深度CTR之AFM：基于Attention网络的FM模型

原创 大厂机器学习 大厂机器学习实战 3月2日

1 解决的问题

该模型于2017年提出，由浙大与新加坡国立大学合作推出。文章题目-《Attentional Factorization Machines：Learning the Weight of Feature Interactions via Attention Networks》

模型与FM算法有关，FM算法使用了二阶特征来提升线性模型的性能，但是FM在使用所有二阶交叉特征时，默认每个交叉特征的权重是一样的，都是1，但实际上每个交叉特征的用处大小并不是相同的，而且有些没用的交互特征可能会给模型的学习带来负面影响，阻碍了模型性能的提升。因此，paper中提出的新模型的核心就是用于**区分不同交互特征的重要性，而不是都是一样的重要性**，也就是文章提出的新模型**AFM**。在两个真实世界数据集上，AFM模型体现出了其有效性。其中在回归任务上相比于FM模型提升显著，并且相比于W&D和Deep&Cross模型效果更好，且模型结构更加简单、参数数量更少。

2 介绍

监督学习是机器学习和数据挖掘中一种非常基础的任务，不管是回归任务的实值预测还是分类任务的分类标签预测，都是需要基于给定的特征输入来学习一个预测器函数。对预测器来说，提**供两个特征之间的相互关系的可解释性**是非常重要的。

在学习器的构建过程中，有这样的一个roadmap：

1. （LR模型）普通的简单学习模型，例如逻辑回归LR，其缺点是**无法学习不同特征之间的联系**；
2. （W&D中Wide模块，即在LR中加入人工特征工程的交互特征）提出了polynomial regression（PR）结构模型，使用交叉结构的特征，模型也可以学习出来交叉特征的权重值，例如W&D模型结构中的Wide模块，其缺点是**对于一些稀疏特征只有极少的交叉特征在样本中存在，这无法保证模型能够学习到真实的特征权重，而且这对于样本中未出现的交叉特征是无法学习的**。
3. （FMs模型）为了解决PR结构模型的无法泛化的缺点，FMs被提出来，**用于将交叉特征权重参数化为特征embedding向量的内积**，通过学习每个特征的embedding向量表达，FM可以评估计任何交叉特征的权重，这种泛化性能使得FMs可以用于很多类型的任务，其缺点是FMs同等对待所有交互特征，这与真实世界中有的特征可回忆发挥作用，而有的特征

无法对预测结果产生有效影响的事实是相悖的，因此**FM缺乏区分不同交互特征的重要性的能力，这有时候导致只能产生次优解**。

4. （本paper的AFM模型）为了解决FM模型无法区分交互特征的重要性的缺点，文章中提出的AFM模型中引入了attention 机制---这可以使得不同的交互特征对预测的结果产生不同重要性程度的影响，更重要的是，**交互特征的重要性大小可以不用人类的领域知识就能从数据中自动学习出来**。

AFM在内容方面和个性化推荐方面的两个数据集上，进行的试验表明在FM上结合attention的使用时具有两方面的优势：

1. 学习的模型效果更好；
2. 深入洞察哪个特征可以对模型产生更重要的作用。AFM能够极大地增强了FM模型的可解释性和透明度，这允许我们对模型涉及的行为进行更深入的分析。

3 FM模型

来简单回顾下FM模型。

首先看下FM的预测公式：

$$\hat{y}(x) := w_0 + \sum_{i=1}^{n} w_i x_i + \sum_{i=1}^{n} \sum_{j=i+1}^{n} < v_i, v_j > x_i x_j \qquad (1)$$

其中需要学习的参数为：

$$w_0 \in \mathbb{R}, \qquad w \in \mathbb{R}^n, \qquad V \in \mathbb{R}^{n \times k} \qquad (2)$$

公式（1）中的$\langle \cdot, \cdot \rangle$表示两个k维向量的点积：

$$< v_i, v_j >:= \sum_{f=1}^{k} v_{i,f} \cdot v_{j,f} \qquad (3)$$

其中V的第i行vi表示第i个特征的k维向量表示，而$k \in Z_+$是一个表示因子分解的维数的超参数。

在2阶FM模型中，可以表达出所有单特征以及变量之间二阶交互特征，具体参数如下：

1. w_0表示全局的偏差；
2. w_i表示第i个特征的强度；
3. $w^{i,j} := \langle vi, vj \rangle$表示第i个特征和第j个特征之间的交互，在实际参数学习中不是直接学习交互特征的权重参数w^i,j的，而是通过**学习因式分解参数**来学习交互特征的参数。

4 AFM

4.1 模型结构

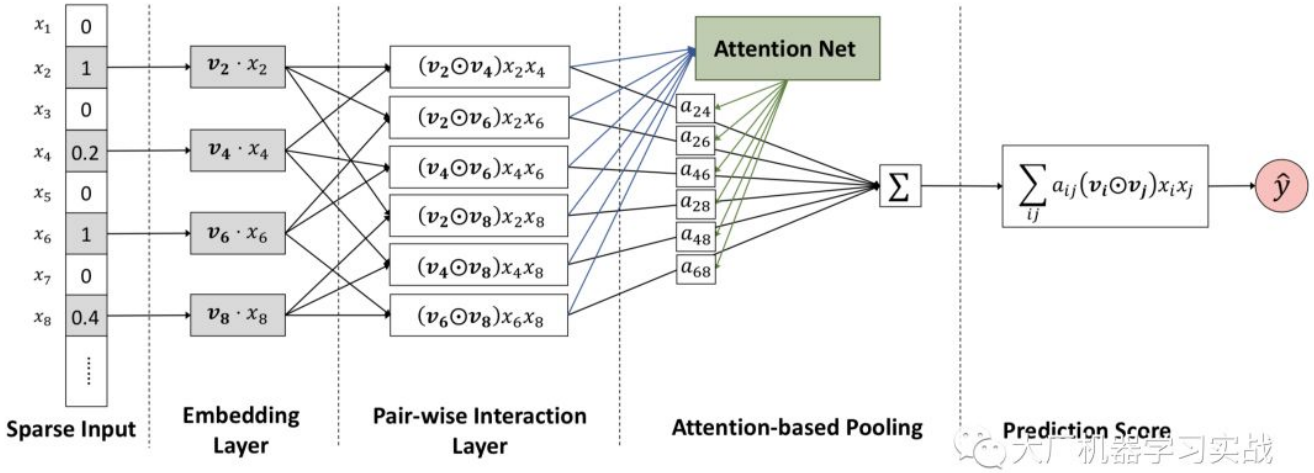

Figure 1: The neural network architecture of our proposed Attentional Factorization Machine model.

在这里插入图片描述

为简单起见，我们在图中省略了线性部分。输入层和embedding层与FM模型是一样的，其中对于输入特征都采取了稀疏表示，即将所有的非零特征都嵌入到dense特征。

文章的核心贡献在于下面将要介绍的 pair-wise 交互层、attention-based 池化层。

4.1.1 Pair-wise 交互层

用X表示特征向量的非零特征的集合。

$$f_{PI}(\varepsilon) = \{(v_i \odot v_j)x_i x_j\}_{(i,j)\in\mathcal{R}} \qquad (4)$$

$$\mathcal{R}_x = \{(i,j)\}_{i\in\chi, j\in\chi, j>i} \qquad (5)$$

其中，⊙表示两个向量的element-wise内积。这看起来其实跟FM无异。因此**定义Pair-wise 交互层的目的就是为了在神经网络中表达FM的计算逻辑，**

$$\hat{y} = p^T \sum_{(i,j)\in\mathcal{R}_x} \{(v_i \odot v_j)x_i x_j\} + b \qquad (6)$$

其中

$$p \in \mathbb{R}^k, \qquad b \in \mathbb{R} \qquad (7)$$

分别表示预测网络的权重值和偏差值。对于将p置为值全为1的向量以及b=0，那么公式（6）则可以完全复现FM模型的计算公式。

> 文章中还特别给作者团队的另一篇文章 Bilinear Interaction pooling operation 打了个广告。

4.1.2 Attention-based 池化层

自从attention机制被引入到神经网络建模中以来，其在推荐、信息检索、计算机视觉等很多任务中都获得了广泛的应用。这个idea是指在将他们压缩成一个单独的表示时，允许不同的部分贡献不同大小。由于FM缺点的影响，我们通过对交互向量计算加权和，来将attention机制应用于特征交互。

$$f_{Att}(f_{PI}(\varepsilon)) = \sum_{a_(i,j)\in\mathcal{R}_x} a_{ij}(v_i \odot v_j)x_i x_j \qquad (8)$$

其中，aij表示交互特征的wij的注意力分数，也就是表示wij的在预测目标值时的重要性程度。

为了能够估计aij，一个比较直接的方法就是通过最小化loss函数去学习其值，虽然看起来是可行的，但是这又会碰到之前的问题：当某个交互特征没有出现在样本中时，就没法某个交互特征的attention分数了。为了解决这个泛化能力方面的问题，我们使用MLP网络去参数化这个attention分数，该MLP网络称之为attention network。attention network的输入是两个特征的交互向量，当然这里是已经对交互信息进行了嵌入编码了，最后attention network定义如下：

$$a_{ij}^{'} = h^T ReLU(W(v_i \odot v_j)x_i x_j + b) \qquad (9)$$

$$a_{ij} = \frac{e^{a_{ij}^{'}}}{\sum_{(i,j)\in\mathcal{R}_x} e^{a_{ij}^{'}}} \qquad (10)$$

$$W \in \mathbb{R}^{t*k}, b \in \mathbb{R}^t, h \in \mathbb{R}^t \qquad (11)$$

其中，w、b、h都是模型参数，t表示attention network的隐层的大小，我们将t称为 `attention factor` (后面的实验环节中，t和embedding size都设置为了256)，attention分数是通过softmax函数进行归一化的，这也是一个常规操作。我们在激活函数上选择了ReLU函数，效果也比较好。

Attention-based 池化层的输出是一个k维的向量，其在embedding空间中通过区分出他们各自的重要性，来压缩了所有的特征交互，我们将这些映射到最终的预测结果上面，即**AFM模型的完整公式**如下：

$$\hat{y}_{AFM}(x) = w_0 + \sum_{i=1}^{n} w_i x_i + p^T \sum_{i=1}^{n}\sum_{j=i+1}^{n} a_{ij}(v_i \odot v_j)x_i x_j \qquad (12)$$

其中，aij在公式（10）已经定义，模型参数为：

$$\ominus = \{w_0, \{w_i\}_{i=1}^n, \{v_i\}_{i=1}^n, p, W, b, h\}$$

4.2 模型的学习

AFM模型可以用于回归、分类、排序等任务中，但是对于不同的学习任务需要定制不同的目标函数，对于回归任务，目标label是一个实值，一个比较常见的额loss函数就是mse函数，而对于分类和排序任务可以使用常见的logloss函数，在这篇文章中，我们使用聚焦在使用mse函数的回归任务上。

过拟合问题本身就不多说了。主要提的是，因为AFM模型相比于FM模型具有更强的表达能力，因此在训练数据上有可能更容易过拟合，文章中主要考虑了dropout和L2正则这两种防止过拟合的方式。

dropout方式是通过防止神经元之间的共现性从而防止过拟合。由于AFM模型中会学习所有的特征之间的二阶交互特征，因此更加容易导致模型学习特征之间的共现性从而更容易导致过拟合，因此在pair-wise交互层使用了dropout方法来避免共适性。

对于AFM模型中的attention network，它是一个单层的MLP网络，这里使用L2正则化来防止过拟合，对于attention network，不选择dropout防止过拟合。

因此我们实际需要优化的目标函数为：

$$L = \sum_{x \in \tau} (\hat{y}_{AFM} - y(x))^2 + \lambda ||w||^2 \qquad (13)$$

5 相关的工作

在之前的工中，FMs在建模稀疏特征的任务中发挥了很重要的作用，但是相比于**MF（矩阵分解）模型的只能建模两个实体之间的交互作用，FM模可以作为更一般的学习器去建模任意数量的实体之间的交互性**。通过指定特定的输入特征向量，**FM模型可以囊括很多不同的分解模型**，包括 MF、parallel factor analysis、SVD++等。因此梳理了下常见的建模稀疏特征的方式有（这也是实验部分做对对照的依据）

1. FM
2. neural FM：在神经网络中加深FM的深度从而学习到更高阶的特征交互关系；
3. FFM：将一个特征的多个embedding向量和其他不同阈特征的交互关系区别开来；
4. GBFM：使用梯度提升算法选择优秀的特征，并且只建模优秀特征之间的交互关系；
5. Wide&Deep
6. Deep&Cross

6 Experiments

6.1 Experimental Settings

试验数据集为Frappe和MovieLens这两个，Frappe是用于上下文感知的推荐，MovieLens是用于用户电影评分的推荐。

评估方式：Frappe和MovieLens中已有的日志作为正样本1，对每条日志随机配对两条负样本-1。70% for training, 20% for validation, and 10% for testing。

用于跟AFM模型对比的算法模型为：

LibFM：FM的C++实现

HOFM：高阶交互特征的Tensorflow实现，阶数设置为3，Movielens只有user、item、tag这三种类型的预测变量

Wide&Deep

DeepCross

其中需要特别提到的是：**Wide&Deep**、**DeepCross**和**AFM**模型中，使用**FM**进行特征**embedding**的预训练相比于特征**embedding**的随机初始化方式，能得到更小的**RMSE**指标，因此使用embedding预训练的方式。

6.2 超参数设置

在AFM模型中，pair-wise交互层中设置了dropout正则化的方式，attention网络层设置了L2正则化的方式。

其他见paper中，也讲解了对FM和LibFm的实验方式。

6.3 Attention网络层的影响

paper需要选择合适的attention factor值t，总稳重给出的实验结果可以看出AFM模型随着不同的attention factor值，模型的效果比较稳定。当attention factor值t=1时，attention网络层就退化成为一个线性回归模型。AFM模型比较稳定，相比于FM模型提升明显，这也证明了AFM涉及的合理性，即其通过评估基于交互向量的特征交互的重要性分数来构成AFM模型的关键设计思想。

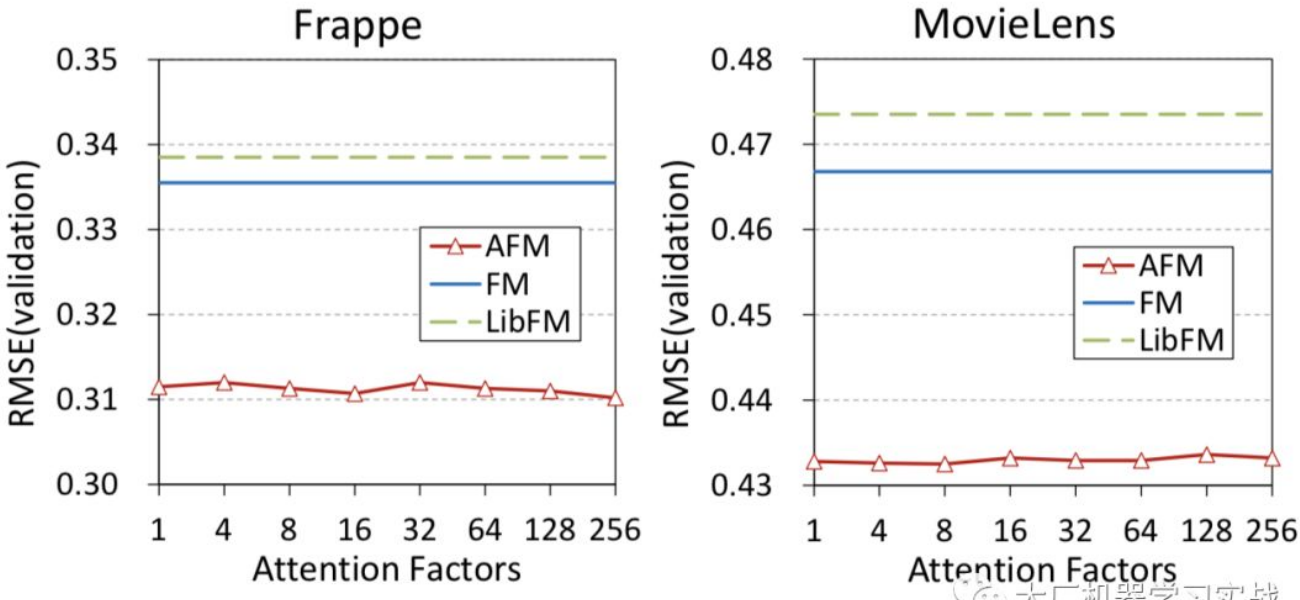

Figure 4: Validation error of AFM *w.r.t.* different attention factors

下图也显示出了AFM模型相比于FM模型的更快的收敛速度和在测试集上更好的模型效果。

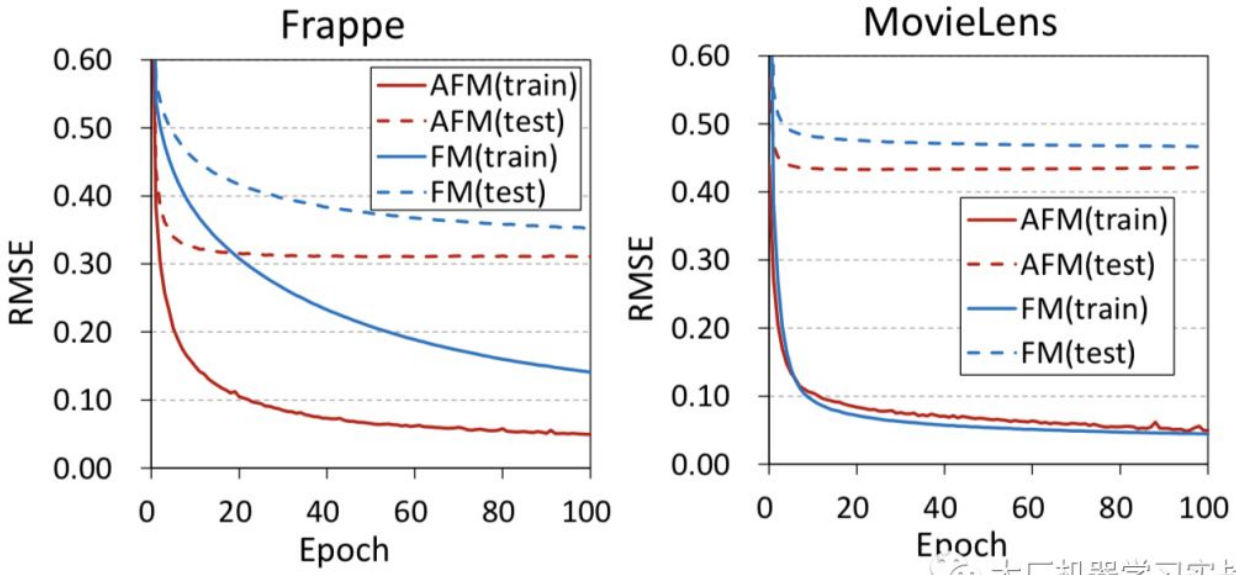

Figure 5: Training and test error of each epoch

6.3.1 微观分析

主要提到了文章通过AFM的网络结构设计，实现了交互特征的以attntion分数作为可解释性的指标。在这一部分，paper也通过实验，计算并展示了每个特征交互的attention分数和交互分数，可以从表1中看到 `attention_score * interaction_score` 的结果作为一个交互特征的重要性，相比而言，FM模型对每个交互特征的attention分数是完全一样的(表格中FM对应的row中的0.33的值)。在FM基础上引入attention网络，可以加强Item-Tag交互特征的重要性，因此使得预测结果的误差更小（三个测试用例的label都为1）

Table 1: The attention_score*interaction_score of each feature interaction of three test examples on MovieLens.

#	Model	User-Item	User-Tag	Item-Tag	$\hat{y}$
1	FM	0.33*-1.81	0.33*-2.65	0.33*4.55	0.03
	FM+A	0.34*-1.81	0.27*-2.65	0.38*4.55	0.39
2	FM	0.33*-1.62	0.33*-1.00	0.33*3.32	0.24
	FM+A	0.38*-1.62	0.20*-1.00	0.42*3.32	0.56
3	FM	0.33*-1.40	0.33*-1.26	0.33*4.68	0.67
	FM+A	0.33*-1.40	0.29*-1.26	0.3[illegible]4.68	0.89

6.3.2 模型效果对比

Table 2: Test error and number of parameters of different methods on embedding size 256. M denotes “million”.

	Frappe		MovieLens	
Method	**Param#**	**RMSE**	**Param#**	**RMSE**
LibFM	1.38M	0.3385	23.24M	0.4735
HOFM	2.76M	0.3331	46.40M	0.4636
Wide&Deep	4.66M	0.3246	24.69M	0.4512
DeepCross	8.93M	0.3548	25.42M	0.5130
AFM	**1.45M**	**0.3102**	**23.26M**	**0.4225**

文章对几个模型在模型效果以及模型容量上进行对比，比较容易从表格中得出下面的结论：

1. AFM模型的效果最好，尽管AFM模型是一个浅层模型，但其具有优于深度学习方法的效果；
2. HOFM算法模型相比于FM模型，效果有轻微的提升，但由于HOFM使用一组单独的embedding集合来建模每一阶的特征交互，导致模型容量几乎翻倍，因此性价比不高。但是这也给后续的研究提出了新的研究方向-使用更高效的方法来捕捉告诫特征交互关系。
3. DeepCorss模型，效果甚至比FM和HOFM模型效果更差，主要原因是DeepCorss模型的过拟合问题，因为交互特征的阶数较高，导致模型容易过拟合，这个问题在DeepCorss原文中也有说道（使用early stopping来代替L2和dropout方法防止过拟合）。

7 文章总结

基于FM模型进行改进，主要通过引入attention网络来学习交互特征的重要性，以此提高了模型的表达能力和可解释性。

作者也提到了AFM模型待改进的方面：0. 优化AFM模型版本，在基于attention的池化层上，堆叠多层非线性网络（目前只有一层）

1. AFM模型的复杂度为非零特征数量的平方，例如可以借鉴使用学习hash的方式和数据采样技术来降低复杂度
2. 转向半监督和多视角学习的方式
3. 探索AFM模型在其他领域的应用，例如问答系统等。

8 参考代码

https://github.com/hexiangnan/attentional_factorization_machine