

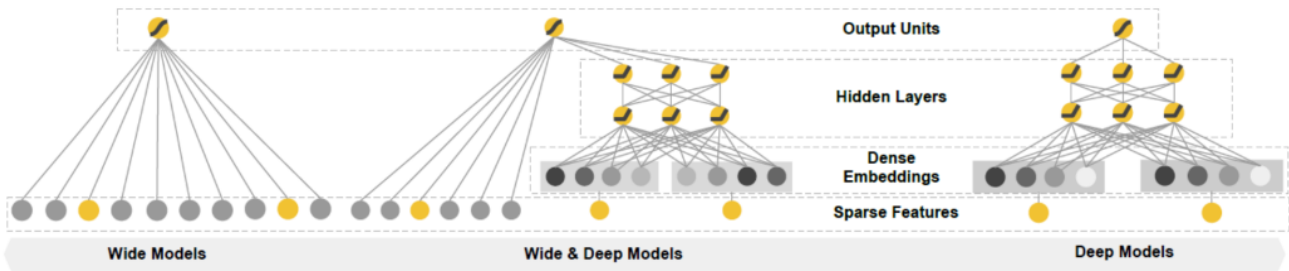
# [深度模型] 经典Wide & Deep模型架构介绍

原创 xxxhuang 推荐算法学习笔记 5月9日

本文介绍一个每个搞推荐的人都必须会的经典推荐深度模型 Wide & Deep。完整的 paper名称是《Wide & Deep Learning for Recommender Systems》

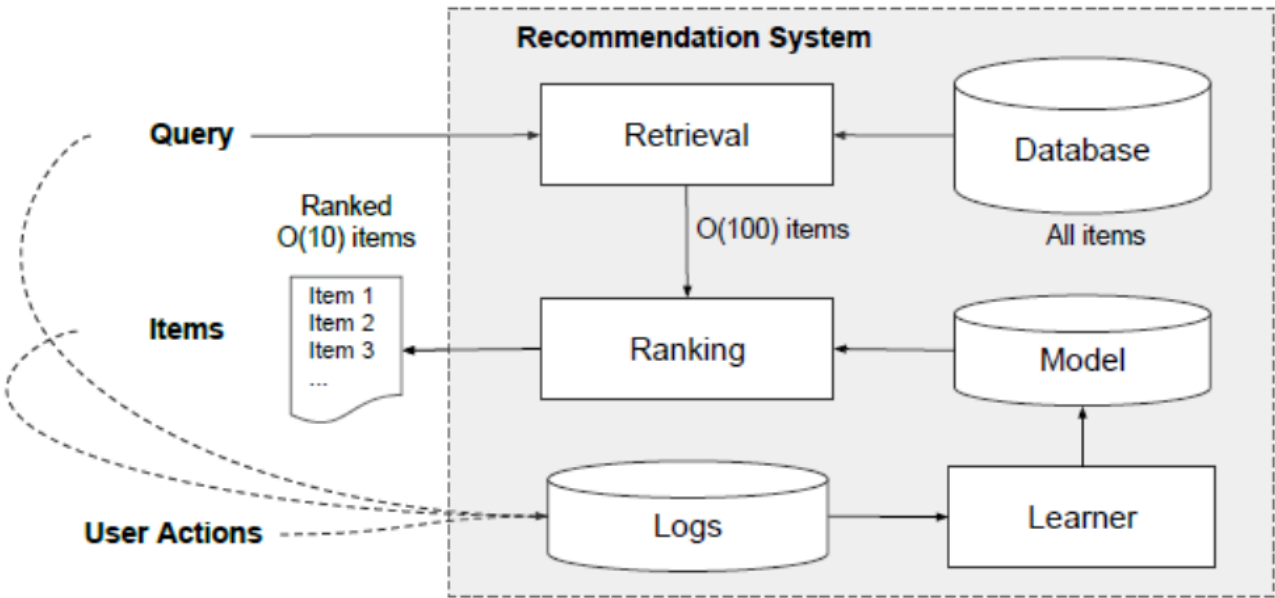
## 一. 介绍

wide & deep的模型架构如下图所示



可以看到wide & deep模型分成wide和deep两部分。wide部分就是一个简单的线性模型，deep部分是一个前馈神经网络模型。将线性模型和前馈神经网络模型合并在一起训练。

## 二. 推荐系统架构



当一个用户请求过来的时候，推荐系统首先会从海量的item里面挑选出 $O(100)$  个用户可能感兴趣的item(召回阶段)。然后这  $O(100)$ 个item将会输入到模型里面进行排序。根据模型的排序结果再选择出topN个item返回给用户。同时，用户会对展示的item进行点击，购买等等。最终，用户的feature，上下文feature，item的feature和user

action会以log的信息保存起来，经过处理后生成新的训练数据，提供给模型进行训练。paper的重点放在使用wide & deep架构为基础的排序模型。

### 三. Wide部分

wide部分其实就是一个简单的线性模型  $y = \mathbf{w}\mathbf{x} + b$ 。y是我们的预测目标， $\mathbf{x} = [x_1, x_2, \dots, x_d]$  是d个feature的向量， $\mathbf{w} = [w_1, w_2, \dots, w_d]$ 是模型的参数，b是bias。这里的d个feature包括原始的输入feature和经过转换的feature。

其中一种很重要的转换feature叫做cross-product转换。假如x1是性别，x1=0表示男性，x1=1表示女性。x2是爱好，x2=0表示不喜欢吃西瓜，x2=1表示喜欢吃西瓜。那么我们就可以利用x1和x2构造出新的feature，令 $x_3 = (x_1 \&\& x_2)$ ，则x3=1表示是女生并且喜欢吃西瓜，如果不是女生或者不喜欢吃西瓜，则x3=0。这样经过转换的来的x3就是cross-product转化。这样转换的目的是为了获取交叉特征对预测目标的影响，给线性模型增加非线性。

### 四. Deep部分

deep部分就是前馈神经网络模型。对于高维稀疏的分类特征，首先会转化成低维的稠密的向量，然后作为神经网络hidden layers的输入进行训练。Hidden layers的计算公式如下

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

f是激活函数(例如ReLU)，a是上一个hidden layer的输出，W是要训练的参数，b是bias

### 五. Wide和Deep一起训练

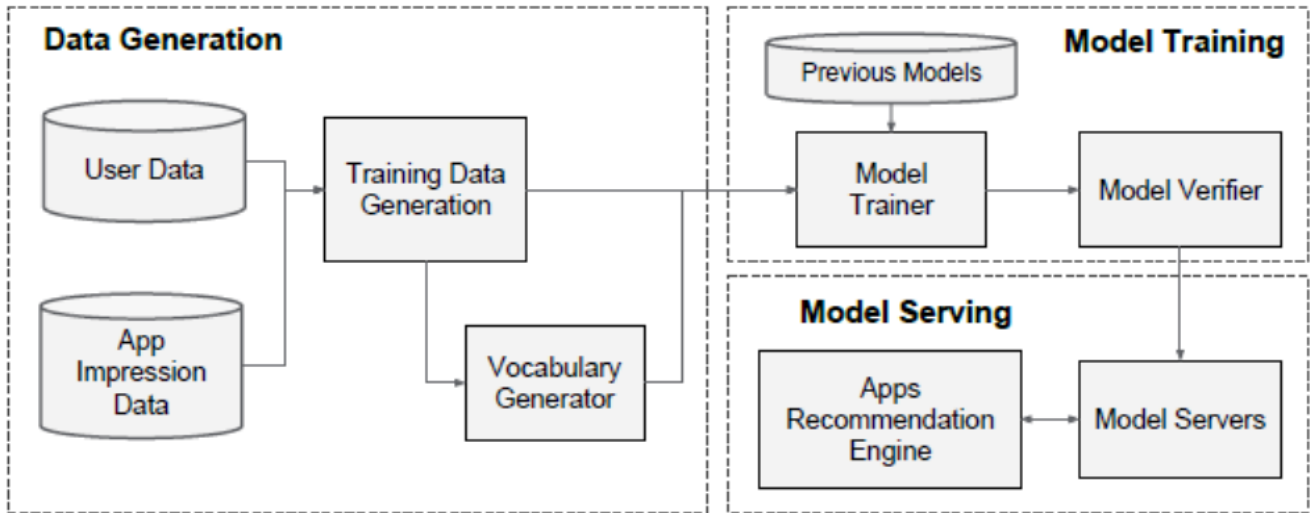
通过weight sum的方式将wide和deep的输出组合起来，然后通过logistic loss函数联合起来一起训练。对于wide的部分，一般采用FTRL进行训练。对于deep的部分则采用AdaGrad进行训练。

对于一个逻辑回归问题，预测公式如下所示

$$P(Y = 1|\mathbf{x}) = \sigma(\mathbf{w}_{wide}^T[\mathbf{x}, \phi(\mathbf{x})] + \mathbf{w}_{deep}^T a^{(lf)} + b) \quad (3)$$

### 六. 系统实现

推荐系统的实现一共分成了三个阶段：数据生成，模型训练和模型服务。如下图所示



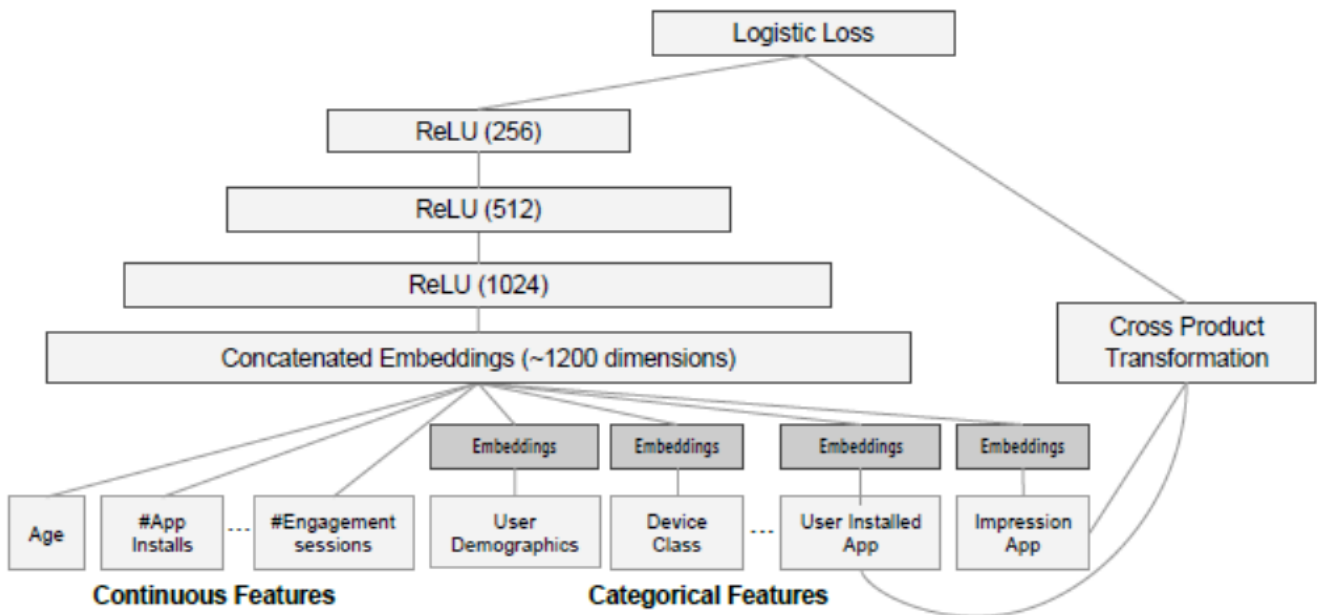
### (1) 数据生成阶段

在这个阶段，最近N天的用户和item将会用来生成训练数据。每条展示过的item将会对应有一个目标label。例如1表示用用户点击过，0表示用户没点击过。

图中的Vocabulary Generation主要用来做数据转换。例如需要把分类特征转换成对应的整数Id，连续的实数特征将会按照累积概率分布映射到[0, 1]等等。

### (2) 模型训练阶段

在数据生成阶段我们产生了包含稀疏特征，稠密特征和label的训练样本，这些样本将作为输入放入到模型里面训练。如下图所示



wide的部分包含了经过Cross Product转换的特征。对于deep的部分，分类特征首先会经过一层embedding，然后和稠密的特征concatenate起来后，经过3层的hidden layers，最后和wide部分联合起来通过sigmoid输出。

paper中还提到，因为google在训练的时候训练样本数超过5000亿，每次所有样本重新训练的成本和延迟非常大。为了解决这个问题，在初始化一个新的模型的时候，将会使用老模型的embedding参数和线性模型的weight参数初始化新模型。

### (3) 模型服务阶段

确认训练没问题以后，模型就可以上线。对于用户的请求，服务器首先会选出用户感兴趣的候选集，然后这些候选集将会放入到模型里面进行预测。将预测结果的分数从高到低排序，然后取排序结果的topN返回给用户。

## 七. 总结

以上便是wide & deep模型架构的全部内容。如有问题，欢迎随时联系~