

密级： 保密期限：

北京邮电大学

硕士学位论文



题目： 基于深度学习的排序模型的研究与实现

学 号： _____

姓 名： _____

专 业： 计算机科学与技术

导 师： _____

学 院： 网络技术研究院

2018 年 3 月 12 日



**BEIJING UNIVERSITY OF
POSTS AND
TELECOMMUNICATIONS**

Thesis for Master Degree

Topic: RESEARCH AND IMPLEMENTATION OF
RANKING MODEL
BASED ON DEEP LEARNING

Student No.: _____

Candidate: _____

Subject: Computer Science and Technology

Supervisor: _____

Institute: Institute of Network Technology

2018-3-12

基于深度学习的排序模型的研究与实现

摘 要

排序问题是信息检索和推荐系统等领域的核心问题之一。排序结果的精准度和合理性会直接影响检索和推荐的质量。因此，对排序模型设计和排序结果优化等相关问题的研究具有实用价值。传统的排序方法包括由人工确定排序打分计算公式、基于重要性排序和基于相关性排序等，这些方法存在一定的不足，例如：依赖专家经验、主观性较强，没有综合考虑排序的各项条件因素等。

近年来，深度学习成为学术研究的热点方向，取得了一系列瞩目的研究成果。与传统机器学习算法模型相比，深度神经网络模型对于复杂函数的拟合能力更强，同时也具备更强的特征抽取和表示学习能力。随着机器学习技术的发展，使用机器学习算法模型解决排序问题逐渐成为学术界和工业界的主流做法。

在此背景下，本文提出了一种基于深度学习的排序模型。论文第一章阐述了课题研究的背景及意义。第二章介绍了深度学习的背景理论和相关技术。第三章分析了目前常用的排序方法的基本原理和相关排序模型中存在的 key 问题。第四章介绍了论文的主要研究成果，本文利用 Boosting、Stacking 等多种模型融合方法，将深度神经网络、逻辑回归和梯度提升树三种基础算法模型进行组合，形成融合模型，并将得到的融合模型用于排序打分，提升了排序的精准度；排序模型采用分层结构设计，兼顾了排序的精准度和计算效率；提出了一种基于循环神经网络模型的历史行为数据编码方法，解决了已有方法存在的特征向量长度不固定、信息丢失等问题。第五章给出了模型的具体设计与实现，提出了基于排序模型的地铁单边交易处理方法，实验结果显示融合模型相比于其他算法模型具有更高的预测准确率，同时基于排序模型的地铁单边交易处理方法也比现有方法更加合理，验证了该排序模型的有效性和实用性。

关键词 排序学习 深度学习 深度神经网络 循环神经网络

RESEARCH AND IMPLEMENTATION OF RANKING MODEL BASED ON DEEP LEARNING

ABSTRACT

Ranking problem is one of the key issues in information retrieval and recommendation system. The accuracy and rationality of the ranking results directly affect the quality of retrieval and recommendation. Therefore, it is of great value to study the ranking model and the optimization method of ranking results. Traditional ranking methods are available including determined by artificial scoring formula, based on the importance and based on the correlation and so on. However, these methods have some disadvantages such as relying on experts' experience, the lack of objectivity and without considering the factors synthetically.

In recent years, deep learning has become a hot topic of academic research and has made a series of remarkable achievements. Compared with the traditional machine learning algorithm models, the deep neural network model has stronger fitting ability for complex functions and stronger feature extraction and presentation learning ability. With the development of machine learning technology, using machine learning algorithm model to solve the ranking problem has gradually become the mainstream practice in academia and industry.

In this context, a ranking model based on deep learning is proposed. The first chapter describes the background and significance of the research. The second chapter introduces the background theories and related technologies of deep learning. In the third chapter, this paper introduces the commonly used ranking methods, and analyzes the key problems existing in the ranking model. The fourth chapter describes the paper's main research results. Fusion methods such as Boosting and Stacking are applied to combining three kinds of algorithms, including the logistic regression, deep neural network and gradient boosting tree to a fusion model for scoring, and improve the ranking effect of

the model. The ranking model adopts the layered structure design, which takes into account the accuracy and efficiency of the ranking. This paper presents a method of data encoding of historical behaviors based on recurrent neural network model, which solves the disadvantages of the previous methods such as feature vector length is not fixed and the information loss problem. The fifth chapter gives the detailed design and implementation of the model, and proposes a subway unilateral transaction processing method based on the ranking model. The experimental results show that the fusion model has higher prediction accuracy compared with other algorithms. At the same time, the subway unilateral transaction processing method based on the ranking model is more reasonable than the existing methods, verifying the validity and practicability of this ranking model.

KEY WORDS learning to rank deep learning deep neural network
recurrent neural network

目 录

第一章 绪论.....	1
1.1 研究背景.....	1
1.2 研究意义.....	2
1.3 研究创新点总结.....	3
1.4 硕士研究生期间的工作总结.....	4
1.5 论文组织结构.....	4
第二章 背景理论及相关技术介绍.....	6
2.1 深度学习简介.....	6
2.1.1 深度学习概述.....	6
2.1.2 神经网络结构介绍.....	7
2.1.3 深度学习框架介绍.....	11
2.2 本章小结.....	13
第三章 常用排序方法的介绍与分析.....	14
3.1 常用排序方法介绍.....	14
3.2 排序学习模型训练方法.....	15
3.2.1 Pointwise 方法.....	16
3.2.2 Pairwise 方法.....	16
3.2.3 Listwise 方法.....	17
3.3 常用算法模型和评价指标.....	18
3.3.1 常用算法模型.....	19
3.3.2 常用评价指标.....	21
3.4 排序模型关键问题分析.....	23
3.4.1 训练集构造方法.....	23
3.4.2 历史行为数据处理方法.....	24
3.4.3 排序模型设计方法.....	27
3.5 本章小结.....	29
第四章 基于深度学习的排序模型关键问题研究.....	30
4.1 排序模型训练方法选择.....	30
4.2 排序打分算法模型设计.....	31
4.2.1 分类任务效果对比.....	31
4.2.2 回归任务效果对比.....	33
4.2.3 打分模型设计.....	36
4.3 损失函数及优化方法选择.....	38
4.3.1 损失函数.....	38
4.3.2 优化方法.....	40
4.4 基于循环神经网络的历史行为数据编码方法.....	42
4.5 本章小结.....	44
第五章 基于深度学习的排序模型的设计与实现.....	45
5.1 基于深度学习的排序模型设计.....	45

5.1.1 模型总体结构设计.....	45
5.1.2 模型训练模块.....	46
5.1.3 排序模块.....	48
5.2 排序模型在地铁单边交易处理中的应用.....	49
5.2.1 问题描述及分析.....	49
5.2.2 基于排序模型的地铁单边交易处理方法.....	50
5.2.3 实验设计及结果分析.....	52
5.3 本章小结.....	55
第六章 总结与展望.....	56
6.1 论文总结.....	56
6.2 工作展望.....	57
参考文献.....	58
附录.....	62
攻读学位期间发表的学术论文目录.....	63

第一章 绪论

1.1 研究背景

随着互联网的高速发展，计算机相关技术已经渗透入人们日常生活中。现在，人们可以使用搜索引擎获取各类信息知识，可以在电商平台进行网上购物，可以使用社交软件与亲友保持联系。得益于信息检索和推荐系统领域相关技术的发展进步，各类互联网服务正趋于个性化和智能化：搜索引擎会根据用户的搜索点击历史，为其展示个性化的搜索结果；新闻资讯类应用软件可以主动给用户推送其可能感兴趣的信息内容；电商平台会根据消费者的特点和消费习惯进行商品推荐；社交应用通过分析用户的人际关系，向其推荐可能认识的人，帮助用户结识新朋友、拓展人脉。在此背景下，如何提高检索和推荐结果的准确度，成为当前迫切需要解决的问题，针对这一问题的研究具有十分重要的现实意义。

排序问题是信息检索和推荐系统等领域的核心问题之一。例如，搜索引擎需要将网页搜索结果按照与用户的检索目的的符合程度进行排序；推荐系统需要把候选物品按照用户可能感兴趣的程度进行排序。排序结果的准确性和合理性会直接影响检索和推荐的质量。因此，排序模型设计和排序结果优化等问题具有很高的研究价值。针对排序问题，传统解决方案大多依赖于人工经验，由专家根据历史数据和待排序项的特征，通过组合一系列排序规则得到排序公式。随着对排序问题研究的不断深入，目前比较常用的做法是利用机器学习相关技术解决排序问题。与传统解决方案相比，基于机器学习的排序模型具有更高的计算效率和排序准确度，得到的排序结果也具有更强的客观性。

近年来，深度学习（Deep Learning, DL）作为学术领域的重点研究方向之一，取得了一系列显著的进展。例如，卷积神经网络（Convolutional Neural Network, CNN）在处理计算机视觉领域中的物体识别、图像分类等问题中，取得了明显优于以支持向量机（Support Vector Machine, SVM）为代表的传统算法模型的效果^[1]；循环神经网络（Recurrent Neural Network, RNN）则被广泛用于处理时间序列信息数据，特别是其变体：长短时记忆网络（Long Short-Term Memory, LSTM）和门控循环单元（Gated Recurrent Unit, GRU），通过特殊的结构设计解决了 RNN 中的长期依赖问题，在处理如机器翻译、语音识别、机器创作等任务中取得了令人惊喜的效果^[2]。众多研究成果表明，在训练数据充足的条件下，深度神经网络模型在处理如分类、回归等机器学习任务时，能够取得显著优于传统机器学习算法模型的效果。关于深度学习技术在排序问题中的应用方面，谷歌公司在 2016 年发表的论文中^[3]，提出 YouTube 的视频推荐

系统使用深度神经网络模型实现候选生成模块和排序模块。同时，谷歌公司还在其另一篇论文中提出了一种由广义线性模型和深度神经网络模型组合而成的混合模型结构，论文的实验部分将该模型用于 Google Play 商店的应用推荐^[4]，取得了良好的效果。

1.2 研究意义

如上所述，信息检索、推荐系统、在线广告等计算机技术应用方向已经广泛存在于人们日常生活的各类场景中，为用户提供了各种便捷的服务。而排序问题作为上述领域中的一类通用问题，对其进行研究，对于优化应用效果、改进服务质量、提高用户满意度具有重要的意义。

针对排序问题的研究，可以主要分为排序系统结构设计、排序打分算法模型设计优化和排序效果评价方法设计等方面。其中，排序系统的结构设计和排序打分算法模型的设计优化是研究的重点，也是难点问题。早期解决排序问题的方法主要是由专家根据业务场景特点，结合对历史数据的分析，制定一系列排序规则，或者是利用待排序项的各属性特征值确定出打分公式，根据候选项得分高低确定最终的排序结果。这种方法的主要缺点在于排序完全依赖于专家的经验，缺少客观性和说服力。同时，人工分析历史数据的效率十分低下，在数据海量增长的当前环境下已变得不可行。随着研究的深入，研究者开始尝试使用机器学习相关技术手段来解决排序问题，并逐步发展为一个热点主题：排序学习（Learning to Rank, LTR）^[5]。与依靠专家的传统方法相比，机器学习算法模型对数据的分析更加全面，能够挖掘出数据中隐含的、人工难以发现的规律和性质。同时，得益于计算机硬件的飞速发展和分布式计算领域取得的一系列研究进展，如今计算机单机/集群的数据处理能力相比过去都得到了巨大的提升，这使得算法模型能够在大规模数据集上进行高效的训练。

排序学习的主要流程包括数据采集、特征抽取、模型训练、场景应用等步骤。其中，特征抽取和模型训练是最为关键的两个阶段，在很大程度上影响最终的排序结果。模型训练阶段的重点则在于算法模型的设计和优化。目前排序学习中广泛使用的机器学习算法模型包括逻辑回归（Logistic Regression, LR）、支持向量机、梯度提升树（Gradient Boosting Decision Tree, GBDT）、随机森林（Random Forest, RF）等。近几年来，深度学习成为广大学术研究者关注的热点问题，各种结构的深度神经网络模型不断涌现。然而深度学习的成功实践主要集中在计算机视觉和自然语言处理等领域，针对深度学习技术在排序问题中的应用的相关研究较少，发展空间很大。此外，在样本数据特征抽取方面，用户行为特征是一种广泛存在于各类应用场景中的特征数据，例如，在搜索引擎中，用户具有历史查询记录和浏览行为记录；在推荐系统中，用户

具有资讯阅读、商品浏览/购买等行为记录。此类数据的共同特点是具有较强的时序特性和前后关联性，且能在一定程度上反应用户的行为特点和偏好习惯。因此，研究用户行为数据处理方法十分重要，也是特征抽取步骤需要解决的重点问题之一。

本课题的目标是研究一种基于深度学习的排序模型。为了提高排序结果的准确度，本课题将深度神经网络结构融入排序模型，弥补了现有算法模型存在的不足之处。此外，经过对数据特征的充分理解和分析，本课题提出了一种使用循环神经网络模型对用户历史行为数据进行编码的数据处理方法，用训练得到的网络参数表示用户历史行为特征，并作为输入特征向量加入排序模型参与训练。为了验证所提出排序模型的可行性和有效性，本课题将该排序模型应用于解决面向地铁交易数据的单边交易处理问题。地铁单边交易指进出站信息存在缺失的交易记录，目前对此类交易记录的处理方式通常是按照最长距离计算票价，这种方法缺乏合理性，容易造成用户损失而导致发生纠纷和投诉。本课题利用所提出的排序模型，根据用户的历史出行特征和地铁站点特征，对各候选站点按照其预估概率进行排序，从而进行合理计费。该方法有助于提升地铁运营的服务质量，提高用户满意度。同时，本课题所提出的排序模型也可以用于完成信息检索、推荐系统和在线广告等应用场景中的排序任务，具有较强的通用性和可扩展性。综上所述，本课题具有十分重要的研究意义和极为广阔的应用前景。

1.3 研究创新点总结

本课题研究的主要创新点总结如下：

(1) 设计并实现了包含深度神经网络结构的融合排序打分模型，优化了排序效果。本课题利用 **Boosting**、**Stacking** 等多种模型融合方法，设计了包含深度神经网络模型、广义线性模型和梯度提升树模型的融合排序打分模型。经过实验测试，相比于单一算法模型，融合模型具有更高的预测准确率和排序精准度。

(2) 提出了一种基于循环神经网络的历史行为数据编码表示方法。如何有效利用历史行为数据，从中提取出供算法模型学习的有价值的特征数据是许多机器学习任务中需要解决的重点难点问题之一。本课题提出的基于循环神经网络的历史行为数据编码表示方法，解决了现有方法存在的特征向量长度不固定和信息丢失等问题。

(3) 设计了分层结构的排序打分模型，综合考虑了模型的排序效果和计算复杂度，实现了排序精准度和计算效率的平衡，增强了排序模型的实际可用性。同时，采用参数控制中间输出候选项数量，使模型结构更加灵活。

(4) 将所提出的排序模型应用于地铁单边交易处理问题，减少了计费偏差，提高了计费策略的合理性。

1.4 硕士研究生期间的工作总结

本人于 2015 年 9 月进入北京邮电大学网络技术研究院的网络与交换国家重点实验室交换与智能控制研究中心学习。研究生学习期间，本人主要参与了一项国家 863 课题项目“大数据分析技术在输变电设备状态评估中的研究与应用”和一项校企合作项目“基于移动支付的新型绿色出行规划系统的设计与实现”的相关工作，负责项目中算法模型的研究、设计和实现，具体内容包括：

(1) 研究关联规则挖掘相关算法，设计并实现了用于量化关联规则挖掘的并行粒子群优化算法。该方法应用在 863 课题项目开发的电网大数据分析平台中，用于挖掘电网设备缺陷的发生与可能诱发因素之间存在的隐含关联关系。

(2) 基于深度学习的排序模型的研究与实现。通过对深度学习相关理论技术进行学习研究，设计了一种包含深度神经网络结构的排序打分模型，同时还提出了一种基于循环神经网络的历史行为数据编码方法。提出了基于该排序模型的地铁单边交易记录处理方法，针对上下车站点信息存在缺失的交易数据，利用排序模型对各站点的概率分布进行预测，从而协助系统制定更加合理的计费方案。

此外，本人还调研了分布式计算技术、搭建了实验室的 Hadoop 和 Spark 集群，并参与多项技术解决方案和设计文档的撰写工作。本论文的内容主要围绕上述第二点工作展开，对基于深度学习的排序模型的设计方法和实现细节进行了介绍。

1.5 论文组织结构

本论文的组织结构如下表所示：

表 1-1 论文组织结构表

第一章	绪论：研究背景及意义，创新点和工作总结，论文结构概述
第二章	论文背景理论及相关技术介绍
第三章	常用排序方法的介绍与分析
第四章	基于深度学习的排序模型关键问题研究
第五章	基于深度学习的排序模型的设计与实现
第六章	总结与展望
附录	包含参考文献、致谢、攻读学位期间发表的学术论文等

第一章：主要介绍了本课题提出的研究背景、研究意义和主要创新点。同时对本人在研究生阶段参与的相关工作内容进行了总结，并对论文的整体组织结构进行了梳理和总结。

第二章：主要介绍了论文所涉及的背景理论和相关技术。具体包括深度学习的概

念、发展过程、与传统算法模型的区别，常见神经网络模型的结构、特点和适用场景，以及常用的开源深度学习框架等内容。

第三章：主要介绍了目前常用的排序方法。针对排序学习问题，分析了 **Pointwise**、**Pairwise** 和 **Listwise** 三种典型的模型训练方法的原理和特点，介绍了常用的打分算法模型及评价指标。最后，从训练集构造、历史行为数据处理和模型设计三个方面分析了排序模型中的重点难点问题，为第四章介绍课题相关研究工作及成果做了铺垫。

第四章：针对第三章提出的关键问题，介绍了本课题的相关研究成果及给出的解决方案。本章是论文的重点论述部分，涉及内容包括排序模型训练方法的选择，排序打分算法模型的结构设计，神经网络模型损失函数和优化方法的选取，以及基于循环神经网络的历史行为数据处理方法等。

第五章：介绍了基于深度学习的排序模型的详细设计与实现方案，具体包括模型的总体结构设计，模型训练模块和排序模块的功能设计和主要流程等。同时，论文将排序模型应用于地铁单边交易处理问题，通过实验对排序模型的预测准确率和方法的有效性进行了测试和分析。

第六章：总结了论文的主要工作和创新点，同时展望了下一步的工作，提出了可优化的方向和可能的改进方案。

第二章 背景理论及相关技术介绍

本章对深度学习相关技术进行了介绍，主要包括深度学习概述，典型神经网络结构和目前比较流行的深度学习框架三个部分。第一部分介绍了深度学习概念的提出及发展历程，同时将深度学习与传统机器学习算法进行了对比分析；第二部分重点介绍了目前应用最为广泛的三种神经网络结构：全连接深度神经网络、卷积神经网络和循环神经网络；第三部分介绍了 TensorFlow、Caffe 等当前比较流行的深度学习框架，分析说明了每种框架的优缺点。

2.1 深度学习简介

2.1.1 深度学习概述

2006 年，机器学习领域顶级专家、加拿大多伦多大学教授 Geoffrey Hinton 等人第一次提出了“深度学习”的概念，指基于样本数据并通过一定的训练方法得到具有多层网络结构的深度神经网络模型的机器学习过程^[6]。

早期神经网络模型也称为感知机（Perceptron），其模型结构包括输入层、一层隐含层和输出层。特征向量由输入层输入，经过隐含层的变换操作，最后由输出层输出模型预测结果^[7]。早期感知机模型存在一个严重的问题，即无法模拟复杂的函数，甚至不能表示异或计算。随着技术的发展，Hinton、LeCun 等人于 20 世纪 80 年代提出了多层感知机（Multilayer Perceptron, MLP）的概念^[8]，解决了这一问题。多层感知机是网络结构中包含多个隐含层的感知机，通过增加非线性操作的次数实现对复杂函数的拟合。多层感知机解决了感知机模型无法表示异或逻辑的问题，提高了模型的表达能力，但同时也带来了网络参数求解易取得局部最优解、梯度消失等严重的问题。另外，随着网络层数的加深，权值参数的数量成倍增长，受限于当时的计算能力水平，训练神经网络模型十分困难。直到 2006 年，Hinton 及其学生 Ruslan 在世界顶级学术期刊《科学》上发表论文^[6]，提出了两个主要观点：第一，含有多个隐含层的神经网络模型具备很强的特征表示学习能力，深度模型学习获得的特征数据能够从更加本质的角度代表原数据；第二，针对深度神经网络模型训练不易取得全局最优解的问题，可以使用逐层训练的方法解决。这一研究成果引发了深度学习技术在学术研究和行业应用中的飞速发展，由此，深度学习逐步发展为机器学习的一个新领域。近年来，由 Google 旗下的人工智能公司 DeepMind 开发的围棋人工智能程序 AlphaGo 在各项比赛中战胜了多位人类世界冠军；百度公司开发的智能机器人小度，在人脸识别比赛中战胜了此前的人类冠军，这些成绩充分证明了深度学习的强大能力，同时也让普通民

众直观感受到了深度学习和人工智能技术，促进了深度学习的进一步发展。

深度学习算法模型与逻辑回归模型、支持向量机以及决策树类算法等传统机器学习算法模型相比，主要区别体现在深度学习模型的网络结构包含更多更深的层级，并且明确强调特征表示学习的重要性。深度学习模型通常包含 5 层以上的隐含层，多隐含层的结构设计增加了非线性操作的次数，从而可以使用较少的参数拟合高度非线性函数，这也是深度学习的“深度”所在。在特征提取方面，浅层学习算法模型主要依靠人工经验提取训练样本的特征，如使用核函数或手工编码等方法。这些方法不仅要求模型的使用者需要具备一定的专业知识，而且特征提取过程需要进行大量的计算，对计算资源的要求较高。同时，模型只能学习得到单层特征，而无法表示特征之间的层次结构；而在深度学习中，原始输入信号在每层网络中进行特征变换，实现了训练样本由原特征空间到新特征空间的变换映射，从而自动学习出样本特征的层次化表示。多项研究成果表明，在自然语言处理、计算机视觉和语音识别等领域的相关问题中，深度学习模型取得的效果已经远远超过此前的相关技术。例如，Google 和 Microsoft 的研究学者们使用深度神经网络模型，把语音识别问题的识别错误率降低了 20%-30%^[9]；在计算机视觉领域图像识别应用的著名项目 ImageNet 中，深度神经网络模型将原来的错误率降低了 9%。此外，Google、Facebook 等公司也开始将深度神经网络模型应用在推荐系统、广告点击率预估等业务场景中，并取得了一定的成效。

2.1.2 神经网络结构介绍

如上所述，在神经网络技术的发展历程中，为了针对性的解决其在各类应用中所面临的问题，研究人员设计发明了多种网络结构。本节对其中三种最基础，同时也是应用最广泛的网络结构进行介绍，具体包括深度神经网络（Deep Neural Network, DNN）、卷积神经网络和循环神经网络。

● 深度神经网络

目前学术研究和行业应用中所使用的深度神经网络模型通常是多种神经网络结构的融合，包括全连接层、卷积层和循环神经网络单元等。与之不同，本节所介绍的深度神经网络具体指具有全连接结构的深度神经网络模型，也可以称为多层感知机。全连接深度神经网络的结构如图 2-1 所示。

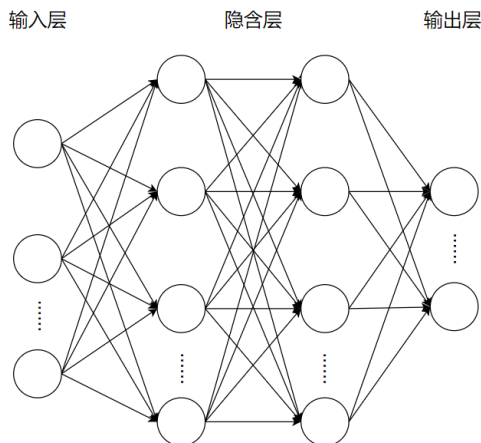


图 2-1 全连接深度神经网络结构

传统神经网络模型通常使用 Sigmoid 或双曲正切等函数作为激励函数模拟神经元对激励信号的响应，并使用由 Werbos 提出的反向传播（Backpropagation, BP）算法进行训练^[10]。在神经网络模型中，隐含层的数量在很大程度上影响模型对复杂函数的拟合能力，然而随着隐含层数量的增加，优化方法在求解过程中变得十分容易取得局部最优解，同时局部最优解与全局最优解的偏差也会逐渐增大。此外，传统神经网络模型还存在一个严重的问题，即当隐含层数量增加时，梯度消失问题会变得愈加突出。具体来说，当使用 Sigmoid 函数作为网络单元的激活函数时，在使用反向传播算法计算梯度时，每传播一层，反馈信号会衰减为上一层的四分之一。在这种情况下，当神经网络模型包含多个隐含层时，就会导致低层的网络由于梯度传播的指数衰减而接收不到有效的反馈信号，无法完成模型的训练。

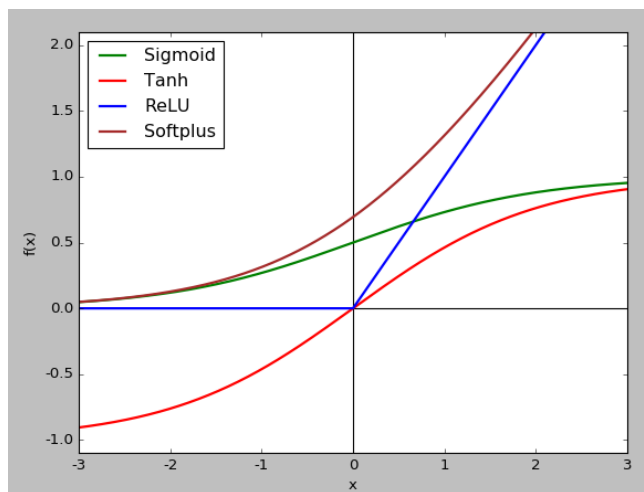


图 2-2 四种激活函数对比

为了解决神经网络模型训练容易取得局部最优解的问题，Hinton 在其 2006 年发表的论文中使用了通过无监督预训练过程来优化模型参数初始值的方法，成功将网络的隐含层数量扩展到了 7 层。具体而言，预训练过程基于贪心思想逐层训练模型，将每一层训练得到的参数设定为后一层的初始权值，最终再进行权值微调。同时，为了

克服梯度消失问题，研究人员提出使用 ReLU、Softplus、Maxout 等函数替代 Sigmoid 函数作为神经元的激活函数，并逐步形成了当前深度神经网络的基本结构。如图 2-2 所示，与 Sigmoid 和 Tanh 函数相比，ReLU 等函数具有单侧抑制、相对宽阔的兴奋边界和稀疏激活性等特点^[11]，能够有效解决神经网络模型存在的训练速度慢、容易过拟合等问题。

● 卷积神经网络

在全连接神经网络结构中，相邻两层的网络单元都互相连接，由此产生的一个严重问题是参数个数的爆炸增长。举例来说，假设以一幅分辨率为 1000×1000 的图片作为神经网络模型的输入，当使用全连接结构的神经网络模型时，每层隐含层含有 10^6 个神经单元，对应 10^{12} 个网络参数^[12]。训练包含如此巨大数量的参数的神经网络模型，不仅需要耗费大量的时间，同时还会导致过拟合、容易取到局部最优解等严重问题。

上世纪 60 年代，Hubel 和 Wiesel 在对猫的脑皮层中用于局部敏感和方向选择的神经元进行研究时，发现其所具有的特殊结构能够显著降低反馈神经网络的复杂度^[13]。受到这一研究成果的启示，研究人员继而提出了卷积神经网络的概念^[14]。卷积神经网络具有两个关键的特性：局部感受野和权重参数共享。区别于全连接的神经网络结构，卷积神经网络模型的结构中具有一个由卷积层和子采样层所组成的特征提取器。在卷积层中，一个网络单元只与邻层的一部分网络单元相连，即局部感受野。卷积神经网络的一个卷积层通常由多个特征平面构成，其中每个特征平面由一系列呈矩形排列的网络单元构成，属于相同特征平面的网络单元具有相同的权值参数，即卷积核^[15,16]。子采样，又被称为池化（Pooling），是在特征向量上进行的一种聚合操作，通常包括均值子采样和最大值子采样两种形式。局部感受野和权重参数共享在很大程度上缩减了神经网络中的参数规模，使模型更加简单，不易产生过拟合等问题。

卷积神经网络模型由于其自身结构特点，非常适用于处理图像特征，因此被广泛应用在计算机视觉领域中的图片分类、图像分割等问题中，并且取得了良好的效果。随着相关研究的深入，卷积神经网络衍生出了众多网络结构，其中比较著名的包括 1986 年提出的 Lenet、2012 年提出的 Alexnet、2014 年提出的 GoogleNet 和 VGG、2015 年提出的 ResNet 以及 2017 年最新提出的 DenseNet 等。

● 循环神经网络

全连接结构的深度神经网络模型还具有一个严重问题，即无法表示数据的时序关联特性。然而在机器翻译、语音识别、趋势预测等应用中，样本出现的先后顺序及上下文关联性是十分重要的。在此背景下，研究人员设计了另一种网络模型结构：循环神经网络。在循环神经网络中，序列在某一时间点的输出不只由此时的输入所决定，还受到之前一段时间的输出的影响。具体表现为神经网络单元会对该时刻之前的网络

状态进行记忆，并参与到对该时刻输出的计算过程中。如图 2-3 所示，区别于普通神经网络中同层网络单元之间不存在连接的结构，在循环神经网络中，同层网络单元之间存在连接，同时将当前时刻输入层和隐含层上一时刻的结果共同作为当前时刻隐含层的输入。

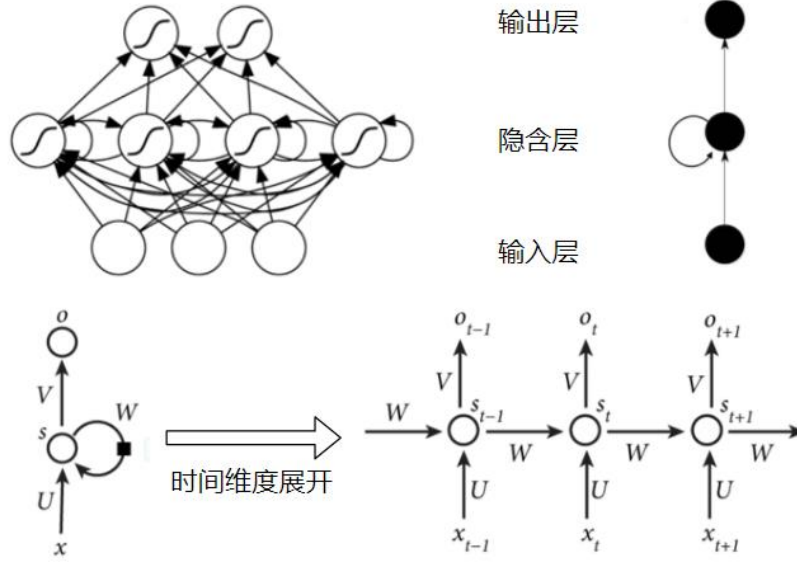


图 2-3 循环神经网络典型结构及在时间维度上的展开

与其他神经网络模型相同，循环神经网络的训练过程也利用了反向传播算法，区别之处在于如果对循环神经网络进行时间维度上的展开，其参数是共享的，同时梯度误差不仅依赖当前网络的状态，还依赖于此前网络的状态。针对循环神经网络的学习算法称为随时间反向传播算法 (Backpropagation Through Time, BPTT)^[17]。然而 BPTT 算法存在的一个严重问题是其无法解决长时依赖问题，即当此刻的输出与之前较长一段时间的数据相关时，使用该算法无法对网络进行有效的训练。这是因为当循环神经网络在时间维度上展开的层数过多时，BPTT 同样会产生普通神经网络使用 BP 算法训练时存在的梯度消失或梯度爆炸问题。为了解决梯度在时间维度上无法进行有效传播的问题，研究者们提出了多种改进的循环神经网络结构，如长短时记忆单元 LSTM 和门控循环单元 GRU。

1997 年，Hochreiter 和 Schmidhuber 提出了 LSTM 单元结构^[18]。LSTM 使用被称为 Cell 的结构来保存当前输入之前的状态，并使用精心设计的称为门 (Gate) 的结构来增加或去除 Cell 中的信息。如图 2-4 所示，LSTM 单元中共包括三个门结构，分别为输入门 (Input Gate)、输出门 (Output Gate) 和忘记门 (Forget Gate)。多项研究成果已经证明 LSTM 网络结构在解决长序列依赖问题中能够取得十分显著的效果，在自然语言处理中的词向量表达、词性标注、机器翻译和语音识别及图像描述生成等应用中都取得了显著的效果。2014 年，Cho 等人提出了 LSTM 单元的一种变化形式：GRU^[19]。GRU 将 LSTM 中的忘记门和输入门组合成了单一的更新门，使模型结构更

加精简，同样得到了广泛的应用。除 GRU 之外，还存在多种 LSTM 变体，如 2014 年由 Koutnik 等人提出的 Clockwork RNN^[20]，2015 年由 Yao 等人提出的 Depth Gated RNN^[21]等。

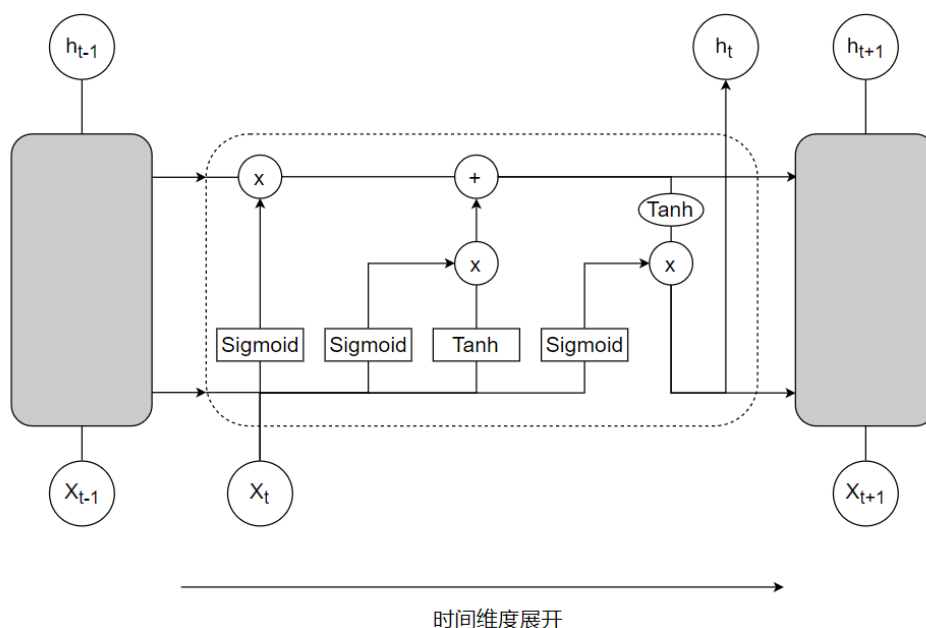


图 2-4 LSTM 单元结构

2.1.3 深度学习框架介绍

深度学习的火热发展及其取得的众多瞩目成绩，吸引了谷歌、微软等众多科技巨头公司针对其建设了一批重点项目，并因此推动了一系列开源框架的诞生和发展。目前在学术和工业界比较流行的深度学习框架主要包括 TensorFlow、Theano、Torch、Caffe 和 Keras 等。这些深度学习框架各具特点，在计算机视觉、语音识别和自然语言处理等领域的各项应用中取得了十分优秀的成果。下面对这些框架进行简要介绍。

● TensorFlow

TensorFlow 由谷歌公司的 Google Brain 团队开发，基于 DistBelief，面向机器学习任务和深度学习相关研究，目前已经开源。作为一款数学计算软件，TensorFlow 使用数据流图表示计算过程。在数据流图中，图的节点表示具体的数据计算操作，边表示以多维数组形式存储的数据。TensorFlow 的名称取自于其运行机制：张量(tensor)从数据流图的一端流动(flow)到另一端的计算过程^[22]。

TensorFlow 的主要特点包括适应性强、可扩展性强、自动差分、计算高效等。具体表现在 TensorFlow 不仅提供神经网络库，还支持使用者自主设计流图、编写相关代码实现计算过程；能够部署在各种类型的终端上，如图形处理器、移动设备、云主机等；提供包括 Python 和 C++在内的多种编程语言的接口支持；TensorFlow 能够实现计算任务分发，能够充分利用硬件资源。

- Torch

Torch 是一个已有十余年历史的科学计算库，采用 Lua 语言编写，支持大量机器学习算法。Facebook 开源了许多建立在 Torch 上的深度学习组件和拓展功能，促进了 Torch 的进一步发展^[23]。

Torch 框架的优点在于使用其搭建神经网络模型非常简单，易于使用。框架高度模块化，同时提供良好的图形处理器支持。Torch 框架提供跨平台支持，能够嵌入到运行 Android 和 iOS 系统的移动设备中。

- Caffe

Caffe 全称 Convolutional Architecture for Fast Feature Embedding，是一个由美国加州大学伯克利分校的贾扬清博士开发的开源深度学习框架，现在由伯克利视觉学习中心进行维护^[24]。Caffe 中的计算是以层的形式表示的，其中每一层的基本流程为接受输入数据，进行计算，输出计算结果。其中计算包括由输入计算输出的前向传播和梯度的反向传播。定义好每一层的计算函数后，即可将多个层连接成网络进行训练。

Caffe 的优点包括计算速度快，可用于海量数据的训练，特别适用于计算机视觉领域；开放性好，提供良好的社区支持。不足之处则包括文档匮乏，需要大量依赖包、安装过于复杂等。

- Theano

Theano 由加拿大蒙特利尔理工学院开发，其设计目标为解决深度学习任务中神经网络模型的大量计算需求^[25]。在其发展过程中，衍生了许多深度学习工具包，如 Blocks 和 Keras 等。

Theano 的优势体现在其集成 NumPy 并使用其中的 ndarray 数据类型、使用 GPU 加速计算、提供速度和稳定性优化、可以动态生成 C 代码、具备较强的灵活性等。而缺点则包括偏向底层，模型实现比较复杂，不支持多图形处理器等。

- Keras

Keras 是一个完全使用 Python 语言编写而成的高层神经网络工具库，其设计目标是为快速科学实验提供支持，方便进行简便高效的模型设计，十分适合用于先进的研究工作^[26]。

Keras 提供简洁一致的应用程序编程结构，可以大大减少工作量，用户体验优秀；模块性强，网络单元、代价函数、优化方法、激励函数等都作为独立模块，便于构建模型；支持以编写类或函数的方式创建新模块，具备很强的扩展性。

经过对上述各框架各自优缺点的分析对比，并充分考虑易用性、可扩展性及计算效率等因素，本课题最终选择使用 TensorFlow 实现课题研究中所涉及到的深度神经网络模型。

2.2 本章小结

本章主要介绍了深度学习的相关概念技术，包括深度学习的发展历程，经典神经网络模型的结构等，并对当前主流的深度学习框架进行了简要介绍和对比分析，为后续章节中涉及的算法设计、模型选取、模型实现等内容做了铺垫。

第三章 常用排序方法的介绍与分析

本章首先对现有的各种排序方法进行了介绍，并分析了不同方法各自的特点。之后总结了排序模型中常用来拟合排序打分函数的算法模型及各自的优缺点，并介绍了三种常用的评价指标。最后概括分析了排序问题中的重点难点问题，包括特征数据的提取，历史行为特征数据的处理，排序模型的设计和优化等内容。在后续章节中将具体介绍本课题解决上述问题所采用的技术方案，并针对地铁单边交易处理的应用场景进行具体排序模型的设计与实现。

3.1 常用排序方法介绍

排序问题最初是作为信息检索领域的核心研究问题而被人们所关注。在搜索引擎等信息检索系统中，排序问题可以被抽象定义为给定查询 Q 和文档集合 $S=\{D_1, D_2, \dots, D_n\}$ ，按照某种规则返回 S 中文档的一个序列。传统的排序模型大致可以分成两种类型：相关度排序模型和重要性排序模型。其中，相关度排序模型根据文档 D 与查询 Q 之间的相关程度对候选文档进行排序，通常做法是利用词频、逆文档频率、文档长度等因子，由人工确定排序公式，此类模型包括 TF-IDF、隐语义分析（Latent Semantic Analysis, LSA）、BM25、基于语言建模的信息检索模型等。与之相对，重要性模型则不考虑文档和查询之间的相似性，而是将包含查询信息的文档按照其重要程度进行排序。文档的重要程度通常根据文档之间的图结构来计算，典型算法如 PageRank、HITS、TrustRank 等^[27-29]。

然而，无论是相关度排序模型还是重要性排序模型，都存在一个严重的问题，即只考虑了某一方面的因素（相关度或重要性），而没有对查询和文档的各项属性特征进行综合考虑。随着机器学习技术的发展，研究者们开始使用统计学习方法解决排序问题，即排序学习。如图 3-1 所示，排序学习将查询关键词、文档信息、甚至是传统模型的结果（如查询与文档的相似度、PageRank 值等）等数据作为输入特征向量，将由人工标注或从日志数据中挖掘出的文档排序列表作为输出，通过监督学习方法训练算法模型进行排序学习。与传统排序模型相比，排序学习训练得到的排序模型使用更多的特征拟合排序公式，充分考虑了各方面因素，同时由算法根据历史数据学习得到的模型参数，也比依赖人工的方法更加客观和高效。

现在，排序学习已经演化为一类独立问题，并且发展成为研究人员所关注的热点研究方向。随着相关研究的深入，排序学习方法已经不仅仅局限于信息检索领域，在推荐系统和计算广告等领域中，排序学习同样得到了广泛的应用，排序对象也从单一

的网页等文档对象扩展为包括商品、新闻资讯、广告内容等在内的各类实体。本课题将研究的排序问题由信息检索领域向外进行了推广扩展，具体定义为给定待排序候选项集合 $S1=\{I_1, I_2, \dots, I_n\}$ ，在条件集合 $S2=\{C_1, C_2, \dots, C_m\}$ 下，确定集合 $S1$ 中候选项的一个序列。其中，候选项 I 可以是网页、新闻、商品、音乐等存在排序需求的对象，条件 C 则包括搜索关键词、用户自身属性特征、历史数据记录等排序的条件因素。本文后续针对排序问题的相关研究，均基于此定义进行。

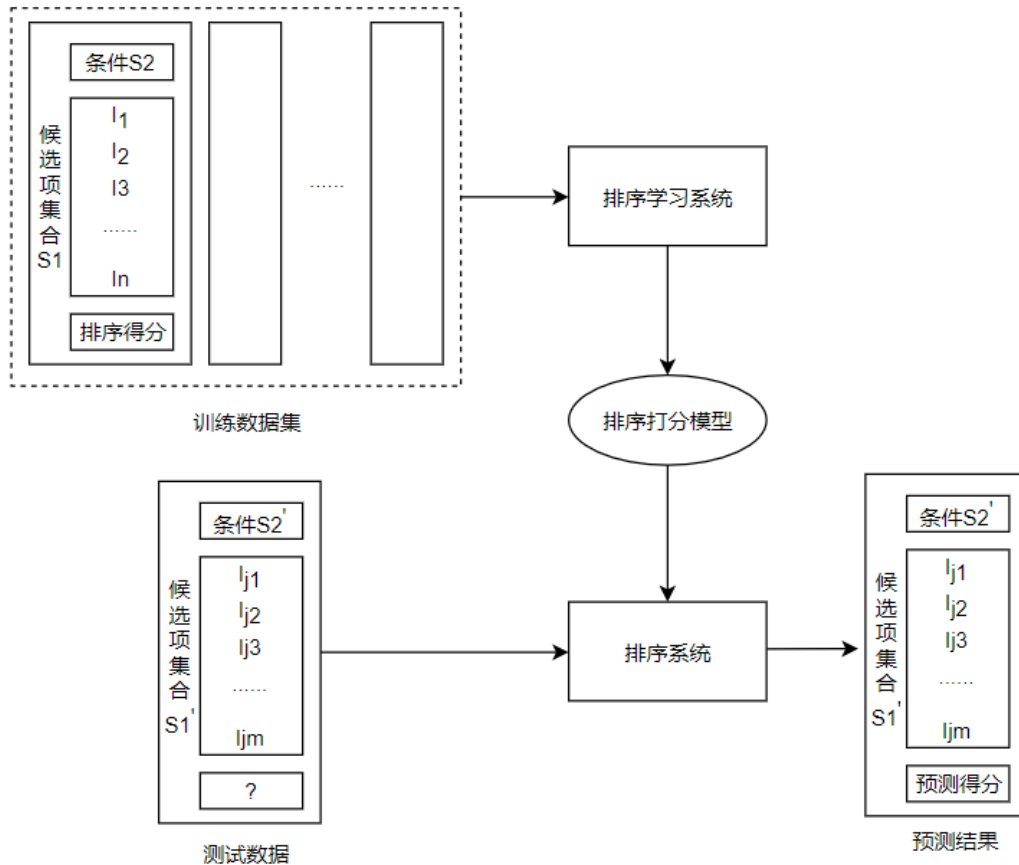


图 3-1 排序学习框架

3.2 排序学习模型训练方法

排序学习的主要流程包括训练数据收集、特征提取转换、排序打分模型训练优化和模型应用等步骤。其中，排序打分模型的设计和训练是排序学习问题的核心，也是技术难点所在。排序学习的模型训练方法按照原理的不同可以分成三种：**Pointwise** 方法、**Pairwise** 方法和 **Listwise** 方法。下面分别对三种方法的思想原理及特点进行分析说明。

3.2.1 Pointwise 方法

排序问题的核心是确定一个打分函数,使用此函数对待排序候选项集合中的每一项进行打分,并根据得分结果对候选项进行排序。**Pointwise** 方法是传统排序方法的延伸和发展。传统排序方法中由专家确定的排序公式、查询关键词与候选文档的相似度、PageRank 值等,都可以视为是打分函数的特殊形式。在 **Pointwise** 方法中,排序问题首先被转换成等价的分类、回归问题,再通过训练机器学习算法模型拟合打分函数^[30]。举例来说,假设排序特征向量为 $\mathbf{x} = (x_1, x_2, \dots, x_n)$, 待排序项的期望得分为 y , 使用线性回归模型拟合打分函数, 损失函数定义为平方误差, 则打分函数表示为:

$$f(\mathbf{x}) = \sum_{i=1}^n w_i x_i \quad (3-1)$$

$$\text{Loss} = \frac{1}{m} \sum_{j=1}^m (y_j - f(\mathbf{x}_j))^2 \quad (3-2)$$

其中, 参数 w_i 由模型经过训练确定, 训练的优化目标为使全部训练数据的损失函数数值之和最小。

Pointwise 方法是最简单也是应用最为广泛的排序学习模型训练方法。在 **Pointwise** 方法中, 训练数据集比较容易构造, 同时可以直接应用机器学习解决分类回归等问题中的一些比较成熟的技术手段和解决方案。此外, 在实际应用时, **Pointwise** 方法对各候选项的打分计算可以并行执行, 具有很高的效率。此方法的不足之处则在于独立地对每一个候选项进行处理, 而没有考虑待排序项之间的关联关系。

3.2.2 Pairwise 方法

与 **Pointwise** 方法相比, **Pairwise** 方法不是单独计算某一候选项的排序得分, 而是将分析的重点转向判断待排序项之间的相对顺序关系是否合理。**Pairwise** 方法的思想原理如图 3-2 所示, 其具体的训练目标为给定一组由两个待排序候选项 I_1, I_2 所构成的候选项对 $\langle I_1, I_2 \rangle$, 由模型判断这一组候选项对是否满足此种顺序关系。通过这种方式, **Pairwise** 方法把排序问题转化为二分类问题, 再使用 SVM、决策树等常用的分类算法模型求解分类判别函数。常用的 **Pairwise** 排序算法包括 GBRank、RankBoost 和 Ranking SVM 等^[31]。

Pairwise 方法分析了候选项之间的相互顺序关系, 通过判断候选项两两之间的顺序关系确定最终的排序结果, 相比于 **Pointwise** 方法做出了一定改进。然而, **Pairwise** 方法也存在明显的问题: 第一, **Pairwise** 方法只考虑了候选项间的相对位置顺序前后关系, 而没有考虑候选项在结果列表中的具体位置, 这导致 **Pairwise** 方法的优化目标与衡量实际排序结果优劣性的相关指标之间存在一定差距, 甚至成负相关关系。第二,

对于 N 个待排序候选项，Pointwise 方法共需要进行 N 次打分运算，而在 Pairwise 方法中则需要对共计 $N*(N-1)/2$ 个候选项对的先后顺序进行判断。这导致在 N 取值较大时，使用 Pairwise 方法的时间消耗会远远大于使用 Pointwise 方法，此外，对候选项对的顺序判断还可能出现排序结果之间存在冲突的情况，此时需要额外的策略规则来确定最终的排序结果。第三，在待排序项数量存在差异时，会导致机器学习系统的效果评价产生波动，而进一步影响模型的参数优化，导致模型的泛用性较差。

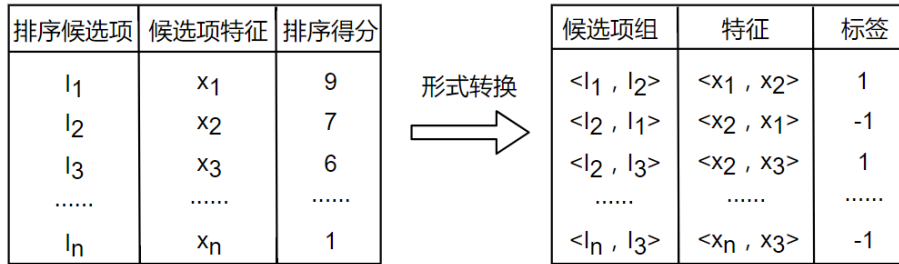


图 3-2 Pairwise 方法示意

3.2.3 Listwise 方法

Pointwise 方法和 Pairwise 方法本质上都是从候选项的角度考虑排序问题，分别以单个候选项和候选项对作为基本单位构造训练实例，通过将排序问题转变为分类/回归问题求解，最终再组合得到整体的排序结果。区别于这两种方法，Listwise 方法将一次排序结果的列表整体作为一个训练实例，并且直接对模型输出的候选项序列进行优化，使其接近真实序列^[32]。类似于 Pointwise 方法，Listwise 方法的学习目标同样是拟合一个排序打分函数，两种方法的不同之处在于 Pointwise 方法的优化目标是使候选项的打分尽可能接近其真实排序得分，而 Listwise 方法则是对每一个候选项打分后，利用 Luce 模型计算候选项集的每一种排列情况的概率值，之后再利用余弦相似度或 KL 距离等作为损失函数来衡量模型输出序列的概率分布与真实序列的排序概率分布之间的差距^[33]。如图 3-3 所示，假设 A、B、C 是三个待排序的候选项，其真实排序得分分别为 6、4、2。 $f(x)$ 与 $h(x)$ 是两个打分函数，对 A、B、C 三个候选项的打分分别为 3、2、1 和 4、6、2。三个候选项对应 $3! = 6$ 种不同的排列方式，使用 Luce 模型分别计算每个打分函数所对应的不同排列的概率分布，从图中可以看出函数 $f(x)$ 与真实打分函数 $g(x)$ 的排列概率分布更加接近，因此 $f(x)$ 比 $h(x)$ 更加接近最优函数 $g(x)$ 。

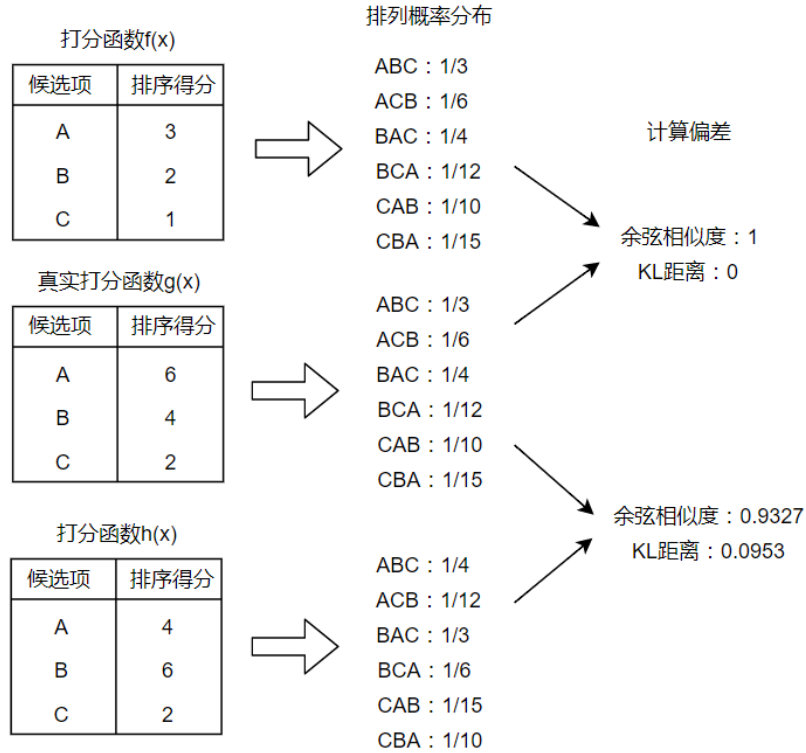


图 3-3 Listwise 方法示意

常用的 Listwise 排序算法包括 ListMLE、ListNet 和 AdaRank 等^[34]。与 Pointwise 方法和 Pairwise 方法相比，Listwise 方法的优点在于综合考虑了候选项在排序结果中的位置和各候选项之间的相对位置关系等因素。同时 Listwise 方法以候选项的序列作为模型输出，并直接针对排序结果整体进行优化，使其更加接近真实序列，从而有效避免了优化目标与实际衡量指标之间的差异问题。但是 Listwise 方法同样存在一些问题：首先，候选项的准确排序得分通常需要依靠人工标注获得，真实排序序列难以获取导致训练集的构造比较困难。比如在推荐系统中，通过日志数据只能获取用户的点击、购买等行为记录，而无法得到用户对各推荐物品的喜好顺序。其次，Listwise 方法需要针对每一种排列情况计算其出现的概率，对于 N 个候选项，算法的时间复杂度为 $O(N!)$ ，这使得当 N 取较大值时 Listwise 方法存在比较严重的性能瓶颈的问题。

3.3 常用算法模型和评价指标

排序学习算法通常使用线性函数来拟合打分函数，即通过对各项排序特征进行加权求和确定候选项的得分。常用的算法包括逻辑回归、分解机模型（Factorization Machine, FM）等。另一类常用模型是决策树类模型。梯度提升树模型，特别是由陈天奇博士改进实现的版本 XGBoost，凭借其优秀的效果和高效的性能，在多项机器学习比赛中表现亮眼。下面分别对这几种模型的原理及特点进行介绍。

3.3.1 常用算法模型

● 逻辑回归

逻辑回归是机器学习中的一种分类模型，具有简单高效的特点，被广泛应用于点击率预估等问题中。逻辑回归与线性回归同属于广义线性模型，其具体定义为给定输入特征向量 x 和参数 θ ，决策函数 $h(x)$ 为：

$$h_{\theta}(x) = g(\theta^T x) = \frac{1}{1 + e^{-\theta^T x}} \quad (3-3)$$

其中，函数 $g(z)$ 称为 Logistic 函数，或 Sigmoid 函数，形式如下：

$$g(z) = \frac{1}{1 + e^{-z}} \quad (3-4)$$

对应的输出 y 的概率分布满足：

$$\begin{aligned} P(y = 1|x; \theta) &= h_{\theta}(x) \\ P(y = 0|x; \theta) &= 1 - h_{\theta}(x) \end{aligned} \quad (3-5)$$

逻辑回归的损失函数为基于最大似然估计推导得到的对数损失函数，具体形式为：

$$Cost(h_{\theta}(x), y) = \begin{cases} -\log(h_{\theta}(x)) & y = 1 \\ -\log(1 - h_{\theta}(x)) & y = 0 \end{cases} \quad (3-6)$$

$$\begin{aligned} J(\theta) &= \frac{1}{m} \sum_{i=1}^n Cost(h_{\theta}(x_i), y_i) \\ &= -\frac{1}{m} \left[\sum_{i=1}^n y_i \log(h_{\theta}(x_i)) + (1 - y_i) \log(1 - h_{\theta}(x_i)) \right] \end{aligned} \quad (3-7)$$

逻辑回归通常被用来处理二分类问题，如预测某一事件发生的概率。在 Pointwise 方法中，排序问题可以被转化为分类问题，因此可以使用逻辑回归模型估计排序候选项的得分。

逻辑回归模型由于其结构简单，计算效率非常高，而且易于并行化，能够处理上亿维度的输入特征向量。与线性回归相同，逻辑回归算法也是对输入特征向量每一维度的线性加和，而没有对特征进行非线性组合的能力。这一问题导致逻辑回归算法依赖人工进行大量的特征工程工作来实现组合特征的提取。

● 分解机

分解机模型由日本大阪大学的研究学者 Steffen Rendle 于 2010 年提出^[35]，是一种基于矩阵分解、对于稀疏数据具有很强学习能力的机器学习算法模型。与逻辑回归模型相比，分解机模型在其公式中引入了组合项，使得其能够自动学习出交叉特征，模型表示为：

$$\begin{aligned}
\hat{y}(\mathbf{x}) &= w_0 + \sum_{i=1}^n w_i x_i + \sum_{i=1}^{n-1} \sum_{j=i+1}^n (\mathbf{v}_i^T \mathbf{v}_j) x_i x_j \\
\mathbf{v}_i &= (v_{i1}, v_{i2}, \dots, v_{ik})^T \in R^k, i = 1, 2, \dots, n \\
\hat{w}_{ij} &= \mathbf{v}_i^T \mathbf{v}_j = \sum_{l=1}^k v_{il} v_{jl}
\end{aligned} \tag{3-8}$$

从公式中可以看出，在计算组合项的权重系数时，分解机模型并不是直接对参数进行估计，而是引入了辅助向量 \mathbf{v} 进行计算。这是因为 N 维特征向量共对应 $N*(N-1)/2$ 个组合特征项，在特征向量高度稀疏时，大部分组合项的值为 0，这导致无法对参数 w_{ij} 进行有效的估计。当使用维度为 k 的辅助向量时，参数规模从 $N*(N-1)/2$ 个减少到 $k*N$ 个，从而大大提高了模型在特征稀疏的样本上的学习能力。同时，Rendle 在论文中给出了一种复杂度为 $O(N)$ 的计算二次项求和的算法，使得分解机模型同样具有较高的计算效率。分解机模型主要使用随机梯度下降法、交替最小二乘法和马尔科夫蒙特卡洛法等三种算法进行参数学习，各部分参数的梯度为：

$$\frac{\partial}{\partial \theta} \hat{y}(\mathbf{x}) = \begin{cases} 1 & \theta = w_0 \\ x_i & \theta = w_i \\ x_i \sum_{j=1}^n v_{jf} x_j - v_{if} x_i^2 & \theta = v_{if} \end{cases} \tag{3-9}$$

如上所述，分解机模型的优点在于与逻辑回归模型相比，其具有主动学习出交叉特征的能力，可以在一定程度上减少人工特征工程的工作量。与加入二次项的普通线性回归算法相比，其通过引入辅助向量，减少了参数数量，使模型具备对稀疏特征的学习能力。同时，与 SVD 等其他矩阵分解类算法相比，分解机模型无需将训练样本数据转化为类似 User-Item 的二维矩阵形式，输入更加灵活，模型更具一般性和扩展性。

● 梯度提升树

梯度提升树 (GBDT)，又称为 GBRT、MART、GBM 等，由 Friedman 提出^[36]，属于集成学习算法中的提升算法的一种。提升 (Boosting) 算法通常是一个迭代的过程，在新一代的训练过程中改进上一代的结果。在 AdaBoost 等传统提升算法中，每一步训练完成之后，会根据当前模型在训练样本上的预测情况对样例进行加权，即增加学习偏差大的样例的权重，减少学习偏差小的样例的权重^[37]，如此训练得到多个简单的分类/回归器，再使用加权或投票等方法组合这些简单模型，确定最后的预估结果。区别于这种训练方法，梯度提升算法以此前的预测结果与真实值之间的残差作为学习目标，通过在残差减少的梯度方向上训练新模型来逐步消除残差。GBDT 模型是梯度提升类算法的典型代表之一，使用分类回归树模型作为基学习器。与逻辑回归等线性模型相比，决策树类算法具有更强的特征非线性变换和特征组合的能力。同时因为树模型在分裂节点时会遍历每种特征计算最佳分割点，因此还可以用于有效特征筛

选。目前，GBDT 已被广泛应用于数据竞赛和工业生产中的各类问题中。

XGBoost^[38]是由陈天奇博士使用 C++ 语言编写开发的一个 GBDT 模型的改进版本。XGBoost 对传统 GBDT 模型做了部分改进，具体表现在 XGBoost 同时支持分类决策树模型和线性模型作为基学习器；求解过程中计算了损失函数的二阶泰勒展开，使用二阶导数实现更加快速的收敛；在损失函数中加入正则项，包括叶子节点个数和叶子节点输出预测值的 L2 范数模的平方和，降低了模型的复杂度，有效防止过拟合；节点分裂特征选择过程引入了随机森林中的列抽样方法，减少计算量，同时降低过拟合风险；对特征进行预排序并保存为 block 结构，避免了节点分裂时的重复计算，同时支持 CPU 多线程并行计算等。这些改进优化使 XGBoost 相比于传统 GBDT 模型具有更好的预测效果和更高的计算效率，因此，本课题在后续实验部分选用 XGBoost 模型进行对比实验。

3.3.2 常用评价指标

在机器学习问题中，研究者们设计了多种评价指标来衡量算法模型的效果。这些评价指标适用于不同的问题，需要按照任务类型进行选择。例如，分类问题常用的评价指标包括精度（Precision）、准确率（Accuracy）、召回率（Recall）、F1 值等，而回归问题中则通常使用平均绝对误差（Mean Absolute Error, MAE）、均方误差（Mean Square Error, MSE）、均方根误差（Root Mean Square Error, RMSE）等作为评价算法模型优劣的标准。在排序问题中，两种比较常用的评价标准是平均精度均值（Mean Average Precision, MAP）和归一化折损累积增益（Normalized Discounted Cumulative Gain, NDCG）。除此之外，本部分还会对 Listwise 方法中用于衡量预测序列与真实序列之间相似度的 KL 距离的概念进行简要介绍。

● 平均精度均值

MAP 基于精度和排序结果的位置信息计算得到。在信息检索领域中，精度定义为相关文档数量在返回结果文档总数中所占的比例^[39]，即：

$$\text{精度} = \frac{|\{\text{相关文档}\} \cap \{\text{检索文档}\}|}{|\{\text{检索文档}\}|} \quad (3-10)$$

可以看出，精度指标只考虑了检索结果中相关文档的个数，而没有考虑文档的位置因素和顺序关系^[40]。事实上，在搜索引擎和推荐系统等应用中，相关性越高的候选项应该处于序列中越靠前的位置上，因此研究者提出了平均精度的概念。平均精度的计算方法为首先计算出返回结果列表从开始位置到之后每一个位置上的精度，如果某一个位置上的候选项是不相关的则该位置的精度为零，最后对所有位置的精度取平均值。举例说明，假设某个查询 Q 应该对应 4 个相关网页，搜索引擎将这 4 个网页分别排在结果列表中的第 1、2、5、8 个位置上，则对于查询 Q，该搜索引擎的平均精

度均值为: $(1/1 + 2/2 + 3/5 + 4/8)/4=0.775$ 。

MAP 的计算方法对排序位置敏感, 但是仍没有考虑相关文档之间存在相关程度的差异这一关键因素。另一种评价指标 NDCG 针对这一问题进行了优化。

● 归一化折损累积增益

如上所述, NDCG 同时考虑了相关候选项在结果序列中的位置和其相关程度^[41]。NDCG 方法将序列每一个位置上的候选项的相关程度记为增益, 不相关的项的增益为零, 将这些增益进行加和, 便得到累积增益。为了引入位置因素, 将每个位置的增益除以一个递增的数值 (通常定义为该位置的对数值), 即折损值。对各位置折损后的增益值求和, 结果即为折损累积增益 (Discounted Cumulative Gain, DCG)。对于不同的查询, 结果的数量及相关性数值可能不同, 因此 DCG 之间不能进行直接比较, 需要进行归一化处理。具体做法为取 DCG 与理想 DCG 的比值, 其中理想 DCG 按照返回结果中的候选项按其增益值排序得到的序列计算得到。归一化处理后的指标称为 NDCG@K, 值域为 $[0, 1]$, 其中 K 表示结果列表的长度, 通常取 10、20、50 等数值。

$$N(n) = \underbrace{Z_n}_{\text{归一化}} \underbrace{\sum_{i=1}^n}_{\text{累积}} \underbrace{(2^{r(j)} - 1)}_{\text{增益}} / \underbrace{\log(1 + j)}_{\text{位置折损}}$$

图 3-5 一种 NDCG 指标计算公式

● KL 距离

KL 距离, 全称 Kullback-Leibler 差异, 也称为 KL 散度或相对熵, 是用来描述同一事件空间中的两个概率分布 P 和 Q 之间的差异的一种方法。KL 距离的物理意义为使用基于 Q 的编码对来自 P 的样本进行编码时, 平均每个样本所需要的额外的比特数。香农的信息论中使用 KL 距离来表示利用概率分布 Q 对真实分布 P 进行拟合时产生的信息损失^[42]。通常用 $D(P||Q)$ 表示 KL 距离, 对于离散概率分布和连续概率分布, 其计算公式分别为:

$$D(P||Q) = \sum_{x \in X} P(x) \log \frac{P(x)}{Q(x)} \quad (3-11)$$

$$D(P||Q) = \int P(x) \log \frac{P(x)}{Q(x)} dx \quad (3-12)$$

KL 距离具备非负性, 当且仅当概率分布 P、Q 相同时, 其 KL 距离等于 0。KL 距离虽然称为“距离”, 但事实上却不是一个距离度量函数, 因为其不满足对称性, 即 $D(P||Q) \neq D(Q||P)$, 同时也不满足三角不等式。KL 距离被广泛应用于信息检索和统计自然语言处理等领域中。例如在上文介绍的 Listwise 方法中, KL 散度被用于衡量预测序列的概率分布与真实序列概率分布之间的差异, 并作为优化目标指导模型参

数的更新。

3.4 排序模型关键问题分析

排序学习需要解决的关键问题大致可以分为三个方面：训练集构造方法、历史行为数据处理方法和排序模型设计方法。其中，训练集构造包括原始训练数据的获取、特征抽取及筛选等问题；历史行为数据处理方法包括历史行为特征向量的构造和编码等；模型设计方法则主要包括排序模型的总体结构设计、排序训练方法和打分算法模型的设计优化等。下面分别从这三个方面对课题研究的关键问题进行分析。

3.4.1 训练集构造方法

在排序问题中，样本数据通常有两种来源：人工标注数据集和通过挖掘日志数据得到的显式/隐式反馈。其中，人工标注方法的具体做法是给定排序的场景条件和待排序候选项集合，由专业的数据评估人员对每个候选项进行相关性判断，相关性评分可以是离散的分档评分也可以是连续的数值评分，最终将给出相关性评分的候选项作为训练数据。人工标注方法取决于数据标注人员的主观判断，受其背景知识和历史经验等因素的影响较大，存在一定的局限性。与之相比，通过挖掘日志数据获取训练数据是一种应用更为广泛的方法。



图 3-6 显式反馈举例：用户在亚马逊网站上对图书的评分

日志数据包括用户对网页的点击浏览、对商品的购买评价、对新闻的评论/点赞等多种类型，并可以分为显式反馈和隐式反馈两种类型。显式反馈指用户明确给出自己对候选项相关性的评价，如图 3-6 所示，在亚马逊、淘宝、豆瓣、IMDb 等网站中，用户可以对商品、图书、电影等物品给出自己的评分；在推荐功能中，用户可以主动选择对被推荐物品感兴趣或不感兴趣等。显式反馈数据是用户对候选项相关性评价的态度的直观体现，适合用于构造训练集。但是，显式反馈数据的数据量通常较少，在全部日志数据中的占比较低，且在搜索推荐等大部分应用场景中难以获取，同时还可能存在“刷分”等作弊行为，因此需要使用其他挖掘方法来弥补其不足。隐式反馈指的

是通过用户的行为推断其对候选项偏好性的方法。与显式反馈数据相比,隐式反馈数据更加稠密,同时也能更加真实地反映候选项的相关性。例如,网页的点击次数、页面停留时间、商品的收藏、新闻的评论转发等隐式反馈,虽然没有显式的给出对候选项的评分,却是用户真实行为和意图的反映。除此之外,隐式反馈通常和模型的优化目标函数关联更加密切,如广告点击率预估的任务就是对广告点击行为这一隐式反馈发生概率的预测。值得注意的是隐式反馈数据往往包含更多的噪声数据,需要进行数据预处理以过滤噪声。

在得到原始训练数据之后,一个至关重要的步骤就是特征的抽取和有效性筛选,因为提取的数据特征的质量直接决定模型训练效果的上限。数据特征主要可以分为基础数据特征、统计类特征和组合特征三类。在排序问题中,基础数据特征包括排序的条件特征、待排序候选项的属性特征等,如搜索关键词、网页内容、商品的价格、新闻的类型等都属于此类特征。数据统计类特征指在训练集上对原始数据特征进行统计分析,并将统计结果作为新的特征维度加入模型参与训练。例如,当用户在搜索引擎中查询关键词 A 时,可以对历史数据记录中同样搜索关键词 A 的其他用户最终点击的网页进行统计,并将每个网页的点击次数作为新的特征。统计类特征是对训练数据更深层次的挖掘,能够从更本质的层面反映用户或候选项的属性,在模型中通常具有更高的权重系数。在添加数据统计类特征时,一个需要注意的问题是要避免发生“数据泄露”。数据泄露是指在统计时使用了样本发生时间之后的数据信息,而事实上此部分数据在预测时是无法获知的。数据泄露通常会产生严重的过拟合问题,导致模型的训练效果和实际预测效果差距很大^[43]。组合特征,顾名思义是对已有的特征通过笛卡尔积等方式进行组合,主要作用是增强模型对非线性特征的表达能力。

提取出各类特征后,还需要对特征的有效性进行验证,以确定是否将其加入模型中。通常使用的特征有效性验证方法包括过滤法、包装法和集成法。其中,过滤方法主要是针对每一维度特征,计算其取值分布的方差等统计指标,去除统计指标低于设定阈值的特征;或通过计算特征与预测值之间的相关系数、互信息及卡方值等方式,衡量特征与预测值的相关性。包装法指的是选择一种基础模型进行数次训练,在每次训练之后去除部分特征,再使用其余特征进行下一次的训练。集成法的原理是使用具有特征选择能力的模型,如带有 L1、L2 正则项的逻辑回归模型和决策树模型,进行预训练,筛选出权重值较大的特征。最后,通常还需要对筛选出的数值特征进行标准化、归一化、离散化,类别特征进行独热编码等预处理操作,才能最终作为训练数据供排序模型学习。

3.4.2 历史行为数据处理方法

用户历史行为数据是搜索引擎、推荐系统等应用中的一类常见数据。用户历史行

为数据记录反映了用户的个性偏好和行为习惯,对于预测用户此后的行为具有重要的参考价值和指导意义^[44]。在排序模型中,历史行为数据是一类十分重要的特征,能够在很大程度上影响最终排序结果的准确率。举例说明,如图 3-7 所示,用户 A 和用户 B 在搜索引擎中输入相同的关键词“挪威的森林”,但却具有不同的检索意图,其中用户 A 期望的搜索结果为日本作家村上春树的小说《挪威的森林》,而用户 B 则希望查找由歌手伍佰演唱的同名歌曲。如果只考虑查询关键词这一特征,排序模型无法判断两个用户的真实检索目的,因而不能提供个性化的搜索结果。通过分析两个用户的历史搜索行为数据记录,可以发现用户 A 经常搜索文学作品的相关信息,而用户 B 则对音乐类的相关内容更具偏好。在这种情况下,将历史行为数据加入排序模型的输入特征中,可以使排序结果更加贴近用户的真实需求,从而提高排序精准度。

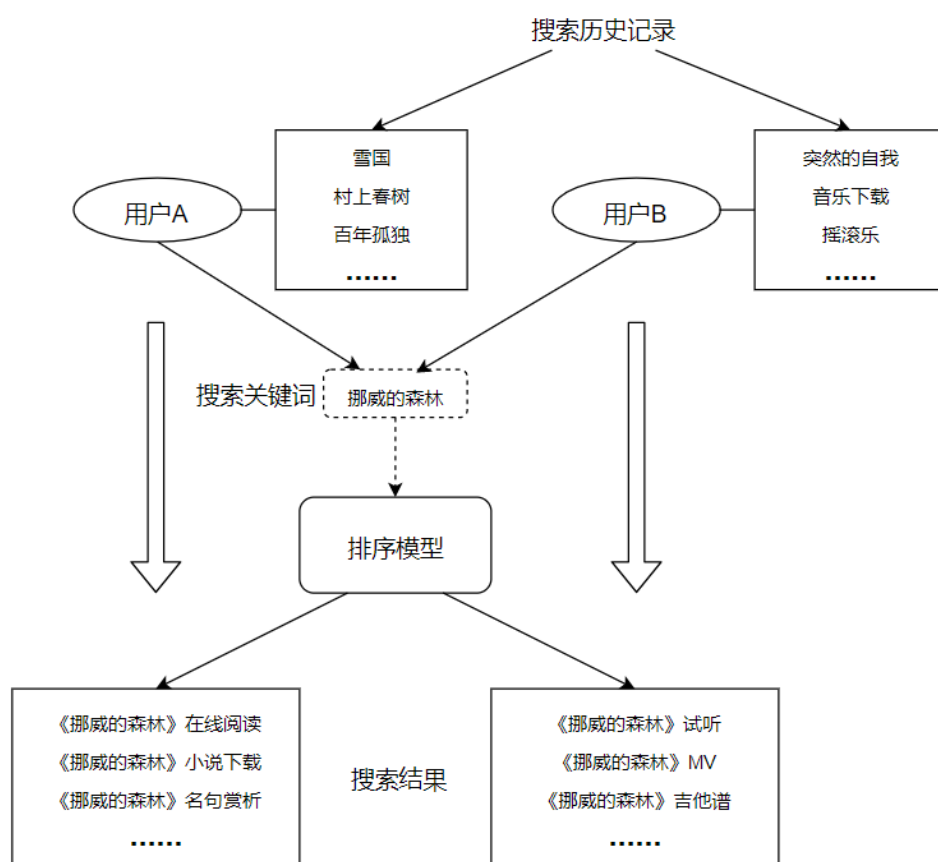


图 3-7 历史行为数据应用举例

如上所述,历史行为数据的处理方法是排序模型中的关键问题之一。目前通常采用的历史行为数据处理方法包括按时序连接的方法和基于卷积神经网络编码^[45]的方法等。下面对两种方法的原理和优缺点分别进行分析说明。

● 按时序连接方法

按时序连接的历史行为数据处理方法将用户的历史行为特征按照其产生的时间顺序进行连接,得到最终的历史行为特征向量。这种方法存在一些不足。首先,因为

历史行为记录的数据量不确定，此方法无法得到固定长度的特征向量。这一问题对应的一种解决方案是先确定需要的历史数据记录条数，对于数据量过多或过少的情况分别进行截断和补齐处理，如图3-8所示。这种方法解决了特征向量长度不固定的问题，但是可能引起数据信息丢失或与真实行为产生偏差。此外，按时序连接的历史行为处理方法只通过特征位置反映其时序关系，对历史行为数据的时序关联特性的建模能力较差，无法准确反映时间维度上的行为偏好特征。

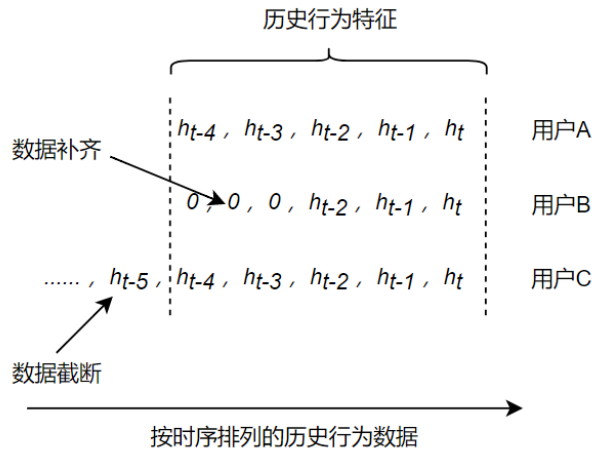


图 3-8 针对历史行为数据的截断补齐方法

● 基于卷积神经网络编码方法

基于卷积神经网络编码的历史行为数据处理方法主要解决了历史行为特征向量长度不固定的问题。如图3-9所示，编码模型主要划分为输入层、卷积层、池化层和全连接层，其中输入层接收原始历史行为向量，经过卷积层和池化层完成历史行为数据的特征提取，最后由全连接层得到最终的历史行为特征向量。

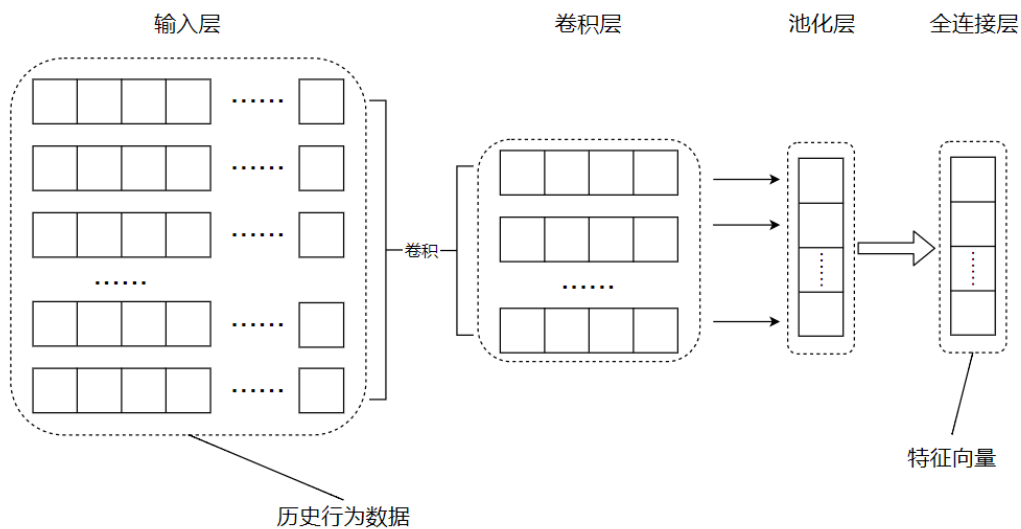


图 3-9 基于卷积神经网络的历史行为数据处理方法

基于卷积神经网络的编码方法解决了历史行为特征向量长度不固定的问题，且不

会因为截断补齐等操作而导致产生噪声或信息丢失。然而这种方法同样没有考虑历史行为数据的时序特性和前后关联，无法充分表示数据的特征。

综合上述分析，针对现有历史行为数据处理方法中存在的问题，本课题提出了一种使用循环神经网络对历史行为数据进行编码的方法，解决了特征向量长度不固定、数据噪声/信息丢失、无法表达数据时序特点等问题。这种方法的思想原理和具体操作过程将在论文第四章中进行详细介绍。

3.4.3 排序模型设计方法

排序模型设计方法包括排序模型的总体结构设计、训练方法选取和排序打分算法模型的设计优化等方面，需要解决的重点难点问题包括如何提升模型的排序精准度、提高模型的训练和预测计算效率等。目前工业界在广告投放和推荐系统等业务场景的排序模型中大多使用分层的架构设计。分层排序模型通常包含资源召回层和排序层，其中资源召回层也被称为候选生成层，排序层则可以进一步划分为粗排层和精排层。图 3-10 描述了一个分层排序模型的主体结构。

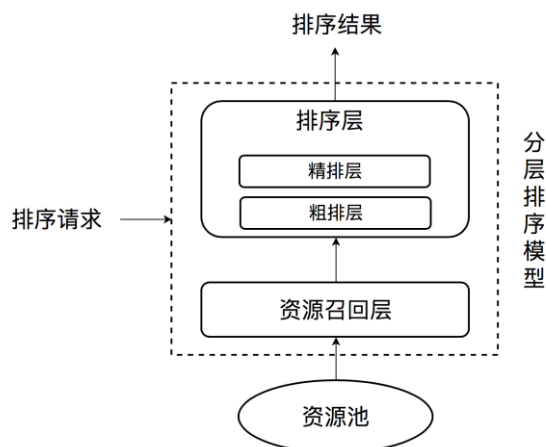


图 3-10 分层排序模型结构

其中，资源召回层实现从整体资源池中初步召回待排序候选项的功能。生成的待排序候选项集由粗排层进行进一步的过滤筛选，其结果送入精排层完成最终的排序并输出。之所以采用分层的系统结构设计是出于对模型排序效率的考虑。如上文所介绍，排序模型的核心思想是对候选项进行打分，再根据分数对其排序，因此排序过程的计算耗时正比于候选项的数量。在实际应用中，资源池中的全部资源的数量是十分巨大的，例如互联网上全部网页数、购物网站中全部的商品数等，在每次排序请求中使用复杂模型对全部的资源进行打分计算是不可实现的。同时，因为与排序请求相关的候选资源只占资源总数的极少一部分，所以首先使用资源召回层过滤掉大量的无关资源，生成与排序请求相关的候选项集合。资源召回层通常基于规则和高级数据结构实现，如使用基于关键字的倒排索引存储结构。在大多数情况，资源召回层输出的相关候选

项的数量仍比较多,不适合直接使用复杂模型计算每一个候选项的得分,需要使用由线性回归模型等简单模型构成的粗排层对候选项做进一步过滤,最后再由精排层确定候选项的排序得分。为了提高排序精准度,精排层打分模型通常采用多个基础模型集成得到的融合模型实现。

精排层常用的模型融合方法为对各个基础模型的结果进行线性加权求和。各基模型的权重系数可以人为指定,也可以再通过模型学习得到。后一种方式相当于以基模型的预测值作为特征向量训练更高层次的机器学习模型,因此又被称为模型堆叠(Stacking)。模型堆叠的优点在于能够找到各个样本下表现最好的基模型,组合不同模型的优势方面,所以当基模型显著不同时,堆叠的效果最好。堆叠方法的优点明显,但是也存在一个问题:各级模型的训练过程相互独立,高层次模型在其训练过程中只会优化对应于低层模型输出结果的参数,而不会对低层模型的内部参数进行更新。不同层级的模型分开训练,增加了时间消耗。如果能同步训练高层模型和各低层模型,无疑可以大大提高模型的训练效率,也可能会取得更好的预测效果。针对这一问题,Google 提出了一种新的融合模型结构,称为宽度&深度模型(Wide & Deep Model),其结构如下图所示^[4]:

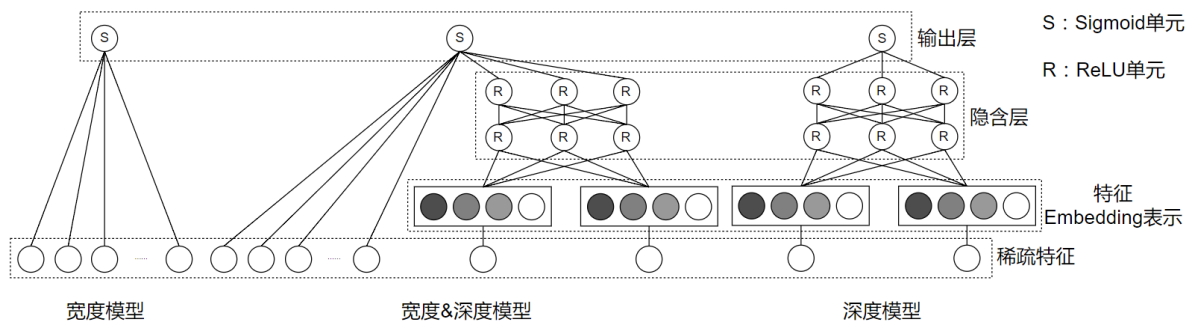


图 3-11 宽度&深度模型结构

宽度&深度模型是一种新颖的对线性模型和深度神经网络模型进行组合的方法。区别于普通的堆叠融合方法,宽度&深度模型使用一个逻辑回归模型实现了对由广义线性模型构成的模型宽度组成部分所利用的组合特征和由深度神经网络模型构成的模型深度组成部分对稀疏特征进行集成而得到的高层特征的集成。在宽度&深度模型的训练过程中,宽度部分和深度部分的各个参数会根据模型的预测误差,分别使用 FTRL 算法^[46]和 AdaGrad 等深度神经网络优化方法进行同步迭代更新。宽度&深度模型真正从结构上实现了不同类型模型的融合,是一种颇具创新性的方法。但是该模型也有比较强的局限性,即只能对特定类型的基础模型进行融合,而无法融合 GBDT 等决策树类模型。

3.5 本章小结

本章首先分析了现有的各种排序方法,以及排序学习中的三种模型训练方法各自的思想及特点。其次介绍了常用于拟合排序打分函数的三种算法模型,及三种常用的评价指标。最后从训练集构造、历史行为数据处理和排序模型设计三个方面概括总结了排序模型中的关键问题和技术难点,为第四章课题针对相关问题的研究和解决方案的提出做了铺垫。

第四章基于深度学习的排序模型关键问题研究

本章主要针对第三章中所提出的排序模型中的各关键问题,介绍了本课题的相关研究成果和解决方案。具体包括不同应用场景下排序模型训练方法的选择和排序打分算法模型的设计,深度神经网络模型损失函数及优化方法的选择,以及对历史行为数据的处理方法等内容。

4.1 排序模型训练方法选择

本文在 3.2 节介绍了排序模型的三种常用训练方法,并对每种方法的原理和特点进行了分析。根据分析的结果,可以得到如下结论:

(1) 从方法计算效率的角度考虑, Pointwise 方法的效率最高。具体来说,假设存在 N 个待排序候选项, Pointwise 方法对每个候选项进行打分,计算时间复杂度为 $O(N)$; Pairwise 方法需要计算候选项两两之间的相对位置关系,最坏情况下需要进行 $N(N-1)/2$ 次计算,时间复杂度为 $O(N^2)$; Listwise 方法需要计算模型预测结果输出的概率分布,为此需要计算 N 个候选项的每一种可能的排列序列的出现概率,计算次数等于其全排列数,即 $N!$ 。因为 $O(N!) > O(N^2) > O(N)$,所以 Pointwise 方法是三种训练方法中时间复杂度最低的一种,训练效率最高,适合用于样本数量大、时效性要求高的应用场景。

(2) 从排序精准度的角度考虑, Listwise 方法的效果通常优于另外两种方法。Wei Chen 等人的论文^[47]在 OHSUMED 数据集上进行了实验,以 NDCG 作为评价指标对比了多种排序算法的效果,实验结果如下表所示:

表 4-1 三种训练方法效果对比

训练方法	算法	评价指标	
		NDCG@5	NDCG@10
Pointwise 方法	Regression	0.4278	0.4110
Pairwise 方法	Ranking SVM	0.4164	0.4140
	RankBoost	0.4494	0.4302
Listwise 方法	SVM-MAP	0.4516	0.4319
	ListNet	0.4432	0.4410
	ListMLE	0.4471	0.4319

从实验结果可以看出,使用 Listwise 方法进行训练的排序算法的相关评价指标通常高于 Pointwise 方法和 Pairwise 方法中的算法。另外,在 2010 年举办的 Yahoo Learning to Rank Challenge 比赛中,Listwise 方法中的基于 RankNet 算法和 LambdaRank 算法改进得到的 LambdaMART 算法表现出了最佳的性能^[48]。综上所述,在对排序精准度要求较高的应用场景中,使用 Listwise 方法对排序模型进行训练更为合适。

除此之外,排序模型训练方法的选择还需要考虑训练数据的具体形式。Listwise 方法以排序列表作为学习对象,因此要求每条训练样本必须具有一组待排序候选项及相应的排序得分,例如在搜索引擎中,可以将每个查询词对应的结果文档集合及其相关程度作为一条训练样本。而在在线效果广告等应用场景中,通过挖掘日志数据中的隐式反馈,通常只能得到用户对单个候选项的行为记录。这种情况下,无法构造出 Listwise 方法所需的训练数据集,因此只能选择使用 Pointwise 方法将排序问题转化为分类问题进行建模。

综上所述,本课题设计的排序模型针对不同的应用场景的需求和特点,分别使用 Pointwise 方法和 Listwise 方法进行训练。具体来说,在训练样本数据量大、对排序效率要求较高以及将单个候选项作为训练样本的场景下,使用 Pointwise 方法训练排序模型。在对排序精准度要求高,能够获取到真实排序列表作为训练数据的情况下,使用 Listwise 训练方法。

4.2 排序打分算法模型设计

排序打分函数是排序模型的核心部分,使用何种算法模型对排序打分函数进行拟合是排序学习中的关键问题之一。本文在 3.3.1 节中对常用的算法模型的原理和特点进行了介绍,下面将通过实验对比分析各模型的实际应用效果。因为本课题所设计的排序模型在不同应用场景下,会分别以类别型数据和连续数值型数据作为学习的目标,所以对比实验分别针对分类任务和回归任务使用不同的数据集进行。本文使用的数据集取自 UCI 数据库^[49]。UCI 数据库是机器学习领域常用的标准测试数据来源,现总共包括三百多个数据集,其中包括知名数据集鸢尾花(Iris)、人口收入普查(Adult)和葡萄酒(Wine)等。

4.2.1 分类任务效果对比

针对分类任务的实验部分,本文使用了 UCI 数据库中的人口收入普查数据集和威斯康星州乳腺癌诊断数据集,下面分别简称为 Adult 数据集和 Cancer 数据集。其中 Adult 数据集从美国 1994 年的人口普查数据库中抽取得到,其学习目标是根据人口特征预测其年收入水平。Adult 数据集共包含 48842 个训练样本,每条样本包括 14 个属

性特征，包括年龄、人种、受教育水平、职业、每周工作时间等，从特征的数据类型上分为 7 个类别型特征和 7 个连续型数值特征，数据集的预测目标为该人员的年收入是否超过 50k 美元。Cancer 数据集是从 1995 年美国威斯康星州的乳腺癌诊断记录数据库中收集得到，用于根据从医疗图像中提取出的各项病理特征判断乳腺癌的种类。Cancer 数据集共包含 569 条训练样本，其中每条样本包括 32 个分类特征和 1 个分类标识，特征的数据类型全部为连续数值类型，具体包括细胞半径、质地的灰度值标准偏差、光滑度等，分类标识为乳腺癌的种类，即良性或恶性。Adult 数据集和 Cancer 数据集都是标准的二分类问题数据集，非常适合用于评价算法模型处理分类任务的能力和效果。

本文使用 Adult 数据集和 Cancer 数据集，对逻辑回归、分解机、GBDT、深度神经网络和宽度&深度模型五种算法模型分别进行了实验测试。其中，逻辑回归和深度神经网络模型使用了开源机器学习算法工具库 scikit-learn 中所实现的模型组件，分解机模型使用了由其发明者提供的开源库 libFM^[50]，GBDT 模型分别使用了 scikit-learn 中提供的版本和其改进后的开源实现库 XGBoost，宽度&深度模型则使用深度学习框架 TensorFlow 编码实现。实验过程针对两个数据集进行了简单预处理，具体包括空缺属性值填充、连续数值型特征的标准化、类别型特征的独热编码等，并对各模型的参数进行了调优。同时按照 20%：80% 的比例分别将两个数据集划分为训练集和测试集，采取 5 折交叉验证，记录 5 次实验结果的平均值。实验结果评估部分选用分类准确率和 AUC 值两项评价指标衡量各算法模型的分类效果。具体实验结果如下面的两个表格所示：

表 4-2 Adult 数据集实验结果

算法模型	评价指标	
	准确率	AUC
逻辑回归	0.9011	0.8625
分解机	0.9225	0.8697
GBDT	0.9213	0.8681
XGBoost	0.9286	0.8706
MLP	0.9101	0.8711
宽度&深度模型	0.9267	0.8714

表 4-3 Cancer 数据集实验结果

算法模型	评价指标	
	准确率	AUC
逻辑回归	0.9473	0.9947
分解机	0.9415	0.9962
GBDT	0.9310	0.9989
XGBoost	0.9298	0.9983
MLP	0.9643	0.9991
宽度&深度模型	0.9672	0.9995

各算法模型在两个数据集上的预测效果对比如图 4-1 所示。

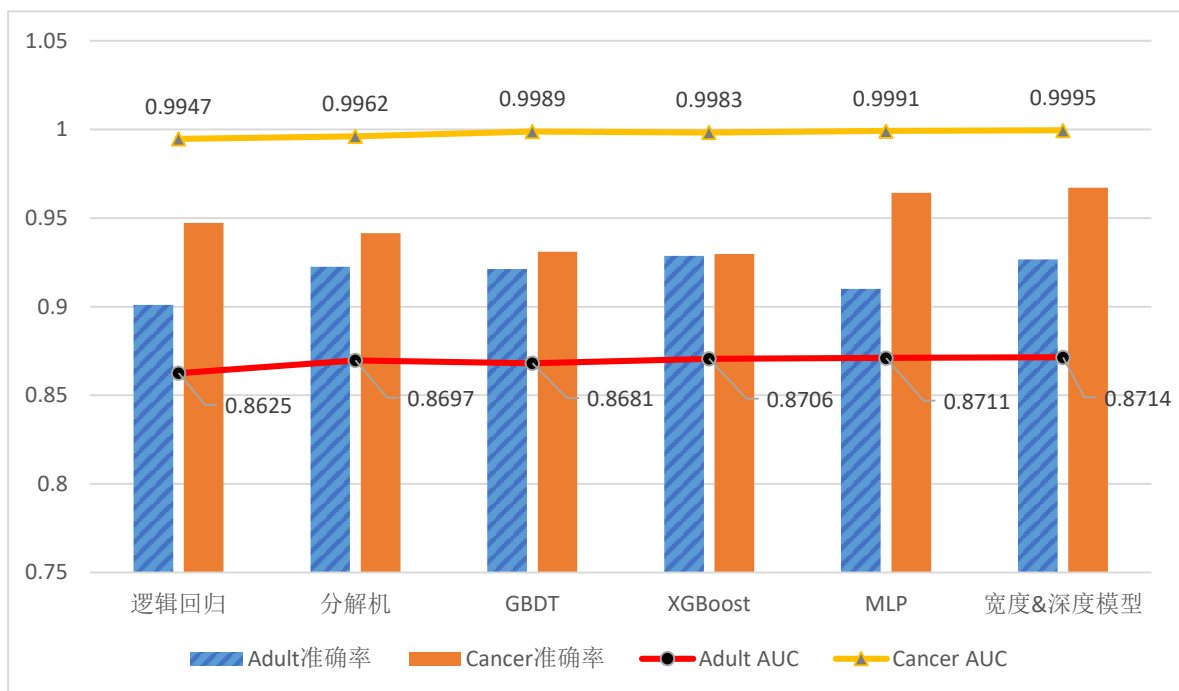


图 4-1 Adult 和 Cancer 数据集上各算法分类效果对比

实验结果显示，不同算法模型在不同数据上的预测效果存在差异性。其中，在 Adult 数据集上，XGBoost 模型预测准确率最高，为 0.9286，宽度&深度模型预测结果 AUC 最高，为 0.8714；而在 Cancer 数据集中，宽度&深度模型在预测准确率和 AUC 值两项评价指标上都排在全部测试模型的第一位。同时，宽度&深度模型的预测效果均好于作为其独立部分的逻辑回归模型和神经网络模型，验证了该模型组合方法的有效性。

4.2.2 回归任务效果对比

本文对各算法模型处理回归任务的能力的实验测试部分选用了 UCI 数据库中的森林火灾数据集和葡萄酒质量数据集。其中，森林火灾数据集的训练目标是利用气象

及其他特征数据预测发生在葡萄牙东北部地区的森林火灾所造成的烧毁区域面积。森林火灾数据集共包含 517 条训练样本，每条样本包含 13 维度的特征，具体包括火灾发生的月份、火灾发生时的风速和降水量及森林火险气候指数系统所提供的各项监测指标等，特征全部为连续数值型特征，回归预测的目标为由实数表示的森林焚毁面积。葡萄酒质量数据集由 Paulo Cortez 于 2009 年收集得到，目的是通过建立模型实现基于物理化学测试的结果预测葡萄酒的质量。该数据集共包括 4898 条样本数据，每条样本含有 12 个连续数值类型的特征值，包括葡萄酒样本的酸度、密度、pH 值、酒精含量等物理化学属性信息，回归预测的目标为葡萄酒样本的质量，使用取值为 0 到 10 的得分表示。

关于实验算法模型的选择部分，因为要处理的是回归预测问题，本文将分类任务实验部分中的逻辑回归算法替换为线性回归算法，同时修改了解析机、GBDT 等算法模型中的相关参数，将宽度&深度模型输出层的 Sigmoid 单元替换为 ReLU 单元。实验结果评估部分分别使用平均绝对误差 MAE 和平均平方误差 MSE 两项评价指标计算各模型预测结果与真实值之间的误差。实验详细结果如下面的两个表格所示。

表 4-4 森林火灾数据集实验结果

算法模型	评价指标	
	MAE	MSE
线性回归	23.75	1950.59
解析机	21.88	1876.76
GBDT	17.93	1405.75
XGBoost	14.23	1589.91
MLP	18.14	1654.38
宽度&深度模型	18.09	1622.33

表 4-5 葡萄酒质量数据集实验结果

算法模型	评价指标	
	MAE	MSE
线性回归	0.5415	0.4957
解析机	0.5132	0.4016
GBDT	0.5134	0.3994
XGBoost	0.5085	0.3979
MLP	0.5136	0.4320
宽度&深度模型	0.5101	0.4008

各模型的实验效果对比如图 4-2 和图 4-3 所示。

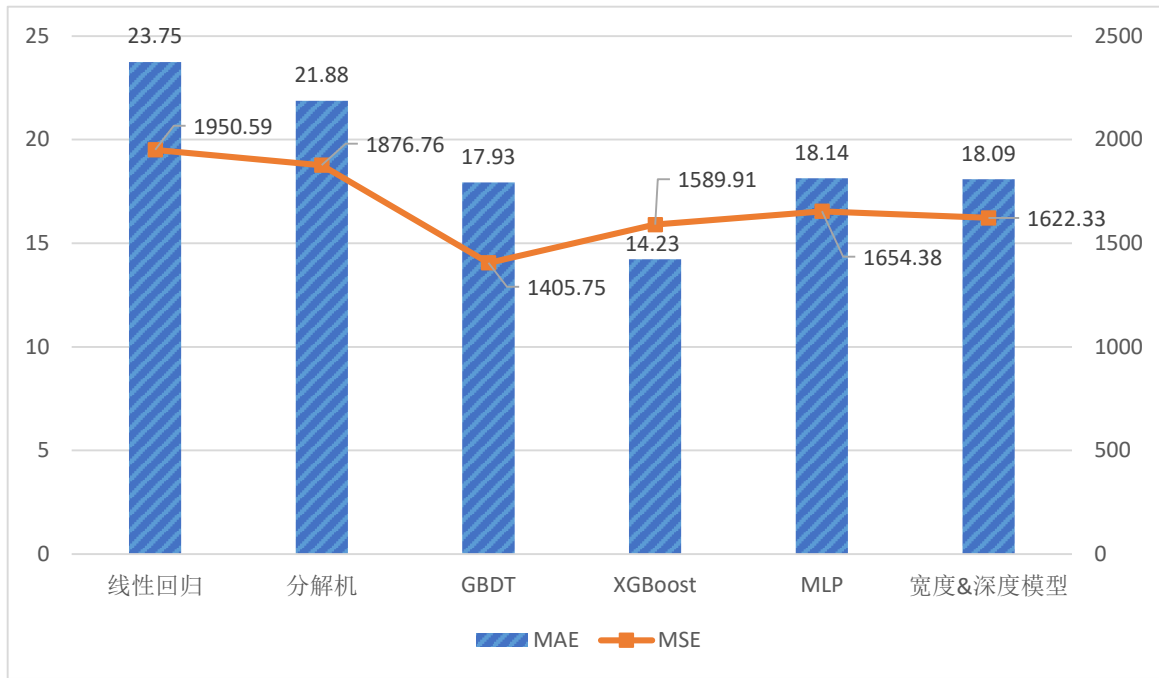


图 4-2 森林火灾数据集上各算法回归效果对比

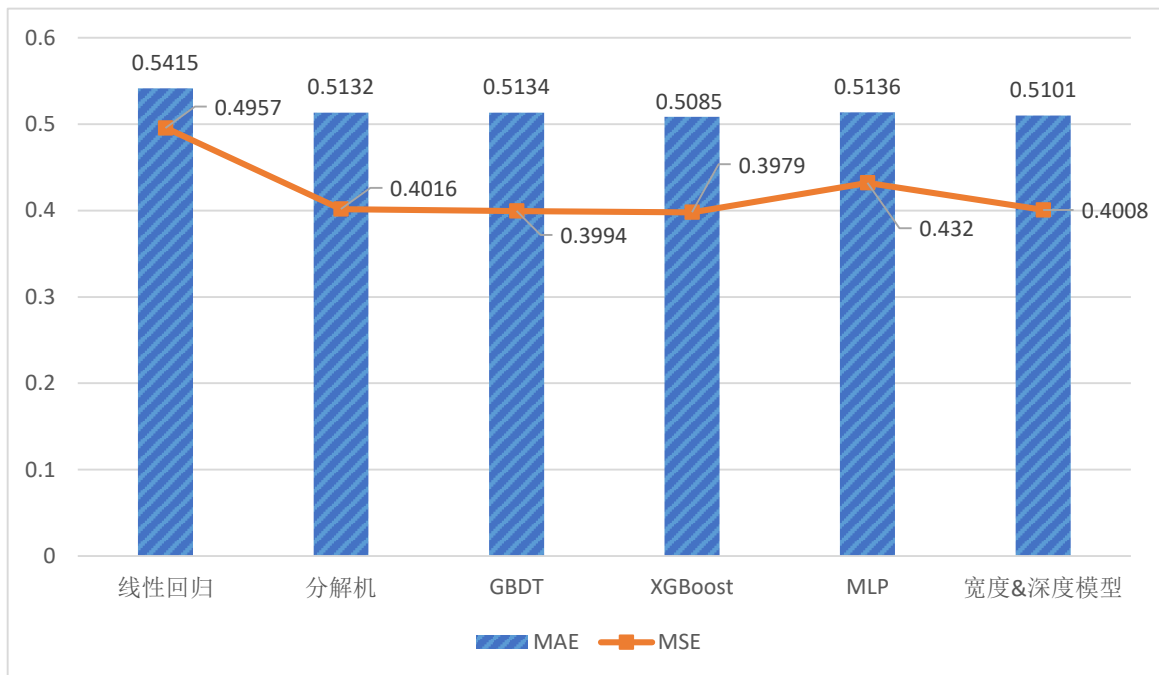


图 4-3 葡萄酒质量数据集上各算法回归效果对比

在处理回归任务的实验中,梯度提升树算法模型的预测效果普遍优于线性模型和神经网络模型。其中,在森林火灾数据集上,GBDT 模型的 MSE 误差最小,XGBoost 模型的 MAE 误差最小;在葡萄酒质量数据集上,XGBoost 模型的两项误差指标均取得全部测试模型中的最小值。同时,组合模型的预测误差低于线性回归模型和神经网络模型,证明该模型组合方法同样能够减少基础模型的回归预测误差。

4.2.3 打分模型设计

根据上述实验结论，本课题设计了如图 4-4 所示的排序打分算法模型。

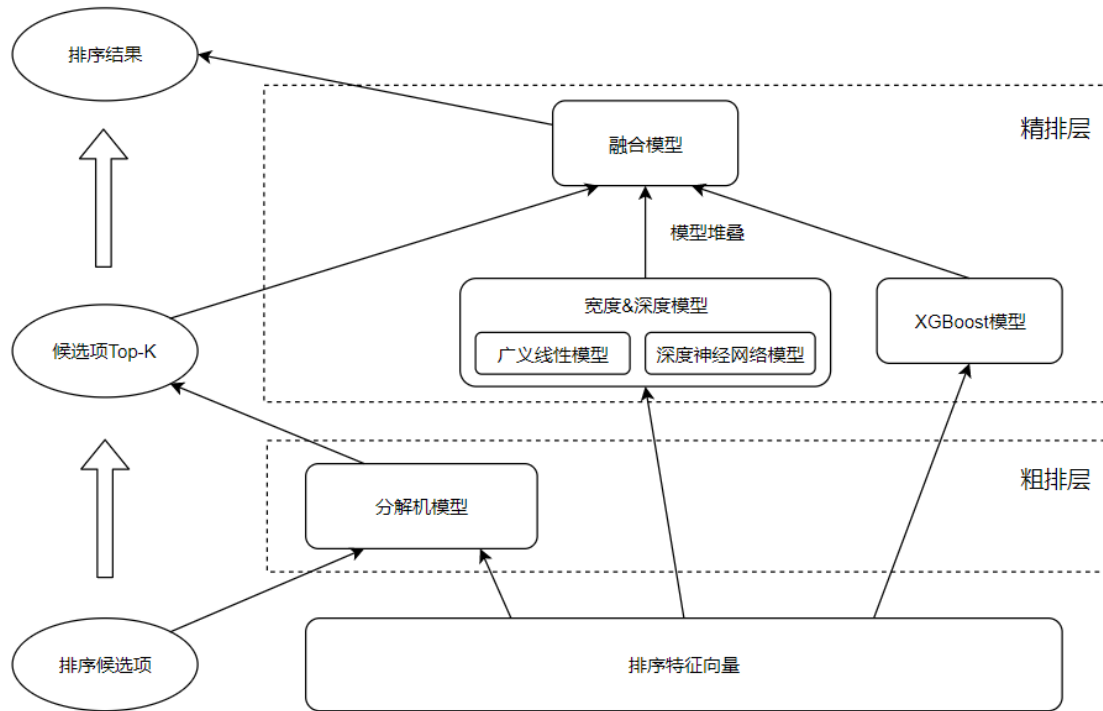


图 4-4 排序打分算法模型设计

本课题设计的排序打分模型具有两层结构，分为粗排层和精排层。如 3.4.3 节所介绍，粗排层完成对待排序候选项的初步排序，以减少精排层的输入数据量。粗排层输出排序结果后，取结果序列中的前 K 个候选项作为精排层的输入。精排层对各候选项打分，确定最终的排序结果。其中，粗排层模型选择分解机模型，而精排层使用融合模型作为打分模型，融合的基础模型包括广义线性模型、深度神经网络模型和 XGBoost 模型，融合方法为首先将广义线性模型和深度神经网络模型组合为宽度&深度模型结构，再与 XGBoost 模型使用堆叠方法进行融合。

该排序打分模型的主要优点包括采用了分层的结构设计和引入了深度神经网络结构。分层的结构设计的优点在于使排序打分模型取得了在预测准确率和计算效率两个方面的平衡，同时使打分模型具备更强的弹性，能够满足不同应用场景的需求。通过调节 K 的取值，可以控制粗排层的过滤比例，特别是当待排序候选项数量不多时，可以跳过粗排层，直接使用精排层模型对全部候选项进行打分。模型选择方面，粗排层使用了线性模型中预测效果较好的分解机模型，精排层则根据相关实验结果确定了三种基础算法模型。通过引入深度神经网络结构，增强了融合模型对稀疏输入的特征提取和泛化能力，弥补了 XGBoost 模型存在的不足。

本文对所提出的融合打分模型的预测精准度和模型分层结构设计的计算效率进行了实验测试。模型预测精准度实验部分使用上文介绍的四个数据集对融合打分模型

的效果进行了测试，评价指标分别使用准确率和 MAE，同时以各数据集上误差最小的单模型为基准计算提升比，实验结果如下表所示。

表 4-6 融合模型预测精准度实验

数据集	评价指标	单模型最优结果	融合模型结果	提升比 (%)
Adult	准确率	0.9286	0.9304	0.19
Cancer	准确率	0.9643	0.9681	0.39
森林火灾	MAE	14.23	13.56	4.71
葡萄酒质量	MAE	0.5085	0.5063	0.43

实验结果显示，融合模型在全部四个测试数据集上均取得了优于单一模型的预测效果，验证了本文所提出的模型融合方法的有效性。其中，因为各单模型在森林火灾数据集上的预测误差较大，融合模型在该数据集上的预测效果提升最为显著，提升比达 4.71%。同时，在 Cancer 数据集上，融合模型取得的预测准确率为 0.9681，除了高于单模型的最优结果 0.9643 外，也高于宽度&深度模型的结果 0.9672，这表明加入 XGBoost 模型能够在一定程度上进一步提升融合模型的预测效果。

模型计算效率实验部分测试了不同数据规模下粗排层使用的线性模型和精排层使用的融合模型的打分计算耗时，以及在 K 的不同取值下分层排序打分模型的整体耗时。实验环境为运行 64 位 Windows 10 操作系统的桌面计算机，CPU 为 Intel Core i5-4590，频率为 3.30GHz，内存 8GB。待排序候选项数量从 200 到 1000，每增加 200 个进行一次实验，K 值分别取 20、50 和 100。为减少随机因素的影响，各组实验重复进行 5 次，记录 5 次实验的平均结果如下表所示。

表 4-7 融合模型打分计算耗时实验

候选项数量 (个)	打分计算耗时 (毫秒)				
	线性模型	融合模型	K=20	K=50	K=100
200	0.001	0.029	0.006	0.009	0.014
400	0.001	0.061	0.006	0.010	0.014
600	0.001	0.084	0.007	0.010	0.015
800	0.001	0.107	0.007	0.011	0.015
1000	0.001	0.128	0.007	0.011	0.015

各模型在不同参数条件下的打分计算耗时随候选项数量增长的变化趋势对比如图 4-5 所示

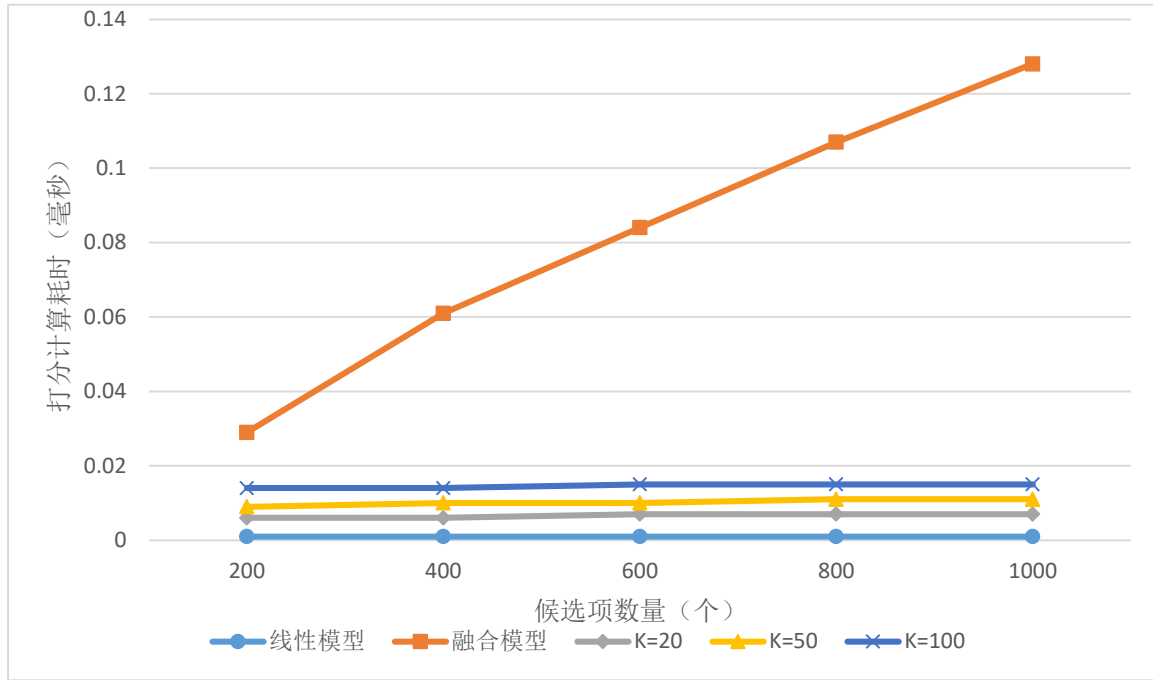


图 4-5 模型打分计算耗时对比

实验结果显示粗排层所使用的线性模型的打分计算耗时远低于融合模型。随着待排序项数据规模的增加，线性模型的计算耗时基本保持不变，而融合模型的打分计算耗时则呈线性增长趋势。对于采用分层结构设计的排序模型，其整体打分计算耗时随待排序候选项数量的增加而增长的趋势不明显，而是与参数 K 的取值呈正比关系。这是因为融合模型的计算耗时占据整体打分耗时的主要部分，融合模型的输入候选项数量由参数 K 的取值确定，而其余候选项则被计算耗时很少的粗排层模型所过滤。实验结果表明分层结构设计降低了候选项数量对排序模型打分计算耗时的影响，使排序打分模型取得了在预测准确率和计算效率两个方面的平衡。

4.3 损失函数及优化方法选择

损失函数和优化方法的选择是深度学习中的两个重要问题。其中，损失函数作为算法模型的优化目标，直接影响模型参数的更新过程及模型最终的训练效果。而优化器选择的合适与否，则会决定神经网络模型训练过程的收敛速度，以及是否会陷入局部最优的情况。针对这两个问题，本文对目前常用的几种损失函数和优化方法进行了分析对比，并说明了课题的选择和理由。

4.3.1 损失函数

深度学习中常用的损失函数包括平方损失函数、绝对值损失函数以及交叉熵损失函数等。使用 a 表示神经网络单元的输出， W 和 b 分别表示权值参数向量和偏移量，三

者之间的关系表示为：

$$a = \sigma(z) = \sigma(Wx + b) \quad (4-1)$$

使用 y 表示需要预测的真实值， C 表示误差损失，则绝对值损失函数表示为：

$$C = |a - y| \quad (4-2)$$

平方损失函数的定义为：

$$C = \frac{1}{2}(a - y)^2 \quad (4-3)$$

交叉熵损失函数的定义为：

$$C = H(y, a) = -\sum y_i \log(a_i) = -[y \ln a + (1 - y) \ln(1 - a)] \quad (4-4)$$

线性模型通常使用绝对值损失函数或平方损失函数，而在深度神经网络模型中，交叉熵损失函数则更为合适。这是因为当使用反向传播算法训练模型时，根据链式求导法则，对于平方损失函数，权重参数 W 和偏置 b 的梯度计算公式为：

$$\frac{\partial C}{\partial W} = (a - y)\sigma'(z)x \quad (4-5)$$

$$\frac{\partial C}{\partial b} = (a - y)\sigma'(z) \quad (4-6)$$

可以看出，权值参数 W 和偏置 b 的梯度更新公式中均包含激活函数的导数项 $\sigma'(z)$ 。当神经网络模型使用 Sigmoid 和双曲正切等函数作为激励函数时，大部分情况下其导数值会落在左右两边的饱和区间中，对应导数值较小，导致参数的更新过程非常缓慢。而对于交叉熵损失函数，参数 W 对应的偏导数为：

$$\frac{\partial C}{\partial W} = \frac{\partial C}{\partial a} * \frac{\partial a}{\partial z} * \frac{\partial z}{\partial W} = \left(\frac{y}{a} - \frac{1-y}{1-a}\right)\sigma'(z)x \quad (4-7)$$

化简得到：

$$\frac{\partial C}{\partial W} = \frac{\sigma'(z)x}{a(1-a)}(a - y) \quad (4-8)$$

而由函数的定义可以得到：

$$\sigma'(z) = \sigma(z)(1 - \sigma(z)) = a(1 - a) \quad (4-9)$$

因此，可以化简得到偏导数的最终形式：

$$\frac{\partial C}{\partial W} = (a - y)x \quad (4-10)$$

同理可以得到偏置 b 的偏导数为：

$$\frac{\partial C}{\partial b} = (a - y) \quad (4-11)$$

与公式（4-5）和（4-6）对比可以看出交叉熵损失函数对应的参数偏导数公式中已经不包含激活函数的导数项，因此参数的学习速率不再受激活函数形式的制约，而是只与预测值和真实值之间的误差有关，即误差越大，学习速率越快。综上所述，本

课题设计的排序打分模型中的神经网络部分选择使用交叉熵损失函数。

4.3.2 优化方法

神经网络模型常用的优化方法包括梯度下降方法、Momentum 方法、Adagrad 方法、Adam 方法等^[51-53]。下面分别对每种方法的原理和特点进行分析说明。

● SGD

梯度下降方法作为机器学习最为常用的一种优化方法，具有计算简便、快速迭代等优点。梯度下降法包括批量梯度下降法（Batch Gradient Descent, BGD）、随机梯度下降（Stochastic Gradient Descent, SGD）和小批量梯度下降法（Mini-batch Gradient Descent, MBGD）三种形式。三种方法的区别体现在每一轮迭代过程计算梯度时用到的样本数量不同：BGD 方法考虑全体样本，SGD 方法每次使用一条样本，而 MBGD 方法作为前两种方法的折中，每次使用固定数量（Batch Size）的样本计算梯度值。其中，批量梯度下降方法使用由全部样本计算得到的梯度更新参数，能够保证在全局方向上的优化，但是当样本数量很大时计算效率十分低下；与之相比，SGD 方法迭代速度更快，但是无法保证每次参数更新都更加接近全局最优解。

深度学习中使用的随机梯度下降方法，实际上指的是小批量梯度下降法，只是将固定数量的样本视为计算单位。使用 η 表示学习速率， g_t 表示第 t 轮迭代时的梯度，参数 x 的更新公式为：

$$\begin{aligned} x_{t+1} &= x_t + \Delta x_t \\ \Delta x_t &= -\eta * g_t \end{aligned} \quad (4-12)$$

随机梯度下降方法的主要缺点在于即使使用了小批量样本数据作为梯度计算单位，收敛过程仍然存在震荡现象，且在用于非凸函数优化时容易陷入局部最优的陷阱。

● Momentum

针对随机梯度下降方法存在的参数更新不稳定的问题，Momentum 方法提出了一种改进方法，即引入动量（Momentum）因子，这也是该方法名称的由来。

Momentum 方法的主要思想是参数在每一轮更新时，会在一定程度上延续上一轮的更新方向，同时使用当前的梯度对更新方向进行调整。Momentum 方法的参数更新公式为：

$$\Delta x_t = \rho * \Delta x_{t-1} - \eta * g_t \quad (4-13)$$

其中， ρ 即表示动量因子，值域为 $[0, 1]$ ，表示在多大程度上保持此前的更新方向。 η 代表学习速率，表示当前梯度值对更新方向的影响程度。通过调整 ρ 和 η 的取值，可以控制参数的最终更新方向在两个影响因素上的分布比例。

Momentum 方法还有一种改进形式，称为 Nesterov Momentum 方法。该方法的主要思想如下图所示：

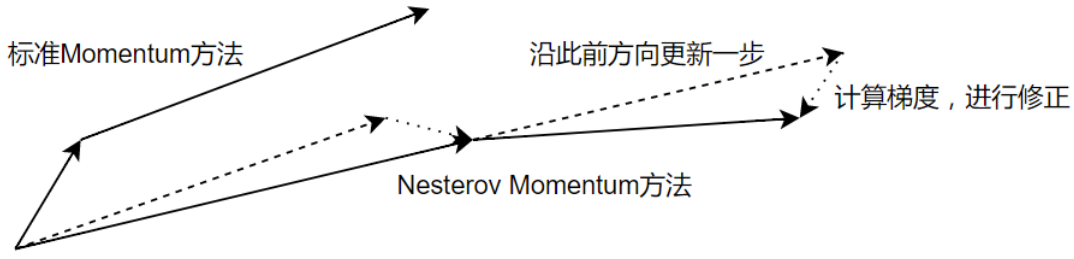


图 4-6 Nesterov Momentum 方法示意

Nesterov Momentum 方法的参数更新过程为：首先沿此前的更新方向更新一步，并计算结果位置的梯度值，再使用梯度值进行一步更新，并修正得到最终的更新方向。参数更新公式为：

$$\Delta x_t = \rho * \Delta x_{t-1} - \eta * \Delta f(x_t + \rho * \Delta x_{t-1}) \quad (4-14)$$

● Adagrad

上面介绍的优化方法都存在一个共同的问题，即各个不同的参数都具有相同的学习速率。而实际上各参数的收敛速度并不一致，当某些参数到达仅需要细微调整的阶段时，另一些参数可能因为训练样本数量较少等因素而导致仍然需要进行较大幅度的更新。针对这一问题，有研究者提出了一种称为 Adagrad 的优化方法，顾名思义，该方法能够在不同参数上自适应的选择合适的学习速率。Adagrad 方法的公式为：

$$\Delta x_t = - \frac{\eta}{\sqrt{\sum_{\tau=1}^t g_{\tau}^2 + \epsilon}} g_t \quad (4-15)$$

其中， g_t 表示当前的梯度， g_{τ} 表示参数的历史梯度， ϵ 表示初始学习速率。因为 Adagrad 方法能够自适应的调整学习速率，所以 ϵ 的取值的影响不大，通常将其设为一个较小的正数值来保证分母非零。Adagrad 方法通过记录每个参数的历史更新过程，使其对应的学习速率随着更新的总体距离的增加而逐渐减少，这样就使得每个参数在训练开始阶段能够以较大的幅度进行更新，并随着迭代次数的增加而逐步趋于稳定，即实现不同参数各自学习速率的自适应调整。

● Adam

Adam 方法是 Momentum 方法和 Adagrad 方法的结合，该方法使用动量因子加速参数收敛过程、减少震荡，同时针对不同参数自适应地动态调整其学习速率。Adam 方法的计算公式为：

$$\begin{aligned}
m_t &= \mu * m_{t-1} + (1 - \mu) * g_t \\
n_t &= \nu * n_t + (1 - \nu) * g_t^2 \\
\hat{m}_t &= \frac{m_t}{1 - \mu^t} \\
\hat{n}_t &= \frac{n_t}{1 - \nu^t} \\
\Delta x_t &= -\frac{\hat{m}_t}{\sqrt{\hat{n}_t} + \epsilon} * \eta
\end{aligned} \tag{4-16}$$

其中， m_t 和 n_t 分别表示对当前梯度的一阶和二阶矩估计，即对期望 $E|g_t|$ 和 $E|g_t^2|$ 的估计，而 \hat{m}_t 和 \hat{n}_t 是对估计 m_t 和 n_t 的修正，其结果近似于对期望的无偏估计。Adam方法结合了动量方法和Adagrad方法的优点，即适用于处理稀疏梯度和非平稳目标函数，同时通过偏置校正使得每一轮迭代的学习速率保持在一个确定范围之内，保证了参数更新过程的平稳性。多项实验结果表明，相比于其他方法，使用Adam方法作为优化方法能够取得更好的效果。综上所述，本课题选择使用Adam方法作为深度神经网络模型的参数优化方法。

4.4 基于循环神经网络的历史行为数据编码方法

如3.4.2节所介绍，历史行为数据是排序模型中的一类十分重要的数据类型。通过挖掘历史行为数据，对行为习惯偏好进行建模并抽取出对应特征，对于提高模型排序结果的精准度有重要意义。如果使用目前常用的按时序连接的方法处理历史行为数据，会产生历史行为记录数据数量不一致、特征向量长度不固定的问题。如果采用截断补齐的处理方法，虽然解决了特征向量长度不固定的问题，但是又会引起数据信息丢失或与真实行为产生偏差。基于卷积神经网络编码的方法弥补了上述缺陷，但是与简单的按照时间顺序对历史行为记录进行连接的方法相同，没有充分利用到数据中蕴含的时序关系和前后关联特性。对于上面提出的这些问题，本课题进行了针对性的研究，并提出了一种基于循环神经网络模型对历史行为数据进行编码的解决方案。

这种解决方案参考了自然语言处理领域流行的编码-解码模型结构。在自然语言处理中的机器翻译、问答系统等Seq2Seq问题中，输入和输出序列通常是不等长的文本数据，这导致无法直接使用输入和输出序列作为模型的输入和训练目标。在这种情况下，研究者提出了如图4-7所示的编码-解码模型结构^[54]，首先利用编码器将输入序列转换为确定长度的语义编码向量，再使用解码器将此向量转换为对应的结果输出序列。可以认为连接编码器和解码器的固定向量包含了输入序列的语义信息。而循环神经网络模型由于其独特的结构设计，十分适合用于对序列数据进行建模，因此常作为编/解码器的模型实现。

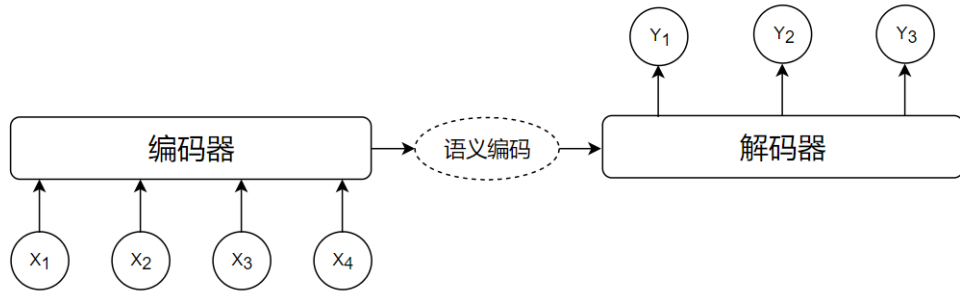


图 4-7 编码-解码模型结构

因为对历史行为数据处理的主要目的是得到其特征向量表示，而无需对历史行为记录进行还原，因此历史行为数据编码模型舍弃了编码-解码模型中的解码器结构，只保留其编码器部分。图 4-8 描述了历史行为数据编码模型的详细结构。

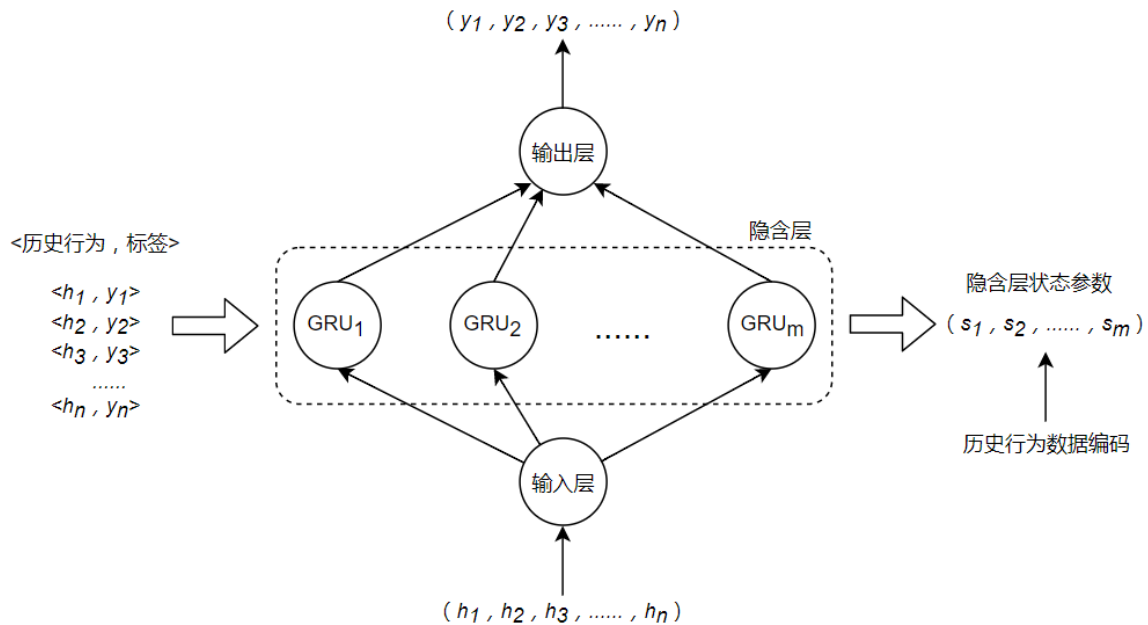


图 4-8 历史行为数据编码模型

如图所示，历史行为数据编码模型的主体结构为包含一层隐含层， m 个 GRU 单元的循环神经网络模型。模型的输入为每条历史行为数据对应的特征向量 h_i ，输出为每条历史行为对应的标签特征向量 y_i ，这里的标签指的是历史行为的结果，如点击的网页、购买的商品、阅读的新闻等。因为标签通常是类别型的数据，所以需要进行如独热编码等编码操作得到标签特征向量。通过对编码模型进行训练，并将用户的历史行为输入模型，可以计算得到模型隐含层单元对应的一组最终状态参数 (s_1, s_2, \dots, s_m) ，这组参数包含了历史行为信息，即为模型对历史行为数据的编码结果。编码向量的维度可以通过调节隐含层神经单元的数量进行控制。

4.5 本章小结

本章主要介绍了本课题对于基于深度学习的排序模型中的关键问题的相关研究成果和给出的解决方案。针对不同应用场景下的排序任务，本课题分别使用 **Pointwise** 方法和 **Listwise** 方法进行模型训练。本文使用 UCI 数据库中的公开数据集，分别对各个算法模型在分类和回归任务下的性能效果进行了实验，根据实验结果，确定使用由深度神经网络模型、广义线性模型和梯度提升树模型组合得到的融合模型拟合排序打分函数，同时确定使用交叉熵函数作为深度神经网络模型的损失函数，并使用 **Adam** 方法作为模型参数的优化方法。此外，本章还提出了一种基于循环神经网络的历史行为数据编码器模型，解决了此前历史行为数据相关处理方法存在的一系列问题。

第五章 基于深度学习的排序模型的设计与实现

本文在第四章对基于深度学习的排序模型中的模型训练方法、打分算法模型、损失函数和优化方法的选择以及历史行为数据的处理方法等关键问题进行了详细的分析，并介绍了本课题的相关研究成果和针对以上问题给出的解决方案。在此基础上，本章阐述了排序模型的详细设计与实现方案，具体包括模型的总体结构设计，功能模块的划分及每个模块的具体功能设计等内容。最后，本章将课题提出的排序模型应用于解决地铁单边交易处理问题，对模型的效果进行了验证。

5.1 基于深度学习的排序模型设计

5.1.1 模型总体结构设计

基于深度学习的排序模型的总体结构设计如图 5-1 所示：

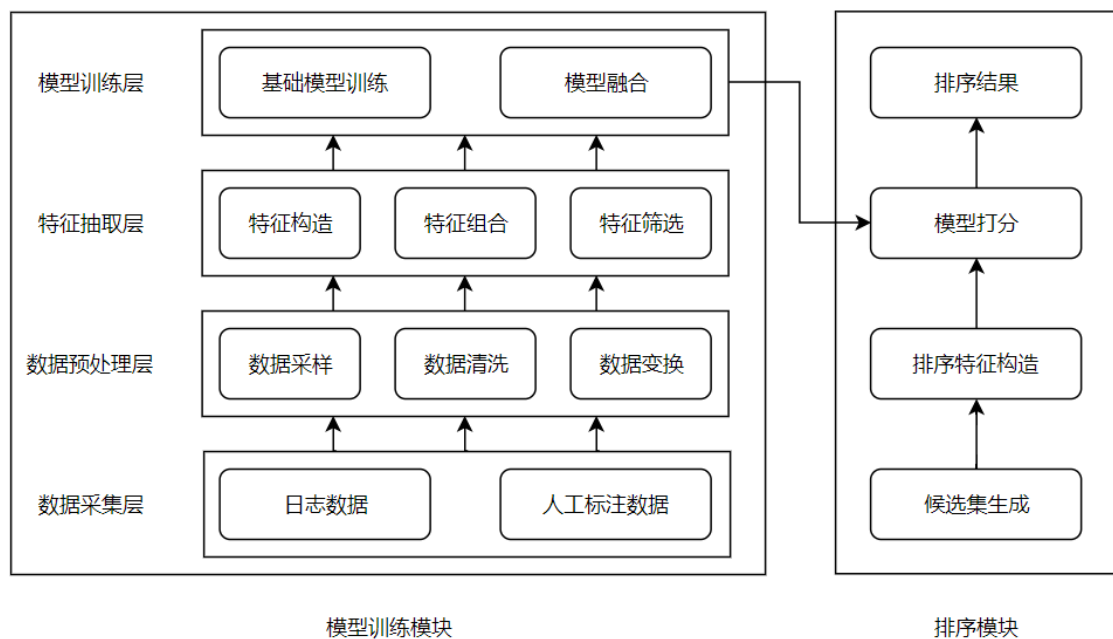


图 5-1 基于深度学习的排序模型总体结构设计

模型整体分为两个主要模块：模型训练模块和排序模块。其中，模型训练模块的主要功能为基于训练数据集训练得到用于拟合排序打分函数的算法模型，具体划分为四个层次，包括数据采集层、数据预处理层、特征抽取层和模型训练层。排序模块的主要功能为使用模型训练模块产出的打分模型，针对具体的排序任务给出预测序列结果。排序模块主体流程分为四个步骤，分别为候选集生成、排序特征构造、模型打分和排序结果输出。下面分别对两个模块的结构和功能进行具体介绍。

5.1.2 模型训练模块

模型训练模块属于排序模型的离线计算部分，模块的核心功能是生成用于对待排序候选项进行评分的算法模型。模型训练模块包含从获取训练数据到算法模型产出的全部流程，自底向上具体划分为数据采集层、数据预处理层、特征抽取层和模型训练层四个层次。

● 数据采集层

数据采集层主要完成训练数据的收集功能。训练数据的来源主要包括人工标注数据集和历史日志数据集。其中，人工标注数据中的记录具有明确的排序得分，而历史日志数据则根据显式反馈和隐式反馈类型的不同，其样本的排序预测标记可能为类别型或数值型数据。

● 数据预处理层

数据预处理层的主要功能为对从数据采集层获取的原始数据记录进行初步整理，主要操作包括数据采样、数据清洗、数据变换等。数据预处理层的具体功能设计如图 5-2 所示。

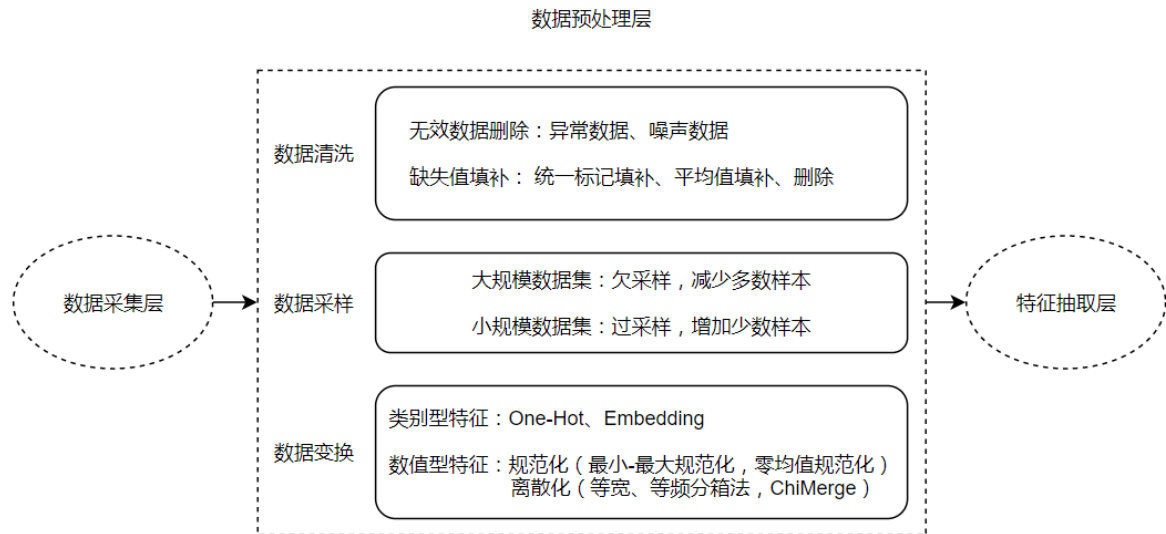


图 5-2 数据预处理层功能设计

其中，数据清洗操作具体包括无效数据删除、缺失值填补等处理过程。在数据采集层中，由于人工失误或系统故障等无法避免的因素的影响，原始数据记录中可能包含明显偏离常值的异常数据，或是混入其他噪声数据。如果不事先将这部分无效数据从训练集中剔除，则将严重影响算法模型最终的预测效果。同时，对于记录属性存在缺失的样本数据，需要进行缺失值填补处理，具体方法包括使用统一标记填补、使用平均值填补或将存在缺失的数据记录直接删除等。

当训练数据为从日志中挖掘得到的隐式反馈数据时，本课题所提出的排序模型会使用 Pointwise 训练方法，将排序问题转化为二分类问题进行求解。在这种情况下，

正负样本的数量比例通常极不平衡,因此需要对原始数据记录进行采样。采样方法包括过采样和欠采样,其中过采样方法通过采用复制等手段增加少数样本的数量,使样本比例达到平衡,而欠采样则是通过删除一部分多数样本的方法达到平衡^[55]。采样方法的选择需要根据具体应用场景确定,通常情况下,在大规模数据集上使用欠采样,小规模数据集上使用过采样效果较好。

数据变换是数据预处理层的一个关键步骤,包括类别型数据的编码,连续数值型数据的规范化、离散化等操作。多数算法模型,如逻辑回归模型、GBDT 模型等,都无法直接处理类别型特征,因此在训练前必须先将类别型数据转换为数值型数据,常用编码方法包括独热编码或使用类别的 Embedding 表示等。对数值型数据进行规范化处理,有助于提高模型参数优化的收敛速度,而离散化操作能够使数据特征更加突出,有助于提高模型对特征信息的提取能力。

● 特征抽取层

如图 5-3 所示,特征抽取层实现了从样本数据记录中提取特征构造训练集的功能,主要包括特征构造、特征组合和特征筛选等过程。

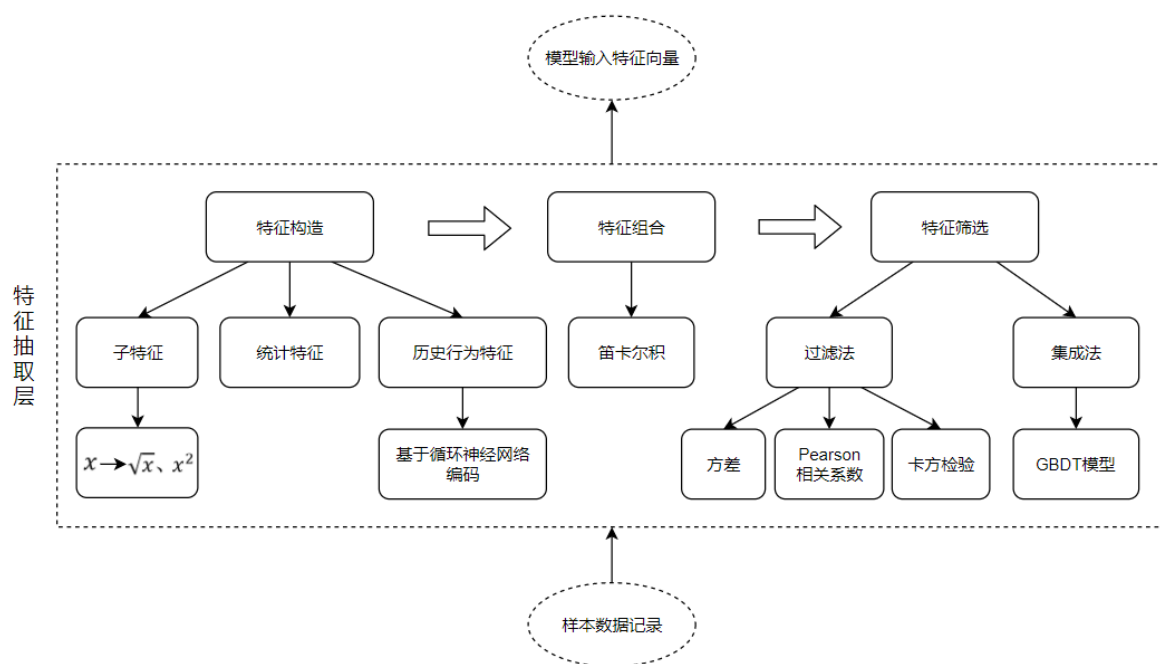


图 5-3 特征抽取层功能设计

特征构造过程具体可以分为子特征构造、统计特征构造、历史行为特征构造三个部分。其中,子特征构造指的是对于每一维度的基础特征 x ,将 \sqrt{x} 、 x^2 等 x 的非线性变换形式作为新的特征加入到模型中。子特征构造增强了模型的非线性处理能力,能够在很大程度上提升逻辑回归、分解机模型等线性模型的预测效果。统计特征构造指的是将在训练样本上计算得到的相关统计量作为新特征,例如某两个特征的共同出现次数、特征与预测目标间的相关函数等。历史行为特征构造是本课题特别提出的一种特

征构造方法，该方法的具体介绍已在 4.4 节给出，这里不再赘述。特别地，在冷启动的情况下，记录中不存在对应历史行为数据，此时的历史行为特征向量使用零向量表示。

特征组合操作同样能够增强模型的非线性表达能力，同时还能起到个性化建模的作用。对于大规模训练样本而言，使用全部数据训练模型可能会忽略掉样本的局部特征，对每一类样本建模又存在模型数量过多、单类样本数量少、容易过拟合等问题。特征组合方法在一定程度上弥补了上述不足，较好的兼顾了训练样本的全局特性和局部差异。

特征筛选过程对各维度特征的有效性进行衡量，并确定最终的模型输入特征向量。特征筛选过程需要权衡模型的预测效果和计算效率，选取前 K 个有效特征作为模型输入。基于深度学习的排序模型使用过滤法和集成法进行特征筛选，即先通过计算方差、相关系数等指标过滤无效特征，再使用 GBDT 模型进行预训练得到各维度特征的重要性。

● 模型训练层

模型训练层接收特征抽取层输出的训练集，完成各个基础模型的训练及模型融合等步骤，输出最终得到的排序打分模型。

模型训练层包含三种基础模型，分别是神经网络模型、包含组合特征的广义线性模型和梯度提升树模型。其中，神经网络模型和广义线性模型通过宽度&深度模型的形式进行组合，得到的组合模型再和梯度提升树模型进行堆叠融合。根据训练方法的不同，高层模型分别使用线性回归模型和逻辑回归模型。

通过这种方法训练得到的融合模型，兼顾了三种模型各自的优点，具备较强的稀疏特征学习能力和非线性表达能力，相比于单一的基础模型具有更好的排序效果。

5.1.3 排序模块

排序模块的主要功能为使用模型训练模块得到的打分算法模型对待排序候选项进行评分，并按照得分高低将排序结果序列返回。排序模块的主要操作步骤包括候选集生成、排序特征构造、模型打分和排序结果输出等。

其中，候选集生成步骤实现从全部资源中筛选出待排序候选项的功能。候选集生成通过规则结合简单算法模型的方式实现，具体方法包括使用基于关键字的倒排索引结构、使用待排序项的 PageRank 值及历史排序得分等。候选集生成步骤的主要作用是缩小排序打分模型的输入数据规模，减少计算量，提高排序模型的实际可用性。

在候选集生成步骤筛选得到待排序候选项集合后，由排序特征构造步骤完成打分模型输入特征向量的拼接功能。特征向量由排序条件特征、候选项属性特征、历史数据特征等部分组成。排序特征构造步骤的作用是将上述特征组织成与模型训练模块中

使用的训练集特征相同的形式，以便使用训练得到的算法模型进行排序打分。

模型打分步骤分为两个阶段：粗排阶段和精排阶段。待排序候选项集合首先由线性算法模型构成的粗排模型进行打分，得到排序预测得分后，取得分最高的前 K 个候选项送入由融合模型构成的精排模型，由精排模型进行打分并得到最终的排序序列。使用这种结构设计的原因是模型训练模块得到的融合模型的结构比较复杂，虽然候选集生成步骤过滤了大部分不相关资源，但是通常情况下得到的待排序候选项的数量仍然较多，使用融合模型对每一个候选项进行打分的计算量偏大，无法满足在线排序等使用场景的要求。而通过使用这种分层排序的模型结构，由粗排模型对待排序候选项集合进行了二次过滤，进一步减少了候选项的数量，再由精排模型控制最终的排序质量，保证了模型对排序效果和计算效率的兼顾。

排序结果输出步骤负责根据排序应用场景的特点将结果以合适的形式进行呈现。例如将全部待排序候选项按照其排序得分高低组织成序列输出，或将得分最高的候选项单独输出等。

5.2 排序模型在地铁单边交易处理中的应用

5.2.1 问题描述及分析

地铁单边交易数据，指进站或出站信息存在缺失的交易记录，是地铁运行过程中不可避免会产生的一类异常交易数据。为了保证地铁运营的有序进行，需要对单边交易数据进行合理的处理。单边交易数据因为进出站信息的缺失，无法进行准确计费，目前通常的处理方法是按照最长距离收取费用，然而这种方法容易导致用户的经济损失，引发纠纷或投诉。为了改进地铁单边交易的处理方法，减少乘客和地铁运营公司的损失，本课题应用所提出的基于深度学习的排序模型，利用历史交易数据和单边交易记录中的已知信息对缺失站点信息进行预测，给出站点或票价的预测概率序列，为业务系统提供参考信息，以确定适当的计费方式。

地铁单边交易数据处理的核心问题是对交易的应扣费额进行预测。针对该问题存在两种解决思路，第一种思路是先对缺失的进出站信息进行预测，再根据预测结果计算应扣票价；第二种思路是以应扣费额作为目标训练模型，直接输出单边交易的应扣费额在各档费用上的分布概率。以站点作为预测目标的方法能够给出计费的依据，预测结果更具有说服力，但存在的主要问题是站点数量偏多，每个站点对应的训练样本数量偏少且存在严重的样本不平衡问题，因此预测准确率相比于以应扣费额作为预测目标的方法较低；以应扣费额作为预测目标的方法则恰恰相反，因为票价的种类较少，每种票价能够对应更多数量的训练样本且样本在不同票价类别中的分布更加均衡，这使得针对应扣费额的预测具有更高的准确率，同时，应扣费额还可以作为数值型预测

目标，使用回归分析的方法进行估计。

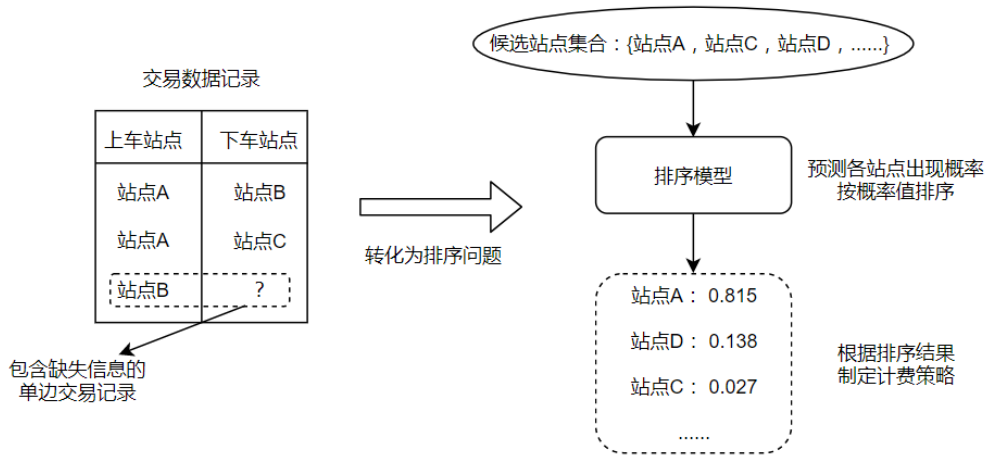


图 5-4 基于排序模型的站点预测

如图 5-4 所示，本课题将该问题转化为排序问题，使用历史交易数据训练得到排序模型，对于包含缺失信息的单边交易记录，由排序模型预测各个站点的概率分布序列，并根据排序结果制定计费策略。下面详细介绍对于该问题的处理过程，并对相关实验结果进行分析。

5.2.2 基于排序模型的地铁单边交易处理方法

问题的处理过程主要分为训练集构造、排序模型训练和效果测试等步骤。其中，训练集的数据来源为广州地铁交易数据记录，共包含 3445749 条样本。将全部样本数据按照 1: 8: 1 的比例进行切分，分别用于历史行为特征学习、排序模型训练和模型效果测试。如图 5-5 所示，原始数据记录中包含订单号、手机号、订单时间、订单金额、订单状态、进出站站点代码和站点名称等信息。

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	手机号	订单号	订单来源	商品编码	订单金额 (分)	退款金额 (分)	下单时间	支付时间	票卡类型	订单状态	票单价	取票数量	上车线路代码	上车站点代码	下车线路代码	下车站点代码	上车站点名称	下车站点名称
2	1.3E+10	0420171	0	QRCD_A	200	200	2017/7/17 19:55	2017/7/17 20:03	2	3	200	1	3	305	3	306	广州塔	春村
3	1.3E+10	0420171	0	QRCD_A	300	300	2017/8/1 18:15	2017/8/1 18:15	2	10	300	1	2	209	5	505	江南西	西村
4	1.3E+10	0420171	1	QRCD_J	300	300	2017/7/4 18:24	2017/7/4 18:25	2	11	300	1	2	216	2	210	三元里	市二宫
5	1.3E+10	0420171	1	QRCD_J	200	200	2017/7/9 14:50	2017/7/9 14:51	2	11	200	1	2	216	2	218	白云公园	白云公园
6	1.3E+10	0420171	1	QRCD_J	400	400	2017/7/9 17:48	2017/7/9 17:48	2	11	400	1	2	218	2	210	白云公园	市二宫
7	1.3E+10	0420171	1	QRCD_J	300	300	2017/7/9 20:08	2017/7/9 20:09	2	11	300	1	2	215	2	210	广州火车站	市二宫
8	1.3E+10	0420171	1	QRCD_J	600	600	2017/7/13 10:51	2017/7/13 10:51	2	11	600	1	2	215	4	410	广州火车站	大学城北
9	1.3E+10	0420171	1	QRCD_J	300	300	2017/7/13 21:59	2017/7/13 21:59	2	11	300	1	2	210	2	215	市二宫	广州火车站
10	1.3E+10	0420171	1	QRCD_J	300	300	2017/7/14 12:53	2017/7/14 12:53	2	11	300	1	2	215	5	510	广州火车站	动物园
11	1.3E+10	0420171	1	QRCD_J	300	300	2017/7/14 17:52	2017/7/14 17:52	2	11	300	1	5	510	5	506	动物园	广州火车站
12	1.3E+10	0420171	1	QRCD_J	300	300	2017/7/15 11:41	2017/7/15 11:41	2	11	300	1	2	215	2	210	广州火车站	市二宫
13	1.3E+10	0420171	1	QRCD_J	300	300	2017/7/15 16:16	2017/7/15 16:16	2	11	300	1	2	210	2	215	市二宫	广州火车站
14	1.3E+10	0420171	1	QRCD_J	200	200	2017/7/16 15:03	2017/7/16 15:03	2	11	200	1	2	214	2	215	越秀公园	广州火车站
15	1.3E+10	0420171	1	QRCD_J	300	300	2017/7/17 11:38	2017/7/17 11:39	2	11	300	1	2	215	2	210	广州火车站	市二宫

图 5-5 地铁交易数据记录

训练集构造步骤的主要流程如图 5-6 所示。首先对原始数据记录进行预处理。原始数据记录中包含已取消和处于退款流程中的订单记录，这部分订单记录大多是由用户的误操作造成，属于噪声数据，需要从数据集中剔除。此外，排序模型将在已知交易记录包含的信息的条件下，各个站点作为下车站点的概率作为预测目标，而交易订单数据记录中只包含正样本，而不包含负样本。一种添加负样本的方法是将除了上车

站点和真实下车站点之外的全部站点作为负样本，但是考虑到站点数量众多，这种方法会导致严重正负样本不平衡问题，同时也没有考虑用户出行习惯的差异性。据此，本课题使用对全部站点进行欠采样的方法添加负样本，具体做法为根据用户的历史交易记录，统计出现频率最高的 K 个下车站点，对于历史记录数量不足的情况，使用基于用户相似度聚类的方法填补数据，最后将结果中除上车站点和真实下车站点之外的站点作为负样本加入到样本集合中。

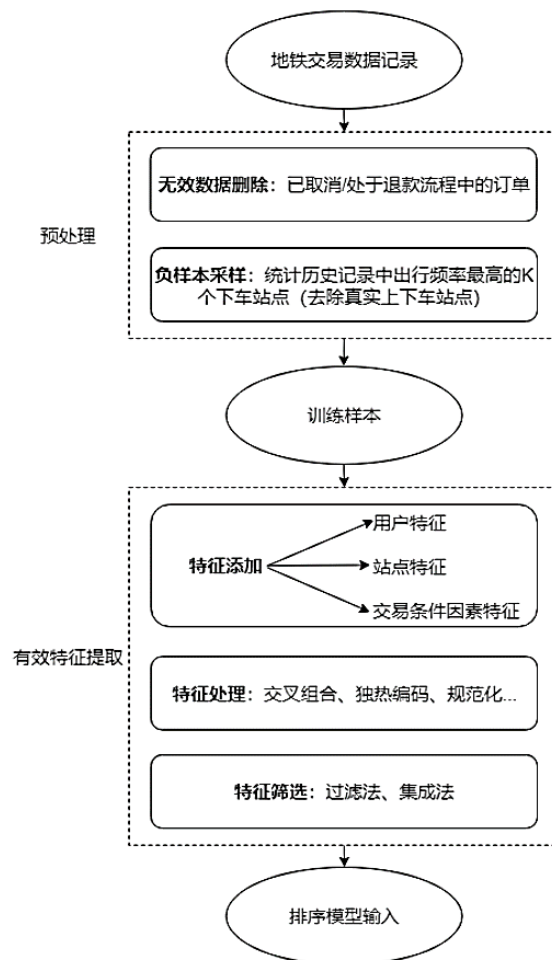


图 5-6 地铁单边交易数据训练集构造

之后对样本数据进行特征抽取。将原始数据记录中的订单手机号、订单时间、上下车站点代码提取为单特征，并进行特征扩充和统计特征添加。添加的特征分为用户特征、站点特征及交易条件因素特征三大类，具体特征包括：

（1）用户特征：包括手机号，归属地，购票频率，是否本地居住，以及由用户历史交易行为数据编码得到的历史行为特征向量；

（2）站点特征：包括经纬度等地理位置信息，站点周边的餐饮商家分布特征及商家平均团购订单数量；

（3）交易条件因素特征：包括交易当天的天气、温度、降水、风速等气象条件特征，并从订单时间中抽取出工作日/周末、月份、星期、小时等特征。

得到经过扩充的特征向量后,对单特征进行交叉组合,分别对类别型特征和数值型特征进行独热编码和规范化处理,并使用过滤法和集成法对各个维度特征进行有效性检验和特征筛选,得到最终的模型输入特征向量。

最后,使用由上述处理得到的训练集训练深度神经网络模型、广义线性模型和梯度提升树模型,并对这三个基础模型进行模型融合。训练完成后,使用测试集对排序模型的预测效果进行实验验证。

5.2.3 实验设计及结果分析

对于测试数据集中的每一条样本,使用模型的排序模块对其下车站点进行预测。首先生成待排序候选项集合,在该问题中,候选项集合为除上车站点外的全部站点集合。之后将各候选站点的与样本记录已知信息拼接成符合模型输入特征格式的排序特征向量,并使用两层结构的排序模型对各站点进行打分。最后,使用预测准确率 Accuracy 和平均精度均值 MAP 两项指标对模型的预测效果进行评估。评价指标的公式定义为:

$$\text{Accuracy} = \frac{|\{s_i | s_i = y_i\}|}{|\{y_i\}|} \quad (5-1)$$

$$\text{MAP} = \frac{\sum \frac{1}{p_i}}{|\{y_i\}|} \quad (5-2)$$

其中, s_i 表示针对第 i 条测试数据模型预测概率最高的站点, y_i 表示该条数据对应的实际下车站点, p_i 表示实际下车站点在结果排序列表中的位置。

首先,实验验证了课题提出的模型融合方法的有效性及其排序模型的排序预测效果。实验选取逻辑回归模型、XGBoost 模型和神经网络模型与融合排序模型进行对比,实验结果如下表所示:

表 5-1 排序预测效果实验结果

算法模型	评价指标	
	Accuracy	MAP
逻辑回归	0.7862	0.8213
XGBoost	0.8115	0.9022
MLP	0.7991	0.8907
融合模型	0.8188	0.9094

各算法实验结果评价指标对比如图 5-7 所示。

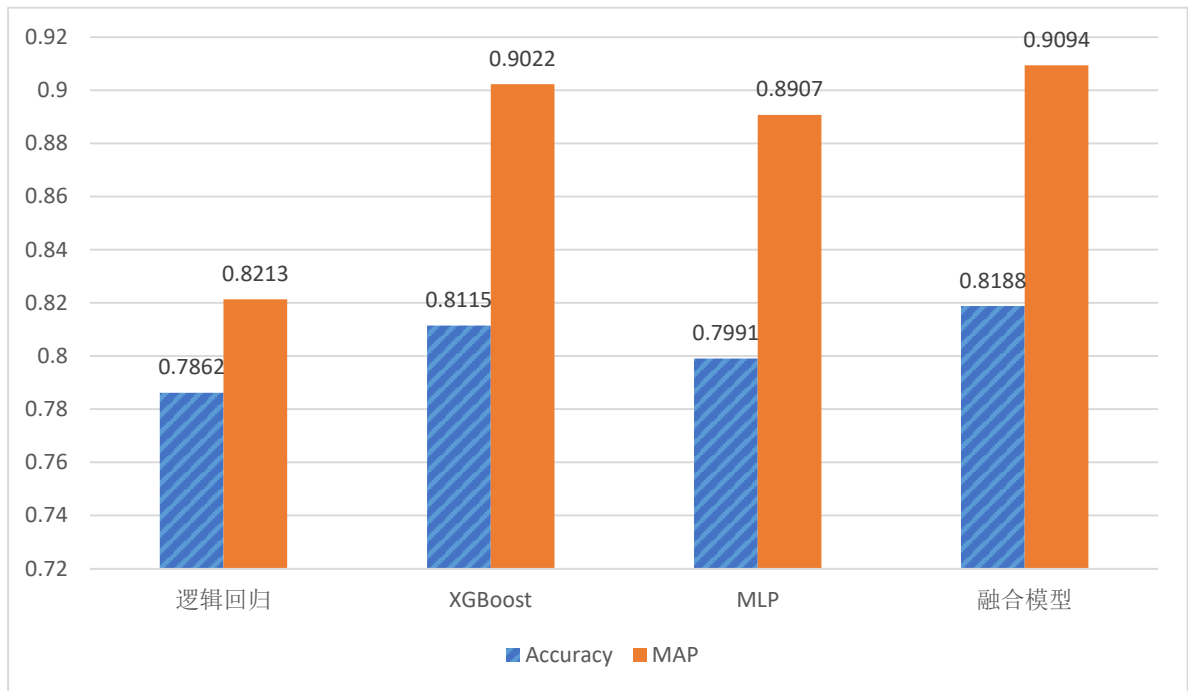


图 5-7 各算法排序预测效果对比

可以看出,在三种基础模型中,XGBoost 模型的预测效果最好,其准确度和 MAP 分别达到 0.8115 和 0.9056,在两项评价指标上均高于逻辑回归模型和神经网络模型。而融合模型的准确率和 MAP 指标均高于三种对比算法模型,说明本文提出的模型融合方法有效,能够在一定程度上提高预测的准确性。另外,从准确率和 MAP 指标的实验结果可以计算得出,对于每一条测试数据记录,其真实下车站点均处于融合模型预测结果序列的前两位,这表明融合模型具有较高的排序精准度。

其次,实验对课题提出的历史行为特征编码方法的有效性进行了测试。这部分实验使用了两组对照模型,分别对应不添加历史行为特征数据和使用按时序连接的历史行为特征向量的方法。实验使用准确率作为评价指标,详细结果如下表所示:

表 5-2 历史行为特征编码实验结果

算法模型	无历史行为特征	按时序连接		基于 RNN 编码	
	准确率	准确率	提升比	准确率	提升比
逻辑回归	0.7779	0.7805	0.33%	0.7862	1.07%
XGBoost	0.8096	0.8084	-0.15%	0.8115	0.23%
MLP	0.7898	0.7940	0.52%	0.7991	1.18%
融合模型	0.8111	0.8116	0.06%	0.8188	0.95%

实验结果对比如图 5-8 所示。

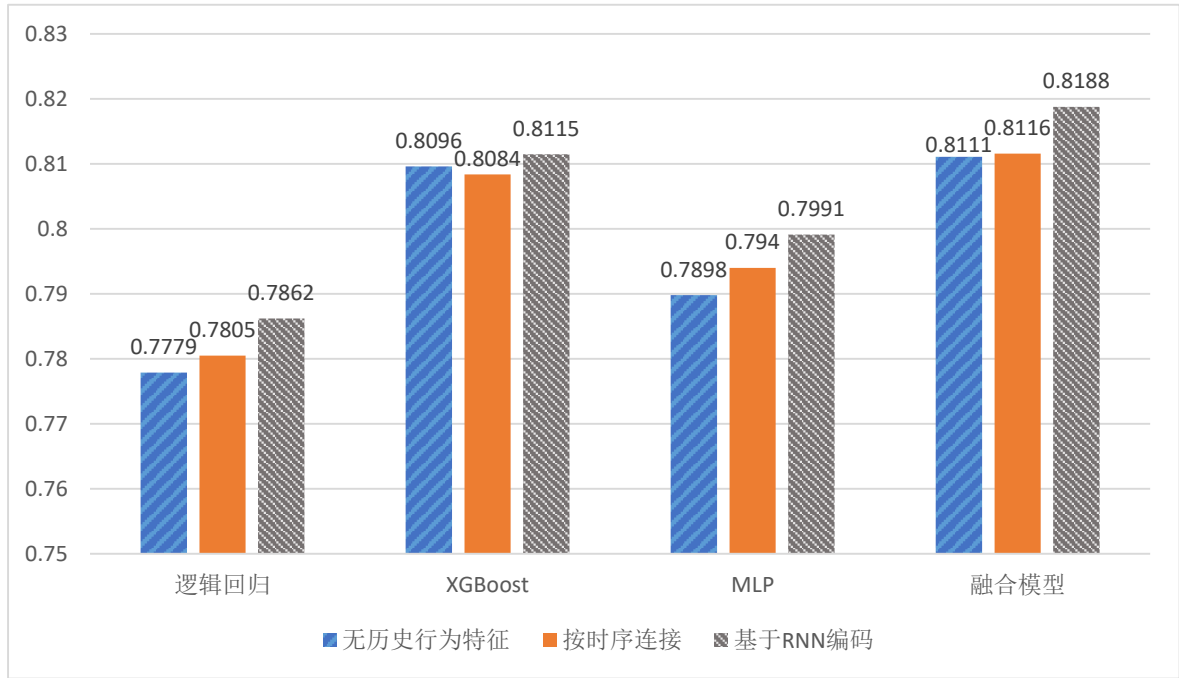


图 5-8 历史行为特征处理方法效果对比

实验结果显示,对于逻辑回归模型、神经网络模型和融合模型,添加历史行为特征后,模型的预测准确率均有所提升。其中神经网络模型的提升幅度最大:加入按时序连接的历史行为特征后,预测准确率提升了 0.52%;而加入基于 RNN 编码的历史行为特征后,准确率提升幅度达到 1.18%。对于融合模型,加入按时序连接的历史行为特征后模型预测准确率基本持平;加入基于 RNN 编码的历史行为特征后,预测准确率达到 0.8188,提升比为 0.95%。相比之下,XGBoost 模型的提升效果比较不明显,在加入按时序连接的特征向量后预测准确率甚至有所下降。这是因为梯度提升树类模型不适用于输入特征为高维度稀疏向量的情况,而广义线性模型和神经网络模型在高维度稀疏特征上的学习效果更好。实验结果表明添加基于 RNN 编码的历史行为特征能够在一定程度上提高各算法模型的预测准确率,验证了课题提出的历史行为特征编码方法的有效性。

最后,通过实验测试基于排序模型的地铁单边交易计费方法的效果。实验选取了三种计费策略,分别为按最远距离计算票价、按预测概率最高的站点计算票价和使用预测概率排序靠前的两个站点对应票价的平均值进行计费,实验结果如下表所示:

表 5-3 计费策略实验结果

计费策略	总费用(元)	偏差
实际站点	1480076.00	0
按最远距离	3910805.00	+164.23%
按预测概率最高站点计费	1606918.00	+8.57%
两站点费用取平均	1386683.00	-6.31%

实验结果表明传统基于最远距离计费的方法比较不合理,该方法的计费总额为应扣费用的 264.23%,与实际费用的偏差过大。使用按预测概率最高的站点计算票价的方法,总费用为实际费用的 108.57%,将偏差从 164.23%减少到 8.57%,效果十分明显。而使用预测概率排序靠前的两个站点对应票价取平均值的计费方式,计算得到的总费用与实际费用最接近,占实际费用的 93.69%,偏差为 6.31%。实验结果符合预期假设,证明了基于排序模型的地铁单边交易计费方法更具合理性。

5.3 本章小结

本章首先介绍了基于深度学习的排序模型的总体结构设计,并对模型训练模块和排序模块两个主要组成模块的结构和功能进行了详细说明。其中,模型训练模块按照其层次划分,分别对数据采集层、数据预处理层、特征抽取层和模型训练层四个层次的具体设计和功能进行了阐述;排序模块部分介绍了候选集生成、排序特征构造、模型打分和排序结果输出四个关键步骤各自的具体操作和实现方式。最后,将提出的基于深度学习的排序模型应用于地铁单边交易处理问题,从预测准确率等方面测试了排序模型的性能。实验结果显示该模型相比其他算法模型具有更好的预测效果,基于循环神经网络模型的历史行为数据特征编码方法也能够在一定程度上提升模型的预测准确率,同时与传统处理方式相比,基于该排序模型的地铁单边交易处理方法减少了计费偏差,更具合理性。

第六章 总结与展望

6.1 论文总结

近年来,深度学习成为学术研究的热点方向,取得了一系列瞩目的研究成果。深度神经网络模型因其所具有的独特优势,被广泛应用于解决计算机视觉和自然语言处理等领域中的相关问题。同时,Google、Facebook 等科技公司也开始尝试在推荐系统、在线广告等业务场景中应用深度学习技术。在此背景下,本文提出了一种基于深度学习的排序模型,给出了模型的总体结构设计和各组成模块的具体功能设计,并将该模型应用于处理地铁单边交易数据记录,验证了模型的有效性和实用性。相比于其他方法,该模型具有更高的预测准确率和排序效果,同时基于该排序模型的地铁单边交易处理方法相比于传统计费方法也更具合理性。本课题研究解决的关键问题,归纳总结如下:

(1) 设计并实现了包含深度神经网络结构的融合排序打分模型,优化了排序效果。本课题根据相关实验结果,利用 Boosting、Stacking 等多种模型融合方法,设计了包含深度神经网络模型、广义线性模型和梯度提升树模型的融合排序打分模型。经过实验测试,相比于单一算法模型,融合模型具有更高的预测准确率和排序精准度。

(2) 提出了一种基于循环神经网络的历史行为数据编码表示方法。如何有效利用历史行为数据,从中提取出供算法模型学习的有价值的特征数据是许多机器学习任务中需要解决的重点难点问题之一。本课题提出的基于循环神经网络的历史行为数据编码表示方法,解决了此前历史行为数据处理方法存在的特征向量长度不固定和信息丢失等问题。实验结果显示,在添加由此方法编码得到的历史行为特征后,模型的预测准确率有所提升,证实了该方法的可行性和有效性。

(3) 针对排序模型的训练方法,本课题根据具体应用场景的训练数据特点,分别选用 Pointwise 方法和 Listwise 方法。这使得排序模型可以有针对性的利用日志数据中的显式反馈和隐式反馈,应用场景更加广泛,模型的通用性和可扩展性更强。

(4) 排序模型使用了分层结构设计,实现了排序精准度和计算效率的平衡。将排序模型划分为粗排层和精排层,并通过参数控制粗排层的排序候选项过滤比例。这种结构设计综合考虑了排序的排序效果和计算复杂度,增强了排序模型的实际可用性。同时,使用参数控制中间输出候选项数量的方法也使得模型结构更加灵活。

(5) 本课题将排序模型应用于地铁单边交易处理,验证了模型的实用性。实验结果显示,模型对缺失站点信息的预测准确率达到 81.88%,排序评价指标 MAP 达到 0.9094。同时,基于排序模型的地铁单边交易处理方法将计费偏差从传统方法的

164.23%降低到 6.31%，这表明采用该方法能够显著提高计费的合理性，具有较强的实际应用价值。

综上所述，本课题所提出的基于深度学习的排序模型具有较好的排序效果和较高的计算效率，实验表明基于该排序模型的地铁单边交易处理方法更加合理，验证了该模型的有效性和实用性。同时模型也具备较强的通用性和扩展性，可以推广应用在推荐系统等业务场景中，具有较高的实际应用价值。

6.2 工作展望

本课题提出的基于深度学习的排序模型具有较高的排序精准度和实用性，但是在一些方面还存在可改进优化的空间，需要进行进一步的研究。可优化问题和可能的解决方法具体总结如下：

(1) 针对历史行为数据编码模型的优化。课题提出的基于循环神经网络的历史行为数据编码模型在隐含层中使用了 GRU 单元，在后续工作中将尝试使用 LSTM、双向 RNN 等多种模型结构，并通过实验对比分析编码效果。此外，如何确定最优的隐含层神经单元数量，即历史行为特征向量的维度，也是一个需要研究解决的问题。

(2) 冷启动问题有待解决。历史行为特征编码依赖于样本中的历史行为数据记录，当这部分数据不存在或数量较少时，历史行为特征对排序模型排序精准度的提升效果将会下降。后续改进方案考虑采用聚类方法，即利用其他维度特征计算用户间的相似度，对用户进行聚类，再将同类用户的历史行为记录作为编码模型的输入。

(3) 在地铁单边交易处理方法中尝试更多的计费策略。根据当前实验结果，采取排序预测结果前两位的站点对应费用平均值的计费策略最贴近实际费用，后续工作中将尝试使用如基于预测概率值线性加权等更多的费用计算策略，进一步减少计费偏差。

参考文献

- [1] 许可. 卷积神经网络在图像识别上的应用的研究[D]. 浙江大学, 2012.
- [2] 胡新辰. 基于 LSTM 的语义关系分类研究[D]. 哈尔滨工业大学, 2015.
- [3] Covington P, Adams J, Sargin E. Deep Neural Networks for YouTube Recommendations[C]. ACM Conference on Recommender Systems. ACM, 2016:191-198.
- [4] Cheng H T, Koc L, Harmsen J, et al. Wide & Deep Learning for Recommender Systems[J]. 2016:7-10.
- [5] Tie-Yan Liu. Learning to Rank for Information Retrieval[J]. Acm Sigir Forum, 2009, 41(2):58-62.
- [6] Hinton G E, Osindero S, Teh Y W. A Fast Learning Algorithm for Deep Belief Nets[J]. Neural Computation, 2006, 18(7):1527.
- [7] Rosenblatt F. The perceptron: A probabilistic model for information storage and organization in the brain [M]. Neurocomputing: foundations of research. MIT Press, 1988:386-408.
- [8] M.W Gardner, S.R Dorling. Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences[J]. Atmospheric Environment, 1998, 32(14–15):2627-2636.
- [9] 陈先昌. 基于卷积神经网络的深度学习算法与应用研究[D]. 浙江工商大学, 2014.
- [10] Werbos P J. The Roots of Backpropagation: From Ordered Derivatives to Neural Networks and Political Forecasting[J]. 1994.
- [11] Krizhevsky A, Sutskever I, Hinton G E. ImageNet classification with deep convolutional neural networks[J]. Communications of the ACM, 2013, 60(2):2012.
- [12] 黄华强. 基于深度神经网络的短时交通流量预测算法研究[D]. 厦门大学, 2016.
- [13] Hubel D H, Wiesel T N. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex.[J]. Journal of Physiology, 1962, 160(1):106.
- [14] Lawrence S, Giles C L, Tsoi A C, et al. Face recognition: a convolutional neural-network approach[J]. IEEE Transactions on Neural Networks, 1997, 8(1):98.
- [15] 李思琴. 基于深度学习的搜索广告点击率预测方法研究[D]. 哈尔滨工业大学, 2015.
- [16] 张效荣. 基于卷积神经网络的深度学习算法与应用研究[D]. 西安电子科技大学,

- 2015.
- [17] Werbos P J. Backpropagation through time: what it does and how to do it[J]. Proceedings of the IEEE, 1990, 78(10):1550-1560.
- [18] Hochreiter S, Schmidhuber J. Long short-term memory.[J]. Neural Computation, 1997, 9(8):1735-1780.
- [19] Chung J, Gulcehre C, Cho K H, et al. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling[J]. Eprint Arxiv, 2014.
- [20] Koutník J, Greff K, Gomez F, et al. A Clockwork RNN[J]. Computer Science, 2014:1863-1871.
- [21] Yao K, Cohn T, Vylomova K, et al. Depth-Gated LSTM[J]. Computer Science, 2015.
- [22] TensorFlow 官方网站[EB/OL]. <https://www.tensorflow.org/>.
- [23] Torch 官方网站[EB/OL]. <http://torch.ch/>.
- [24] Caffe 官方网站[EB/OL]. <http://caffe.berkeleyvision.org/>.
- [25] Theano 官方网站[EB/OL].<http://deeplearning.net/software/theano/>.
- [26] Keras 中文文档[EB/OL]. <http://keras-cn.readthedocs.io/en/latest/>.
- [27] PAGE,L. The PageRank Citation Ranking : Bringing Order to the Web, Online manuscript[J]. <http://www-db.stanford.edu/~backrub/pageranksub.ps>, 1998, 9(1):1-14.
- [28] Kleinberg J M. Authoritative Sources in a Hyperlinked Environment[C]. Acm-Siam Symposium on Discrete Algorithms. Society for Industrial and Applied Mathematics, 1998:668-677.
- [29] Garcia-Molina H, Pedersen J. Combating web spam with trustrank[C]. Thirtieth International Conference on Very Large Data Bases. VLDB Endowment, 2004:576-587.
- [30] 王鹏. 基于排序学习的商品搜索算法研究[D]. 重庆大学, 2015.
- [31] Li H. Learning to Rank for Information Retrieval and Natural Language Processing[C]. Morgan & Claypool, 2011:113.
- [32] Xia F, Liu T Y, Wang J, et al. Listwise approach to learning to rank: theory and algorithm[C]. International Conference on Machine Learning. ACM, 2008:1192-1199.
- [33] McMahan H B, Holt G, Sculley D, et al. Ad click prediction: a view from the trenches[C]. ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 2013:1222-1230.
- [34] Xia F, Liu T Y, Wang J, et al. Listwise approach to learning to rank: theory and algorithm[C]. International Conference on Machine Learning. ACM, 2008:1192-1199.
- [35] Rendle S. Factorization Machines[C]. IEEE International Conference on Data Mining.

- IEEE Computer Society, 2010:995-1000.
- [36] Jerome H. Friedman. Greedy Function Approximation: A Gradient Boosting Machine[J]. The Annals of Statistics, 2001, 29(5):1189-1232.
- [37] 李航. 统计学习方法[M]. 清华大学出版社, 2012.
- [38] Chen T, Guestrin C. XGBoost: A Scalable Tree Boosting System[J]. 2016:785-794.
- [39] 徐宝文. 搜索引擎与信息获取技术[M]. 清华大学出版社, 2003.
- [40] 单栋栋. 搜索引擎中索引剪枝的研究[D]. 北京大学, 2013.
- [41] Valizadegan H, Jin R, Zhang R, et al. Learning to Rank by Optimizing NDCG Measure[C]. Advances in Neural Information Processing Systems 22, Conference on Neural Information Processing Systems 2009. Proceedings of A Meeting Held 7-10 December 2009, Vancouver, British Columbia, Canada. DBLP, 2009:1883-1891.
- [42] 王育民. 信息论与编码理论[M]. 高等教育出版社, 2013.
- [43] 曾春, 邢春晓, 周立柱. 基于内容过滤的个性化搜索算法[J]. 软件学报, 2003, 14(5):999-1004.
- [44] Yu P S, Yu P S, Yu P S. Joint Deep Modeling of Users and Items Using Reviews for Recommendation[C]. Tenth ACM International Conference on Web Search and Data Mining. ACM, 2017:425-434.
- [45] 机器学习知识库《机器学习算法中的过拟合与欠拟合》[EB/OL]. <http://lib.csdn.net/article/machinelearning/68441>.
- [46] 李贤慧, 余正涛, 魏斯超, 等. 基于 Listwise 的深度学习专家排序方法[J]. 模式识别与人工智能, 2015, 28(11):000976-982.
- [47] Chen W, Liu T Y, Lan Y, et al. Ranking Measures and Loss Functions in Learning to Rank[C]. Advances in Neural Information Processing Systems 22:, Conference on Neural Information Processing Systems 2009. Proceedings of A Meeting Held 7-10 December 2009, Vancouver, British Columbia, Canada. DBLP, 2009:315-323.
- [48] Burges C J C. From ranknet to lambdarank to lambdamart: An overview[J]. Learning, 2010, 11.
- [49] UCI 公开数据库[DB/OL]. <http://archive.ics.uci.edu/ml/index.php>.
- [50] Rendle S. Factorization Machines with libFM[J]. Acm Transactions on Intelligent Systems & Technology, 2012, 3(3):1-22.
- [51] Duchi J, Hazan E, Singer Y. Adaptive Subgradient Methods for Online Learning and Stochastic Optimization[J]. Journal of Machine Learning Research, 2011, 12(7):257-269.
- [52] Zeiler M D. ADADELTA: An Adaptive Learning Rate Method[J]. Computer Science,

- 2012.
- [53] Kingma D P, Ba J. Adam: A Method for Stochastic Optimization[J]. Computer Science, 2014.
- [54] Cho K, Van Merriënboer B, Gulcehre C, et al. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation[J]. Computer Science, 2014.
- [55] 范先念. 数据不平衡分类问题研究[D]. 中国科学技术大学, 2011.

附录

缩略语	全称	描述
DL	Deep Learning	深度学习
CNN	Convolutional Neural Network	卷积神经网络
RNN	Recurrent Neural Network	循环神经网络
LSTM	Long Short-Term Memory	长短时记忆网络
GRU	Gated Recurrent Unit	门控循环单元
LTR	Learning to Rank	排序学习
LR	Logistic Regression	逻辑回归
GBDT	Gradient Boosting Decision Tree	梯度提升决策树
MLP	Multilayer Perceptron	多层感知机
DNN	Deep Neural Network	深度神经网络
BP	Backpropagation	反向传播
BPTT	Backpropagation Through Time	随时间反向传播算法
FM	Factorization Machine	分解机
MAE	Mean Absolute Error	平均绝对误差
MSE	Mean Square Error	均方误差
RMSE	Root Mean Square Error	均方根误差
MAP	Mean Average Precision	平均精度均值
NDCG	Normalized Discounted Cumulative Gain	归一化折损累积增益

攻读学位期间发表的学术论文目录

- [1] Design and Implementation of a Multidimensional Data Retrieval Sorting Optimization Model[C]. IEEE International Congress on Big Data. IEEE, 2016. (三作)
- [2] Predicting Stock Using Microblog Moods[J]. China Communication, 2016, 13(8):244-257. (三作)
- [3] PPQAR: Parallel PSO for Quantitative Association Rule Mining[C]. IEEE International Conference on Big Data and Smart Computing. IEEE, 2018. (二作, 学生一作)