人工智能概述

四要素：**数据**，算法，算力，场景

数据的质量决定了模型的上限，算法是帮助模型无限的接近这个上限，算力是帮我模型加快效率

机器学习

- 有监督学习（有标签的数据集进行训练）
    - 分类：预测值是离散值
    - 回归：预测值是连续值

- 无监督学习（无标签的数据集进行训练）
    - 聚类：寻找样本之间的相似性（物以类聚，人以群分）
    - 关联规则
    - 推荐
- 半监督学习
- 强化学习

机器学习整体流程

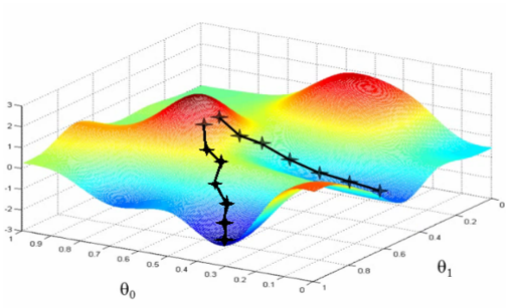
准备数据（脏数据）— 数据预处理 — 特征工程 — 选择算法搭建模型 — 模型评估和优化 — 部署上线

- 数据集
    - 训练集：用来训练模型
    - 测试集：用来做模型评估

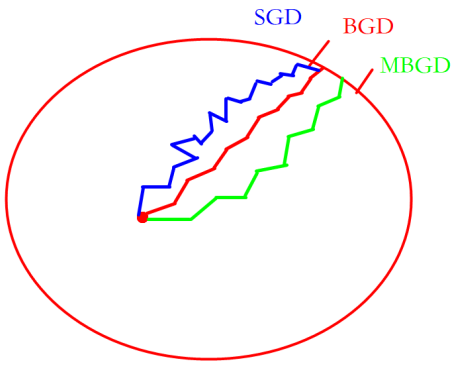
- 数据预处理
    - 数据清洗—缺失值处理（删除，插补：均值，中位数...）
    - 数据转换
        - 将离散特征值转化为对应的数值表示：label encode，独热编码
        - 标准化：统一量纲，避免量级不同对模型训练造成影响，min—max标准化
        - 数据降维
            - PCA主成分分析：将样本从高维空间映射到低维空间
            - 特征选择
                - 基于数据理解直接筛选
                - 过滤法：不依赖于模型，只关注于特征本身，计算特征中的一些相关性系数，通过设定阈值选择。（方差法，皮尔逊相关系数）

        - 包装法：依赖于模型，将特征选择问题视为一个搜索问题，评估和比较不同的组合，根据模型的准确性进行评分。（特征递归消除）
        - 嵌入法：依赖于模型，通过算法本身去做特征选择（决策树）

- 模型泛化能力：面对新样本的一个处理能力
    - 过拟合：模型过于复杂，在训练集中拟合的很好，但是泛化能力不行
    - 欠拟合：模型过于简单，没有训练到什么东西
- 模型的评估
    - 回归：平均绝对误差，均方误差，R2分数
    - 分类：混淆矩阵
- 梯度下降：模型的输出是预测值，预测值与真实值（标签）之间必然存在**误差**，将误差函数化得到的就是**损失函数**，模型训练的过程就是不断的去**优化**这个损失函数，优化损失函数就是去找损失函数的最小值，所以模型训练的过程就转化为如何去求一个函数的最小值，求函数最小值可以使用**令导数为零**的方法，但不是最普适的方法，因为损失函数往往是一个复杂的高维曲面，所以使用**梯度下降**的方法更为合适。
    - 梯度（偏导数）：方向导数中上升最快的方向
    - 学习率（超参数）：控制下降一步的大小
        - 学习率过大：可能会在最低点附近来回震荡
        - 学习率过小：会造成收敛速度过慢



- 全局梯度下降（BGD）：每次更新权重，用所有训练样本计算梯度
    - 优点：下降方向稳定
    - 缺点：下降速度比较慢
- 随机梯度下降（SGD）：每次更新权重，随机选取一条样本计算梯度
    - 优点：下降速度快
    - 缺点：下降过程比较震荡
- 小批量梯度下降（MBGD）：结合以上两者，随机选取n条样本计算梯度



- 超参数：需要人为去设定的参数
    - 搜索方法：网格搜索（枚举出所有的超参数组合），随机搜索（可能的参数值的分布中进行取样）
    - 验证集：评估模型指标，进行超参数调优（K折交叉验证）

有监督学习—回归—线性回归

1. 初始化权重W，初始化模型
2. 根据初始化模型的预测值，得到**损失函数**
3. 利用**梯度下降**算法优化损失函数，求得最小值
4. 找到对应的权重参数，做为最终模型的结果

- 过拟合问题：通过添加L2范数构成新的损失函数，优化问题变成了新的问题：要同时优化之前的均方误差和新的惩罚项，它们之间相互制约，在一定程度上能解决过拟合问题。

有监督学习—分类—逻辑回归

实现二分类功能，在线性回归的基础上套上了sigimod函数，引入了非线性，将值域映射到了0—1范围内，设定阈值来处理二分类功能，损失函数为最大似然估计。

有监督学习—分类—决策树

ID3

- 熵：一个物体的混乱程度
- 信息熵：一件事情的不确定性 公式：—∑PlogP
- 信息增益：初始信息熵 — 条件熵（某个条件让这件事情变得确定了多少，量化）

选择信息增益最大的特征做为优先划分依据

缺点：偏向于选择特征值较多的特征做为优先划分依据

C4.5

- 信息增益率 = 信息增益/分裂信息

当特征值较多的时候带来的信息增益会比较大，但同时特征本身的信息熵也会比较大，分裂信息就是特征本身的信息熵，把分裂信息做为分母，相当给原来的信息增益加上了一个惩罚项，从而解决ID3算法偏向于选择特征值较多的特征做为划分依据的缺点。

过拟合：

树的过拟合可以通过剪枝来解决，预剪枝：在建树之前提前限定好树的深度

有监督学习—分类—KNN

1. 计算待分类样本离其它所有样本之间的距离（相似度）
2. 根据相距离的远近，将样本由近到远进行排序
3. 取前K（超参数）个样本，根据类别少数服从多数进行分类

优点：逻辑简单 缺点：计算量大

无监督学习—聚类—Kmeans

1. 初始化K个质心（超参数）
2. 计算所有样本离这K个质心的距离
3. 选择距离最近的质心，聚为一类
4. 根据已聚好的样本的均值来更新质心的位置
5. 循环执行2,3,4步骤
6. 直到质心的位置不再发生改变的时候，算法停止

优点：逻辑简单

缺点：非常容易受到初始质心位置的影响，导致聚类效果不好，处理不了非凸数据集

无监督学习—聚类—自下而上层次聚类

1. 首先将N个样本视为N个簇
2. 计算簇与簇之间的距离，找到最近的两个簇合并为一个
3. 更新合并之后簇的中心点
4. 重复执行2,3两步
5. 直到N个样本被聚成一个簇后，算法停止

集成学习

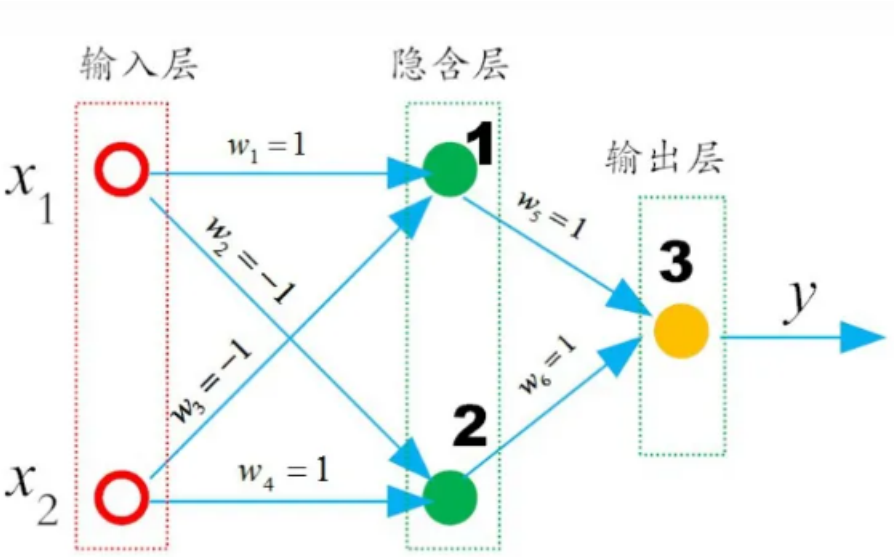
- Bagging：随机森林，通过对原数据集进行随机有放回的采样，根据采样后的子集构造一颗弱决策树，最后根据所有弱决策树的分类结果，进行平权投票做为最终的结果。
- Boosting

深度学习

神经网络：输入层，隐藏层，输出层

起源：单层感知机，只能处理线性可分，不能解决异或问题

多层感知机模型可以解决异或问题



(a) 实现"异或"的网络结构

(神经元节点阈值均为0.5)

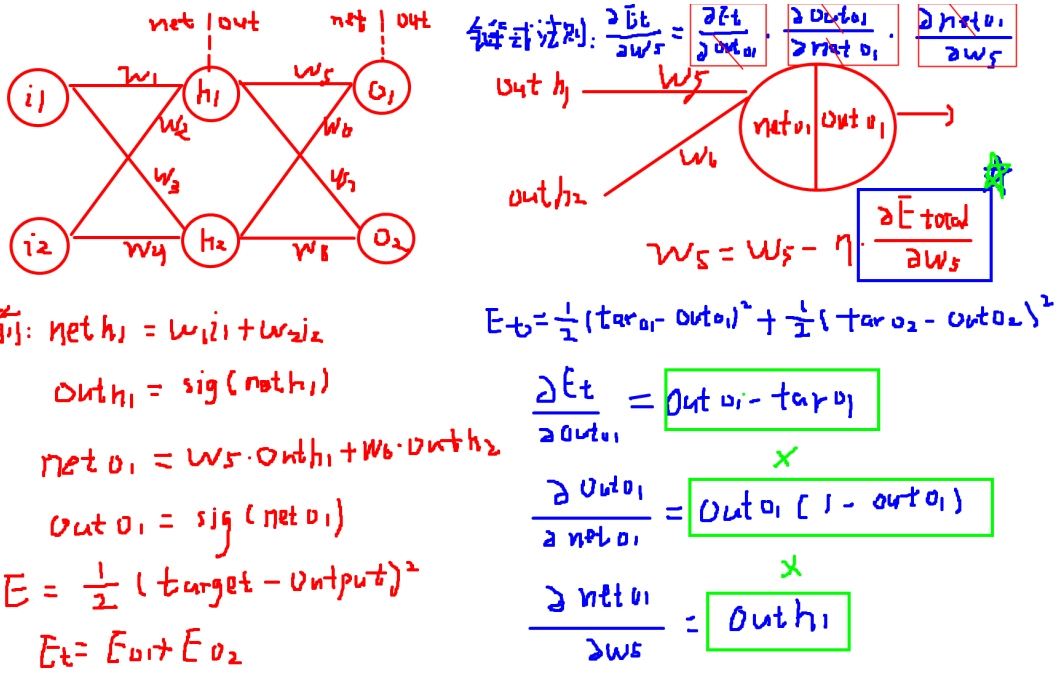
前馈神经网络：每个神经元只与前一层的神经元相连，同一层的神经元之间互相没有联系，信号从输入层向输出层单向传播，隐藏层越多，模型的表达能力越强，但是也会发生过拟合问题。

训练法则

梯度下降是解决损失函数最小值问题

反向传播是解决梯度下降中的梯度计算

两者组合在一起，共同解决前馈神经网络中的训练问题



激活函数

如果没有激活函数的话，无论隐藏层有多少，无论怎么调整权重，其输出的值仍然为线性，真实世界中模型必然是非线性的，为了让模型有拟合非线性的功能，因此必须要加入激活函数。

- sigmoid
- tanh
- relu
- softmax

  sigmoid激活函数的导数落于0—0.25之间，tanh激活函数的导数落于0—1之间，当隐藏层过多时，容易造成梯度消失现象，当X过大或者过小时，梯度会变得很小，会拖慢整个梯度下降法。
  relu激活函数可以解决梯度消失的现象，但是它的缺点是当X小于零时，会造成神经元失活。
  softmax激活函数主要用于多分类任务。

过拟合问题解决

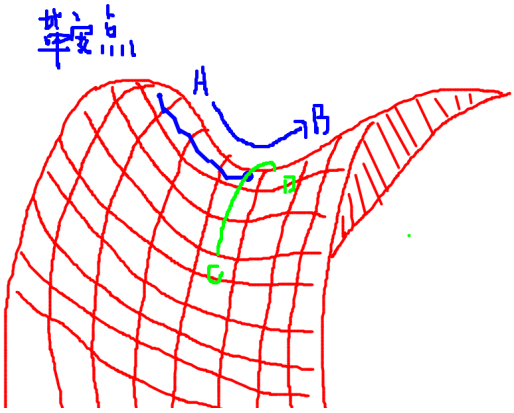
- L1,L2正则，通过添加L1,L2范数做为惩罚项构造新的目标函数
- 数据增强：扩充训练集，训练集合越大时，过拟合的概率越小
  - 图像识别领域：旋转图片，P图
  - NLP领域：近义词替换
- 提前停止，配合验证集，当发现模型在验证集中的损失开始变大时，提前停止训练
- Dropout：集成的思想，通过随机丢弃一部分神经元形成子网络，最后将子网络进行合并

数据不平衡问题解决

欠采样，过采样，合成采样

优化器

优化的目标在于降低训练中的损失，优化遇到的挑战：局部最优（鞍点问题）等



- 动量法

  思想：指数加权平均，主要是用来处理序列化数据的方法 ,公式： $S_t = \begin{cases} Y_1 & (t>1) \\ \beta S_{t-1} + (1-\beta) Y_t & (t>1) \end{cases}$

  Yt为t时刻下的真实值，St为t时刻下加权平权后的值，β是权重

  当β越大，曲线越平滑而且滞后，称为偏差修正

动量梯度下降就是计算梯度的加权平均数，并利用该值去更新参数，β通常设为0.9

通过累加过去的梯度值来减少抵达最小值路径上的波动，加速了收敛。当前后梯度方向一致时，动量法能加　速学习，不一致时能抑制震荡或者冲破鞍点。形象理解：动量法类似于小球下山，小球向下运动过程中会有　加速度，导致越来越快，如果遇到了局部最小值，会冲到对面的山坡上然后继续下降，可以避过鞍点。

卷积神经网络CNN

核心思想：局部感知，参数共享

网络架构：卷积层，池化层，全连接层

- 卷积层：本质上就是一个参数矩阵，用来扫描整张图片，提取特征，卷积层的通道数一定要与图片保持一致
- 池化层：最大池化的作用就是保留最大特征，降维
- 全连接层：使用Softmax激活函数，输出每一个类别的得分

在卷积神经网络中，没有固定的结构，根据具体的场景，可以将卷积层和池化结合使用

循环神经网络RNN

主要的作用就是理解上下文信息，通过将上一时刻的输出做为下一时刻的输入的方法，对以前的信息进行记忆，但是因为记忆的东西太多，所以长期的记忆效果不好（梯度消失），所以为了改良这一点，有了后面的LSTM，通过添加遗忘门来对重要信息进行选择性记忆。