DEEP CTR讲稿

这篇文章就是要解决CTR prediction的一个任务。CTR即CLICK per visit，这个丁洪鑫和张级昃都给我详细讲述过，

根据用户特征准确地预估CTR，才能更好得完成RTB(real time biding )的任务。

第一张ppt是我理解下来，一个简单的流程图：一个预测任务：数据特征比如IP地址, region,city, 然后经过one hot encoding将这一些数据变成一长串非0 即1 的二进制特征。通过不同的MODEL，经过sigmoid函数，值域变化到0-1之间，阈值设定为0.5。>0.5就是1，<0.5就是0，进行预测了。训练好模型后，通过evaluate 来比较模型之间的好坏。

这个都是大家都很了解的过程，我就稍微提一下。

然后讲这篇论文：

在related work 中我们可以看到，CTR问题，在工业界最常用的就是：线性模型——logistics regression。

但是不可避免的问题有2：

1. 随着特征值的增多， feature的维数越来越高，会造成很大的计算负担，
2. 不同的特征之间如何组合的问题。

比如用户标签user tag 和 creative id 这两个不同的特征之间有内部联系，用它们一起对CTR预测的时候，应该考虑两个特征值结合后的预测效果。

而且：(挑选哪两个特征，来结合预测)；(怎么结合)都是很麻烦的问题。

这就是Combination feature的问题。

然后就讲这篇论文怎么解决这个问题：

这篇文章针对这个问题，提出了两个改进。

一个是在 embedding（嵌入） 方式上的改进；一个是在模型上的改进

第一个Embedding的改进：不要one-hot了，改成另一种FM\_embeding 方式。

第二个是模型改进：不要Logistic Regression了，改成deep neural network.深度神经网络。

其实第二个模型改进囊括了第一个，这个放到后面讲

我这边先讲一下第一个embedding的改进。Embedding大家都懂么？

**就是把输入变成机器能够理解的形式，比如最简单的embeddding就是onehot，特征变成010101…**

Embed 这个词中文解释为嵌入，理解成把一个很长feature用公式压一压，变成一个很短的feature。

那怎么实现这个embedding：

论文方法:叫做FM模型 Factorisation model——分解模型。最早是用作协同过滤推荐，现在被认为是一种最为成功的embeding手段了。它不仅能将一个很长的高维度特征变成低纬度特征，达成降维效果，而且通过自动学习出交叉特征.因为这种特性,FM在CTR预估上的效果会远超LR。

这个FM模型达成的效果是什么：

最直观的说：比如有一个用户的URL特征：<http://mail.163.com/index.html>

比如有100w种取值，那么这个特征的维度就是100w维，其中有5个维的取值是1

分别是mail; 163;com ;index;htmL

现在我通过FM算法学习以后，100w的维度就变成了5维！

然后我们具体的来说：FM是怎么实现的：

这是那篇论文中给出的公式：你发现，光看这些公式，看不明白。

但是你参考这幅图，听我讲，就能明白。

这幅图就是模型的摘要图

上面部分仍旧是Logistic Regression

花头就在一开始的下面部分的embedding里面。

其实公式没讲什么，这个降维就是一个变量代换！

从变量X变化到变量Z，

变量X是一个sparse的变量：

变量Z是一个dense的变量：

大家听我举个例子就明白了。

我现在有一个输入，输入里有两个特征，一个用户ID,一个是用户的兴趣点，每个用户的ID可以用2维表示，每个用户有一个兴趣点。

X这一层就是one-hot embedding 的结果：比如前者20维，里面有两个值是1，后者10维里面有一个1。

X=【（00000000010000001），（010000000000）】30维

我现在目标是把X它变成一个低维度的特征Z：

Z=其实就是5维（2+1+1+1）维。加两个w0截距量，所以加个2

用什么？FM模型，分解模型！，那就在当中划一刀。

前面乘以个W0,(1) 后面乘一个W0(2)

大家想一想这两个个矩阵是几维？

W0(1)=3\*20维

W0(2)=2\*10维

这样一乘积，是不是就在Z这个层降维了？30维马上变成5维

好，然后再往上在Z层的特征去估计，你把Z层的值去往上做的时候，直接加权求和做个全连接了！

（纸上公式）

这就是FM全过程

但是，你把z再代换回去x，公式就是这个。你看这个公式这个部分内积就是交叉特征。

所以，大家可以看到降维和自动学习出交叉特征.就是一个变量代换加一个全连接。

这边大家有没有问题：

没有问题的话，我开始讲本文另一个改进：DNNs [深度神经网络]

他这篇文章用到了两个神经网络，一个叫FNN，一个叫SNN

我先讲FNN，为什么先讲他。因为FNN这个其实是FM+NN 就是我刚刚讲的分解模型+NEURAL netWORK NEURAL NETWORK代替的是原来的LOGISTIC REGRESSION ， FM代替的是one-hot enconding

大家神经网络模型框架都知道么？

他也是个模型，为什么叫神经网络。因为它就是模仿生物，神经信息传播的方式，来构建机器学习模型训练的框架。

首先你的神经信息传播不是一下次传过去的，他分成很多层，因为生物内部就是分层传播的，神经网络就是模拟这一点。以这个图为例，一共4层，他是一层层传递的。

然后再看每一层。这一个个的小球，就类似一个个的细胞元，存取信息，并且传递信息。

这一个个细胞元，每个细胞元存取一个数值。

层与层之间。这些线，就类似人类的神经，用作连接层。

放到数学里，每根神经上有个weight，就是一个加权和，一个SIGMA——又被称作全连接。

然后我们来看这个具体的CTR任务

顶层是你的output，输出的也就是目标值,目标就是一个0-1的估计值，>0.5 就是1，<0.5就是0

底层就是Input，你输入的数据，在CTR问题中 ，就是onehot -embeding 以后的那个01,01，特征.

底层到顶层的传递，就是刚刚我说的全连接。一个加权和，一个SIGMA =w1\*V1+w2\*V2+…..+wn\*Vn，一个矩阵乘法。n就是每层细胞源的个数。

有了这些连接，模型过程中，你就可以把Input正向传播一遍，一层一层传上去,传到顶端，顶端的值就是CTR估值。

但你第一遍传上去的，CTR估值肯定不准确，因为你这些weight都是未知，你初始设定的。

所以你要根据CTR的估计值与真实值的差额的大小，通过BACK PROPOGATION逆传算法，（这是个公式）由SGD随机梯度下降法确定具体公式，去更新全部的weight。

更新到最下面。

这样一个传上去，再传下来，更新的来回，称为一个EPOCH，循环。

你每经过一次循环，你的参数更新一次，结果的预估就更准确一点。

等到循环了10次，50次，你的结果就接近真实情况了。

此外，有些参数是需要实现确定的，比如神经网络有多少层，每一层有多少个细胞源，每一个weight 的初始值，以及每一层的activate函数。这些都是事先确定好才可以开始训练的。

修改他们，称之为调参。

以上是NN

那么FNN是什么，FNN的精华在于第一层INPUT，因为和之前讲的一样，把一个复杂的高维的的Input ,变低维度了，加快了训练的速度，提升了训练的效果。

然后，我们再来细看这个神经网络传递的方式

我刚刚讲，层与层间的传播是用加权求sigma的方法计算的，

但是实际计算中，为了适应具体不同的问题。每一层用的实际上都是不同激活公式。

这次我们从上往下讲几个计算公式。

首先第三层到最顶层：sigmoid激活函数

第一层到第二层；第二层到第三层 ，外面包一个tanh函数。

他们都长这样，所以输出都在0-1之间。

而最底层的Input 到第一层的传播函数是： FM（factorisation machine FM）

FNN就是在有了z和x的基础上再用这个公式。

这个模型更有办法挖掘数据内部的关系，并且能得到更好的结果。

还有最后一个问题，就是这些weight的如何更新？

是SGD随机梯度下降+back propogation的方法，更新的。用论文中给出的推到好的公式即可。

本文另一个深度神经网络模型SNN

Sampling-based neural network

SNN 和FNN的不同只有底层的embeding不同，

不同于 one-hot encoding 和fm embedding

他用到一个关键参数m,强行把100w 维的变成m维，他就规定每次亮m个细胞元。

因为这是用到别人的想法，他只是提及了。

第4部分论文experiment 里面主要讲述了参数的选择和调参的一些结果，我们可以看这几个表格。

总结：

本文用DNN来做CTR预测；

两个模型：FNN，SNN

主要就是做了一个结合,FM embedding 方式结合深度神经网络。

(使得计算更高效！)

然后我们来看一下代码：

我跑了一下这个代码，这个FNN的模型。

我们看一下他的输入的用户标签有：

WEEKDAY

HUOR

USERAGENT

IP

REGION

14个向量.

[weekday]：0-5的铺开

[hour] :6-29的铺开

每一条数据，他是1的地方。那就记录他的位置。

比如这个数据weekday是3，hour=12，那么 它第一个向量就是3，第二个向量是18

数据集是26773 train；21150——test的一个量

**我想了一想这篇论文怎么能有用？**

我们公司的任务：CTR ，RTB是分开的两个任务嘛

我这个DNN，主要是解决CTR的问题，也就是给入数据，得到结果0；1

我在想可以真正用来改进CTR的地方：可能1个是运行时间，1个是运算capacity。 就是需要存的东西的容量。

DNN模型+FM (改进)，最大的地方就是降维，降维效果应该是指模型的第一层参数少了。应该结果是会更好的。

但是这个事情是需要验证的。

我现在在逐步研究代码，我觉得这个研究还是很有价值的。因为他有可行性。

http://blog.csdn.net/xiewenbo/article/details/52038493

我们主要看的是AUC这个指标，业内一般模型提升2个百分点就会带来巨大收益，可以看到从LR到PNN，模型效果提升了近5个百分点，由此可见FM、FNN、PNN这几类模型效果比LR显著要好。

所以我想做的第一件事，就是把丁的原始数据190w的，拿过来试一下。

我对比了一下原始数据，跑的数据，就一处不同。这个改一改是快速的。

* [iPinYou data formalizing repository](https://github.com/wnzhang/make-ipinyou-data)
* [Cretio 1T dataset](http://labs.criteo.com/downloads/download-terabyte-click-logs/)

我想记录两个东西，一个是时间

如果时间可行。我准备的第二件事是对比模型。

丁和我说现在已经做过的模型都是LOG-LOSS 来评估模型好坏的。

因为Log-loss和lr很适配的方法。

然后我这个用AUC，mse来做评估的嘛。所以

其实和Log-loss很像的

如果这两件事都做好了，我想可以是不是想着结合一下公司现有模型，实现降维的办法。

因为我现在没看过别的模型。所以对公司现在在用的移动平均模型这边相关知识还比较空白。

我想先去了解一下这类的知识？