AFM模型理论与实践

# 背景

推荐系统或者CTR预估中输入中类别型特征比较多，因为这些类别型特征不是独立的，所以他们的组合特征就显得非常重要。

一个简单的办法就是给每一个组合特征 (cross feature) 一个权重 。 但是这种 cross feature-based方法的通病就在于训练集中很多组合特征并没有出现，导致无法有效学习。

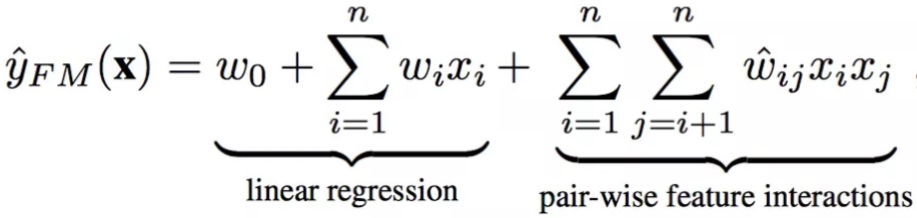
FM通过为每一个特征学习一个嵌入向量，也叫做隐向量，通过两个隐向量的内积来表示这个组合特征的权重。但是同样有个问题就是，在预测中有一部分特征是不重要甚至是没用的，它们会引入噪声并对预测造成干扰。对于这样的特征，在预测的时候应该赋予一个比较小的权重，但是FM并没有考虑到这一点。对于不同的特征组合，FM并没有区分它们的权重 (可以认为内积之后看成一个组合特征，它们的权重都是1)。

AFM模型由浙大与新加坡国立大学联合推出，全称是Attentional Factorization Machine，NFM的作者也参与了研究。通过引入Attention机制，用来赋予不同的特征组合不同的重要程度。权重可以在网络中自动学习，不需要引入额外的领域知识。AFM最大的特点就是使用一个attention network来学习不同组合特征的重要性。

AFM的贡献在于，将attention机智引入到特征交叉模块，不同于NFM在Bi-Interaction Layer中的处理，AFM在二阶交叉特征进行sum pooling时，使用attention network计算得到的attention score对交叉特征进行加权求和。因为不同的交叉特征有着不同的重要性，所以通过模型显式的学习出每种交叉特征的权重，不仅有益于模型的性能提升的，还增强了模型可解释性。

# FM回顾

FM全称是Factorization Machine，形式化公式如下：



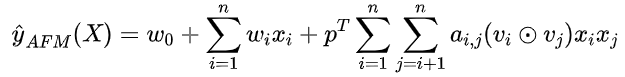
其中w是两个特征隐向量v的内积。FM有下面两个问题：

1. 一个特征针对其他不同特征都使用同一个隐向量。所以有了FFM用来解决这个问题。
2. 所有组合特征的权重w都有着相同的权重1。AFM就是用来解决这个问题的。

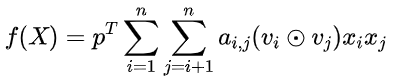
在一次预测中，并不是所有的特征都有用的，但是FM对于所有的组合特征都使用相同的权重。AFM就是从这个角度进行优化的，针对不同的特征组合使用不同的权重。这也使得模型更加可解释性，方便后续针对重要的特征组合进行深入研究。

# AFM模型

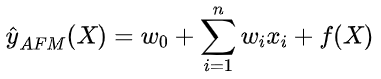
AFM全称是Attentional Factorization Machine，通过引入attention机制（使用一个attention network）来学习不同组合特征的重要性。AFM的公式化定义如下：



我们先采用对公式逐步分析的方式对AFM有个笼统的认识。首先，假设：

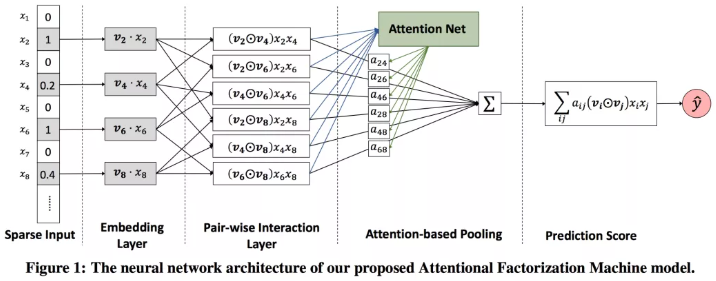


因此，AFM的公式可以简化为：



其中求和的三项分别为：全局偏置项、一阶项、高阶项。前两项就是线性模型，AFM的重点在于第三项，这部分是AFM模型的创新和特点所在。

## 模型结构



注意，这里面省去了线性部分，只考虑特征组合部分。在AFM中Sparse Input和Embedding Layer和FM中的是相同的，Embedding Layer把输入特征中非零部分特征embed成一个dense vector。

从图中可以看出，AFM模型的特征交叉部分可以划分为五个部分，下面依次进行介绍：

## Sparse Input

这个部分主要是为了说明输入数据特点。在对输入特征进行二值化、离散化等处理之后，输入数据往往会变成高维稀疏数据。

## Embedding Layer

高维稀疏数据不利于模型的学习。因此，AFM使用了Embedding Layer 来对高维稀疏特征进行降维，将其表示为低维稠密特征。每个特征对应一个隐向量，将特征值与隐向量进行乘积，得到特征向量作为实际特征表示。

## Pair-wise Interaction Layer

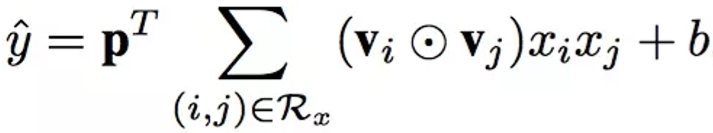
这一层主要是对组合特征进行建模（即二阶特征交叉），原来的n个嵌入向量，通过element-wise product（两两特征交叉）操作得到了n(n-1)/2个组合向量，这些向量的维度都是嵌入向量的维度k。形式化如下：



其中，表示 Embedding Layer的输出，表示所有可能的特征交叉组合，xi表示特征值，vi表示维度为k的隐向量，运算符表示向量对应元素相乘。假设m = n(n-1)/2，那么。也就是Pair-wise Interaction Layer的输入是所有嵌入向量，输出也是一组向量。输出是任意两个嵌入向量的element-wise product。任意两个嵌入向量都组合得到一个 Interacted vector，所以n个嵌入向量得到n(n-1)/2个向量。

注：Pair-wise Interaction Layer与NFM中的Bi-Interaction Layer在交叉过程完全一样，只是Bi-Interaction Layer在交叉完了之后再进行向量等权求和处理。

如果不考虑Attention机制，在Pair-wise Interaction Layer之后直接得到最终输出，我们可以形式化如下（下面这个式子也是Generalize FM的表达形式）：



其中p和b分别是权重向量和偏置，表示向量对应元素求和， 输出为一个向量。当p全为1向量，b为0的时候，上式就是FM的二阶交叉项，如果此时再把全局偏置和一阶项考虑进来，此时的模型就是FM模型。这个只是说明 AFM的表达能力是在FM之上的，实际的情况中我们还使用了Attention机制。

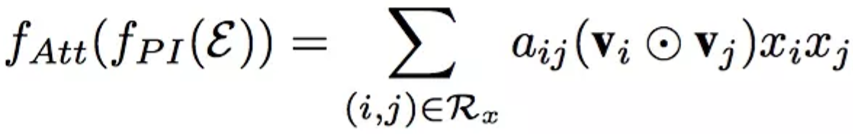
注意：NFM中的 Bilinear Interaction Layer也是把任意两个嵌入向量做element-wise product，然后进行sum pooling操作。

## Attention-based Pooling Layer

在NFM中，特征向量进行两两交叉之后，直接进行sum pooling，将二阶交叉向量进行等权求和处理。但是直觉上来说，不同的交叉特征应该有着不同的重要性。不重要的交叉特征应该降低其权重，而重要性高的交叉特征应该提高其权重。Attention概念与该思想不谋而合，AFM作者顺势将其引入到模型之中，为每个交叉特征引入重要性权重，最终在对特征向量进行sum pooling时，利用重要性权重对二阶交叉特征进行加权求和。

为了计算特征重要性权重（Attention score），作者构建了一个Attention Network，其本质是含有一个隐藏层的多层感知机（MLP）

Attentio机制的核心思想在于：当把不同的部分压缩在一起的时候，让不同的部分的贡献程度不一样。AFM通过在Interacted vector后增加一个weighted sum来实现Attention机制。形式化如下：



aij是Attention score，表示不同的组合特征对于最终的预测的贡献程度。可以看到：

1. Attention-based Pooling Layer的输入是Pair-wise Interaction Layer的输出。它包含 n(n-1)/2 个向量，每个向量的维度是k。（k是嵌入向量的维度，n是 Embedding Layer中嵌入向量的个数）

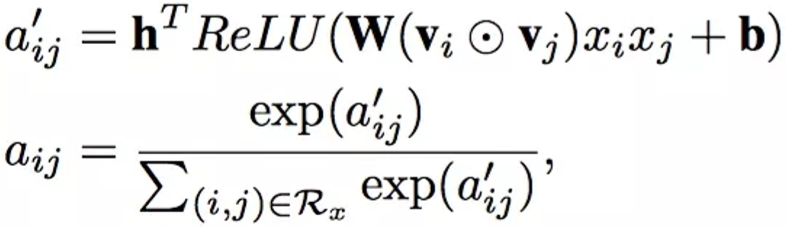
2. Attention-based Pooling Layer的输出是一个k维向量。它对Interacted vector使用Attention score进行weighted sum pooling（加权求和）操作。

计算Attention score的一个常规的想法就是随着最小化loss来学习，但是这样做对于训练集中从来没有一起出现过的特征组合的Attention score无法学习。

AFM用一个Attention Network来学习Attention score。

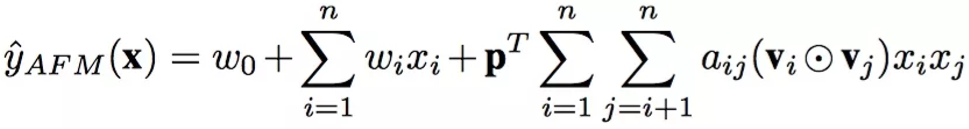
Attention network实际上是含有一个隐藏层的多层感知机（MLP），激活函数使用ReLU，网络大小用attention factor表示，就是神经元的个数。

Attention network的输入是两个嵌入向量element-wise product之后的结果 (interacted vector，用来在嵌入空间中对组合特征进行编码)；它的输出是组合特征对应 的Attention score。最后，使用softmax对得到的Attention score进行规范化， Attention Network形式化如下：



其中，是模型参数。参数t是Attention Network隐藏层节点数，在论文中被称之为 attention factor。计算得到的aij即表示对应的二阶交叉特征的重要性权重。得到权重之后，就可以对二阶交叉特征进行加权求和，可以表示为，其结果仍然是一个向量，作为下一模型结构的输入。

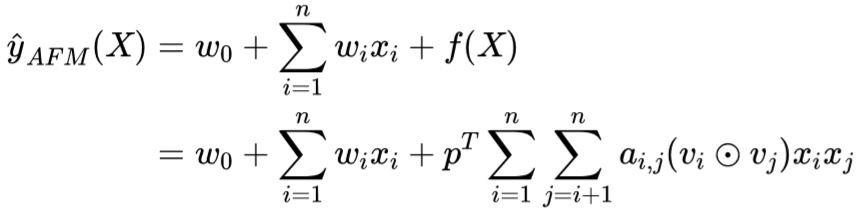
总结，AFM模型总形式化如下：



前面一部分是线性部分；后面一部分对每两个嵌入向量进行element-wise product得到 Interacted vector；然后使用Attention机制得到每个组合特征的Attention score，并用这个score来进行weighted sum pooling；最后将这个k维的向量通过权重矩阵p得到最终的预测结果。

## Prediction Score

Attention-based Pooling得到二阶交叉信息，在这个部分负责将其表示为一个标量。具体定义为： ，其中表示Attention-based Pooling的输出，p为参数向量。至此，我们完成了对AFM核心模块的分析。最终AFM的公式定义如下：



模型参数集合为{w0,wi,vi,p,h,W,b}。

# 实践

参考：<https://github.com/jpegbert/code_study/tree/master/AFM>

# 实践Tricks

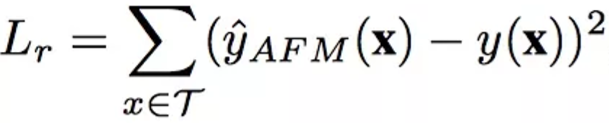
## 模型训练

AFM针对不同的任务有不同的损失函数。

1. 回归问题。square loss。

2. 分类问题。log loss。

论文中针对回归问题来讨论，所以使用的是square loss，形式化如下：



模型参数估计使用的是SGD。

## 过拟合

AFM是对于NFM的改进，但在改进同时引入了更多的参数，增加了模型过拟合风险。作者使用了dropout与L2正则的方式来避免过拟合问题。

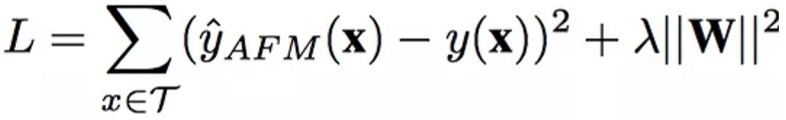
防止过拟合常用的方法是Dropout或者L2、L1正则化。AFM的做法是：

1. 在Pair-wise Interaction Layer的输出使用Dropout防止过拟合

2. 在Attention Network中使用L2正则化

Attention Network是一个one layer MLP。不给他使用Dropout是因为，作者发现如果同时在interaction layer和Attention Network中使用Dropout会使得训练不稳定，并且降低性能。

所以，AFM的最终loss函数为：



其中W是Attention Network的参数矩阵，代表全体训练样本。

# 模型效果评估

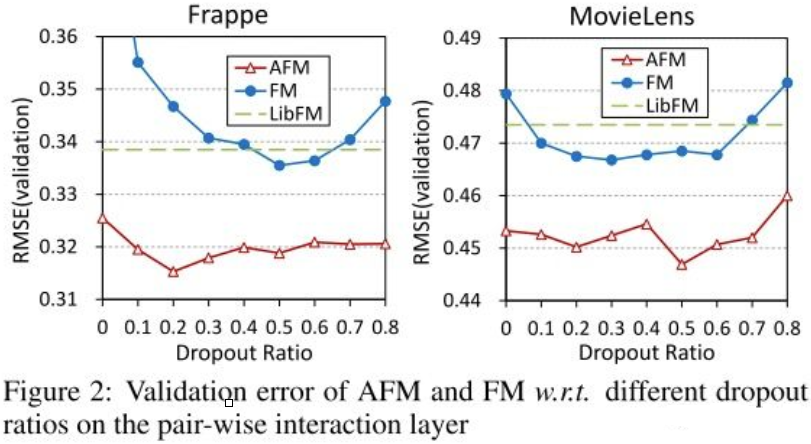
论文中作者在对模型进行消融分析时，作者从三个问题出发，设计实验证明模型的优越性。实验数据集为Frappe与MovieLens。对比模型分别有：LibFM、HOFM、Wide&Deep、DeepCross。对于所有实验Embedding Size都为256，如未提及，attention factor均为256。

在实验过程中，作者发现对比AFM、Wide&Deep、DeepCross模型，相对于随机初始化参数，使用预训练的FM参数进行初始化可以提高模型性能。

## 问题一

**Q：AFM的关键超参数（如，dropout与L2正则化）对于性能的影响如何？**

对比三种模型（LibFM、FM、AFM）在不同dropout比例下的表现，其中的FM是将AFM的Attention Network去除得到的等价模型。实验效果如下：



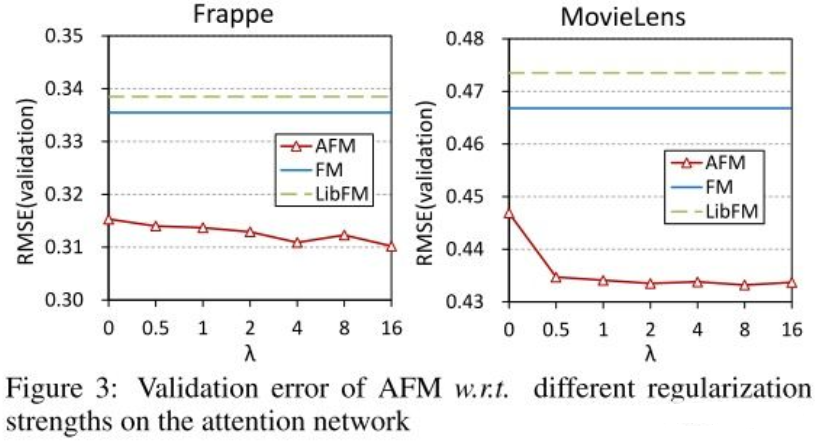
该实验表明了一些结论，

1）在Pair-wise Interaction Layer使用dropout，可以提高模型的表现。

2）对比LibFM与FM，发现FM的效果优于LibFM，可能的原因在于优化器的不同，LibFM使用普通的SGD算法，而FM使用Adagrad进行优化。还有一个原因在于，LibFM使用L2正则来防止过拟合，而该FM模型使用dropout。

3）即使在AFM不使用dropout时，其效果仍大幅优于FM与LibFM，说明Attention Network的有效性。

将AFM设置为最佳的dropout比例，然后我们来看在不同的  （L2正则强度）下，AFM的表现如何。对比效果如下：

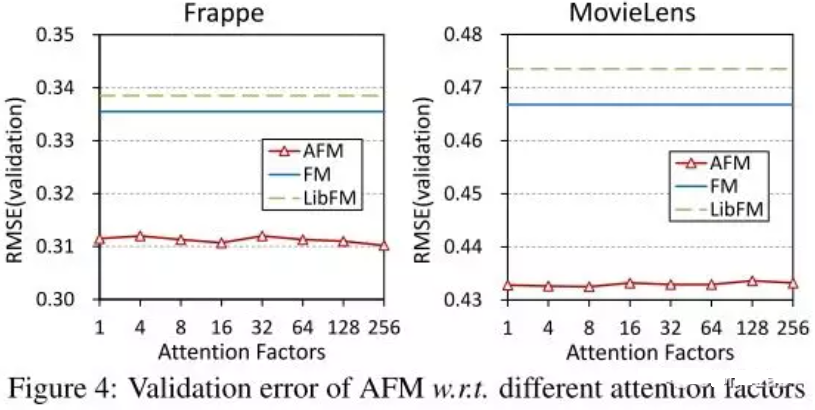


可以看出，随着正则化强度的提高，AFM的性能得到进一步提高。这说明仅对Pair-wise Interaction Layer使用dropout是不够的，加上Attention Network的L2正则才能达到最佳效果。

## 问题二

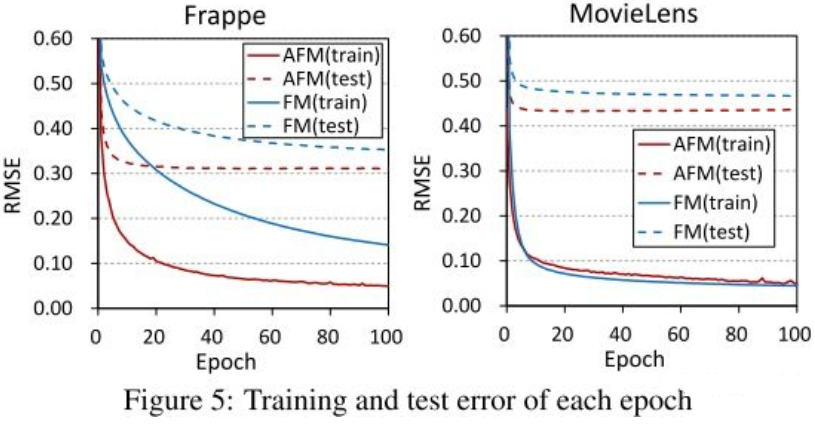
**Q：Attention Network能够有效地学习交叉特征的重要性吗？**

在回答这个问题之前，我们应该选择一个合适的 attention factor。下图展示了在不同的attention factor下模型的表现，每次试验都将 dropout与 设置到最佳。



从图上可以看出，随着 attention factor的改变，AFM的表现相当稳定。上文中提到，Attention Network是一个含有一层隐藏层的感知机。所以当 attention factor等于1时，Attention Network的隐藏层只有一个节点。尽管条件设定很严格，AFM仍然能有突出的表现，这说明了Attention Network的合理性，不同的交叉特征根据重要性权重进行加权求和，能够有效提高模型表现。

此外，对比AFM与FM在训练过程中的收敛速度，结果如下。在两个数据集上，AFM都有着不错的表现。



为了能够更深入的分析Attention Network的优势，作者设计实验。首先将Attention Network参数固定，将所有的设置为，模拟FM训练Embedding向量，直到模型收敛。然后再将Embedding Layer进行固定，放开Attention Network进行训练，直到模型再次收敛。结果显示，第二次收敛模型性能提高了3%。这充分说明了Attention Network的有效性。

在实验过程中抽取三条样本进行观察，在FM阶段与放开Attention Network之后的阶段，Item-Tag的交叉特征都是最重要的，但是在FM阶段模型是将所有交叉特征等权求和的（权重均为0.33）。而在加入Attention Network之后，模型给Item-Tag分配了更高的权重，三条样本分别分配了0.38、0.42、0.37权重。这说明模型确实能够自适应的调整不同交叉特征的权重，提高模型表现。

## 问题三

**Q：在稀疏数据集上，AFM与最好的模型比较是否有提高呢？**

在作者对比的几种模型中，AFM参数少，但是表现最佳。

# 总结

AFM是在FM的基础上改进的。相比于其他的DNN模型，比如Wide&Deep，DeepCross 都是通过MLP来隐式学习组合特征。这些Deep Methods都缺乏解释性，因为并不知道各个组合特征的情况。相比之下，FM通过两个隐向量内积来学习组合特征，解释性就比较好。

通过直接扩展FM，AFM引入Attention机制来学习不同组合特征的权重，即保证了模型的可解释性又提高了模型性能。但是，DNN的另一个作用是提取高阶组合特征，AFM依旧只考虑了二阶组合特征，这应该算是AFM的一个缺点吧。

参考：

https://mp.weixin.qq.com/s/V-qXXQxsT3aOhUqqMRxdIA

<https://mp.weixin.qq.com/s/q-M3Z93pD5CDRFJjFzuwPg>