**Thinking1: 在CTR点击率预估中，使用GBDT+LR的原理是什么？**

答: CTR预估中用的最多的模型是LR,LR很容易并行化,但是线性模型的学习能力有限,需要

大量特征工程预先分析出有效的特征、特征组合.而GDBT是一种常用的非线性模型,它

基于继承学习中的boosting思想,每次迭代都在减少残差的梯度方向新建立一颗决策

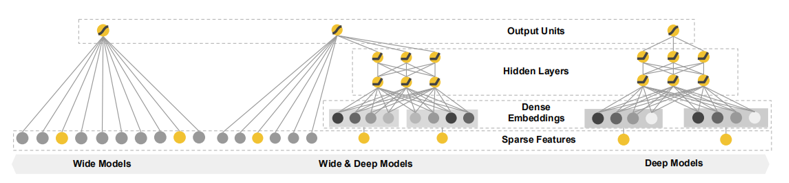
树.GDBT的思想使其具有天然优势可以发现多种有区分性的特征以及特征组合,决策树的路径可以直接作为LR的输入特征使用,省去了人工寻找特征,组合特征的步骤.

**Thinking2: Wide & Deep的模型结构是怎样的，为什么能通过具备记忆和泛化能力（memorization and generalization）**

答: 首先,记忆能力指从历史数据中发现item和特征之间的相关性,而泛化能力指相关性的

传递,发现再历史数据中很少或者没有出现的新的特征组合.

Wide & Deep包括两部分: 线性模型 + DNN部分.



Wide & Deep模型中,Wide指的就是LR模型,LR模型是线性模型,是用来找数据中的相

关性,因而可以解释成有记忆能力.而Deep模型,就是DNN模型,是一种深度模型,深度

神经网络模型,通常需要的是输入是连续的稠密特征,对于稀疏,高维的类别特征,通常

将其转化为低维的向量,进而进行组合可以学习到更深层次的隐藏特征.也就是具有泛化能力.

**Thinking3: 在CTR预估中，使用FM与DNN结合的方式，有哪些结合的方式，代表模型有哪些？**

答: 在CTR预估中,常常使用FM和DNN结合的方式,其主要分为并行结构和串行结构.

并行结构指,FM部分和DNN部分分开计算,只在输出层进行一次融合得到结果,常见的模

型有DeepFM,DCN,Wide&Deep

串行结构指,将FM的一次项和二次项结果(或者其中之一)作为DNN部分的输入,经DNN

得到最终结果,常见的模型有NFM,PNN,AFM

**Thinking4: Surprise工具中的baseline算法原理是怎样的？BaselineOnly和KNNBaseline有什么区别？**

答: BasicOnly是基于统计的基准预测线打分,主要算法原理是:

需要注意的是,如果用户u位置,则就默认为0,这个同样适用于商品和.

而BaselineOnly和KNNBaseLine的区别主要是一个是基于统计的,一个是基于协同过滤算法的baseline算法.

**Thinking5: GBDT和随机森林都是基于树的算法，它们有什么区别？**

答: 1.随机森林采用的bagging思想，而GBDT采用的boosting思想。这两种方法都是

Bootstrap思想的应用，Bootstrap是一种有放回的抽样方法思想。虽然都是有放回的

抽样，但二者的区别在于：Bagging采用有放回的均匀取样，而Boosting根据错误率来取样（Boosting初始化时对每一个训练样例赋相等的权重1／n，然后用该算法对训练集训练t轮，每次训练后，对训练失败的样例赋以较大的权重），因此Boosting的分类精度要优于Bagging。Bagging的训练集的选择是随机的，各训练集之间相互独立，弱分类器可并行，而Boosting的训练集的选择与前一轮的学习结果有关，是串行的。

2.组成随机森林的树可以是分类树，也可以是回归树；而GBDT只能由回归树组成。组

成随机森林的树可以并行生成；而GBDT只能是串行生成。

3.对于最终的输出结果而言，随机森林采用多数投票等；而GBDT则是将所有结果累加

起来，或者加权累加起来。

4.随机森林对异常值不敏感；GBDT对异常值非常敏感。

5.随机森林对训练集一视同仁；GBDT是基于权值的弱分类器的集成。

6.随机森林是通过减少模型方差提高性能；GBDT是通过减少模型偏差提高性能。

**Thinking6: 基于邻域的协同过滤都有哪些算法，请简述原理?**

答: 主要有UserCF和ItemCF

UserCF:推荐和当前用户相似度搞的N个用户产生过行为的物品给当前用户

ItemCF:推荐和当前用户历史上行为过的物品相似的物品给当前用户

UserCF原理:

首先利用行为的相似度计算用户的相似度,然后找到和目标用户兴趣相似的用户集合,然后用户u对物品的相似度,等价于K个邻居对物品的相似度,最后为用户u生成推荐列表.

ItemCF原理:

首先利用行为的相似度计算物品的相似度,然后计算物品之间的相似度,用户u对物品的兴趣度,等价于物品的K个邻居物品,收到用户u的兴趣度,最后为用户u生成推荐列表.

**Action1: 使用Wide&Deep模型对movielens进行评分预测?**

答:



**Action2: 使用基于邻域的协同过滤（KNNBasic, KNNWithMeans, KNNWithZScore, KNNBaseline中的任意一种）对MovieLens数据集进行协同过滤，采用k折交叉验证(k=3)，输出每次计算的RMSE, MAE?**

答:

