关于集成学习、模型蒸馏、模型部署的理解

1. 集成学习

集成学习的思想：通过将弱学习器的偏置和/或方差结合起来，从而学习一个强学习器，从而获得更好的性能

基于投票思想的多数票机制

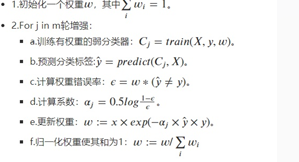
分别学习多个弱学习器，对每个弱分类器输出预测结果，每个样本取投票数最多预测的那个预测为改样本最终分类预测，也就是少数服从多数。

Bagging思想

它与投票机制不同在于投票机制用的是全部样本，Bagging方法则是使用全部样本的一个随机抽样，每个分类器都是使用不同的样本进行训练

Boosting思想

先学习一个弱分类器，然后在弱分类器的基础上，基于预测残差的思想，错误的样本给的权重更高，正确的样本给的低的基础上，不断学习分类器，最终将多个分类器整合起来的方法

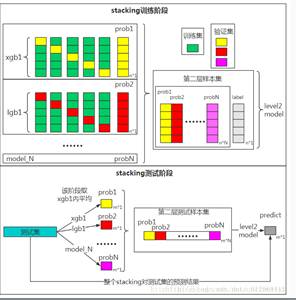


Bagging与它相比，bagging是降低方差的，Boosting是降低偏差的，bagging它是可以并行训练的，Boosting只能顺序学习，Bagging基于投票机制，Boosting则是预测残差的思想，错误的样本给的权重更高，正确的样本给的低思想。

Stacking

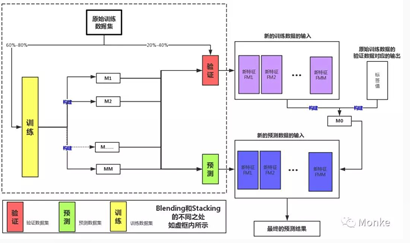
Stacking集成算法可以理解为一个两层的集成，第一层含有一个分类器，把预测的结果(元特征)提供给第二层， 而第二层的分类器通常是逻辑回归，他把一层分类器的结果当做特征做拟合输出预测结果。

Stacking第一层时候在预测结果时候会有一个交叉验证阶段，它会出现一个数据泄露的问题



Blending

Blending它则是为了解决stacking中的数据泄露问题的，训练集不是通过K-Fold的CV策略来获得预测值从而生成第二阶段模型的特征，而是建立一个Holdout集，例如10%的训练数据，第二阶段的stacker模型就基于第一阶段模型对这10%训练数据的预测值进行拟合，即把Stacking流程中的K-Fold CV 改成 HoldOut CV。



•**Blending的优点：**

•比stacking简单（因为不用进行k次的交叉验证）

•避开了一个信息泄露问题

•在团队建模过程中，不需要给队友分享自己的随机种子

•**缺点：**

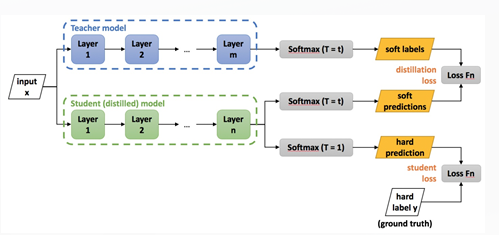
•使用了很少的数据

•blender可能会过拟合（其实大概率是第一点导致的）

•stacking使用多次的CV会比较稳健

1. 模型蒸馏

它是将一个较大的、效果较好的ensemble模型压缩成一个轻量级的模型，同时解决labeled data不足问题。先用过ensemble模型对大量的unlabeled data 打标签得到label，然后对这部分样本训练一个轻量的模型。它包括两个模型teacher model 和student model

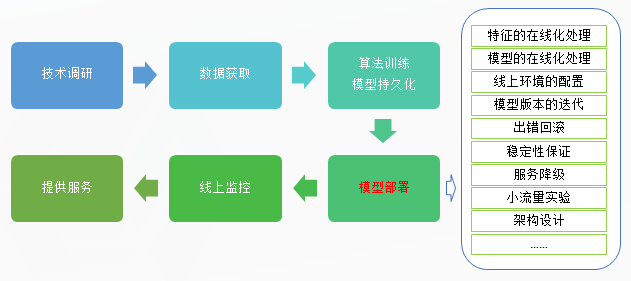


它要求student model在训练时，除了要使得其输出与true label尽量拟合外，也要求最后softmax之前的logits和teacher network中对应的logits，在一个temperature参数T的变换下，能尽量接近。



其中Zs是student,Zt则是老师的

1. 模型部署



在线项目流程

模型的服务方式

离线预测

在线预测

部署方式

server framework + deeplearning framework api

server framework + deeplearning serving

我们需要考虑的问题

是否是长耗时操作

是否是一次性操作

接口控制

处理时长有多少

模型的大小

服务降级策略

回滚兼容策略

实验方式

多模型是否并行或者串行等等