

分类号: _____
UDC _____

密级: 公开
学号: 2016022292

华南师范大学

South China Normal University

硕士学位论文

(学术学位)

基于序列感知的推荐算法研究

学位申请人: 付海林

专业学位名称: 软件工程

专业学位领域: 数据挖掘

所在院系: 计算机学院

导师姓名及职称: 李建国

2019 年 3 月 20 日

基于序列感知的推荐算法研究

专业名称： 软件工程

申请者： 付海林

导师： 李建国

摘 要

随着互联网信息的不断增长，推荐系统已成为克服信息过载的有效策略。推荐系统作为帮助人们从浩如烟海的信息中发现自己所需的重要工具，在许多网络应用中被广泛集成，不仅辅助人们做出选择还帮助服务商提升了交易量，推荐系统的效用不容小觑，已成为信息社会日常生活的重要组成部分。推荐算法作为推荐系统背后的核心力量，支撑着推荐系统的表现。因此研究泛化能力强鲁棒性好的推荐算法一直是数据挖掘领域的热点。

近年来，深度学习在计算机视觉和自然语言处理等许多研究领域引起了相当大的兴趣，这不仅源于其出色的大数据计算能力，还归功于其能从原始数据学习特征表达的迷人特性，这一影响早已扩散到了信息检索和推荐系统的研究领域。在现代推荐算法中，大多数方法都忽略了用户所消费物品之间的时序关系，并且对于缺少数据的用户，现代推荐系统也不能给出精准的结果。所以本文基于双向长短期记忆神经网络提出了一个新颖的时序推荐模型，它通过捕捉用户消费物品的时序特征来预测用户下一个的兴趣物品会是哪一个。借用迁移学习的能力来处理冷启动问题。

为了验证这一模型的性能，本文分别在真实的用户行为数据集 Movielens, Last.FM 上进行了实验，实验结果表明，基于双向长短期记忆神经网络的序列推荐模型，面对稀疏、大规模数据时，也能较好地反映用户的短期兴趣；迁移学习应用在推荐领域也能利用辅助信息提升推荐准确度。

关键词：推荐系统；循环神经网络；迁移学习；数据挖掘；冷启动

Research on sequence-aware recommendation

Major: Software Engineering

Name: Hailin Fu

Supervisor: Jianguo Li

ABSTRACT

KEY WORDS: Recommendation system; Recurrent neural network; Transfer learning; Data mining; Cold start

目 录

摘 要	I
ABSTRACT	III
第 1 章 绪论	1
1.1 研究背景与意义	1
1.2 国内外研究现状	2
1.2.1 个性化推荐研究现状	3
1.2.2 序列感知推荐研究现状	3
1.2.3 迁移学习研究现状	3
1.3 研究内容和方法	3
1.3.1 研究内容	3
1.3.2 研究方法	4
1.4 本文主要贡献	4
1.5 论文组织架构	5
第 2 章 个性化推荐、序列感知推荐及迁移学习	7
2.1 推荐系统	7
2.1.1 个性化推荐概述	7
2.1.2 推荐算法	8
2.1.3 推荐系统存在的问题	8
2.2 序列感知模型	9
2.2.1 序列感知推荐任务及场景	9
2.2.2 序列感知推荐技术分类	9
2.2.3 序列感知推荐算法评价指标	14

2.3	图神经网络	14
2.4	本章小结	14
第 3 章	基于双向长短期记忆网络的序列感知推荐	15
3.1	问题的提出	15
3.2	算法框架描述	15
3.3	基于双向长短期记忆网络的序列感知推荐模型	16
3.3.1	目标问题定义	16
3.3.2	嵌入层	17
3.3.3	用户短期兴趣学习	17
3.3.4	流行趋势学习	18
3.3.5	模型的训练	19
3.4	复杂度分析	19
3.4.1	算法实现	20
3.5	本章小结	20
第 4 章	基于自注意力机制的序列感知推荐	21
4.1	问题的提出	21
4.1.1	目标问题定义	21
4.1.2	嵌入层	22
4.2	自注意力模块	23
4.3	复杂度分析	27
4.3.1	算法实现	27
4.4	本章小结	27
第 5 章	实验结果与分析	29
5.1	数据集介绍	29
5.2	评价指标	29

5.3 实验结果与分析	30
5.3.1 关键词抽取结果	30
5.3.2 学者推荐结果	30
5.3.3 结果分析	30
5.4 本章小结	30
第 6 章 总结与展望	31
符号列表	33
致 谢	39
作者攻读学位期间发表的学术论文目录	41

第 1 章 绪论

1.1 研究背景与意义

我国经过二十余年的信息化建设，互联网用户仍在持续增长，根据中国互联网信息中心发布的第 43 次《中国互联网络发展状况统计报告》^[2]显示，截止 2018 年 12 月，我国网民规模达 8.29 亿，较 2017 年末仍增加了 3.8%，互联网普及率达 59.6%，其中移动互联网用户比例更是高达 98.6%。推荐系统是帮助用户从海量产品集合里面找到感兴趣目标的软件应用，具有千人千面的特点，也是数据挖掘和机器学习相关科技在实践领域最成功的应用之一。推荐系统面对的任务主要有两部分：评分预测和产品推荐。所以根据用户的历史记录预测他下一次会选择什么也是推荐领域一个严峻的挑战。在许多网站和应用程序中，例如电子商务、新闻和视频网站、音乐和广播电台等，他们都需要为用户推荐可能喜欢物品的杰出服务，如今接收不同形式的自动推荐已经成为我们日常在线用户体验的一部分。在典型的在线网站上面，可以收集用户各种类型的相关动作，例如：用户点击、浏览、收藏、购买了某个商品。现有的推荐策略都主要关注如何为用户或项目找到临近集，或者利用其它的显式或隐式信息（如标签、评论、项目属性和用户个人信息）来提升近邻感知能力。然而，这些静态算法都没有考虑用户兴趣变化的实时动态性。时序信息就是反应用户兴趣实时变化的一个重要特征，在许多任务比如用户下一行为预测中，用户下一首会听什么歌与用户的喜好、当前所处环境的上下文都高度相关。

推荐系统面临的另外一个巨大挑战就是处理新用户和新物品，也就是所谓的冷启动问题，因为这些用户 / 物品的特征在缺少数据时很难正确判断。近来，迁移学习^[2]被用于解决推荐系统的冷启动问题。迁移学习通过利用辅助域来在目标域上获得提升

本文的研究内容是结合神经网络的序列模型和推荐系统相关理论知识，实现快速向用户推荐可能感兴趣的物品。充分利用深度神经网络强大的建模能力，提取用户行为的重要时序特征，构建推荐对象的兴趣模型，再结合迁移学习的泛化

能力，构建相关模型来识别用户的兴趣意图，综合考虑推荐对象和用户两者之间的特征信息挖掘两者之间的隐式联系，从而发掘出用户感兴趣的物品，以此来帮助用户快速的找到有感兴趣的物品，提升网络应用的流量及用户的黏性。

1.2 国内外研究现状

大数据时代，既给人们的生活带来了很大便利，同时也面临着很多的问题。你可以足不出户便轻松了解到所有的大小事情，同时大量的信息也阻碍我们获取高质量的信息。面对信息膨胀的问题，各个领域的研究者也都已经提出了解决办法，例如分类目录、信息检索和推荐，毫无疑问推荐系统在解决信息过载问题上是非常成功，自从推荐系统被提出之日起，便吸引了广大研究者的注意，而且被广泛的应用在各个领域中，例如电影推荐^{[2][3]}、音乐推荐^{[2][3]}、图书推荐^[2]、广告推荐^[2]、电商推荐^[2]以及学术领域推荐。本文的研究领域则是学术领域相关的推荐，科研社交网站作为一种新型的专业社交网络平台，主要关注的是为科研人员提供在线的以科学研究为导向的活动及构建学者间学术网络^[2]。从 2007 年开始，国内外就出现一些科研社交相关的网站，如国内外的 ResearchGate、Academic 和科研之友和学者网等等，为科研交流有了新的有效途径。

经研究表明，在对文献数据进行挖掘时，论文正文篇幅较长，且包含的冗余信息较多，最能代表论文的内容的是论文的题目、摘要和关键词等部分^[2]，关键词短小精悍，但其包含的信息却非常的丰富，因此本文就采用文本挖掘和自然语言处理相关技术抽取论文摘要中的信息作为关键词，以抽取出的关键词作为论文作者的兴趣。在此基础上提出基于关键词网络的学者推荐模型，这里的关键词即是通过文本特征工程相关技术从文本中抽取出来的能代表作者研究兴趣的一些词或者短语；学者推荐则是属于社交网络范畴，因此本文将从特征提取之关键词抽取和社交网络两个维度去阐述目前的研究现状。

首先对关键词提取进行调研得知，关键词的提取是文本挖掘的一个子领域，而文本挖掘技术又是数据挖掘的一个分支，所以关键词抽取也就属于数据挖掘领域范畴。1995 年 Feldman 提出了文本挖掘概念和框架。国外研究比较早，在特征工程、文本分类等方面都取得丰富的研究成果。国内 1998 年开始才陆续开展文本

挖掘的研究，并且由于中文自身的特点，难度系数也相对大，所以跟国外相比还存在着一定差距^[2]。该技术在数据指数增长的时代扮演者重要的作用，很多文本挖掘工具都已经应用在商业方面^[2]。通过将非结构化的文本转化为结构化数据的形式，让计算机能够计算，从而抽取隐含的、有用的知识。文本特征工程的技术包括预处理、特征提取等，数据的预处理对结果准确率有很大的影响，是一个很重要的环节。该项技术被广泛应用在文本分类、聚类和情感分析等各项研究中。其中文献^[2]就是采用将文本挖掘和深度学习结合对文本进行情感分析。50 年代末期，国外学者.P.Luh 就已经提出了词频统计的思想，用于自动分类，随后众多学者在该领域也取得卓越的成效，最近研究者主要围绕文本的挖掘模型、特征抽取和文本表示^[2]。国内起步较晚，针对中文信息处理还未形成完整的技术理论和框架，不过进展也在逐步加快。

2018 年的 ACM RecSys 中还专门设立了关于序列感知推荐的课程并发布了一篇关于序列感知推荐的研究综述^[2]。

1.2.1 个性化推荐研究现状

1.2.2 序列感知推荐研究现状

1.2.3 迁移学习研究现状

1.3 研究内容和方法

1.3.1 研究内容

本文通过调研分析科研社交网络的发展状况后，分析了目前科研社交网络在国内外理论和实际应用的情况，本文提出一种新的思路去解决学者推荐模型，详细阐述了该学者推荐模型的工作步骤和方式，受文本特征提取之关键词抽取的启发，本文还提出了结果评价的指标。最后本文通过爬取“微软学术”官网的真实数据对提出的推荐模型的有效性进行验证。本文总体分为六个章节，具体每章的内容安排如下所示：

1.3.2 研究方法

本文主要采用的研究方法有如下几点：第一，文献调研法。通过互联网技术访问线上各个数据库中检索了大量研究领域的相关书籍、论文等学术成果，经过对国内外的相关研究文献与资料的全方位收集和分析，确立本文研究方向和主题，设计本文推荐模型框架及各模块之间的耦合。第二，迁移法，本文是建立在自然语言处理、文本挖掘、推荐系统等相关技术的研究基础之上，通过综合探索以上技术理论，将其迁移到本文的模型框架及设计的评价指标上。第三，实验仿真与分析法，本文通过调研和分析大量文献，提出本文的研究内容，为了验证本文提出的模型的有效性，本文在多个真实的数据上进行了实验，从而检验模型的可靠性。

1.4 本文主要贡献

本文通过前期大量文献调研，对比国内外科研社交网站在学者推荐技术上进行的研究，通过深入探索后提出了本文的研究问题，并针对提出的问题进行了大规模的对比实验，直到得出最后的结论。整个过程中本文的主要贡献体现出如下几点：

1. 本文提出了一个基于关键词网络的学者推荐模型，该模型能够根据新用户搜索的关键词进行及时推荐。该模型包含两个网络图，即关键词共现图 (Keywords Co-occurrence Graph)，为了描述方便，本文简称该图为 *KCG*，*KCG* 是根据用户学者检索的关键词和匹配到的文献的关键词共同构建的，其目的有二，第一是在用户学者没有明确的意图的情况下，通过挖掘 *KCG* 中的核心点作为用户学者的搜索意图，第二是通过该图确立用户学者的研究兴趣。另外一个为论文学者 (Author) 和关键词 (Keyword) 构建的二部图 (Graph)，简称为 *AKG*，该图的目的是采用某种算法对论文学者进行打分排序，以关键词为纽带，建立论文学者和用户学者之间的映射关系，从而向用户学者推荐最有可能感兴趣的论文学者。本文设计的推荐模型通过耦合并挖掘 *KCG* 和 *AKG* 两个网络图，最后完成最终的推荐目标。
2. 本文设计多层过滤器从每篇摘要中抽取关键词集，抽取出的关键词即为论

文学者的研究领域或者兴趣。多层过滤器包括采用自然语言处理相关的 Pos-Tag 标注词性、过滤停用词、正则匹配等启发式的算法过滤生成关键词候选集，还包括文本特征提取中的多个经典特征算法对候选集做进一步过滤，最后采用聚合排序算法对候选关键词集进行重排序，从而生成高质量的关键词集。

3. 在学者推荐领域，本文提出了全新的假设，即以关键词网络作为切入点，完成推荐任务。该假设是在缺乏其他指标，如行为记录、基本信息和论文引用数、影响因子等特征，仅仅只使用关键词这一个特征的前提下，设计出了本文的学者推荐模型，该模型能有效解决推荐系统的冷启动问题。为了验证有效性，本文爬取了微软学术官网的文献数据，在真实的数据上验证了模型的可行性。

1.5 论文组织架构

第 2 章 个性化推荐、序列感知推荐及迁移学习

2.1 推荐系统

推荐系统（RS）已发展成为帮助用户做出明智决策和选择的基本工具，尤其是在大数据时代，客户必须从大量产品和服务中做出选择。因此现代推荐系统是在当前大数据环境下应运而生的，现代推荐系统的架构如图3-1

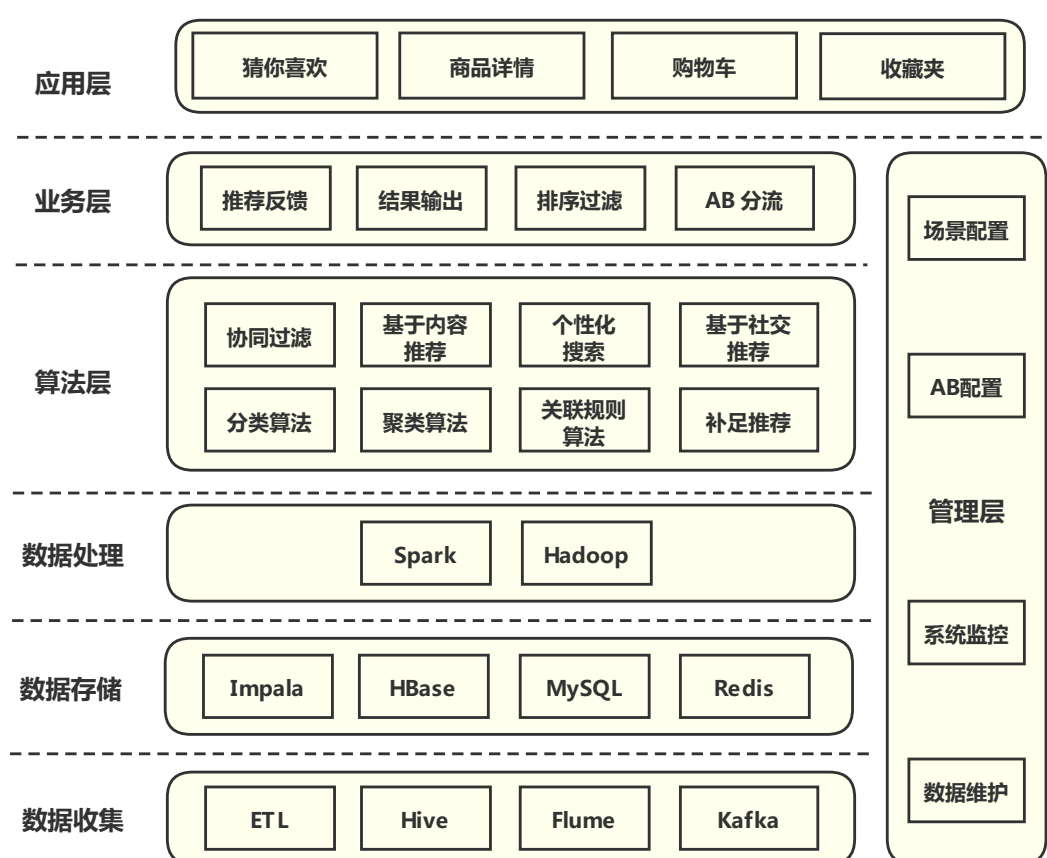


图 2-1 现代推荐系统架构图

2.1.1 个性化推荐概述

个性化推荐能成功，需要具备两个条件，第一是海量的信息，因为只有信息量很大的时候用户才需要系统的自动推荐。第二是用户没有明确的需求，如果有明确的需求，用户都会选择通过浏览器搜索，快速发现感兴趣的东西，而不需要

推荐了。随着互联网行业的快速发展,信息量的增长速度飞快,各种新闻的推送(微博、微信公众号等),占用了很多的时间,严重的影响了获取信息的质量问题,大量的垃圾信息导致人们获取有价值的信息的成本有所增加,并且,人们的生活节奏也日益加快,需要在有限的时间内获取对自己有用的信息便难上加难,为了解决信息过载的难题,研究人员边通过用户历史行为数据开始对用户兴趣进行建模,从而实现个性化推荐的功能,让每个用户都有不一样的个性化页面。个性化推荐系统的价值便在于此。至今,国内外许多大型的公司投入了大量的精力到推荐系统的研究中,因此也给公司带来了很大收益,如最早研究推荐系统的亚马逊。

在学术社交研究领域个性化人物推荐包括社交好友推荐,论文合作作者推荐等。收集用户的历史行为记录,如评分、分享、收藏、自定义标签等,通过分析用户的行为,给其推荐可能感兴趣的对象,以及用户的个人基本信息,如性别、年龄,在推荐系统的研究中,真正难以把握的是用户个性化需求,

2.1.2 推荐算法

2.1.3 推荐系统存在的问题

冷启动问题推荐系统需要数据作为支撑,通常需要根据用户历史行为记录去预测未来可能产生的行为和兴趣。现实中,我们面对大量的新用户又或者对于全新上线的系统不仅没有有效的用户行为数据而且也缺乏用户的个人基本信息(年龄、性别等),我们没有任何数据对用户的偏好进行建模,这个问题被称为“冷启动”,是推荐系统中面临的一个难题问题。在缺乏用户行为数据的情况下,并不是就没办法给用户推荐,早期的推荐系统会基于商品内容数据做推荐,例如给商品打大量的标签或者通过推荐热门商品等方式来解决冷启动问题。Yao^[2]等学者提出通过用户搜索的关键字信息去预测用户的偏好,具体方式是通过比较搜索的关键词和已有的产品的信息做相似性比较,实验证明,他们提出的方法比推荐最受欢迎的方法效果更好。

噪音问题大数据时代面临一个很严重的问题,不仅数据量多,而且数据不干净要获得高质量的数据往往是不太现实的,需要前期花费人力物力进行数据预处理,高效的数据处理能力,能快速的挖掘数据的重要信息。随着需求的提出,研

究者们也提出了很多数据处理的方式，例如减噪、归一化等。

数据长尾问题个性化推荐系统一开始主要解决的问题就是通过发掘长尾数据来提升商家产品的销售额，而长尾商品却仅仅只能代表小部分用户的需求，所以只能充分发掘用户的行为，研究用户的兴趣，以此找到用户的个性化需求。所以，推荐系统可以更好地发掘数据的长尾。

2.2 序列感知模型

序列感知模型是把数据根据时间日期排序之后，按照数据之间的时间先后顺序，发现时间上近邻的数据之间的隐藏关系或数据的周期性变化规律等与时间有关系的一类数据挖掘模型，数据挖掘领域又常称这类模型为时序模型。因此序列感知模型面对的数据必须包含时间戳或者数据的存储形式能够不丢失数据的诞生先后顺序。时序模型数据分析的目的就是为了挖掘出数据之间的内在时间规律，找到这种时间规律之后利用其归纳、类推、演绎未来的数据变化趋势，从而进行建模样本之外的数据预测。

2.2.1 序列感知推荐任务及场景

当待分析的数据具有固有的顺序性质，序列学习方法就会在这些应用领域中 useful，比较常见的应用有如自然语言处理、语音识别、时间序列预测、DNA 建模，以及作为本文工作的核心内容，序列感知推荐。

2.2.2 序列感知推荐技术分类

频繁集挖掘

马尔可夫链

循环神经网络

因为传统前馈深度神经网络 (FNN) 无法了解给定输入的上下文环境关系，循环神经网络 (RNN)[?] 被发明的目的就是用来进行对可变长度的序列数据进行建模。循环神经网络与传统的 FNN 模型之间的主要区别在于组成网络的单元中存在内部隐藏状态，在一个序列建模步骤中的每个内部隐藏状态节点都接收来自上一

个节点的输入，因此这可以用一个循环来表示，其结构如图2-2所示，隐藏状态层保留了过去序列编码的摘要，每当 RNN 呈现新的输入时，就会更新该隐藏层的状态。对于一个最简单的标准循环神经网络其通过以下形式来更新隐藏单元的状态 h ：

$$h_n = f(Ux_n + Wh_{n-1} + b) \quad (2.1)$$

其中 h_{n-1} 是第 $n-1$ 层神经网络的向量化表示， x_n 是传递给第 n 层的输入序列编码， U, W 是该层包含的权重矩阵， b 是该层向量的偏置。函数 $f(\cdot)$ 为非线性转换函数，也称激活函数，常用的激活函数有 Logistic Sigmoid 函数 $\sigma(\cdot)$ ， $\tanh(\cdot)$ ，线性整流单元 $ReLU(\cdot)$ 和一些它们的变体。经过以上隐藏层状态的更新之后，输出层的计算公式如下：

$$\hat{y}_n = g(Vh_n) \quad (2.2)$$

其中 \hat{y}_n 是 n 时刻的输出值， V 是输出层的权重矩阵， $g(\cdot)$ 是输出层的激活函数。

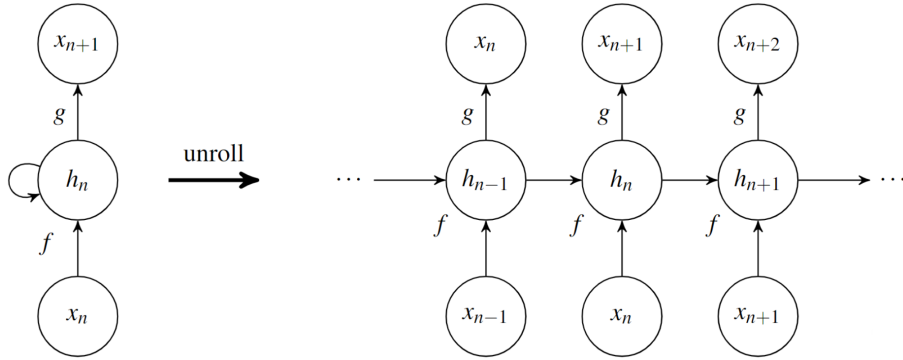


图 2-2 简单循环神经网络结构图

通过公式2.1,2.2可以明白，RNN 的每一个时间步骤都会有一个新的输入，并且特定时间步骤的输出依赖于之前所有步骤的输入，这意味着时间步骤 N 时刻的损失函数的计算要回溯到时间步骤 1，这一过程也称为基于时间的反向传播算法 (*backpropagation through time, BPTT*)^[2]。但是如果处理的序列很长的话，经过多层反向传播，BPTT 会产生梯度消失或者梯度爆炸的问题，以至于无法从差的很远的时间步骤中感知上下文环境，使得 RNN 的训练变得非常麻烦，RNN 这一

明显的缺点也称为长期依赖问题。

长短期记忆神经网络

为了解决 RNN 的长期依赖问题，一些基于公式2.1的变形工作诞生，其中最广为人知的是长短期记忆网络 (long short-term memory, LSTM)^[2]。LSTM 通过精巧设计的记忆单元更换了 RNN 中的隐藏单元，其核心计算单元如图2-3所示。LSTM 神经元内部通过精心设计的分别称为遗忘门、输入门、输出门的三个门结构来决定哪些信息更新到内部或者从内部去除。在遗忘门(2.3)当中，前一时间步的掩藏状态和当前时间步的输入经过 Sigmoid 函数非线性转换之后得到一个 $[0, 1]$ 之间的值，以表示需要遗忘信息的概率。在输入门(2.4),(2.5)中，将当前时间步骤中的输入和前一时间步骤学习到的隐藏状态经过 \tanh 激活函数的计算生成一些候选值，并通过 Sigmoid 函数的传递从候选值中选出一些进行更新。而输出门(2.6)就决定了当前单元要输出哪些部分。LSTM 通过如下的组合函数来更新隐藏单元的状态：

$$f_t = \sigma(U_f x_t + W_f [h_{t-1} + c_{t-1}] + b_f) \quad (2.3)$$

$$i_t = \sigma(U_i x_t + W_i [h_{t-1} + c_{t-1}] + b_i) \quad (2.4)$$

$$c_t = f_t c_{t-1} + i_t \tanh(U_c x_t + W_c h_{t-1} + b_c) \quad (2.5)$$

$$o_t = \sigma(U_o x_t + W_o [h_{t-1} + c_t] + b_o) \quad (2.6)$$

$$h_t = o_t \tanh(c_t) \quad (2.7)$$

其中 $\sigma(\cdot)$ 是 Logistic Sigmoid 激活函数；而 i 、 f 、 o 和 c 分别是输入门、遗忘门、输出门和隐藏层的激活向量；变量 b 表示偏置向量，例如 b_f 表示为遗忘门的偏置向量。 U 、 W 分别代表输入向量的权重和上一层输出的权重，例如在遗忘门中， U_f 代表遗忘门输入向量 x_t 的权重， W_f 代表上一个 LSTM 神经元输出 h_{t-1} 的权重。

门控循环单元 Gated Recurrent Unit

由于 LSTM 在循环神经单元中增加了三个门结构，与 RNN 相比，在一个神经元当中要完成更多的复杂计算。当使用更大的网络的时候，训练时间相比 RNN 也将显著增加。为了减少训练的时间复杂度并同时保留 LSTM 对长期依赖关系

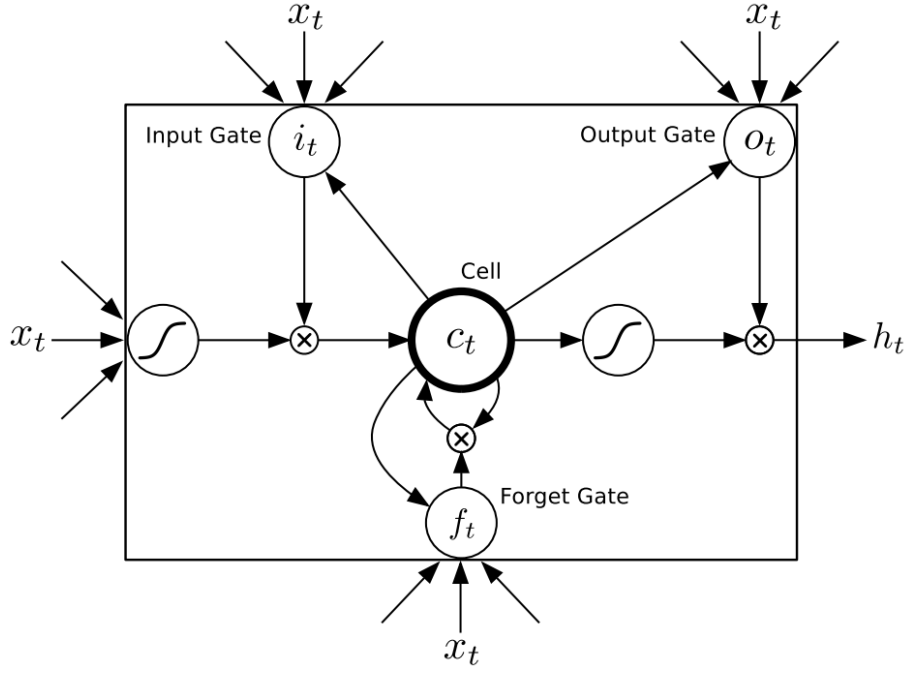


图 2-3 长短期记忆神经网络结构图

的记忆能力，2014 年 Cho 等人提出门控循环单元 (Gated Recurrent Unit, GRU)^[21]。与 LSTM 相似，GRU 使用门结构建模单元内部信息的流动，不同的是，GRU 将 LSTM 三个门减少为两个。GRU 使用更新门来决定是否遗忘上时刻的信息或者记忆此时刻新的外部输入信息，其功能相当于组合了 LSTM 中的输入门与遗忘门。使用重置门来决定如何将新的输入信息与内部已有放入记忆相结合。在 t 时刻 GRU 单元中更新门的状态表达式为：

$$z_t^j = \sigma (U_z \mathbf{x}_t + W_z \mathbf{h}_{t-1})^j$$

其中， \mathbf{x}_t 为第 t 个时间步骤的输入向量， \mathbf{h}_{t-1} 中保存的是上一个时间步骤的信息。 U ， W 是更新门当中输入向量与上一时间步信息的权重矩阵。 t 时刻 GRU 单元中重置门为：

$$r_t^j = \sigma (U_r \mathbf{x}_t + W_r \mathbf{h}_{t-1})^j$$

结合更新门与重置门，整个 GRU 神经元在 t 时间步的内部状态更新为：

$$\tilde{z}_t^j = \sigma (U_z \mathbf{x}_t + U_z \mathbf{h}_{t-1})^j$$

其中 \tilde{h}_t^j 为候选信息，表示当前记忆的内容，其计算表达式为：

$$h_t^j = (1 - z_t^j) h_{t-1}^j + z_t^j \tilde{h}_t^j$$

单个 GRU 神经元的总体结构如图2-4所示。

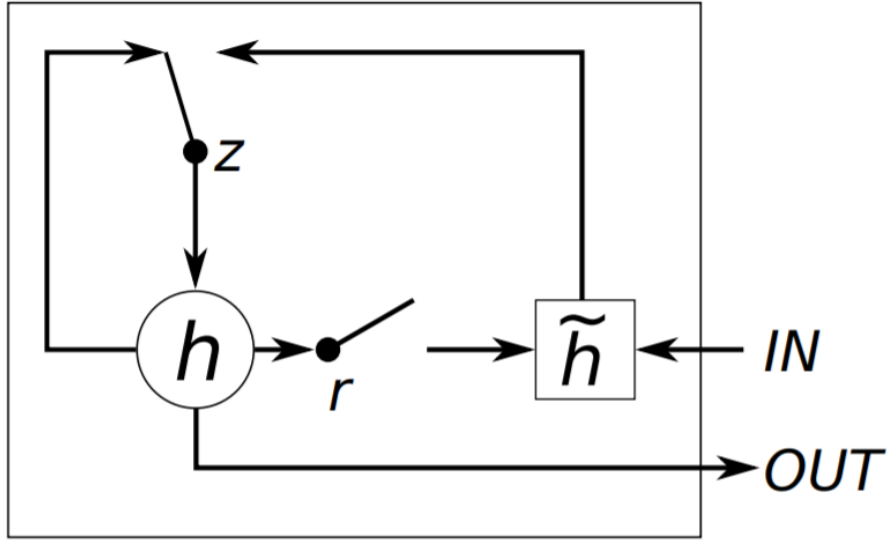


图 2-4 门控循环单元神经元结构图

双向循环神经网络

在自然语言处理的实体识别技术中，双向循环神经网络 (Bidirectional recurrent neural networks, BRNN)^[2] 弥补了单项循环神经网络对于上下文感知能力的不足，因为单向 RNN 预测下一个单词时使用的只是此单次出现之前的信息，而 BRNN 则从两个方向获取信息，上下文感知能力也就更强了。BRNN 将隐藏层分为两个部分，前向状态层 \vec{h} 和反向状态层 \overleftarrow{h} ，其输出层的输入由 \vec{h} 和 \overleftarrow{h} 堆叠而成，其迭代公式如下：

$$\vec{h}_t = \sigma(U_{\vec{h}} x_t + W_{\vec{h}\vec{h}} \vec{h}_{t-1} + b_{\vec{h}}) \quad (2.8)$$

$$\overleftarrow{h}_t = \sigma(U_{\overleftarrow{h}} x_t + W_{\overleftarrow{h}\overleftarrow{h}} \overleftarrow{h}_{t+1} + b_{\overleftarrow{h}}) \quad (2.9)$$

$$y_t = W_{\vec{h}y} \vec{h}_t + W_{\overleftarrow{h}y} \overleftarrow{h}_t + b_y \quad (2.10)$$

BRNN 的隐藏层结构如图2-5所示。

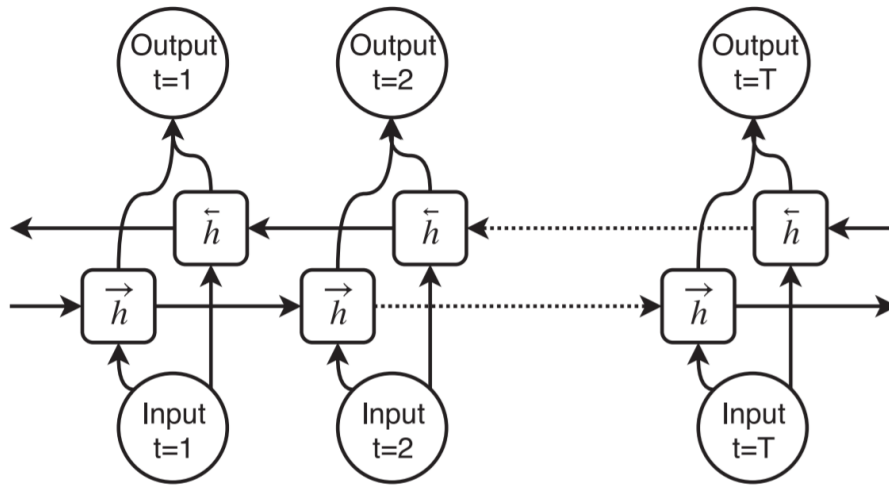


图 2-5 双向循环神经网络结构图

2.2.3 序列感知推荐算法评价指标

2.3 图神经网络

2.3.0.1 迁移学习研究内容

2.3.0.2 迁移学习分类

2.3.0.3 迁移学习方法

2.3.0.4 迁移学习面临的问题

2.4 本章小结

第3章 基于双向长短期记忆网络的序列感知推荐

3.1 问题的提出

推荐系统面临的问题主要有两大类：评分预测和项目推荐。所以在推荐领域根据用户的历史活动记录预测用户下一次行为可能会选择什么项目也是一个重要的问题。在许多在线网站和应用程序当中，如在线电子商务、新闻或视频推荐网站、音乐或广播电台，它们都需要为用户提供一个杰出服务来推荐用户在未来可能会喜欢的东西。现有的推荐系统主要关注于找出用户或项目额近邻集，或者利用隐式或显式信息(如标签、评论、物品内容、用户属性)来提升近邻感知能力。然而，却少有工作利用数据当中的时序属性来直接构建推荐系统。在本篇论文中，我们发现数据的序列中其实包含着许多有价值的且激动人心的信息，以视频网站为例，一个用户看了纪录片《河西走廊》第一集《使者》之后，接下来看的另一个节目很有可能会是《河西走廊》第一集《通道》。甚至早在2011年举办的 Recsys 推荐系统大会上，来自音乐应用 Pandora¹的研究人员给出的演讲上都提到了许多用户听音乐具有时序特点。

在某些特别的应用场景下，常规的推荐系统甚至无法起作用。现有的推荐系统都需要分析用户的数据，因此每个网站和应用的使用到需要让用户完成注册以及登录，然而用户每次使用网站或者应用的服务时都不一定会愿意登录，这种场景下对匿名用户的推荐显然挑战更大，常规的推荐策略显然无法起作用，基于匿名用户本地浏览器和缓存的会话所蕴含的序列进行推荐则显现出很重要的实践意义与价值。

3.2 算法框架描述

为了挖掘许多现有算法忽视掉的用户行为序列特征，本文提出了使用神经网络来对用户的序列进行建模的思路。我们的模型主要由五个部分组成，分别以此是嵌入层、循环结构、全连接层、池化层和输出预测层。其整体框架结构如

¹www.pandora.com

图3-1所示。

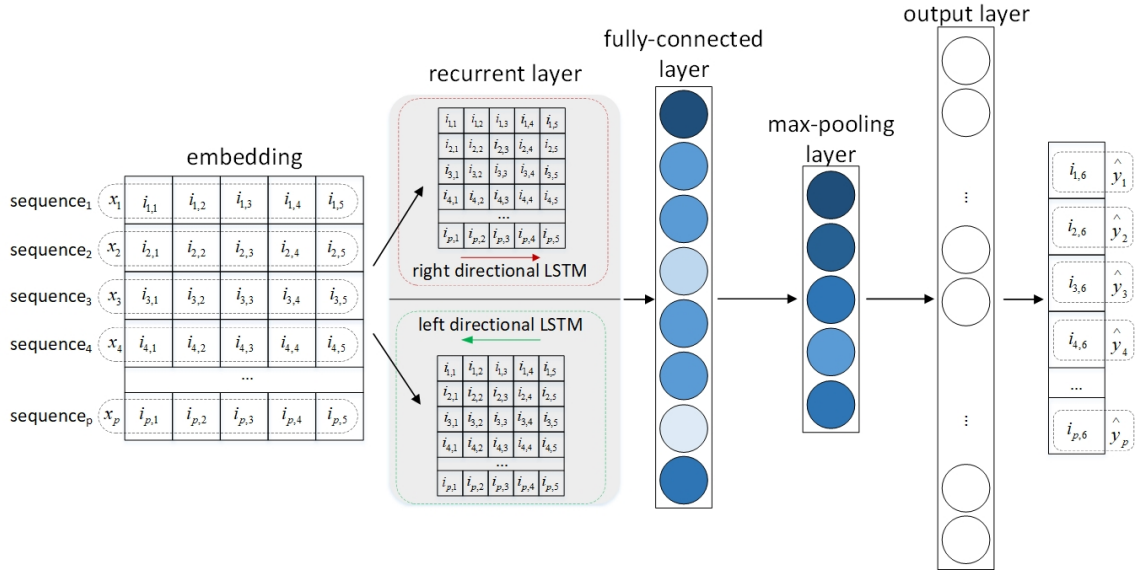


图 3-1 基于双向循环神经网络的序列推荐算法框架图

3.3 基于双向长短期记忆网络的序列感知推荐模型

3.3.1 目标问题定义

序列感知推荐与传统的单类协同过滤推荐是有很大的不同的，序列感知推荐的主要目标是预测用户下一步将会点击什么，而且利用数据仅仅包含用户当前历史行为的序列集合，而不接触为用户设置的长期偏好属性。接下来我们将定义序列感知这一问题的形式。

在序列感知推荐当中，我们定义 $\mathbb{U} = \{u_1, u_2, \dots, u_N\}$ 代表不同的用户集合，定义 $\mathbb{I} = \{i_1, i_2, \dots, i_M\}$ 代表在有序列中出现过的不同物品集合， $s_u^t \in \mathbb{I}$ 表示用户 u 在时刻 t 点击某一个物品的记录，该记录对应的物品包含在物品集合 \mathbb{I} 当中。对于每一个用户 u ，都记录一个按照数据诞生时间戳顺序排列的用户点击记录序列 $\mathbb{S}_u = \{s_u^1, s_u^2, \dots, s_u^{t-1}, s_u^t\}$ 序列感知推荐的目标是预测下一次点击行为，也就是做出 $t+1$ 时刻的推荐 s_u^{t+1} 。在序列感知推荐模型当中，对于序列 s ，模型的输出是所有候选物品对象可能被点击的概率 \hat{y} ，而概率最大的 K 个输出所对应的候选物品将作为推荐项目给到用户。

3.3.2 嵌入层

我们使用用户消费历史的最近几个序列当做特征，用户消费的最后一个物品当做标签，来构建一个超多分类的有监督学习模型。因此，在特征工程阶段，我们需要将原始序列特征数据转换为计算机容易处理的向量形式并且与标签映射。One_hot 编码是用来表达离散特征的最常用的向量表达形式，然而 One_hot 编码的向量会遇到高维和稀疏的问题。如果我们使用 One_hot 编码来处理具有 1000 个类别的特征，那么每个特征会被一个拥有 1000 个数字的向量来表示，但其中 999 个数字会是 0。在一个大规模数据集中，就计算效率而言这种方式是不可取的。而词嵌入技术在自然语言处理领域大放异彩，我们可以借用词嵌入技术，也构造一个嵌入矩阵来得到比 One_hot 形式小得多的向量结构：

$$e(I_i) = EI_i$$

其中 $E \in \mathbb{R}^{|e| \times |M|}$, $|e|$ 是嵌入层的大小， $|M|$ 是训练集中不同项目的数量。所以 $e(I_i)$ 是 I_i 的嵌入表达，其是一个具有 e 个实数的稠密矩阵。与一个 $|M| \times |M|$ 大小的 One_hot 编码形式相比，我们的序列嵌入矩阵的大小为 $|e| \times |M|$ ，当处理大数据集时，这显著减小了内存消耗。

3.3.3 用户短期兴趣学习

我们通过结合用户 u 在 t 时刻消费的项目 I_u^t 和其先后的项目来表达用户在此时刻的兴趣。行为序列帮助我们更精确地揭示了用户的短期兴趣。在这个推荐系统中，我们使用了一个双向的长短期记忆神经网络构建的循环结构来捕捉用户短期的兴趣变化。

我们定义 $h_b(I_i)$ 为用户他在消费物品 I_i 之前的兴趣， $h_a(I_i)$ 为用户消费物品 I_i 之后的兴趣。 $h_b(I_i)$ 和 $h_a(I_i)$ 都是具有 $|h|$ 个实数的稠密向量。 $W^{(b)}$ 是隐藏层的权重矩阵，用来继承用户之前的兴趣状态，矩阵 $W^{(cb)}$ 用来结合前一个物品的嵌入表达， σ 是一个非线性激活函数，因此通过学习表达式 $h_b(I_i)$ 来学习用户消费物品 I_i 之前的兴趣。同理，用户消费物品 I_i 之后的兴趣通过学习表达式 $h_a(I_i)$ 来学习。所有

用户的初始兴趣使用同样的参数 $h_b(I_1)$ ，用户消费历史中的最后的兴趣则共享参数 $h_a(I_n)$

$$h_b(I_i) = \sigma(W^{(b)}h_b(I_{i-1}) + W^{(cb)}e(I_{i-1}))$$

$$h_a(I_i) = \sigma(W^{(a)}h_a(I_{i+1}) + W^{(ca)}e(I_{i+1}))$$

通过以上公式，学习用户某个时刻之前和之后的兴趣，将它们和用户当前消费物品的嵌入矩阵集合来表达此时刻用户的临时兴趣状态，其结合形式如下：

$$x_i = [h_b(I_i); e(I_i); h_a(I_i)]$$

所以通过使用大量用户的历史行为序列 $\{i_1, i_2, \dots, i_{n-1}, i_n\}$ ，如果我们的模型学习到了某个用户消费物品 i_{n-1} 时的临时兴趣 x_{n-1} ，他将更有可能得到一个物品推荐 i_n 。双向循环结构能够对序列的前向扫描中捕获所有的 h_b ，反向扫描则捕获了所有的 h_a 。当训练集中所有的临时兴趣状态 x_i 都被捕获之后，运用一个线性转换与 \tanh 激活函数将结果送到下一层。

$$y_i^{(2)} = \tanh(W^{(2)}x_i + b^{(2)})$$

$y_i^{(2)}$ 是潜在的兴趣向量，其中的每一个兴趣向量将通过上面权重和参数的更新来决定影响用户消费序列中最重要的因素。

3.3.4 流行趋势学习

当用户消费项目的所有序列被计算完之后，接下来应用一个最大池化层：

$$y^{(3)} = \max_{i=1}^n y_i^{(2)}$$

最大池化通过应用一个最大过滤器到上层代表的非重叠子区域，有了池化层，模型参数或权重迅速减小了，这样也能减小上层输入的空间维度，减小计算消耗。

通过对全局序列属性的捕获最大输出层能在用户的这个历史记录里找到那些最流行的序列组合。模型的最后一层就是常规的输出层了：

$$y^{(4)} = W^{(4)}y^{(3)} + b^{(4)}$$

输出层通过应用一个 softmax 激活函数到 $y^{(4)}$ 来转换成下一个类别的输出概率：

$$p_i = \frac{e^{y_i^{(4)}}}{\sum_{k=1}^n e^{y_k^{(4)}}}$$

3.3.5 模型的训练

将模型训练过程中所有需要更新的参数定义为 θ 。

$$\theta = \{E, b^{(2)}, b^{(4)}, h_b(B_1), h_a(B_n), W^{(2)}, W^{(4)}, W^{(b)}, W^{(a)}, W^{(cb)}, W^{(ca)}\} \quad (3.1)$$

模型训练的优化目标是最小化交叉熵损失函数：

$$\mathcal{L}(y, S, \theta) = - \sum_{u \in \mathbb{U}} [y_u \log p(y_u | S_u, \theta) + (1 - y_u) \log(1 - p(y_u | S_u, \theta))] \quad (3.2)$$

3.4 复杂度分析

在嵌入层当中，主要包含一个矩阵的向量相乘操作，这一部分的时间复杂度是 $O(n)$ 。在双向长短期记忆网络中，网络的主体是 LSTM，其时间复杂度为 $O(n \cdot d^2)$ ， n 表示序列的长度， d 表示嵌入层因子的维度。而双向 LSTM 的时间复杂度我们可以知道为 $2 \cdot O(n \cdot d^2)$ ，最大池化层的时间复杂度为 $O(n)$ ，全连接层的时间复杂度为 $O(n)$ ，整个模型是上述这些结构的串联，所以整个基于双向循环神经网络的序列推荐算法的时间复杂度为 $O(n \cdot d^2)$ 。相比于基于用户的协同过滤算法 (UBCF)^[2, 1] 的时间复杂度 $O(n_u^2)$ 和基于物品的协同过滤 (IBCF)^[2, 1] 时间复杂度 $O(n_i^2)$ ，其中 n_u 表示用户数量， n_i 表示物品数量。与其相比，序列长度 n 和嵌入维度 d 存在这样的关系： $n_u > n_i \gg d > n$ ，所以基于双向循环神经网络的序列推荐算法在选择合适的序列长度和嵌入维度大小时面对大数据处理的压力时还是可以接

受的。

3.4.1 算法实现

算法 3.1 基于双向长短期记忆网络的序列感知推荐算法

已知：用户序列数据： S ；序列长度： $maxLen$ ；最大物品数： $maxNum$

求：推荐结果： $Items$

```

1:  $sequence \leftarrow S$ , 其中  $sequence = \{seq_1, seq_2 \dots seq_n\}$ 
2: if  $len(sequence) \geq maxLen$  then
3:     选取序列最后的  $N$  物品:  $sequence_{lastN} \subseteq sequence$ 
4:      $items \leftarrow sequence_{topN}$ , 其中  $items = \{word_1, word_2 \dots word_m\}$ 
5:     for each  $i \in [1, maxLen]$  do
6:         将每个序列的物品 ID 编码到嵌入矩阵:  $E = Embedding_{seq_i}$ 
7:     end for
8: else
9:     continue
10: end if
11: for each  $i \in [1, maxLen]$  do
12:      $output_{right} = LSTM(E)$ 
13: end for
14: for each  $i \in [maxLen, 1]$  do
15:      $output_{left} = LSTM(E)$ 
16: end for
17:  $output = [output_{right}; output_{left}]$ 
18:  $output = maxpolling(output)$ 
19:  $NextItem = SoftMax(output)$ 
20: return  $NextItem$ 

```

3.5 本章小结

本章节主要介绍本文提出的一个针对用户下一个点击推荐的新颖神经网络推荐模型 BiLSTM4Rec，其利用双向长短期记忆网络来对用户按照时间先后点击物品的历史记录进行序列建模，挖掘隐藏在序列当中的用户短期兴趣变化，来精确地对用户下一个可能感兴趣的物品做出推荐。通过对大规模物品 ID 进行矩阵嵌入压缩处理，借助该时序推荐算法，能够取得比传统协同过滤更高效更精确地短期推荐策略。

第 4 章 基于自注意力机制的序列感知推荐

4.1 问题的提出

在多个不同的机器学习领域，例如图像描述、机器翻译、阅读理解、摘要生成还有一些其他的应用上，已经证明注意力机制是有明显效果的这一机制背后的重要思想是影响序列输出的因素只与序列中某些重要信息相关，而无需过多关注其他无用的信息。这一机制的命名受到人类视觉系统中视觉焦点的启示，所以称为注意力机制。

2017 年 NIPS 大会上谷歌发表了一个完全用注意力机制构建的机器翻译框架 *Transfomer*^[2]，到达了当时最好的效果。

在基于循环神经网络的序列感知推荐系统方面，由于循环神经网络本身的特性也带来了性能上面的一些缺陷。由于循环神经网络的结构，当前神经元的输入有上一个神经元的输出，反向传播梯度更新时，下一个神经元的梯度更新也需要上一个神经元计算完再开始，这一串行的性质限制了循环神经网络的训练速度，特别是当需要使用更深的循环神经网络的时候。而由于注意力机制完全抛弃了循环神经网络的串行结构而

使用注意力机制的另一好处是基于注意力的方法在推荐可解释性方面也更好说明。

因此在这一部分，我们选用自注意力机制来建模用户短期行为模式的依赖因素，

4.1.1 目标问题定义

与第三章中基于双向长短期记忆网络的序列感知推荐模型问题一样，我们定义 $\mathbb{U} = \{u_1, u_2, \dots, u_N\}$ 代表不同的用户集合，定义 $\mathbb{I} = \{i_1, i_2, \dots, i_M\}$ 代表在所有序列中出现过的不同物品集合， $s_u^t \in \mathbb{I}$ 表示用户 u 在时间刻 t 点击某一个物品的记录，该记录对应的物品包含在物品集合 \mathbb{I} 当中。对于每一个用户 u ，都记录一个按照数据诞生时间戳顺序排列的用户点击记录序列 $\mathbb{S}_u = \{s_u^1, s_u^2, \dots, s_u^{t-1}, s_u^t\}$ 序列感知推荐

的目标是预测下一次点击行为，也就是做出 $t + 1$ 时刻的推荐 s_u^{t+1} 。在序列感知推荐模型当中，对于序列 s ，模型的输出是所有候选物品对象可能被点击的概率 \hat{y} ，而概率最大的 K 个输出所对应的候选物品将作为推荐项目给到用户。

4.1.2 嵌入层

我们使用用户消费历史的最近几个序列当做特征，用户消费的最后一个物品当做标签，来构建一个超多分类的有监督学习模型。因此，在特征工程阶段，我们需要将原始序列特征数据转换为计算机容易处理的向量形式并且与标签映射。One_hot 编码是用来表达离散特征的最常用的向量表达形式，然而 One_hot 编码的向量会遇到高维和稀疏的问题。如果我们使用 One_hot 编码来处理具有 1000 个类别的特征，那么每个特征会被一个拥有 1000 个数字的向量来表示，但其中 999 个数字会是 0。在一个大规模数据集中，就计算效率而言这种方式是不可取的。而词嵌入技术在自然语言处理领域大放异彩，我们可以借用词嵌入技术，也构造一个嵌入矩阵来得到比 One_hot 形式小得多的向量结构：

$$e(I_i) = EI_i$$

其中 $E \in \mathbb{R}^{|e| \times |M|}$ ， $|e|$ 是嵌入层的大小， $|M|$ 是训练集中不同项目的数量。所以 $e(I_i)$ 是 I_i 的嵌入表达，其是一个具有 e 个实数的稠密矩阵。与一个 $|M| \times |M|$ 大小的 One_hot 编码形式相比，我们的序列嵌入矩阵的大小为 $|e| \times |M|$ ，当处理大数据集时，这显著减小了内存消耗。

位置信息嵌入层

由于在利用自注意力方法构建的模型当中，没有了循环和卷积结构，因此抛弃了项目之间的序列信息，但用户的短期兴趣关注点隐藏在序列之中，为了使我们的模型能够捕捉到序列变化信息，必须将序列中物品的相对或者绝对位置信息加入到模型当中去。位置编码有可学习形式 (learned) 和固定参数形式 (fixed)^[2]。这里我们使用了与 Transfer^[2] 一样的方式，在模型输入的嵌入层中加入物品序列的固定参数形式“位置编码”，即位置信息嵌入层。我们使用正弦和余弦函数并结合不同频率将物品的固定位置编码传递给模型，位置编码的一个维度对应于正

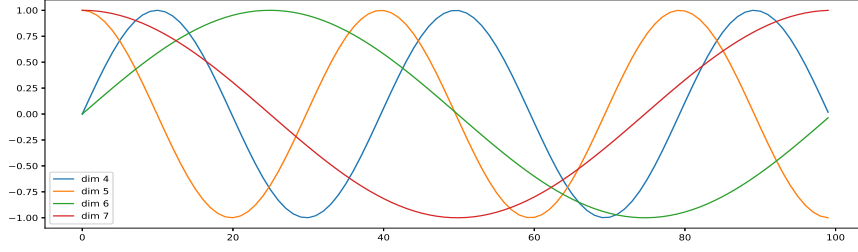


图 4-1 使用不同维度大小的位置编码波形图

弦曲线上的一个值，从而形成从 2π 到 $10000 \cdot 2\pi$ 的几何级数波长，这些信号作为额外的信息加入到输入（或输出）中以表达时间的流逝，使模型能够感知到当前正在处理的是输入（或输出）序列的哪个部分。：

$$PE(t, 2e) = \sin(t/10000^{2e/d_{model}}) \quad (4.1)$$

$$PE(t, 2e + 1) = \cos(t/10000^{2e/d_{model}}) \quad (4.2)$$

其中 t 就是位置信息，表示该物品出现在序列中的第 t 个时间步骤， e 是位置嵌入层的维度。图??显示，不同的位置嵌入层维度大小，位置编码波形图具有不同的波长。由于位置信息嵌入层与嵌入层有相同的维度 $|e|$ ，因此可以把两者相加起来构成我们最终的输入嵌入层 $\widehat{\mathbf{E}}$ ：

$$\widehat{\mathbf{E}} = \begin{bmatrix} \mathbf{E}_{s_1} + \mathbf{PE}_1 \\ \mathbf{E}_{s_2} + \mathbf{PE}_2 \\ \dots \\ \mathbf{E}_{s_n} + \mathbf{PE}_n \end{bmatrix} \quad (4.3)$$

4.2 自注意力模块

自注意力方法是一种特殊的注意力机制，已有在各种任务上的成功应用。与最简单的注意力方法通过对整个语境学习得到有限的知识表示不同，自注意力方法即使面对相隔距离较远的两个元素，其依然能保存上下文中的序列信息并且捕获这两个元素在序列上的关系。因此我们这里使用自注意力方法来捕获用户历史

行为中的序列特征。

构建自注意力模块的基本单元由缩放点乘注意力 *Scaled Dot-product Attention* 构成，一个自注意力模块的输入由查询 (*query*)，键 (*key*) 和值 (*value*) 三部分组成。自注意力模块的输出是模块中值 (*value*) 的加权和得来，而这里的权重矩阵则有查询 (*query*) 和其对应的键 (*key*) 决定。

我们用 Q 表示查询 (*query*) 向量，用 K 表示键 (*key*) 向量，用 V 表示值 (*value*) 向量，向量中的每一行代表一个物品。首先，我们通过使用共享参数的非线性变换将查询和键投影到同一空间：

$$Q' = \text{ReLU}(W_Q \hat{\mathbf{E}}) \quad (4.4)$$

$$K' = \text{ReLU}(W_K \hat{\mathbf{E}}) \quad (4.5)$$

其中 W_Q , W_K 分别是查询和键向量的权重矩阵，得到 Q' 和 K' 之后我们就可以通过如下形式得到值 (*value*) 的权重矩阵：

$$V' = \text{softmax}\left(\frac{Q' K'^T}{\sqrt{d}}\right) \quad (4.6)$$

其中 \sqrt{d} 为缩放因子，为了避免点乘之后出现过大的值导致 *softmax* 的输出被推到梯度消失的位置，因此加上缩放因子来优化矩阵点乘。于是 *Self-Attention* 模块的输出就可以计算出来了：

$$\mathbf{S} = \text{SA}(\hat{\mathbf{E}}) = \text{Attention}(W_Q \hat{\mathbf{E}}, W_K \hat{\mathbf{E}}, W_V \hat{\mathbf{E}}) = V' \hat{\mathbf{E}} \quad (4.7)$$

多头注意力机制 **Multi-Head Attention**

与其只让单个 *Self-Attention* 模块来学习 d 维的查询 (*query*)，键 (*key*) 和值 (*value*)，可以通过集成学习的方法，让多个 *Self-Attention* 模块分别学习不同位置子空间的表达，由于随机初始化的不同，每个 *Self-Attention* 模块的关注点将会有所差异，所以每个 *Self-Attention* 模块可以分别学习到三个不同的 d_k 维的查询 (*query*)、 d_k 维的键 (*key*) 和 d_v 维的值 (*value*) 向量。每个 *Self-Attention* 模块有了查询 (*query*)、键 (*key*) 和值 (*value*) 之后，就能得到 d_v 维的输出。而 *Multi-Head Attention*

是多个 Scaled Dot-product Attention 的并行叠加，因此 Multi-Head Attention 的计算复杂度与 Scaled Dot-product Attention 相当，但是却获得了更好的泛化性能。因此这部分 Multi-Head Attention 的计算方式如下：

$$\text{MultiHead}(\widehat{\mathbf{E}}) = \text{Concat}[\text{SA}_1(\widehat{\mathbf{E}}), \dots, \text{SA}_h(\widehat{\mathbf{E}})]\mathbf{W}^O \quad (4.8)$$

其中 h 表示我们将 h 个 Self-Attention 模块并联堆叠在一起得到 Multi-Head Attention 模块。

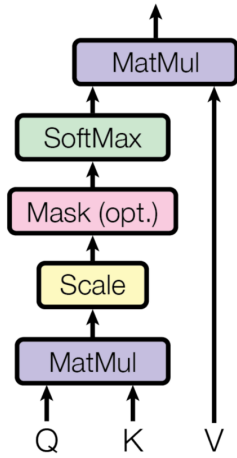


图 4-2 Scaled_Dot-Product_Attention 缩放点乘注意力结构图

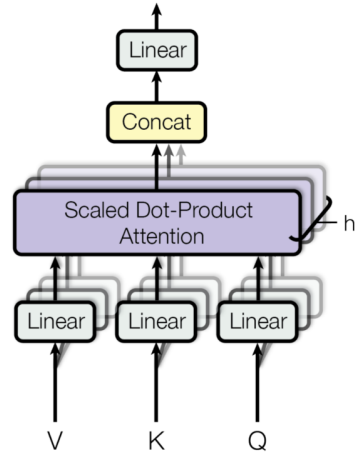


图 4-3 Multi-Head_Attention 多头注意力模块结构图

结构中逐项的 feed-forward 网络作用

考虑到 Self-Attention 模块在学习物品嵌入向量的过程中仍然是一个广义线性模型，为了增强整个 Transformer 模型的非线性拟合能力，Multi-Head Attention 子层后面跟了一个前馈神经网络 FFN 层，为了保证模型计算的效率，它仅由两个线性变换组成，中间嵌入一个 Relu 激活函数来提升非线性表达能力，因此这一部分的形式如下：

$$\mathbf{F}_i = \text{FFN}(\mathbf{S}_i) = \text{ReLU}(\mathbf{S}_i \mathbf{W}^{(1)} + \mathbf{b}^{(1)}) \mathbf{W}^{(2)} + \mathbf{b}^{(2)} \quad (4.9)$$

其中 \mathbf{S}_i 表示嵌入层经过第 i 个 Self-Attention 模块计算后的输出

Residual Connections 残差连接

在序列建模的问题当中，用户最后一个点击浏览的物品通常会与下一个物品

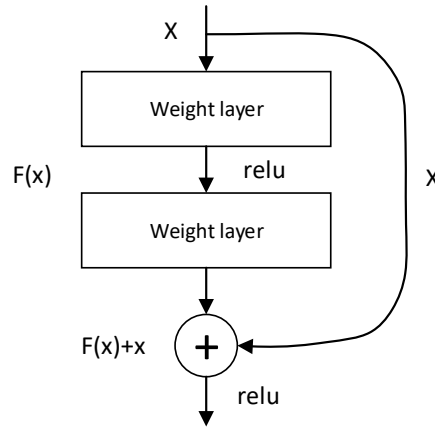


图 4-4 残差连接结构图

的联系较大，所以最后一个物品的特征权重应该会更加大。由于 **Self-Attention** 模块在处理最后一个物品的嵌入向量的时候会将其与之前所有的其他物品嵌入向量做交互，因此最后一个物品的特征并没有特别关注。为了提升对这种浅层特征的提取能力，可以为模型增加残差连接操作，其主要思想是如果浅层特征对于目标学习有效的话，就直接将低层网络学习的浅层特征直接传递给输出层，而不用再经过深层网络提取复杂特征，残差连接的做法就是将浅层网络的输出建立旁路与输出层相连，跳过了部分中间复杂计算。

层次归一化 **Layer normalization**

为了使得模型训练的时候更加容易收敛，减少训练时间，在深度神经网络的模型中加入归一化操作是一个比较常见的做法。在训练卷积神经网络时候，常见的操作是加入批量归一化 (**Batch Normalization**)^[7]，批量归一化通过计算均值和方差将一个批量的输入数据装换到 $[0, 1]$ 之间，加速梯度下降，但其高度依赖批量的大小，同时也不适用于输入序列长度不固定的循环神经网络等模型。层次归一化 (**Layer Normalization**)^[7] 则不是在样本层面，而是将该层神经元的输入通过计算均值和方差转换到在 $[0, 1]$ 之间，这样的操作是跨特征计算的，与其他样本无关，因此可以使用任意大小的批量大小。

$$LN(x_i) = \alpha \times \frac{x_i - u_B}{\sqrt{\sigma_B^2 + \epsilon}} + \beta \quad (4.10)$$

Dropout

使用神经网络时，随着模型变得更大和更深，也带了过拟合的潜在风险，对于神经网络来说，除了在损失函数处加上 L1 和 L2 正则项之外，在模型的结构当中加入 Dropout 也是防止过拟合的常见操作之一，为了防止我们的模型过拟合、同时为了让训练更稳定，减少部分模型参数加速训练过程，在 Self-Attention 模块的输出之后加上 Dropout 操作：

$$\hat{y} = x + \text{Dropout}(\text{SA}(\text{LN}(x))) \quad (4.11)$$

Network Training

4.3 复杂度分析

模型的时间复杂度只要由 Self-Attention 模块主导，Self-Attention 的时间复杂度为 $O(n^2d)$ ，其中 n 为序列的长度， d 为嵌入层维度，而 Multi-Head Attention 由多个 Self-Attention 并行叠加，假设有 h 个 Self-Attention，则 Multi-Head Attention 的时间复杂度仍为 $O(n^2d)$ ，

4.3.1 算法实现

4.4 本章小结

算法 4.1 基于自注意力机制的序列感知推荐算法

已知: 用户序列数据: S ; 最大序列长度: $maxLen$; 最大字数: $maxNum$

求: 推荐结果: $Items$

```
1:  $sequences \leftarrow S$ , 其中  $sequences = \{sen_1, sen_2 \dots sen_n\}$ 
2: if  $n > maxNum$  then
3:   for  $i = 0$  to  $n$  do
4:     计算文本中每个语句的权重:  $score_{sen_i}$ 
5:   end for
6:   根据  $score$  对语句有大到小排序
7:   选取 TOP N 语句:  $sentences_{topn} \subseteq sequences$ 
8:    $words \leftarrow sentences_{topn}$ , 其中  $words = \{word_1, word_2 \dots word_m\}$ 
9: else
10:   $words \leftarrow sentence$ , 其中  $words = \{word_1, word_2 \dots word_m\}$ 
11: end if
12: if  $m > maxLen$  then
13:    $Tokens \subseteq words$ 
14: else
15:    $Tokens = words$ 
16: end if
17: return  $Tokens$ 
```

第 5 章 实验结果与分析

5.1 数据集介绍

为了比较上文提出的两个算法模型，分别从推荐结果的准确性、和算法训练与结果查询返回效率两个方面分别进行了两组实验。因此需要设计两组实验分别从算法推荐准确性和性能两个方面进行验证。本文评估上述两个算法在两个来自真实应用上的数据集，这些数据集都包涵用户与物品产生交互的时间戳：

MovieLens: MovieLens¹是用来对推荐模型进行评估的最流行的基准数据集，它是由 GroupLens 研究组织从 MovieLens 网站收集的关于用户对电影评分的数据集，其中主要的评分文件以“用户 ID | 电影 ID | 评分 | 时间戳”为一行的格式保存了某个用户在某时刻对某个电影做出的评分标准，按照数据量的大小不同，本文选用的 MovieLens 数据集有 Movielens 100K 和 Movielens 1M。

Yoochoose: Yoochoose²数据集是 2015 年推荐系统顶级会议 RecSys 举办的挑战赛 RecSys Challenge 2015 上公开使用的目标数据集。Yoochoose 包含了一个在线电子商务网站在 6 个月时间内用户所有的点击会话流。

5.2 评价指标

作为一个推荐模型，模型会对查询目标给出点击可能性最高的 N 个物品，对于本文提出的序列推荐形式，模型的输入是 t 时刻及其以前的用户点击物品序列，需要的模型输出目标是 $t + 1$ 时刻用户可能点击的物品，以 \hat{I}_u^{t+1} 表示，所以本文采用了 $Precision@N$, $Recall@N$ 指标来评估我们的模型，其计算形式如下：

$$Precision@N = \frac{\sum_u |\hat{I}_u^{t+1} \cap I_u^{t+1}|}{|\mathbb{U}| * N} \quad (5.1)$$

¹<https://grouplens.org/datasets/movielens/>

²<http://2015.recsyschallenge.com>

$$Recall@N = \frac{\sum_u |\hat{I}_u^{t+1} \cap I_u^{t+1}|}{|\sum_u I_u^{t+1}|} \quad (5.2)$$

由于 $Precision@N$ 和 $Recall@N$ 值可以通过阈值调整来相互权衡，为了得到一个更加容易量化评估的指标，本文引入了 $F1@N$ 指标，如果 $F1@N$ 分数越大，可以认为模型的效果更好。而当一次为用户推荐多个结果时，推荐结果在屏幕上展示的位置也会影响物品被点击的概率，越靠前的物品越可能被点击，而引入 $NDGG@N$ 指标的原因就是为了感知这种位置的影响，让点击率越高的物品位置越靠前， $NDGG@N$ 值越大，得到的相关推荐结果中前 N 个物品的排序越准确，所以 $F1@N$ 和 $NDGG@N$ 的计算形式如下：

$$F1@N = \frac{2 \times Precision@N \times Recall@N}{Precision@N + Recall@N} \quad (5.3)$$

$$NDGG@N = \frac{\sum_{i=1}^N \frac{2^{rel_i} - 1}{\log_2(i+1)}}{\sum_{i=1}^{|REL|} \frac{2^{rel_i} - 1}{\log_2(i+1)}} \quad (5.4)$$

5.3 实验结果与分析

5.3.1 关键词抽取结果

5.3.2 学者推荐结果

5.3.3 结果分析

5.4 本章小结

第 6 章 总结与展望

数据集不充分，数据获取不全，可获取数据有限，使得模型应用中部分特征维度得不到数据支持

在未来笔者将对该研究课题进行更为深入的研究与探索第一，进一步完善推荐模型中的组成模块，挖掘更多可以表达学者需求的内容特征；第二，笔者将尝试采用更为充分的数据集与数据量，对模型的实现进行更精确得仿真模拟，并且采用更有效的数据清晰、预处理方法，减少因数据、参数等降低推荐准确度的影响。第三，采用深度学习相关技术提取文本特征

符号列表

RNN	循环神经网络 (Recurrent Neural Network)
LSTM	长短期记忆神经网络 (Long Short-Term Memory)
ANN	人工神经网络 (Artificial Neural Network)
FNN	前馈神经网络 (Feedforward Neural Network)
MLP	多层感知机 (Multi-Layer Perceptron)
BRNN	双向循环神经网络 (Bi-directional Recurrent Neural Network)
GRU	门控循环单元 (Gated Recurrent Unit)

表 目 录

图 目 录

图 2-1	现代推荐系统架构图	7
图 2-2	简单循环神经网络结构图	10
图 2-3	长短期记忆网络神经元结构图	12
图 2-4	门控循环单元神经元结构图	13
图 2-5	双向循环神经网络结构图	14
图 3-1	基于双向循环神经网络的序列推荐算法框架图	16
图 4-1	使用不同维度大小的位置编码波形图	23
图 4-2	Scaled_Dot-Product_Attention 缩放点乘注意力结构图	25
图 4-3	Multi-Head_Attention 多头注意力模块结构图	25
图 4-4	残差连接结构图	26

致 谢

光阴似箭，日月如梭，三年的时间，我的硕士生涯已接近尾声，回头想想这段短暂的求学路，时而喜悦，时而惆怅，感谢命运的安排，让我有幸结识了许多良师益友。首先感谢我的恩师朱佳教授，感谢您一直以来对我的照顾，让我有幸遇到您这样对学生亦师亦友的好导师，您和蔼可亲、学识渊博、没有架子以及乐观向上的工作生活态度深深感染着我。感谢您耐心指导我的论文工作，感谢您陪我走过人生最重要的结婚典礼，感谢您带领我们参加各种会议开阔视野，感谢您带我们吃各种美食，感谢您跟我们聊各种八卦新闻。。。再次向您致以衷心的感谢和崇高的敬意。

感谢学者网团队的各位小伙伴，肖丹阳、林雪琴、韦经敏，感谢你们的3年陪伴和对我的学术帮助，感谢332实验室的师弟师妹们，杨芬、汪序明、郑泽涛、伦家琪、胡迎彬、于晗宇，感谢你们带给我的欢乐和帮助，感谢已经毕业的师兄师姐，董浩业、丁蕊、陈凌潇、许传华、孔剑龙，跟你们在一起真的很开心快乐，感谢生活学习中有你们。

感谢我的最佳室友们男神玉、波波、嘉良，跟你们在一起的日子真的很幸福，记得大半夜去外面吃烧烤喝啤酒，记得去自助KTV嗨翻全场，记得在宿舍吃着零食聊学术八卦到深夜，这些美好的时光将是我一生宝贵的财富，谢谢你们的陪伴，让我的研究生生活丰富多彩。

感谢我的父母、岳父母、爱人、女儿及所有的亲朋好友，你们是我永远的支持者，让我在学习和生活中都受到你们的无私关怀和帮助，才能完成研究生求学生涯，感谢你们的付出，让我开心的度过每一天。

感谢所有的2016级的同窗们，谢谢你们陪我一起成长，祝各位同窗前程似锦；感谢我的母校华南师范大学，感谢您为我们提供美好的校园环境和学习氛围，祝您的办学越来越好；最后感谢各位答辩评委老师，谢谢你们为我指点出不足，祝各位老师永远健康快乐。

作者攻读学位期间发表的学术论文目录

发表的学术论文

- [1] Hailin F, Jianguo L, Jiemin C, et al. Sequence-based Recommendation with Bidirectional LSTM Network[C]. PCM 2018: 428–438.(第一作者)
- [2] Jiemin C, Jianguo L, Jing X, et al. A Hybrid Collaborative Filtering Model: RSVD Meets Weighted-Network Based Inference[J]. 網際網路技術學刊 2016: 1221-1233.(合作作者)