1. 栈：先进后出，后进先出
2. 顺序表和链表的比较

存取方式

顺序表可以顺序存取，也可以随机存取，链表只能从表头顺序存取元素。

逻辑结构与物理结构

顺序存储时，逻辑上相邻的元素，对应的物理存储位置也相邻；

链式存储时，逻辑上相邻的元素，物理存储位置不一定相邻，其逻辑关系是通过指针链接来表示的。

查找、插入和删除操作

对于按值查找，顺序表无序时，两者的时间复杂度均为O(n)；顺序表有序时，可采用折半查找，此时的时间复

杂度为O(log2n）

对于按序号查找，顺序表支持随机访问，时间复杂度仅为O(1)，而链表的平均时间复杂度为O(n)。

顺序表的插入、删除操作，平均需要移动半个表长的元素。

链表的插入、删除操作，只需要修改相关结点的指针域既可。由于链表的每个结点都有指针域，所以在存储空间上要比顺序表付出的代价大，存储密度不够大。

1. 关于图的说法正确的是

可以使用邻接表或邻接矩阵来存储图；

对有n 个顶点、 e 条边且使用邻接表存储的有向图进行广度优先遍历的时间复杂度是O(n+e)；

没有回路的连通图也是树；

1. 大驼峰命名法：函数名，命名空间，属性
2. 参数初始化策略：标准初始化，随机初始化，Xavier初始化（为了使得网络中信息更好的流动，每一层输出的方差应该尽量相等。）
3. 常见的判别模型：K邻近，线性回归，条件随机场
4. 监督学习可以分为生成方法与判别方法，所学到的模型可以分为生成模型与判别模型。

生成模型由数据学习联合概率分布P(X,Y)，然后求出条件概率分布P(Y|X)作为预测的模型，即生成模型：P(Y|X)=P(X,Y)P(X) 这样的方法之所以称为生成方法，是因为模型表示了给定输入X产生输出Y的生成关系。典型的生成模型有：朴素贝叶斯法和隐马尔可夫模型

判别方法由数据直接学习决策函数f(X)或者条件概率分布P(Y|X)作为预测的模型，即判别模型，判别方法关心的是对给定的输入X，应该预测什么样的输出Y，典型的判别模型包括：K近邻法、感知机、决策树、逻辑斯蒂回归模型、最大熵模型、SVM、CRF等等

生成方法的特点

生成方法可以还原出联合概率分布P(X,Y)，而判别方法则不能；

生成方法的学习收敛速度更快，即当样本容量增加的时候，学到的模型可以更快的收敛于真实模型；

当存在隐变量时，仍可以用生成方法学习，此时判别方法就不能用。

判别方法的特点

判别方法直接学习的是条件概率P(Y|X)或决策函数f(X)，直接面对预测，往往学习的准确率更高；

由于直接学习P(Y|X)或f(X)，可以对数据进行各种程度上的抽象、定义特征并使用特征，因此可以简化学习问题。

总结，生成模型在学习过程中可以得到联合分布，但其最终学习的目标可以不是联合分布，可以与判别模型一样学习条件概率分布。联合分布虽然能提供更多的信息，但也需要更多的样本和更多的计算。因此，对于样本数量较少的问题，优先考虑判别模型。

1. 交叉验证的方式

第一种是简单交叉验证，所谓的简单，是和其他交叉验证方法相对而言的。首先，我们随机的将样本数据分为两部分（比如：70%的训练集，30%的测试集），然后用训练集来训练模型，在测试集上验证模型及参数。接着，我们再把样本打乱，重新选择训练集和测试集，继续训练数据和检验模型。最后我们选择损失函数评估最优的模型和参数。

第二种是S折交叉验证（S-Folder Cross Validation）。和第一种方法不同，S折交叉验证会把样本数据随机的分成S份，每次随机的选择S-1份作为训练集，剩下的1份做测试集。当这一轮完成后，重新随机选择S-1份来训练数据。若干轮（小于S）之后，选择损失函数评估最优的模型和参数。

第三种是留一交叉验证（Leave-one-out Cross Validation），它是第二种情况的特例，此时S等于样本数N，这样对于N个样本，每次选择N-1个样本来训练数据，留一个样本来验证模型预测的好坏。此方法主要用于样本量非常少的情况，比如对于普通 适中问题，N小于50时，我一般采用留一交叉验证。留一法的缺点是：当n很大的时候，计算量会很大，因为需要进行n次模型的训练，而且训练集的大小为n-1

1. 二元分类的准确率和召回率

阈值提高：准确率增加或不变，召回率减小或不变

1. 朴素贝叶斯

贝叶斯方法

贝叶斯方法是以贝叶斯原理为基础，使用概率统计的知识对样本数据集进行分类。由于其有着坚实的数学基础，贝叶斯分类算法的误判率是很低的。贝叶斯方法的特点是结合先验概率和后验概率，即避免了只使用先验概率的主观偏见，也避免了单独使用样本信息的过拟合现象。贝叶斯分类算法在数据集较大的情况下表现出较高的准确率，同时算法本身也比较简单。、

朴素贝叶斯算法

朴素贝叶斯算法（Naive Bayesian algorithm) 是应用最为广泛的分类算法之一。

朴素贝叶斯方法是在贝叶斯算法的基础上进行了相应的简化，即假定给定目标值时属性之间相互条件独立。也就是说没有哪个属性变量对于决策结果来说占有着较大的比重，也没有哪个属性变量对于决策结果占有着较小的比重。虽然这个简化方式在一定程度上降低了贝叶斯分类算法的分类效果，但是在实际的应用场景中，极大地简化了贝叶斯方法的复杂性。

1. Caffe（全称：Convolutional Architecture for Fast Feature Embedding），是一个计算CNN相关算法的框架，用im2col和gemm计算卷积神经网络
2. 集成学习（ensemble learning) <https://zhuanlan.zhihu.com/p/27689464>

集成方法中，用加权平均代替投票方法

1. bootstrap 自助法 从总的样本N中，有放回地抽取n个样本
2. 线性回顾与逻辑回归：https://blog.csdn.net/album\_gyd/article/details/83537992

线性回归：预测连续独立变量

1. 广度优先遍历
2. 中央处理器；

中央处理器（CPU，Central Processing Unit）是一块超大规模的集成电路，是一台计算机的运算核心（Core）和控制核心（ Control Unit）。它的功能主要是解释计算机指令以及处理计算机软件中的数据。

中央处理器主要包括运算器（算术逻辑运算单元，ALU，Arithmetic Logic Unit）和高速缓冲存储器（Cache）及实现它们之间联系的数据（Data）、控制及状态的总线。它与内部存储器（Memory）和输入/输出（I/O）设备合称为电子计算机三大核心部件。

1. 进程与线程

看进程是资源分配的最小单位，线程是CPU调度的最小单位

做个简单的比喻：进程=火车，线程=车厢

线程在进程下行进（单纯的车厢无法运行）

一个进程可以包含多个线程（一辆火车可以有多个车厢）

不同进程间数据很难共享（一辆火车上的乘客很难换到另外一辆火车，比如站点换乘）

同一进程下不同线程间数据很易共享（A车厢换到B车厢很容易）

进程要比线程消耗更多的计算机资源（采用多列火车相比多个车厢更耗资源）

进程间不会相互影响，一个线程挂掉将导致整个进程挂掉（一列火车不会影响到另外一列火车，但是如果一列火车上中间的一节车厢着火了，将影响到所有车厢）

进程可以拓展到多机，进程最多适合多核（不同火车可以开在多个轨道上，同一火车的车厢不能在行进的不同的轨道上）

进程使用的内存地址可以上锁，即一个线程使用某些共享内存时，其他线程必须等它结束，才能使用这一块内存。（比如火车上的洗手间）

"互斥锁"进程使用的内存地址可以限定使用量（比如火车上的餐厅，最多只允许多少人进入，如果满了需要在门口等，等有人出来了才能进去）－“信号量”

1. 高级语言的程序控制结构；顺序结构，分支结构，循环结构
2. 带头结点的循环双向链表（头指针记为L）为空的判定条件是：L->next==L