

苏州大学学位论文独创性声明

**本人郑重声明：所提交的学位论文是本人在导师的指导下，独立进行研究工作所取得的成果。除文中已经注明引用的内容外，本论文不含其他个人或集体已经发表或撰写过的研究成果，也不含为获得苏州大学或其它教育机构的学位证书而使用过的材料。对本文的研究作出重要贡献的个人和集体，均已在文中以明确方式标明。本人承担本声明的法律责任。**

**论文作者签名： 日 期：**

苏州大学学位论文使用授权声明

**本人完全了解苏州大学关于收集、保存和使用学位论文的规定，即：学位论文著作权归属苏州大学。本学位论文电子文档的内容和纸质论文的内容相一致。苏州大学有权向国家图书馆、中国社科院文献信息情报中心、中国科学技术信息研究所（含万方数据电子出版社）、中国学术期刊（光盘版）电子杂志社送交本学位论文的复印件和电子文档，允许论文被查阅和借阅，可以采用影印、缩印或其他复制手段保存和汇编学位论文，可以将学位论文的全部或部分内容编入有关数据库进行检索。**

**涉密论文□**

**本学位论文属 在 年 月解密后适用本规定。**

**非涉密论文□**

**论文作者签名： 日 期：**

**导师签名： 日 期：**

基于宽度&深度学习的离网预测模型

中文摘要

随着4G网络的普及以及互联网的快速发展，电信运营商联合其他公司推出了各种各样的优惠活动吸引用户。这些活动在吸引越来越多新用户入网的同时也导致了老用户加速离网，这种加速离网的现象已经引起了业界的广泛重视。如何在用户离网之前准确预测出用户的离网倾向，从而采取一系列措施挽留用户，减少运营商损失，成为了一个研究的热点。

针对电信用户的离网预测问题，业界实验了一系列的机器学习算法。由于近期深度学习技术的快速发展，越来越多的研究开始在离网预测任务中尝试各种神经网络模型。针对现有的基于神经网络的离网预测模型存在的一些问题，本文在电信运营商提供的真实用户数据上，设计构造了一种基于宽度&深度学习的神经网络框架用于离网预测。本文的工作主要有以下三个内容：

1）本文提出一种基于宽度&深度学习的离网预测模型框架。加入了交叉积转换的线性模型可以有效的记忆特征间的共现关系，而神经网络则可以自动生成大量复杂的特征组合，具有更好的泛化能力。然而当神经网络的输入包含离散型特征时，可能会存在过度泛化的问题。我们在离网预测任务上借鉴宽度&深度学习的思想，线性模型作为宽度模型，神经网络作为深度模型，将两部分模型组合起来共同训练，从而将两者的优势结合。实验证明，加入了线性模型的神经网络比不加入线性模型的神经网络在PR-AUC上最高可提升5.39%。

2）目前应用于离网预测任务的神经网络模型层数都比较浅，当数据量较小时可以比较好的拟合数据。然而随着数据量的增加，仍然使用浅层模型就会导致比较严重的欠拟合问题。本文使用一种较深层的卷积神经网络结构，并在卷积层中尝试加入短连接或门机制来充分拟合大量数据。本文使用的深层的神经网络与作为baseline的浅层网络相比，在PR-AUC上最高可提升72.33%。

3）现有的大数据驱动的离网预测模型通常只选择一种时间粒度聚合时序特征，然后在静态特征和聚合的时序特征上探索各种机器学习算法的效果。这样的方法只考虑了模型对分类性能的影响，没有充分考虑数据的作用。本文尝试特征级融合和决策级融合两种方式来同时利用月和日两种时间粒度聚合的特征，在聚合的特征上做一系列的模型训练和模型融合，来进一步提高模型的性能。实验证明，同时进行特征级融合和决策级融合的方法在PR-AUC上最多可以比不进行任何融合的模型提升21.94%。

**关键词：**离网预测，神经网络，深度学习，多粒度，时序数据

作 者：石鸿斌

指导教师：曾 嘉

**Churn Prediction Model Based on Wide & Deep Learning**

Abstract

With the popularity of the 4G network and the rapid development of the Internet, telecom operators have launched a variety of promotions to attract users together with other companies. These activities have attracted more and more new users in the network and also causes the accelerated depature of old users from the network. This phenomenon of accelerating off-line has attracted widespread attention in the industry. How to accurately make churn prediction before users leave the network, so as to take a series of measures to retain the users and reduce the loss of the operator has become a focus of research.

To solve the churn prediction problem of telecommunication users, the industry has experimented with a series of machine learning algorithms. Due to the rapid development of the deep learning technologies, more and more researches have utilized neural network models in the task of churn prediction. To solve the problems existing in churn prediction models based on neural network, we design and construct a neural network framework based on wide & deep learning in this paper which is based on the real user data provided by telecom operators. The work of this paper mainly has the following three contents:

1)This paper proposes a churn prediction framework based on wide & deep learning. Linear model with cross product transform can effectively memorize the co-occurrence relationships between features, while neural network can automatically generate a large number of complex feature combinations to get better generalization capability. However, when the neural network’s input contains discrete features, there may be an issue of over-generalization. So we learn the idea of wide & deep learning in churn prediction. The linear model is used as wide model and the neural network is used as deep model. The two parts are combined and trained together to combine the benefits of linear model and neural network. Experiments results show that the performance of the neural network with linear model can be increased by up to 5.39% on PR-AUC compared with the neural network without linear model.

2) At present, the layers of the neural network models applied to the churn prediction are relatively shallow, and can fit data well only when the amount of the data is small. However, the use of shallow models will lead to serious under-fitting problems as the amount of data increase. To solve the problem, we propose a deep convolutional neural network structure and attempt to add a short link or gate mechanism in the convolutional layers to fit a large amount of data. Compared with the shallow network as baseline, the performance of our deep network can be increased by up to 72.33% on PR-AUC.

3)The existing data-driven churn prediction models usually select only one kind of time granularity to aggregate time series features and build models on static features and time series features using machine learning algorithms. Such approaches only consider the impact of the models on classification performance, and do not fully consider the role of data. In this paper, we attempt to use monthly and daily time granularity to aggregate features by feature-levle fusion and decision-level fusion, and make a series of model training and model ensemble based on the extracted features to further improve the performance of the model. Experimental results show that the performance of our method which simultaneously uses feature-levle fusion and decision-level fusion can be increased by up to 21.94% on PR-AUC.

**Keywords:** churn prediction, neural network, deep learning, multi-granularity , time series data

Written by Hongbin Shi

Supervised by Jia Zeng

目 录

[第一章 绪论 1](#_Toc513566404)

[1.1 研究背景与意义](#_Toc513566405) 1

[1.1.1 研究背景 1](#_Toc513566406)

[1.1.2 研究意义 1](#_Toc513566407)

[1.2 国内外研究现状 3](#_Toc513566408)

[1.2.1 构建特征 3](#_Toc513566409)

[1.2.2 构造分类器 3](#_Toc513566410)

[1.2.3 分类器集成 4](#_Toc513566411)

[1.3 目前存在的主要问题 4](#_Toc513566412)

[1.4 本文的主要工作 5](#_Toc513566413)

[1.5 论文组织结构 6](#_Toc513566414)

[第二章 相关理论及技术 7](#_Toc513566415)

[2.1 树型模型 7](#_Toc513566416)

[2.1.1 随机森林 7](#_Toc513566417)

[2.1.2 梯度提升树 8](#_Toc513566418)

[2.2 神经网络 8](#_Toc513566419)

[2.2.1 神经网络基础 9](#_Toc513566420)

[2.2.2 卷积神经网络 13](#_Toc513566421)

[2.2.3 残差网络 14](#_Toc513566422)

[2.2.4 门卷积网络 15](#_Toc513566423)

[2.3 集成学习 16](#_Toc513566424)

[2.4 本章小结 17](#_Toc513566425)

[第三章 实验数据准备及评价指标 18](#_Toc513566426)

[3.1 实验数据准备 18](#_Toc513566427)

[3.2 评价指标 23](#_Toc513566428)

[3.2.1 查准率和查全率 23](#_Toc513566429)

[3.2.2 AUC 23](#_Toc513566430)

[3.2.3 PR-AUC 24](#_Toc513566431)

[3.3 本章小结 24](#_Toc513566432)

[第四章 基于神经网络的离网预测模型 25](#_Toc513566433)

[4.1 网络结构框架 25](#_Toc513566434)

[4.1.1 过度泛化 26](#_Toc513566435)

[4.1.2 宽度模型 27](#_Toc513566436)

[4.1.3 深度模型 29](#_Toc513566437)

[4.1.4 宽度&深度模型拼接 31](#_Toc513566438)

[4.2 实验设置 32](#_Toc513566439)

[4.2.1 数据准备及评价指标 32](#_Toc513566440)

[4.2.2 实验环境 32](#_Toc513566441)

[4.2.3 模型训练参数设置 32](#_Toc513566442)

[4.2.4 对比实验 33](#_Toc513566443)

[4.3 实验结果与分析 35](#_Toc513566444)

[4.3.1 模型训练时间分析 37](#_Toc513566445)

[4.3.2 与深度/宽度模型比较 38](#_Toc513566446)

[4.3.3 与浅层神经网络模型比较 39](#_Toc513566447)

[4.3.4 与树型模型比较 41](#_Toc513566448)

[4.4 本章小结 42](#_Toc513566449)

[第五章 融合多粒度时序特征的离网预测模型 43](#_Toc513566450)

[5.1 基于特征级融合利用多粒度时序特征 43](#_Toc513566451)

[5.1.1 特征级融合 43](#_Toc513566452)

[5.1.2 特征级融合应用于离网预测模型 44](#_Toc513566453)

[5.2 基于决策级融合利用多粒度时序特征 45](#_Toc513566454)

[5.2.1 决策级融合 45](#_Toc513566455)

[5.2.2 决策级融合应用于离网预测模型 45](#_Toc513566456)

[5.3 基于特征级融合+决策级融合利用多粒度时序特征 46](#_Toc513566457)

[5.4 实验设置 46](#_Toc513566458)

[5.4.1 数据准备及评价指标 46](#_Toc513566459)

[5.4.2 实验环境 47](#_Toc513566460)

[5.4.3 模型训练参数设置 47](#_Toc513566461)

[5.4.4 对比实验 47](#_Toc513566462)

[5.5 实验结果分析 48](#_Toc513566463)

[5.5.1 特征级融合实验结果 49](#_Toc513566464)

[5.5.2 决策级融合实验结果 50](#_Toc513566465)

[5.5.3 特征级融合+决策级融合实验结果 51](#_Toc513566466)

[5.6 本章小结 52](#_Toc513566467)

[第六章 总结与展望 54](#_Toc513566468)

[6.1 工作总结 54](#_Toc513566469)

[6.2 工作展望 55](#_Toc513566470)

[参考文献 56](#_Toc513566471)

[攻读硕士学位期间主要的工作 62](#_Toc513566472)

[致 谢 63](#_Toc513566473)

1. 绪论
   1. 研究背景与意义
      1. **研究背景**

近年来，随着互联网技术和人工智能的发展，各个行业的存储的数据都出现了指数型的增长，数据的形式也越来越多元化，如图片、新闻、对话、出行轨迹、医疗数据等等。如何充分利用这些数据来提升自己的竞争力和利润，成为了大量企业研究的热点。新闻媒体可以通过研究新闻的内容，使新闻自动归类，减少人工归类成本；医院可以通过分析历史病历数据，辅助医生给病人更加个性化的治疗方案；共享单车公司可以根据自行车的行车轨迹，计算出更加合理的自行车投放地点。在电信行业，电信运营商们通过离网预测来利用海量数据帮助自己提高利润。离网预测主要通过现有的用户行为数据等数据，预测用户在未来的一段时间内是否会流失，它的目的主要是使运营商在用户可能离网前知道用户的意向，从而采取各种挽留手段来阻止用户流失，提高运营商的利润。

* + 1. **研究意义**

随着各个互联网公司和运营商的合作越来越多，电信运营商联合其他公司推出了各种各样价格优惠的新套餐，这些新套餐在吸引大量用户换新号码的同时也导致抛弃旧号码的用户越来越多。这种现象使用户流失问题越来越成为电信运营商亟待解决的问题，同时也引起了业界的广泛关注和研究。由于新套餐的资费普遍比老套餐便宜，这就使得即使用户没有换运营商，只是换新号码也会导致电信运营商的利润降低。研究表明，运营商成功发展一个新用户的成本是挽留现有用户的3倍以上，而采用合理的维系挽留策略挽留用户的成功率高于发展新用户的成功率[1-3]。可见，离网预测是一个很有意义的任务，对离网预测的研究可以使运营商的利润得到提升。本文在研究离网预测任务的基础上聚焦预付费活跃用户的离网。

对于电信运营商来说，用户通常分为两种：预付费用户和后付费用户。预付费指的是先缴费后使用，一旦余额降为0后就会停机；后付费指先使用后缴费，余额降为0后仍可继续使用。相比而言，后付费用户比较稳定，一般不会流失，而预付费用户则随时会流失，在离网预测任务中主要预测预付费用户的离网情况。

图1-1统计了上海某运营商的2/3G预付费用户在2015年9月到2016年8月的离网率。其中活跃用户指的是每月的通话时长、上网流量或发短信次数超过给定阈值的用户，非活跃用户则指未达到阈值的用户。图1-2统计了2015年9月到2016年8月所有用户中活跃用户和非活跃用户的占比，可以看到活跃用户在这12个月中的平均占比是63%，占总体用户中的大多数。由于非活跃用户较容易离网，通常可以根据规则直接判断出，不需要使用机器学习的模型，并且人数占比少，给运营商带来的利润没有活跃用户多，因此在本文中主要聚焦活跃用户的离网。

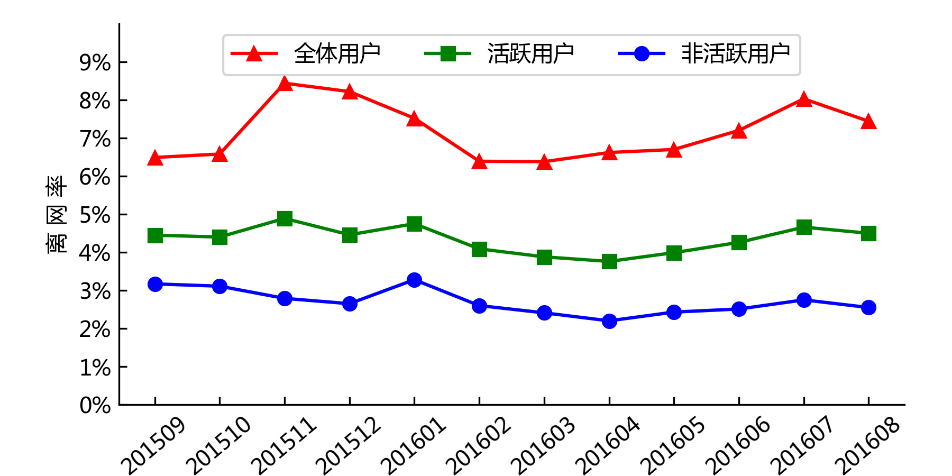


图1-1 2/3G预付费用户的离网率

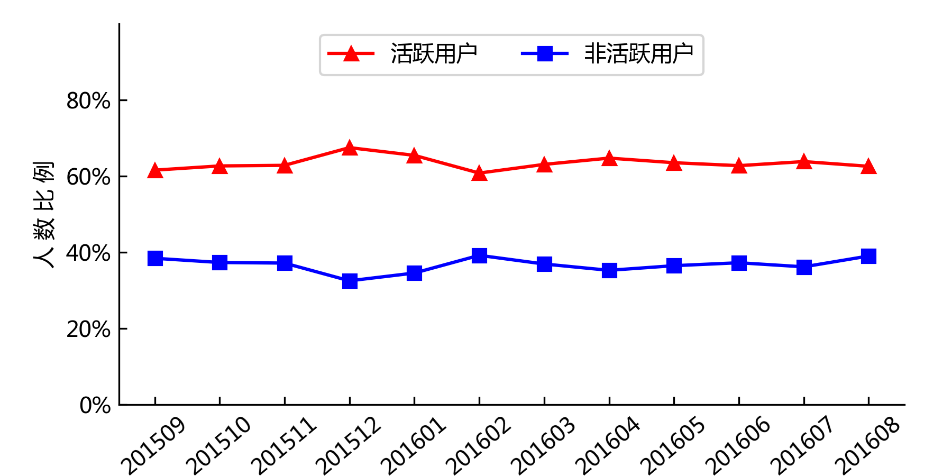


图1-2 2/3G活跃用户和非活跃用户的比例

综上所述，本文研究预付费活跃用户的离网预测是很有价值的。

* 1. 国内外研究现状

目前国内外的离网预测研究主要从三个方面提高性能：构建特征、构造分类器和分类器集成[4-5]。

* + 1. **构建特征**

在构建特征阶段主要有两个步骤：特征选取和时序特征聚合。大部分用户相关的特征来源于运营商的业务支持系统（BSS）中，主要包括通话详情记录（电话号码、开始结束时间、数据使用情况等）、账单记录（账户余额、缴费记录、每个用户的平均收入等）、用户个人信息（年龄、性别、职业、家庭住址等）、生命周期（入网时间、在网时长等）、社交网络信息[10-12]、购买历史（订购套餐等）、投诉记录[13]以及用户等级（是否是VIP）。各个运营商之间存储的数据格式和内容不一样，具体抽取哪些特征没有统一的标准。由于运营商数据库中的时序特征通常是秒级的，直接使用会导致数据量过大，通常需要对时序特征按某种时间粒度进行聚合。例如按月[6-7]为时间粒度，统计用户在一个月内发生的行为，如一个月的话费、通话量、短信量等。也有研究按日[8]或按2小时[9]等其他粒度来聚合特征。

* + 1. **构造分类器**

在分类器的选择中，不同的学者选用了不同的分类器。比如逻辑回归[14]-[16]、支持向量机[17-19]、决策树[20-22]、K近邻[23]和神经网络等。其中，随着深度学习在这几年的快速发展，神经网络越来越多的用于离网预测的研究和应用中。Prashanth等人[24]比较了常见的线性模型（逻辑回归等）和非线性模型（随机森林、神经网络等）在离网预测任务上的效果，发现非线性模型比线性模型的效果好。Castanedo[7]等人提出一种包含4个全连接层的神经网络，实验证明这种网络的AUC值高于随机森林和需要大量特征工程的模型。Wangperawong等人[8]在卷积层和全连接层后面加入Dropout来降低过拟合，与没有dropout的模型相比在测试集上的AUC值更高。Zaratiegui等人[9]提出一种具有2个卷积层、2个池化层和4个全连接层的卷积神经网络，发现这种网络与依赖人工做特征的机器学习算法相比有明显优势。Zhang等人[25]采用联合学习的方法将神经网络与线性模型结合，神经网络部分采用全连接的网络，实验证明这种方法训练的网络性能优于卷积神经网络、循环神经网络等常见的网络。

* + 1. **分类器集成**

集成学习是一种在现有模型基础上进一步提高性能的有效方法。在离网预测的研究中有支持向量机集成[1][26-27]、决策树集成[28-29]等。其中Olle等人[30]在一个包含2000个用户的数据集上进行了实验，观察到逻辑回归和感知机算法的组合可以得到最好的性能。Vafeiadis等人[31]在一个包含5000个样本的离网数据集上对不同的机器学习算法进行了综合的比较，发现使用AdaBoost算法融合多项式核的支持向量机效果最好。Lu等人[32]提出使用提升算法来改进用户离网预测模型的性能，实验结果证明提升算法可以从数据集中很好的区分出离网用户。Azeem等人[33]提出了一种基于模糊分类的离网预测模型，他们将构造的模糊分类器与神经网络、线性回归、决策树、支持向量机、AdaBoost、梯度提升树和随机森林做对比，发现模糊分类器优于其他分类器。Idris等人[34]提出了一种电信行业使用的智能离网预测系统，这种系统结合了基于过滤和基于包装的方法进行特征选择，并将随机森林、旋转树、支持向量机等分类器作为基分类器进行集成。

* 1. 目前存在的主要问题

近年来，越来越多的学者和工程师开始关注离网预测这个任务，离网预测已经成为了电信行业乃至其他相关行业一个关注的热点。逻辑回归、随机森林、支持向量机、神经网络等机器学习算法均已被尝试应用在了离网预测任务中。随着各种深度学习方法的快速发展，基于神经网络的离网预测模型已经渐渐成为了主流。这些模型与以往的模型相比性能获得了巨大的提升，但仍然存在着一些问题：

1. 容易过度泛化

当神经网络的输入中包含离散型特征时，我们通常会将特征的值转换为低维稠密的向量，然后再与连续型特征拼接，共同输入到网络。当模型在预测时出现之前较少出现或从未出现的值，就可能导致过度泛化的问题[35]，目前的离网预测研究中很少有人关注这个问题。

1. 模型的网络层数较浅

目前的基于神经网络的离网预测模型，无论是使用多层感知机、卷积神经网络还是循环神经网络，网络深度都比较浅，通常不超过5层，这样的网络在小数据集上可以有比较好的效果，但工业界中的数据量通常都比较大，随着样本的数量和样本的维度渐渐增大，浅层的网络会出现欠拟合的现象。

1. 特征的时间粒度单一

在目前的离网预测模型中，学者们都是首先选取一种时间粒度后，对特征按选取的粒度进行聚合，然后尝试使用各种机器学习算法对数据建模。学者们将过多的精力放在如何构建一个好的模型上，忽视了使用不同的时间粒度聚合特征对模型最终性能的影响。

* 1. 本文的主要工作

针对现有的离网预测研究中出现的一系列问题，本文在电信运营商提供的真实用户数据上，结合了现有数据的特点，给出了一种基于宽度&深度学习的离网预测模型框架，主要从以下两个方面改进现有的模型：

在模型的网络结构方面，针对问题1，我们将模型分为宽度模型和深度模型两部分。宽度模型部分为一个线性模型，线性模型可以通过交叉积转换的方式来记住特征间的共现关系，由于没有将特征的值转换为低维稠密向量的操作，因此不会产生过度泛化的问题。深度模型部分为一个包含多层的卷积神经网络模型，针对问题2，我们尝试使用了两种较深层的网络来代替现有的浅层网络。

在模型的输入特征方面，针对传统方法只选取一种时间粒度来聚合时序特征，本文提出同时使用月和日两种不同的时间粒度来聚合特征，并使用特征级融合和决策级融合来对在不同时间粒度的特征上训练的模型进行融合，来进一步提高模型的性能。

* 1. 论文组织结构

本文共分为6章，各章的安排如下：

第一章为绪论，首先介绍了离网预测的研究背景以及研究意义，然后介绍了研究现状和目前存在的主要问题，最后提出了本文的研究内容和文章组织结构。

第二章介绍了本文用到的神经网络结构的相关理论基础，以及树型模型和常用的集成学习方法。

第三章介绍了本文实验的实验数据准备和实验的评价指标。

第四章针对目前的基于神经网络的离网预测模型的不足，给出了一种新的基于神经网络的离网预测模型，并给出了模型的结构图。该模型由深度模型和宽度模型两部分组成，深度模型部分构造了一个较深层的卷积神经网络，宽度模型部分对部分离散型特征做交叉特征，作为深度模型的补充。

第五章探索了如何充分利用不同时间粒度的时序特征来进一步提高神经网络模型的性能。本章从特征级融合和决策级融合两个角度尝试了一系列可以融合的模型组合，并将这些方法的实验结果与基于单粒度特征的模型作比较，找出最好的融合方案。

第六章为对本文研究工作的总结，以及对未来工作的展望。

1. 相关理论及技术

本章主要给出树型模型（包括随机森林、梯度提升树）、神经网络以及集成学习的基础原理介绍。

* 1. 树型模型
     1. **随机森林**

Leo Breiman在2001年[36]提出了随机森林的概念。随机森林是一种以决策树为基学习器，使用Bagging[37]进行集成的算法。Bagging方法为每个基学习器构造不同的样本来训练，对于一个包含个样本的数据集，从中有放回的采样个样本得到新数据集，当样本数无穷大时，中的样本在中没有出现的概率是



即中约有63.2%的数据出现在。使用上面的方法采样出个，然后基于每个训练出一个基学习器，再将这些学习器融合。表2-1是Bagging算法的描述：

表2-1 Bagging算法

|  |
| --- |
| 算法：Bagging |
| 输入：训练集  基学习算法  训练轮数  输出： |
| for t=1,2,…,T  中无放回抽样得到 |

随机森林在Bagging的基础上加入了随机属性划分。假设样本一共有个不同的属性，基决策树在训练模型时候，随机选取其中的个属性，再从这个属性中选择一个最好的用来划分。通常选取。随机森林在同时加入了样本扰动和属性扰动后，显著提高了模型的泛化性能。

* + 1. **梯度提升树**

梯度提升树（Gradient Boosting Decision Tree, GBDT）[38]与随机森林一样，是一种基学习器为决策树的集成模型。与随机森林不同的地方是，随机森林使用Bagging来集成决策树，GBDT则使用Boosting集成决策树。Boosting的基本思想是：先训练一个初始的基学习器，根据基学习器在数据集上的表现调整数据集，然后基于调整后的数据集来训练下一个基学习器。GBDT在训练第棵树时，训练的目标值依赖于前棵树的输出。假设用来表示第棵树，表示第个样本的值，表示数的棵数，则第个样本的预测值可以表示为。对于GBDT第棵树的训练，我们有以下目标函数：



其中是损失函数，是正则项。我们以GBDT的一种实现：XGBoost[39]为例，将损失函数在处用泰勒公式展开得到：



其中和分别是一阶导和二阶导，定义为：



可以看到，在计算前棵树的预测值时需要首先知道前棵树的预测值。

* 1. 神经网络

在生物神经网络中，每个神经元与其他神经元相连，当它“兴奋”时，就会向相连的神经元发送化学物质，从而改变这些神经元内的电位；如果某种神经元的电位超过了一个“阈值”，那么它就会被激活，即“兴奋”起来，向其它神经元发送化学物质。人工神经元网络是对生物神经网络的一种模拟和近似，是由大量神经元通过相互连接而构成的自适应非线性动态网络系统。

对于一个监督学习的任务，当特征之间存在线性相关性或者对速度要求很高的时候，逻辑回归是一种特别适合的学习算法。但是当特征之间的关系比较复杂，并且可以接受较慢的训练速度的时候，神经网络就是一类有效的算法。

* + 1. **神经网络基础**

1943年，McCulloch等人[40]首次提出“M-P神经元模型”。在这个模型中，神经网络由多个神经元构成，神经元接收来自n个其他神经元传递过来的输入信号，这些输入信号通过带权重的连接进行传递，然后通过“激活函数”处理以产生神经元的输出。图2-1是单个神经元的结构图：

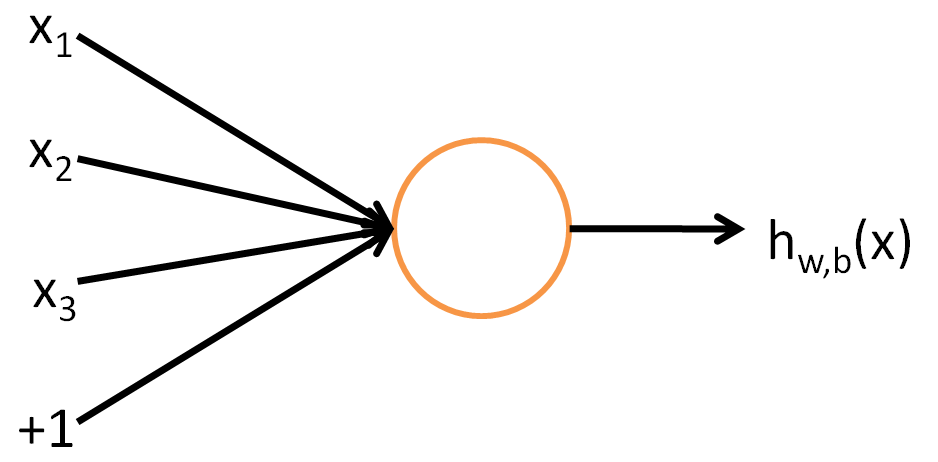


图2-1 单个神经元

从图中可以看到，这个神经元由3个输入、和组成，经过变换得到输出。

激活函数用来加入非线性因素，解决线性模型不能解决的问题。理想中的激活函数为阶跃函数，它将输入值映射为输出值“0”或“1”，然而，阶跃函数具有不连续、不光滑等不太好的性质，因此常用Sigmoid函数作为激活函数，它把可能在较大范围内变化的输入值挤压到(0,1)输出范围内。此外，常用的激活函数还有：Tanh函数、ReLU[41]函数、SoftPlus函数，这些激活函数的表达式见公式2-5到公式2-8，图形见图2-2到图2-5。

Sigmoid函数：



Tanh函数：



ReLU函数：



SoftPlus函数：



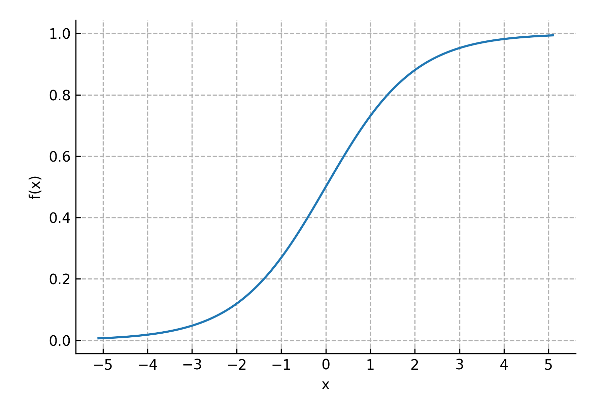
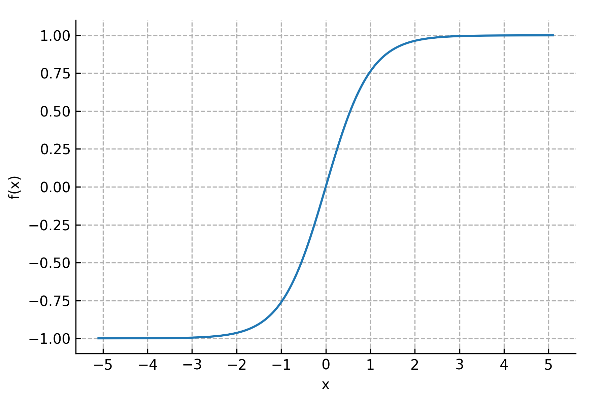
 

图2-2 Sigmoid函数 图2-3 Tanh函数

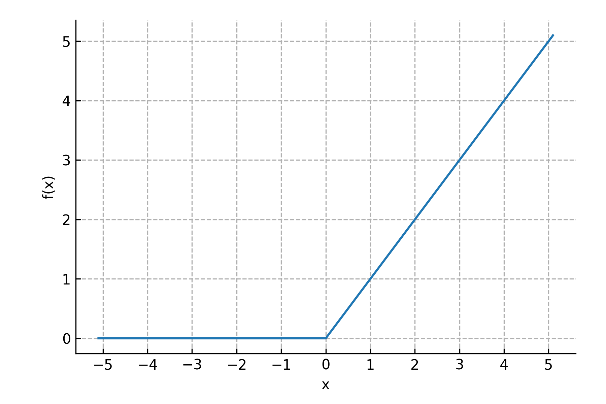
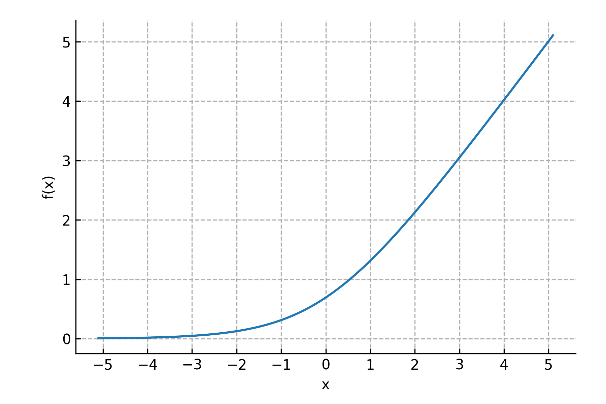
 

图2-4 ReLU函数 图2-5 SoftPlus函数

一个神经网络通过把许多简单的神经元连接在一起，这样神经元的输出就可以成为另一个神经元的输入。图2-6为一个含有一个隐层的神经网络：



图2-6 一个隐层的神经网络

上图中，这个神经网络由3个输入、和组成，“+1”表示偏置单元。代表输入层，仅接收外界输入，不进行函数处理；代表输出层，代表隐层，这两层包含功能神经元对信号进行加工，最终结果由输出层神经元输出。因此该神经网络包含三个输入单元，三个隐藏单元以及一个输出单元。

令表示第层单元与第层单元连接的权重，表示第层单元的偏置，因此该网络的参数包括()=()。令表示第层单元的激活值。对于，用表示第个输入。该神经网络的计算如下：









令表示到第层单元输入的加权和，包含偏置项。则上述等式可写为：









上述过程被称为“前向传播”，给定第层激活值，我们便可以按如下公式计算第层激活值：





一般用反向传播（简称BP）算法来训练神经网络，反向传播是利用链式法则递归计算表达式的梯度的方法。在整个计算线路图中，每个单元都会获得输入并立即计算：单元的输出值以及其关于输入值的局部梯度。一旦前向传播完毕，在反向传播的过程中，单元将最终获得整个网络的最终输出值在自己的输出值上的梯度。根据链式法则，

单元应该将回传的梯度乘以关于输入的局部梯度，从而得到整个网络的输出对该单元的每个输入值的梯度。

假设训练集有个训练样本，对于每一个样本，以损失函数为均方差函数为例，定义损失如下：



反向传播算法的流程如下：

1. 进行前向传播，计算神经网络中每层的激活值。
2. 对于输出层，计算残差如下：



1. 对于,计算残差如下：



1. 计算最终需要的偏导数：





1. 更新参数：





* + 1. **卷积神经网络**

传统的神经网络在层与层之间是全连接的，即每一层的神经元与下一层的所有神经元都有连接。这种连接方式虽然可以使神经网络有很强的表达能力，但会导致模型参数过多，训练困难。为了解决这个问题，Hubel等人[42]在1962年提出了感受野的概念。在视觉神经系统中，一个神经元的感受野是指视网膜上的特定区域，只有这个区域内的刺激才能够激活该神经元。以图像为例，图像中的每一个像素只与周围的像素有关联，而与离的较远的像素没有太强的关联，因此对于每一个神经元，只需要与图像中的一小块区域连接即可。每个神经元感受局部的图像，然后在更高层将局部的信息汇总得到全局的信息。

卷积神经网络[43-45]（CNN）根据以上机制提出。如图2-7为CNN的基本结构图，CNN的基本结构由输入层、卷积层、池化层、全连接层及输出层构成。卷积层和池化层一般会取若干个，早期的CNN采用卷积层和池化层交替设置，即一个卷积层连接一个池化层，池化层后再连接一个卷积层。卷积层中输出特征面的每个神经元与其输入进行局部连接，并通过对应的连接权重与局部输入进行加权求和再加上偏置值，得到该神经元输入值，该过程等同于卷积过程。



图2-7 卷积神经网络结构图

在一个卷积层中，上一层的特征图与可学习的卷积核卷积，并对输出使用激活函数，得到下一层的特征图。每个输出的特征图由多个输入的特征图组合得到，可以用公式2.26表示这一过程：



其中表示输入特征图的可选集合，为输入特征图编号，为输出特征图编号，表示第层网络。

池化层对输入的特征图进行下采样。池化层输出的特征图数量与输入的特征图数量保持一致，但输出的特征图会更小。池化层可以有效减少模型的训练参数数量，加速训练。池化层的计算方式如下：



其中表示一个下采样函数，常用的下采样函数有取最大值和取平均值。

* + 1. **残差网络**

在传统的神经网络中，网络深度越来越深后，会出现难以训练的情况，这通常是由梯度消失或梯度爆炸导致的[46-47]。梯度消失或爆炸的问题目前已经被一些正则化的方法[47-50]大幅的减轻了。当一个较深的神经网络解决了梯度的问题，开始收敛后，会产生退化问题，即网络深度增加后，分类精度达到饱和保持不变，甚至开始下降。He等人[51]提出一种深度残差学习框架来解决这个问题。

对于传统的神经网络，我们堆叠几个网络层来拟合函数。在残差网络中，我们让堆叠的非线性层来拟合另一个函数：。原始映射就被转换成。具体可以使用带短连接的前馈神经网络实现，如图2-8所示。



图2-8 残差网络构造单元

其中短连接指可以跳过神经网络中的一层或多层的连接，在残差网络中短连接使用恒等映射来表示，这种方式的优点是不会增加额外的参数和计算复杂度。 在残差网络实现过程中会遇到两种情况：1）当的维度和的维度相同时直接将和相加；2）当维度不同时我们使用1\*1的卷积核对进行卷积，卷积核的数量与的维度相同，这样就可以将进行卷积操作后的与相加。实验证明残差网络可以使得模型的精度随着网络深度的增加而增加，大幅减轻了退化问题带来的影响。

* + 1. **门卷积网络**

网络层数加深后除了有退化问题，还有一个更加常见的问题：梯度消失，即训练时距离输出较远的层梯度接近0。梯度消失的主要原因是神经网络在反向传播求梯度时，会有大量的连乘操作，如果连乘中的大部分元素都小于1，就会导致最后的梯度接近0，这种现象称为梯度消失。梯度消失会导致网络训练困难，甚至无法收敛。在卷积神经网络中，为了缓解梯度消失问题，除了激活函数使用ReLU以外，还可以模仿LSTM的门机制，在卷积层中加入门结构[52]。如图2-9所示为一种加入了门的卷积，假设输入为，左边的卷积为，右边的卷积为，则门卷积网络的输出为。我们用来表示,来表示，在输出上求关于的梯度可得



从等式中可以看到，加号右边一项由于对激活函数进行了求导，会导致梯度下降较快，而第一项则不存在这种情况，因此两项相加后保证了梯度不会快速下降。



图2-9 门卷积单元结构图

* 1. 集成学习

在一个学习任务中，单个学习器往往会由于陷入局部极小值而导致最终的泛化性能不佳，集成学习可以较好的解决这类问题。集成学习是一类通过组合多个单模型，来进一步提升模型性能的方法。如图2-10所示为集成学习的示意图：



图2-10 集成学习示意图

从图中可以看出，集成学习一般分为两步：1）训练一些不同的个体学习器。这些个体学习器可以是使用不同的学习算法，比如使用随机森林和神经网络各训练一批个体学习器，也可以都来自于同一种学习算法，比如所有个体学习器都使用神经网络来训练。当个体学习器的学习算法相同时，可以通过设定不同的算法参数来达到学习器之间的差异化。2）选择一种模型融合算法，对个体学习器进行融合。

对于数值类型的输出（包括概率值），一种最常见且有效的集成学习方法就是平均法（averaging）。式2.29和2.30分别是简单平均法和加权平均法的公式：





其中表示个体学习器的数量，表示每个学习器的权重，通常，。加权平均法给每个个体学习器一个权重，通过加权和的方式得到最终的输出，权重可以人为的根据个体学习器的表现分配，也可以通过模型计算得到。简单平均法可以看成是加权平均法在时的一种特例。目前常见的集成学习方法均可以看成是使用不同的方法来计算个体学习器的权重。

* 1. 本章小结

本章首先介绍了两种目前比较常用的树型模型，包括随机森林和梯度提升树，说明了两种模型大致的构造方法与不同点。随后介绍了神经网络，包括神经网络常用的激活函数，全连接网络的反向传播过程等。传统神经网络可能存在训练困难的问题，又介绍了卷积神经网络来解决。在卷积神经网络中给出了网络的结构图，并介绍了卷积和池化操作。最后为了解决梯度消失或者网络退化问题，在卷积神经网络的基础上介绍了残差网络和门卷积网络。

1. 实验数据准备及评价指标
   1. 实验数据准备

由于本文所有实验均使用同一份数据，因此在介绍实验方法前介绍实验数据的准备。本文实验所需的数据来自于上海某运营商在生产环境中的真实数据，运营商的数据主要包括BSS(Business Support System)数据和OSS(Operation Support System)数据两部分。BSS又叫做业务支持系统，主要保存各种业务相关的数据，例如用户的基本信息（性别、年龄、入网时长等）、用户的账户余额、语音相关数据（用户时长、通话次数等）、短信相关数据（短信条数、彩信条数等）等。OSS叫做运营支持系统，保存了电路交换（Circuit Switch, CS）数据、分组交换（Packet Switch, PS）数据和测量报告（Measurement Report, MR）数据。其中电路交换数据主要是一些和通话质量相关的数据，比如用户的掉话率和连接成功率等。分组交换数据主要是和网络相关的数据，包括了用户的上网速度、网络的连接成功率、网页浏览记录等。测量报告数据记录了用户的行动轨迹数据。

为了得到离网预测模型需要的输入，我们需要从原始数据中抽取需要的数据，并做一系列的转换，具体有以下几个步骤：

1、导出原始数据

运营商的原始数据保存在Oracle数据库中，首先需要从Oracle中导出可能会用到的表到运营商的大数据平台中。其中大数据平台是一个由13个节点组成的Hadoop/Spark平台。我们将数据导出到大数据平台的HDFS中，然后创建对应的Hive[53]表来保存。由于Spark SQL运行速度比Hive快得多，在后续步骤中使用Spark SQL[54]来处理表中的数据。Spark SQL是一种运行在Spark上的新的SQL引擎。

2、抽取特征

在抽取特征阶段我们以用户为单位，抽取每个用户的行为特征。将待抽取的特征分为静态特征和时序特征，从数据库中抽取这两类特征。静态特征指的是特征的值不随时间变化，或者随时间线性增长的值，例如性别、年龄、入网时长等。对于随时间线性增长的静态特征，在训练集中抽取截止到2015年12月时特征的值，在测试集中抽取截止到2016年1月时特征的值。时序特征指特征的值随时间变化而变化。对于时序特征，我们使用月和日两种不同的时间粒度来聚合特征的值，其中月粒度的特征选取2015年1月到2016年1月共13个月的数据，日粒度的特征选取2015年10月1日到2016年1月31日共123天的数据。具体的特征清单见表3-1到表3-2：

表3-1 静态特征清单

|  |  |
| --- | --- |
| **特征名称** | **描述** |
| is\_shanghai | 是否为本地人 |
| sex | 性别 |
| age | 年龄 |
| innet\_dura | 入网时长 |

表3-2 日粒度时序特征清单

|  |  |  |  |
| --- | --- | --- | --- |
| **特征名称** | **描述** | **特征名称** | **描述** |
| localbase\_outer\_call\_dur | 本地网外时长 | all\_call\_cnt | 总话单条数 |
| ld\_call\_dur | 长途主叫时长 | local\_base\_call\_cnt | 本地语音通话次数 |
| roam\_call\_dur | 漫游主叫时长 | ld\_call\_cnt | 长途语音通话次数 |
| localbase\_called\_dur | 本地被叫时长 | caller\_cnt | 主叫次数 |
| roam\_called\_dur | 漫游被叫时长 | voice\_dur | 语音通话计费时长 |
| cm\_dur | 和移动通话时长 | caller\_dur | 主叫计费时长 |
| ct\_dur | 和电信通话时长 | localbase\_inner\_call\_dur | 本地网内时长 |
| busy\_call\_dur | 忙时通话时长 | call\_10010\_cnt | 客户拨打10010次数 |
| fest\_call\_dur | 节假日通话时长 | sms\_bill\_cnt | 短信计费量 |
| sms\_p2p\_mo\_cnt | 上行点对点短信条数 | sms\_p2p\_mt\_cnt | 下行点对点短信条数 |
| sms\_p2p\_inner\_mo\_cnt | 上行网内点对点短信条数 | mms\_cnt | 点对点彩信次数 |
| sms\_p2p\_cm\_mo\_cnt | 上行与移动点对点短信条数 | mms\_p2p\_mt\_cnt | 下行点对点彩信条数 |
| sms\_p2p\_ct\_mo\_cnt | 上行与电信点对点短信条数 | gprs\_all\_flux | gprs\_总流量 |
| sms\_p2p\_roam\_int\_mo\_cnt | 上行国际漫出点对点短信条数 | balance | 余额 |
| mms\_p2p\_mo\_cnt | 上行点对点彩信条数 | total\_payment\_amount | 充值 |
| mms\_p2p\_inner\_mo\_cnt | 上行网内点对点彩信条数 | distinct\_serve\_count | 服务类短信号码个数 |
| mms\_p2p\_other\_mo\_cnt | 上行其他点对点彩信条数 | serve\_sms\_count | 服务类短信总条数 |

表3-3 月粒度时序特征清单

|  |  |  |  |
| --- | --- | --- | --- |
| **特征名称** | **描述** | **特征名称** | **描述** |
| terminal | 终端 | call\_10010\_cnt | 客户拨打10010次数 |
| localbase\_outer\_call\_dur | 本地网外时长 | sms\_bill\_cnt | 短信计费量 |
| ld\_call\_dur | 长途主叫时长 | sms\_p2p\_mt\_cnt | 下行点对点短信条数 |
| roam\_call\_dur | 漫游主叫时长 | mms\_cnt | 点对点彩信次数 |
| localbase\_called\_dur | 本地被叫时长 | mms\_p2p\_mt\_cnt | 下行点对点彩信条数 |
| roam\_called\_dur | 漫游被叫时长 | gprs\_all\_flux | gprs\_总流量 |
| cm\_dur | 和移动通话时长 | total\_charge | 本月总收入 |
| ct\_dur | 和电信通话时长 | gprs\_flux | gprs流量 |
| busy\_call\_dur | 忙时通话时长 | gprs\_charge | gprs费用 |
| fest\_call\_dur | 节假日通话时长 | local\_call\_minutes | 本地通话分钟数 |
| sms\_p2p\_mo\_cnt | 上行点对点短信条数 | toll\_call\_minutes | 长途通话分钟数 |
| sms\_p2p\_inner\_mo\_cnt | 上行网内点对点短信条数 | roam\_call\_minutes | 漫游通话分钟数 |
| sms\_p2p\_cm\_mo\_cnt | 上行与移动点对点短信条数 | voice\_call\_minutes | 语音通话分钟数 |
| sms\_p2p\_ct\_mo\_cnt | 上行与电信点对点短信条数 | p2p\_sms\_mo\_cnt | 短信\_点对点短信发送条数 |
| sms\_p2p\_roam\_int\_mo\_cnt | 上行国际漫出点对点短信条数 | p2p\_sms\_mo\_charge | 短信\_点对点短信费 |
| mms\_p2p\_mo\_cnt | 上行点对点彩信条数 | gift\_voice\_call\_dur | 套餐内语音时长 |
| mms\_p2p\_inner\_mo\_cnt | 上行网内点对点彩信条数 | gift\_sms\_mo\_cnt | 套餐内短信条数 |
| mms\_p2p\_other\_mo\_cnt | 上行其他点对点彩信条数 | gift\_flex\_value | 套餐内流量 |
| all\_call\_cnt | 总话单条数 | balance | 余额 |
| local\_base\_call\_cnt | 本地语音通话次数 | total\_payment\_amount | 充值 |
| ld\_call\_cnt | 长途语音通话次数 | pagerank | 用户重要性 |
| caller\_cnt | 主叫次数 | lp | 标签传播算法 |
| voice\_dur | 语音通话计费时长 | distinct\_serve\_count | 服务类短信号码个数 |
| caller\_dur | 主叫计费时长 | serve\_sms\_count | 服务类短信总条数 |
| localbase\_inner\_call\_dur | 本地网内时长 | balance\_diff | 余额与账单差值 |
| sms\_entropy | 熵值 |  |  |

3、抽取标签

在离网预测任务中，标签有离网和在网两类。由于数据库中没有明确的字段用来表示离网状态，需要我们自行定义标签。根据电信领域专家们的建议，如果一个用户进入充值期且一定天数没有充值，那么这个用户就被标记为离网用户。其中进入充值期指的是用户手机号码处于欠费状态。图3-1是2014年一整年用户进入充值期后多久进行充值的人数比例分布图。从图中可以看出，进入充值期后15天内充值的用户占到了93%，因此我们将15天没有充值的用户定义为离网用户。共抽取2016年1月和2016年2月两个月的离网和在网标签。

图3-1 用户进入充值期后充值时间分布图

4、生成数据集

特征和标签抽取完成后开始生成数据集。对于抽取的静态特征，直接将特征的值分配给训练集和测试集。对于抽取的时序特征和标签，我们使用滑动窗口的方式分配给训练集和测试集。不同时间粒度数据的滑动窗口如图3-2和图3-3：



图3-2 月粒度数据的滑动窗口 图3-3 日粒度数据的滑动窗口

对于日粒度数据，每个方格表示当月所有天的数据。以2015年10月为例，如图3-4所示：



图3-4 2015年10月的日粒度数据

最终形成的训练集和测试集由静态特征，月特征、日特征和标签组成，如图3-5所示：



图3-5 数据集格式

最后我们将训练集打乱，抽取20%数据作为验证集，将所有数据保存为hdf5文件。统计各个数据集的样本数以及正负例比例如表3-4：

表3-4 数据集样本数统计

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **数据集** | **样本数** | **正例数** | **负例数** | **正例数占比** |
| 训练集 | 1132343 | 43645 | 1088698 | 3.85% |
| 验证集 | 283086 | 11120 | 271966 | 3.93% |
| 测试集 | 1380154 | 50376 | 1329778 | 3.65% |

统计各个数据集中特征的基本情况如表3-5：

表3-5 数据集各类特征统计

|  |  |  |
| --- | --- | --- |
| **特征类型** | **时间序列长度** | **特征数量** |
| 月粒度时序特征 | 12月 | 51 |
| 日粒度时序特征 | 92天 | 34 |
| 静态特征 | - | 4 |

* 1. 评价指标

本文的所有实验使用查准率（Precision）、查全率（Recall）、AUC和PR-AUC作为评价指标。

* + 1. **查准率和查全率**

首先给出模型预测结果的混淆矩阵：

表3-6 混淆矩阵

|  |  |  |
| --- | --- | --- |
| 真实情况 | 预测结果 | |
| 正例 | 负例 |
| 正例 | TP（真正例） | FN（假负例） |
| 负例 | FP（假正例） | TN（真负例） |

由混淆矩阵我们可以得到查准率P和查全率R的计算公式：





在本文实验中，我们先对所有预测样本按预测概率值降序排序，然后将概率值Top N高的样本预测为正例，然后得到和。参考公式3.1和3.2，我们给出和的计算公式：





* + 1. AUC

AUC的值是ROC曲线下的面积。ROC曲线的纵坐标为真正例率（TPR），横坐标为假正例率（FPR），分别定义为：





计算AUC值的具体实现方法为：将模型预测得到的概率值降序排序，所有样本都标记为负例，首先在坐标轴中的(0, 0)标记一个点。依次将当前概率值最大的样本标记为正例，并且重新计算TPR和FPR，在坐标轴中标记计算出的点。连接相邻的标记的点得到ROC曲线。由于得到的ROC曲线是由一系列矩形拼接而成，可以很容易的计算出ROC曲线的面积。

* + 1. PR-AUC

PR-AUC是PR曲线下的面积。PR曲线的纵坐标为查准率（P），横坐标为查全率（R），计算方法与AUC类似。由于我们的数据集正负样本极不平衡，PR-AUC比AUC更适合评价模型的性能[55]。

* 1. 本章小结

本章首先给出了本文所有实验所需要的数据的生成过程，然后介绍了本文所有实验用到的评价指标。为了获得本文实验所需要的数据，我们首先从运营商的Oracle数据库中提取出我们需要的特征到Hive中，然后使用月和日两种时间粒度对时序特征进行聚合，最后使用滑动窗口的方式得到数据集。在评价指标部分，本章介绍了查准率、查全率、AUC和PR-AUC四个评价指标的计算方法。

1. 基于神经网络的离网预测模型

在过去的几年中，随机森林和梯度提升树等树型模型由于较好的性能和较快的训练速度，在各个运营商的离网预测任务中均得到了广泛的应用。随着硬件性能的提升和运营商数据量的增加，神经网络这类计算量大且需要大量数据的模型逐渐进入了人们的视野。本章针对现有的基于神经网络的离网预测模型中存在的两个问题：1）模型易过度泛化；2）网络层数较浅，提出了一种基于宽度&深度学习[35]框架的改进的离网预测模型来解决问题。本章提出的神经网络模型相比于现有模型有以下两个优势：1）结合了宽度模型与深度模型的优势，神经网络作为深度模型，线性模型作为宽度模型，两部分进行加权和得到最终输出；2）模型层数更多，网络更深，能更好的拟合数据。

* 1. 网络结构框架

图4-1为本章提出的基于神经网络的离网预测模型的网络结构框架图。



图4-1 网络结构框架图

从框架图中可以看到整个网络分为宽度模型和深度模型两个部分，将这两个部分的输出进行拼接后，输入到整个网络的最后一个全连接层。注意到在深度模型部分我们使用卷积网络来处理时序特征，为了解决网络层数较浅的问题，我们在卷积网络部分尝试了两种深层的卷积网络。我们在4.1.1节解释过度泛化问题以及为什么要加入宽度模型，在4.1.2节介绍宽度模型的结构，在4.1.3节介绍深度模型的结构，在4.1.4节介绍两部分模型如何进行拼接。

* + 1. **过度泛化**

Cheng等人[35]在介绍宽度&深度学习时提到了神经网络应用于推荐系统时的过度泛化问题，在离网预测任务中同样存在这个问题。当神经网络的输入包含离散型特征时，一种常见的做法是模仿自然语言处理领域训练语言模型的方法，将离散稀疏的特征值转换为低维稠密的向量。具体地，我们将特征看作是一个字典，特征中具体的取值则是字典中的单词，参考语言模型中词向量的概念，我们将离散型特征中的每一种取值转换为一个特征向量。将特征向量作为神经网络参数的一部分，更新参数的同时训练特征向量。

当特征向量训练完成后，可以使用向量间的欧氏距离来判断不同向量之间的相似度。越相似的特征值，在输入到模型后越容易得到相似的预测结果。这样当模型在测试集中遇到没有遇到过的特征组合时，就可以很好的进行泛化。这种训练特征向量的操作在特征向量被充分训练后是可行的，但是在真实数据中特征的取值通常只集中在很少的几种值上。这样就会导致那些较少出现的取值，对应的特征向量不能被充分的训练，导致特征向量的值失去意义，从而进行一些错误的泛化，降低模型性能，即过度泛化。

加入了交叉积转换的线性模型可以有效的记忆特征间的共现关系，特征组合共现的频率越高，则线性模型对应的权重就越大。对于一些很少或没有出现过的特征组合，线性模型对应的权重也会接近0，即直接忽略这种特征组合，不会进行泛化，从而避免过度泛化的问题。因此，为了解决第一个问题，即神经网络容易过度泛化的问题，可以将线性模型和神经网络结合，充分发挥线性模型记忆特征的共现关系的优势和神经网络泛化能力强的优势。

* + 1. **宽度模型**

宽度模型部分是一个线性模型，具有类似的形式。宽度模型在线性模型中加入交叉积转换，来捕捉特征之间的共现性。线性模型被工业界广泛用于带有稀疏特征的大规模分类/回归任务上。线性模型的一个常用的应用场景是使用被one-hot编码处理过的二值化稀疏特征来训练模型。比如有一个二值化特征叫“sex=male”，当特征的值为1时表示性别为男。稀疏特征之间的交叉积还可以使我们更直观的发现两个特征是如何共同影响标签的值的。例如有一个特征叫“sex=male&city=suzhou”，特征值为1时表示性别为男且城市为苏州。当这个特征对应的权重的值较大时，就可以认为sex特征和city特征的组合对标签的值影响较大。

在我们的模型中，宽度模型的输入由稀疏特征和稀疏特征间的交叉积两部分拼接构成。为了得到所需要的输入，我们从第三章抽取的特征中选取终端、性别、是否本地人3个离散型特征，同时将年龄特征分段离散化，作为第4个离散型特征，对于第一部分的输入，对抽取的这些特征进行one-hot编码。对于第二部分输入，从所有的one-hot编码中选择部分编码做交叉积转换。选取的特征见图4-2到图4-5：

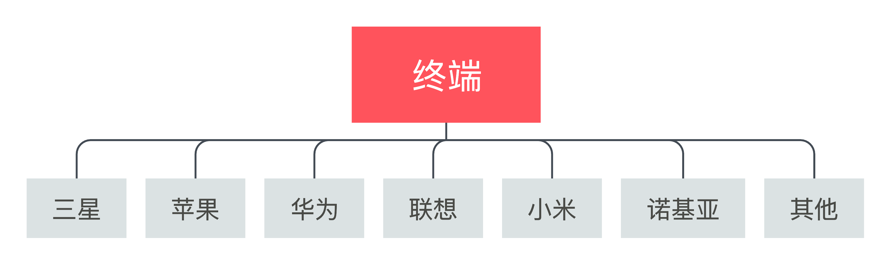


图4-2 终端

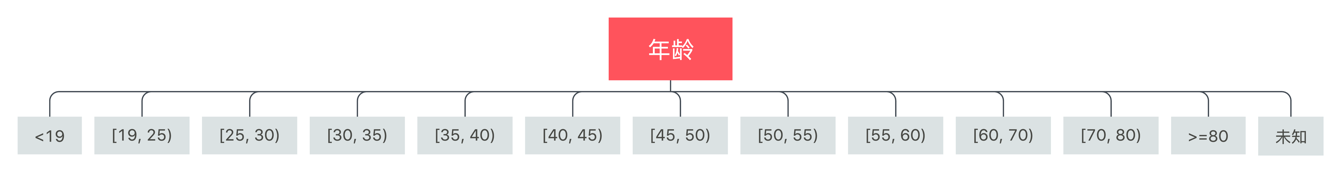


图4-3 年龄

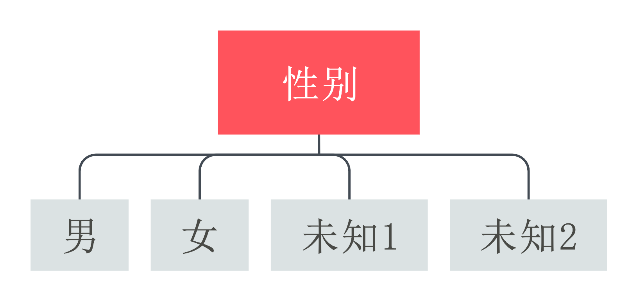
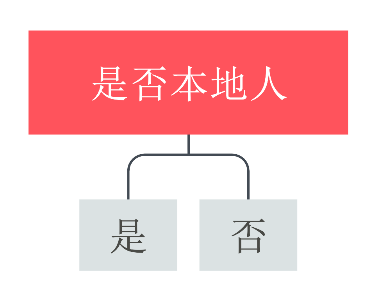
 

图4-4 性别 图4-5 是否本地人

4个特征具体的特征处理方法如下：

终端特征。由于终端特征为月粒度的时序特征，我们取离标签所在月最近的那个月的终端的值用于后续的计算。例如标签为2016年1月的，则取2015年12月的终端特征的值。由图4-2可知，终端的值有6种具体的终端型号再加上“其他”共7个取值构成，在计算交叉积时，我们去除“其他”这个值，保留其余的值。

年龄特征。原始的年龄特征为离散型和连续型特征的结合，具体表现为年龄的值为1时表示小于等于18岁，值为2时表示大于等于80岁，值为3时表示缺失值，其他值表示实际年龄。我们将原始特征的值分段表示，转换为离散型特征，如图4-3，共有13种不同的取值。计算交叉特征时不考虑缺失值。

性别特征。性别特征为离散型特征，在数据库中有4种取值，除了男、女两种取值，我们将没有输入性别的用户标记为“未知1”，将其他异常情况标记为“未知2”。同样的，在计算交叉特征时不考虑“未知1”和“未知2”这两种性别不明的取值。

是否本地人特征。是否本地人特征同样是离散型特征，只有是和否两种取值，在计算交叉特征时计算所有两种取值。

最后我们使用终端特征的值的one-hot编码分别与年龄、性别、是否本地人这三个特征的值的one-hot编码相乘，得到交叉特征部分的输入。宽度模型部分完整的输入清单见表4-1：

表4-1 宽度模型输入统计

|  |  |
| --- | --- |
| **特征名** | **特征数量** |
| 终端 | 7 |
| 年龄 | 13 |
| 性别 | 4 |
| 是否本地人 | 2 |
| 终端\*年龄 | 72(6\*12) |
| 终端\*性别 | 12(6\*2) |
| 终端\*是否本地人 | 12(6\*2) |
| 总计 | 122 |

* + 1. **深度模型**

深度模型部分是一个神经网络，网络的大致结构见图4-1的左半部分。本节将从输入特征预处理和神经网络结构两方面来介绍深度模型部分。

#### 输入特征预处理

我们将输入的特征分为两种类型，分别为时序特征和静态特征，每种类型的特征又有连续型和离散型两种。对于神经网络这类模型，离散型的特征不适合直接作为输入。与线性模型中对离散型特征进行one-hot编码不同，在神经网络中对特征进行one-hot编码可能会导致特征维度过大，难以训练。一种比较常用的方法是模仿自然语言处理中对单词进行embedding的做法，将每一个离散型特征的值转换为稠密的embedding向量后，与连续型特征拼接作为输入。其中embedding向量的值作为网络的参数，在网络更新参数的同时更新向量的值。表4-2为被转换的离散型特征和对应的embedding向量。

表4-2 特征embedding

|  |  |  |
| --- | --- | --- |
| **特征名** | **特征取值数量** | **embedding维度** |
| 终端 | 7 | 5 |
| 性别 | 4 | 5 |
| 是否本地人 | 2 | 5 |

#### 神经网络结构

深度模型部分的网络结构主要由两部分构成，对于时序特征，使用一维的卷积来处理，对时间序列的时间维度进行卷积，将每个时间点的特征数作为通道数。对于静态特征，将静态特征与卷积的输出拼接，共同输入到第一个全连接层中。从图4-1中可以看到，网络的全连接层部分共有3个全连接层，3层的输出维度分别为256、64和1，其中前两层的激活函数为ReLU，最后一层为Sigmoid。为了解决第二个问题，即网络层数较浅的问题，在经过大量实验后，我们在卷积层部分尝试了两种卷积网络的结构来实现深层的网络，分别为残差网络和门卷积网络。其中残差网络可以达到更好的性能，而门卷积网络的性能略差于残差网络，但训练时间更短。

（1）残差网络

表4-3 残差网络卷积层参数设置

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **名称** | **卷积核长度** | **卷积核数量** | **滑动间隔** | **填充** |
| conv1 | 1 | 64 | 1 | valid |
| res2a | 1  3  1 | 64  64  256 | 1  1  1 | valid  same  valid |
| res2b | 1  3  1 | 64  64  256 | 1  1  1 | same  same  valid |
| res3a | 1  3  1 | 128  128  512 | 2  1  1 | valid  same  valid |
| res3b | 1  3  1 | 128  128  512 | 1  1  1 | same  same  valid |
| res3c | 1  3  1 | 128  128  512 | 1  1  1 | same  same  valid |

当网络层数加深后，可能会出现退化问题，即网络加深后，精度趋于饱和不再提升，甚至会开始快速下降。为了缓解这个问题，我们借鉴深度残差网络的方法，在网络中加入短连接。我们把被一个短连接跳过的卷积层的集合称为一个残差块，残差块的具体结构可以参考2.2.3节。表4-3为加入了短连接的卷积网络的参数设置，其中conv1为一个正常的卷积层，res2a-res3c为残差块。

对于每一个卷积层，我们在卷积操作之后先使用Batch Normalization，然后使用ReLU函数作为激活函数。

（2）加入了门的卷积神经网络

网络层数加深后除了有退化问题，还会存在更常见的梯度消失的问题。梯度消失会导致网络训练困难，甚至难以收敛。为了缓解梯度消失问题，在卷积神经网络中，我们可以通过模仿LSTM中的门机制，在卷积层中加入门得到门卷积单元，门卷积单元的具体结构可以参考2.2.4节。表4-4为加入了门的卷积网络的参数设置，其中conv1为一个正常的卷积层，gat2a-gat4c表示门卷积单元。由于一个门卷积单元中的两个卷积层是完全相同的参数，表中只列出其中一个卷积层的参数。

表4-4 门卷积网络卷积层参数设置

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **名称** | **卷积核长度** | **卷积核数量** | **滑动间隔** | **填充** |
| conv1 | 1 | 64 | 1 | valid |
| gat2a | 5 | 64 | 1 | same |
| gat2b | 5 | 64 | 1 | same |
| gat3a | 3 | 128 | 1 | same |
| gat3b | 3 | 128 | 1 | same |
| gat3c | 3 | 128 | 1 | same |
| gat4a | 3 | 256 | 1 | same |
| gat4b | 3 | 256 | 1 | same |
| gat4c | 3 | 256 | 1 | same |

对于每个门卷积单元的输出，我们先使用Batch Normalization，然后使用ReLU函数作为激活函数。

* + 1. **宽度&深度模型拼接**

当得到宽度模型和深度模型的输出后，我们可以用公式计算得到模型的最终预测结果：



其中表示Sigmoid函数，是宽度模型中交叉积转换的输入，是交叉积转换的输出，是深度模型的输出，和分别是宽度模型和深度模型的权重。可以看到模型的最终输出是宽度模型输出和深度模型输出的加权和。我们将宽度模型和深度模型的权重拼接，作为网络的最后一层全连接层，在网络反向传播时作为参数训练。

* 1. 实验设置
     1. **数据准备及评价指标**

我们使用3.1节中准备好的数据进行训练。3.1节中将数据分为静态特征，月粒度特征和日粒度特征三部分，为了方便与其他论文中的方法作对比，我们将数据集拆分为静态特征和月粒度时序特征（）、静态特征和日粒度时序特征两组数据（）。对于本章提出的网络和用于作对比实验的每个网络，分别使用这两组数据训练模型和评估模型。

实验的评价指标使用3.2节中提到的指标，分别为查准率、查全率、AUC和PR-AUC。

* + 1. **实验环境**

本文全部实验均在单机上进行，其中树型模型和神经网络模型在两台不同的计算机上训练。树型模型使用的计算机有4个Intel X5690、140GB内存，神经网络使用的计算机有2个Intel E5-2620、128GB内存、3个GeForce GTX 1080Ti。在训练每个神经网络时，我们只使用一块GPU。

* + 1. **模型训练参数设置**

经过大量的实验调试，我们将模型的训练参数设置为以下的值：模型的损失函数为交叉熵损失函数，即，其中是样本数，表示预测函数。优化算法采用Adam，学习率设置为0.001，设置为0.9，设置为0.999,设置为1e-7。Batch size的大小为128。在使用训练集训练的过程中，每训练1000个batch，我们计算一次模型在验证集上的损失，如果本次验证损失小于上次计算的验证损失，则保存当前模型。同时为了防止模型过拟合，加速训练过程，还使用了Earlystopping，当验证集损失连续100次没有下降后，终止训练过程。

* + 1. **对比实验**

表4-5列出了所有baseline模型和本章提出的模型，我们共设置7个模型与本章提出的模型进行对比，分别为1）的深度模型部分（）；2）的深度模型部分（）；3）宽度模型部分（）；4）Umayaparvathi等人[6]使用的浅层神经网络模型；5）Umayaparvathi等人[6]使用的浅层神经网络模型；6）随机森林（）；7）梯度提升树（），其中、、和使用与4.2.2节相同的训练参数。为了公平的比较浅层神经网络和本章提出的模型，我们为和这两个浅层神经网络同样加入宽度模型部分。

表4-5 实验模型总结

|  |  |
| --- | --- |
| **模型** | **含义** |
|  | 仅包含深度模型部分且使用残差网络 |
|  | 仅包含深度模型部分且使用门卷积网络 |
|  | 仅包含宽度模型部分 |
|  | 深度模型部分使用浅层全连接网络 |
|  | 深度模型部分使用浅层卷积网络 |
|  | 随机森林 |
|  | 梯度提升树 |
|  | 深度模型部分使用残差网络 |
|  | 深度模型部分使用门卷积网络 |

以下为对比模型的具体参数设置：

（1）、、

是的深度模型部分，如图4-1，我们将宽度模型去掉，仅保留深度模型部分。类似的是的深度模型部分。则为将深度模型去掉，仅保留宽度模型部分。

（2）

是一个全连接的神经网络，各全连接层的参数设置见表4-6，其中优化算法和batch size的设置与4.2.2节相同。

表4-6 参数设置

|  |  |  |  |
| --- | --- | --- | --- |
| **网络层** | **输出维度** | **初始化方式** | **激活函数** |
| Dense\_1 | 74 | Normal | ReLU |
| Dense\_2 | 37 | Normal | ReLU |
| Dense\_3 | 1 | Normal | Sigmoid |

（3）

是一个卷积神经网络，网络由以下几层构成：

Layer 1:一维卷积，卷积核大小为3，卷积核数量为250，滑动间隔为1。

Layer 2：全局最大池化。

Layer 3：全连接层，输出维度为250。

Layer 4：Dropout，比例为0.5。

Layer 5：全连接层，输出维度为1，即模型的最终输出。

（4）

使用Scikit-learn中的RandomForestClassifier类来训练模型。其中n\_estimators参数设为200，即随机森林中包含200棵树，n\_jobs参数设置为-1，表示使用cpu所有核心来训练，其余参数使用默认值。

（5）

使用XGBoost来训练，模型的参数设置表4-7，未设置的参数均使用默认值。

表4-7 参数设置

|  |  |
| --- | --- |
| **参数** | **值** |
| booster | gbtree |
| silent | 0 |
| eta | 0.1 |
| gamma | 0.1 |
| max\_depth | 12 |
| min\_child\_weight | 3 |
| subsample | 0.8 |
| colsample\_bytree | 0.8 |
| lambda | 100 |
| alpha | 0 |
| max\_delta\_step | 1 |
| objective | binary:logistic |
| eval\_metric | logloss |
| seed | 519 |
| num\_boost\_round | 10000 |
| early\_stopping\_rounds | 50 |

* 1. 实验结果与分析

表4-8到表4-13是各个模型在不同数据集上的训练结果。其中模型由于输入只有静态特征，不包含时序特征，训练结果单独列出。

表4-8 模型在上的Precision

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **top N**  **model** | **10000** | **15000** | **20000** | **25000** | **50000** | **100000** | **150000** | **200000** |
|  | 0.6720 | 0.6051 | 0.5543 | 0.5147 | 0.3849 | 0.2692 | 0.2125 | 0.1767 |
|  | 0.5376 | 0.5127 | 0.4868 | 0.4628 | 0.3812 | 0.2756 | 0.2168 | 0.1801 |
|  | 0.6428 | 0.5846 | 0.5410 | 0.5024 | 0.3792 | 0.2696 | 0.2129 | 0.1778 |
|  | 0.6100 | 0.5483 | 0.5020 | 0.4645 | 0.3469 | 0.2418 | 0.1913 | 0.1611 |
|  | 0.6528 | 0.6013 | 0.5571 | 0.5245 | 0.3995 | 0.2830 | 0.2228 | 0.1849 |
|  | 0.6395 | 0.5877 | 0.5451 | 0.5104 | 0.3919 | 0.2816 | 0.2224 | 0.1847 |
|  | 0.6787 | 0.6198 | 0.5730 | 0.5350 | 0.4018 | 0.2831 | 0.2231 | 0.1854 |
|  | 0.6501 | 0.5937 | 0.5550 | 0.5179 | 0.3916 | 0.2744 | 0.2175 | 0.1825 |

表4-9 模型在上的Precision

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **top N**  **model** | **10000** | **15000** | **20000** | **25000** | **50000** | **100000** | **150000** | **200000** |
|  | 0.6447 | 0.5918 | 0.5444 | 0.5072 | 0.3782 | 0.2588 | 0.2018 | 0.1671 |
|  | 0.5360 | 0.5235 | 0.5021 | 0.4801 | 0.3841 | 0.2760 | 0.2179 | 0.1806 |
|  | 0.4534 | 0.4105 | 0.3773 | 0.3512 | 0.2669 | 0.1952 | 0.1606 | 0.1383 |
|  | 0.4709 | 0.4293 | 0.4028 | 0.3786 | 0.3034 | 0.2264 | 0.1835 | 0.1564 |
|  | 0.6153 | 0.5814 | 0.5467 | 0.5167 | 0.4023 | 0.2819 | 0.2200 | 0.1822 |
|  | 0.5769 | 0.5491 | 0.5157 | 0.4853 | 0.3844 | 0.2751 | 0.2167 | 0.1806 |
|  | 0.6518 | 0.6045 | 0.5656 | 0.5295 | 0.4017 | 0.2833 | 0.2221 | 0.1839 |
|  | 0.6172 | 0.5777 | 0.5423 | 0.5125 | 0.3959 | 0.2794 | 0.2194 | 0.1817 |

表4-10 模型在上的Recall

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **top N**  **model** | **10000** | **15000** | **20000** | **25000** | **50000** | **100000** | **150000** | **200000** |
|  | 0.1334 | 0.1802 | 0.2200 | 0.2554 | 0.3821 | 0.5344 | 0.6327 | 0.7015 |
|  | 0.1067 | 0.1527 | 0.1932 | 0.2297 | 0.3783 | 0.5471 | 0.6455 | 0.7151 |
|  | 0.1276 | 0.1741 | 0.2148 | 0.2493 | 0.3763 | 0.5352 | 0.6339 | 0.7060 |
|  | 0.1211 | 0.1633 | 0.1993 | 0.2305 | 0.3443 | 0.4800 | 0.5695 | 0.6396 |
|  | 0.1296 | 0.1790 | 0.2212 | 0.2603 | 0.3966 | 0.5617 | 0.6633 | 0.7340 |
|  | 0.1269 | 0.1750 | 0.2164 | 0.2533 | 0.3890 | 0.5589 | 0.6622 | 0.7334 |
|  | 0.1347 | 0.1846 | 0.2275 | 0.2655 | 0.3988 | 0.5620 | 0.6642 | 0.7361 |
|  | 0.1290 | 0.1768 | 0.2203 | 0.2570 | 0.3887 | 0.5446 | 0.6477 | 0.7244 |

表4-11 模型在上的Recall

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **top N**  **model** | **10000** | **15000** | **20000** | **25000** | **50000** | **100000** | **150000** | **200000** |
|  | 0.1280 | 0.1762 | 0.2162 | 0.2517 | 0.3753 | 0.5137 | 0.6009 | 0.6634 |
|  | 0.1064 | 0.1559 | 0.1993 | 0.2382 | 0.3813 | 0.5479 | 0.6487 | 0.7168 |
|  | 0.0900 | 0.1222 | 0.1498 | 0.1743 | 0.2649 | 0.3875 | 0.4781 | 0.5491 |
|  | 0.0935 | 0.1278 | 0.1599 | 0.1879 | 0.3011 | 0.4495 | 0.5464 | 0.6210 |
|  | 0.1221 | 0.1731 | 0.2170 | 0.2564 | 0.3993 | 0.5597 | 0.6551 | 0.7235 |
|  | 0.1145 | 0.1635 | 0.2047 | 0.2408 | 0.3815 | 0.5462 | 0.6451 | 0.7169 |
|  | 0.1294 | 0.1800 | 0.2246 | 0.2628 | 0.3987 | 0.5624 | 0.6612 | 0.7300 |
|  | 0.1225 | 0.1720 | 0.2153 | 0.2543 | 0.3930 | 0.5546 | 0.6534 | 0.7214 |

表4-12 模型在上的AUC和PR-AUC 表4-13 模型在上的AUC和PR-AUC

|  |  |  |
| --- | --- | --- |
| **model** | **AUC** | **PR-AUC** |
|  | 0.8815 | 0.3493 |
|  | 0.8879 | 0.3134 |
|  | 0.8896 | 0.3385 |
|  | 0.8667 | 0.2955 |
|  | 0.9016 | 0.3624 |
|  | 0.9000 | 0.3469 |
|  | 0.9027 | 0.3719 |
|  | 0.8997 | 0.3556 |

|  |  |  |
| --- | --- | --- |
| **model** | **AUC** | **PR-AUC** |
|  | 0.8583 | 0.3294 |
|  | 0.8899 | 0.3140 |
|  | 0.8209 | 0.2067 |
|  | 0.8530 | 0.2424 |
|  | 0.8929 | 0.3465 |
|  | 0.8899 | 0.3226 |
|  | 0.8959 | 0.3562 |
|  | 0.8912 | 0.3400 |

表4-14 模型

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **top N** | **Precision** | **Recall** | **AUC** | **PR-AUC** |
| 10000 | 0.0776 | 0.0154 | 0.6397 | 0.0552 |
| 15000 | 0.0733 | 0.0218 |
| 20000 | 0.0712 | 0.0283 |
| 25000 | 0.0704 | 0.0349 |
| 50000 | 0.0655 | 0.0650 |
| 100000 | 0.0622 | 0.1235 |
| 150000 | 0.0607 | 0.1808 |
| 200000 | 0.0592 | 0.2350 |

* + 1. **模型训练时间分析**

表4-15比较了训练各个模型所需要的时间，对于神经网络模型，训练时间指的是从模型开始训练到发生Earlystopping后终止训练所花费的时间。

可以看到在训练时间上，将模型的网络层数加深后，训练时间明显变长。同样的在神经网络中加入宽度模型后，也会导致训练时间变长。对于和这两个宽度&深度学习框架的模型，在训练时间上会明显好于，而观察表4-10到4-13会发现，在性能上优于，这两个模型在时间和性能上各有优势。虽然我们提出的模型在训练时间上会明显长于现有的模型，但由于在电信运行商实际使用时，离网预测模型仅需每月训练一次，因此训练时间的延长也是可以接受的。

表4-15 模型训练时间比较

|  |  |  |
| --- | --- | --- |
| **模型** | **训练时间（分钟）** | |
|  |  |
|  | 20 | |
|  | 31 | 72 |
|  | 110 | 200 |
|  | 81 | 69 |
|  | 71 | 114 |
|  | 176 | 183 |
|  | 121 | 138 |
|  | 249 | 231 |
|  | 177 | 145 |

* + 1. **与深度/宽度模型比较**

为了解决神经网络过度泛化问题，我们比较了宽度&深度模型是否真的能在仅深度模型的基础上进一步提高性能。我们根据表4-8到表4-13画出、、、和 5个模型在不同数据集上的对比图。其中模型的输入为静态特征，并不包含时序特征，为了方便与其他模型对比，一同画入图中。对比图见图4-6到图4-9：

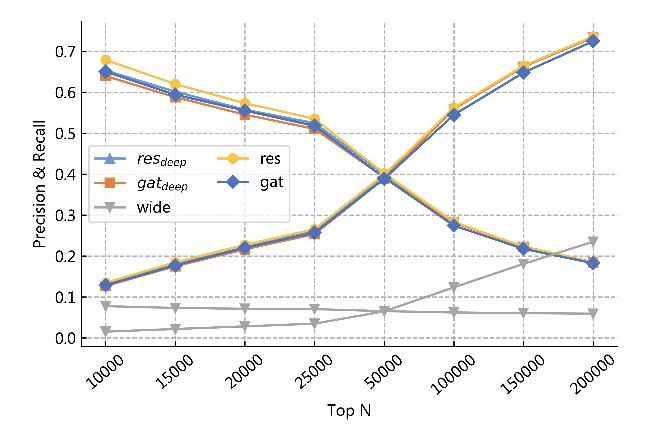
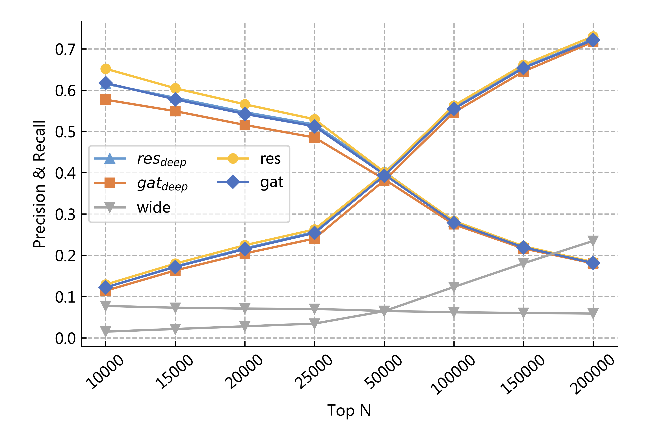
 

图4-6 数据集上的查准率和查全率 图4-7 数据集上的查准率和查全率

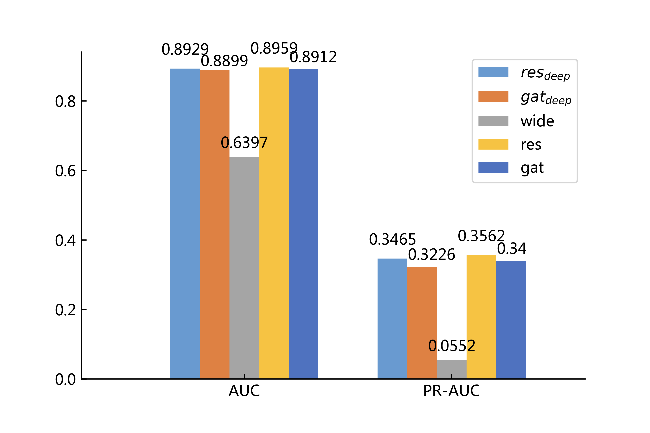
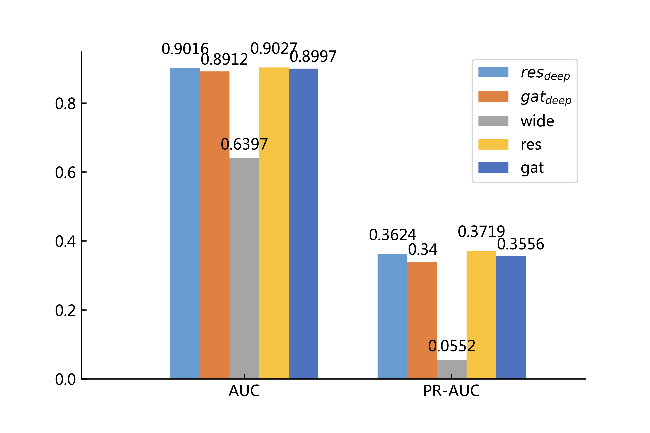


图4-8 数据集上的AUC和PR-AUC 图4-9 数据集上的AUC和PR-AUC

可以看到模型在查准率、查全率、AUC和PR-AUC的表现均大幅落后于其他模型，而和的查准率和查全率比和的略差。当Top N的N取值为10000时，我们比较各个模型的查准率和查全率，在数据集上的查准率和查全率比高3.97%，比高1.66%，在数据集上比高5.93%，比高6.99%。深度模型的AUC与完整模型相比几乎相同，PR-AUC则低于完整模型。在数据集上的PR-AUC比高2.62%，的PR-AUC比高2.51%，在数据集上的PR-AUC比高2.8%，的PR-AUC比高5.39%。

宽度模型本身没有很好的性能，这是由于本章使用的宽度模型的输入维度有限，没有做复杂的特征工程。但与深度模型结合后，可以在深度模型已有较好性能的基础上进一步提高性能。因此可以认为基于宽度&深度学习的神经网络模型在一定程度上解决了神经网络的过度泛化的问题。

* + 1. **与浅层神经网络模型比较**

为了解决离网预测任务中现有的神经网络层数过浅的问题，我们比较了在网络的深度模型部分使用深层的网络相比于使用浅层的网络是否能提升性能。我们根据表4-8到4-13画出、、和 4个模型在不同数据集上的对比图，对比图见图4-10到图4-13：

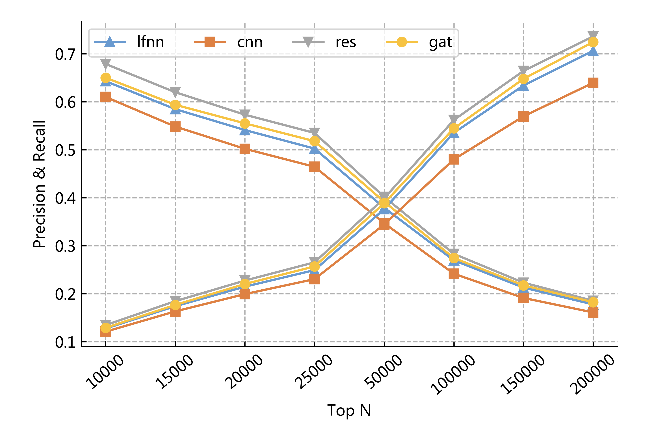
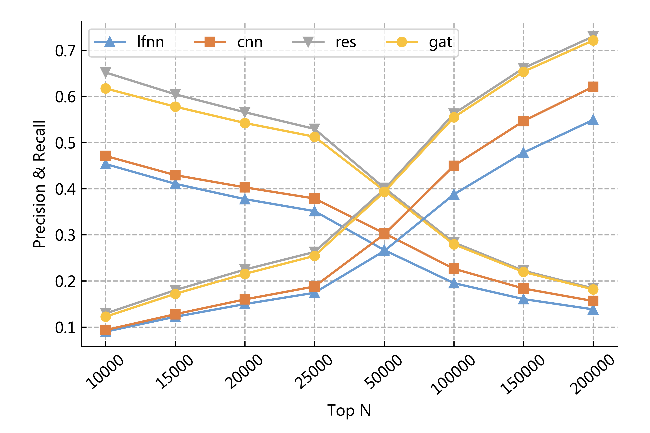
 

图4-10 数据集上的查准率和查全率 图4-11 数据集上的查准率和查全率

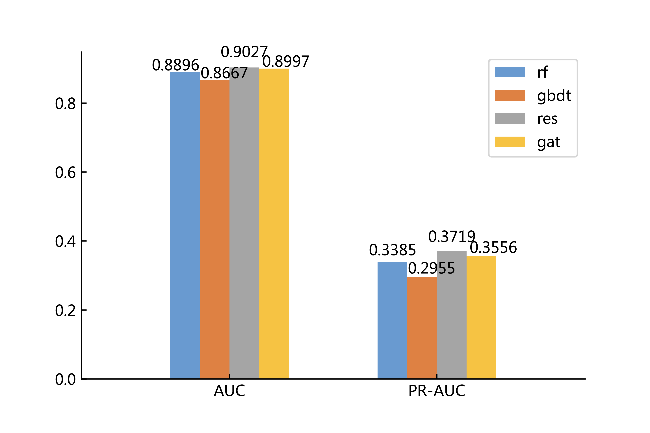
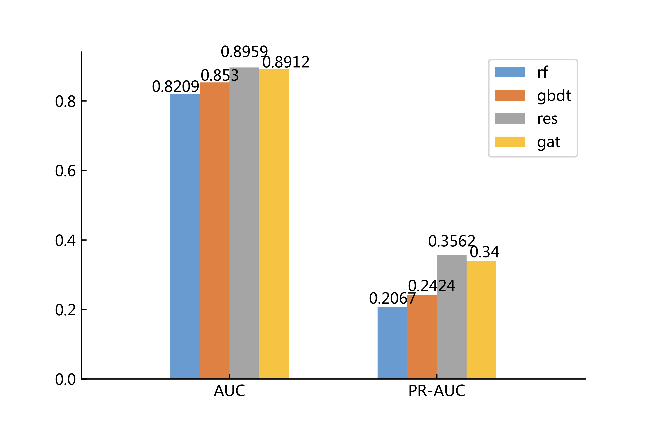
 

图4-12 数据集上的AUC和PR-AUC 图4-13 数据集上的AUC和PR-AUC

可以看到无论是在数据集还是在数据集，本章提出的两个模型在各个评价指标上均优于浅层的神经网络。以Top N取值25000为例，在数据集上的查准率和查全率比高6.49%，比高15.18%，在数据集上的查准率和查全率比高50.77%，比高39.86%。在数据集上的查准率和查全率比高3.09%，比高11.5%，在数据集上的查准率和查全率比高45.93%，比高35.37%。和在AUC和PR-AUC的表现同样优于两个浅层神经网络，对于数据集，本章提出的模型在AUC上最多比浅层神经网络提升4.15%，在PR-AUC上提升25.85%。对于数据集，本章提出的模型在AUC上最多比浅层神经网络提升9.14%，在PR-AUC上提升72.33%。

浅层网络的参数较少，相比于深层的网络，在拟合小数据量的数据时不容易出现过拟合的情况。然而随着数据量的增加，浅层网络开始欠拟合，性能明显比深层网络差。综上所述，本章提出的深层的神经网络相比于浅层网络在性能上有明显的提升，由此可以证明在本文给出的深层的网络在离网预测任务上性能确实优于浅层网络。

* + 1. **与树型模型比较**

前两个实验验证了我们给出的神经网络结构解决了在离网预测任务中网络过度泛化和网络层数较浅的问题。由于目前工业界在离网预测任务上通常使用树型模型，在本节实验中验证我们的神经网络结构是否能优于传统的树型模型。我们画出、、和 4个模型在不同数据集上的对比图，对比图见图4-14到图4-17：

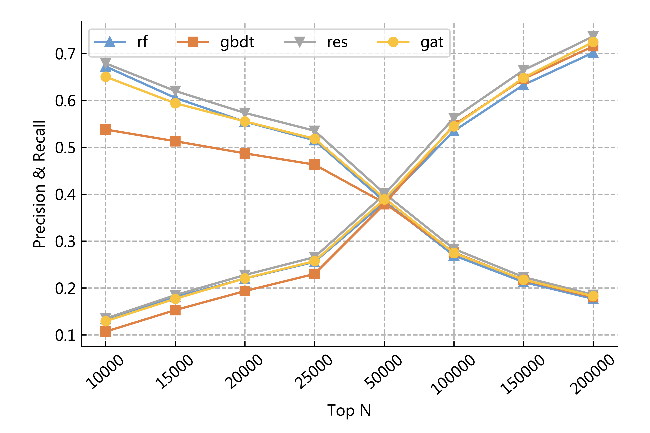
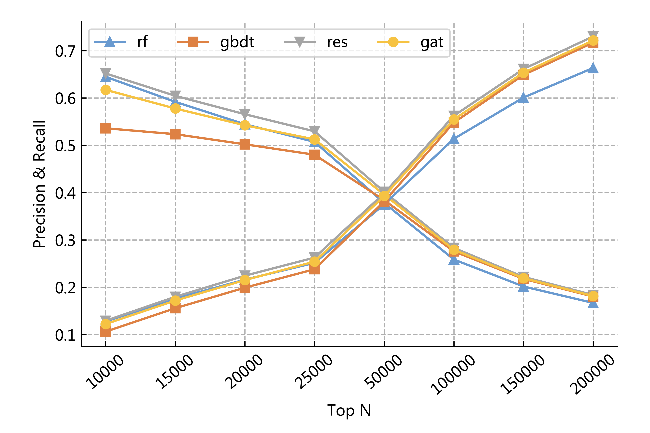
 

图4-14 数据集上的查准率和查全率 图4-15 数据集上的查准率和查全率

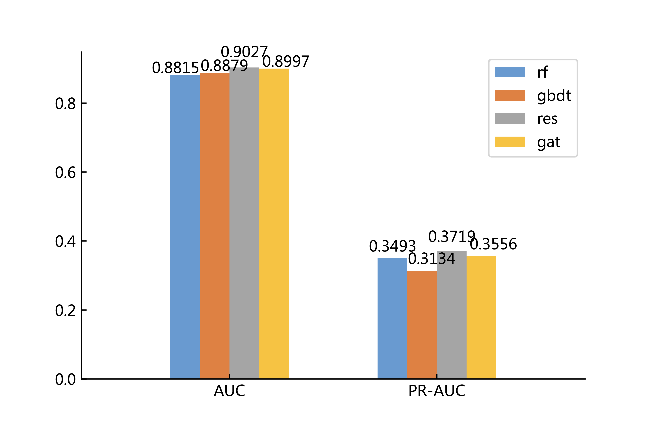
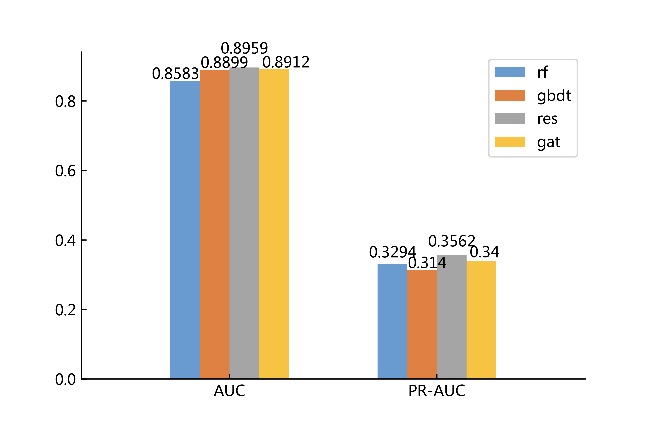
 

图4-16 数据集上的AUC和PR-AUC 图4-17 数据集上的AUC和PR-AUC

可以看到本章提出的两个模型在查准率和查全率上性能略好于，当Top N取值小于50000时，在查准率和查全率上性能明显优于。以Top N取值25000为例，在数据集上的查准率和查全率比高3.94%，比高15.6%，在数据集上的查准率和查全率比高4.4%，比高10.29%。在数据集上的查准率和查全率比高0.62%，比高11.91%，在数据集上的查准率和查全率比高1.04%，比高6.75%。和在AUC和PR-AUC的表现同样优于两个树型模型，对于数据集，本章提出的模型在AUC上最多比树型模型提升2.4%，在PR-AUC上提升18.7%。对于数据集，本章提出的模型在AUC上最多比树型模型提升4.38%，在PR-AUC上提升13.44%。

综上所述，在使用相同数据训练模型的情况下，本章提出的神经网络模型比目前工业界常用的树型模型有更好的效果。这是因为带卷积结构的神经网络在处理时间序列特征时可以捕获到更多的信息，从而更好的理解数据。可以认为本章提出的基于神经网络的离网预测模型有一定的应用价值。

* 1. 本章小结

本章提出了一种基于神经网络的离网预测模型，网络由宽度模型和深度模型两个部分组成。在网络模型的宽度模型部分对离散型特征做交叉积，深度模型部分提出了两种不同的卷积网络结构和。实验中将提出的模型分别与深度/宽度模型、浅层的神经网络以及目前工业界常用的树型模型对比，实验证明本章提出的两个模型在查准率、查全率、AUC和PR-AUC上均比baseline的模型高，在一定程度上解决了离网预测任务上神经网络过度泛化和层数较浅这两个问题。

1. 融合多粒度时序特征的离网预测模型

在目前的离网预测的研究和应用中，通常首先从数据库中选择部分特征，对于时序特征，使用一种时间粒度对这些特征进行聚合，然后尝试各种特征工程方法，最后使用机器学习算法对聚合的数据进行建模。在学术界中，主要的工作集中在设计分类器和模型融合方面，在工业界中则相反，大部分工作都在特征工程上。然而无论学术界还是工业界，目前在聚合时序特征阶段都只使用一种时间粒度来聚合，没有考虑使用多种时间粒度来共同聚合时序特征[56]。本章主要尝试使用特征级融合和决策级融合的方法来同时利用月和日两种时间粒度聚合的特征，实验证明，本章提出的特征级融合+决策级融合的融合方法在PR-AUC上比未进行融合的模型提升了21.94%。

* 1. 特征级融合

本节首先介绍了特征级融合的概念，然后基于特征级融合的特点改进第四章图 4-1的网络结构框架，来将特征级融合应用到离网预测任务中。

* + 1. **特征级融合概念**



图5-1 特征级融合

特征级融合即特征层面上的融合，将提取的特征进行融合与分析[57-58]。特征级融合是中间层次的融合，融合的过程如图5-1所示。首先从原始数据中使用不同的方式提取特征，然后将提取的特征进行融合，最后在融合的特征上训练模型进行预测。

* + 1. **特征级融合在离网预测模型的应用**

为了能将月粒度时序特征和日粒度时序特征进行特征级融合并同时输入到一个模型中，我们改进第四章图4-1中的网络框架图，改进后的框架图如图5-2。从图中可以看到，在网络的深度模型部分，我们在原有框架使用一个卷积网络处理时序特征的基础上新加入一个卷积网络，两个卷积网络分别用来处理月粒度的时序特征和日粒度的时序特征。我们将两个卷积网络的输出和静态特征拼接，共同输入到第一个全连接层。



图5-2 改进后的模型框架图

在本节实验中我们尝试以下两种特征级融合方式：

1、新模型框架的卷积网络部分使用残差网络（）；

2、新模型框架的卷积网络部分使用门卷积网络（）。

* 1. 决策级融合

本节首先介绍了决策级融合的概念，然后基于决策级融合的特点给出在离网预测任务中应用决策级融合的方法。

* + 1. **决策级融合概念**

决策级融合是指将在每个数据集上训练的模型的预测结果进行融合，实现共同预测的过程[59-60]。决策级融合是一种高层次的融合，理论上这种共同决策的方式比单个结果更精确。图5-3是决策级融合的整个过程，首先从数据集中提取特征，对每一种特征训练模型进行预测，然后将各个预测结果进行融合，得到最终的预测结果。



图5-3 决策级融合

* + 1. **决策级融合在离网预测模型的应用**

为了将月粒度时序特征和日粒度时序特征进行决策级融合，我们使用第四章提出的神经网络模型框架，首先在月粒度的时序特征+静态特征上训练模型，然后在日粒度的时序特征+静态特征训练模型，最后对两部分模型的预测概率取平均值融合，得到最终的结果。为了验证决策级融合后性能的提升是由不同时间粒度的时序特征带来的，而不是模型的差异导致的，在本节的融合实验中，我们尝试以下两种决策级融合方式：

1、网络模型框架中卷积网络部分为残差网络，分别使用月粒度时序特征+静态特征和日粒度时序特征+静态特征训练模型，在进行预测时对两个模型的预测概率取平均值（）；

2、网络模型框架中卷积网络部分为门卷积网络，分别使用月粒度时序特征+静态特征和日粒度时序特征+静态特征训练模型，在进行预测时对两个模型的预测概率取平均值（）；

* 1. 同时使用特征级融合和决策级融合

我们也可以将两种融合方式结合起来，将特征级融合中训练的模型作为决策级融合的基模型，在决策级融合阶段与其他模型进行融合。本节的融合实验中，我们尝试以下三种融合方式：

1、对特征级融合中实验1的模型与决策级融合中实验1的两个模型进行决策级融合，在进行预测时对3个模型的预测概率取平均值（）；

2、对特征级融合中实验2的模型与决策级融合中实验2的两个模型进行决策级融合，在进行预测时对3个模型的预测概率取平均值（）；

3、对特征级融合中实验1和2的模型与决策级融合中实验1和2的模型进行决策级融合，在进行预测时对6个模型的预测概率取平均值（）。

其中实验3验证在融合多粒度时序特征提高模型性能的前提下，融合不同网络结构的模型是否同样能提高性能。

* 1. 实验设置
     1. **数据准备及评价指标**

我们使用3.1节中准备好的月粒度时序特征、日粒度时序特征和静态特征数据进行训练。其中每个模型具体使用哪些数据训练见5.1到5.3节。

实验的评价指标使用3.2节中介绍的查准率、查全率AUC和PR-AUC。

* + 1. **实验环境**

本节所有的融合实验均在配置为2个Intel E5-2620、128GB内存、3个GeForce GTX 1080Ti显卡的计算机上进行。

* + 1. **模型训练参数设置**

模型的训练参数设置与第四章中的参数设置相同。模型的损失函数使用交叉熵损失，优化算法为Adam，优化算法的学习率设置为0.001，设置为0.9，设置为0.999，设置为1e-7。采用mini-batch的方式训练，batch size的大小为128。在模型训练阶段，每训练1000个batch，我们计算一次模型在验证集上的损失，如果本次计算的验证集损失小于上次计算的损失，就保存当前训练的模型。同时还使用了Earlystopping来防止模型过拟合，加速训练过程。当验证集的损失连续100次计算都没有下降，则提前终止训练过程。

* + 1. **对比实验**

我们将第四章中提出的模型作为本章的对比模型，用于验证融合多个时间粒度特征后的有效性，以下为对比模型的介绍：

1、模型输入为月粒度时序特征+静态特征，深度模型的卷积部分为残差网络（）；

2、模型输入为日粒度时序特征+静态特征，深度模型的卷积部分为残差网络（）；

3、模型输入为月粒度时序特征+静态特征，深度模型的卷积部分为门卷积网络（）；

4、模型输入为日粒度时序特征+静态特征，深度模型的卷积部分为门卷积网络（）。

* 1. 实验结果分析

表5-1到表5-3是4个baseline模型和7个融合了多粒度时序特征的模型在查准率、查全率、AUC和PR-AUC上的表现：

表5-1 模型的查准率

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **top N**  **model** | **10000** | **15000** | **20000** | **25000** | **50000** | **100000** | **150000** | **200000** |
|  | 0.6787 | 0.6198 | 0.5730 | 0.5350 | 0.4018 | 0.2831 | 0.2231 | 0.1854 |
|  | 0.6518 | 0.6045 | 0.5656 | 0.5295 | 0.4017 | 0.2833 | 0.2221 | 0.1839 |
|  | 0.6501 | 0.5937 | 0.5550 | 0.5179 | 0.3916 | 0.2744 | 0.2175 | 0.1825 |
|  | 0.6172 | 0.5777 | 0.5423 | 0.5125 | 0.3959 | 0.2794 | 0.2194 | 0.1817 |
|  | 0.6667 | 0.6242 | 0.5851 | 0.5541 | 0.4335 | 0.3022 | 0.2344 | 0.1929 |
|  | 0.6746 | 0.6267 | 0.5898 | 0.5556 | 0.4325 | 0.3021 | 0.2339 | 0.1922 |
|  | 0.7001 | 0.6419 | 0.5965 | 0.5576 | 0.4235 | 0.2939 | 0.2296 | 0.1902 |
|  | 0.6809 | 0.6270 | 0.5850 | 0.5486 | 0.4196 | 0.2928 | 0.2281 | 0.1888 |
|  | 0.7067 | 0.6511 | 0.6078 | 0.5698 | 0.4357 | 0.3013 | 0.2346 | 0.1928 |
|  | 0.6947 | 0.6454 | 0.6036 | 0.5658 | 0.4364 | 0.3018 | 0.2345 | 0.1931 |
|  | 0.7071 | 0.6561 | 0.6113 | 0.5749 | 0.4419 | 0.3052 | 0.2368 | 0.1945 |

表5-2 模型的查全率

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **top N**  **model** | **10000** | **15000** | **20000** | **25000** | **50000** | **100000** | **150000** | **200000** |
|  | 0.1347 | 0.1846 | 0.2275 | 0.2655 | 0.3988 | 0.5620 | 0.6642 | 0.7361 |
|  | 0.1294 | 0.1800 | 0.2246 | 0.2628 | 0.3987 | 0.5624 | 0.6612 | 0.7300 |
|  | 0.1290 | 0.1768 | 0.2203 | 0.2570 | 0.3887 | 0.5446 | 0.6477 | 0.7244 |
|  | 0.1225 | 0.1720 | 0.2153 | 0.2543 | 0.3930 | 0.5546 | 0.6534 | 0.7214 |
|  | 0.1323 | 0.1859 | 0.2323 | 0.2750 | 0.4302 | 0.5998 | 0.6980 | 0.7659 |
|  | 0.1339 | 0.1866 | 0.2342 | 0.2757 | 0.4293 | 0.5997 | 0.6964 | 0.7632 |
|  | 0.1390 | 0.1911 | 0.2368 | 0.2767 | 0.4203 | 0.5834 | 0.6835 | 0.7551 |
|  | 0.1352 | 0.1867 | 0.2323 | 0.2722 | 0.4165 | 0.5812 | 0.6791 | 0.7494 |
|  | 0.1403 | 0.1939 | 0.2413 | 0.2828 | 0.4324 | 0.5981 | 0.6985 | 0.7655 |
|  | 0.1379 | 0.1922 | 0.2396 | 0.2808 | 0.4331 | 0.5991 | 0.6982 | 0.7668 |
|  | 0.1404 | 0.1954 | 0.2427 | 0.2853 | 0.4386 | 0.6058 | 0.7051 | 0.7721 |

表5-3 模型的AUC和PR-AUC

|  |  |  |
| --- | --- | --- |
| **model** | **AUC** | **PR-AUC** |
|  | 0.9027 | 0.3719 |
|  | 0.8959 | 0.3562 |
|  | 0.9050 | 0.3749 |
|  | 0.8912 | 0.3400 |
|  | 0.9128 | 0.3933 |
|  | 0.9120 | 0.3903 |
|  | 0.9081 | 0.3955 |
|  | 0.9063 | 0.3867 |
|  | 0.9128 | 0.4076 |
|  | 0.9134 | 0.4055 |
|  | 0.9158 | 0.4146 |

* + 1. **特征级融合实验结果**

本节验证对不同粒度的时序特征进行特征级融合是否能提高模型性能。我们比较了进行特征级融合的模型和没有进行融合的模型，对比图见图5-4到图5-5。

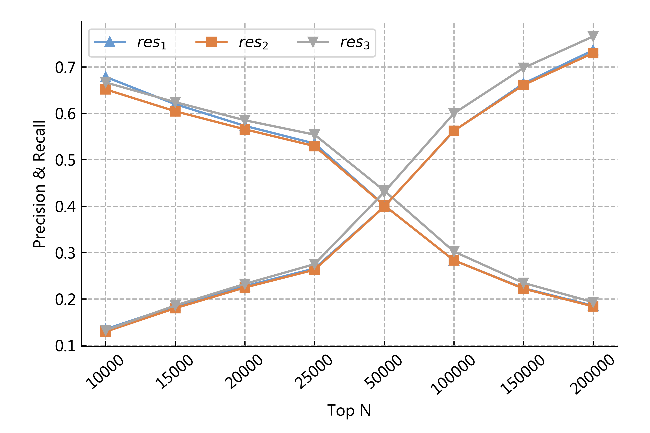
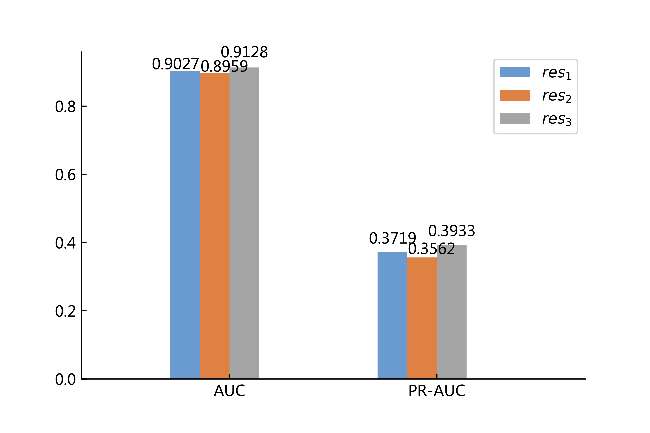
 

图5-4 在上进行特征级融合的性能比较

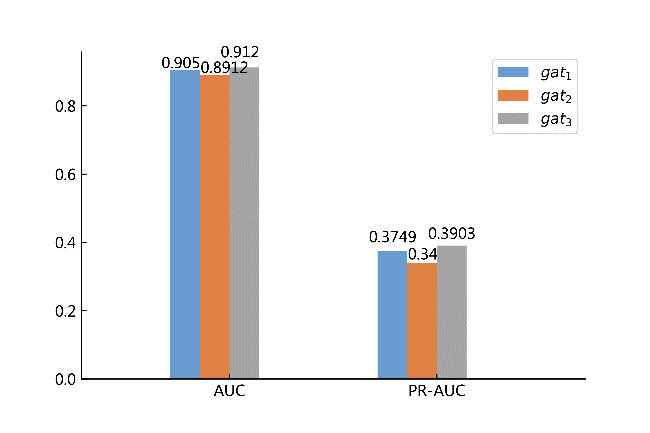
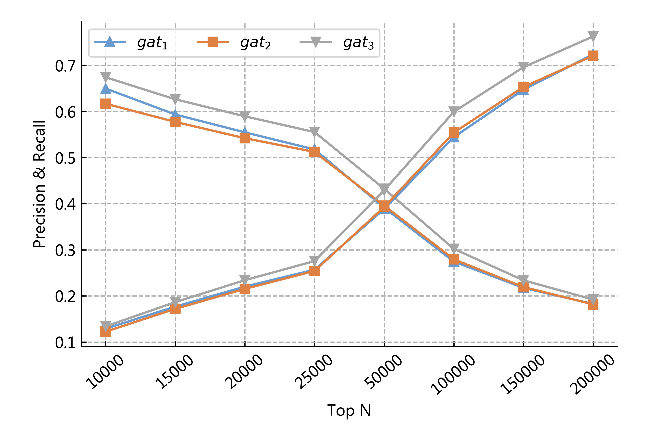


图5-5 在上进行特征级融合的性能比较

可以看到无论是还是，在进行特征级融合后性能均有不同程度的提升，以Top N取值25000为例，在上进行特征级融合后查准率和查全率相比没有进行融合的模型最高提升4.65%，AUC上最高提升1.89%，PR-AUC上最高提升10.41%。在上进行特征级融合后查准率和查全率相比没有进行融合的模型最高提升8.41%，AUC上最高提升2.33%，PR-AUC上最高提升14.79%。

因此我们认为对多粒度时序特征进行特征级融合后可以提升模型的性能。可以将不同时间粒度的时序特征看作是不同的刻画数据的角度，将不同时间粒度特征进行特征级融合后模型可以更加全面的了解数据，从而有更好的性能。

* + 1. **决策级融合实验结果**

本节验证对不同粒度的时序特征进行决策级融合是否能提高模型性能。我们比较了进行决策级融合的模型和没有进行融合的模型，对比图见图5-6到图5-7。

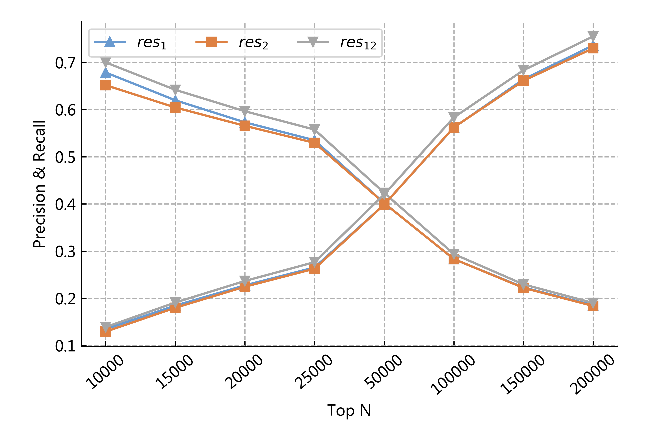
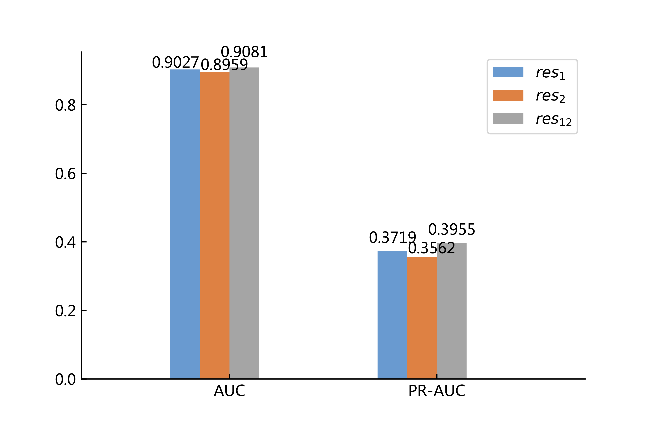
 

图5-6 在上进行决策级融合的性能比较

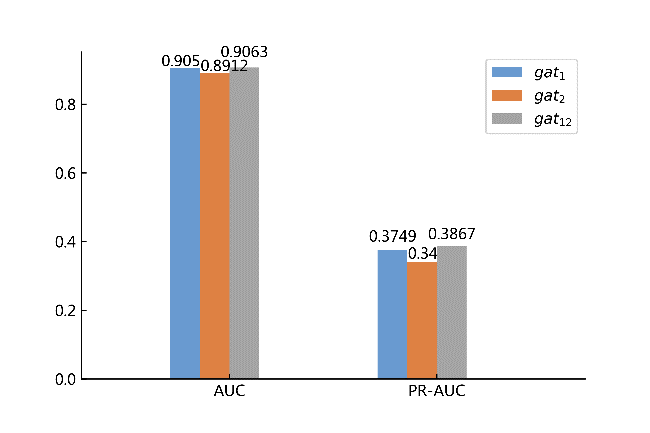
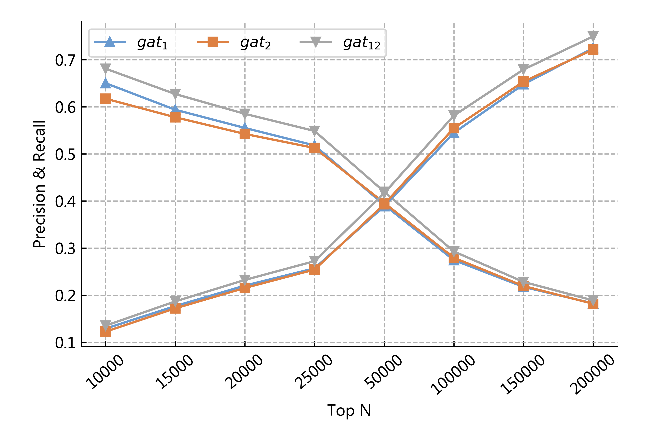


图5-7 在上进行决策级融合的性能比较

同样的，无论是还是，在进行决策级融合后性能均有不同程度的提升，以Top N取值25000为例，在上进行决策级融合后查准率和查全率相比没有进行融合的模型最高提升5.31%，AUC上最高提升1.36%，PR-AUC上最高提升10.03%。在上进行决策级融合后查准率和查全率相比没有进行融合的模型最高提升3.61%，AUC上最高提升1.69%，PR-AUC上最高提升13.74%。

可以看到对多粒度时序特征进行决策级融合后同样可以提升模型的性能。在决策级融合中，每个模型负责观察数据的一个视角，对模型进行决策融合的过程就可以看作是结合过个视角观察数据，从而达到比单个视角更好的效果。

* + 1. **特征级融合+决策级融合实验结果**

本节实验中我们尝试将特征级融合和决策级融合两种融合方法结合起来，并观察结合后的模型是否能在只使用一种融合方法的基础上进一步提升性能。图5-8到 图5-9是实验结果。

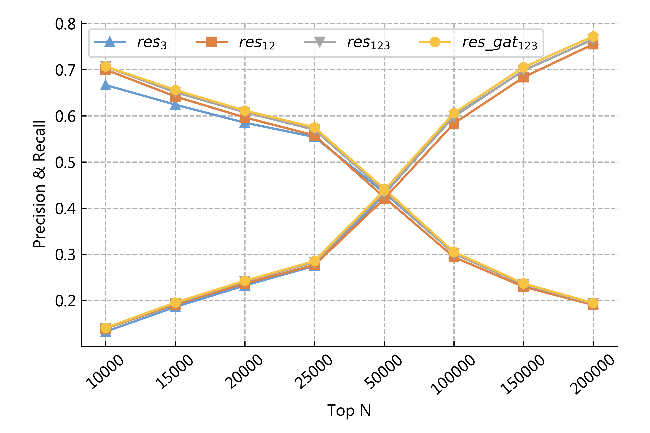
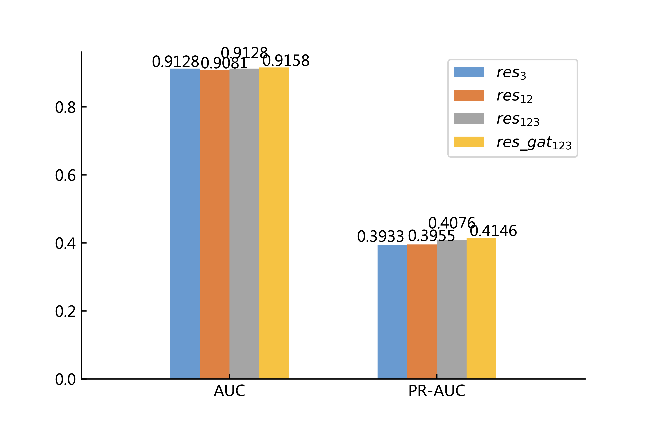
 

图5-8 在上进行特征级融合+决策级融合的性能比较

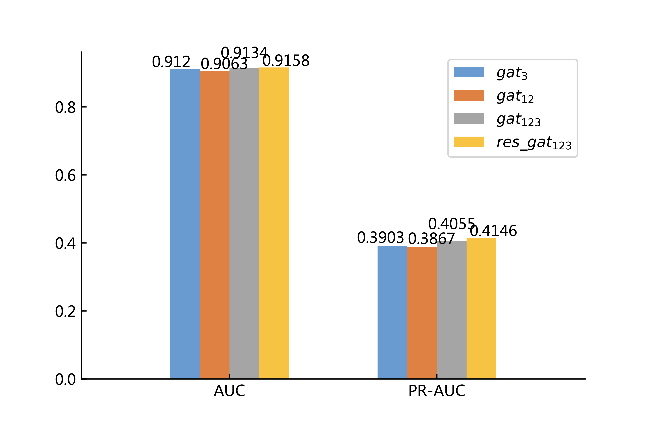
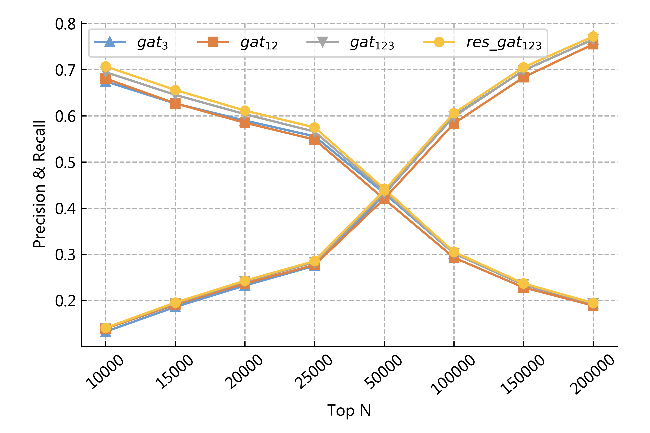


图5-9 在上进行特征级融合+决策级融合的性能比较

可以看到无论是还是，在进行特征级融合+决策级融合后均能进一步提升性能。以Top N取值25000为例，在上进行特征级融合+决策级融合后查准率和查全率相比只进行了一种融合的模型最高提升2.83%，PR-AUC上最高提升3.64%。在上进行特征级融合+决策级融合后查准率和查全率相比只进行了一种融合的模型最高提升3.14%，PR-AUC上最高提升4.86%。另外还观察到在进行了特征级融合+决策级融合的基础上，进一步融合不同结构的神经网络后，性能得到了再次提升，在PR-AUC上性能比只进行了一种融合的模型最高提升7.21%。

因此可以得出结论，在不同时间粒度的时序特征上同时进行特征级融合+决策级融合，相比于只进行一种融合的方法，可以进一步提升性能。将两种方法融合起来，可以使模型更好的观察数据。同时在决策级融合中加入不同结构的神经网络，相当于使用了不同的模型观察数据，相比于只使用一种模型观察数据，多个模型共同观察数据可以得到更真实的结果。

* 1. 本章小结

本章针对现有方法只利用一种时间粒度来聚合时序特征的不足，提出使用特征级融合和决策级融合两种方法来同时利用多种时间粒度特征。介绍了两种融合不同时间粒度特征的方法，然后将这些方法与没有进行过任何融合的方法作对比。实验证明，无论使用哪种融合方法，融合了多种不同时间粒度的特征后均能有效提高模型性能，并且同时使用特征级融合+决策级融合，且融合不同结构的神经网络时性能最好，在PR-AUC上比第二章中提出的性能最好的神经网络模型高10.59%，比树型模型中性能最好的模型高18.69%。

1. 总结与展望
   1. 工作总结

针对现有的离网预测研究中出现的一系列问题，本文在电信运营商提供的真实用户数据上，结合了现有数据的特点，设计并构造了一种基于宽度&深度学习的离网预测模型框架，这种神经网络相比于现有的应用于离网预测的神经网络，缓解了离网预测任务中神经网络过度泛化的问题，同时框架的深度模型部分使用了层数更深的网络来代替现有的浅层的网络，能更好的拟合数据。另外针对现有的应用于离网预测的神经网络只使用一种时间粒度来聚合时序特征的问题，本文探索了同时利用多种时间粒度特征进行建模的方法。本文的工作主要包括以下几个方面：

1）本文首先提出了一种基于宽度&深度学习的离网预测模型框架。当神经网络的输入包含离散型特征时，网络可能会存在过度泛化的问题。而加入了交叉积转换的线性模型可以有效的记忆特征间的共现关系，不会有泛化问题。我们借鉴宽度&深度学习的思想，将线性模型作为宽度模型，神经网络作为深度模型，充分发挥两者优势，两部分模型共同决策。实验证明，加入了线性模型的神经网络相比于没有线性模型的神经网络，在PR-AUC上最高可以提升5.39%。

2）本文针对现有的基于神经网络的离网预测模型层数都较浅的情况，提出构造较深层的神经网络来充分拟合大数据。经过大量实验我们选用了残差网络和门卷积网络两种结构来构造深层网络。其中残差网络具有更好的性能，而门卷积网络则在性能略差的情况下有更短的训练时间。实验证明，基于深层网络的离网预测模型相比于基于浅层网络的离网预测模型，在PR-AUC上最高可以提升72.33%。

3）基于以上工作提出的神经网络模型，本文提出将多种时间粒度聚合的时序特征同时引入到神经网络中来进一步提高模型性能。本文尝试了特征级融合和决策级融合两种融合方式。对于特征级融合，我们改造提出的神经网络的结构，将不同时间粒度的特征同时输入到一个神经网络中，每种时间粒度的特征使用一个卷积网络处理，最后将各个卷积层的输出进行拼接。对于决策级融合，我们使用每种时间粒度的特征各训练一批模型，将各个模型的预测值取平均值作为输出。最后我们将两种融合方式结合，得到最终的输出。实验结果证明，该方法的预测结果与只使用一种时间粒度训练的最好的模型相比，在PR-AUC上可以提升10.59%。

* 1. 工作展望

本文提出的融合了多粒度时序特征的基于宽度&深度学习的离网预测模型在实验中取得了较好的性能，但在训练和使用过程中仍然存在一些问题有待解决，具体体现在以下几个方面：

1）本文提出的基于神经网络的离网预测模型虽然在性能上要比随机森林和GBDT等常用的树型模型好，但在训练时间上远比随机森林慢，也略慢于GBDT。同时由于网络层数较深的缘故，与浅层模型相比，本文的网络模型存在较严重的过拟合现象。接下来的工作需要继续优化网络模型，在提升模型训练速度的同时加入更多防止过拟合的机制。

2）本文的网络模型分为宽度模型和深度模型两部分，在宽度模型部分目前只是简单的选取少量离散型特征做交叉积，选取的特征较少，且做特征的方式比较单一。接下来的工作中需要在宽度模型中使用不同的特征提取的方式加入更多的特征。

3）本文在使用决策级融合的方法融合不同时间粒度特征训练的模型时，只是简单的对预测结果取平均值，没有充分考虑到各个模型本身的性能对最终预测结果的影响，接下来的工作中可以在模型融合时根据各个模型的性能，对预测结果进行加权求和。

参考文献

1. Coussement K, Van den Poel D. Churn prediction in subscription services: An application of support vector machines while comparing two parameter-selection techniques [J]. Expert systems with applications, 2008, 34(1): 313-327
2. Verbeke W, Dejaeger K, Martens D, et al. New insights into churn prediction in the telecommunication sector: A profit driven data mining approach [J]. European Journal of Operational Research, 2012, 218(1): 211-229.
3. Reinartz W J, Kumar V. The impact of customer relationship characteristics on profitable lifetime duration [J]. Journal of marketing, 2003, 67(1): 77-99.
4. Guyon I, Lemaire V, Boullé M, et al. Design and analysis of the kdd cup 2009: fast scoring on a large orange customer database [J]. ACM SIGKDD Explorations Newsletter, 2010, 11(2): 68-76.
5. Yu H F, Lo H Y, Hsieh H P, et al. Feature Engineering and Classifier Ensemble for KDD Cup 2010 [A]. Proceedings of the KDD Cup 2010 Workshop [C]. Washington DC, USA: ACM, 2010
6. Umayaparvathi V, Iyakutti K. Automated Feature Selection and Churn Prediction using Deep Learning Models [J]. International Research Journal of Engineering and Technology, 2017, 4(3): 1846-1854.
7. Castanedo F, Valverde G, Zaratiegui J, et al. Using deep learning to predict customer churn in a mobile telecommunication network[DB/OL]. http://www.wiseathena.com /pdf/wa\_dl.pdf, 2014.
8. Wangperawong A, Brun C, Laudy O, et al. Churn analysis using deep convolutional neural networks and autoencoders [J]. arXiv preprint arXiv:1604.05377, 2016.
9. Zaratiegui J, Montoro A, Castanedo F. Performing highly accurate predictions through convolutional networks for actual telecommunication challenges [J]. arXiv preprint arXiv:1511.04906, 2015.
10. Dasgupta K, Singh R, Viswanathan B, et al. Social ties and their relevance to churn in mobile telecom networks [A] Proceedings of the 11th international conference on Extending database technology: Advances in database technology [C]. Nantes, France: ACM, 2008.
11. Karnstedt M, Rowe M, Chan J, et al. The effect of user features on churn in social networks [A]. Proceedings of the 3rd International Web Science Conference [C]. Koblenz, Germany: ACM, 2011.
12. Pushpa S. An efficient method of building the telecom social network for churn prediction [J]. International Journal of Data Mining & Knowled Management Process, 2012, 2(3): 31-39.
13. Phua C, Cao H, Gomes J B, et al. Predicting near-future churners and win-backs in the telecommunications industry [J]. arXiv preprint arXiv:1210.6891, 2012.
14. Stripling E, vanden Broucke S, Antonio K, et al. Profit maximizing logistic regression modeling for customer churn prediction [A]. Data Science and Advanced Analytics (DSAA), 2015. 36678 2015. IEEE International Conference on [C]. Paris, France: IEEE, 2015.
15. Lu Ning, Lin Hua, Lu Jie, et al. A customer churn prediction model in telecom industry using boosting [J]. IEEE Transactions on Industrial Informatics, 2014, 10(2): 1659-1665.
16. Owczarczuk M. Churn models for prepaid customers in the cellular telecommunication industry using large data marts [J]. Expert Systems with Applications, 2010, 37(6): 4710-4712.
17. Zhao Xi, Shi Yong, Lee Jongwon, et al. Customer Churn Prediction Based on Feature Clustering and Nonparallel Support Vector Machine [J]. International Journal of Information Technology & Decision Making, 2014, 13(05): 1013-1027.
18. Shaaban E, Helmy Y, Khedr A, et al. A proposed churn prediction model [J]. IJERA, 2012, 2: 693-697.
19. Abbasimehr H, Setak M, Tarokh M J. A comparative assessment of the performance of ensemble learning in customer churn prediction [J]. Int. Arab J. Inf. Technol., 2014, 11(6): 599-606.
20. binti Oseman K, Haris N A, bin Abu Bakar F. Data Mining in Churn Analysis Model for Telecommunication Industry [J]. Journal of Statistical Modeling and Analytics Vol, 2010, 1(19-27).
21. Kirui C, Hong Li, Cheruiyot W, et al. Predicting customer churn in mobile telephony industry using probabilistic classifiers in data mining [J]. IJCSI International Journal of Computer Science Issues, 2013, 10(2): 1694-0784.
22. Huang Y, Zhu F, Yuan M, et al. Telco churn prediction with big data [A]. Proceedings of the 2015 ACM SIGMOD international conference on management of data [C]. Melbourne, Victoria, Australia: ACM, 2015.
23. Idris A, Khan A. Ensemble Based Efficient Churn Prediction Model for Telecom [A]. Frontiers of Information Technology (FIT), 2014 12th International Conference on [C]. Islamabad, Pakistan: IEEE, 2014.
24. Prashanth R, Deepak K, Meher A K. High accuracy predictive modelling for customer churn prediction in telecom industry [A]. 6th International Conference on Machine Learning and Data Mining in Pattern Recognition [C]. Leipzig, Germany: Springer, 2017
25. Zhang R, Li W, Tan W, et al. Deep and Shallow Model for Insurance Churn Prediction Service [A]. Services Computing (SCC), 2017 IEEE International Conference on [C]. Honolulu, USA: IEEE, 2017
26. Verbeke W, Martens D, Mues C, et al. Building comprehensible customer churn prediction models with advanced rule induction techniques [J]. Expert Systems with Applications, 2011, 38(3): 2354-2364.
27. Kim N, Jung K H, Kim Y S, et al. Uniformly subsampled ensemble (USE) for churn management: Theory and implementation [J]. Expert Systems with Applications, 2012, 39(15): 11839-11845.
28. Wei C P, Chiu I T. Turning telecommunications call details to churn prediction: a data mining approach [J]. Expert systems with applications, 2002, 23(2): 103-112.
29. 石鸿斌,严建峰,白瑞瑞,等. 多粒度时序特征在离网预测中的应用 [J/OL]. 计算机应用研究, 2019(04): 1-7.
30. Olle G D O, Cai S. A hybrid churn prediction model in mobile telecommunication industry [J]. International Journal of e-Education, e-Business, e-Management and e-Learning, 2014, 4(1): 55.
31. Vafeiadis T, Diamantaras K I, Sarigiannidis G, et al. A comparison of machine learning techniques for customer churn prediction [J]. Simulation Modelling Practice and Theory, 2015, 55: 1-9.
32. Lu N, Lin H, Lu J, et al. A customer churn prediction model in telecom industry using boosting [J]. IEEE Transactions on Industrial Informatics, 2014, 10(2): 1659-1665.
33. Azeem M, Usman M, Fong A C M. A churn prediction model for prepaid customers in telecom using fuzzy classifiers [J]. Telecommunication Systems, 2017, 66(4): 603-614.
34. Idris A, Khan A. Churn prediction system for telecom using filter–wrapper and ensemble classification [J]. The Computer Journal, 2016, 60(3): 410-430.
35. Cheng H T, Koc L, Harmsen J, et al. Wide & deep learning for recommender systems [A]. Proceedings of the 1st Workshop on Deep Learning for Recommender Systems [C]. Boston, MA, USA: ACM, 2016.
36. Breiman L. Random forests [J]. Machine learning, 2001, 45(1): 5-32.
37. Breiman L. Bagging predictors [J]. Machine learning, 1996, 24(2): 123-140.
38. Friedman J H. Greedy function approximation: a gradient boosting machine [J]. Annals of statistics, 2001: 1189-1232.
39. Chen T, Guestrin C. Xgboost: A scalable tree boosting system [A]. Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining [C]. San Francisco, California, USA: ACM, 2016
40. McCulloch W S, Pitts W. A logical calculus of the ideas immanent in nervous activity [J]. The bulletin of mathematical biophysics, 1943, 5(4): 115-133.
41. Glorot X, Bordes A, Bengio Y. Deep sparse rectifier neural networks [A]. Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics [C]. Fort Lauderdale, Florida, USA: JMLR, 2011
42. Hubel D H, Wiesel T N. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex [J]. The Journal of physiology, 1962, 160(1): 106-154.
43. Fukushima K, Miyake S. Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition [M]//Competition and cooperation in neural nets. Springer, Berlin, Heidelberg, 1982: 267-285.
44. LeCun Y, Bottou L, Bengio Y, et al. Gradient-based learning applied to document recognition [J]. Proceedings of the IEEE, 1998, 86(11): 2278-2324.
45. Krizhevsky A, Sutskever I, Hinton G E. Imagenet classification with deep convolutional neural networks [A]. Proceedings of the 25th International Conference on Neural Information Processing Systems [C]. Lake Tahoe, Nevada, USA: Curran Associates Inc., 2012
46. Bengio Y, Simard P, Frasconi P. Learning long-term dependencies with gradient descent is difficult [J]. IEEE transactions on neural networks, 1994, 5(2): 157-166.
47. Glorot X, Bengio Y. Understanding the difficulty of training deep feedforward neural networks [J]. Journal of Machine Learning Research, 2010, 9: 249-256.
48. LeCun Y, Bottou L, Orr G, et al. Efficient BackProp in Neural Networks: Tricks of the Trade (Orr, G. and Müller, K., eds.) [J]. Lecture Notes in Computer Science, 1524.
49. Saxe A M, McClelland J L, Ganguli S. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks [J]. arXiv preprint arXiv:1312.6120, 2013.
50. He K, Zhang X, Ren S, et al. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification [A]. Proceedings of the IEEE international conference on computer vision [C]. Santiago, Chile: IEEE, 2015.
51. He K, Zhang X, Ren S, et al. Deep residual learning for image recognition [A]. Proceedings of the IEEE conference on computer vision and pattern recognition [C]. Las Vegas, USA: IEEE, 2016.
52. Dauphin Y N, Fan A, Auli M, et al. Language modeling with gated convolutional networks [J]. arXiv preprint arXiv:1612.08083, 2016.
53. Thusoo A, Sarma J S, Jain N, et al. Hive: a warehousing solution over a map-reduce framework [J]. Proceedings of the VLDB Endowment, 2009, 2(2): 1626-1629.
54. Armbrust M, Xin R S, Lian C, et al. Spark sql: Relational data processing in spark [A]. Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data [C]. Melbourne, Victoria, Australia: ACM, 2015
55. Davis J, Goadrich M. The relationship between Precision-Recall and ROC curves [A]. Proceedings of the 23rd international conference on Machine learning [C]. Pittsburgh, Pennsylvania, USA: ACM, 2006.
56. Zhang J, Zheng Y, Qi D. Deep Spatio-Temporal Residual Networks for Citywide Crowd Flows Prediction [A]. Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17) [C]. San Francisco, USA: AAAI, 2017
57. Luo R C, Yih C C, Su K L. Multisensor fusion and integration: approaches, applications, and future research directions [J]. IEEE Sensors journal, 2002, 2(2): 107-119.
58. Fortmann T E, Bar-Shalom Y, Scheffe M. Multi-target tracking using joint probabilistic data association [A]. Proceedings of the 19th IEEE Conference on Decision and Control including the Symposium on Adaptive Processes [C]. Albuquerque, USA: IEEE, 1980.
59. Bossé É, Valin P, Boury-Brisset A C, et al. Exploitation of a priori knowledge for information fusion [J]. Information Fusion, 2006, 7(2): 161-175.
60. Morbee M, Tessens L, Aghajan H, et al. Dempster-Shafer based multi-view occupancy maps [J]. Electronics letters, 2010, 46(5): 341-343.

攻读硕士学位期间主要的工作

论文发表情况

1. **石鸿斌**,严建峰,白瑞瑞,徐彩旭,徐广根.多粒度时序特征在离网预测中的应用[J/OL].计算机应用研究,2019(04):1-7 （已录用）。
2. 徐广根,杨璐,严建峰,徐彩旭,**石鸿斌**. 基于LDA主题模型的用户电信轨迹恢复算法[J/OL].计算机应用研究,2019(08):1-8 （已录用）。
3. Caixu Xu, JianFeng Yan, Lu Yang, Guanggen Xu and **Hongbin Shi**. Context Co-occurrence Based Relationship Strength Prediction in Spatiotemporal Data [C]. Proceedings of the 2018 International Conference on Computer and Mechatronics. ICCM 2018. （EI Indexed，已录用）。

软件著作权

1. 大数据分析平台软件V1.0. 登记号：2016SR369257。

专利成果

1. 徐彩旭,徐广根,**石鸿斌**. 基于位置社交网络的用户关系强度预测方法、装置及设备.申请号：201711422233.0。

科研项目

1. 参与“融合文本网数据的深度学习技术研究”，国家自然科学基金(61572339)。
2. 参与“基于超图的主题建模算法研究”，国家自然科学基金(61373092)。
3. 参与“基于时空相关性的无线传感器网络节能问题研究”，国家自然科学基金(61272449)。

致谢

时光荏苒，三年的研究生生活转眼就要接近尾声了。回首这三年的研究生生涯，我学到了很多本科期间没有接触过的知识。我要向所有帮助过我的家人、老师和同学表示感谢。

首先感谢我的导师曾嘉教授，感谢曾老师在我初试成绩出来之后就叫我去公司实习，让我积累了宝贵的工程实践经验，为我研二找实习期间和研三找工作期间奠定了扎实的基础。虽然曾老师不在学校，见面的机会不多，但他给了我充分的时间来学自己想学的东西。

其次感谢严建峰副教授，总是慷慨地为我们提供实验设备，并且在我写软著时提供资金支持。同时也要感谢实验室的同学们，感谢罗晨师兄和白瑞瑞师姐对我科研上指导，感谢周捷师弟帮我修改论文，感谢夏鹏师弟教我自然语言处理方面的知识，以及王猛师弟、居亚亚师妹、徐广根、徐彩旭在学习和生活上提供的指导和帮助。

接着感谢刘梦眙，在我实验和论文遇到困难时，不断给我鼓励，帮助我一起想解决办法，克服难关。

最后，感谢各位专家教授在百忙之中对我的论文进行评审。