统计学习方法

三要素：模型、策略、算法
监督学习(重点)、非监督学习、半监督学习、强化学习
监督学习：学习一个模型，使模型对任意给定输入，对其相应输出做出预测
输入输出变量均为连续变量的预测问题为回归问题
输出为有限个离散变量，为分类问题
输入输出均为变量序列，为标注问题
假设空间是输入空间到输出空间映射的集合

决策函数为非概率模型，条件概率表示为概率模型
策略：最优化，损失函数用来度量预测错误的程度，越小越好；风险函数为联合分布下的平均损失期望，但一般不能得到。经验风险，是损失函数的直接算术平均。
当样本容量较大时，经验风险效果较好，趋于风险函数；样本较小时，会产生“过拟合”（对已知数据的预测得很好，对未知预测很差）。
结构风险最小化防止过拟合，在经验风险上加表示模型复杂度的正则化项（泛函，定义域为函数），模型越复杂其值越大。
训练误差与测试误差，后者主要是过拟合问题。

正则化项：
L0范数，向量中的非0元素个数
L1范数，向量中各元素绝对值之和
L2范数，向量中各元素的平方和再求平方根
奥卡姆剃刀原理，简单并能够解释已知数据的模型应首先选择。
交叉验证：数据分为训练集，验证集，测试集，验证集用于模型选择，测试集用于学习方法评估。
模型的不同主要是参数的不同。
S折交叉验证，已知数据切分成S个子集，用S-1个子集训练，剩下一个测试，S种重复进行。当S=N时为留一交叉验证，数据缺乏时适用。

泛化误差，评价方法学习到的模型对未知数据预测的误差。

生成模型：朴素贝叶斯，隐含马尔可夫
判别模型：直接学习决策函数或条件概率作为预测模型，准确率更高。k近邻，感知机，决策树，最大熵，支持向量机，条件随机场等。

分类问题：输出变量为有限离散，评价指标为精确率与召回率。标注问题，为分类的推广，输入为观测序列，输出为标记序列。标注问题通常的学习方法为隐马尔可夫模型、条件随机场。
回归问题，预测输入变量与输出变量之间的关系，特别当输入变量变化时输出的变化,其学习等价于函数拟合

梯度下降法，选择学习速率，根据梯度大小调整学习速率

感知机，输入为实例特征向量，输出为类别，取+1，-1，属于判别模型
f(x) = sign(w*x + b)
w*x + b = 0 对应于欧式空间的超平面，w是超平面法向量，b是截距。
感知机学习算法：选初值w0,b0；选（xi，yi），若yi(w*xi+b)<=0发生错误，w=w+n*xi*yi，b=b+n*yi；转第二步直到没有误分类点。采用不同初值或顺序，解不同。算法具有收敛性，也有对称形式。

KNN，对新输入实例，在训练数据集中，找到与该实例最邻近的k个实例，这k个实例多数属于某个类，就把该输入实例分为这个类。K近邻法没有显式的学习过程。
模型三要素：距离度量，k值选择，分类决策决定
距离度量，主要为Lp距离，p=1为曼哈顿距离，p=2为欧拉距离。
k值越小，近似误差小，但估计误差越大，容易发生过拟合。
kd树方法，减少计算距离的次数。
K-INN逆向最近邻，考虑周边点的最近邻包含该点
K-NN~,和最近邻的点互为最近邻的点

朴素贝叶斯法。将实例分到后验概率最大的类中，朴素，条件独立性假设

支持向量机SVM

C：对错误样例的惩罚程度，越大拟合度越高，越不想丢弃错误点，可能过拟合

gmma：rbf中是均方差，一般为类别数的倒数，反应数据波动大小。不同核函数，gamma定义不同，越大拟合越好，也越容易过拟合

集成学习ensemble learning

bagging：从训练集从进行子抽样组成每个基模型所需要的子训练集，对所有基模型预测的结果进行综合产生最终的预测结果

boosting提升方法：通过改变训练样本权重，学习多个分类器，并将分类器线性组合，提升分类性能。训练过程阶梯状。如何改变训练数据权值，如何将弱分类器组合成强分类器。

stacking：将训练好的所有基模型对训练基进行预测，第j个基模型对第i个训练样本的预测值将作为新的训练集中第i个样本的第j个特征值，最后基于新的训练集进行训练。同理，预测的过程也要先经过所有基模型的预测形成新的测试集，最后再对测试集进行预测：

AdaBoost算法

强可学习：一个概念，存在一个多项式学习，并正确率很高。弱可学习：一个概念，存在多项式学习，正确率仅比随机猜测略好。 两者等价，强可学习充要是弱可学习。提升方法，把多个弱分类器组合成强分类器。

如何改变训练数据权值？

提高前一轮被弱分类器错误分类的样本权值，降低被正确分类样本权值。

如何将弱分类器组合成强分类器？

多数表决方法，加大分类误差小的弱分类器权值(表决中大作用)，减小分类误差率大的权值。

算法流程：

输入：二分类的训练数据集 $T=\{(x_1,y_1),(x_2,y_2),\cdots,(x_N,y_N)\}$

$x_i \in \mathcal{X} \subseteq \mathbf{R}^n$ ， $y_i \in \mathcal{Y}=\{-1,+1\}$

输出：最终分类器$G(x)$

初始化训练数据的起始权值分布

$$D_1=(w_{11},\cdots,w_{1i},\cdots,w_{1N}) \quad w_{1i}=\frac{1}{N}, \quad i=1,2,\cdots,N$$

c步骤中，当训练误差<=0.5，系数将>=0，误差越小组合权值越大。

对m个弱分类器 m=1,2，。。M

a、在权值D_m下训练数据集，得到弱分类器

$$G_m(x):\mathcal{X}\rightarrow\{-1,+1\}$$

b、计算G_m的训练误差

$$e_m=P(G_m(x_i)\neq y_i)=\sum_{i=1}^{N} w_{mi} I(G_m(x_i)\neq y_i)$$

c、计算G_m的系数 $\alpha_m=\frac{1}{2}\log\frac{1-e_m}{e_m}$

d、更新训练数据集的权值分布

步骤 d

$$w_{m+1,i}=\begin{cases}\frac{w_{mi}}{Z_m}e^{-\alpha_m}, & G_m(x_i)=y_i\\ \frac{w_{mi}}{Z_m}e^{\alpha_m}, & G_m(x_i)\neq y_i\end{cases}$$

误分类的样本权值放大 $e^{2\alpha_m}=\frac{e_m}{1-e_m}$

$$D_{m+1}=(w_{m+1,1},\cdots,w_{m+1,i},\cdots,w_{m+1,N}) \quad w_{m+1,i}=\frac{w_{mi}}{Z_m}\exp(-\alpha_m y_i G_m(x_i))$$

Z是规范化因子

$$Z_m=\sum_{i=1}^{N} w_{mi}\exp(-\alpha_m y_i G_m(x_i))$$

构建弱分类器的线性组合，

$$f(x)=\sum_{m=1}^{M}\alpha_m G_m(x)$$

$$G(x)=\text{sign}(f(x))=\text{sign}\left(\sum_{m=1}^{M}\alpha_m G_m(x)\right)$$

AdaBoost训练误差分析每轮选取适当Gm使Zm最小，使训练误差下降最快

AdaBoost算法最终分类器的训练误差界为：

$$\frac{1}{N}\sum_{i=1}^{N}I(G(x_i)\neq y_i)\leqslant\frac{1}{N}\sum_{i}\exp(-y_i f(x_i))=\prod_{m}Z_m$$

exp(x)与根号(1-x)在0处泰勒展开，可得下面不等号

定理：二分类问题AdaBoost的训练误差界为：

$$\prod_{m=1}^{M}Z_m=\prod_{m=1}^{M}[2\sqrt{e_m(1-e_m)}]=\prod_{m=1}^{M}\sqrt{(1-4\gamma_m^{2})}\leqslant\exp\left(-2\sum_{m=1}^{M}\gamma_m^{2}\right)$$

$$\gamma_m=\frac{1}{2}-e_m$$

训练误差，指数级下降

定理：如果存在$\gamma>0$，对所有的m有$\gamma_m\geq\gamma$，则

$$\frac{1}{N}\sum_{i=1}^{N}I(G(x_i)\neq y_i)\leqslant\exp(-2M\gamma^{2})$$

AdaBoost算法，模型为加法模型，损失函数为指数函数，学习算法为向前分步算法

提升树(分类树、回归树)。以决策树为基函数

对二分类问题，将AdaBoost算法中的基本分类器限制为二类分类树

下面对于回归树问题：

$$f_M(x)=\sum_{m=1}^{M}T(x;\Theta_m)$$

已知训练数据集：

$$T=\{(x_1,y_1),(x_2,y_2),\cdots,(x_N,y_N)\},\quad x_i\in\mathcal{X}\subseteq\mathbf{R}^n$$

X为输入空间，y为输出空间，$y_i\in\mathcal{Y}\subseteq\mathbf{R}$

将X划分为J个互不相交的区域R1,R2,..Rj, 并且在每个区域上确定输出的常量c_j，那么，树可表示为：

$$T(x;\Theta)=\sum_{j=1}^{J}c_j I(x\in R_j)$$

$$\Theta=\{(R_1,c_1),(R_2,c_2),\cdots,(R_J,c_J)\}$$

J是回归树的复杂度即叶结点个数

首先确定初始提升树：$f_0(x)=0$

第m步的模型：$f_m(x)=f_{m-1}(x)+T(x;\Theta_m)$

其中，$f_{m-1}(x)$为当前模型，通过经验风险极小化确定下一棵决策树的参数θ_m

$$\hat{\Theta}_m=\arg\min_{\Theta}\sum^{N}L(y_i,f_{m-1}(x_i)+T(x_i;\Theta_m))$$

采用平方损失函数时：$L(y,f(x))=(y-f(x))^2$

$$\begin{aligned}&L(y,f_{m-1}(x)+T(x;\Theta_m))\\&=[y-f_{m-1}(x)-T(x;\Theta_m)]^2\\&=[r-T(x;\Theta_m)]^2\end{aligned}$$

1）Bagging + 决策树 = 随机森林

2）AdaBoost + 决策树 = 提升树

3）Gradient Boosting + 决策树 = GBDT

决策树

ID3，C4.5，CART 输入：训练集D，特征集A，阈值eps 输出：决策树T

1. 若D中所有样本属于同一类Ck，则T为单节点树，将类Ck作为该结点的类标记，返回T
2. 若A为空集，即没有特征作为划分依据，则T为单节点树，并将D中实例数最大的类Ck作为该结点的类标记，返回T
3. 否则，计算A中各特征对D的信息增益(ID3)/信息增益比(C4.5)，选择信息增益最大的特征Ag
4. 若Ag的信息增益（比）小于阈值eps，则置T为单节点树，并将D中实例数最大的类Ck作为该结点的类标记，返回T
5. 否则，依照特征Ag将D划分为若干非空子集Di，将Di中实例数最大的类作为标记，构建子节点，由结点及其子节点构成树T，返回T
6. 对第i个子节点，以Di为训练集，以A-{Ag}为特征集，递归地调用1~5，得到子树Ti，返回Ti

CART ：

CART树是二叉树，而ID3和C4.5可以是多叉树

CART在生成子树时，是选择一个特征一个取值作为切分点，生成两个子树

选择特征和切分点的依据是基尼指数，选择基尼指数最小的特征及切分点生成子树

剪枝：预防过拟合，从叶节点向上回溯，尝试对某个节点进行剪枝，比较剪枝前后的决策树的损失函数值。动态规划（树形dp）

随机森林算法(bagging + 决策树)，分类，回归

优点：准确率高，随机性不易过拟合，随机性抗噪，处理高维数据不用特征选择

处理离散数据连续数据无需规范化，需连速度快，并行化

缺点：决策树个数多时间空间大，不易解释类似黑盒

1. 从原始训练集中使用Bootstraping(有放回地抽样)方法随机有放回采样选出m个样本，共进行n_tree次采样，生成n_tree个训练集
2. 对于n_tree个训练集，我们分别训练n_tree个决策树模型
3. 对于单个决策树模型，假设训练样本特征的个数为n，那么每次分裂时根据信息增益/信息增益比/基尼指数选择最好的特征进行分裂
4. 每棵树都一直这样分裂下去，直到该节点的所有训练样例都属于同一类。在决策树的分裂过程中不需要剪枝
5. 将生成的多棵决策树组成随机森林。对于分类问题，按多棵树分类器投票决定最终分类结果；对于回归问题，由多棵树预测值的均值决定最终预测结果

#ANN人工神经网络

卷积神经网络CNN，参考 https://www.cnblogs.com/skyfsm/p/6790245.html

层级结构：输入层，卷积计算层，ReLU激励层，池化层，全连接层

一、数据输入层：原始图像数据预处理。去均值：把输入数据各个维度都中心化为0，如下图所示，其目的就是把样本的中心拉回到坐标系原点上。归一化：各维度特征幅度归一化到同样的范围。PCA/白化：用PCA降维；白化是对数据各个特征轴上的幅度归一化。

二、卷积计算层（Conv）：两种操作：局部关联，每个神经元看做一个滤波器(filter)；窗口(receptive field)滑动， filter对局部数据

计算。三种名词：深度、步长、填充值。

有多少个神经元，深度就是多少。

填充值，让滑动窗口能够吧所有像素遍历完，而添加的行与列的像素，使它为滑动窗口大小整数倍。比如5*5的图像，滑动窗口2*2，步长取2，这样需要添加填充值，使得编程6*6矩阵,就可以不重复地遍历完所有像素。

输出宽度 = (输入宽度 - 滤波器宽度)/步长 + 1

一个滤波器参数，有 宽度*宽度+1

卷积计算，输入与神经元内积加上偏移（内积不是矩阵乘法，而是对应元素相乘然后相加）。深度多大，就输出多少个。

参数共享机制，每个神经元连接数据窗的权重固定，每个神经元只关注一种特征（垂直边缘，水平边缘，颜色，纹理等），所有神经元加起来就是图像特征的提取集合。

三、激励层：把卷积层输出结果做非线性映射，ReLU修正线性单元，收敛快，求梯度简单。

激励层实践经验： ①不要用sigmoid！不要用sigmoid！不要用sigmoid！② 首先试RELU，因为快，但要小心点 ③ 如果2失效，请用Leaky ReLU或者Maxout ④ 某些情况下tanh倒是有不错的结果，但是很少。

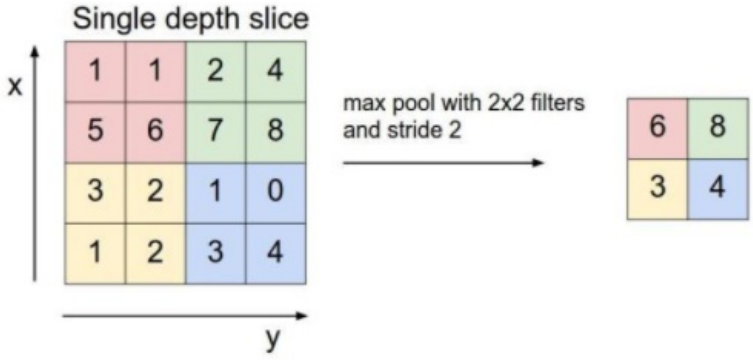
四、池化层：池化层夹在连续的卷积层中间， 用于压缩数据和参数的量，减小过拟合.(压缩图像)。

1.特征不变性，压缩只是去掉无关信息。

2.特征降维。把重要特征提取出来。

3.防过拟合。

如Max pooling ,average pooling.实际前者常用。



五、全连接层：起到分类器作用，两层之间所有神经元都有权重连接，通常全连接层在卷积神经网络尾部。也就是跟传统的神经网络神经元的连接方式是一样的。

六、训练方法

1. 先定义Loss function，衡量和实际结果之间差距。

2. 找到最小化损失函数的W和b， CNN中用的算法是SGD（随机梯度下降）。

fine-tuning，已用于其它目标，训练好的模型的权重或部分权重，收敛较快。

牛顿法，拟牛顿法，求解无约束最优化问题

keras网络：

batch,梯度下降方式，一种是批梯度下降Batch gradient descent，遍历全部数据计算损失函数；另一种随机梯度下降stochastic gradient descent，每一个数据就算一下损失函数，速度快，收敛性不太好，震荡。小批的梯度下降mini-batch，结合两者。

epochs,训练过程中，数据被使用“轮”多少次

rnn:一个一阶的张量[1,2,3]的shape是(3,);一个二阶的张量[[1,2,3],[4,5,6]]的shape是(2,3);一个三阶的张量[[[1],[2],[3]],[[4],[5],[6]]]的shape是(2,3,1)。

input_shape的三个维度samples, time_steps, features

features: 是一个原始样本的特征维数， 对你的样本 6

time_steps: 是输入时间序列的长度，即用多少个连续样本预测一个输出。如果你希望用连续m个序列（每个序列即是一个原始样本），那么就应该设为m。

当然，特殊情况是m=1

samples：经过格式化后的样本数。假设原始样本(3000*6), 你选择features=6, time_steps=m,则samples=3000/m

无论你如何设置time_steps需要注意，原始样本集合是二维向量， 但网络的输入的样本集必须是三维张量（单个样本是二维向量）