**

## HUNAN UNIVERSITY

本科生毕业设计(论文)

|  |  |  |
| --- | --- | --- |
| **设计（论文）题目：** | | 基于概率矩阵分解的深度推荐神经网络的设计与实现 |
|  |  |  |
|  | 学生姓名： |  |
|  | 学生学号： |  |
|  | 专业班级： |  |
|  | 学院名称： | 信息科学与工程学院 |
|  | 指导老师： | 黄尚戎 |
|  | 学院院长： | 李肯立 |

2019年5月15日

湖 南 大 学

毕业设计（论文）原创性声明

本人郑重声明：所呈交的设计（论文）是本人在导师的指导下独立进行研究所取得的研究成果。除了文中特别加以标注引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写的成果作品。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到本声明的法律后果由本人承担。

学生签名： 日期：2019年5月15日

毕业设计（论文）版权使用授权书

本毕业设计（论文）作者完全了解学校有关保留、使用设计（论文）的规定，同意学校保留并向国家有关部门或机构送交设计（论文）的复印件和电子版，允许设计（论文）被查阅和借阅。本人授权湖南大学可以将本设计（论文）的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存和汇编本设计（论文）。

本设计（论文）属于

1、保 密 ，在 年解密后适用本授权书。

2、不保密 √ 。

（请在以上相应方框内打“√”）

学生签名： 日期：2019年 5 月 15日

导师签名： 日期：20 年 月 日

基于概率矩阵分解的深度推荐神经网络的设计与实现

摘 要

在推荐系统领域中，由于用户到项目评分的稀疏性，往往会造成推荐系统推荐质量的恶化和功能衰退。特别是存在需要处理冷启动的情况时，更是会造成不可预知的恶劣效果。为了解决这个问题，一些考虑辅助因素的推荐算法在近几年来成为了主流。比如词袋模型，它通过利用一些文本信息（例如评论，星级等方法）来对评分模型进行构建。在最近的工作中，出现了一种通过集成卷积神经网络（CNN）到概率矩阵分解的方法来进行模型的适配的方法。这种方法的好处在于能够利用卷积核来获取局部特征。但是，如果考虑评分模型的序列性，为了能够达到更好的效果，我们通过考虑加入循环神经网络（RNN）来对评分信息的时序性进行规约。本文提出并实现了一种集成RNN到ConvFM的方法，并利用在KAGGLE 2017 KKBOX的比赛中的数据进行了实验。实验结果表明，我们的模型取得了较为良好的效果。即使在评分数据十分稀疏的情况下，我们依然能够从上下文中取得能够用于建模的文档信息。

关键词：推荐系统；卷积神经网络；循环神经网络；kaggle竞赛

**Design and Implementation of Depth Recommendation Neural Network Based on Probabilistic Matrix Decomposition**

In the field of recommendation system, the sparseness of user to item scoring often leads to deterioration of recommendation quality and decline of function. Especially when there is a need to deal with cold start, it will cause unpredictable adverse effects. In order to solve this problem, some recommendation algorithms considering auxiliary factors have become the mainstream in recent years. For example, the word bag model, which uses some text information (such as comments, stars, etc.) to construct the scoring model. In recent work, a method of adapting models by integrating convolutional neural network (CNN) into probability matrix decomposition (PMD) has emerged. The advantage of this method is that convolution kernels can be used to obtain local features. However, if we consider the sequentiality of the scoring model, in order to achieve better results, we consider adding a recurrent neural network (RNN) to regulate the sequentiality of the scoring information. This paper presents and implements a method of integrating RNN into ConvFM, and makes experiments using data from KAGLE 2017 KKBOX. The experimental results show that our model achieves good results. Even when the scoring data is sparse, we can still get document information from the context that can be used for modeling.

**Key Words:** **Recommendation System; Convolutional Neural Network; Recurent Neural Network; Kaggle Competition**

目 录

毕业设计（论文）原创性声明和毕业设计（论文）版权使用授权书…………….…….Ⅰ

摘 要………………………………………………………………………………….………Ⅱ

Abstract……………………………………………………………………………Ⅲ

插图索引……………………………………………………………………………Ⅴ

附表索引……………………………………………………………………………Ⅵ

**1**绪论………………………………………………………..………………………..……….1

1.1课题背景及目的……………………………………………………………….………1

1.2 国内外研究状况………………………………………………..…………….….……2

1.3 课题研究方法………………………………………………………………....………3

1.4 论文构成及研究内容……………………………………………………...….………4

**2** I级叶/盘协调转子固有振动特性分析…………………………………..………......……5

2.1基础知识………………..…………………………………………………...……….5

2.1.1有限元法……………………………………….……………..…….…………5

2.1.2 循环对称结构的分析方法……….……………………………….…………6

**3** 结论……………………………………………………………………………….………13

参考文献…………………………………………………………………….………..14

致谢………………………………………………………………………………….…..15

附录…………………………………………………………………………………….……..16

附录A……………………………………………………………………………….……16

附录B………………………………………………………….……………………….…17

插图索引

图2.3部分相干解调与相干和非相干解调平均误码性能的比较………………3

图3.1单管换热系统流程图………………………………………..……………….5

附表索引

表2.1 方法—干扰抑制结果…………………………………………..…………………5

表3.1 各组分lgBi值………………………………………………………….…………7

**1** 绪论

1.1课题背景及目的

在推荐系统中，通常通过观察用户-项目评分矩阵来进行算法的推导和研究。但是，随着电子商务服务中用户和项目数量的爆炸式增长，导致用户-项目矩阵的维度呈现指数增加的趋势，伴随而来的就会造成评分矩阵的稀疏。该特性也是推荐系统质量恶化的主要因素之一。为了提高准确性，用于特定场景的推荐技术要求不仅考虑用户的评分信息，还要考虑用户所在地区的人口统计信息，基于用户行为的社会网络以及指定相关项目描述的一些辅助信息。我们的目的在于通过设计一些中间的过渡过程（比如文档的时序性等附加信息），来平滑在预测过程中出现的系统推荐效果波动过大的问题。

1.2 国内外研究状况

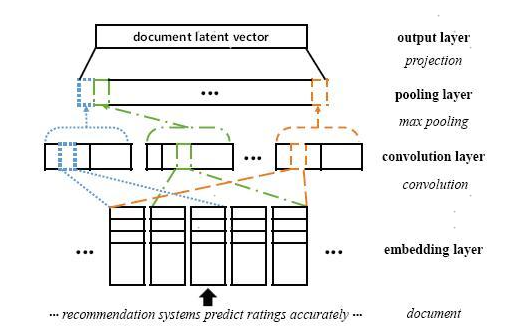
目前，研究者们已经提出了基于文档建模方法（如LDA主题模型等方法）以及能够产生更加准确的潜在模型的CDL协同深度学习的方法。在最近的工作成果里，利用卷积神经网络（CNN）来进行文档感受野的采集，并利用下采样来防止过拟合的思路获得了一定成都的成功。但是，使用CNN进行用户-项目矩阵分解（ConvMF）的缺陷在于其不能体现出单个句子中的词序关系。而词序在一定程度上决定了语句的内在含义和思想。

但是，这种方法造成的缺陷在于，它们是基于词袋模型进行模型搭建的，也就是说每一个词在句子中的顺序是不确定的，也就是说，现有的集成模型并不能完全捕获句子的内建信息，因为他们不能利用预先训练的词嵌入模型（如Glove）

在最近的工作中，提出了一种CNN和MF结合的模型ConvMF，这种模型的优点在于其通过模型组件（卷积核）捕获了文档之间的上下文信息，通过增加CNN层和下采样层到PMF中，比较好地处理了上下文之间地信息稀疏的问题。但是，针对于单个语句中词序的关系，该模型没有对词粒度的信息进行挖掘。因此，对词粒度信息的挖掘是提升推荐系统性能的下一个发展方向

1.3 课题研究方法

目前提出的CNN具有如下的结构：



可以看到，这样一个包含了卷积层（CNN），池化层（PL）和全连接层（ProL）的结构，可以比较好地在用户-项目矩阵上发挥效果，其中卷积层用户在上下文提取特征，由于文档输入的特殊性，我们不能直接将词向量对卷积层进行输入。因此，我们在卷积层之前加入了一个嵌入层以保证将原始文档转换成表示下一个卷积层的文档的数字密集矩阵。从更进一步的角度，关于该文件，我们可以作为I个单词的序列，我们将通过把文档中的单词向量进行连接来保证文档可以被组织为一个可用于提取向量的矩阵。为了优化模型训练的时间，我们可以利用一些预训练的模型如GLOVE。通过预训练的权重因子的加入，我们可以进一步缩短训练过程，并且达到跟从0开始训练同等的效果。

但是，由于词袋模型的存在，CNN结构仅能从无序的感受野上获取信息，因此，我们的想法实在enbedding layer之后CNN之前加入RNN层以满足词序性的要求。这样，我们就可以在输入之前就充分挖掘用户向量单元的顺序性和相关性，更能够达到我们需要的效果。

考虑到推荐系统的长短时间相关性，我们考虑用RNN的变体LSTM来解决上述问题，LSTM由遗忘门，输入门和输出门组成，其原理在于通过遗忘门过滤掉间隔较远的时间的信息，之后通过输入门改变隐向量的状态，最后通过输出门的到相应结果

1.4 论文构成及研究内容

在第二小节中，我们将较为详细的介绍我们对问题的理解以及一些可用于解决该问题的方法，该小节有利于读者从我们介绍的基本信息中获取我们的思路逻辑；在第三节中，我们将介绍我们所设计的算法的具体思路及实现过程，包括对CNN层的改进以及对RNN/LSTM层的参数设定和框架描述；在第四节中，我们将利用KAGGLE 2017 KKBOX竞赛的数据和MOVIELEN的数据对我们的算法准确度进行实验，以对比该算法在这两种不同数据集上的精度表现；最后，我们将对我们的算法和现在主流的算法进行一个直接的比较，以便于我们更加了解不同算法的优劣性。

**2 论文概述**

**2.1 问题定义**

在开篇中，我们对问题进行了初步的定义：即我们想设计一种模式，以便在CNN层中间加入信息保留的机制。这样就算应对稀疏的数据，我们也能根据之前获取的信息，对之后的信息进行较为准确的预测。于是，我们需要解决的问题，在于如何确定这种机制的结构，以及如何保证该结构对于数据不同程度的稀疏能够保持较为良好的适应性和稳定性。

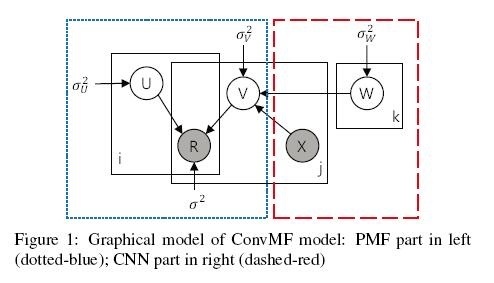
**2.2 ConvFM**

ConvFM是一种上下文环境设置推荐模型。通过利用卷积神经网络（CNN）捕获项目描述文档的上下文信息，进一步提高了评分预测精度。ConvMF正是将CNN无缝集成到PMF中，而PMF通常用于推荐任务。因此，综合模式遵循推荐目标，最终有效地利用协作信息和上下文信息。取得较为良好的效果。

虽然CNN最初是为计算机视觉开发的，但是CNN的关键思想已经积极应用于信息检索和NLP，如搜索查询检索，句子建模和分类和其他传统的NLP任务。虽然针对NLP任务需要对CNN架构进行大量修改，但最终都有助于提高各种NLP任务的性能。

然而，CNN尚未被积极应用于推荐系统领域。据我们所知，van den Oord等人首先将CNN应用于音乐推荐，通过声学分析观点利用CNN分析了歌曲，并提出了一种基于声学CNN获得的项目潜在模型来预测评分的模型。然而，他们的CNN模型，专为声信号处理而设计，不适合处理文档。文档和声学信号对周围特征的质量有固有的差异。一定时间的信号固有地类似于其周围的信号，即具有轻微时间差的信号，而文档中某个位置处的一个字与周围的字有很大的语义差异。周围特征之间的这种相似度的差异影响局部特征的质量，最终需要不同的CNN架构。此外，该模型还没有充分反映协同信息。具体来说，项目潜在模型主要由通过CNN进行音频信号分析的结果而不是协同信息决定。因此，总体推荐的性能甚至没有达到加权矩阵分解（WMF），WMF这是处理隐式反馈数据集的常规基于MF的协同过滤技术之一。

ConvFM作为解决稀疏问题的主流方法之一，其基本的框架图如下：



图一显示了ConvMF的概率模型的概述，并且展示了如何将CNN集成到PMF的过程：假设我们有N个用户和M个项目，那么通过因子聚合方法我们可以得到R=M\*N的用户-项目评分矩阵。然后，我们的目标是通过概率分解模型重建R模型。即我们需要得到一个U\*V的矩阵，使得R=M\*N≈U\*V。

在上一步过程中，得到分解矩阵U\*V后，我们在这两个矩阵的基础上分开训练两个CNN模型，这个CNN模型和上文提到的一样，同样包含了四个层：嵌入层（Emb）,卷积层（CNN），池化层（PL）和全连接层（ProL）。同样的，对于Emb层，我们可以使用预训练的Glove模型来缩短模型构建时间。

**3 算法设计**

在上面的描述中，我们介绍了一种用于解决特征矩阵稀疏性的方法ConvMF。在这一节中，我们将对该方法进行改进。得益于时间序列模型建模方式的启发，我们将在CNN层之前加入LSTM层以保证数据组织的有序性能够被有效地探查到。同时，我们将加深CNN网络的规模，尝试利用模型层过拟合+dropout的方式来缓解由于模型层数过深而产生的过拟合的情况。

**3.1 CNN层设计**

在这里，我们希望能够找到一种CNN结构，以便我们在加深网络高度和宽度的同时避免一些由于网络自身产生的原因而造成的模型不可用的情况。可能造成的情况有以下几点：

1.参数太多，若训练数据集有限，容易过拟合；

2.网络越大计算复杂度越大，难以应用；

3.网络越深，梯度越往后穿越容易消失，难以优化模型。

解决上面几个问题的根本方法是减少网络节点之间的连接数量。因此在近几年的研究当中，许多取得了不错效果的CNN通常取消了全连接层，改而用一般的多维卷积或者其他特殊的方法将稠密连接转化为稀疏连接。在2011年以前，为了打破网络的对称性和提高学习能力，传统的网络都是用了一种名为随机稀疏连接的方法来维持网络的稀疏性。但是，由于计算机硬件能力的制约，使得当时的计算水平对于非均匀分布的稀疏数据的计算能力远远不能满足模型本身设计的需求。因此现在的问题就可以归结为有没有一种方法，既能保持网络结构的稀疏性，又能利用密集矩阵的高计算性能。

为了解决这个问题，我们在实验过程中引入可GOOGLE与2013年提出的Inception V模型。首先，Inception架构的主要思想是找出如何用密集成分来近似最优的局部稀疏解。也就是说，我们可以通过Inception结构，代替人工确定卷积层中的过滤器类型或者确定是否需要创建卷积层和池化层。简单的说，通过加入Inception层，我们可以让网络自行决定需要添加的卷积核以及池化层的参数，可以给网络提供所有的可能值，并且将输出连接起来，让网络自己学习他需要什么样的参数。

最开始的Inception V1的结构有22层（见附录）。但是由于V1引入了一些较大的卷积核参与计算（7\*7卷积核），导致了运算复杂度的成倍增加。在V3中对V1的结构进行了改进。改进的主要思想在于：大的卷积核完全可以有一系列小的卷积核来替代。由于替代过程是一个线性叠加的过程，因此采用若干个3\*3和1\*1的卷积核代替7\*7的卷积核被认为是可行的。并且，V3将一个较大的二位卷积核按照方向拆成了两个较小的一维卷积，并且加入了著名的BN（Batch Normalization）方法。BN方法是一种非常有效的正则化的方法，可以让大型卷积神经网络的训练速度加快很多倍，同时收敛后的分类准确率也可以得到大幅度提高。

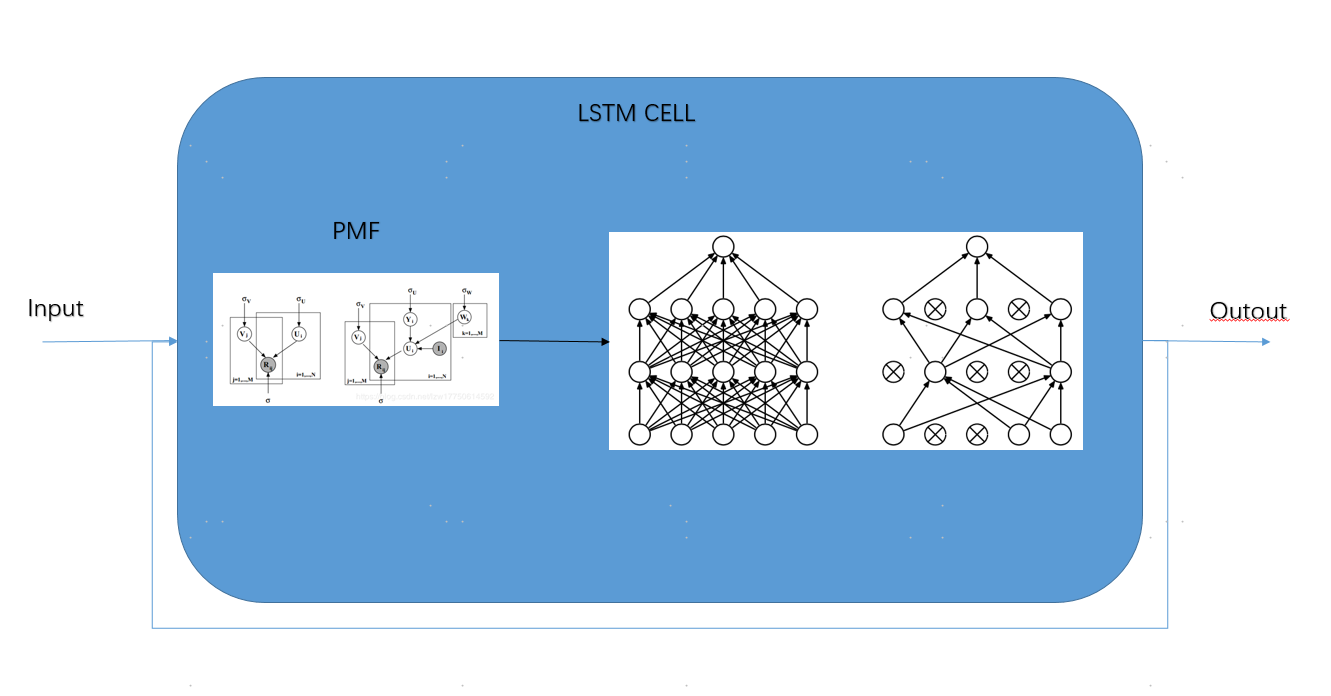
基于上面的介绍，我们拟采用Inception V3模型替代原有模型中的CNN模型。

**3.2 LSTM层设计**

在上文中提到，我们计划在CNN层之前加入LSTM层以挖掘文本信息中非本身特征（包括序列特征和隐性特征）。在我们的设计中，仅使用单一LSTM CELL作为特征筛选的门矩阵。在进行建模的过程中，采用可变的学习率（learning rate）进行梯度的获取和迭代，最终获得一个能够反应CNN过程中不能体现的时序特征的LSTM模型。

**3.3 模型总体设计**

基于上面的讨论，我们设计出的模型的整体框架如下图所示：



其中我们在Input层之后并没有直接跟Embbeding层，这是为了减少Embedding层拟合过程中的error对于CNN建模的影响。另外，我们的Embedding层加在PMF之后能够更好地获取PMF提供的特征隐藏信息。这样，数据通过Input层-PMF-CNN-Output，构成了一个End-to-End的结构系统。

其中CNN和LSTM层的模型参数见附录。

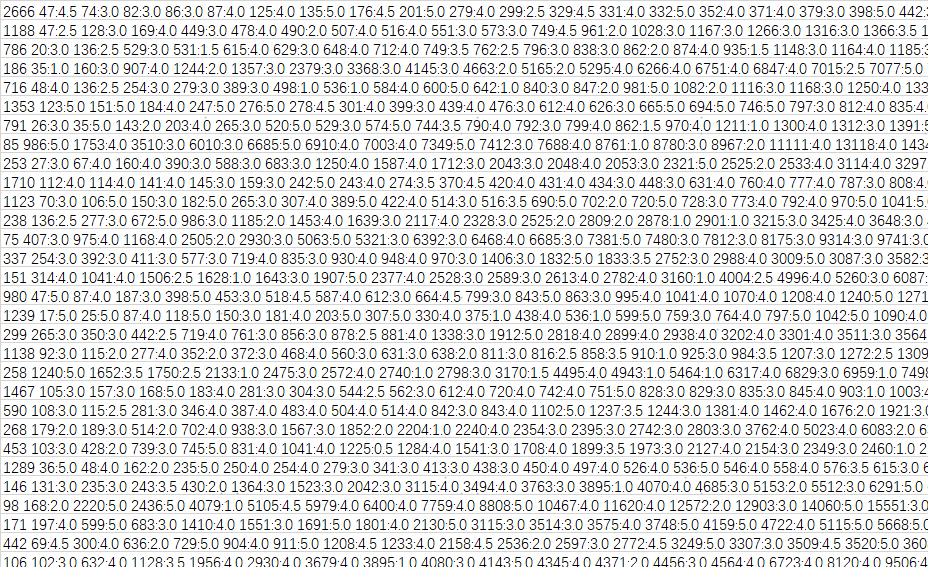
**4 实验**

在本节中，我们将利用上文中提到了Kaggle 2017 KKBOX的比赛数据和MovieLen的数据对我们的算法进行检验。

在模型构建的过程中，一个非常重要的步骤是对数据进行归一化。归一化的好处在于我们可以针对不同数据类型的单位进行统一处理，以消除因为不同单位造成的数据尺度不一的问题。在我们的实验中，进行归一化的具体方法如下：

def normalizeRatings(rating, record):  
 m, n = rating.shape  
 # m代表电影数量，n代表用户数量  
 rating\_mean = np.zeros((m, 1))  
 # 每部电影的平均得分  
 rating\_norm = np.zeros((m, n))  
 # 处理过的评分  
 for i in range(m):  
 idx = (record[i, :] != 0)  
 # 每部电影的评分，[i，:]表示每一行的所有列  
 rating\_mean[i] = np.mean(rating[i, idx])  
 # 第i行，评过份idx的用户的平均得分  
 # np.mean() 对所有元素求均值  
 rating\_norm[i, idx] = rating[i, idx] - rating\_mean[i]  
 # rating\_norm = 原始得分-平均得分  
 return rating\_norm, rating\_mean

采用的数据的基本格式如下（数据的基本格式解析见MOVIELEN官网）：

****

我们对于的设想是能够指定一个K值，模型在构建完毕之后能够输出K个最匹配的项目。因此，我们在代码构建的过程中加入了一个process模块，来进行模型预测结果的输出：

def process():  
 user\_id = input(u'您要想哪位用户进行推荐？请输入用户编号：')  
 sortedResult = predicts[:, int(user\_id)].argsort()[::-1]  
 # argsort()函数返回的是数组值从小到大的索引值; argsort()[::-1] 返回的是数组值从大到小的索引值  
 print(u'为该用户推荐的评分最高的20部电影是：'.center(80, '='))  
 # center() 返回一个原字符串居中,并使用空格填充至长度 width 的新字符串。默认填充字符为空格。  
 idx = 0  
 for i in sortedResult:  
 print(u'评分: %.2f, 电影名: %s' % (predicts[i, int(user\_id)] - 2, movies\_df.iloc[i]['title']))  
 # .iloc的用法：https://www.cnblogs.com/harvey888/p/6006200.html  
 idx += 1  
 if idx == 20:  
 break

在构建Inception网络结构的过程中，我们拟采用tensorflow slim进行实现。TF-Slim 是 TensorFlow 中一个用来构建、训练、评估复杂模型的轻量化库。TF-Slim 模块可以和 TensorFlow 中其它API混合使用。Slim 模块可以使模型的构建、训练、评估变得简单：

1.允许用户用紧凑的代码定义模型。这主要由 arg\_scope、大量的高级 layers 和 variables 来实现。这些工具增加了代码的可读性和维护性，减少了复制、粘贴超参数值出错的可能性，并且简化了超参数的调整。

2.通过提供常用的 regularizers 来简化模型的开发。很多常用的计算机视觉模型（例如 VGG、AlexNet）在 Slim 里面已经有了实现。这些模型开箱可用，并且能够以多种方式进行扩展（例如，给内部的不同层添加 multiple heads）。

3.Slim使得 “复杂模型的扩展” 及 “从一些现存的模型 ckpt 开始训练” 变得容易。

基于上面的考虑，我们决定使用slim作为模型背后的框架。

执行代码后，我们得到的输出结果如下：

您要想哪位用户进行推荐？请输入用户编号：1

==============================为该用户推荐的评分最高的20部电影是：===============================

评分: 5.06, 电影名: Like Water for Chocolate (Como agua para chocolate) (1992)

评分: 4.68, 电影名: Sense and Sensibility (1995)

评分: 4.55, 电影名: Evita (1996)

评分: 4.47, 电影名: Shallow Grave (1994)

评分: 4.41, 电影名: Mighty Aphrodite (1995)

评分: 4.34, 电影名: Whale Rider (2002)

评分: 4.31, 电影名: Annie (1982)

评分: 4.30, 电影名: Interstellar (2014)

评分: 4.27, 电影名: To Catch a Thief (1955)

评分: 4.26, 电影名: Arrival (2016)

评分: 4.24, 电影名: Dead Again (1991)

评分: 4.15, 电影名: Dead Man Walking (1995)

评分: 4.12, 电影名: Orgazmo (1997)

评分: 4.10, 电影名: Saving Grace (2000)

评分: 4.09, 电影名: Purple Rose of Cairo, The (1985)

评分: 4.09, 电影名: Let the Right One In (Låt den rätte komma in) (2008)

评分: 4.07, 电影名: Half Baked (1998)

评分: 4.06, 电影名: Out of Sight (1998)

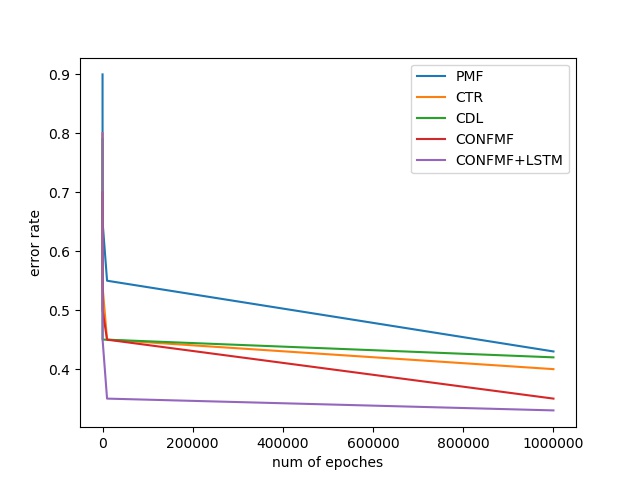
评分: 4.05, 电影名: Princess and the Warrior, The (Krieger und die Kaiserin, Der) (2000)

评分: 4.04, 电影名: Doctor Strange (2016)

样7（3）

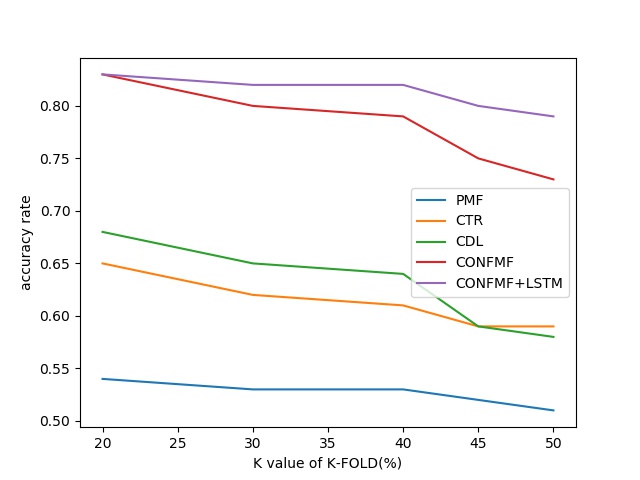
**4.1 实验结果分析**

首先，我们通过对比现有主流算法和我们构建模型在相同数据集上的错误率来比较它们的性能差别，如下图：



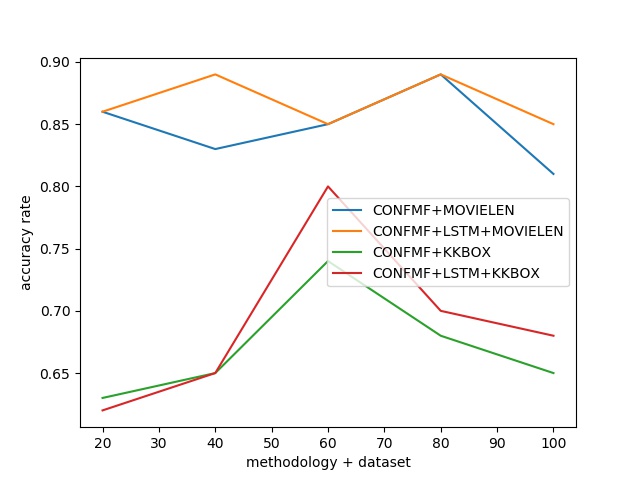
可以看出，对比起主流的模型，我们的模型拥有更加低的错误率，这就意味着我们的模型在特定数据集上表现出了极高的性能。但是由于网络的深度和宽度太过于巨大，因此可能会出现过拟合的情况，我们可以通过K-fold检验对模型的拟合程度进行分析。

K-fold检验结果如下图：



可以看到，在以accuracy rate为衡量标准的K-fold检测环节，我们的模型同样表现出了很高的竞争力。其中值得一提的是远远高出了原有的传统PMF算法。

下面展示的是我们的模型在不同数据集上与现有主流的ConvMF算法的区别：



可以看到，我们的模型在相同的数据集上同样表现得十分优异。

**5 结论**

在本文中，我们提出了一种基于深度卷积神经网络的概率矩阵分解模型的改进方法。改进的过程有两点：（1）.采用Inception V3结构替代原有的CNN结构，扩大了特征提取网络的感知域，并且能够在稀疏连接的前提下提升了计算的性能。（2）在CNN层之前加入了LSTM层，实现了隐性特征（如序列特征）的提取，并且LSTM层以包装器的形式加入，既不影响内部CNN+PMF的结构，又将整个CELL作为一个整体提取了对应的特征。在迭代过程中，我们采用了0.5的learning rate和0.9的momenton作为迭代参数，进一步加快了迭代的速度，同时减少了过拟合的情况。最后，我们将设计的模型用于KAGGLE 2017 KKBOX比赛数据集和MOVIELEN数据集并与现有的主流模型进行了对比，对比结果表明我们的模型在accuracy rate上占有优势，但是可能会出现一定程度的过拟合。在今后的工作中，我们将尝试将BN（Batch Normalization）加入不同的CNN 层中，以减少训练数据之间的关联性，保证每条训练数据都可以在一个独立的环境下被独立地迭代和运算，以取得不依赖于数据的隐性序列关系的更好的推荐效果。

参考文献

[1] Dario Amodei, Sundaram Ananthanarayanan, Rishita Anubhai, Jingliang Bai, Eric，Battenberg, Carl Case, Jared Casper, Bryan Catanzaro, Qiang Cheng, Guoliang

Chen, et al. 2016. Deep speech 2: End-to-end speech recognition in english and

mandarin. In International Conference on Machine Learning. 173–182.

[2] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine

translation by jointly learning to align and translate. arXiv preprint

arXiv:1409.0473 (2014).

[3] DavidMBlei, AndrewY Ng, and Michael I Jordan. 2003. Latent dirichlet allocation.

Journal of machine Learning research 3, Jan (2003), 993–1022.

[4] Emmanuel J Candès and Benjamin Recht. 2009. Exact matrix completion via

convex optimization. Foundations of Computational mathematics 9, 6 (2009), 717.

[5] Li Chen, Guanliang Chen, and Feng Wang. 2015. Recommender systems based

on user reviews: the state of the art. User Modeling and User-Adapted Interaction

25, 2 (2015), 99–154.

[6] Kyunghyun Cho, Bart Van Merri boer, Caglar Gulcehre, Dzmitry Bahdanau,

Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase

representations using RNN encoder-decoder for statistical machine translation.

arXiv preprint arXiv:1406.1078 (2014).

[7] Robert Desimone and John Duncan. 1995. Neural mechanisms of selective visual

attention. Annual review of neuroscience 18, 1 (1995), 193–222.

[8] Qiming Diao, Minghui Qiu, Chao-Yuan Wu, Alexander J Smola, Jing Jiang, and

Chong Wang. 2014. Jointly modeling aspects, ratings and sentiments for movie

recommendation (jmars). In Proceedings of the 20th ACM SIGKDD international

conference on Knowledge discovery and data mining. ACM, 193–202.

[9] Ruihai Dong, Michael P O’Mahony, Markus Schaal, Kevin McCarthy, and Barry

Smyth. 2013. Sentimental product recommendation. In Proceedings of the 7th

ACM conference on Recommender systems. ACM, 411–414.

[10] Ruihai Dong, Michael P O’Mahony, and Barry Smyth. 2014. Further experiments

in opinionated product recommendation. In International Conference on Case-

Based Reasoning. Springer, 110–124.

[11] Ruihai Dong, Markus Schaal, Michael P O’Mahony, Kevin McCarthy, and Barry

Smyth. 2013. Opinionated product recommendation. In International Conference

on Case-Based Reasoning. Springer, 44–58.

[12] Ruihai Dong, Markus Schaal, Michael P O’Mahony, and Barry Smyth. 2013. Topic

Extraction from Online Reviews for Classification and Recommendation.. In

IJCAI, Vol. 13. 1310–1316.

[13] Ruihai Dong and Barry Smyth. 2016. Personalized Opinion-Based Recommendation.

In International Conference on Case-Based Reasoning. Springer, 93–107.

[14] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N

Dauphin. 2017. Convolutional sequence to sequence learning. arXiv preprint

arXiv:1705.03122 (2017).

[15] Alex Graves and Jürgen Schmidhuber. 2005. Framewise phoneme classification

with bidirectional LSTM and other neural network architectures. Neural Networks

18, 5-6 (2005), 602–610.

[16] Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will

Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and

comprehend. In Advances in Neural Information Processing Systems. 1693–1701.

[17] Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A convolutional

neural network for modelling sentences. arXiv preprint arXiv:1404.2188 (2014).

[18] Timothy L Keiningham, Bruce Cooil, Lerzan Aksoy, Tor W Andreassen, and Jay

Weiner. 2007. The value of different customer satisfaction and loyalty metrics in

predicting customer retention, recommendation, and share-of-wallet. Managing

Service Quality: An International Journal 17, 4 (2007), 361–384.

[19] Donghyun Kim, Chanyoung Park, Jinoh Oh, Sungyoung Lee, and Hwanjo Yu.

2016. Convolutional matrix factorization for document context-aware recommendation.

In Proceedings of the 10th ACM Conference on Recommender Systems.

ACM, 233–240.

[20] Yoon Kim. 2014. Convolutional neural networks for sentence classification. arXiv

preprint arXiv:1408.5882 (2014).

[21] Yehuda Koren. 2008. Factorization meets the neighborhood: a multifaceted

collaborative filtering model. In Proceedings of the 14th ACM SIGKDD international

conference on Knowledge discovery and data mining. ACM, 426–434.

[22] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques

for recommender systems. Computer 42, 8 (2009).

[23] Jiwei Li, Will Monroe, Alan Ritter, Michel Galley, Jianfeng Gao, and Dan Jurafsky.

2016. Deep reinforcement learning for dialogue generation. arXiv preprint

arXiv:1606.01541 (2016).

[24] Greg Linden, Brent Smith, and Jeremy York. 2003. Amazon. com recommendations:

Item-to-item collaborative filtering. IEEE Internet computing 7, 1 (2003),

76–80.

[25] Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective

approaches to attention-based neural machine translation. arXiv preprint

arXiv:1508.04025 (2015).

[26] Julian McAuley and Jure Leskovec. 2013. Hidden factors and hidden topics:

understanding rating dimensions with review text. In Proceedings of the 7th ACM

conference on Recommender systems. ACM, 165–172.

致 谢

这篇耗时半年的毕业论文终于写完,在电脑上敲下最后一个字的时候，我有一些成就感

……………

最后，感谢大学四年的生活!

附录A Inception V3参数

层 参数量

conv0 864

conv1 9216

conv2 18432

conv3 5120

conv4 138240

256a 254976

288a 276480

288b 284160

768a 1152000

768b 1294336

768c 1687552

768d 1687552

768e 2138112

aux 2712576

1280a 1695744

2048a 5038080

2048b 5922816

fc 417792

共 24734048

样张8（1）