摘要

推荐系统在日常的网络应用中无处不在，比如网上购物、网上买书、新闻app、社交网络、音乐网站、电影网站等等等等，有人的地方就有推荐。根据个人的喜好，相同喜好人群的习惯等信息进行个性化的内容推荐。比如打开新闻类的 app，因为有了个性化的内容，每个人看到的新闻首页都是不一样的。这当然是很有用的，在信息爆炸的今天，获取信息的途径和方式多种多样，人们花费时间最多的不再是去哪获取信息，而是要在众多的信息中寻找自己感兴趣的，这就是信息超载问题。为了解决这个问题，推荐系统应运而生。

协同过滤技术是推荐系统最为核心的技术之一，也是目前应用最为广泛和成功的技术。协同过滤算法可以分为基于全局的算法和基于模型的算法两种，其中基于全局的算法主要依赖最近邻算法，又可以分为基于用户的全局算法和基于项目的全局算法。基于模型的方法则有贝叶斯网络方法、奇异值分解的方法等，本文利用 MovieLens 数据集，在深度学习文本分类的基础上，通过对 MSE 值的计算优化模型，采用协同过滤算法，完成电影推荐的任务

关键词：个性化推荐；协同过滤；MSE

Abstract

Recommendation system is ubiquitous in daily network applications, such as online shopping, online book buying, news app, social network, music website, movie website and so on. According to personal preferences, the habits of people with the same preferences and other information for personalized content recommendation. For example, open the news app, because of the personalized content, everyone sees the news home page is different. This is of course very useful, in the information explosion today, access to information in a variety of ways and means, people spend the most time is no longer where to get information, but to find their interest in a lot of information, this is the information overload problem. In order to solve this problem, recommendation system comes into being.

Collaborative filtering technology is one of the most core technologies of recommendation system, and it is also the most widely used and successful technology at present. Collaborative filtering algorithms can be divided into global algorithms and model-based algorithms. Among them, global algorithms mainly rely on nearest neighbor algorithms, and can be divided into user-based global algorithms and project-based global algorithms. The methods based on model include bayesian network and singular value decomposition. In this paper, the collaborative filtering algorithm of the recommendation system is taken as the research target, and the task of movie recommendation is completed by calculating the MSE value of the algorithm by using text convolutional neural network and MovieLens data set

Key words: personalized recommendation; Collaborative filtering; MSE

目录

第一章 绪论.....1
1.1 研究背景与意义.....1
1.2 国内外研究现状.....1
1.2.1 国外研究现状.....1
1.2.2 国内研究现状.....3
1.3 本文研究工作与组织结构.....4
第二章 个性化推荐系统及其相关算法.....6
2.1 个性化推荐系统概述.....6
2.2 个性化推荐算法.....8
2.2.1 基于关联规则的推荐算法.....8
2.2.2 基于内容的推荐算法.....8
2.2.3 基于知识的推荐算法.....9
2.2.4 协同过滤推荐算法.....10
第三章 卷积神经网络.....19
3.1 卷积神经网络的相关研究.....19
3.2 卷积神经网络的基本结构.....19
3.2.1 卷积层.....20
3.2.2 激活函数.....20
3.2.3 池化层.....22
3.2.4 全连接层.....22
3.2.5 分类层.....23
3.3 卷积神经网络的特性.....24
3.3.1 局部感受野.....24
3.3.2 权值共享.....25
3.3.3 多卷积核与池化采样.....26
3.4 卷积神经网络的应用.....26
第四章 文本分类.....29
4.1 传统文本分类方法.....29
4.1.1 特征工程.....29
4.1.2 分类器.....31
4.2 深度学习文本分类方法.....31
4.2.1 文本的分布式表示：词向量(word embedding).....31
4.2.2 深度学习文本分类模型.....33
第五章 项目过程.....38
5.1 数据.....38
5.2 模型训练及推荐结果.....42
5.2.1 模型.....42
5.2.2 训练过程.....43
5.2.3 推荐结果.....44
5.3 前端页面说明文档.....46
5.3.1 布局.....46
5.3.2 特效.....46

5.3.3 功能 48
5.4 前端效果展示 48
第六章 项目实践总结 49
6.1 张庆轩个人工作总结 49
6.2 刘家钰个人工作总结 49
6.3 于晓刚个人工作总结 50
6.4 王梓瑞个人工作总结 51
参考文献 52

第一章 绪论

1.1 研究背景与意义

互联网的出现和普及给用户带来了大量的信息，据 IDC《数字宇宙》的研究报告表明，2020 年全球新建和复制的信息量将超过 40ZB，是 2012 年的 12 倍；中国的数据量在 2020 年超过 8ZB，比 2012 年增长 22 倍。互联网满足了用户在信息时代对信息的需求，但随着网络的迅速发展而带来的网上信息量的大幅增长，使得用户在面对大量信息时无法从中获得对自己真正有用的那部分信息，对信息的使用效率反而降低了，这就是所谓的信息超载（information overload）问题[1]。信息超载是目前网络用户面临的一个严重问题，个性化推荐系统是解决该问题的一个有力工具，并受到了众多的关注和研究。个性化推荐系统，它是根据用户的信息需求、兴趣等，将用户感兴趣的信息、产品等推荐给用户的个性化信息推荐系统。和搜索引擎相比推荐系统通过研究用户的兴趣偏好，进行个性化计算，由系统发现用户的兴趣点，从而引导用户发现自己的信息需求[2]。一个好的推荐系统不仅能为用户提供个性化的服务，还能和用户之间建立密切关系，让用户对推荐产生依赖[3]。对于商家而言，推荐系统可以给用户提供个性化服务，提供用户的信任度和粘性，增加营收，或者精确投放对应的广告，提高收入。通过一组简单的数据，我们即可了解推荐系统的价值：Netflix：2/3 被观看的电影来自推荐；Google news：38%的点击量来自推荐；Amazon：35%的销量来自推荐。性化推荐系统具有良好的发展和应用前景。目前，几乎所有的大型电子商务系统，如 Amazon、eBay 等，都不同程度的使用了各种形式的推荐系统。各种提供个性化服务的 Web 站点也需要推荐系统的大力支持。在日趋激烈的竞争环境下，个性化推荐系统能有效的保留客户，提高电子商务系统的服务能力。成功的推荐系统会带来巨大的效益。

1.2 国内外研究现状

1.2.1 国外研究现状

推荐系统最早应用于 Palo Alto 研究中心，Xerox 公司在 90 年代运用以 Tapestry 命名的协同过滤推荐系统来克服此研究中心的邮件信息过载难题由于

当时技术的限制，研究中心每日收到的邮件量超过了他们的处理能力，而且邮件系统不能将收到的邮件按照各自的类型合理的筛选归类，所以研究中心的研究人员就以此为契机开发出了一套可以帮助员工自动分类筛选邮件信息的邮件推荐系统。明尼苏达大学 GroupLens 研究组[4]，以用户主动为项目评分为基础，使用协同过滤算法开发出了一套 GroupLens 系统，并将其应用在 Usenet 新闻组中。这是首次真正有针对性的研究。在随后的深入研究中，该研究中心成员又在已开发的邮件分类处理系统基础上推出了在后来知名度较高的 MovieLens 电影推荐系统，这是一个专门针对协同过滤推荐系统进行研究的系统[5]。该系统通过长期观察用户观看电影后的评价，分析预测用户的观影兴趣，给用户推荐他们没有评分但可能会喜欢的电影。由于 MovieLens 电影推荐系统是最早研究的推荐系统，在互联网推荐领域起着标杆的作用，因此在后来相关学者研究推荐算法时，MovieLens 数据集一直作为首要标准来测试算法的性能[6]。国外相关领域的学者 Herlocker 是基于用户协同过滤推荐算法这个概念的最早提出以及应用者，其后来的广泛应用使推荐系统真正的进入了公共视野[25]。

基于用户的协同过滤推荐系统的核心算法是通过各用户间的相似程度，结合被推荐对象最近邻的几个相关用户对项目的评价，最后判断目标对象对未评分项目的喜爱程度[7]。不过最初的协同过滤算法也有缺陷，因为用户对项目的评分会不断变化以及新项目源源不断的添加，因此系统再次给目标用户推荐结果时都需要从新计算用户间的相似度以及最近邻用户。这对于少量用户数据信息的网站还可以应付，但是面对庞大的用户群以及亿万级的用户数据时，系统就不能及时的计算推荐结果，也就达不到推荐实时性的效果[8]。

针对上述缺陷，Sarwar 等研究者在协同过滤算法基础上提出了考虑物品相似度的推荐算法，即基于物品的协同过滤推荐算法[9]。该算法的核心技术是建立用户与项目之间的评分矩阵（通过搜集不同用户对目标项目的不同评价信息建立），通过用户-项目之间的评分矩阵，利用目标用户最近邻用户对项目的喜爱程度来确定目标用户对项目的兴趣爱好[10]。而在计算项目之间的相似度时可以通过离线数据来计算，用户对项目的评分短时间内不会改变，因此可以减少计算的频率以提高推荐系统的执行效率[11]。

此后，Slope One 由 Lemire 和 Daniel Lemire 等研究者[12]提出。它虽然是一种基于物品的协同过滤，但该算法不需要计算用户与物品之间相似度。此算法通过用户间对项目的评分平均值代替评分差，而两个项目之间的差异主要通过用户对项目的评分差来体现。因此，在获得用户对某物品的评分时，结合此物品对另一物品的差异，可以对用户对另一物品的评分做出相对可靠的预测[13]。由此可见，相比其他算法，此算法更加简单高效。 由于推荐系统在互联网行业的快速发展，使得学术界和工业界也看到了发展的曙光[14]。在学术界，ACM 从 2008 年开始每年都主办了专门的推荐系统国际会议(ACM Conference on Recommender System)，会议的研究热点话题引起了全球相关研究者的高度关注，该会议已成为推荐系统领域最具权威的会议。在工业界，Pandora、Jinni 这样的网站不断出现，以推荐系统为核心，这些网站成为了许多著名网站的辅助系统，当今主流社交软件均以此为基础向用户推荐可能认识的人，以及像 Youtube、Netflix、腾讯视频、爱奇艺视频等多媒体网站推荐给用户可能喜爱的视频。

1.2.2 国内研究现状

推荐系统已经在我国的电子商务和社交领域得到不同程度的应用，但相比于国外的成熟技术，目前我国在个性化推荐和自动推荐等方面的自主化研究尚处于初步阶段，大部分是借鉴国外成熟的技术成果[15]。当前，国内常见的应用个性化推荐技术的电子商务网站有当当网（dangdang.com）、豆瓣网（douban.com）、互动出版网（china-pub.com）、淘宝网（taobao.com）以及 360doc 网（360doc.com）等。随着电子商务和社交网络的迅速发展，协同过滤推荐也成为了国内个性化服务领域的热点研究对象。这些在线的推荐服务一般采用基于简单的用户历史行为统计的方法，而无需依赖用户提供特征信息，具有较强的学习能力，能够给用户带来较快和较好的操作体验。推荐系统作为个性化信息服务的重要组成，现已成为 Internet 领域新一代信息服务的有效应用形式。2009 年，国内首个个性化推荐系统研究团队成立，即百分点推荐技术研究中心（baifendian.com）。该团队专注于个性化推荐、推荐引擎技术与解决方案，在其个性化推荐引擎技术和数据平台上汇集了国内外百余家知名电子商务网站与资讯类网站[16]。

针对传统协同过滤技术仅通过单一的评分数据来度量用户兴趣相似性并以此建立用户兴趣模型的不足，国内学者在项目协同过滤基本方式的基础上，设计

了不同的改进方法，如文献[17]提出了一种基于多目标的兴趣度度量方法，在评价推荐质量时不仅适用推荐精度，同时考虑用户对推荐列表的感兴趣程度。文献[18]提出了一种基于项目特征属性的推荐算法，通过分析用户的历史行为和项目的特征进而挖掘出用户的行为模式和潜在兴趣，从而得到用户对项目特征的偏好程度，依此实现对新产品的推荐。文献[19]利用了统计信息对协同过滤进行了改进，在产生推荐时不仅考虑传统协同过滤的推荐结果，而且还加权考虑了用户和项目的统计信息，通过引入统计信息来达到增强推荐效果的目的。

协同过滤推荐算法的推荐精度作为一个重要的指标受到了企业界和学术界的共同关注，虽然研究学者们在如何提高推荐算法精度方面提出了各种各样的方法，然而现有的协同过滤推荐算法仍然存在一些缺陷与不足，国内外学者一直在尝试引入机器学习、概率算法、图论、矩阵降维、聚类等方法进行进一步的改进，所取得的成果仍存在较大的上升空间。

1.3 本文研究工作与组织结构

本文的主要的研究内容是通过用户对电影的评分运用深度学习及协同过滤算法构建用户电影推荐系统。利用 MovieLens 数据集，tensorflow 深度学习框架，在深度学习文本分类的基础上，通过对 MSE 值的计算优化模型，最后采用协同过滤算法，完成电影推荐的任务。

第一章重点说明本课题研究背景与意义,介绍了国内外新闻资讯应用的现状,并分析国内外个性化推荐技术的研究与应用进展,确定论文基本内容与结构。

第二章重点分析个性化推荐系统相关算法，重点对协同过滤算法做了详细的描述。

第三章介绍了卷积神经网络的基本结构和理论基础。首先概述了人工神经网络的发展历程以及所遇到的问题；其次介绍了卷积神经网络的基本理论，以及对卷积神经网络特有的权值共享、局部感知和多卷积核等特性都进行了概述。并介绍了卷积神经网络技术的应用现状和发展前景。

第四章讲的文本分类内容，主要介绍了传统的文本分类方法，以及深度学习的集中常见的文本分类方法，分析了传统文本分类方法的不足，并着重介绍了我们此次实践用到的 TextCNN 文本分类模型。

第五章对整个推荐系统的项目做了详细的描述并对系统的各功能模块进行了实现。主要分为数据预处理，项目模型设计，模型训练的过程和结果以及前端展示页面的说明。

第六章内容为总结与展望。本章主要对全文的工作内容进行总结,并指出了一些本课题研究中遇到的问题和未来可能有待改进的地方。

第二章 个性化推荐系统及其相关算法

2.1 个性化推荐系统概述

随着互联网的不断普及以及电子商务的快速发展,个性化推荐系统得到了广泛的应用,也逐渐成为当今的研究热点。个性化推荐系统是建立在大量数据的基础上,在用户没有明确的查找目的的情况下,模拟现实中的销售人员,来引导用户,从而满足用户的需求。所以,推荐系统有别于我们常使用的百度、谷歌等搜索引擎系统。搜索引擎系统是根据用户在搜索栏中的搜索关键词,或是对于物品 的描述去匹配与用户当前搜索内容最相近的物品,从而产生搜索结果。对当前对于用户来说,推荐系统就好比一个看不到黑盒子,用户通过在推荐系统中的一系列行为,都被操作系统获取并记录下来,进而推荐系统给出对该用户的推荐结果。推荐系统的框架如图 2-1 所示。

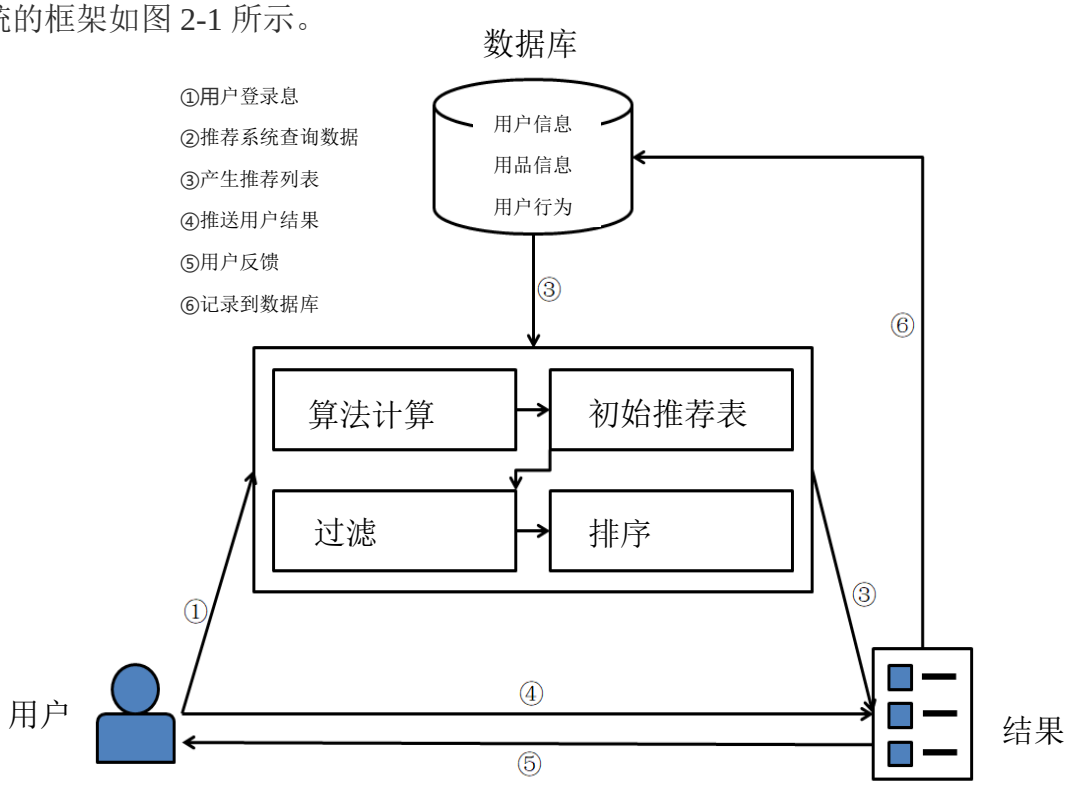

图 2-1 推荐系统框架

从图 2-1 中的结构我们可以看出,个性化推荐系统主要包括三个要素:用户,数据库和推荐算法。数据库用来存放物品信息、用户信息以及用户历史行为记录。当用户登录到系统中时,推荐系统会从数据库中读取该用户的历史行为记录以及相关物品资源信息。经过推荐算法的一系列运算,产生推荐结果,并将该推荐结果

推送给用户。当用户接收到推荐结果后,可能会有相应的点击、浏览等行为, 推荐系统可以将这些用户反馈信息记录到数据库中,以便以后进行推荐。常用的推荐结果包含以下几个方面：

1）商品推荐

商品推荐方式为电子商务网站中最常用的方式，当用户登录进入电商网站中，比如淘宝、亚马逊、ebay 等，这些网站会根据用户的历史行为信息对用户进行商品推荐，或者比较热门的商品的也会处被推荐给用户。这些推荐的商品信息会出新在页面的重要位置，比如页面的下面或者右侧面，且常常处于悬浮状态。

2）个体评分推荐

用户对于该物品没有相应的行为记录，如果对商品进行评分，那么需要根据其相邻用户对该商品的评价信息来进行评价。但是，个体评分不适合过多，在用户量很大的情况下，会加大数据处理的难度。

3）个体文本评价推荐

个体文本评价推荐在电子商务网站中往往和个体评分数据结合起来使用，目前中国使用个体文本评价推荐方式最好的电商网站是大众点评网，通过查看个体用户对商品的评价进行推荐。

4）平均评分推荐

平均评分推荐体现了在系统中所有用户对于目标物品的群体意见和评价。这些评价信息虽然隐藏了用户的个体评价信息，但是用户却可以藉此来参考这些推荐信息。

5）编辑推荐

编辑推荐是通过专家效应，通过专家对于目标物品的评价信息来向用户进行推荐。此类推荐在推荐系统中的应用非常广泛，比如在电商平台的首页，一般就会看到一些明星对于某系列商品的广告信息。

6）电子邮件推荐

电子邮件推荐是比较传统的推荐方式，推荐系统会定期的向用户在系统中预留的邮箱信息向用户发送推荐信息。这些信息可以是商品也可以是促销活动，推荐的方式可以是针对用户进行的个性化推荐也可以是整体推荐。目前，ebay 网还在坚持使用此类推荐方式。

2.2 个性化推荐算法

2.2.1 基于关联规则的推荐算法

基于关联规则的推荐是根据历史数据统计不同规则出现的关系，形如：X->Y，表示 X 事件发生后，Y 事件会有一定概率发生，这个概率是通过历史数据统计而来。关联规则的目的在于在一个数据集中找出项之间的关系，也称之为购物篮分析 (market basket analysis)。例如，购买鞋的顾客，有 10%的可能也会买袜子，60%的买面包的顾客，也会买牛奶。这其中最有名的例子就是"尿布和啤酒"的故事了。对于一个规则 X->Y，有两个指标对该规则进行衡量。一个是支持度，表示在所有样本数据中，同时包含 X 和 Y 样本的占比。另一个是置信度，表示在所有包含 X 的样本中，包含 Y 的样本占比。在关联推荐算法中，最主要的是如何找到最大频繁项，业界主要的做法有两种，分别为 Apriori 算法和 FP 树。Apriori 算法是生成频繁集的一种算法。Apriori 原理有个重要假设，如果某个项集是频繁的，那么它的所有子集势必也是频繁的。如果一个项集是非频繁项集，那么它所对应的超集就全都是非频繁项集。传统的 Apriori 算法的计算量很大，当商品数据量大时基本上效率很低，所以后来有 FP-Tree 算法优化了该算法。

2.2.2 基于内容的推荐算法

基于内容的推荐算法(Content-based Recommendations, CB)也是一种工业界应用比较广的一种推荐算法。基于内容的推荐技术广泛应用于推荐系统中,这项技术结合用户历史关注内容、特征和信息,计算与之相似的物品,再将相似度最高的若干项推送给用户。简单来说,就是通过用户以往关注的物品,发现并推荐与之类似的物品。例如用户曾经阅读过一本书刊,并对其给予了较高的评价,基于内容的推荐技术会分析这个书刊与其他书刊的相似程度,再将与该书刊相似程度最高的其他书刊推荐给用户。基于内容的推荐技术的流程主要分为三步,分别是物品表示、特征学习以及信息过滤。物品表示是将诸如新闻、网页等对象用合适的方式进行描述,方便算法下一步的处理。这些对象可以通过提取特征,将其表示为可处理的格式,对于新闻而言,可以将其用关键词向量进行处理。特征学习是根据用户以往的行为习惯,构建出用户的“画像这一步可以采用 KNN 算法、朴素贝叶斯分类等技术实现。训练用户行为信息,将用户喜欢和不喜欢的物品挑选出来,生成

用户的兴趣模型。信息过滤是指将系统中的物品与用户的兴趣模型进行匹配,得到用户最感兴趣的物品列表,将用户喜欢的物品推荐给用户。基于内容的推荐算法有以下两方面的优点。首先是可解释性,因为推荐的都是用户曾经喜欢的且相似度较高的物品,使得推荐结果更有可信度。其次是解决了冷启动问题,在物品初次进入系统时,由于确定了物品的内容特征,可以直接向感兴趣的用户进行推荐,因此不需要考虑用户的评分数据。但这项技术同样也存在一些现阶段无法解决的问题。例如基于内容的推荐目前只能对文本信息进行分析,其他的诸如视频、音乐等内容由于难以分析其内部特征,因此不适宜采用基于内容的推荐技术。其次,基于内容的推荐是根据历史记录进行预测,不能找出用户的潜在信息,只能向用户推荐以往感兴趣内容,不利于挖掘用户潜在的兴趣内容。

2.2.3 基于知识的推荐算法

传统的推荐算法适用于推荐特性或者口味相似的物品，比如：书籍、电影或者新闻。但是在对某些产品进行推荐的过程中，就有可能不是特别适合的方法，比如汽车、电脑、房屋、或者理财产品等等。主要是两个原因：很难在一个产品上获取大量的用户评分信息以及获得推荐的用户不会对这些已经过时的产品产生一个满意的回馈。

基于知识的推荐技术(Knowledge-based Recommendations, KB)是专门解决这类问题的一种新的推荐技术，高度重视知识源，不会存在冷启动的问题，因为推荐的需求都是被直接引出的。缺点是：所谓的知识的获取比较难，需要知识整理工程师将领域专家的知识整理成为规范的、可用的表达形式。基于知识的推荐技术需要主动的询问用户的需求，然后返回推荐结果。基于知识的推荐系统分为两大类：基于样列的推荐和基于约束的推荐；这两种方法非常相似：先收集用户需求，在找不到推荐方案的情况下，自动修复与需求的不一致性，并给出推荐的解释。区别在于：推荐方案是如何被计算出来的。1. 基于样列的推荐方法通过相似度衡量标准从目录中检索物品。 2. 基于约束的推荐方式主要是利用预先定义好的推荐知识库，即一些描述用户需求以及与这些需求相关的产品信息特征的显示关联规则；也就是使用约束求解器解决的约束满足问题或者通过数据库引擎执行并解决的合取查询形式。基于知识的推荐系统一般情况下需要依赖物品特征的详细知识；简单来讲，推荐就是从物品特征数量表中挑出能够匹配用户需求、

偏好和硬件需求的物品；用户的需求可能会表达成为：价格不超过在 2200 元的物品或者能够防水等等

2.2.4 协同过滤推荐算法

协同过滤推荐算法是诞生最早，并且较为著名的推荐算法，也是本文要介绍的重点。主要的功能是预测和推荐。算法通过对用户历史行为数据的挖掘发现用户的偏好，基于不同的偏好对用户进行群组划分并推荐品味相似的商品。协同过滤推荐算法分为两类，分别是基于用户的协同过滤算法(user-based collaboratIve filtering)，和基于物品的协同过滤算法(item-based collaborative filtering)。简单的说就是：人以类聚，物以群分。下面我们将分别说明这两类推荐算法的原理和实现方法。

2.2.4.1 基于用户的协同过滤算法

基于用户的协同过滤算法是通过用户的历史行为数据发现用户对商品或内容的喜欢(如商品购买，收藏，内容评论或分享)，并对这些喜好进行度量和打分。根据不同用户对相同商品或内容的态度和偏好程度计算用户之间的关系。在有相同喜好的用户间进行商品推荐。简单的说就是如果 user1，user3 两个用户都购买了 product2,product3 两个物品，并且给出了 5 星的好评。那么 user1 和 user3 就属于同一类用户。可以将 user1 买过的 product1，product4 也推荐给 user3。如下图所示

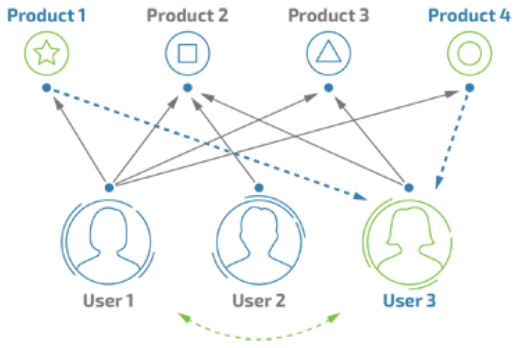

图 2-2 用户商品关系图

1）寻找偏好相似的用户

我们模拟了 5 个用户对两件商品的评分，来说明如何通过用户对不同商品的态度和偏好寻找相似的用户。在示例中，5 个用户分别对两件商品进行了评分。这里的分值可能表示真实的购买，也可以是用户对商品不同行为的量化指标。例

如，浏览商品的次数，向朋友推荐商品，收藏，分享，或评论等等。这些行为都可以表示用户对商品的态度和偏好程度。

	商品1	商品2
用户A	3.3	6.5
用户B	5.8	2.6
用户C	3.6	6.3
用户D	3.4	5.8
用户E	5.2	3.1

图 2-3 用户商品偏好程度

从表格中很难直观发现 5 个用户间的联系，我们将 5 个用户对两件商品的评分用散点图表示出来后，用户间的关系就很容易发现了。在散点图中，Y 轴是商品 1 的评分，X 轴是商品 2 的评分，通过用户的分布情况可以发现，A,C,D 三个用户距离较近。用户 A(3.3 6.5)和用户 C(3.6 6.3)，用户 D(3.4 5.8)对两件商品的评分较为接近。而用户 E 和用户 B 则形成了另一个群体。

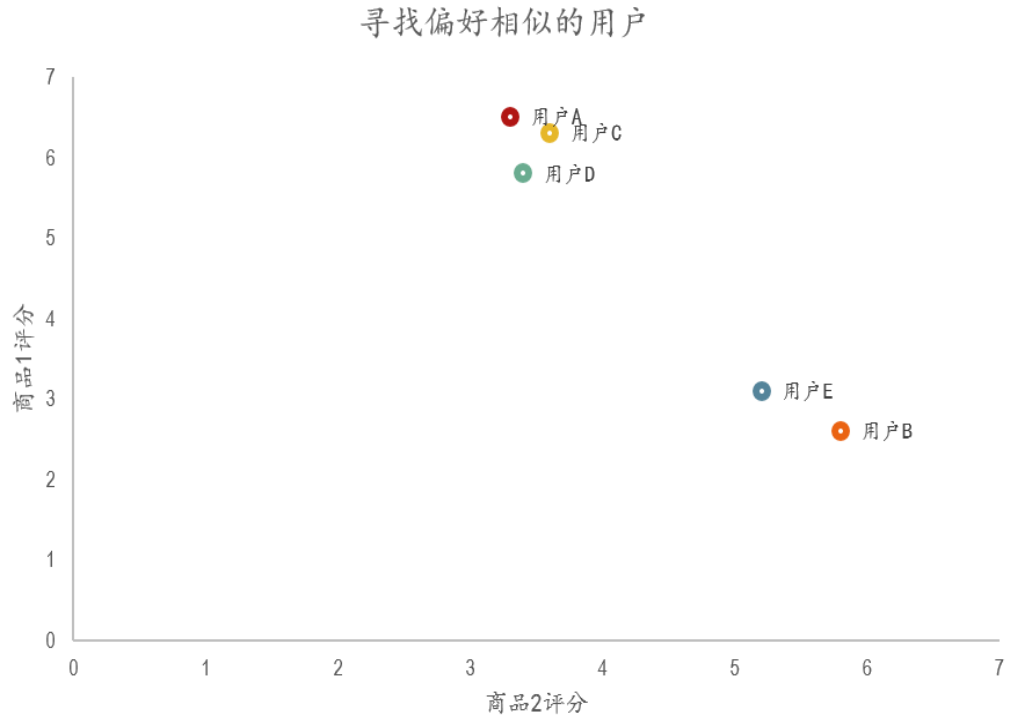

图 2-3 用户商品打分散点图

散点图虽然直观，但无法投入实际的应用，也不能准确的度量用户间的关系。因此我们需要通过数字对用户的关系进行准确的度量，并依据这些关系完成商品的推荐。

2） 欧几里德距离评价

欧几里德距离评价是一个较为简单的用户关系评价方法。原理是通过计算两个用户在散点图中的距离来判断不同的用户是否有相同的偏好。以下是欧几里德距离评价的计算公式。

$$d(x,y)=\sqrt{(\sum(x_i-y_i)^2)} \qquad \text{(公式 2-1)}$$

通过公式我们获得了 5 个用户相互间的欧几里德系数，也就是用户间的距离。系数越小表示两个用户间的距离越近，偏好也越是接近。不过这里有个问题，太小的数值可能无法准确的表现出不同用户间距离的差异，因此我们对求得的系数取倒数，使用户间的距离约接近，数值越大。在下面的表格中，可以发现，用户 A&C 用户 A&D 和用户 C&D 距离较近。同时用户 B&E 的距离也较为接近。与我们前面在散点图中看到的情况一致。

欧几里德距离评价

	系数	倒数
用户A&B	4.63	0.18
用户A&C	0.36	0.73
用户A&D	0.71	0.59
用户A&E	3.89	0.20
用户B&C	4.30	0.19
用户B&D	4.00	0.20
用户B&E	0.78	0.56
用户C&D	0.54	0.65
用户C&E	3.58	0.22
用户D&E	3.24	0.24

图 2-4 欧几里得距离评价

3）皮尔逊相关系数

皮尔逊相关度评价是另一种计算用户间关系的方法。他比欧几里德距离评价的计算要复杂一些，但对于评分数据不规范时皮尔逊相关度评价能够给出更好的结果。皮尔逊相关系数的计算公式如下，

$$\rho_{X,Y}=\frac{\sum(X-\overline{X})(Y-\overline{Y})}{\sqrt{\sum(X-\overline{X})^2\sum(Y-\overline{Y})^2}} \qquad \text{(公式 2-2)}$$

结果是一个在-1 与 1 之间的系数。该系数用来说明两个用户间联系的强弱程度。相关系数的分类：0.8-1.0 极强相关；0.6-0.8 强相关；0.4-0.6 中等程度相；0.2-0.4 弱相关；0.0-0.2 极弱相关或无相关通过计算 5 个用户对 5 件商品的评分我们获得了用户间的相似度数据。这里可以看到用户 A&B，C&D，C&E 和 D&E 之间相似度较高。下一步，我们可以依照相似度对用户进行商品推荐。

	相似度
用户A&B	0.9998
用户A&C	-0.8478
用户A&D	-0.8418
用户A&E	-0.9152
用户B&C	-0.8417
用户B&D	-0.8353
用户B&E	-0.9100
用户C&D	0.9990
用户C&E	0.9763
用户D&E	0.9698

图 2-5 皮尔逊相关系数

4）为相似的用户推荐物品

当我们需要对用户 C 推荐商品时，首先我们检查之前的相似度列表，发现用户 C 和用户 D 和 E 的相似度较高。换句话说这三个用户是一个群体，拥有相同的偏好。因此，我们可以对用户 C 推荐 D 和 E 的商品。但这里有一个问题。我们不能直接推荐前面的商品。因为这这些商品用户 C 以及浏览或者购买过了。不能重复推荐。因此我们要推荐用户 C 还没有浏览或购买过的商品。我们提取了用户 D 和用户 E 评价过的另外 5 件商品 A—商品 F 的商品。并对不同商品的评分进行相似度加权。按加权后的结果对 5 件商品进行排序，然后推荐给用户 C。这样，用户 C 就获得了与他偏好相似的用户 D 和 E 评价的商品。而在具体的推荐顺序和展示上我们依照用户 D 和用户 E 与用户 C 的相似度进行排序。

为用户C推荐商品

	相似度	商品A	商品A*	商品B	商品B*	商品C	商品C*	商品D	商品D*	商品E	商品E*	商品F	商品F*
用户D	0.99899	3.4	3.39656	4.4	4.395544	5.8	5.7941262	2.1	2.097873		0	3.8	3.796152
用户E	0.97627	3.2	3.12407		0	4.1	4.0027152	3.7	3.612206	5.3	5.174242	3.1	3.026443
总计			6.52063		4.395544		9.7968414		5.71008		5.174242		6.822595
相似度总计			1.97526		1.975259		1.9752593		1.975259		1.975259		1.975259
总计/相似度			3.30115		2.2253		4.9597749		2.8908		2.619525		3.454025

图 2-6 相似用户推荐

以上是基于用户的协同过滤算法。这个算法依靠用户的历史行为数据来计算相关度。也就是说必须要有一定的数据积累(冷启动问题)。对于新网站或数据量较少的网站，还有一种方法是基于物品的协同过滤算法。

2.2.4.2 基于物品的协同过滤算法

基于物品的协同过滤算法与基于用户的协同过滤算法很像，将商品和用户互换。通过计算不同用户对不同物品的评分获得物品间的关系。基于物品间的关系对用户进行相似物品的推荐。这里的评分代表用户对商品的态度和偏好。简单来说就是如果 user1，user2 同时购买了 product1 和 product3，那么说明商品 1 和商品 2 的相关度较高。当用户 user3 也购买了 product3 时，可以推断他也有购买 product1 的需求。

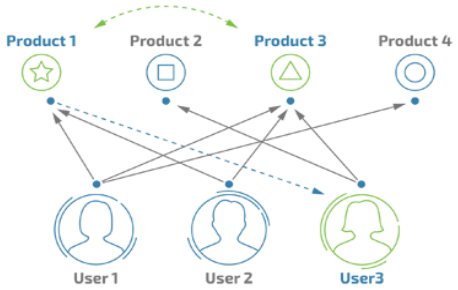

图 2-7 商品用户关系图

1) 寻找相似的物品

表格中是两个用户对 5 件商品的评分。在这个表格中我们用户和商品的位置进行了互换，通过两个用户的评分来获得 5 件商品之间的相似度情况。单从表格中我们依然很难发现其中的联系，因此我们选择通过散点图进行展示。

	用户A	用户B
商品1	3.3	6.5
商品2	5.8	2.6
商品3	3.6	6.3
商品4	3.4	5.8
商品5	5.2	3.1

图 2-8 户关对商品评分图

在散点图中，X 轴和 Y 轴分别是两个用户的评分。5 件商品按照所获的评分值分布在散点图中。我们可以发现，商品 1,3,4 在用户 A 和 B 中有着近似的评分，说明这三件商品的相关度较高。而商品 5 和 2 则在另一个群体中。

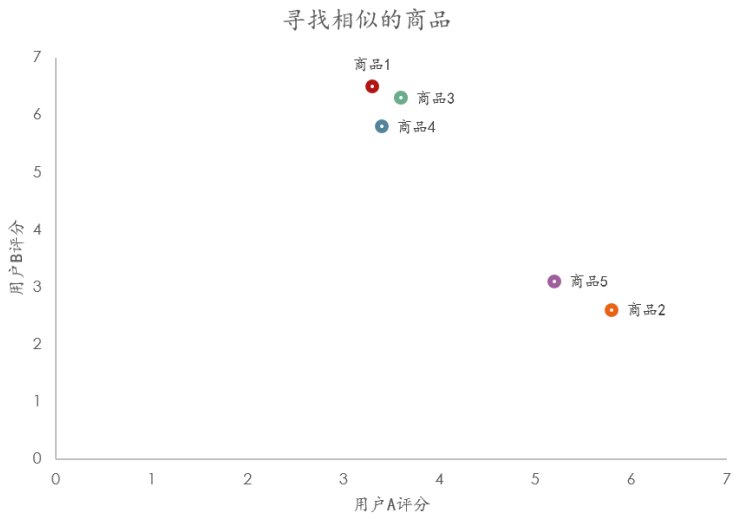

图 2-9 户关对商品散点图

在基于物品的协同过滤算法中，我们依然可以使用欧几里德距离评价来计算不同商品间的距离和关系。通过欧几里德系数可以发现，商品间的距离和关系与前面散点图中的表现一致，商品 1,3,4 距离较近关系密切。商品 2 和商品 5 距离较近。

欧几里德距离评价

	系数	倒数
商品1&2	4.63	0.18
商品1&3	0.36	0.73
商品1&4	0.71	0.59
商品1&5	3.89	0.20
商品2&3	4.30	0.19
商品2&4	4.00	0.20
商品2&5	0.78	0.56
商品3&4	0.54	0.65
商品3&5	3.58	0.22
商品4&5	3.24	0.24

图 2-10 欧几里得距离评价图

2) 皮尔逊相关系数

通过计算可以发现，商品 1&2，商品 3&4，商品 3&5 和商品 4&5 相似度较高。下一步我们可以依据这些商品间的相关度对用户进行商品推荐。

皮尔逊相关系数

	相似度
商品1&2	0.99977
商品1&3	-0.84776
商品1&4	-0.84182
商品1&5	-0.91524
商品2&3	-0.84174
商品2&4	-0.83532
商品2&5	-0.90998
商品3&4	0.99899
商品3&5	0.97627
商品4&5	0.96978

图 2-11 皮尔逊相关系数图

3) 为用户提供基于物品相似的推荐

这里我们遇到了和基于用户进行商品推荐相同的问题，当需要对用户 C 基于商品 3 推荐商品时，需要一张新的商品与已有商品间的相似度列表。在前面的相似度计算中，商品 3 与商品 4 和商品 5 相似度较高，因此我们计算并获得了商品 4,5 与其他商品的相似度列表。

	用户1	用户2	用户3
商品4	5.4	2.8	4.1
商品5	5.2	3.1	4.7
商品A	3.3	4.2	5.2
商品B	4.1	3.7	3.5
商品C	4.6	4.0	4.1

图 2-12 商品相似度评分图

以下是通过计算获得的新商品与已有商品间的相似度数据。

	相似度
商品4&5	0.95719
商品4&A	-0.47347
商品4&B	0.65465
商品4&C	0.93326
商品5&A	-0.19822
商品5&B	0.40780
商品5&C	0.78932
商品A&B	-0.97579
商品A&C	-0.75826
商品B&C	0.88250

图 2-13 商品与已有商品相似度

这里是用户 C 已经购买过的商品 4,5 与新商品 A,B,C 直接的相似程度。我们将用户 C 对商品 4,5 的评分作为权重。对商品 A,B,C 进行加权排序。用户 C 评

分较高并且与之相似度较高的商品被优先推荐。

对用户C基于商品3进行推荐

	评分	商品A	商品A*	商品B	商品B*	商品C	商品C*
商品4	4.1	-0.473466	-1.941209	0.654654	2.68408	0.93325653	3.826352
商品5	4.7	-0.198223	-0.931647	0.407804	1.916678	0.78931804	3.709795
总计			-2.872856		4.600758		7.536147
评分			8.8		8.8		8.8
总计/评分			-0.326461		0.522813		0.85638

图 2-14 对用户基于已有商品推荐

2.2.4.3 协同过滤算法详解

推荐系统应用数据分析技术，找出用户最可能喜欢的东西推荐给用户，现在很多电子商务网站都有这个应用。目前用的比较多、比较成熟的推荐算法是协同过滤（Collaborative Filtering，简称 CF）推荐算法，CF 的基本思想是根据用户之前的喜好以及其他兴趣相近的用户的选择来给用户推荐物品。

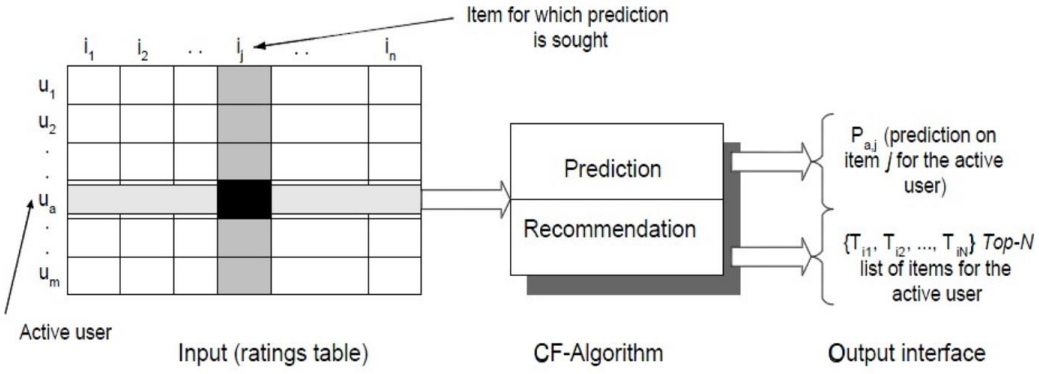

图 2-15 协同过滤算法图

如图所示，在 CF 中，用 m×n 的矩阵表示用户对物品的喜好情况，一般用打分表示用户对物品的喜好程度，分数越高表示越喜欢这个物品，0 表示没有买过该物品。图中行表示一个用户，列表示一个物品，Uij 表示用户 i 对物品 j 的打分情况。CF 分为两个过程，一个为预测过程，另一个为推荐过程。预测过程是预测用户对没有购买过的物品的可能打分值，推荐是根据预测阶段的结果推荐用户最可能喜欢的一个或 Top-N 个物品。

2.2.4.4 User-based 算法与 Item-based 算法对比

User-based 的基本思想是如果用户 A 喜欢物品 a，用户 B 喜欢物品 a、b、c，用户 C 喜欢 a 和 c，那么认为用户 A 与用户 B 和 C 相似，因为他们都喜欢 a，而

喜欢 a 的用户同时也喜欢 c，所以把 c 推荐给用户 A。该算法用最近邻居（nearest-neighbor）算法找出一个用户的邻居集合，该集合的用户和该用户有相似的喜好，算法根据邻居的偏好对该用户进行预测。

Iterm-based 的基本思想是预先根据所有用户的历史偏好数据计算物品之间的相似性，然后把与用户喜欢的物品相类似的物品推荐给用户。还是以之前的例子为例，可以知道物品 a 和 c 非常相似，因为喜欢 a 的用户同时也喜欢 c，而用户 A 喜欢 a，所以把 c 推荐给用户 A。

User-based 算法存在两个重大问题： 1. 数据稀疏性。一个大型的电子商务推荐系统一般有非常多的物品，用户可能买的其中不到 1%的物品，不同用户之间买的物品重叠性较低，导致算法无法找到一个用户的邻居，即偏好相似的用户。2. 算法扩展性。最近邻居算法的计算量随着用户和物品数量的增加而增加，不适合数据量大的情况使用。

因为物品直接的相似性相对比较固定，所以可以预先在线下计算好不同物品之间的相似度，把结果存在表中，当推荐时进行查表，计算用户可能的打分值，可以同时解决上面两个问题。

第三章 卷积神经网络

3.1 卷积神经网络的相关研究

卷积神经网络的框架是受到了自然界的生物视觉感知机制产生的深度学习的框架，在上个世纪六十年代初期，Hubel 和 Wiwsel 在研究动物的视觉系统的时候发现，猫的皮层细胞负责的是分层并且局部的检测光学信号，进而他们提出了感受野的概念，受到这个发现的启发，Fukunshima 在八十年代中期提出了一个新的识别标准[20] [21] [22]，这可以被认为是卷积神经网络的前身，自此 CNN 从网络结构上区别于传统的神经网络模型的结构，CNN 每层的神经元都是按照三维的形式去排列的[23]。传统的人工神经网络在处理数据的时候往往是比较复杂的，这需要大量的先验知识以及预处理操作，二十世纪九十年代，Lecun 等人[24]发表了一篇建立现代 CNN 框架的开创性的论文，后来又在论文[25]中对其进行了补充和改进，他们同时研发了一个名为 LeNet-5 的多层人工神经网络，像其他的神经网络一样，LeNet-5 具有多个层次，可以用来训练反向传播的算法[26]，它容易获得最原始图像的有效表示，这使得它可以直接从最原始的像素中直接去识别视觉的模式，几乎不需要做预处理的工作。

3.2 卷积神经网络的基本结构

卷积神经网络通常由输入层、卷积层、非线性层(激活函数层)、池化层、全连接

层以及分类器组成，其示意图如下。

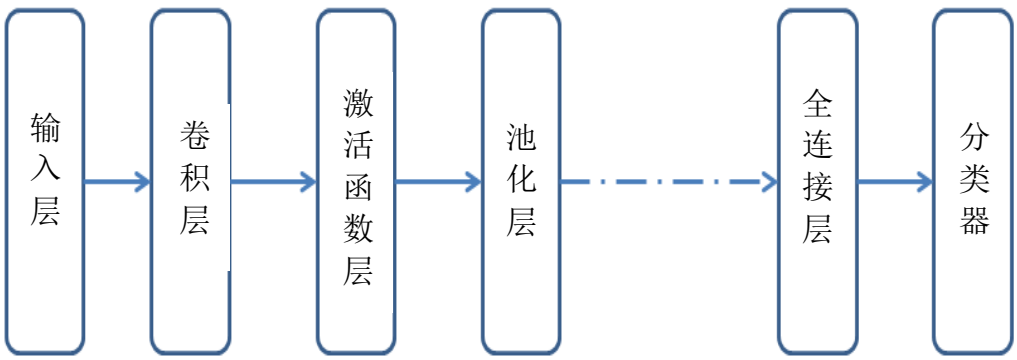

图 3-1 卷积神经网络的基本结构

3.2.1 卷积层

在图像处理中，卷积层常用来提取图像中的特征，不同的卷积核用来提取不同的特征。在实际情况中，卷积核采用一定步长的滑动窗口，对输入图像进行滑动采样，最终得到卷积层输出。图 3-2 为图像处理中常用的卷积计算案例。

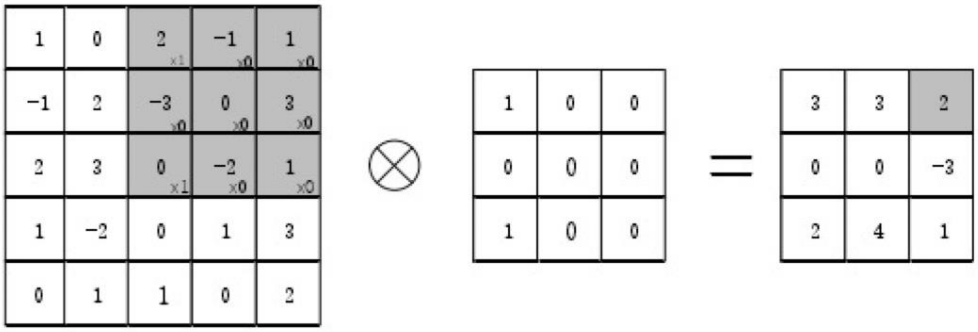

图 3-2 卷积计算案例

3.2.2 激活函数

激活函数是用来加入非线性因素的，因为线性模型的表达力不够

我们知道在神经网络中，对于图像，我们主要采用了卷积的方式来处理，也就是对每个像素点赋予一个权值，这个操作显然就是线性的。但是对于我们样本来说，不一定是线性可分的，为了解决这个问题，我们可以进行线性变化，或者我们引入非线性因素，解决线性模型所不能解决的问题。常用的激活函数 如下：

（1）Sigmoid 函数函数表达式：

$$f(x) = \frac{1}{1+e^{-x}}$$

其函数图像如图 3-3 所示。

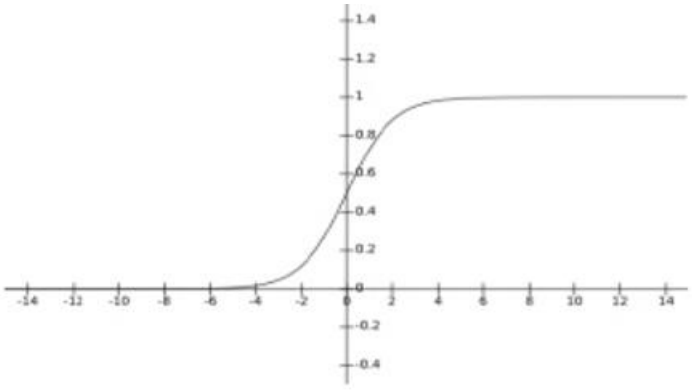

图 3-3 Sigmod 函数图像

由 Sigmod 函数图像可知，其值域在(0,1)之间，可以将输出映射至(0,1)之间。通常使用在二分类问题中。其缺点是求导复杂，在进行反向传播的时候计算量较大。并且函数两端导数接近于 0，在深层网络的反向传播中会引起梯度消失

（2） Tanh 函数

其函数图像如图 3-4 所示：

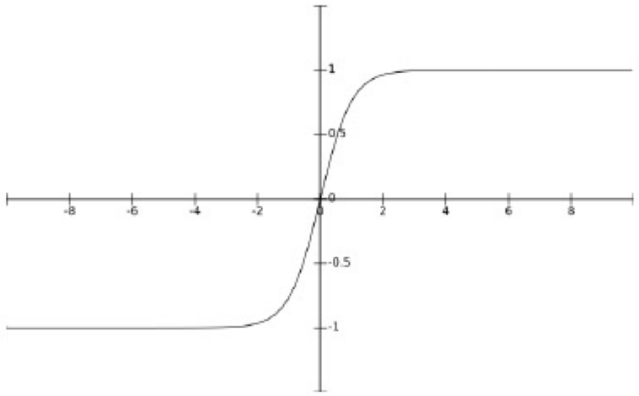

图 3-4 Tanh 函数图像

由 tanh 函数图像可知，tanh 函数的值域在(-1,1)之间。

相比 Sigmoid ,tanh 函数的输出以 0 为中心，输出范围更宽。但同样函数两端求导后接近于 0，容易造成梯度消失，不利于深层网络训练。

（3）ReLU 函数

ReLU 函数的表达式为： $f(z)=\max(0,z)$

其函数图像如图 3-5 所示。

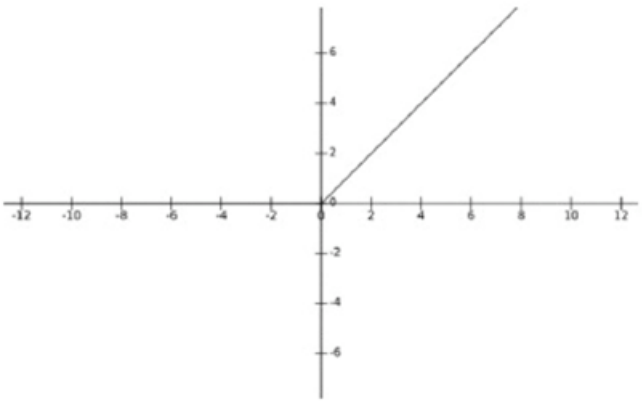

图 3-5 ReLU 函数图像

由 ReLU 函数图像可知，当输入为负时，输出恒为 0，当输入为正时，输出等于输入。ReLU 的优点在于较 Sigmoid 和 tanh 其导数计算量小，在卷积网络中计算中速度更快，并且负数输出为 0 增加了网络的稀疏性。

3.2.3 池化层

通常池化层(pooling layer)的输入为卷积层的输出。尽管卷积的稀疏性已经减少了大部分的计算参数，但是模型中的神经元并没有减少。此时数据维度依然很高，如果直接送入分类器将容易产生过拟合。池化层作用是降低特征数量，避免过拟合的产生。常用的池化操作有最大池化(max pooling)和平均池化(mean-pooing)。

图 3-6 中展示了常用的两种池化操作，最大池化和平均池化。定义一个空间邻域(图中的 2 x 2 窗口)，采用步长为 2 的方式，分别取出窗口中的最大值(最大池化方法)和平均值(平均池化方法)，最终可以得到一个 2x2 的输出图像特征。在实际应用中往往最大池化方法更好。

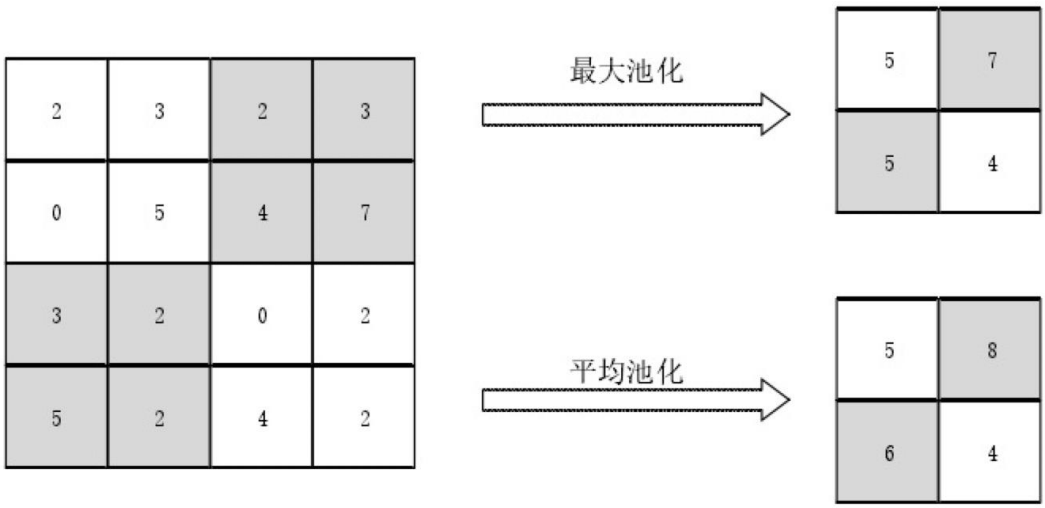

图 3-6 池化操作示意图

3.2.4 全连接层

全连接层中的每个节点均与上一层的每个节点相连接，将卷积神经网络输出的特征图转换为一维向量，将高维数据变到低维数据，在整个卷积神经网络中起到维度变换的作用，并且保留有效的信息，通常在卷积神经网络的最后使用，将输出交给分类器。

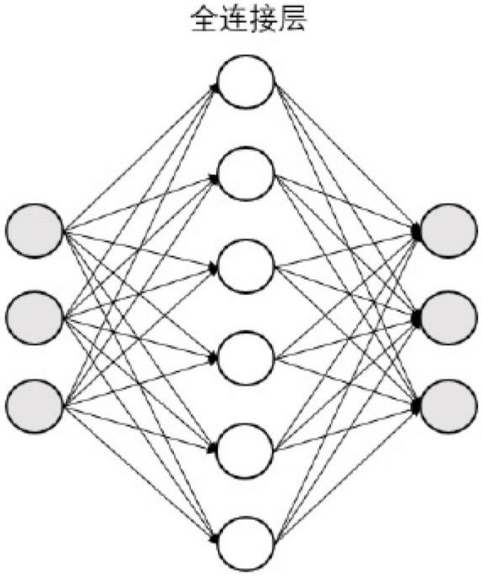

图 3-7 3 层全连接示意图

3.2.5 分类层

分类层是卷积神经网络的最后一层，分类层接收两个输入。其中一个是执行每一层的非线性函数得到的卷积神经网络的预测值，另外一个是手动打上的标签值。分类要做的工作是通过对这两个输入进行一系列的运算从而得到当前网络的损失函数： L (θ)，其中的 θ 代表着当前网络权值构成的向量空间，卷积神经网络训练的目的就是为了在权值空间中找到一个能够获得损失函数整体最小解的权值。目前，针对不同的场合设计出了相应的损失函数。对于单 Label 的分类问题，有 Softmax+交叉熵损失函数、另外还有应用于多分类的欧式损失函数、应用于 SVM 的 Hinge 损失函数、应用于多实行分类的 Sigmoid+交叉熵损失函数。最为理想的分类器是除了真实标签的概率为 1，其余的标签的概率均为 0，这样的分类器得到的损失函数为 ln (1)，也就是 0。一般来说，损失函数越大，则该分类器在真实标签上的分类概率就越小，性能也就越不良。对于那种损失函数都要接近于正无穷的分类器，就是特别差的分类器，也就是俗称的训练发散，需要进行改进

3.3 卷积神经网络的特性

当人工神经网络的层数不断加深之后，就会面临优化函数的解往往只是局部最优解而非全局最优解的问题。在使用有限的数据对深层网络进行训练时，得到的性能往往都比较浅层的网络要差。另外网络层数增高还会带来“梯度消失”的弊端，具体就是在将神经元的输入和输出函数设为 sigmoid 时，在 BP 反向传播梯度的过程中，幅度为 1 的信号会在传递过一层之后其衰减程度高达 75%。这样后面的层级是没有有效的训练信号可接收的。局部最优解的问题在 2006 年由 Hinton 教授率先克服，将能够有效训练的隐含层层级提高到 7 层，也从此引领了深度学习的另一波热潮。将 ReLU 和 maxout 等传输函数替代原有的 sigmoid，可在相当程度上解决梯度消失的现象，并最终形成了 DNN。就结构方面而言，全连接的 DNN 和多层感知机具有高度同一性，其结构中上下层所有神经元都能够连接，当然也引起了参数数量过度膨胀。面对上述的参数过多问题，卷积神经网络通过局部感受野，权值共享，多卷积核，下采样层来解决这个的问题，下面将依次进行介绍。

3.3.1 局部感受野

卷积神经网络能够大幅度的降低参数数目的原因之一在参数连接方面采用了新的处理技术，即局部感知，又称局部感受野。图像数据最为明显的一个特点就是局部像素之间并不是相互独立的，而是彼此相关，即局部相关性，网络隐含层的神经元利用图像数据的局部相关性，通过局部连接的方式实现图像的特征提取。图 3.8 展示的卷积神经网络算法局部感知特性示意图。

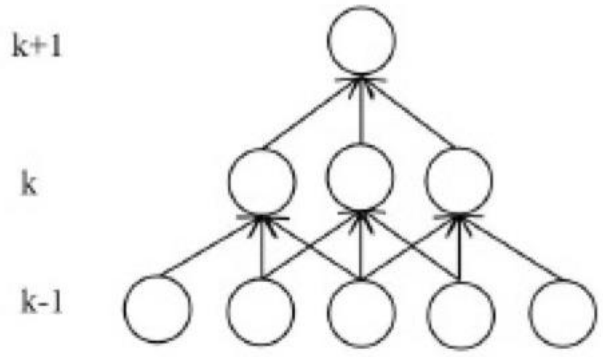

图 3-8 局部感知特性图像

通过分析图 3.8 可以得出以下结论。

（1）下层隐含层的输出神经元均是上层隐含层的输入神经元，例如第 k 层隐含层的输出就是第 k+1 层隐含层的输入。

（2）上层的输出神经元并不是与下层所有层的神经元相关，仅仅是一部分神经　元与输出神经元相关，例如第 k 层神经元仅与 3 个 k-1 层神经元相关联。上述的结论中详细叙述了连接方式和特点，这种连接方式就是卷积神经网络的局部感知特性。

3.3.2 权值共享

上述的局部感知技术对降低神经网络参数的问题具有一定的效果，但神经网络由于层数较多、参数数目过大，局部感知技术无法完全有效地解决问题。生物神经元在功能和结构上具有相似性和可取代性的特点，受此启发，卷积神经网络对得到下层隐含层输出的过程进行了一定的改进：传统人工神经网络中输出神经元与全部输入神经元相相关，而输出神经元数目较多且每个输出神经元的参数都各不相同，导致了参数数目过多的问题，在卷积神经网络中通过权值共享的方式解决了这个问题。如图 3.7k 层有三个输出神经元，k-1 层有六个输入神经元。只经过局部感知技术的参数数目为 3*3 个。权值共享的操作方法是对 k 层输出神经元都采用相同的参数，这就导致了参数数目的大量下降，从而实现降低参数数目的目的。在图 3.9 中，权值共享后的参数数目仅为 3 个，降低了 3 倍。

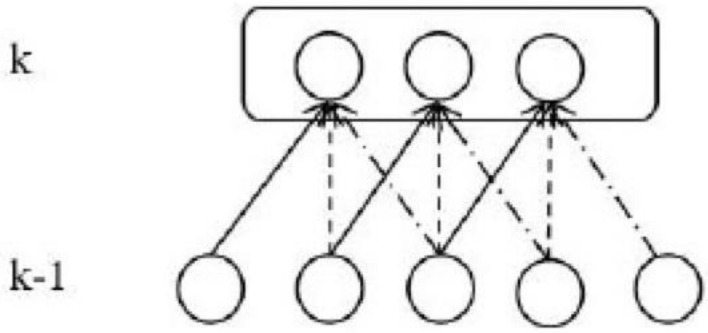

图 3-9 权值共享图

对权值共享的优点的总结如下。

（1）通过权值共享技术，实现了多组输出神经元只需要同一组权值的效果。相较于传统人工神经网络的采取的方式，降低了权值参数的数目，并在训练精度

上和训练时间上都得到了一定的提升，同时网络模型的复杂度也大大的降低了。

（2）使用同一组参数对整体的图像数据进行卷积操作，意味着在卷积神经网络训练的过程中，省去了提前确定图像特征在输入图像的具体位置的操作，不再需要对图像进行一个特征方面的预处理操作，降低了训练所需的时间和困难度。

3.3.3 多卷积核与池化采样

特征提取是图像分类最为关键的一步，也是卷积神经网络与传统神经网络区别最大的一步。卷积神经网络不再使用人工设计特征，而是在卷积运算的过程中实现了特征提取，降低了训练难度和训练时间。随着拍摄技术的不断发展，图片的像素，包含的信息都越来越多，一张图像的全部特征无法使用一个卷积核实现特征提取，所以在卷积神经网络中通过设计多个卷积核实现提取多个特征的目的，这多个卷积核在维度上完全一致，只是在权值上的初始化值不同，根据网络的不同、处理问题的不同，一般采用的初始化的方式也不同。卷积神经网络虽然通过局部感知和权值共享的处理手段完成了降低参数数目的功能，但是，一般卷积核的选取是远小于输入图像数据的大小，这就导致了输出特征图像的维度依旧较大，无法直接在下一层卷积层中进行直接使用，所以，一般会通过下采样的操作来进行特征图的降维工作，下采样又称为池化。对降维操作有关键影响的参数有两个：采样步长和池化函数。可以通过调节这两个参数实现对降维的基本控制。

3.4 卷积神经网络的应用

卷积神经网络经过这几十年的快速发展，已经从最初的手写字识别发展到了目前在生活的各个方面都有大量的应用，下面将简单的介绍相关的应用场景。

1)图像分类

图像分类[27][28][29]可以说是计算机视觉领域目前关注度与实用度均非常高的热点，其中很多较为典型的问题例如物体检测，图像分割等等都可以当作图像分类的问题来进行处理，这个问题虽然看似简单，但的确是计算机视觉领域的核心问题。

2)字符识别光学字符识别（OCR）

OCR[30]就是通过对图像进行一定的处理从而能够提取出来图像中的文字及版面信息的过程。较为常见的应用就是身份证的实名认证，每个购物网站进行分

期购买商品时都是需要实名认证，具体的做法就是通过手机对身份证进行拍摄，购物 APP 会自动扫描出身份信息，这背后的原理就是 OCR。OCR 算法目前有基于传统方法和深度学习的两种方法，深度学习虽然大大提高了 OCR 的识别性能，而且泛化能力更强，但是还是不能够完全的取代传统方法，因为深度学习方法的解码速度较慢而且需要训练的样本也过多。

3）目标检测

目标检测[31]也是计算机视觉中最为常见的问题之一，目标检测最终要实现的目标是机器能够判断它的视野内都存在什么物品，是什么种类，并且处于什么位置。近年来，目标检测算法在强大的深度神经网络的加持下性能得到了大幅度的提升，在其中的目标检测框架 R-CNN 可以说是扮演了重要的角色。R-CNN 框架对静态的图像的目标检测具有非常好的性能，但是 R-CNN 并不是针对视频这种输入数据进行专门设计的，所以在视频的处理上还是有所欠缺。最近提出了另一种学习框架 T-CNN，这种框架能够从视频中获得时域和上下文信息，显著的

改善了视频中的目标检测性能。

4）人脸识别

从 20 世纪 60 年代到目前为止，人脸识别技术[32][33][34]一直是各界较为关注的热门项目，相关的技术在安全领域，医学领域，人机交互中都有着巨大的应用前景。简单的来说，人脸识别技术的目的就是在于能够快速将图片上的人与真人的身份对应上。该技术主要有两个大类：一类是人脸验证，即 1:1 问题；另一类为人脸识别，即 1:n 问题。人脸识别技术现在也面对着许多的挑战，比如：人的表情；人的年龄的变化；光线的问题以及遮挡问题等等，传统的人脸识别技术很难去有效的解决这些问题，但是随着深度学习的不

断发展，基于深度学习的人脸识别研究取得了巨大的突破。许多机构也投入了很多的精力和金钱进行人脸识别研究，这其中 FaceBook 公司的 DeepFace 技术，以及香港中文大学的 FACE++团队最为有名。目前，越来越多学术上的和工业上的研究者投身于人脸识别技术的研究之中，而且根据最新 LFW 数据库的测试结果，可以看出由于深度学习技术的不断进步，人脸识别目前在实用中已经有了超过 99%的准确率。不过，高正确率并不意味着可以进行完全放心的推广应用。究其原因在于，现有模型都是基于某个特殊集合训练后得来的结果，无

法确定在应用到其他集合中时依然保持高度的准确率。因此人脸识别技术在将来的研发与应用中都还要有很长的路要走。

5) 自然语言处理

自然语言处理[35][36]的目的就是让计算机能够处理、理解自然语言，从而可以和人类进行更加直接的交互。目前较为常见的 NLP 应用有：拼写检查、关键字搜索、同义词发现、文本分类、机器翻译等等。NLP 的困难点在于语言表达具有多样性，上下文关联性和模糊性。目前斯坦福大学 NLP 研究组开发的 NLP—Caffe 在 NLP 方面获得了较为不错的效果。

第四章 文本分类

4.1 传统文本分类方法

文本分类问题算是自然语言处理领域中一个非常经典的问题了，相关研究最早可以追溯到上世纪 50 年代，当时是通过专家规则（Pattern）进行分类，甚至在 80 年代初一度发展到利用知识工程建立专家系统，这样做的好处是短平快的解决 top 问题，但显然天花板非常低，不仅费时费力，覆盖的范围和准确率都非常有限。

后来伴随着统计学习方法的发展，特别是 90 年代后互联网在线文本数量增长和机器学习学科的兴起，逐渐形成了一套解决大规模文本分类问题的经典玩法，这个阶段的主要套路是人工特征工程+浅层分类模型。训练文本分类器过程见图 4-1，整个文本分类问题就拆分成了特征工程和分类器两部分。

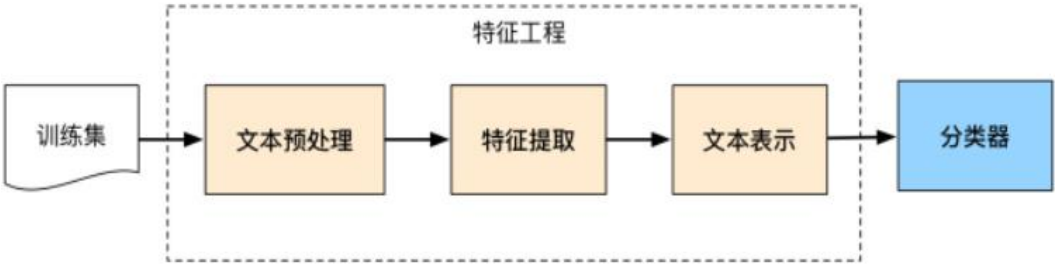

图 4-1 分类过程

4.1.1 特征工程

特征工程在机器学习中往往是最耗时耗力的，但却极其的重要。抽象来讲，机器学习问题是把数据转换成信息再提炼到知识的过程，特征是“数据－>信息”的过程，决定了结果的上限，而分类器是“信息－>知识”的过程，则是去逼近这个上限。然而特征工程不同于分类器模型，不具备很强的通用性，往往需要结合对特征任务的理解。

文本分类问题所在的自然语言领域自然也有其特有的特征处理逻辑，传统分本分类任务大部分工作也在此处。文本特征工程分位文本预处理、特征提取、文本表示三个部分，最终目的是把文本转换成计算机可理解的格式，并封装足够用于分类的信息，即很强的特征表达能力。

1）文本预处理

文本预处理过程是在文本中提取关键词表示文本的过程，中文文本处理中主要包括文本分词和去停用词两个阶段。之所以进行分词，是因为很多研究表明特征粒度为词粒度远好于字粒度，其实很好理解，因为大部分分类算法不考虑词序信息，基于字粒度显然损失了过多“n-gram”信息。

具体到中文分词，不同于英文有天然的空格间隔，需要设计复杂的分词算法。传统算法主要有基于字符串匹配的正向/逆向/双向最大匹配；基于理解的句法和语义分析消歧；基于统计的互信息/CRF 方法。近年来随着深度学习的应用，WordEmbedding + Bi-LSTM+CRF 方法逐渐成为主流，本文重点在文本分类，就不展开了。而停止词是文本中一些高频的代词连词介词等对文本分类无意义的词，通常维护一个停用词表，特征提取过程中删除停用表中出现的词，本质上属于特征选择的一部分。

经过文本分词和去停止词之后淘宝商品示例标题变成了下图“ / ”分割的一个个关键词的形式：

夏装 / 雪纺 / 条纹 / 短袖 / t 恤 / 女 / 春 / 半袖 / 衣服 / 夏天 / 中长款 / 大码 / 胖 mm / 显瘦 / 上衣 / 夏

2）文本表示

文本表示的目的是把文本预处理后的转换成计算机可理解的方式，是决定文本分类质量最重要的部分。传统做法常用词袋模型（BOW, Bag Of Words）或向量空间模型（Vector Space Model），最大的不足是忽略文本上下文关系，每个词之间彼此独立，并且无法表征语义信息。词袋模型的示例如下：

(0, 0, 0, 0, ... , 1, ... 0, 0, 0, 0)

一般来说词库量至少都是百万级别，因此词袋模型有个两个最大的问题：高纬度、高稀疏性。词袋模型是向量空间模型的基础，因此向量空间模型通过特征项选择降低维度，通过特征权重计算增加稠密性。

3）特征提取

向量空间模型的文本表示方法的特征提取对应特征项的选择和特征权重计算两部分。特征选择的基本思路是根据某个评价指标独立的对原始特征项（词项）

进行评分排序，从中选择得分最高的一些特征项，过滤掉其余的特征项。常用的评价有文档频率、互信息、信息增益、x^2统计量等。

特征权重主要是经典的 TF-IDF 方法及其扩展方法，主要思路是一个词的重要度与在类别内的词频成正比，与所有类别出现的次数成反比。

传统做法在文本表示方面除了向量空间模型，还有基于语义的文本表示方法，比如 LDA 主题模型、LSI/PLSI 概率潜在语义索引等方法，一般认为这些方法得到的文本表示可以认为文档的深层表示，而 word embedding 文本分布式表示方法则是深度学习方法的重要基础

4.1.2 分类器

分类器基本都是统计分类方法了，基本上大部分机器学习方法都在文本分类领域有所应用，比如朴素贝叶斯分类算法（Na ïve Bayes）、KNN、SVM、最大熵和神经网络等等

4.2 深度学习文本分类方法

前面介绍了传统的文本分类做法，传统做法主要问题的文本表示是高纬度高稀疏的，特征表达能力很弱，而且神经网络很不擅长对此类数据的处理；此外需要人工进行特征工程，成本很高。而深度学习最初在之所以图像和语音取得巨大成功，一个很重要的原因是图像和语音原始数据是连续和稠密的，有局部相关性，。应用深度学习解决大规模文本分类问题最重要的是解决文本表示，再利用 CNN/RNN 等网络结构自动获取特征表达能力，去掉繁杂的人工特征工程，端到端的解决问题。接下来会分别介绍：

4.2.1 文本的分布式表示：词向量(word embedding)

分布式表示（Distributed Representation）其实 Hinton 最早在 1986 年就提出了，基本思想是将每个词表达成 n 维稠密、连续的实数向量，与之相对的 one-hot encoding 向量空间只有一个维度是 1，其余都是 0。分布式表示最大的优点是具备非常 powerful 的特征表达能力，比如 n 维向量每维 k 个值，可以表征 kn 个概念。事实上，不管是神经网络的隐层，还是多个潜在变量的概率主题模型，都是应用分布式表示。下图是 03 年 Bengio 在 A Neural Probabilistic Language Model 的网络结构：

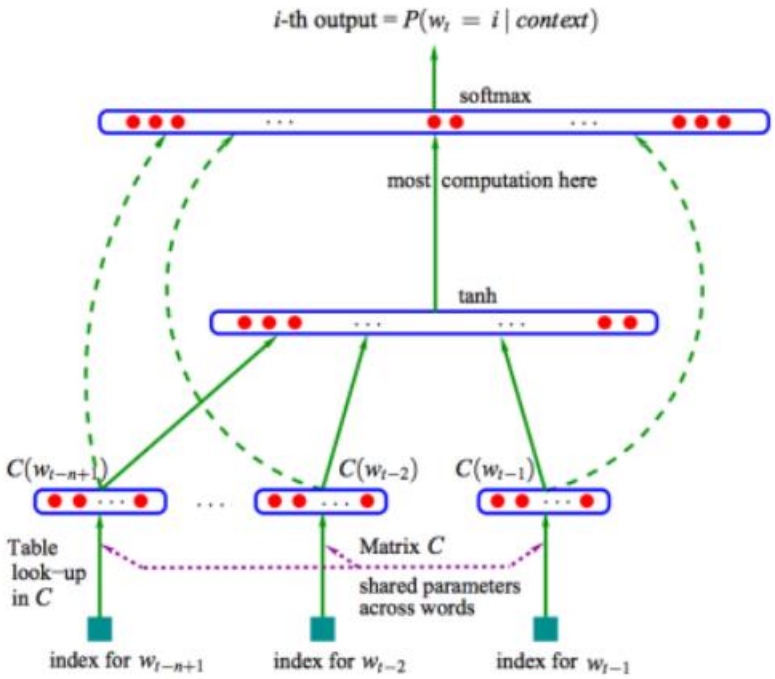

图 4-2 网络结构

这篇文章提出的神经网络语言模型（NNLM，Neural Probabilistic Language Model）采用的是文本分布式表示，即每个词表示为稠密的实数向量。NNLM 模型的目标是构建语言模型：

$$f(w_t,\cdots,w_{t-n+1}) = \tilde{P}(w_t \mid w_1^{t-1}) \qquad \text{(公式 4-1)}$$

词的分布式表示即词向量（word embedding）是训练语言模型的一个附加产物，即图中的 Matrix C。

尽管 Hinton 86 年就提出了词的分布式表示，Bengio 03 年便提出了 NNLM，词向量真正火起来是 google Mikolov 13 年发表的两篇 word2vec 的文章 Efficient Estimation of Word Representations in Vector Space 和 Distributed Representations of Words and Phrases and their Compositionality，更重要的是发布了简单好用的 word2vec 工具包，在语义维度上得到了很好的验证，极大的推进了文本分析的进程。下图 3-3 是文中提出的 CBOW 和 Skip-Gram 两个模型的结构，基本类似于 NNLM，不同的是模型去掉了非线性隐层，预测目标不同，CBOW 是上下文词预测当前词，Skip-Gram 则相反。

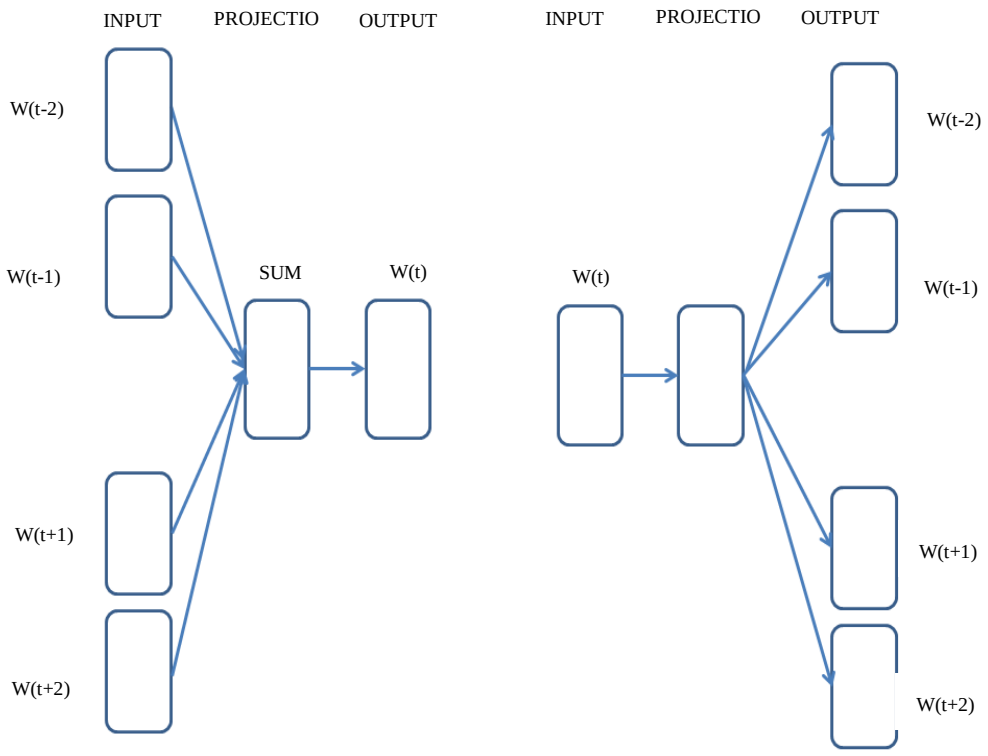

图 4-3 模型结构

除此之外，还提出了 Hierarchical Softmax 和 Negative Sample 两个方法，很好的解决了计算有效性，事实上这两个方法都没有严格的理论证明，有些 trick 之处，非常的实用主义。实际上 word2vec 学习的向量和真正语义还有差距，更多学到的是具备相似上下文的词，比如“good”“bad”相似度也很高，反而是文本分类任务输入有监督的语义能够学到更好的语义表示。

至此，文本的表示通过词向量的表示方式，把文本数据从高纬度高稀疏的神经网络难处理的方式，变成了类似图像、语音的的连续稠密数据。深度学习算法本身有很强的数据迁移性，很多之前在图像领域很适用的深度学习算法比如 CNN 等也可以很好的迁移到文本领域了，下一小节具体阐述下文本分类领域深度学习的方法。

4.2.2 深度学习文本分类模型

CNN/RNN 等深度学习网络及其变体解决自动特征提取（即特征表达）的问题。我们的算法就是利用的 TextCNN 文本分类方法，下面我们将详细介绍我们用的方法，以及几种其他的深度学习文本分类模型。

1） TextCNN

在短文本分析任务中，由于句子句长长度有限、结构紧凑、能够独立表达意思，使得 CNN 在处理这一类问题上成为可能，主要思想是将 ngram 模型与卷积操作结合起来。此图选用的就是 14 年这篇文章提出的 TextCNN 的结构论文 Convolutional Neural Networks for Sentence Classification。卷积神经网络（CNN Convolutional Neural Network）最初在图像领域取得了巨大成功，核心点在于可以捕捉局部相关性，具体到文本分类任务中可以利用 CNN 来提取句子中类似 n-gram 的关键信息。

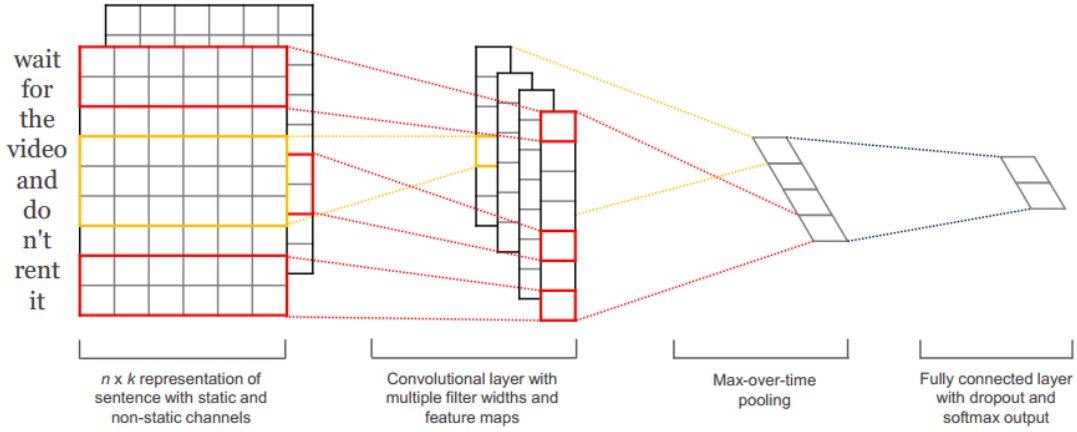

图 4.4 模型结构

TextCNN 详细过程：

第一层是图中最左边的 7 乘 5 的句子矩阵，每行是词向量，维度=5，这个可以类比为图像中的原始像素点了。然后经过有 filter_size=(2,3,4) 的一维卷积层，每个 filter_size 有两个输出 channel。第三层是一个 1-max pooling 层，这样不同长度句子经过 pooling 层之后都能变成定长的表示了，最后接一层全连接的 softmax 层，输出每个类别的概率。

特征：

这里的特征就是词向量，有静态（static）和非静态（non-static）方式。static 方式采用比如 word2vec 预训练的词向量，训练过程不更新词向量，实质上属于迁移学习了，特别是数据量比较小的情况下，采用静态的词向量往往效果不错。non-static 则是在训练过程中更新词向量。推荐的方式是 non-static 中的 fine-tunning 方式，它是以预训练（pre-train）的 word2vec 向量初始化词向量，训

练过程中调整词向量，能加速收敛，当然如果有充足的训练数据和资源，直接随机初始化词向量效果也是可以的。

通道（Channels）：

图像中可以利用 (R, G, B) 作为不同 channel，而文本的输入的 channel 通常是不同方式的 embedding 方式（比如 word2vec 或 Glove），实践中也有利用静态词向量和 fine-tunning 词向量作为不同 channel 的做法。

一维卷积（conv-1d）：

图像是二维数据，经过词向量表达的文本为一维数据，因此在 TextCNN 卷积用的是一维卷积。一维卷积带来的问题是需要设计通过不同 filter_size 的 filter 获取不同宽度的视野。

Pooling 层：

利用 CNN 解决文本分类问题的文章还是很多的，比如这篇 A Convolutional Neural Network for Modelling Sentences 最有意思的输入是在 pooling 改成 (dynamic) k-max pooling ，pooling 阶段保留 k 个最大的信息，保留了全局的序列信息。比如在情感分析场景，举个例子：

“ 我觉得这个地方景色还不错，但是人也实在太多了 ”

虽然前半部分体现情感是正向的，全局文本表达的是偏负面的情感，利用 k-max pooling 能够很好捕捉这类信息。

2） fastText

fastText 是上文提到的 word2vec 作者 Mikolov 转战 Facebook 后 16 年 7 月刚发表的一篇论文 Bag of Tricks for Efficient Text Classification。把 fastText 放在此处并非因为它是文本分类的主流做法，而是它极致简单，模型图见下图 4.6；

原理是把句子中所有的词向量进行平均（某种意义上可以理解为只有一个 avg pooling 特殊 CNN），然后直接接 softmax 层。其实文章也加入了一些 n-gram 特征的 trick 来捕获局部序列信息。文章倒没太多信息量，算是“水文”吧，带来的思考是文本分类问题是有一些“线性”问题的部分[from 项亮]，也就是说不必做过多的非线性转换、特征组合即可捕获很多分类信息，因此有些任务即便简单的模型便可以搞定了。

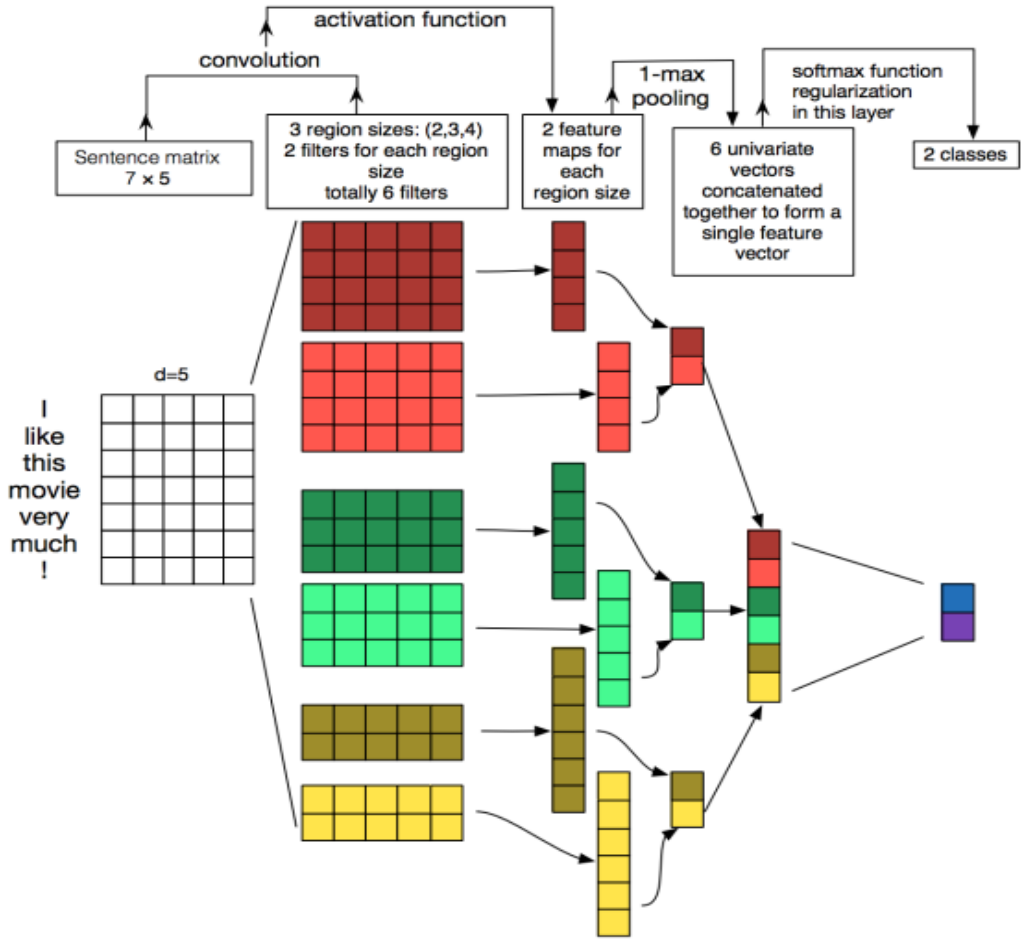

图 4.5 TextCNN 过程原理结构

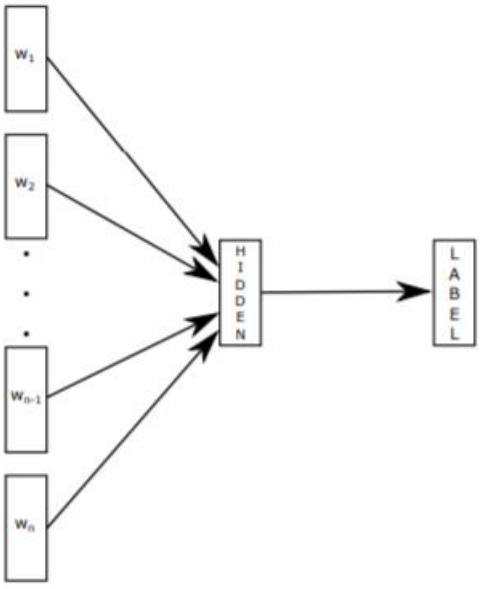

图 4.6 fastText 模型结构

3） TextRNN

尽管 TextCNN 能够在很多任务里面能有不错的表现，但 CNN 有个最大问题是固定 filter_size 的视野，一方面无法建模更长的序列信息，另一方面 filter_size 的超参调节也很繁琐。CNN 本质是做文本的特征表达工作，而自然语言处理中更常用的是递归神经网络（RNN, Recurrent Neural Network），能够更好的表达上下文信息。具体在文本分类任务中，Bi-directional RNN（实际使用的是双向 LSTM）从某种意义上可以理解为可以捕获变长且双向的的 “n-gram” 信息。

双向 LSTM 算是在自然语言处理领域非常一个标配网络了，在序列标注/命名体识别/seq2seq 模型等很多场景都有应用，下图 4.7 是 Bi-LSTM 用于分类问题的网络结构原理示意图，黄色的节点分别是前向和后向 RNN 的输出，示例中的是利用最后一个词的结果直接接全连接层 softmax 输出了。

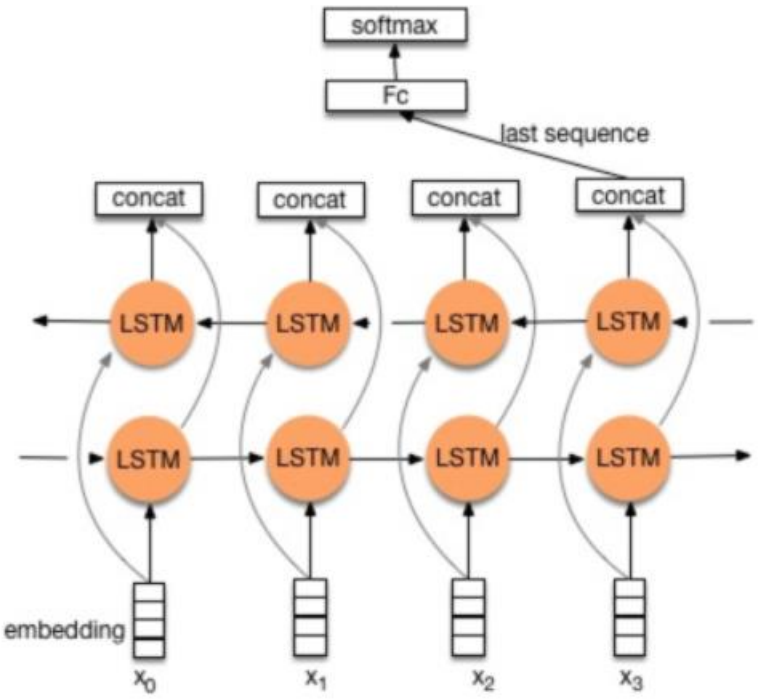

图 4.7 TextRNN 模型结构

第五章 项目过程

本文使用文本卷积神经网络，并使用 MovieLens 数据集完成电影推荐的任务。

5.1 数据

本项目使用的是 MovieLens 1M 数据集，包含 6000 个用户在近 4000 部电影上的 1 亿条评论。数据集分为三个文件：用户数据 users.dat，电影数据 movies.dat 和评分数据 ratings.dat。

1）用户数据

分别有用户 ID、性别、年龄、职业 ID 和邮编等字段。

数据中的格式：UserID::Gender::Age::Occupation::Zip-code

性别表示用 "M" 代表 male 即男性 用 "F" 代表 female 即女性

年龄分布分为以下一个区间:

1: "Under 18"

18: "18-24"

25: "25-34"

35: "35-44"

45: "45-49"

50: "50-55"

56: "56+"

职业有以下几个选项:

0: "other" or not specified

1: "academic/educator"

2: "artist"

3: "clerical/admin"

4: "college/grad student"

5: "customer service"

6: "doctor/health care"

7: "executive/managerial"

8: "farmer"

9: "homemaker"

10: "K-12 student"

11: "lawyer"

12: "programmer"

13: "retired"

14: "sales/marketing"

15: "scientist"

16: "self-employed"

17: "technician/engineer"

18: "tradesman/craftsman"

19: "unemployed"

20: "writer"

序号	UserID	Gender	Age	Occupation	Zip-code
0	1	F	Q	10	48067
1	2	M	56	16	70072
2	3	M	25	15	55117
3	4	M	45	7	02460
4	5	M	25	20	55455

表 5.1 用户数据表

2）电影数据

分别有电影 ID、电影名和电影风格等字段。

数据中的格式：MovieID::Title::Genres

标题与 IMDB 提供的标题相同

电影类型分为以下几个方面:

Action

Adventure

Animation

Children's

Comedy

Crime

Documentary

Drama

Fantasy

Film-Noir

Horror

Musical

Mystery

Romance

Sci-Fi

Thriller

War

Western

序号	MovieID	Title	Genres
0	1	Toy Story(1995)	Animation|Children's|Comedy
1	2	Jumanji(1995)	Adventur|Children's|Fantasy
2	3	Grumpier Old Men(1995)	Comedy|Romance
3	4	Waiting to Exhale(1995)	Comedy|Drama
4	5	Father of the Bride Part II(1995)	Comedy

表 5.2 电影数据表

3）评分数据

分别有用户 ID、电影 ID、评分和时间戳等字段。

数据中的格式：UserID::MovieID::Rating::Timestamp

UserIDs range between 1 and 6040

MovieIDs range between 1 and 3952

Ratings are made on a 5-star scale (whole-star ratings only)

Timestamp is represented in seconds since the epoch as returned by time(2)

Each user has at least 20 ratings

序号	UserID	MovieID	Ratinge	Timestamps
0	1	1193	5	978300760

1	1	661	3	978302109
2	1	914	3	978301968
3	1	3408	4	978300275
4	1	2355	5	978824291

表 5.3 评分数据表

4）数据预处理

UserID、Occupation 和 MovieID 不用变。

Gender 字段：需要将‘F’和‘M’转换成 0 和 1。

Age 字段：要转成 7 个连续数字 0~6。

Genres 字段：是分类字段，要转成数字。首先将 Genres 中的类别转成字符串到数字的字典，然后再将每个电影的 Genres 字段转成数字列表，因为有些电影是多个 Genres 的组合。

Title 字段：处理方式跟 Genres 字段一样，首先创建文本到数字的字典，然后将 Title 中的描述转成数字的列表。另外 Title 中的年份也需要去掉。

Genres 和 Title 字段需要将长度统一，这样在神经网络中方便处理。空白部分用‘< PAD >’对应的数字填充

5）加载数据并保存到本地

title_count：Title 字段的长度（15）

title_set：Title 文本的集合

genres2int：电影类型转数字的字典

features：是输入 X

targets_values：是学习目标 y

ratings：评分数据集的 Pandas 对象

users：用户数据集的 Pandas 对象

movies：电影数据的 Pandas 对象

data：三个数据集组合在一起的 Pandas 对象

movies_orig：没有做数据处理的原始电影数据

users_orig：没有做数据处理的原始用户数据

6） 预处理后的数据集

In [23]:
```
users.head()
```

Out[23]:

	UserID	Gender	Age	JobID
0	1	0	0	10
1	2	1	5	16
2	3	1	6	15
3	4	1	2	7
4	5	1	6	20

In [24]:
```
movies.head()
```

Out[24]:

	MovieID	Title	Genres
0	1	[3001, 5100, 275, 275, 275, 275, 275, 275, 275...	[3, 6, 2, 17, 17, 17, 17, 17, 17, 17, 17, 17, ...
1	2	[2280, 275, 275, 275, 275, 275, 275, 275, 275,...	[16, 6, 14, 17, 17, 17, 17, 17, 17, 17, 17, 17...
2	3	[4339, 3338, 348, 275, 275, 275, 275, 275, 275...	[2, 13, 17, 17, 17, 17, 17, 17, 17, 17, 17, 17...
3	4	[4507, 4093, 596, 275, 275, 275, 275, 275, 275...	[2, 4, 17, 17, 17, 17, 17, 17, 17, 17, 17, 17,...
4	5	[2123, 4479, 2698, 2221, 4495, 3997, 275, 275,...	[2, 17, 17, 17, 17, 17, 17, 17, 17, 17, 17, 17...

图 5.1 处理后的数据展示图

5.2 模型训练及推荐结果

5.2.1 模型

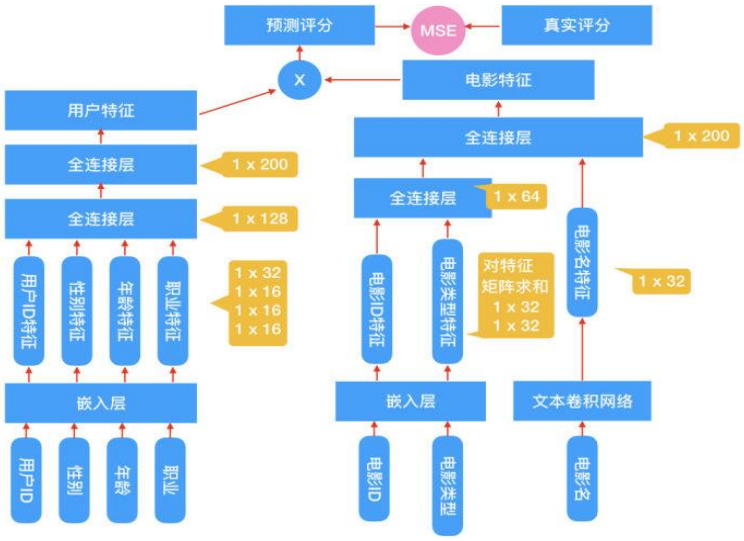

图 5.2 模型结构图

通过研究数据集中的字段类型，我们发现有一些是类别字段，通常的处理是将这些字段转成 one hot 编码，但是像 UserID、MovieID 这样的字段就会变成非常的稀疏，输入的维度急剧膨胀，这是我们不愿意见到的所以在预处理数据时将这些字段转成了数字，我们用这个数字当做嵌入矩阵的索引，在网络的第一层使用了嵌入层，维度是（N，32）和（N，16）。电影类型的处理要多一步，有时一个电影有多个电影类型，这样从嵌入矩阵索引出来是一个（n，32）的矩阵，

因为有多个类型，我们要将这个矩阵求和，变成（1，32）的向量。电影名的处理比较特殊，没有使用循环神经网络，而是用了文本卷积网络。从嵌入层索引出特征以后，将各特征传入全连接层，将输出再次传入全连接层，最终分别得到（1，200）的用户特征和电影特征两个特征向量。我们的目的就是要训练出用户特征和电影特征，在实现推荐功能时使用。得到这两个特征以后，就可以选择任意的方式来拟合评分了。 将电影特征和用户特征这两个特征作为输入，再次传入全连接层，输出一个值，将输出值回归到真实评分，采用 MSE 优化损失

文本卷积网络的第一层是词嵌入层，由每一个单词的嵌入向量组成的嵌入矩阵。下一层使用多个不同尺寸（窗口大小）的卷积核在嵌入矩阵上做卷积，窗口大小指的是每次卷积覆盖几个单词。这里跟对图像做卷积不太一样，图像的卷积通常用 2x2、3x3、5x5 之类的尺寸，而文本卷积要覆盖整个单词的嵌入向量，所以尺寸是（单词数，向量维度），比如每次滑动 3 个，4 个或者 5 个单词。第三层网络是 max pooling 得到一个长向量，最后使用 dropout 做正则化，最终得到了电影 Title 的特征。

5.2.2 训练过程

1）数据集划分使用的是 sklearn 的 train_test_split，训练集和测试集比例为 4:1，随机种子(random_state)不固定。

2）超参设置：

num_epochs = 5
Batch Size
batch_size = 256
dropout_keep = 0.5
Learning Rate
learning_rate = 0.0001
show_every_n_batches = 20

3）更新 loss 函数参数：使用梯度下降优化 loss 函数,最终 loss 收敛到 1 左右，loss 曲线如下：

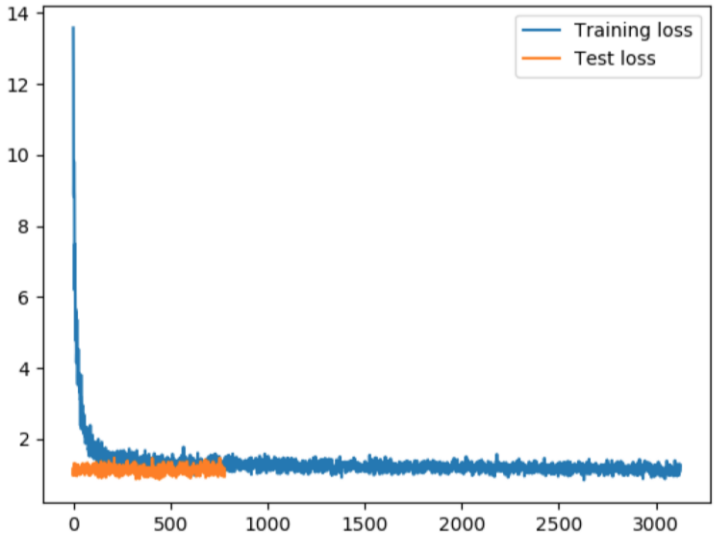

图 5.3　loss 曲线图

5.2.3 推荐结果

1）推荐同类型电影

计算当前看的电影特征向量与整个电影特征矩阵的余弦相似度，取相似度最大的 top_k 个，项目中加了些随机选择在里面，保证每次的推荐稍稍有些不同。

```
recommend_same_type_movie(1401, 20)
```

```
您看的电影是：[1401 'Ghosts of Mississippi (1996)' 'Drama']
以下是给您的推荐：
3385
[3454 'Whatever It Takes (2000)' 'Comedy|Romance']
707
[716 'Switchblade Sisters (1975)' 'Crime']
2351
[2420 'Karate Kid, The (1984)' 'Drama']
2189
[2258 'Master Ninja I (1984)' 'Action']
2191
[2260 'Wisdom (1986)' 'Action|Crime']
```

图 5.4 同类型电影推荐结果展示图

2）推荐喜欢的电影

使用用户特征向量与电影特征矩阵计算所有电影的评分，取评分最高的 top_k 个，同样加了些随机选择部分。

```
recommend_your_favorite_movie(234, 10)
```

```
以下是给您的推荐:
1642
[1688 'Anastasia (1997)' "Animation|Children's|Musical"]
994
[1007 'Apple Dumpling Gang, The (1975)' "Children's|Comedy|Western"]
667
[673 'Space Jam (1996)' "Adventure|Animation|Children's|Comedy|Fantasy"]
1812
[1881 'Quest for Camelot (1998)' "Adventure|Animation|Children's|Fantasy"]
1898
[1967 'Labyrinth (1986)' "Adventure|Children's|Fantasy"]
```

图 5.5 喜欢电影推荐结果展示图

3）看过这个电影的人还看了哪些电影

首先选出喜欢某个电影的 top_k 个人，得到这几个人的用户特征向量。然后计算这几个人对所有电影的评分，选择每个人评分最高的电影作为推荐，此过程中同样加入了随机选择：

```
您看的电影是: [1401 'Ghosts of Mississippi (1996)' 'Drama']
喜欢看这个电影的人是: [[5782 'F' 35 0]
 [5767 'M' 25 2]
 [3936 'F' 35 12]
 [3595 'M' 25 0]
 [1696 'M' 35 7]
 [2728 'M' 35 12]
 [763 'M' 18 10]
 [4404 'M' 25 1]
 [3901 'M' 18 14]
 [371 'M' 18 4]
 [1855 'M' 18 4]
 [2338 'M' 45 17]
 [450 'M' 45 1]
 [1130 'M' 18 7]
 [3035 'F' 25 7]
 [100 'M' 35 17]
 [567 'M' 35 20]
 [5861 'F' 50 1]
 [4800 'M' 18 4]
 [3281 'M' 25 17]]
喜欢看这个电影的人还喜欢看:
1779
[1848 'Borrowers, The (1997)' "Adventure|Children's|Comedy|Fantasy"]
1244
[1264 'Diva (1981)' 'Action|Drama|Mystery|Romance|Thriller']
1812
[1881 'Quest for Camelot (1998)' "Adventure|Animation|Children's|Fantasy"]
1742
[1805 'Wild Things (1998)' 'Crime|Drama|Mystery|Thriller']
2525
```

图 5.6 此人还看了哪些电影展示

5.3 前端页面说明文档

5.3.1 布局

Bootstrap 布局，使用 container 做容器，内置栅格系统来保证不同分辨率下的页面布局兼容。使用 bootstrap 的内联表单来保证不同浏览器下表单样式统一。

```
<div class="container part3">
    <h2 class="text-center">看过这个电影的人还喜欢：</h2>
    <form class="form-inline">
        <div class="form-group">
            <label class="sr-only" for="otherLikeMovie">请输入电影ID</label>
            <input type="number" class="form-control" id="otherLikeMovie" placeholder="请输入电影ID:">
        </div>
        <div class="form-group">
            <label class="sr-only" for="otherLikeMovieNumber">推荐人群数量</label>
            <input type="number" class="form-control" id="otherLikeMovieNumber" placeholder="推荐人群数量">
        </div>
        <button type="button" class="btn btn-primary" id="btn3">提交</button>
    </form>
    <div class="row">
        <div class="col-lg-6 peaple">
            <h3>喜欢这个电影的人是：</h3>
            <p class="none"></p>
        </div>
        <div class="col-lg-6 movie">
            <h3>他们喜欢的电影是：</h3>
            <p class="none"></p>
        </div>
    </div>

</div>
```

图 5.7 布局代码展示

5.3.2 特效

步骤：使用 canvas 标签，动态设置宽高与页面相等。

```
width = window.innerWidth;
main = $("#main");
height = document.documentElement.clientHeight > document.body.clientHeight ? document.documentElement.
// target = {x: width/2, y: height/2};
target = {x: width, y: height};
canvas = document.getElementById( elementId: 'demo-canvas');
canvas.width = width;
canvas.height = height;
ctx = canvas.getContext('2d');
```

图 5.8 特效代码设置宽高

随机生成一定数量的点坐标（粒子位置）、随机生成 X 和 Y 运动速率。将点数组画到 canvas 画布上。

```
// create points
points = [];
for(var x = 0; x < width; x = x + width/20) {
    for(var y = 0; y < height; y = y + height/20) {
        var px = x + Math.random()*width/20;
        var py = y + Math.random()*height/20;
        var p = {x: px, originX: px, y: py, originY: py };
        points.push(p);
    }
}
```

图 5.9 特效代码生成点坐标

遍历点数组，两两比较点坐标，寻找最近的 5 个点连线(drawLines)并形成一个环

```
// for each point find the 5 closest points
for(var i = 0; i < points.length; i++) {
    var closest = [];
    var p1 = points[i];
    for(var j = 0; j < points.length; j++) {
        var p2 = points[j]
        if(!(p1 == p2)) {
            var placed = false;
            for(var k = 0; k < 5; k++) {
                if(!placed) {
                    if(closest[k] == undefined) {
                        closest[k] = p2;
                        placed = true;
                    }
                }
            }

            for(var k = 0; k < 5; k++) {
                if(!placed) {
                Duplicate declaration
                stance(p1, p2) < getDistance(p1, closest[k])) {
                        closest[k] = p2;
                        placed = true;
                    }
                }
            }
        }
    }
    p1.closest = closest;
}

// assign a circle to each point
for(var i in points) {
    var c = new Circle(points[i], 2+Math.random()*2, 'rgba(255,255,255,0.3)');
    points[i].circle = c;
}
```

图 5.10 特效代码连线

```
function Circle(pos,rad,color) {
    var _this = this;

    // constructor
    (function() {
        _this.pos = pos || null;
        _this.radius = rad || null;
        _this.color = color || null;
    })();

    this.draw = function() {
        if(!_this.active) return;
        ctx.beginPath();
        ctx.arc(_this.pos.x, _this.pos.y, _this.radius, 0, 2 * Math.PI, false);
        ctx.fillStyle = 'rgba(156,217,249,'+ _this.active+')';
        ctx.fill();
    };
}
```

图 5.11 特效代码设置宽高

循环调用 shiftPoint 方法，在原点位置附近改变动点位置。

```
function shiftPoint(p) {
    TweenLite.to(p, 1+1*Math.random(), {x:p.originX-50+Math.random()*100,
        y: p.originY-50+Math.random()*100, ease:Circ.easeInOut,
        onComplete: function() {
            shiftPoint(p);
        }});
}
```

图 5.12 特效代码调用 shiftPoint 方法

同时设置监听函数 addListeners，来监听鼠标滑动 mouseMove 位置、以及页面大小改变 resize 情况下的动画重绘。

```
// Event handling
function addListeners() {
    if(!('ontouchstart' in window)) {
        window.addEventListener( type: 'mousemove', mouseMove);
    }
    window.addEventListener( type: 'scroll', scrollCheck);
    window.addEventListener( type: 'resize', resize);
```

图 5.13 特效代码动画重画

5.3.3 功能

jQuery 提交表单数据，后台根据前端表单返回相应数据，前端通过 ajax 将获取到的数据动态渲染到页面中。

5.4 前端效果展示

由于代码演示，执行结果都是在我们的编辑器里面，考虑到这一缺点，我们为了跟好的用户体验，以及操作显示更加方便，我们设计了一个前端界面，将前面的推荐功能整合到我们的前端界面展示，前端界面显示如下图 5.14 所示；我们只需在第一个输入框中输入电影的 id 值，或者用户的 id 值，在第二个输入框中输入 top_n 的个数即可获得推荐的结果。

图 5.14 推荐系统前端界面展示

第六章　项目实践总结

6.1 张庆轩个人工作总结

在本次实践课程中主要负责模型训练调参和部分编码实现的工作。我们的系统利用 MovieLens 数据集，在深度学习文本分类的基础上，采用协同过滤算法，完成电影推荐的任务。推荐分为三个功能：第一，通过输入的电影 id 来推荐同类型的电影；第二，通过输入的用户 id 来推荐其喜欢的电影；第三，通过输入的电影 id 来推测喜欢这个电影的人，以及他们还喜欢哪些电影。通过此次实践，我应用到了很多书本上和课上老师提到的理论知识，不但加深了对理论的理解，更增强了实际动手能力。有很多技术理念，只知道它的概念和作用，却不知如何在工程中使用，比如 dropout 和卷积核的使用等。当然，在项目中也遇到了一下工程性的问题，比如在页面与服务器的前后端分离开发中遇到的跨域问题，还有服务启动时的端口占用问题等，通过这些问题，大家才能更加了解整个应用的生态系统。

本文中使用的协同过滤作为一种经典的推荐算法种类，在业界应用广泛，它的优点很多，模型通用性强，不需要太多对应数据领域的专业知识，工程实现简单，效果也不错，这些都是它流行的原因。

当然，协同过滤也有些难以避免的难题，比如令人头疼的“冷启动”问题，我们没有新用户任何数据的时候，无法较好的为新用户推荐物品。同时也没有考虑情景的差异，比如根据用户所在的场景和用户当前的情绪等等，这些都需要系统进行一些具体的优化。

6.2 刘家钰个人工作总结

随着网络信息量的急剧增长,人们面对海量信息时难以准确得到自己感兴趣的内容。诸如谷歌、百度等搜索引擎一定程度上缓解了这个问题,但是它们无法满足用户的个性化需求。因此,个性化推荐系统应运而生,它根据用户以往的行为偏好,将符合用户喜好的信息呈现给用户。个性化推荐系统给用户带来了私人定制般的服务体验,能有效地增强用户粘度,避免用户流失。协同过滤是推荐系统应

用较广泛的技术，该方法搜集用户的历史记录、个人喜好等信息，计算与其他用户的相似度，利用相似用户的评价来预测目标用户对特定项目的喜好程度。我在本次工程实践中负责的主要任务是数据集的预处理，文献资料的整理以及论文的撰写。我们使用的数据集是 MovieLens 1M 数据集，包含 6000 个用户在近 4000 部电影上的 1 亿条评论，数据集有三个文件，分别是用户数据，电影数据及评分数据。在数据预处理的过程中，有些字段及信息这次没选择作为训练的属性，比如用户邮编及评论的时间戳等，还需要做的就是将数据集中所有的字段数字字典化，这里会遇到一些问题，由于文本长度不一，为了在神经网络中处理方便，需要将 Genres 和 Title 字段统一长度，因此空白部分用‘< PAD >’对应的数字填充。在项目的文档撰写中系统的学习了一个基本的推荐系统需要部分及知识模块，加之在深度学习课上老师对推荐系统的理论讲解，自己动手实践一遍更加印象深刻了，推荐系统的种类繁多还需要更深入细致的钻研。此次实践还有一些不足之处，协同过滤是推荐系统应用较广泛的技术，该方法搜集用户的历史记录、个人喜好等信息，计算与其他用户的相似度，利用相似用户的评价来预测目标用户对特定项目的喜好程度，这个算法是会给用户推荐未浏览过的项目，但是对于新用户来说，由于没有任何历史记录以及个人偏好等信息，就存在冷启动问题，导致该算法无法对新用户进行推荐，然而本次实践没有对冷启动问题进行深入探索。其次可能还存在其他更好的模型，只是我们还没有发现及应用，可多进行尝试几个其他的网络模型。这些还需要进一步的完善。

6.3 于晓刚个人工作总结

本次的工程实践作业，是一个让我从理论到实践的一个学习过程，使我不仅复习巩固了之前学习的知识，也更加的进一步的理解了一些之前理解不到的内容；我们本次大作业是商品推荐算法的设计与实现应用于电影推荐的一个应用，主要用到了 TensorFlow 深度学习框架，使用了文本神经网络，并使用开源数据集 MovieLens 数据集完成电影推荐的任务。我的主要工作内容就是负责测试，验证模型的好坏，以及收集整理需要的资料，完成文档撰写和最终文档的格式标注化，文档格式完全按照《北航硕士论文格式规范》撰写。

在定制化推荐系统里面主要分为协同过滤和基于内容推荐两个部分,而我们最常用的一种算法就是协同过滤算法,而总的来说,协同过滤的目的就是找相似,找相似的人或者相似的东西；该方法搜集用户的历史记录、个人喜好等信息，计算与其他用户的相似度,利用相似用户的评价来预测目标用户对特定项目的喜好程度。优点是会给用户推荐未浏览过的项目，缺点呢，对于新用户来说，没有任何与商品的交互记录和个人喜好等信息，存在冷启动问题，导致模型无法找到相似的用户或商品。但是在我们本次的本次的作业内容没有处理该部分的内容，主要是对有一定浏览信息的用户进行推荐,所以想要做成一个更好更完整的大的系统,还需要更多的时间和精力以及技术条件,这也要求我们组更进一步的去学习,去完善自己的工作内容。最后我们组共同的努力完成本次的实践作业， 真正认识到，每个人的不同擅长内容，团队工作特别重要，所以，我们也会在未来的工作，科研，学习过程中发挥自己特长贡献自己力量，不断学习完善自身，注重团队工作，集体的力量。

6.4 王梓瑞个人工作总结

本次工程实践大作业是商品推荐算法的设计与实现应用于电影推荐应用。主要技术为 TensorFlow 深度学习框架，使用了文本神经网络、开源数据集 MovieLens。个人工作为前端页面的设计和实现。

前端界面功能共分为三个部分。一、通过电影 ID 获取同类型电影。二、通过用户 ID 获取该用户可能喜欢的电影。三、通过电影 ID 获取喜欢这个电影的用户群，以及该用户群喜欢的电影。通过这次作业，在很多工程性的东西理解和使用上，我比以前有了更深的认识。比如，对页面的设计、对不同浏览器和不同分辨率的兼容、对 canvas 的使用以及 JS 性能上的优化有了更深刻的理解。

当然，在团队合作中也遇到了一些问题以及有待优化的地方，比如在前后联调中遇到的跨域问题，经商议确定了最优解决方案。今后要完善的部分如：用户冷启动问题，即在匿名场景下的推荐方案。

参考文献

[1] 刘建国,周涛,汪秉宏. 个性化推荐系统的研究进展[J]. 自然科学进展, 2009,19(1):1-15.

[2] Schafer J B, Konstan J, Riedl J. Recommender system in e-commerce[C]. Proceedings of the first ACM conference on electronic commerce. 1999, 158-166

[3] 贺桂和. 基于用户偏好挖掘的电子商务协同过滤推荐算法研究[J]. 情报科学, 2013,31(12):38-42.

[4] 卓勇霖. 推荐系统中近邻算法与矩阵分解算法效果的比较——基于 Movielens 数据集[D]. 南开大学, 2012.

[5] J. Allan, J. Carbonell, G. Doddington, J. Yamron, and Y. Yang, Topic detection and tracking pilot study final report, Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop, 1998, pp. 194-218.

[6] Shafer, Sen S W, Frankowski, et al. “Collaborative Filtering Recommender Systems”[C]//International Conference on Intelligent Systems Design & Applications. IEEE, 2007:438-443.

[7] 胡为. 一种基于社交网络的协同过滤推荐算法研究[D]. 湖南大学, 2015.

[8] Anand S S, Bell D A, Hughes J G. Evidence based discovery of knowledge in databases[C]// Knowledge Discovery in Databases, Iee Colloquium on. IET, 2002:9/1-9/4.

[9] 王小亮. 基于协同过滤的个性化推荐算法的优化和应用[D]. 浙江工商大学, 2010.

[10] 刘丹, 褚蓓蓓, 郑丽娟. 基于协同过滤的个性化推荐算法研究与实践[J]. 石家庄铁路职业技术学院学报, 2008, 7(2):43-48.

[11] 张明敏. 基于 Spark 平台的协同过滤推荐算法的研究与实现[D]. 南京理工大学, 2015.

[12] Lemire D, Maclachlan A. Slope One Predictors for Online Rating-Based Collaborative Filtering[J]. Computer Science, 2007:21—23.

[13] 孙丽梅, 李晶皎, 孙焕良. 基于动态 k 近邻的 SlopeOne 协同过滤推荐算法[J]. 计算机科学与探索, 2011, 05(9):857-864.

[14] 王鹏. 基于矩阵分解的推荐系统算法研究[D]. 北京交通大学, 2015.

[15] 张光前,雷彩华,吕晓敏. 电子商务推荐的研究现状及其发展前景[J]. 情报杂志, 2011,30(12):60-65.

[16] 于洪,李俊华. 一种解决新项目冷启动问题的推荐算法[J]. 软件学报, 2015,26(6):1395-1408.

[17] 刘鑫钱,松荣. 时间元兴趣度度量方法和扩展 VSM 用户兴趣模型研究[J]. 小型微型计算机系统,2011,32(4):708-712.

[18] 陈志敏,姜艺. 综合项目评分和属性的个性化推荐算法[J]. 微电子学与计算机, 2011, 28(9):186-189.

[19] 黄国言,李有超,高建培等. 基于项目属性的用户聚类协同过滤推荐算法[J]. 计算机工程与设计,2010,31(5):1038-1041.
[20] Perronnin, Florent, and Christopher Dance. Fisher kernels on visual vocabularies for image categorization. Computer Vision and Pattern Recognition, 2007. CVPR07. IEEE Conference on.IEEE, 2007:169-185
[21] Gordo, Albert. Supervised mid-level features for word image representation. ArXiv preprint arXiv:1410.5224 .2014:19-22
[22] Yao, Cong, etal. Strokelets: A learned multi-scale representation for scene text recognition. Computer Vision and Pattern Recognition, 2014 IEEE Conference on. IEEE,2014: 437-442
[23] Jaderberg, Max, Andrea Vedaldi, and Andrew Zisserman. Deep features for text spotting.Computer Vision–ECCV 2014. Springer International Publishing. 512-528 ,2014: 569-571
[24] Jaderberg, Max, et al. Deep structured output learning for unconstrained text recognition. arXiv preprint arXiv:1412.5903 ,2014:389-392
[25] Singh, Saurabh, Abhinav Gupta, and Alexei Efros. Unsupervised discovery of mid-level discriminative patches. Computer Vision–ECCV 2012,2012:73-89
[26] Zhang Xiang, Yann LeCun. Text Understanding from Scratch. arXiv preprint arXiv:1502.01710 ,2015.
[27] 杨莹, 张海仙. 基于卷积神经网络的图像分类研究[J]. 现代计算机, 2016(5):67-71.
[28] 闫蕾芳. 基于深度学习的图像分类的研究[D]. 山东大学, 2017.
[29] 辛晨. 基于深度学习的图像分类及应用研究[D]. 中国科学院大学(中国科学院遥感与数字地球研究所),2017.
[30] 荆涛, 王仲. 光学字符识别技术与展望[J]. 计算机工程, 2003, 29(2):1-2.
[31] Liu F, Picard R W. Finding Periodicity in Space and Time[C]// International Conference on Computer Vision.IEEE, 1998:376-383.
[32] 张翠平, 苏光大. 人脸识别技术综述[J]. 中国图象图形学报, 2000, 5(11):885-894.
[33] Phillips,P.J,Moon H,Rizvi,S.A,et al. The FERET evaluation methodology for face-recognition algorithms[J]. IEEE Transactions on Pattern Analysis and Machine Intellig ence, 2000, 22(10):1090-1104.
[34] Zhai Y, Wang X, Gan J, et al. Towards Practical Face Recognition: A Local Binary Pattern Non Frontal FacesFiltering Approach[C]// Chinese Conference on Biometric Recognition. Springer International Publishing,2015:51-59.
[35] 陈肇雄, 高庆狮. 自然语言处理[J]. 计算机研究与发展, 1989(11):1-16.
[36] 张钹. 自然语言处理的计算模型[J]. 中文信息学报, 2007, 21(3):3-7.