基于深度学习模型Wide&Deep的推荐

凡人机器学习 2018-11-26

本实验选用数据为UCI开源数据集，仅用于学习，请勿商用）

代码使用的是github的开源代码：https://github.com/tensorflow/models/tree/master/official/wide_deep

我是用的环境是PAI-DSW，目前在邀测：https://pai.data.aliyun.com/notebook#/

Wide&Deep推荐算法出自一篇论文《Wide&Deep Learning for RecommenderSystems》，Wide&Deep由两部分组成，分别是Wide和Deep。先来说wide，表示的是generalized的推荐系统，传统的推荐系统都是通过线性算法基于离散特征来做推荐的。Wide推荐通常是这样的：系统通过获得用户的购物日志数据，包括用户点击哪些商品，购买过哪些商品，然后通过one-hot编码的方式构成离散特征或者通过对业务的理解衍生出一些特征，并进行计算，类似于本系列文章第二篇。这种wide推荐方式有非常多的好处，比如对于大规模的稀疏数据有很好的效果，而且模型的解释性很强。什么叫模型的解释性呢？以逻辑回归为例，每个特征都对应模型中的一个权重值，每个特征的权重值的大小跟这个特征对结果的影响是有关的。那么wide方式同样有很多缺点，比如我们一直强调的，特征衍生需要很多人为操作，需要专家经验，另外这种推荐只对用户操作过的商品有效。

接着讲下deep，这里的deep表示的是通过深度学习学习出来的一些向量，这些向量是隐性特征，往往是没有明确可解释性的。这些向量也可以作为特征的一部分参与到训练中。通过deep方式产生的特征会有以下好处，其一可以拟补人为提取特征造成的人力思考维度的限制，试想下一个人可以轻易的思考出二阶乘法的结果，如果是五阶呢？其二这部分特征是深度学习框架自动生成的，无需人力干预。

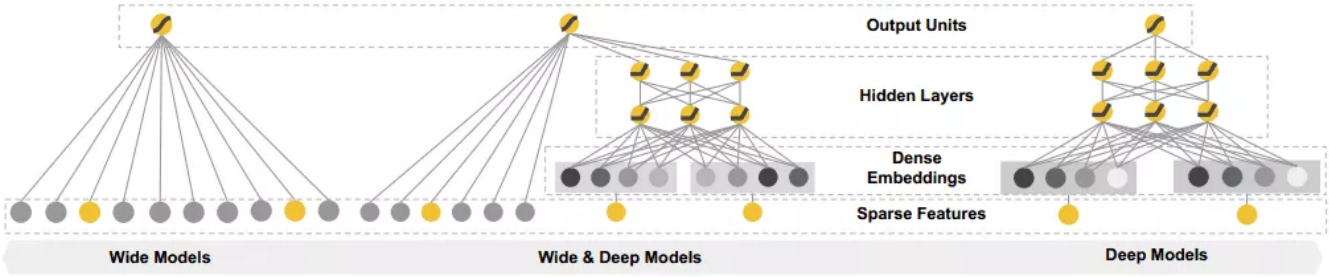


Figure 1: The spectrum of Wide & Deep models.

既然Wide和Deep算法各有千秋，那如果可以将两种算法作为组合，那么一定可以生成更有效的推荐场景的模型，本文就介绍如何在PAI-DSW上实现基于Wide&Deep的预测。

一、业务场景描述

本节使用的是PAI-DSW云端深度学习训练平台和PAI-EAS模型服务平台，使用的是一份开源的基于人的各种背景的统计数据，解决的问题是基于人的各种基础数据预测每个人收入是否

会超过50K。

本实验的全部代码和数据已经内置于PAI-DSW，只要打开DSW就可以安装下方的教程运行实验。

二、数据集介绍

数据源：引用UCI开源数据源，https://archive.ics.uci.edu/ml/datasets/Census+Income
具体特征字段如下：

字段名	含义	类型	描述
age	对象年龄	double	对象的年龄大小
workclass	工作性质	string	自由职业者、私企、企业人员、政府工作者、无业游民等
fnlwgt	连续数据	double	-
education	学历	string	学士、说是、博士、11th、10th、1s-4th等等
education-num	教育年限	double	教育年限
marital-status	婚姻状况	string	单身、未婚、离异等等
occupation	职业	string	工程师、农民、销售等等
relatonship	家庭角色	string	妻子、父亲、没家庭等等

字段名	含义	类型	描述
race	人种	string	亚裔、白人、黑人等等
sex	性别	string	女性、男性
capital-gain	连续数据	double	-
capital-loss	连续数据	double	-
hours-per-week	连续数据	double	-
native-country	祖籍国家	string	美国、哥伦比亚、英格兰、加拿大等等

目标字段：income是否超过50k

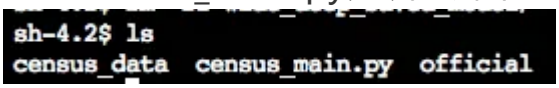
三、数据探索流程

首先进入PAI-DSW，找到左侧的Demo文件夹，下载Wide&Deep数据集及代码包。

（1）工程描述

首先看下整个工程，

- 包含一个census_data文件夹，里面包含一个训练数据和一个测试数据
- official文件夹是一个工具包
- census_main.py为训练脚本

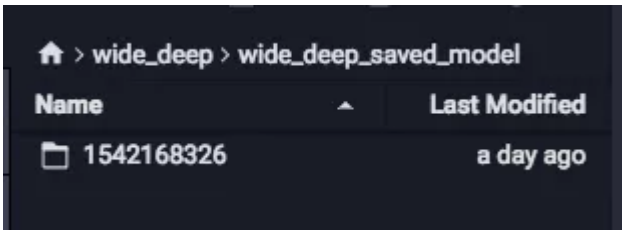


（2）训练模型

打开一个terminal环境，执行

```
python census_main.py --export_dir wide_deep_saved_model
```

wide_deep_saved_model为输出模型所在的文件夹，训练完在文件目录下会找到相应文件，打开后可以看到checkpoint：



把这个checkpoint的号记住。

（3）模型预测

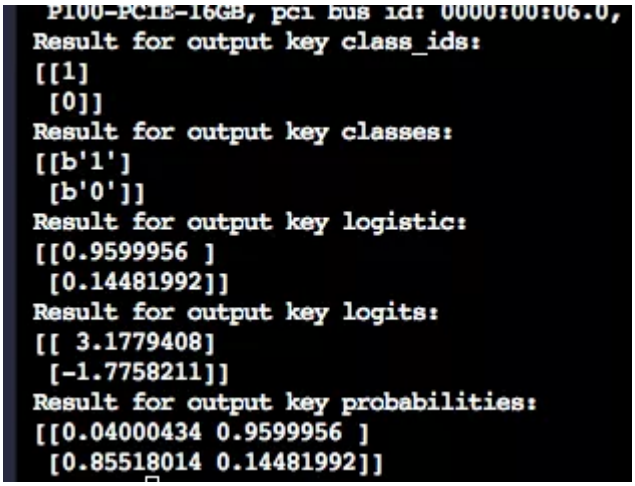
现在已经生成了模型的checkpoint输出，接下来进入terminal，运行以下脚本：

```
saved_model_cli run --dir wide_deep_saved_model/${模型checkpoint号码}/ --tag_set se
```

根据本文的案例可以执行以下脚本拿到预测结果：

```
saved_model_cli run --dir wide_deep_saved_model/1542168326/ --tag_set serve --sign
```

输入了两条预测数据，最终拿到预测结果：



输入了两条预测数据，可以得到预测输出，第一条预测结果为1，第二条结果为0，可以通过output key probabilities判断（注：矩阵第一行对应第一个预测结果，第二列0.9599956>第一列0.04000434，所以第一个预测结果是1。同理第二个预测结果是0）。

可以通过代码official/wide_deep/census_dataset.py来看具体的特征工程的特征和目标值的构建，目标列>50k时目标值为1，目标列<50k时目标值为0。

于是预测结果第一条的人的预测收入为>50k，预测结果第二条的人的预测收入<50k。

（4）模型在线部署

生成的模型是Tensorflow的标准模型格式，可以通过PAI-EAS将模型部署成Http服务供调用。

后续流程可以参考在线预测文档：https://help.aliyun.com/document_detail/92917.html

部署成在线服务之后，这样就可以做到模型跟用户自身的业务结合，完成PAI模型训练和业务应用的打通。