

CTR - Wide&Deep Learning

- [CTR - Wide&Deep Learning](#)
 - [简介](#)
 - [优势](#)
 - [Wide&Deep Learning 结构](#)
 - [Wide Component](#)
 - [Deep Component](#)
 - [joint training](#)
 - [推荐系统实现](#)
 - [Data Generation](#)
 - [Model Training](#)
 - [Model Serving](#)
 - [实验验证](#)

简介

Wide&Deep learning最终的目的是为了服务于google play的推荐系统，所以本文要做的其实是一个推荐系统的设计，推荐系统的主要工作是查询相关items，对其进行排序而后返回推荐列表，所以推荐系统当中最为关键一步的就是推荐物品的排序，Wide&Deep Learning(WDL) 就是为解决这个问题而产生的。WDL将一个线性模块和一个神经网络模块组合为一个模型，可以更好的预测CTR辅助进行推荐排序，该方法已经开源在tensorflow当中；

交叉特征通常有两种生成方式：

- 记忆式 (memorization)
根据历史数据当中不同特征同时出现的频率来学习交叉特征，常见的做法是对大量的特征做cross-product 的特征转换，这种方式有效且具备可解释性；
- 生成式 (generalization)
生成式的方法是基于相关性可传递的性质，挖掘出一些深层难以发现的特征组合，常见的方式有两种方式：
 - 人工：需要具备专业的业务领域知识和大量的特征工程的操作；
 - DNN：将高维稀疏的输入转化为低维稠密的embeddings再通过DNN进行训练，可以挖掘更深层的交叉特征；采用这种方法的主要缺陷是在输入过于稀疏时有可能出现类似于欠拟合的情况而推荐一些不相关的物品；

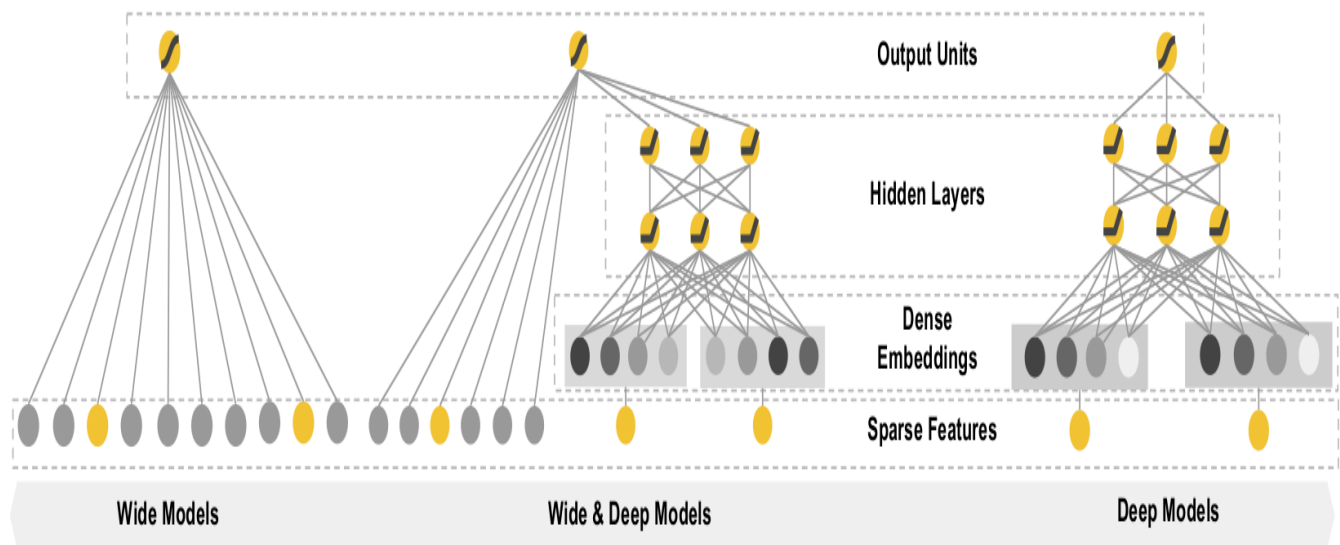
WDL将一个基于cross-product的线性模型（单纯的cross product只能学习到用户有过的行为）和一个基于embeddings（FM/NN都是基于embedding的，可以学习到不易发现的交叉组合）的前馈神经网络模型结合到一起进行联合训练，也就相当于结合了记忆式和生成式两种方法来生成交叉特征，再对生成的交叉特征进行建模学习。

优势

- 线性模型和深度模型联合训练
- 基于google play进行实现和评估

- 以TensorFlow API的形式开源代码

Wide&Deep Learning 结构



如上图所示，左侧是一个单独的线性模型，右侧是单独的深度模型，中间是两者的联合模型 WDL。

Wide Component

线性模型的特征包括原始输入特征和特征转换生成的特征

$$y = W^T x + b$$

其中x包括原始特征和人工生成的交叉特征(对多个不同的one-hot-encoded离散变量做乘积)

$$\phi_k(x) = \prod_{i=1}^d x_i^{c_{ki}}$$

其中，x为输入样本，共有d个维度，若某个维度是one hot encoding的向量，则 $c_{ki} = 1$ ，否则 $c_{ki} = 0$

Deep Component

Deep主要包括三层：input layer, embedding layer和hidden layer，hidden layer的结果输出到output layer;

Embedding层把原本高维稀疏的输入转化为低维、密实的向量，然后输入到隐藏层进行训练。

input layer的input vector经过embedding layer变成了 $a^{(0)}$ ，然后输入到DNN当中，若令l表示神经网络层的深度，则有：

$$a^{(l+1)} = \sigma(W^{(l)} a^{(l)} + b^{(l)})$$

joint training

WDL的wide和deep部分同时训练，采用 mini batch SGD 调节参数值，但是wide和deep两部分的优化器选择不同，wide部分选用L1正则化的FTRL的优化器， deep部分选用AdaGrad优化器。

模型的预测结果输出为：

$$P(Y = 1|x) = \sigma(w_{wide}^T[x, \phi(x)] + w_{deep}^T a^{l_f} + b)$$

推荐系统实现

本文推荐系统主要包括三个模块：Data Generation， Model Training 和 Model Serving。

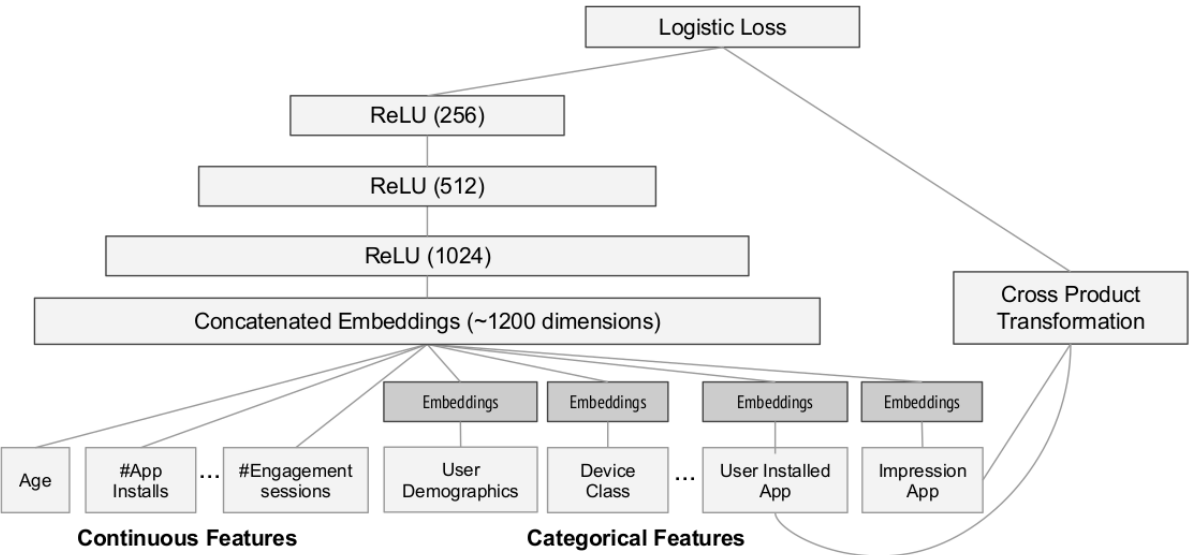
Data Generation

数据生成部分主要是从用户属性信息和行为日志当中提取训练数据集，每条数据对应的是一个用户以及该用户想要获取的app，若该用户安装了该app，则对应的标签字段为1，否则为0。

对于字符型的字段，将其映射为整数ID；对于连续型的字段，进行min-max标准化映射到[0, 1]区间内。

Model Training

下图是用于app推荐的WDL模型结构。
wide部分是用用户想要的app和用户安装的app之间的交叉特征；
deep部分将每个的离散属性转化为32维的embeddings，并和所有的连续特征整合到一起，共计约1200维，然后输入到3层ReLU的隐藏层；
output是一个sigmoid函数；



Model Serving

当用户请求到来时，数据库返回查询列表，通过WDL预估该用户对每个app的下载概率，根据该输出值对app列表进行排序，返回给用户。

实验验证

连续3周的在线A/B测试，随机选取1%的用户采用wide, deep, wide & deep，1%的用户延续以前的模型，训练数据超过5000亿历史数据，结果显示：

wide & deep 在线下载量增加了3.9%，离线AUC达到0.728，提升了约0.002，

单wide模型在线没有任何提升，离线auc达到0.726

单deep模型的auc不升反降，达到0.722，在线下载量却增加了2.9%.