DeepFM原理与实践

# CTR预估

DeepFM是CTR预估领域优秀的模型之一，因此这里简单介绍下CTR预估。

CTR预估数据特点：

1. 输入中包含类别型和连续型特征。类别型特征需要经过one-hot处理, 连续型数据可以先离散化再one-hot，也可以直接保留原值

2. 维度非常高

3. 数据非常稀疏

4. 特征按照Field分组

CTR预估重点在于学习组合特征。注意，组合特征包括二阶、三阶甚至更高阶的，阶数越高越复杂，越不容易学习。Google 的论文研究得出结论：高阶和低阶的组合特征都非常重要，同时学习到这两种组合特征的性能要比只考虑其中一种的性能要好。

那么关键问题转化成：如何高效的提取这些组合特征。一种办法就是引入领域知识人工进行 特征工程。这样做的弊端是高阶组合特征非常难提取，会耗费极大的人力。而且，有些组合特征是隐藏在数据中的，即使是专家也不一定能提取出来，比如著名的“尿布与啤酒”问 题。

在DeepFM提出之前，已有LR，FM，FFM，FNN，PNN（以及三种变体： IPNN,OPNN,PNN\*），Wide&Deep模型，这些模型在CTR或者是推荐系统中被广泛使用。

# 模型演进历史

## 线性模型

最开始 CTR 或者是推荐系统领域，一些线性模型取得了不错的效果。比如：LR，FTRL。线性模型有个致命的缺点：无法提取高阶的组合特征。所以常用的做法是人为的加入 pairwise feature interactions。即使是这样：对于那些出现很少或者没有出现的组合特征以及高阶组合特征依旧无法提取。

LR大的缺点就是无法组合特征，依赖于人工的特征组合，这也直接使得它表达能力受 限，基本上只能处理线性可分或近似线性可分的问题。

## FM模型

线性模型差强人意，直接导致了 FM 模型应运而生（在 Kaggle 上打比赛提出来的，取得了 第一名的成绩）。FM 通过隐向量 latent vector做内积来表示组合特征，从理论上解决了低阶和高阶组合特征提取的问题。但是实际应用中受限于计算复杂度，一般也就只考虑到 2 阶交叉特征。后面又进行了改进，提出了FFM，增加了Field的概念。

## 深度学习模型

随着DNN在图像、语音、NLP等领域取得突破，人们渐渐意识到DNN在特征表示上的天 然优势。相继提出了使用CNN或RNN来做CTR预估的模型。但是，CNN模型的缺点 是：偏向于学习相邻特征的组合特征。RNN模型的缺点是：比较适用于有序列 (时序) 关系的数据。

FNN的提出，应该算是一次非常不错的尝试：先使用预先训练好的FM，得到隐向量，然后作为DNN的输入来训练模型。缺点在于：受限于FM预训练的效果。

随后提出了PNN，PNN为了捕获高阶组合特征，在embedding layer和first hidden layer之间增加了一个product layer。根据 product layer 使用内积、外积、混合分别衍生出IPNN, OPNN, PNN\*三种类型。

无论是FNN还是PNN，他们都有一个绕不过去的缺点：对于低阶的组合特征，学习到的比较少。而前面我们说过，低阶特征对于CTR也是非常重要的。

Google意识到了这个问题，为了同时学习低阶和高阶组合特征，提出了Wide&Deep模 型。它混合了一个 线性模型（Wide part）和Deep模型(Deep part)。这两部分模型需要不同的输入，而Wide part部分的输入，依旧依赖人工特征工程。

但是，这些模型普遍都存在两个问题：

1. 偏向于提取低阶或者高阶的组合特征。不能同时提取这两种类型的特征。

2. 需要专业的领域知识来做特征工程。

DeepFM 在 Wide&Deep 的基础上进行改进，成功解决了这两个问题，并做了一些改进， 其优势/优点如下：

1. 不需要预训练FM得到隐向量

2. 不需要人工特征工程

3. 能同时学习低阶和高阶的组合特征

4. FM模块和Deep模块共享 Feature Embedding部分，可以更快的训练，以及更精确的训练学习

# DeepFM

这里先用两点概况性的语句指出DeepFM的精华。

1、DeepFM（FM Component + Deep Component）包含两部分：因子分解机（FM）部分与神经网络（DNN）部分，FM部分负责低阶特征的提取（包括一阶和二阶，虽然FM也可以实现高于二阶的特征提取，但是由于计算复杂度的限制，一般只计算到二阶），DNN部分负责高阶特征的提取

2、共享feature embedding。FM和Deep共享输入和feature embedding不但使得训练更快，而且使得训练更加准确。

DeepFM的预测公式如下：

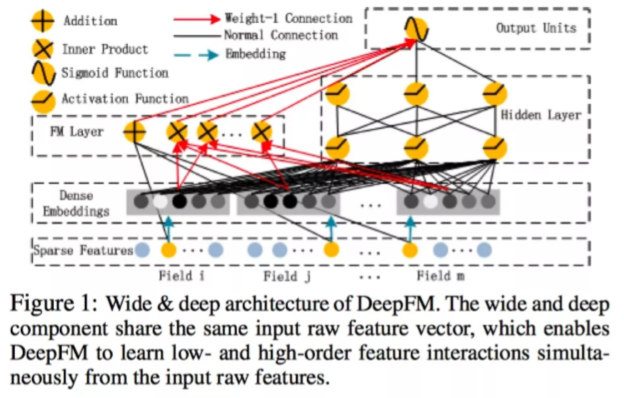


注：DeepFM与Wide&Deep的不同点在于，DeepFM是端到端的训练，不需要人工特征工程，而Wide&Deep中的Wide部分依然依靠人工特征工程。此外，需要明白的是，在Wide&Deep中，input vector非常大，里面包含了大量的人工设计的pairwise组合特征，增加了计算复杂度。

DeepFM中需要清楚理解的几个值：

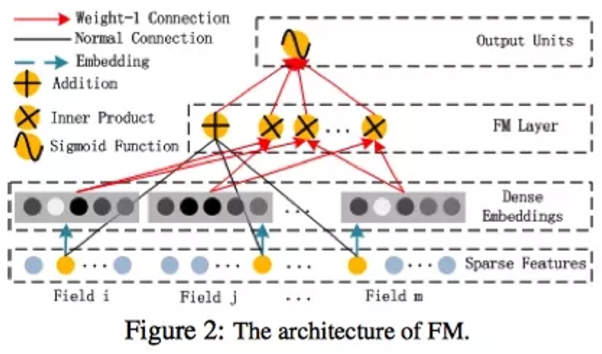
1. field\_size: 输入X在进行one-hot之前的特征维度，对应FM部分一维部分(即<w,x>)
2. feature\_size: 输入X在one-hot之后的特征维度，又记作n
3. embedding\_size: one-hot后的输入，进行嵌入后的维度，又记作k，对应FM二维组合特征部分

## 模型结构

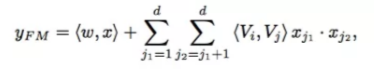


## FM部分

FM部分结构如下：



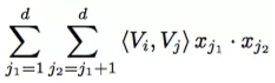
FM部分是因子分解机。FM部分的输出由两部分构成：（1）一个Addition Unit；（2）多个内积单元。FM部分的公式如下：



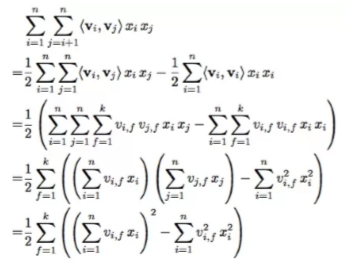
这里的d是输入one-hot之后的维度，我们一般称之为feature\_size（注意：在某些文章中也常用k表示）。

Addition Unit 反映的是1阶的特征。内积单元反映的是2阶的组合特征对于预测结果的 影响。

注意：虽然公式上Y\_fm是所有部分都求和，是一个标量。但是从FM模块的架构图上我们可以看到，输入到输出单元的部分并不是一个标量，应该是一个向量。实际实现中采用的是 FM化简之后的内积公式，最终的维度是：field\_size + embedding\_size。其中field\_size部分对应的是<w,x>，embedding\_size部分对应的是



这一部分可以按照下面的方式简化，在我们coding的过程中通常使用简化后的形式：



K就是W的列数，就是Embedding后的维度，也就是embedding\_size。也就是说，在 DeepFM的FM模块中，最后没有对结果从[1,K]进行求和。而是把这K个数拼接起来形成了一个 K 维度的向量。

FM Component总结：

1. FM 模块实现了对于1 阶和2阶组合特征的建模。

2. 没有使用预训练

3. 没有人工特征工程

4. embedding 矩阵的大小是：特征数量 \* 嵌入维度。然后用一个 index 表示选择了哪个特 征。

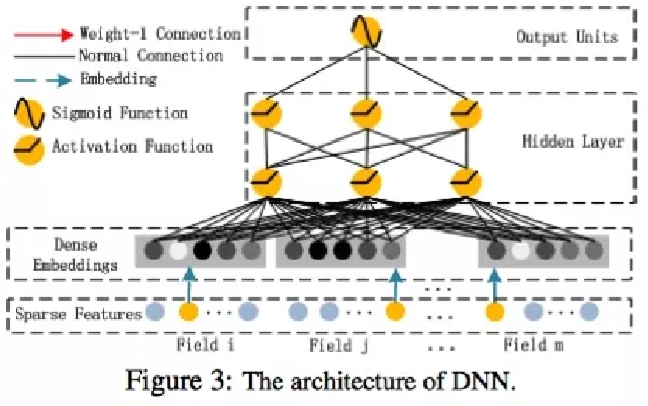
需要训练的有两部分：

1. input\_vector和Addition Unit相连的全连接层，也就是1阶的Embedding矩阵。

2. Sparse Feature到Dense Embedding的Embedding矩阵，中间也是全连接的，要训练的是中间的权重矩阵，这个权重矩阵也就是隐向量V

## 深度（DNN）部分：

Deep Component架构图如下：

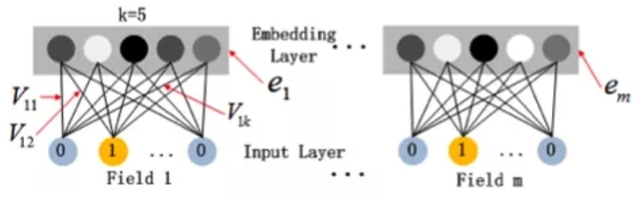


Deep Component 是用来学习高阶组合特征的。网络里面黑色的线是全连接层，参数需要神经网络去学习。

由于CTR或推荐系统的数据one-hot之后特别稀疏，如果直接放入到DNN中，参数非常多，我们没有这么多的数据去训练这样一个网络。所以增加了一个 Embedding层，用于降低纬度。

Embedding层具有两个特点：（1）尽管输入的长度不同，但是映射后长度都是相同的.embedding\_size也就是k，这是我们根据经验预先设置的值；（2）embedding 层的参数其实是全连接的 Weights，是通过神经网络自己学习到的。

Embedding层（嵌入层）的架构图：

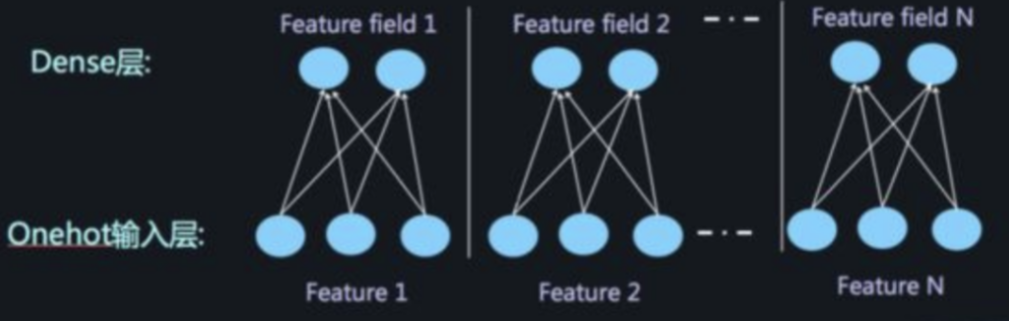


嵌入层(embedding layer)的结构如上图所示。通过嵌入层，尽管不同field的长度不同（不同离散变量的取值个数可能不同），但是embedding之后向量的长度均为K（提前设定好的 embedding-size)。另外，我们要明白的是FM部分与DNN部分是共享feature embedding的。

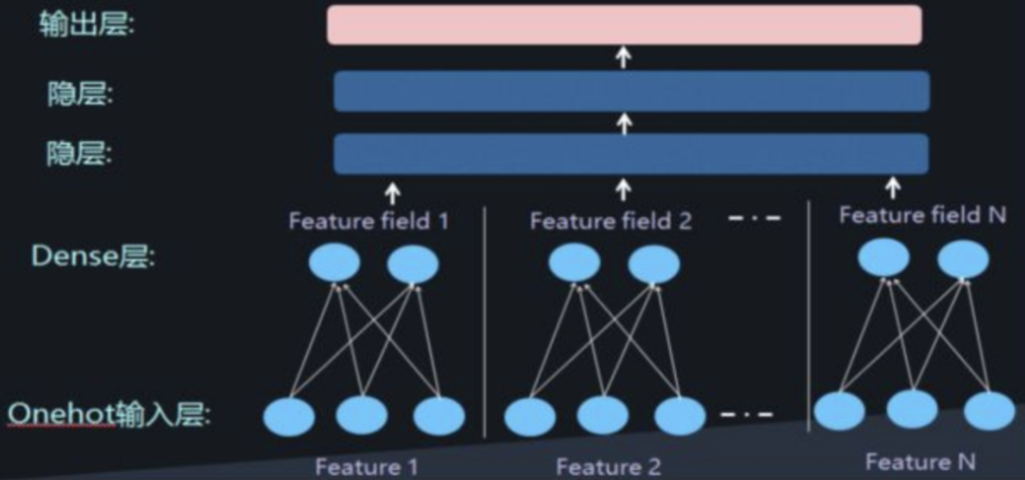
上面已经对DeepFM中DNN部分进行了阐述，这里简单补充下为什么DNN部分需要先做embedding，而不是直接用原始特征直接输到DNN。对于离散特征，一般将特征转换成one-hot的形式，但是如果将one-hot类型的特征输入到DNN中，会导致网络参数太多，计算力撑不住，如下图：假如输入层one-hot之后的维度是1000万，隐藏层有500个节点，则在全连接神经网络中参数数量达到50亿，这么多参数的网络是很难训练的。



因此，在DeepFM中的DNN部分采用了类似于FFM中的思想，将特征分为不同的Field，将one-hot 转换为Dense Vector，如下图：



再加上两层的全链接层，让Dense Vector进行组合，那么高阶特征的组合就出来了，如下图：



DeepFM模型的优点：

1. 模型可以从最原始的特征中同时学习低阶和高阶组合特征，即可以同时对低阶特征组合和高阶特征组合建模，从而能够学习到各阶特征之间的组合关系；
2. DeepFM模型是一个端到端的模型，不需要任何的人工特征工程。

# 实践

理解了DeepFM的原理，完成相应的代码并不困难，然后coding过程中有下面几点需要注意，否则可能出现不必要的麻烦，甚至导致最后结果是错误的。需要注意的问题有：

1. 对于FM部分的一次项，从结构图中我们可以看到，一次向是求和了，但是在实际代码中，我们并不需要对一次项求和，而是输出一个k维的向量，这里的k就是embedding\_size。这样做的原因是：输入由三部分组成： FM-1维，FM-二维，Deep部分。 自然而然，我们不希望FM-1维只是一个标量，一 个数吧。这显然不利于LR模型的学习，那么如果用原始的维度那：公式里是Wi \* Xi， Xi的维度是n，n是one-hot之后的维度，这个维度太大了以至于我们才想出了各种办 法来解决这个问题，用n显然不行。所以，就用field\_size。从逻辑上来说，也是对 one-hot之前每个特征维度的一种建模
2. 对于FM部分的二阶交叉部分，从结构图中也可以看到在最外层是embedding\_size上的求和，在实际代码中不需要求和，而是得到一个embedding\_size维（即k维）的向量，这才是FM二阶部分的实际输出。原因大概同上。

代码请参考：<https://github.com/jpegbert/code_study/tree/master/DeepFM>

## 超参数建议

