1. 特征工程

结构化数据（表） and 非结构化数据（文本、视频）

1. 特征归一化
2. Min-Max 归一化
3. 零均值归一化

作用：

a. 消除量纲影响

b. 提升收敛速度，使不同特征对应参数更新速度变得一致，容易更快通过梯度下降获得最优解。

Ps: 适用于——线性回归、逻辑回归、SVM 、神经网络 等需要通过梯度下降来实现的算法。

不适用与——决策树，for example C4.5 计算信息增益率时跟特征是否归一化没有关系。

1. 属性特征处理
2. 序号编码——保留大小关系
3. One-hot——产生稀疏数据，可以配合特征选择来降低维度
4. 二进制编码
5. 特征组合
6. 提高复杂关系的拟合能力，一阶特征两两组合，构造高阶特征——容易过拟合
7. 如何构造有效特征——利用决策树（GBDT）构造，从根节点到叶子节点可以看成是一个特征组合方式。
8. 文本表示
9. 词袋(忽略顺序，TF-IDF=TF\*IDF) and N-gram（连续的N个词表示一个东西）
10. Embedding
11. Word2Vec——（CBOW and Skip-gram）

CBOW : 根据上下文预测当前词的生成概率

Skip-gram : 根据当前词预测上下文中各个词的概率

输入层，映射层（拼接），输出层

1. 图像数据不足处理

数据不足，过拟合

1. 简化模型，正则化缩小假设空间，集成学习，Dropout
2. 数据扩充（图片旋转、平移，裁减…，颜色变化等），GAN
3. 模型评估
4. 指标
   1. 准确率Accuracy：
   2. 精确率P： 分类正确的正样本/分类器分类的正样本
   3. 召回率Recall : 分类正确的正样本/真正正确的正样本
   4. P-R 曲线： 对模型进行全面评估
   5. F1-score: 
   6. 平方根RMSE：如果存在个别偏离很大的离群点时，即使点的数量很少，RMSE会很大
   7. FPN (假阴性率):， FP 负样本中被预测为正的数量
   8. TPR (真阳性率):, TP 正样本中被分为正的数量
   9. ROC曲线：FPR-TPR
   10. AUC: ROC下的面积
   11. P-R 和 ROC 的区别：
       1. 当正负样本比例发生改变时，ROC曲线形状基本不变，P-R发生剧烈变化。
       2. ROC能够降低不同测试集带来的影响，对于正负样本分布很不均匀的情况，采用不同的测试集，P-R变化明显，而ROC能够很好的反映模型本身的好坏。
       3. 如果研究者希望看到特定数据集上的表示，P-R能够直观的查看性能。
5. Cos距离
   1. 欧式距离体现数值上的绝对差距，余弦距离反映方向差距。
   2. 两部剧的观看行为（1,0）(0,1) cos差距大，欧氏距离差距小，选cos.；活跃度：（1,10）（10,100），选欧式
   3. 曼哈顿距离——
6. 模型评估方法
7. Holdout 留出法：25% and 75%
8. K-Fold 交叉验证
9. 自助法：有放回n次抽样，有1/e 没被抽到，36.8%
10. 超参数调优
11. 网格搜索
12. 随机搜索——收敛
13. 贝叶斯搜索
14. 过拟合与欠拟合
15. 过拟合：训练集表现好 and 测试集表现差
16. 欠拟合：all bad
17. 降低过拟合：
    * 1. 从数据出发，获得更多数据
      2. 降低模型复杂度
      3. 正则化
      4. 集成学习
18. 降低欠拟合：
19. 添加新特征
20. 增加模型复杂度
21. 减小正则化系数
22. 基础模型
23. SVM 严格推导过程-ipad
24. 逻辑回归

a. 逻辑回归用于分类，线性回归用于回归

b. 似然函数

c. 多标签时用softmax处理

1. 决策树：

a. ID3 —— 信息增益，偏好稀疏特征，容易过拟合

b. C4.5 —— 信息增益率

c. CART —— 基尼系数gini

对比：ID3 只能处理离散变量，C4.5 and CART 能够处理连续变量，能够将连续变量划分为离散变量。

CART 能够进行回归

1. 剪枝，预剪枝和后剪枝
2. 降维

降维具有如下一些优点：

1）使得数据集更易使用 2）降低算法的计算开销 3）去除噪声 4）使得结果容易理解

1. PCA ： 最大方差就是最大特征值，最佳投影就是最大特征值对应的特征向量——无监督，非凸优化。
2. LDA ： 线性判别——有监督。

首先我们看看相同点：

1）两者均可以对数据进行降维。

2）两者在降维时均使用了矩阵特征分解的思想。

3）两者都假设数据符合高斯分布。

我们接着看看不同点：

1）LDA是有监督的降维方法，而PCA是无监督的降维方法

2）LDA降维最多降到类别数k-1的维数，而PCA没有这个限制。的秩最大也为N-1

3）LDA除了可以用于降维，还可以用于分类。

4）LDA选择分类性能最好的投影方向，而PCA选择样本点投影具有最大方差的方向。

**PCA——主成分分析**

思想：最大化投影方差，让数据在主轴上的投影方差最大。目的是为了找到一个投影方向，让投影后的数据方差最大，投影后的数据均值为0。

*步骤：*给定数据

1.去中心化，计算均值，

***a.*** 在上投影为

***b.*** 投影后所有数据均值：

***c.*** 投影后所有数据的方差：

***d***. 其中为样本协方差矩阵。最大化目标化为;



运用拉格朗日\*\*法，令得，此时，

2.求样本协方差矩阵

3.对协方差矩阵进行特征值分解，将特征值从大到小排序

4.求特征值前大对应的特征向量，



其中，，信号占比

**LDA——线性判别分析**

思想：最大化类间间距和最小化类内间距。

*步骤：*给定两类样本，

1. 计算类别均值，，最大化两类中心在方向上投影的距离：



1. 考虑加入类别内方差：



其中：

于是，



求导，令得即，对应的最大特征值，不考虑长度，只要求样本内均值+类内方差，即可计算，



扩展到多类时，需要计算全局散度，，表示全样本均值。通过，间接计算，得





求解，为特征值前大对应的特征向量。

*步骤：*

1. 计算每个类的均值和总体均值
2. 计算类内散度、全局散度，类间散度
3. 对矩阵进行特征值分解，将特征值从大到小排序
4. 求特征值前大对应的特征向量，



其中，，信号占比

**总结**

PCA是无监督的可降噪，分解协方差矩阵

LDA是有监督的可分类，分解

1. 非监督学习

1. K-means聚类

2. DBSCAN

3. 聚类的评估——密集程度

1. 优化算法
2. 有监督学习的损失函数
3. Hinge损失
4. Logistics损失 {0,1}
5. Cross 交叉熵损失 {-1,1}
6. 平方损失（离真实值太远值太大）
7. 绝对损失
8. Huber损失
9. 优化算法

a. 一阶展开——梯度下降

b. 二阶展开——牛顿迭代

3. 梯度验证——|定义-迭代|

4. 梯度下降

a. 批量梯度下降

b. 随机梯度下降

c. 小批量梯度下降

加速：

a. Momentum

b. Adagrad

c．Adam

5. L1 L2与lasso 回归如何计算梯度

可梯度下降条件时：凸函数，封闭解。

L1 解空间是多边形，L2 解空间是圆形，多变形的解容易在尖角处于等高线碰撞，因此产出稀疏特征。

L1正则：

1）降低模型的复杂度

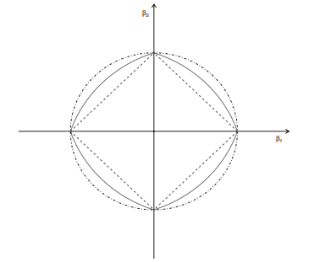
2）做特征选择

这是由于当采用L1正则后模型中对于部分特征的权重会置零。这样可以有效的降低有依赖的特征，起到特征选择的作用，同时特征维度降低后模型的复杂度也随之降低。所以L1正则适用于特征有相互依赖，且对权重是0或者非0相当敏感的模型。

L2正则：

加上L2正则后的损失要求损失函数降低的同时，特征权重也被限制在较小的范围。这样当特征维度较高时候也能保证模型受高次幂的影响较低，是模型能够在经验风险和结构风险之间得以平衡，提高模型的鲁棒性，降低模型的复杂度。所以L2正则适用于特征维度高的模型。

elastic net：在相互依赖的特征中选两个



1. 采样
2. 逆函数方法
3. 重要性采样
4. 接受拒接采样
5. 吉布斯采样
6. 过采样 + 少 ——> 过拟合
7. 欠采样 – 多 ——> 信息学不够
8. 神经网络
9. 激活函数

a. sigmoid

b. Tanh

c. Relu

Relu优缺点：

优点：

* + - * 1. 计算角度，sigmoid和Tanh需要计算指数，计算复杂度高，Relu只需要一个阀值就可以计算激活值。
        2. Relu的非饱和性可以有效解决梯度消失问题，提供相对宽的激活边界
        3. Relu的单边抑制提供网络的稀疏表达能力

缺点：

1. 导致训练过程中神经元坏死
2. 损失函数
3. 平方损失适用于输出连续，并且最后一层不含sigmoid或softmax
4. 交叉熵损失适用于二分类和多分类

由于值太大的话，sigmoid求导后会—>0，导致梯度更新值太小

1. 训练技巧
2. 初始化不能全为0
3. Dropout可以抑制过拟合=集成，[0,1,0,1,1]
4. 批量归一化 batch normalization
5. CNN
6. 卷积

稀疏交互：filter与部分神经元连接，捕获局部特征

参数共享：使卷积层具有平移不变性，先卷积再平移=先平移再卷积

1. 池化

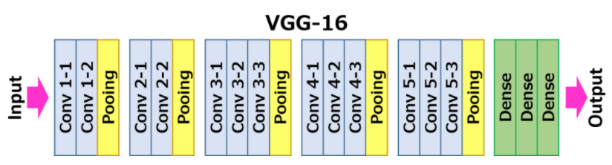
最大值池化、均值池化

显著降低参数量外，还能保持对平移、伸缩、旋转操作的不变性。如输入（1,5,3）右移一位后（0,1,5）都是5

1. 用于文本

输入N K矩阵，两个通道，一个固定一个训练。一个预先训练的词嵌入可以获得更多预料库的先验知识。一个当前网络训练的词向量可以抓住与当前任务相关的特征。

VGG16： 13个卷积层，3个全连接层，5个池化层



1. DRN

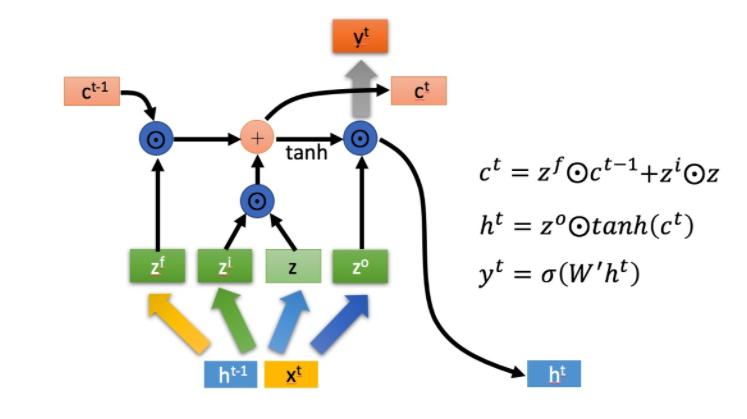
提出背景：为了解决和缓解深层神经网络训练的梯度消失问题

核心理论：因为离输入近的网络层容易出现梯度消失，于是将它短接到里输出较近的的层，训练一个拟合残差的F(x),输出H(x)=F(x)+x

ResNet可以有效改善深层的神经网络学习问题，使得训练成为可能。

传统神经网络，随模型的结构的加深训练误差反而上升，而ResNet随着模型结构的加深，训练误差下降且优于相同层数的传统神经网络。

1. RNN LSTM



1. 集成学习
2. Boosting——各个学习器之间相互依赖，学习器层层叠加，每次学习都会给前面训练错误样本更高的权重，测试时，将各层结果加权叠加，串行训练。
3. Bagging——各个学习器之间相互独立，可以并行运算。

Random Forest——为了使学习器之间相互独立，将训练集分为若干子集，可能重叠。Bagging方法像独立决策过程，每个个体进行独立学习，学习内容可以相同也可以不同，由于个体之间存在差异，最终做出的判断不会完全一致。

1. 基分类器有较大的方差和偏差。
2. Boosting 通过逐步聚集错误样本，减少了模型的训练偏差，但是不会降低方差，因为不是独立的
3. Bagging 分而治之，减少了模型的方差，
4. 单独来看，基分类器有过拟合现象，但是Bagging集成后可以使分类边界平滑，减少方差
5. 模型复杂度过低，方差小偏差高；模型复杂度高，偏差小方差高。
6. 集成学习的步骤和例子
7. 找到误差独立的基分类器
8. 训练基分类器
9. 合成基分类器的结果 Voting and stacking

基分类器：

1. 决策树：可以方便的将样本权重合整合到训练过程中，而不需要使用过采样的方法来调整样本权重。

其表达能力和泛化能力可以通过层数和叶子数来调节，数据样本的扰动对于决策树的影响较大，不

同数据集生成的基分类器的随机性较大，这样的不稳定学习器更适合用来做基分类器。

神经网络：同样具有“不稳定”性，而且可以通过调节神经元和层数来引入随机性。

K-近邻和线性分类器比较稳定，不适合，本身方差就较小

K-NN*：K近邻算法，即是给定一个训练数据集，对新的输入实例，在训练数据集中找到与该实例最邻近K个实例，这K个实例的多数属于某个类，就把该输入实例分类到这个类中。（这就类似于现实生活中少数服从多数的思想），画个圈看K个点中谁的个数多。k值既不能过大，也不能过小，选取k值很重要的关键是实验调参，类似于神经网络选取多少层这种，通过调整超参数来得到一个较好的结果。特征归一化。*

Adaboost:

循环T个学习器，采样，训练基分类器，计算基分类器权重，更新采样点权重，循环，合成结果

GBDT:

每个样本的残差训练下一棵树，直到残差小于某一个值或者树量到达某一个上限。在预测时需要把所有的树的预测值相加。

梯度提升是在损失函数空间进行更新，而梯度下降是在参数空间进行更新。

优点：

1. 预测阶段可以并行计算，速度快
2. 在分布稠密的数据集上，泛化能力和表达能力都比较好
3. 采用决策树可以是GBDT具有较高的解释性和鲁棒性，能自动发现特征间的高维特征，不需要归一化。

缺点：

1. 在高维稀疏数据上不如SVM and DNN
2. GBDT在处理文本数据时优势不明显
3. 训练需要串行，只能在局部提高速度

XGBoost：

1. GBDT是机器学习算法，XGBoost是他的一个实现
2. 在使用CART时，XGBoost使用正则化控制模型的复杂度，有利于防止过拟合，提高模型的泛化能力
3. GBDT只用到代价函数的一阶导，而XGBoost对代价函数进行二阶展开，可同时使用一阶和二阶
4. 传统的GBDT使用CART做分类器，而XGBoost支持多种类型的分类器，比如线性分类器
5. 传统GBDT每轮迭代需要所有数据，而XGBoost采用和随机森林相似的策略，支持数据采样
6. 传统的GBDT没有对缺失值进行处理，而XGBoost能够学习出缺失值处理策略
7. GAN

生成器和判别器，锁定