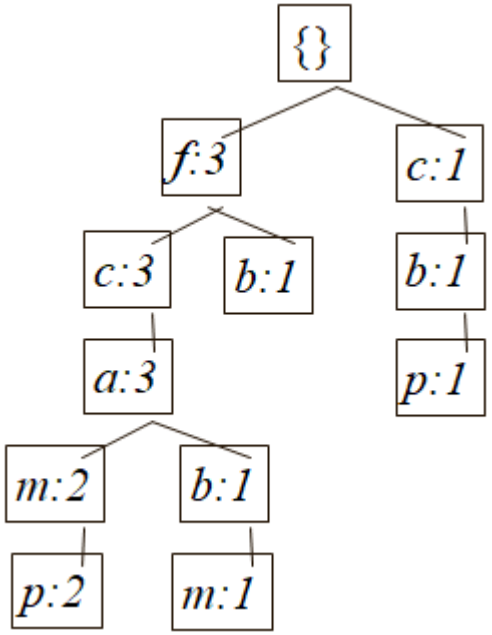


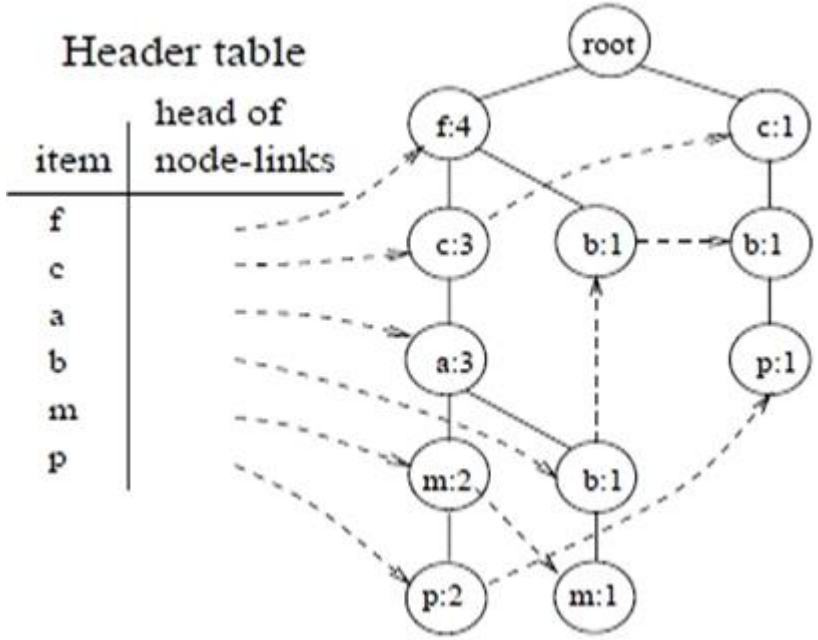

$pc:3$

# 《数据挖掘》计算题复习范围

## 二、频繁模式树的构造

根据构造出**频繁模式树**，可以挖掘频繁模式




包含m的频繁模式

am: 3, cm:3, fm:3

cam:3, fam:3, fcm:3

fcam:3

包含a的频繁模式

ca:3, fa:3

fca:3

包含p的频繁模式

pc：3

包含b的频繁模式

**包含b的频繁模式为空**

包含c的频繁模式

fc：3

根据上面的两步，我们已经构造出了频繁模式树，怎么样通过频繁模式树，找到频繁模式。

其中，我们拿和项$p$有关的频繁模式举例，其他依次类推：

首先，我们找到所有$p$的节点，并沿着树枝路径向上直到根节点(root)，我们发现有两条路径：

$$<(f:4),(c:3),(a:3),(m:2),(p:2)>\text{和}<(c:1),(b:1),(p:1)>,$$

然后，我们可以得出出现的3次$p$，$c$同时出现了3次，是同时和$p$出现次数最多的项，而且次数大于最小支持度阈值。所以$A=\{c,p\}$就是一个频繁模式，依次类推得出其他项的频繁模式：

# 《数据挖掘》计算题复习范围

$$E(X) = -\sum^{n} p_i log_2(p_i)$$

## 三、信息增益的计算（以题目为例（记得带计算器！！学会用计算器算log！！）

| uin | gender | act_info | is_lost |
|---|---|---|---|
| 1 | 男 | 高 | 0 |
| 2 | 女 | 中 | 0 |
| 3 | 男 | 低 | 1 |
| 4 | 女 | 高 | 0 |
| 5 | 男 | 高 | 0 |
| 6 | 男 | 中 | 0 |
| 7 | 男 | 中 | 1 |
| 8 | 女 | 中 | 0 |
| 9 | 女 | 低 | 1 |
| 10 | 女 | 中 | 0 |
| 11 | 女 | 高 | 0 |
| 12 | 男 | 低 | 1 |
| 13 | 女 | 低 | 1 |
| 14 | 男 | 高 | 0 |
| 15 | 男 | 高 | 0 |

| | positive | negative | 汇总 |
|---|---|---|---|
| 整体 | 5 | 10 | 15 |
| 男性 | 3 | 5 | 8 |
| 女性 | 2 | 5 | 7 |
| 高 | 0 | 6 | 6 |
| 中 | 1 | 4 | 5 |
| 低 | 4 | 0 | 4 |

整体熵：（类别熵）

$$\boxed{E(S) = -\frac{5}{15}\log_2(\frac{5}{15}) - \frac{10}{15}\log_2(\frac{10}{15}) = 0.9182}$$

性别熵：

$$E(g_1) = -\frac{3}{8}\log_2(\frac{3}{8}) - \frac{5}{8}\log_2(\frac{5}{8}) = 0.9543$$

$$E(g_2) = -\frac{2}{7}\log_2(\frac{2}{7}) - \frac{5}{7}\log_2(\frac{5}{7}) = 0.8631$$

性别信息增益：

$$IGain(S,g) = \boxed{E(S)} - \frac{8}{15}E(g_1) - \frac{7}{15}E(g_2) = 0.0064$$

整体熵：（类别熵）

$$\boxed{E(S) = -\frac{5}{15}\log_2(\frac{5}{15}) - \frac{10}{15}\log_2(\frac{10}{15}) = 0.9182}$$

同理计算活跃度熵：

$$E(a_1) = 0 \quad E(a_2) = 0.7219 \quad E(a_3) = 0.0$$

活跃度信息增益：

$$IGain(S,a) = \boxed{E(S)} - \frac{6}{15}E(a_1) - \frac{5}{15}E(a_2) - \frac{4}{15}E(a_3) = 0.6776$$

活跃度的信息增益比性别的信息增益大，也就是说，活跃度对用户流失的影响比性别大。在做特征选择或者数据分析的时候，我们应该重点考察活跃度这个指标。

# 《数据挖掘》计算题复习范围

**四、朴素贝叶斯分类**（朴素的原因：朴素贝叶斯算法是假设各个特征之间相互独立） $P(B|A)=\frac{P(A|B)P(B)}{P(A)}$

**朴素有两处优势：第一，下面的式子会成立（成立需要各个特征独立）**

$$p(\text{嫁}|\text{不帅、性格不好、身高矮、不上进})=\frac{p(\text{不帅、性格不好、身高矮、不上进}|\text{嫁})*p(\text{嫁})}{p(\text{不帅、性格不好、身高矮、不上进})}$$

http://blog.csdn.net/zizhen_nlp

**第二：由于满足各个特征独立，可以分开连乘**（*我们就需要在嫁的条件下，去找四种特征全满足分别是不帅，性格不好，身高矮，不上进的人的个数，由于数据的稀疏性，很容易统计到0的情况。 这样是不合适的*）

$$p(\text{嫁}|\text{不帅、性格不好、身高矮、不上进})=\frac{p(\text{不帅、性格不好、身高矮、不上进}|\text{嫁})*p(\text{嫁})}{p(\text{不帅、性格不好、身高矮、不上进})}$$

http://blog.csdn.net/zizhen_nlp

$$=\frac{p(\text{不帅}|\text{嫁})*p(\text{性格不好}|\text{嫁})*p(\text{身高矮}|\text{嫁})*p(\text{不上进}|\text{嫁})*p(\text{嫁})}{p(\text{不帅})*p(\text{性格不好})*p(\text{身高矮})*p(\text{不上进})}$$

# 《数据挖掘》计算题复习范围

## 四、朴素贝叶斯分类

$$p(\text{嫁}\mid\text{不帅、性格不好、身高矮、不上进})=\frac{p(\text{不帅、性格不好、身高矮、不上进}\mid\text{嫁})*p(\text{嫁})}{p(\text{不帅、性格不好、身高矮、不上进})}$$

$$=\frac{p(\text{不帅}\mid\text{嫁})*p(\text{性格不好}\mid\text{嫁})*p(\text{身高矮}\mid\text{嫁})*p(\text{不上进}\mid\text{嫁})*p(\text{嫁})}{p(\text{不帅})*p(\text{性格不好})*p(\text{身高矮})*p(\text{不上进})}$$

http://blog.csdn.net/yizhen_nlp

| 帅？ | 性格好？ | 身高？ | 上进？ | 嫁与否 |
|---|---|---|---|---|
| 帅 | 好 | 矮 | 上进 | 嫁 |
| 不帅 | 好 | 高 | 上进 | 嫁 |
| 帅 | 好 | 高 | 不上进 | 嫁 |
| 不帅 | 好 | 中 | 上进 | 嫁 |
| 帅 | 好 | 中 | 上进 | 嫁 |
| 不帅 | 不好 | 高 | 上进 | 嫁 |

| 帅？ | 性格好？ | 身高？ | 上进？ | 嫁与否 |
|---|---|---|---|---|
| 帅 | 不好 | 矮 | 不上进 | 不嫁 |
| 不帅 | 好 | 矮 | 上进 | 不嫁 |
| 帅 | 好 | 矮 | 上进 | 嫁 |
| 不帅 | 好 | 高 | 上进 | 嫁 |
| 帅 | 不好 | 矮 | 上进 | 不嫁 |
| 不帅 | 不好 | 矮 | 不上进 | 不嫁 |
| 帅 | 好 | 高 | 不上进 | 嫁 |
| 不帅 | 好 | 高 | 上进 | 嫁 |
| 帅 | 好 | 高 | 上进 | 嫁 |
| 不帅 | 不好 | 高 | 上进 | 嫁 |
| 帅 | 好 | 矮 | 不上进 | 不嫁 |
| 帅 | 好 | 矮 | 不上进 | 不嫁 |

$$\frac{3}{6}\quad\frac{1}{6}\quad\frac{1}{6}\quad\frac{1}{6}\quad\frac{6}{12}$$

$$=\frac{p(\text{不帅}\mid\text{嫁})*p(\text{性格不好}\mid\text{嫁})*p(\text{身高矮}\mid\text{嫁})*p(\text{不上进}\mid\text{嫁})*p(\text{嫁})}{p(\text{不帅})*p(\text{性格不好})*p(\text{身高矮})*p(\text{不上进})}$$

$$\frac{4}{12}\quad\frac{4}{12}\quad\frac{7}{12}\quad\frac{4}{12}$$

=0.0535

立即推⇒沉孤生

# 《数据挖掘》计算题复习范围

## 四、朴素贝叶斯分类

$$p(\text{不嫁}|\text{不帅、性格不好、身高矮、不上进})=\frac{p(\text{不帅、性格不好、身高矮、不上进}|\text{不嫁})*p(\text{不嫁})}{p(\text{不帅、性格不好、身高矮、不上进})}$$

$$=\frac{p(\text{不帅}|\text{不嫁})*p(\text{性格不好}|\text{不嫁})*p(\text{身高矮}|\text{不嫁})*p(\text{不上进}|\text{不嫁})*p(\text{不嫁})}{p(\text{不帅})*p(\text{性格不好})*p(\text{身高矮})*p(\text{不上进})}$$

http://blog.csdn.net/yizhen_nlp

| 帅？ | 性格好？ | 身高？ | 上进？ | 嫁与否 |
|---|---|---|---|---|
| 帅 | 不好 | 矮 | 不上进 | 不嫁 |
| 不帅 | 好 | 矮 | 上进 | 不嫁 |
| 帅 | 好 | 矮 | 上进 | 嫁 |
| 不帅 | 好 | 高 | 上进 | 嫁 |
| 帅 | 不好 | 矮 | 上进 | 不嫁 |
| 不帅 | 不好 | 矮 | 不上进 | 不嫁 |
| 帅 | 好 | 高 | 不上进 | 嫁 |
| 不帅 | 好 | 高 | 上进 | 嫁 |
| 帅 | 好 | 高 | 上进 | 嫁 |
| 不帅 | 不好 | 高 | 上进 | 嫁 |
| 帅 | 好 | 矮 | 不上进 | 不嫁 |
| 帅 | 好 | 矮 | 不上进 | 不嫁 |

这题给我们的启发是：
如果男生不帅也不高
一定要性格好而且要上进
不然是没有女生愿意嫁的！

p（不嫁|不帅、性格不好、身高矮、不上进）= ((1/6*1/2*1*1/2)*1/2)/(1/3*1/3*7/12*1/3)
p（嫁|不帅、性格不好、身高矮、不上进）= (1/2*1/6*1/6*1/6*1/2)/(1/3*1/3*7/12*1/3)
很显然(1/6*1/2*1*1/2) > (1/2*1/6*1/6*1/6*1/2)
于是有p（不嫁|不帅、性格不好、身高矮、不上进)>p（嫁|不帅、性格不好、身高矮、不上进)
所以我们根据朴素贝叶斯算法可以给这个女生答案，是不嫁！！！！

# 《数据挖掘》计算题复习范围

四、朴素贝叶斯分类（课件例题-对于如何分类理解还不够透彻，再多看看）

**类:**

**$c_1$:buys_computer='yes'**

**$c_2$:buys_computer='no'**

**数据样本**

**x =(age<=30,Income=medium，Student=yes，Credit_rating=Fair)**

| age | income | student | credit_rating | buys_computer |
|---|---|---|---|---|
| <=30 | high | no | fair | no |
| <=30 | high | no | excellent | no |
| 31…40 | high | no | fair | yes |
| >40 | medium | no | fair | yes |
| >40 | low | yes | fair | yes |
| >40 | low | yes | excellent | no |
| 31…40 | low | yes | excellent | yes |
| <=30 | medium | no | fair | no |
| <=30 | low | yes | fair | yes |
| >40 | medium | yes | fair | yes |
| <=30 | medium | yes | excellent | yes |
| 31…40 | medium | no | excellent | yes |
| 31…40 | high | yes | fair | yes |
| >40 | medium | no | excellent | no |

- 对每一个类，计算$P(\mathbf{x}|c_i)$

  P(age="<30" | buys_computer="yes") = 2/9=0.222

  P(age="<30" | buys_computer="no") = 3/5 =0.6

  P(income="medium" | buys_computer="yes")= 4/9 =0.444

  P(income="medium" | buys_computer="no") = 2/5 = 0.4

  P(student="yes" | buys_computer="yes)= 6/9 =0.667

  P(student="yes" | buys_computer="no")= 1/5=0.2

  P(credit_rating="fair" | buys_computer="yes")=6/9=0.667

  P(credit_rating="fair" | buys_computer="no")=2/5=0.4

- **x = (age<=30 ,income =medium, student=yes,credit_rating=fair)**

  $P(\mathbf{x}|c_1)$ : P(**x**|buys_computer="yes")= 0.222 × 0.444 × 0.667 × 0.0.667 =0.044

  P(**x**|buys_computer="no")= 0.6 × 0.4 × 0.2 × 0.4 =0.019

  $P(\mathbf{x}|c_2) \times P(c_2)$ : P(**x** | yes) × P(yes) = 0.028

  P(**x** | no) × P(no) = 0.007

- 所以：**x** 属于类 **"buys_computer=yes"**

## 五、k-means聚类

k-means算法如下：

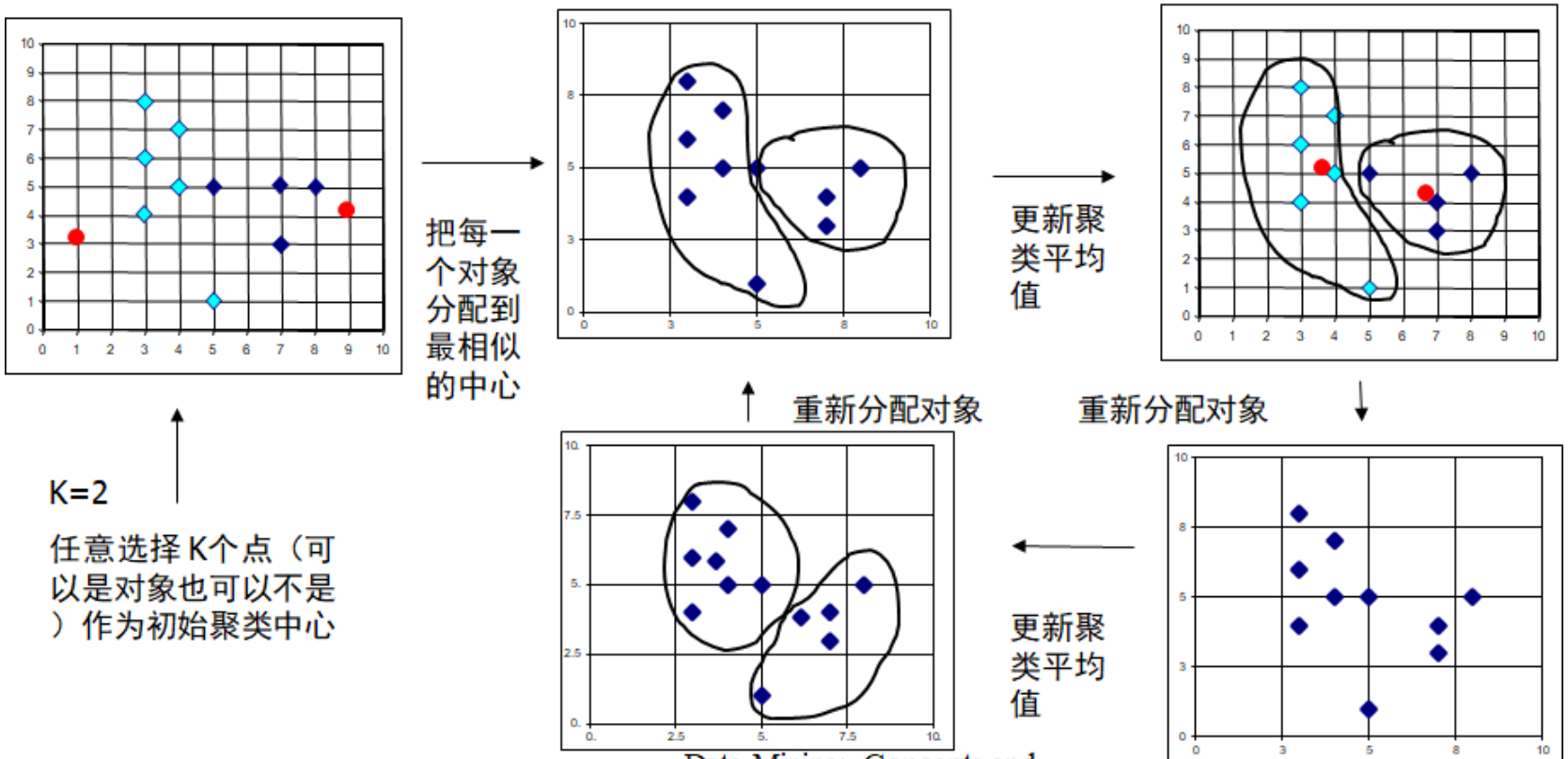
1. 把对象划分成k 个非空子集；
2. 计算当前划分的每个聚类的中心；
3. 把每一个对象分配到离它最近的中心；
4. 返回到第2步，当满足某种停止条件时停止。

停止条件

①当分配不再发生变化时停止；

②当前后两次迭代的目标函数值小于某一给定的阈值时；

③当达到给定的迭代次数时。




Data Mining: Concepts and Techniques

# 《数据挖掘》计算题复习范围

五、k-means聚类（课堂例题）

- 对象如下, $k$=2

| 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|
| (12,8) | (7,9) | (13,11) | (23,10) | (18,23) | (20,18) |

- 步骤 1, 任意选择两个对象初始中心，如2和4。
- 步骤2, 分配剩下的对象。

| No | 2 | 4 |
|---|---|---|
| 1 | 25 | 125 |
| 2 | | |
| 3 | 40 | 101 |
| 4 | | |
| 5 | 317 | 194 |
| 6 | 250 | 73 |

- 因此, 有2个聚类:{1, 2, 3}和{4, 5, 6}，两个聚类内部每个对象与对应的聚类中心的平方误差和为(略)
- 步骤3, 计算每个聚类的中心
  - Cluster 1: $m_1$=((12+7+13)/3, (8+9+11)/3)=(10.67, 9.33)
  - Cluster 2: $m_2$=((23+18+20)/3, (10+23+18)/3)=(20.33, 17)
- 步骤4, 重新分配对象（停止）。

| No | M1(10.67, 9.33) | M2 (20.33, 17) |
|---|---|---|
| 1 | 3.54 | 150.31 |
| 2 | 13.76 | 241.8 |
| 3 | 8.24 | 89.68 |
| 4 | 149.82 | 56.1 |
| 5 | 240.56 | 41.47 |
| 6 | 162.31 | 2.82 |

# 《数据挖掘》计算题复习范围

## 五、k-means聚类（再练一题）

| 像元 | 波段 | |
|---|---|---|
| | B1 | B2 |
| A | 10 | 30 |
| B | 8 | 32 |
| C | 22 | 18 |
| D | 30 | 10 |
| E | 32 | 12 |

（1）确定类别数为2（ABC，DE），计算两个类的中心坐标

| | 类中心坐标 | |
|---|---|---|
| 类 | B1 | B2 |
| ABC | 13 | 27 |
| DE | 31 | 11 |

（2）计算每个像元到类中心的欧氏距离，并将每个像元重新分配给最近的一类。若类中像元

A到两个类的平均距离

$D^2$（A，ABC）=（10-13）$^2$+（30-27）$^2$=18

$D^2$（A，DE）=（10-31）$^2$+（30-11）$^2$=802

B到两个类的平均距离

$D^2$（B，ABC）=（8-13）$^2$+（32-27）$^2$=50

$D^2$（B，DE）=（8-31）$^2$+（32-11）$^2$=970

D到两个类的平均距离

$D^2$（D，ABC）=（30-13）$^2$+（10-27）$^2$=578

$D^2$（D，DE）=（30-31）$^2$+（10-11）$^2$=2

E到两个类的平均距离

$D^2$（E，ABC）=（32-13）$^2$+（12-27）$^2$=586

$D^2$（E，DE）=（32-31）$^2$+（12-11）$^2$=2

结论1：C应重新分配到DE所在类

五、k-means聚类（再练一题）

重复步骤1、2；

| | 新的类中心坐标 | |
|---|---|---|
| 类 | B1 | B2 |
| AB | 9 | 31 |
| CDE | 28 | 13 |

A到两个类的平均距离

$D^2$（A，AB）=（10-9）$^2$+（30-31）$^2$=2

$D^2$（A，CDE）=（10-28）$^2$+（30-13）$^2$=613

D到两个类的平均距离

$D^2$（D，AB）=（30-9）$^2$+（10-31）$^2$=882

$D^2$（D，CDE）=（30-28）$^2$+（10-13）$^2$=13

结论：所有像元不再重新分类

各类平均距离汇总

B到两个类的平均距离

$D^2$（B，AB）=（8-9）$^2$+（32-31）$^2$=2

$D^2$（B，CDE）=（8-28）$^2$+（32-13）$^2$=761

E到两个类的平均距离

$D^2$（E，AB）=（32-9）$^2$+（12-31）$^2$=890

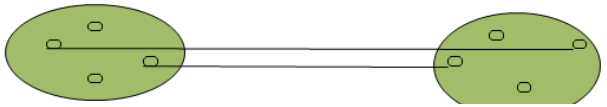
$D^2$（E，CDE）=（32-28）$^2$+（12-13）$^2$=17

# 《数据挖掘》计算题复习范围

## 六、凝聚层次聚类法（全链接与单链接）

### 层次聚类法：聚类之间的距离

- 最短与最长距离：
  - 若定义两个聚类之间的距离为二者对象之间的**最小**距离，则该算法也称为**单链接**算法(Single-Linkage Algorithm，SLA)，也称为最小生成树算法。
  - 若定义两个聚类之间的距离为二者对象之间的**最大**距离，则该算法也称为**全链接**算法(Complete-Linkage Algorithm，CLA)。



- SLA与最小生成树的关系：最大分离度一定等于最小生成树中某条边的值。
- 给定5个对象间的距离如下表

| No | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| 1 | 0 | | | | |
| 2 | 6 | 0 | | | |
| 3 | 2 | 4 | 0 | | |
| 4 | 3 | 4 | 5 | 0 | |
| 5 | 7 | 1 | 5 | 5 | 0 |

- 步骤1: 每个对象当做一个聚类.
- 步骤 2: 找出上述5个聚类中最近的两个聚类2和5，因为它们的距离最小: $d_{25}=1$. 所以, 2和5凝聚成一个新的聚类{2, 5}.
- 步骤3. 计算聚类{2, 5}与聚类 {1}, {3}, {4}的距离
- $d_{\{2,5\}1}=\min\{d_{21}, d_{51}\}=\min\{6,7\}=6$
- $d_{\{2,5\}3}=\min\{d_{23}, d_{53}\}=\min\{4,5\}=4$
- $d_{\{2,5\}4}=\min\{d_{24}, d_{54}\}=\min\{4,5\}=4$
  - 4个聚类 {2,5}, {1}, {3}, {4}中最近的2个聚类是 {1}和{3}. 因此, 1和3凝聚成一个新的聚类. 现在, 我们有3个聚类: {1,3}, {2,5}, {4}.
- 步骤4. 计算聚类 {1,3}与 {2,5}, {4}之间的距离
  - $d_{\{1,3\}\{2,5\}}=\min\{d_{1\{2,5\}}, d_{3\{2,5\}}\}=\min\{6,4\}=4$
  - $d_{\{1,3\}4}=\min\{d_{14}, d_{34}\}=\min\{3,5\}=3$
  - 因此, 聚类{1,3}和 {4}凝聚成一个新的聚类{1,3,4}.
  - 现在, 我们得到2个聚类{1,3,4}和{2,5}
- 步骤5. 计算{1, 3,4}的{2,5}聚类
  - $d_{\{2,5\}\{1,3,4\}}=\min\{d_{\{2,5\}\{1,3\}}, d_{\{2,5\}4}\}=\min\{4,4\}=4$
  - 聚类 {1, 3,4}和{2,5}凝聚成一个唯一的聚类{1,2,3,4,5}.

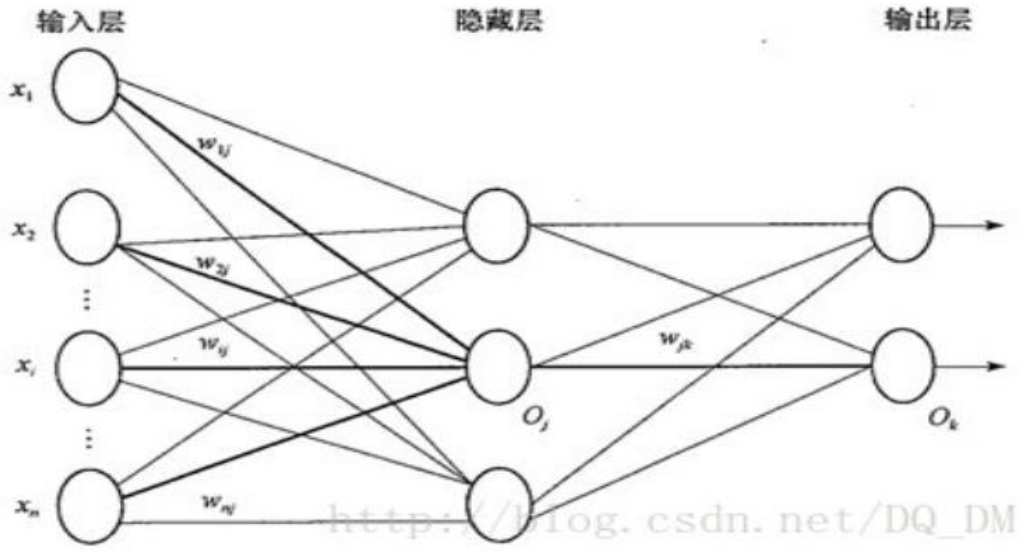
| No | {2,5} | 1 | 3 | 4 |
|---|---|---|---|---|
| {2,5} | 0 | | | |
| 1 | 6 | 0 | | |
| 3 | 4 | 2 | 0 | |
| 4 | 4 | 3 | 5 | 0 |

| No | {2,5} | {1,3} | 4 |
|---|---|---|---|
| {2,5} | 0 | | |
| {1,3} | 4 | 0 | |
| 4 | 4 | 3 | 0 |

| No | {2,5} | {1,3,4} |
|---|---|---|
| {2,5} | 0 | |
| {1,3,4} | 4 | 0 |

# 《数据挖掘》计算题复习范围

## 七、神经网络的正反向传播计算（sigmoid、ReLU、softmax激活函数）




**Softmax**

**数学公式：**

$$\sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}}$$

Softmax用于多分类神经网络输出，目的是让大的更大。函数公式是

所以 Softmax 函数的导数如下面所示：

$$\frac{\partial p_j}{\partial a_j} = \begin{cases} p_i(1-p_j) & if \quad i=j \\ -p_j \cdot p_i & if \quad i \neq j \end{cases}$$

$$\frac{\partial L}{\partial o_i} = -y_i(1-p_i) - \sum_{k \neq i} y_k \frac{1}{p_k}(-p_k \cdot p_i)$$

$$\frac{\partial L}{\partial o_i} = p_i - y_i$$

**Sigmoid**

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

求导：y'=y（1-y）

**ReLU**

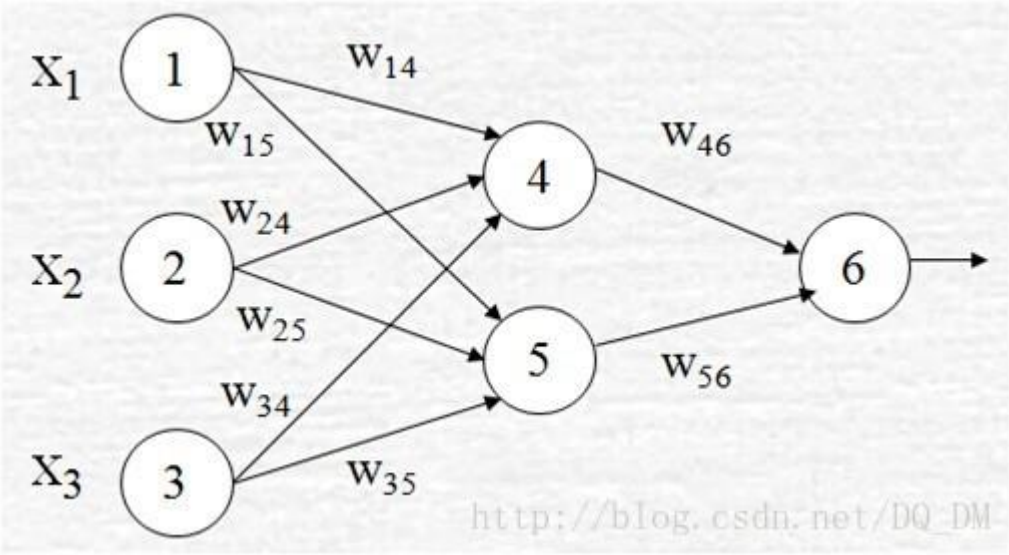
$$f(x) = max(0, x)$$

求导：y'=0或1

$$\frac{\partial L}{\partial o_i} = O_j(1-O_j)(T_j - O_j)$$

# 《数据挖掘》计算题复习范围

## 七、神经网络的正反向传播计算（练一道题）




$$\frac{\partial L}{\partial o_i} = O_j(1-O_j)(T_j-O_j)$$

**向后传播误差**：通过更新权重和反映网络预测误差的偏倚，向后传播误差。

（1）对于输出层单元j, 误差$Err_j$用下式计算

$$Err_j = O_j(1-O_j)(T_j-O_j)$$

其中，$O_j$是单元j的实际输出，而 $T_j$是j给定训练元组的已知目标值。

（2）隐藏层单元j的误差是

$$Err_j = O_j(1-O_j)\sum_k Err_k w_{jk}$$

其中, $w_{kj}$是由下一较高层中单元k到单元j的连接权, 而$Err_k$是单元k的误差。

（3）更新权重和偏倚，以反映传播的误差。

权重由下式更新：

$$\Delta w_{ij} = (l)\,Err_j O_i$$

$$w_{ij} = w_{ij} + \Delta w_{ij}$$

其中,$Dw_{ij}$是权$w_{ij}$的改变量， 变量$l$是学习率,通常取0.0和1.0之间的常数值。

# 《数据挖掘》计算题复习范围

$$\frac{\partial L}{\partial o_i} = O_j(1-O_j)(T_j-O_j)$$

七、神经网络的正反向传播计算（练一道题）

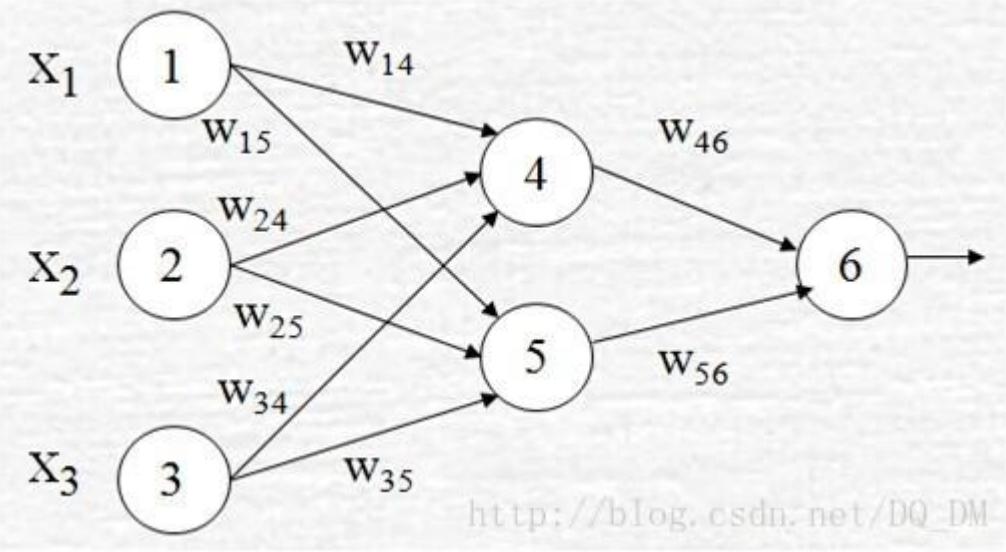



表1 初始输入、权值和偏倚值

| $x_1$ | $x_2$ | $x_3$ | $w_{14}$ | $w_{15}$ | $w_{24}$ | $w_{25}$ | $w_{34}$ | $w_{35}$ | $w_{46}$ | $w_{56}$ | $\theta_4$ | $\theta_5$ | $\theta_6$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 1 | 0.2 | -0.3 | 0.4 | 0.1 | -0.5 | 0.2 | -0.3 | -0.2 | -0.4 | 0.2 | 0.1 |

表2 净输入和输出的计算表

| 单元j | 净输入$I_j$ | 输出$O_j$ |
|---|---|---|
| 4 | 0.2+0-0.5-0.4=-0.7 | $1/(1+e^{0.7})=0.332$ |
| 5 | -0.3+0+0.2+0.2=0.1 | $1/(1+e^{-0.1})=0.525$ |
| 6 | (-0.3)(0.332)-(0.2)(0.525)+0.1=-0.105 | $1/(1+e^{0.105})=0.474$ |

表3 计算每个节点的误差表

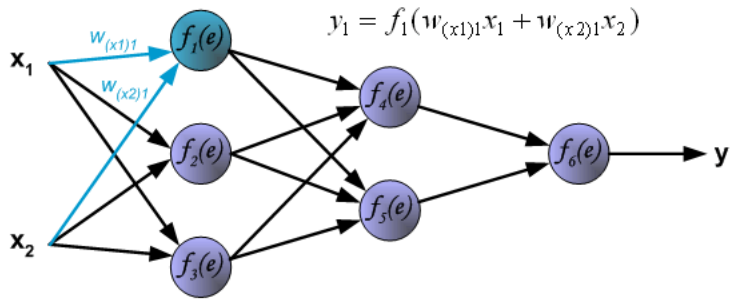
| 单元j | $Err_j$ |
|---|---|
| 6 | (0.474)(1-0.474)(1-0.474)=0.1311 |
| 5 | (0.525)(1-0.525)(0.1311)(-0.2)=-0.0065 |
| 4 | (0.332)(1-0.332)(0.1311)(-0.3)=-0.0087 |

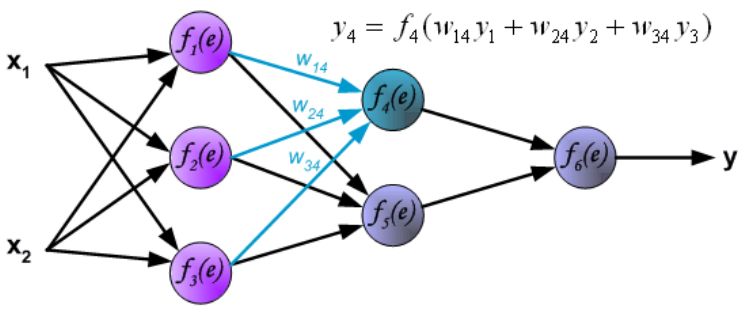
表4 计算权重和偏倚的更新

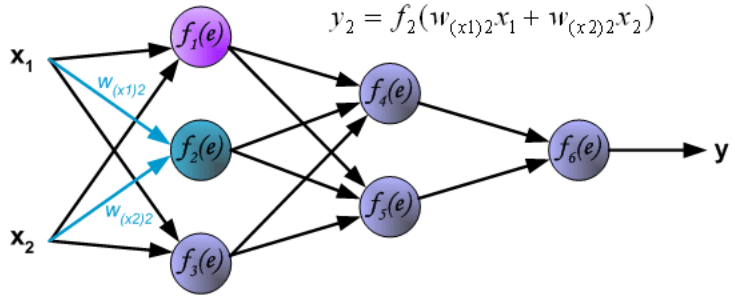
| 权重或偏差 | 新值 |
|---|---|
| $w_{46}$ | $-0.3+(0.9)(0.1311)(0.332)=-0.261$ |
| $w_{56}$ | $-0.2+(0.9)(0.1311)(0.525)=-0.138$ |
| $w_{14}$ | $0.2+(0.9)(-0.0087)(1)=0.192$ |
| $w_{15}$ | $-0.3+(0.9)(-0.0065)(1)=-0.306$ |
| $w_{24}$ | $0.4+(0.9)(-0.0087)(0)=0.4$ |
| $w_{25}$ | $0.1+(0.9)(-0.0065)(0)=0.1$ |
| $w_{34}$ | $-0.5+(0.9)(-0.0087)(1)=-0.508$ |
| $w_{35}$ | $0.2+(0.9)(-0.0065)(1)=0.194$ |
| $\theta_6$ | $0.1+(0.9)(0.1311)=0.218$ |
| $\theta_5$ | $0.2+(0.9)(-0.0065)\approx 0.194$ |
| $\theta_4$ | $-0.4+(0.9)(-0.0087)=-0.408$ |

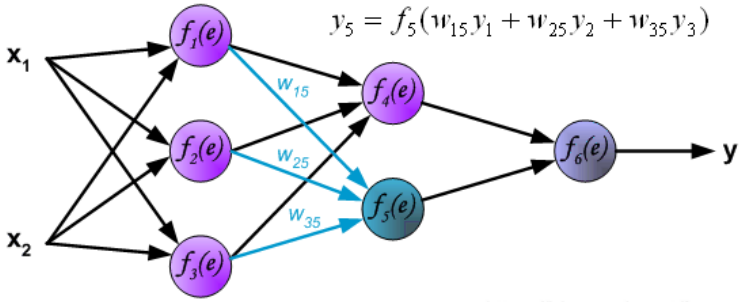
http://blog.csdn.net/DQ_DM

## 七、神经网络的正反向传播计算【正向传播】

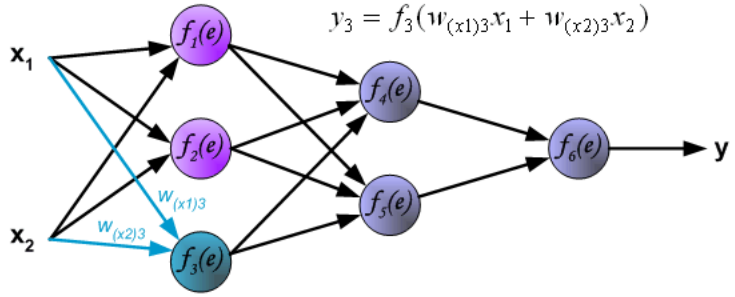


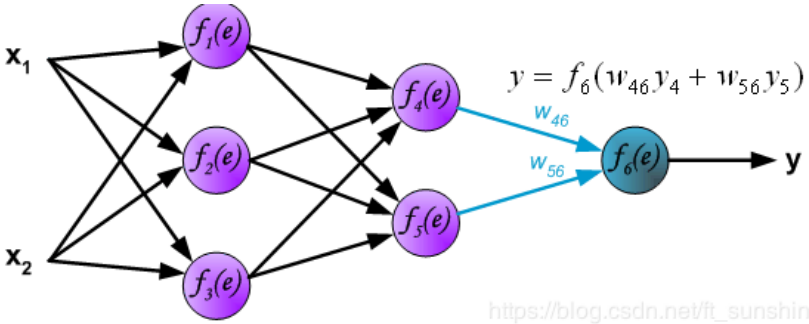




https://blog.csdn.net/ft_sunshine

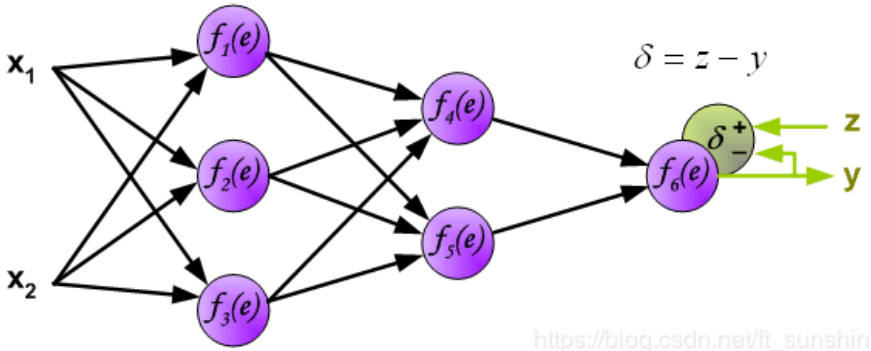


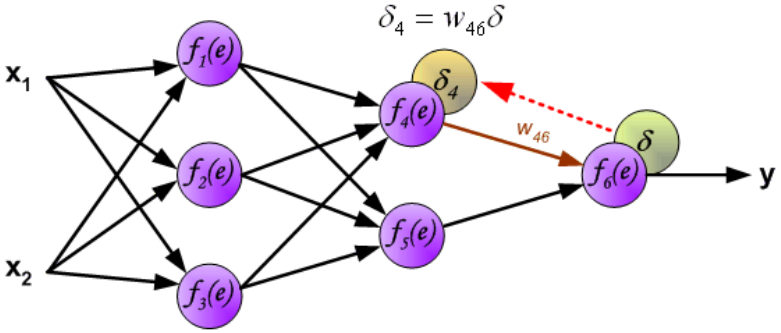

https://blog.csdn.net/ft_sunshine

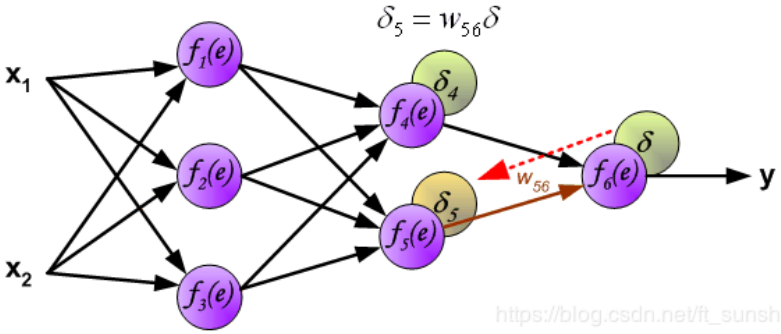
# 《数据挖掘》计算题复习范围

## 七、神经网络的正反向传播计算【反向传播】

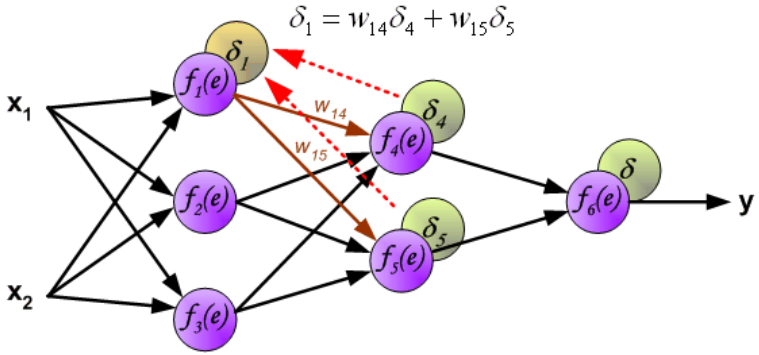


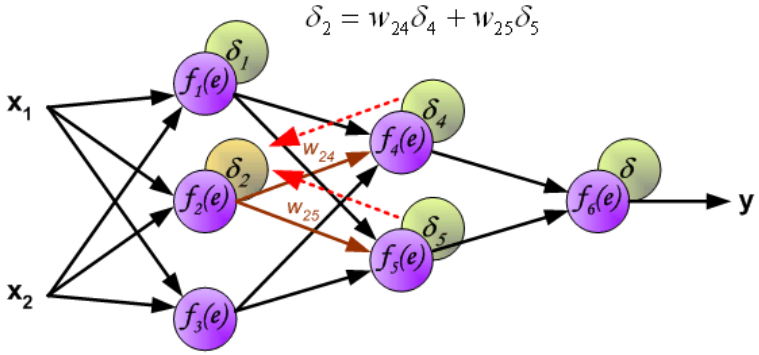

https://blog.csdn.net/ft_sunshine

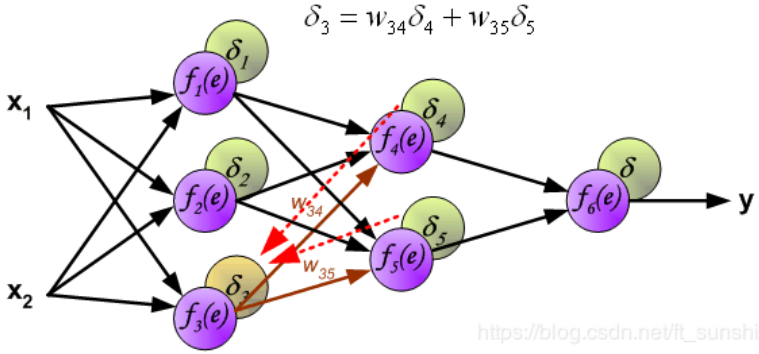



https://blog.csdn.net/ft_sunshine



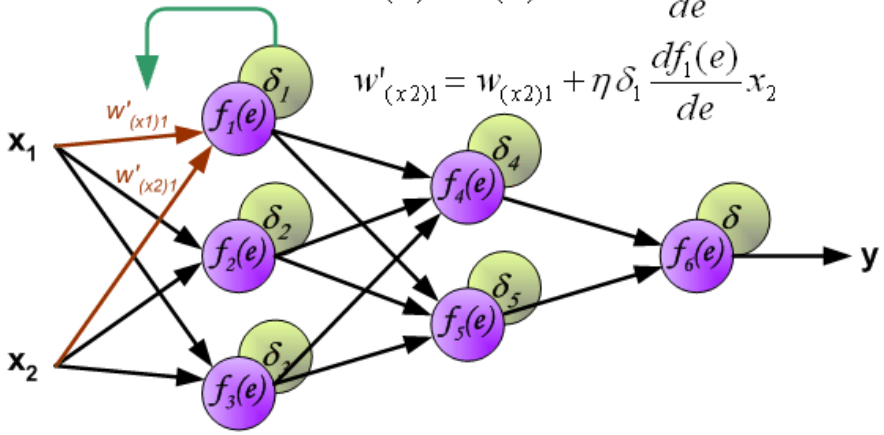

https://blog.csdn.net/ft_sunshine

# 《数据挖掘》计算题复习范围

## 七、神经网络的正反向传播计算【参数修改1】

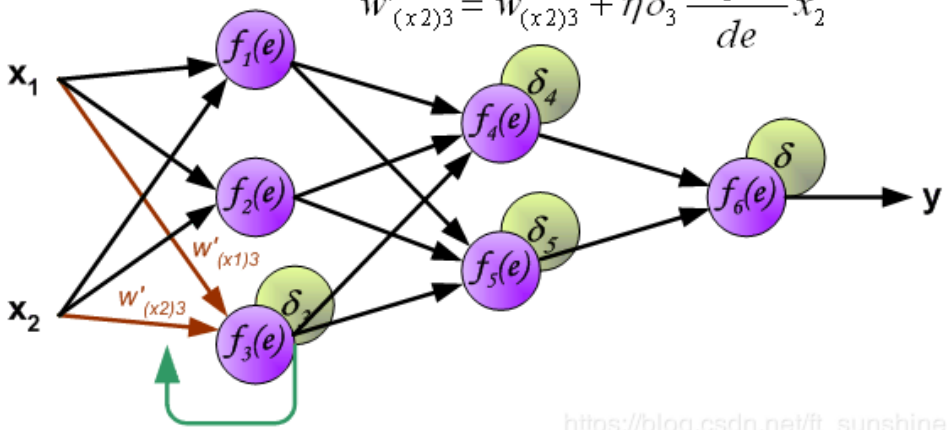
$$w'_{(x1)1} = w_{(x1)1} + \eta\delta_1 \frac{df_1(e)}{de} x_1$$

$$w'_{(x2)1} = w_{(x2)1} + \eta\delta_1 \frac{df_1(e)}{de} x_2$$




$$w'_{(x1)3} = w_{(x1)3} + \eta\delta_3 \frac{df_3(e)}{de} x_1$$

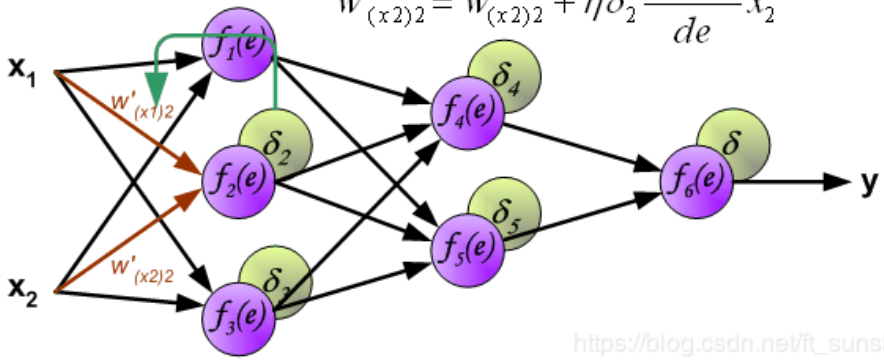
$$w'_{(x2)3} = w_{(x2)3} + \eta\delta_3 \frac{df_3(e)}{de} x_2$$




https://blog.csdn.net/ft_sunshine

$$w'_{(x1)2} = w_{(x1)2} + \eta\delta_2 \frac{df_2(e)}{de} x_1$$

$$w'_{(x2)2} = w_{(x2)2} + \eta\delta_2 \frac{df_2(e)}{de} x_2$$




https://blog.csdn.net/ft_sunshine

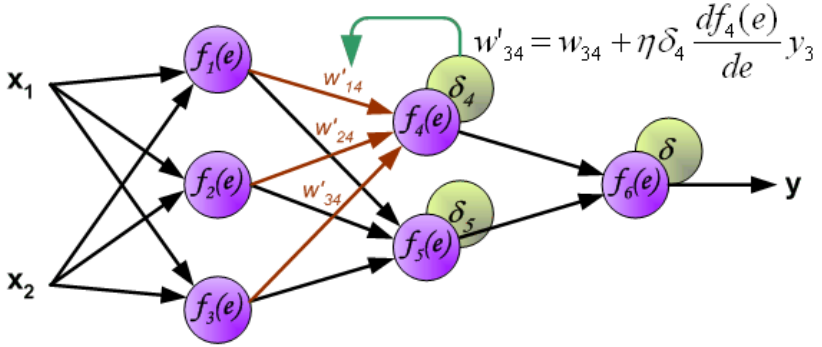
# 《数据挖掘》计算题复习范围

## 七、神经网络的正反向传播计算【参数修改2】

$$w'_{14} = w_{14} + \eta\delta_4 \frac{df_4(e)}{de} y_1$$

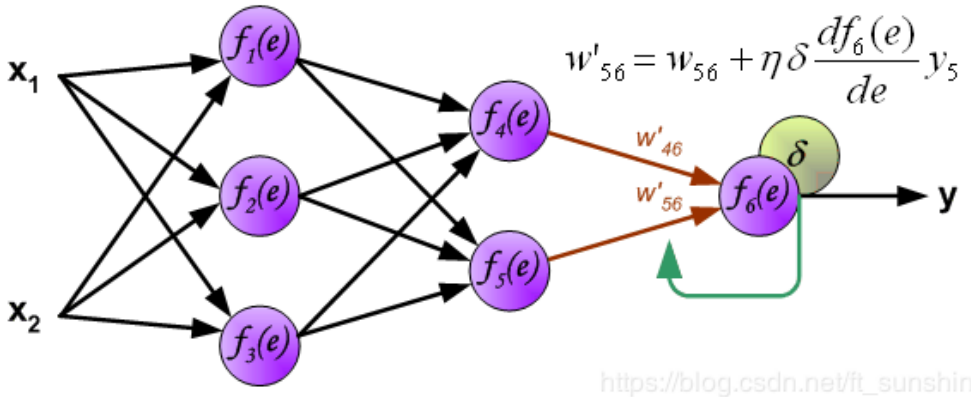
$$w'_{24} = w_{24} + \eta\delta_4 \frac{df_4(e)}{de} y_2$$

$$w'_{34} = w_{34} + \eta\delta_4 \frac{df_4(e)}{de} y_3$$




$$w'_{46} = w_{46} + \eta\delta \frac{df_6(e)}{de} y_4$$

$$w'_{56} = w_{56} + \eta\delta \frac{df_6(e)}{de} y_5$$

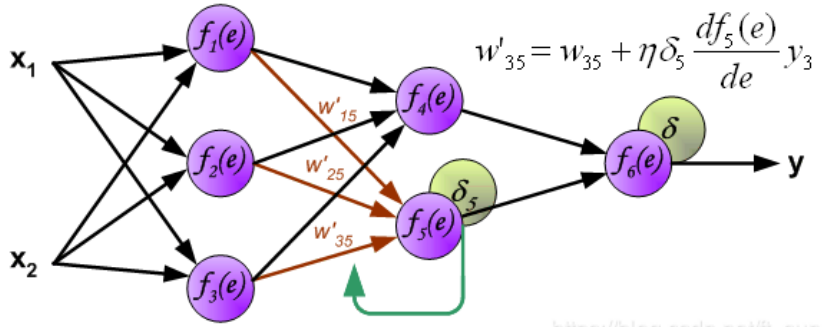



https://blog.csdn.net/ft_sunshine

$$w'_{15} = w_{15} + \eta\delta_5 \frac{df_5(e)}{de} y_1$$

$$w'_{25} = w_{25} + \eta\delta_5 \frac{df_5(e)}{de} y_2$$

$$w'_{35} = w_{35} + \eta\delta_5 \frac{df_5(e)}{de} y_3$$

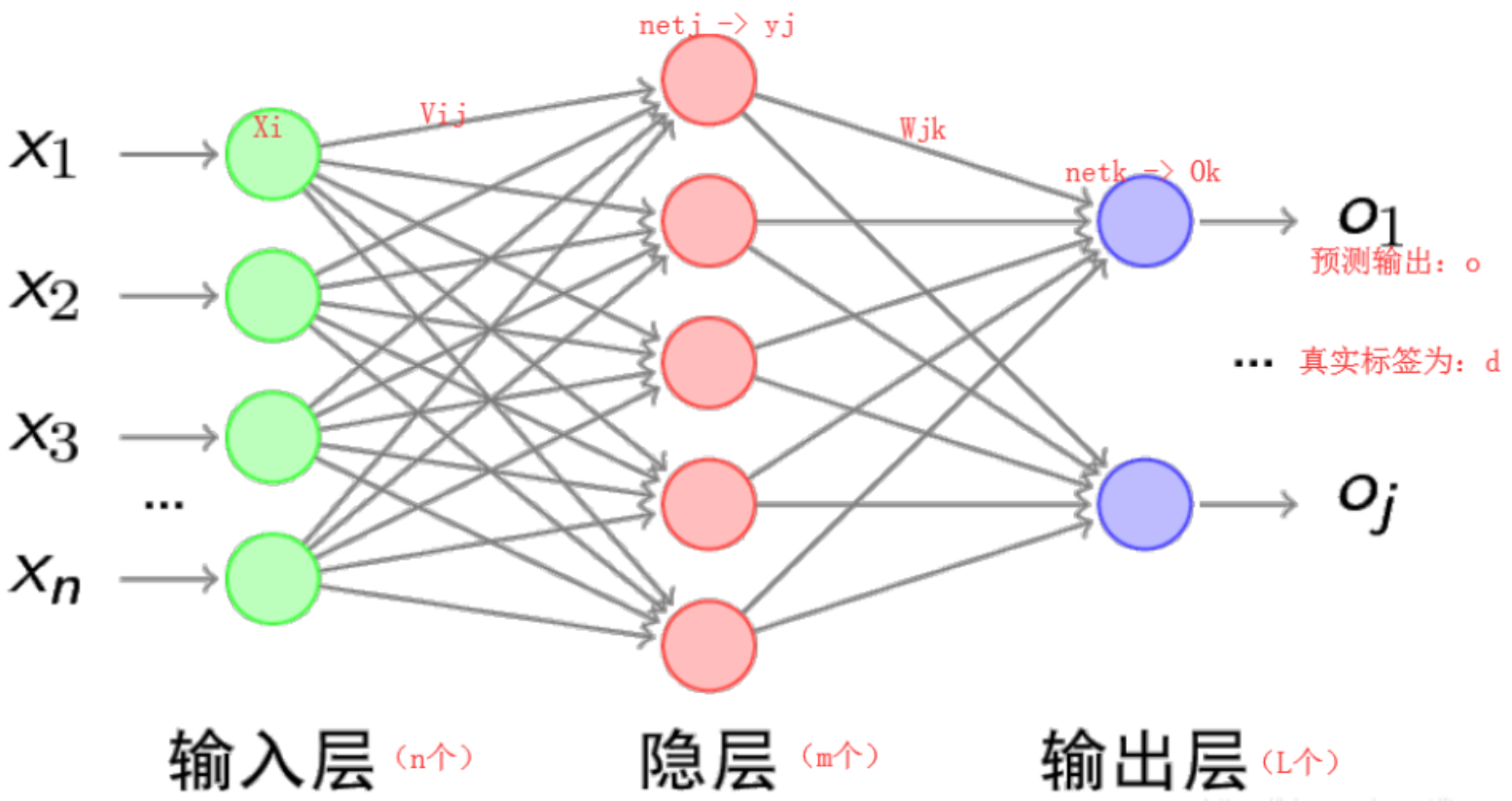



https://blog.csdn.net/ft_sunshine

## 七、神经网络的正反向传播计算【总结】

首先明确，**“正向传播”求损失，“反向传播”回传误差**。同时，神经网络每层的每个神经元都可以**根据误差信号修正每层的权重**，只要能明确上面两点，那么下面的例子，只要会一点**链式求导**规则，就一定能看懂！

**BP算法，也叫$\delta$算法**，下面以3层的感知机为例进行举例讲解。




https://blog.csdn.net/ft_sunshine

七、神经网络的正反向传播计算【总结】

- 输出层

$$E=\frac{1}{2}(d-O)^2 \qquad =\frac{1}{2}\sum_{\kappa=1}^{\ell}(d_k-o_k)^2$$

- 误差展开至隐层

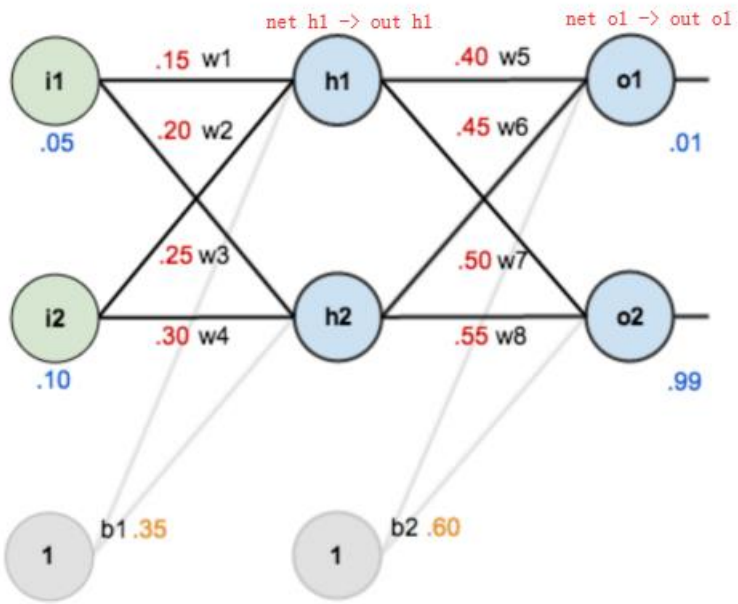
$$E=\frac{1}{2}\sum_{\kappa=1}^{\ell}[d_\kappa-f(net_\kappa)]^2 \qquad =\frac{1}{2}\sum_{\kappa=1}^{\ell}[d_\kappa-f(\sum_{j=0}^{m}\omega_{j\kappa}y_j)]^2$$

- 展开至输入层

$$E=\frac{1}{2}\sum_{\kappa=1}^{\ell}\left(d_\kappa-f[\sum_{j=0}^{m}\omega_{j\kappa}f(net_j)]\right)^2 \qquad =\frac{1}{2}\sum_{\kappa=1}^{\ell}\left(d_\kappa-f[\sum_{j=0}^{m}\omega_{j\kappa}f(\sum_{i=0}^{n}v_{ij}\chi_i)]\right)^2$$

https://blog.csdn.net/ft_sunshine

七、神经网络的正反向传播计算【例题：前向传播（sigmoid）】




$$net_{h1} = w_1 * i_1 + w_2 * i_2 + b_1 * 1$$

$$net_{h1} = 0.15 * 0.05 + 0.2 * 0.1 + 0.35 * 1 = 0.3775$$

$$out_{h1} = \frac{1}{1+e^{-net_{h1}}} = \frac{1}{1+e^{-0.3775}} = 0.593269992$$

$$out_{h2} = 0.596884378$$

$$net_{o1} = w_5 * out_{h1} + w_6 * out_{h2} + b_2 * 1$$

$$net_{o1} = 0.4 * 0.593269992 + 0.45 * 0.596884378 + 0.6 * 1 = 1.105905967$$

$$out_{o1} = \frac{1}{1+e^{-net_{o1}}} = \frac{1}{1+e^{-1.105905967}} = 0.75136507$$

$$out_{o2} = 0.772928465$$

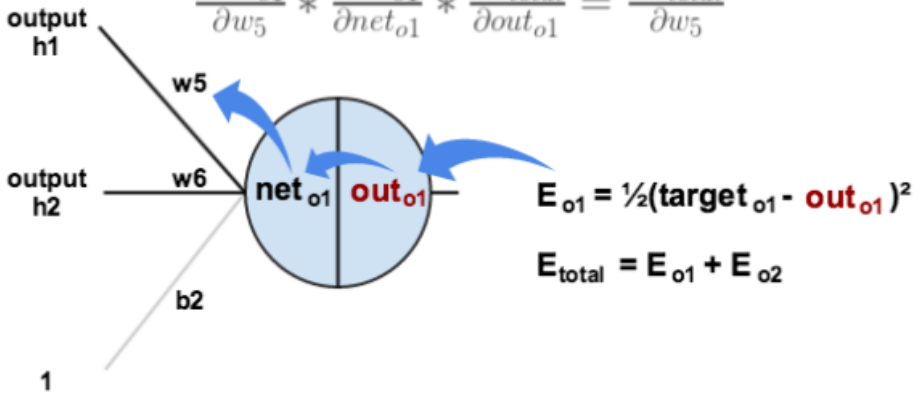
$$E_{total} = \sum \frac{1}{2}(target - output)^2$$

$$E_{total} = E_{o1} + E_{o2} = 0.274811083 + 0.023560026 = 0.298371109$$

https://blog.csdn.net/r_sunshine

七、神经网络的正反向传播计算【例题：反向传播（sigmoid）】

$$\frac{\partial net_{o1}}{\partial w_5} * \frac{\partial out_{o1}}{\partial net_{o1}} * \frac{\partial E_{total}}{\partial out_{o1}} = \frac{\partial E_{total}}{\partial w_5}$$




$$E_{total} = \frac{1}{2}(target_{o1} - out_{o1})^2 + \frac{1}{2}(target_{o2} - out_{o2})^2$$

$$\frac{\partial E_{total}}{\partial out_{o1}} = 2 * \frac{1}{2}(target_{o1} - out_{o1})^{2-1} * -1 + 0$$

$$\frac{\partial E_{total}}{\partial out_{o1}} = -(target_{o1} - out_{o1}) = -(0.01 - 0.75136507) = 0.74136507$$

$$out_{o1} = \frac{1}{1+e^{-net_{o1}}}$$

$$\frac{\partial out_{o1}}{\partial net_{o1}} = out_{o1}(1 - out_{o1}) = 0.75136507(1 - 0.75136507) = 0.186815602$$

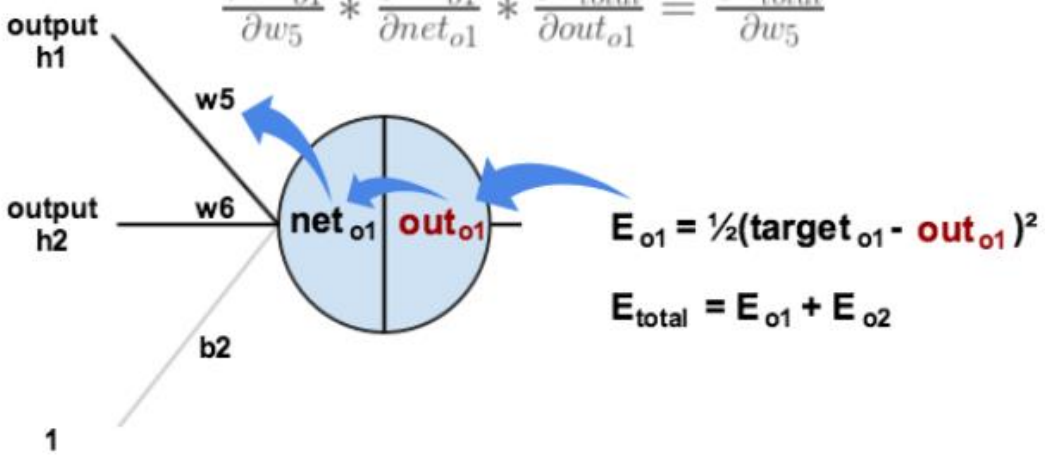
$$net_{o1} = w_5 * out_{h1} + w_6 * out_{h2} + b_2 * 1$$

$$\frac{\partial net_{o1}}{\partial w_5} = 1 * out_{h1} * w_5^{(1-1)} + 0 + 0 = out_{h1} = 0.593269992$$

$$\frac{\partial E_{total}}{\partial w_5} = 0.74136507 * 0.186815602 * 0.593269992 = 0.082167041$$

七、神经网络的正反向传播计算【例题：参数调整（sigmoid）】

$$\frac{\partial net_{o1}}{\partial w_5} * \frac{\partial out_{o1}}{\partial net_{o1}} * \frac{\partial E_{total}}{\partial out_{o1}} = \frac{\partial E_{total}}{\partial w_5}$$




$$\frac{\partial E_{total}}{\partial w_5} = -(target_{o1} - out_{o1}) * out_{o1}(1 - out_{o1}) * out_{h1}$$

$$w_5^+ = w_5 - \eta * \frac{\partial E_{total}}{\partial w_5} = 0.4 - 0.5 * 0.082167041 = 0.35891648$$

$$w_6^+ = 0.408666186$$

$$w_7^+ = 0.511301270$$

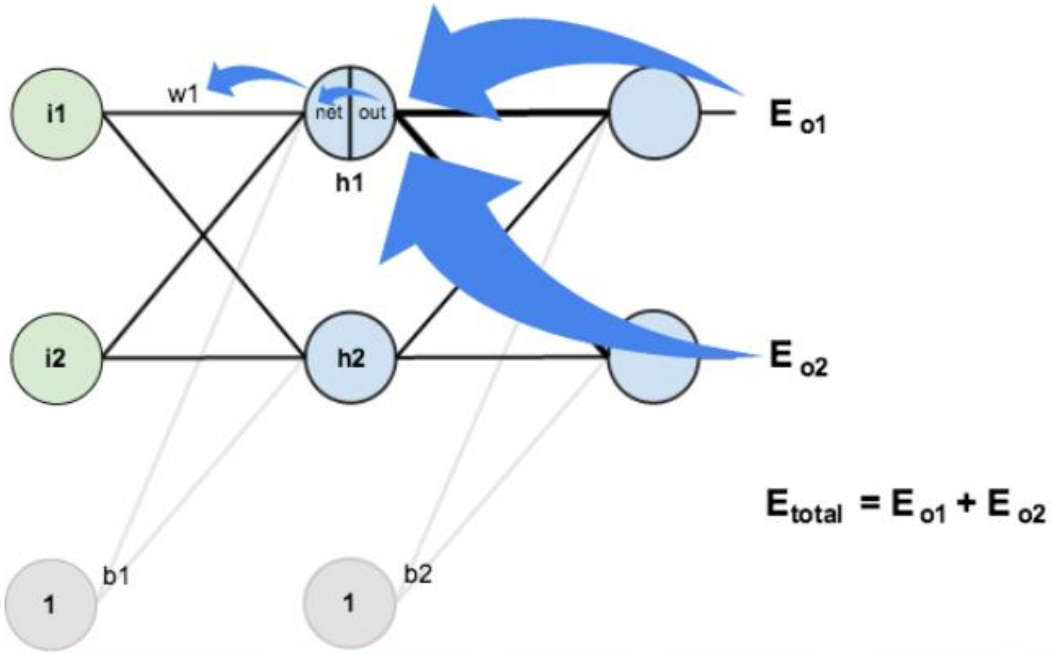
$$w_8^+ = 0.561370121$$

https://blog.csdn.net/ft_sunshine

七、神经网络的正反向传播计算【例题：参数调整（sigmoid）】

$$\frac{\partial E_{total}}{\partial w_1} = \frac{\partial E_{total}}{\partial out_{h1}} * \frac{\partial out_{h1}}{\partial net_{h1}} * \frac{\partial net_{h1}}{\partial w_1}$$

$$\frac{\partial E_{total}}{\partial out_{h1}} = \frac{\partial E_{o1}}{\partial out_{h1}} + \frac{\partial E_{o2}}{\partial out_{h1}}$$




$$\frac{\partial E_{total}}{\partial w_1} = \left(\sum_o \frac{\partial E_{total}}{\partial out_o} * \frac{\partial out_o}{\partial net_o} * \frac{\partial net_o}{\partial out_{h1}}\right) * \frac{\partial out_{h1}}{\partial net_{h1}} * \frac{\partial net_{h1}}{\partial w_1}$$

$$w_1^+ = w_1 - \eta * \frac{\partial E_{total}}{\partial w_1} = 0.15 - 0.5 * 0.000438568 = 0.149780716$$

https://blog.csdn.net/lt_sunshine