论文编号：

基于卷积神经网络（CNN）的网络流量分类

**Network traffic classification based on convolutional neural network (CNN)**

院系名称：

班 级：

学 号：

学生姓名：

指导教师：

2020年 5 月

# 摘要

近年来，随着互联网的快速发展，各种新型的互联网应用接踵而至，网络规模不断扩大，网络流量也日益繁多。网络流量分类技术作为网络管理中的关键手段之一，不仅可以承担分辨加密流量以及恶意流量确保网络安全的任务，还可以为网络管理者提供可靠的网络资源使用情况，以便网络管理者进行合理的资源分配以及科学的调整网络架构。基于端口，深度包检测和经典的机器学习方法等传统网络流量分类技术已被广泛使用，但是由于互联网的急剧变化，特别是加密流量的大大增加，这些方法的准确率已经下降。鉴于深度学习方法在图像分类领域的优良表现，网络流量分类研究学者开始关注于基于深度学习的网络流量分类技术，并且做出了许多尝试，取得了很好的效果。

本次研究中使用卷积神经网络作为流量分类的模型。首先对原始流量数据进行预处理，然后使用深度学习框架Tensorflow完成模型的搭建，接着输入标准数据实现对混合流量分类，模型不仅分辨出10种常规流量以及10种加密流量，还可以分辨出8种常规流量以及2种恶意流量。在此之后，面对数据不平衡问题，在数据预处理阶段本文尝试着从采样角度，交叉验证角度来解决这个问题，做了多组对比实验，F1得分和准确率在原来的基础上都提升了2个百分点。最后对实验数据和结果整理分析，思考此次研究的得失，发现未来可以继续改进的方向。

关键词：流量分类；深度学习；卷积神经网络；Tensorflow

# **Abstract**

In recent years, The Internet has developed rapidly, various new Internet applications have followed one after another, the scale of the network has been continuously expanded, and the network traffic has also become more and more numerous. As one of the key methods in network management, network traffic classification technology can not only undertake the task of distinguishing encrypted traffic and malicious traffic to ensure network security, but also provide network administrators with reliable network resource usage so that network administrators can make reasonable Resource allocation and scientific adjustment of network architecture. Traditional network traffic classification technologies such as port-based, deep packet inspection and classic machine learning methods have been widely used, but due to the rapid changes in the Internet, especially the increase in encrypted traffic, the accuracy of these methods has declined. In view of the excellent performance of deep learning methods in the field of image classification, network traffic classification research scholars have begun to focus on network traffic classification technology based on deep learning, and have made many attempts and achieved good results.

In this study, I will use a convolutional neural network as a model for traffic classification. First, I preprocess the original traffic data, then use the deep learning framework Tensorflow to complete the model construction, and then enter the standard data to classify the mixed traffic. The model not only distinguishes 10 types of conventional traffic and 10 types of encrypted traffic, but also 8 kinds of regular traffic and 2 kinds of malicious traffic. After this, in the face of data imbalance, in the data preprocessing stage, I tried to solve this problem from the perspective of sampling and cross-validation. I did multiple sets of comparative experiments, and the F1 score and accuracy were improved on the original basis. 2 percentage points. Finally, the experimental data and results are sorted and analyzed, and the gains and losses of this study are considered, and the direction of improvement can be found in the future.and the direction of improvement can be found in the future.

**Key words: traffic classification; deep learning; convolutional neural network; Tensorflow**

# **目 录**

[第1章 绪 论 1](#_Toc41162988)

[1.1 研究背景和意义 1](#_Toc41162989)

[1.2 国内外研究现状 1](#_Toc41162990)

[1.2.1 传统网络流量分类技术 2](#_Toc41162991)

[1.2.2 基于深度学习的网络流量分类技术 3](#_Toc41162992)

[1.2.3 卷积神经网络性能差异的主要原因 3](#_Toc41162993)

[1.3 主要工作 5](#_Toc41162994)

[1.4 论文组织结构 6](#_Toc41162995)

[第2章 实验基础 7](#_Toc41162996)

[2.1 实验数据集 7](#_Toc41162997)

[2.1.1 ISCXVPN2016数据集 7](#_Toc41162998)

[2.1.2 USTC-TFC2016数据集 8](#_Toc41162999)

[2.1.3 数据流的结构 9](#_Toc41163000)

[2.2 深度学习框架 10](#_Toc41163001)

[2.3 卷积神经网络 11](#_Toc41163002)

[2.3.1 主要结构 11](#_Toc41163003)

[2.3.2 核心操作（卷积操作和池化操作） 12](#_Toc41163004)

[2.3.3 优点和不足 15](#_Toc41163005)

[2.4 实验环境以及评价标准 17](#_Toc41163006)

[第3章 基于卷积神经网络的流量分类系统 19](#_Toc41163007)

[3.1 系统框架 19](#_Toc41163008)

[3.2 卷积神经网络模型 19](#_Toc41163009)

[3.3 预处理模块 20](#_Toc41163010)

[3.4 训练模块与测试模块 23](#_Toc41163011)

[3.4.1 训练模块 23](#_Toc41163012)

[3.4.2 测试模块 24](#_Toc41163013)

[3.5 实验结果以及分析 24](#_Toc41163014)

[3.5.1 基础五组实验 25](#_Toc41163015)

[3.5.2 采样三组实验 33](#_Toc41163016)

[3.5.3 交叉验证三组实验 35](#_Toc41163017)

[3.5.4 实验总结 36](#_Toc41163018)

[第4章 结 论 38](#_Toc41163019)

[参考文献 40](#_Toc41163022)

[致 谢 41](#_Toc41163023)

[附 录 42](#_Toc41163024)

[附录A: 主要源程序 42](#_Toc41163025)

[附录B: 光盘 44](#_Toc41163026)

# 第1章 绪 论

本章节主要介绍此次研究的背景和意义，网络流量分类领域的研究现状（包括三种传统的流量分类技术，新兴的基于深度学习的流量分类技术和卷积神经网络性能差异的主要原因），本次研究做的主要工作以及论文的整体组织结构。

## 1.1 研究背景和意义

由于互联网的快速发展，互联网用户数量在庞大的基础上仍然不断增加，并且出现了各种互联网服务和互联网应用，随之而来的是网络流量数据的数量不断增加，流量形成的复杂性也在增加。为了确保互联网的可持续发展，需要不断更新互联网相关技术。

流量分类技术作为互联网相关技术之一，在解决互联网服务提供商（ISP）的网络问题中起着重要的作用，可以给网络管理者提供许多帮助。网络运营商也需要了解网络中每个类别流量类型的资源占用情况，以便合理的分配网络资源，保证互联网用户的良好冲浪体验。在网络安全方面流量分类可用于识别加密流量以及有关网络攻击的恶意流量。除此之外，互联网协议制造者和互联网应用设计者要根据互联网中不同流量类型的一些反馈对互联网协议和互联网应用做出及时的调整。

网络流量的组成通常是不平衡的。例如视频流量和文字聊天流量的数据包数目差异很大，正常流量的数据包数目也是远远超出恶意攻击流量的数据包数目。而且网络流量的组成是不断更替的，每隔一段时间就会涌现出一些新的流量，这些流量刚开始也是规模比较小。所以面对数据不平衡的网络流量，在网络流量分类过程中分辨出多数大类流量的同时兼顾少数小类流量也是非常重要的。

流量分类已经研究了二十年：从ISP的Qos设置和计费，到防火墙和入侵检测系统等。基于端口，数据包检测和经典的机器学习方法已被广泛使用，但是由于互联网的急剧变化，这些技术的准确率已经下降。伴随深度学习方法的出现，有学者用深度学习的方法做出了很多尝试，取得了很好的效果。

## 1.2 国内外研究现状

为了满足网络管理的需要，流量分类技术得到了显著发展。目前存在两种流量分类技术：第一种为传统网络流量分类技术，包括基于端口的流量分类技术，基于数据包的流量分类技术以及基于经典机器学习的分类技术，第二种为近年来兴起的基于深度学习的流量分类技术。

### **1.2.1** 传统网络流量分类技术

1. **基于端口的流量分类技术**

基于端口的流量分类技术是首先出现的。仅仅是在TCP或UDP数据包头中找到源端口号和目标端口号，然后将这些端口号与符合端口特点的互联网应用程序对应起来，这样就完成了网络流量分类，这种分类技术是最简单的。标准端口号是Internet分配号码授权机构（IANA）分配给常见应用程序的固定端口号。例如，21端口号对应于FTP服务，22端口号对应于SSH服务，80端口号对应于HTTP服务，443端口号对应于HTTPS服务。

因为在互联网发展前期互联网应用程序很少，所以特定的端口号对应于特定的应用程序，可以快速精准的从端口号库中找出哪个应用程序对应于哪种流量。但是随着互联网的发展，P2P应用程序开始使用动态端口号，某些服务允许用户自定义端口号。基于端口的流量分类技术的分类效果识别率仅仅为50%至70% [19]。现在几乎没有人使用这项技术了。

1. **基于数据包的流量分类技术**

基于数据包的流量分类技术是对网络流量数据的数据包载荷进行深度包检测，在应用层内容搜索协议特征串，例如：BitTorrent Protocol对应BitTorrent协议，GET对应HTTP协议，220对应FTP协议。这个方法与基于端口的流量分类技术类似，都是检测数据的某项特征和已知的规律匹配，从而分析出流量的类别。不同的是基于端口的流量分类技术依赖端口，而基于数据包的流量分类技术依赖于数据包载荷[9]。

此项技术起初对未被封装处理的流量和未加密的流量十分的可靠。但是现在互联网上的加密流量不断增加，许多应用程序使用协议封装或混淆来避免一些网络识别和拦截策略。除此之外，访问全部有效载荷在现在的互联网规模之下也是不可能的。一方面是时间和精力投入很大，成本很高，另一方面是用户的隐私得不到保证。因此，基于数据包的流量分类技术不再是网络流量分类的主流。

1. **基于机器学习的流量分类技术**

基于机器学习的流量分类技术一般是对经典的机器学习算法的改进或者与其他技术结合，用来适配特定场景中的流量数据，提升流量分类的准确率和效率等指标值。

例如J.Kornycky等人将矢量量化算法和决策树结合起来提高了F1得分[3]{Aceto, 2018 #28}。Zhang J等人尝试使用自动优化参数的半监督K-means聚类算法细粒度区分未知流量[15]。Z.Wang等人使用改进的K-SVD字典学习模型来学习特征[12]。Kong等人利用 SVM对多种攻击流量应用进行分类和识别[14]。

基于机器学习的流量分类技术有着它的许多优势：第一是选取合适的特征之后，流量其他的变化不会影响到分类的准确率，具有很强的适应能力；第二是现在有着很多优秀的机器学习算法，生态很好，面对特定的流量数据集都有不错的算法可以支撑，并且这些算法都有着不错的扩展性和灵活性；第三在数据量不大的情况下机器学习算法可以有着不错的效果，在有限数据集进行特征学习时有着自己的优势；第四是在计算过程中CPU参与之后就可以完成工作，GPU不是必须的。除此之外，基于机器学习的流量分类技术还有很多优秀的地方，因此它被许多学者广泛使用着。

但是基于机器学习的流量分类技术也有着它的不足的地方。例如特征工程很大程度上影响着机器学习算法分类的性能，这个过程往往依赖于行业的专家来做特征选取工作。人工的特征选取是有上限的，不能保证提取到所有的有效特征。而且面对如今的大数据时代，此项技术的效率也在慢慢的下降。除此之外，在机器学习算法解决问题的过程中，常常将问题分解为几个小问题，分别解决小问题之后再把几个最优方案组合起来形成完整的解决方案，但是部分最优方案的组合不一定面对整体时是最优的。

### **1.2.2** 基于深度学习的网络流量分类技术

近年来，深度学习在许多领域取得了不凡的成就，许多学者也开始将它应用于网络流量分类。

Wei Wang等人采取两种维度的CNN（一维和二维CNN）对预处理后的原始流量数据进行特征提取 [11]。作者通过观察实验评价指标中的准确率等，验证了这两种方法的优越性，相对于传统网络流量分类技术实现了大的超越。

M.Lopez-Martin等人将CNN和RNN结合起来，用于网络流量分类。网络输入数据的两个维度分别为作者选取的流统计特征（源端口号、目的端口号、包方向、负载字节数、TCP窗口大小、到达间隔等）和数据包序列号[5]。作者在研究组合网络时，保持CNN的输出中特征向量维度不变，其他两个维度平坦化，作为RNN需要的时间维度，以此构成的矩阵作为神经网络的输入，取得了优异的效果。作者还通过实验研究了网络结构、选取的特征、时间序列长度对分类性能的影响。

以上的研究证明了CNN应用于网络流量分类的可行性和有效性。

### **1.2.3** 卷积神经网络性能差异的主要原因

在阅读前人文献时，本文总结了影响基于卷积神经网络的流量分类性能差异的主要原因，并且计划了所做的工作。

1. **数据来源为原始数据包数据或者流统计特征**

W. Wang等人在在数据预处理阶段用一维向量来表示每个流或会话，只使用每个数据包中前784个字节[11]。作者使用了2个卷积层、2个池化层和2个全连接层来完成分类任务。实验数据集包含12个流量类别。在模型评估结果对比中，远远超越了使用时间序列和统计特征的C4.5方法。

S. Rezaei等人中使用基于一维CNN的半监督方法对Google应用程序分类[8]。他们使用采样时间序列特征代替前n个包，取得了不错的效果。这验证了使用采样时间序列特征代替前n个包的可能性。

本次研究选择使用了前n个包，因为它的处理比较简单，效果相比于流统计特征不相上下。

1. **网络维度、深度及卷积核大小的选择**

Wei Wang等人采取一维和二维CNN对相应预处理后的原始流量数据进行特征提取[11]。其中二维网络结构类似于手写数字识别的经典网络LeNet-5，—维网络的层次结构与之类似。

而陈晔欣设计出三维卷积神经网络，与低维卷积神经网络相比，准确率大大增加[17]。在数据预处理阶段提取每个数据包中的前l字节的数据，然后对每字节的数据分别进行m比特的one-hot编码，每个数据包转换为l\*m的二维数据。将n个数据包对应的二维数据按顺序组合为l\*m\*n的三维数据。

本次研究中使用二维卷积神经网络，因为原始数据预处理之后生成的灰度图片（大小为32×32），是恰好对应于二维卷积神经网络的。深度的话不超过10层，因为随着层数的增加，分类效果不会有大的增加，与之而来的还有梯度消失问题。经过后面的测试调整定为64个卷积核，大小为3×3。

1. **是否对样本不均衡问题作出处理**

Zhou H等人在研究基于离差标准化的卷积神经网络时，出现了样本不均衡的负面影响[16]。在Moore数据集中比例较少的muitimedia类别，interactive类别以及game类别的流量并未被重视处理，所以在平均准确率为90%的情况下，这三种类别的准确率分别为54.25%，0%，0%。Wei Wang等人的实验中，训练数据不平衡对实验性能影响很大[11]。例如VPN-VoIP流量有6000个训练样本，VPN-Email只有298个训练样本，其精确度分别为99.5％和80％。

而在其他前人的研究中，有的实验者考虑到了这一点并加以处理，减轻了样本不均衡问题对研究的影响。如Lotfollahi等人开发了一个称为深度包的框架[6]。他在数据预处理阶段使用欠采样方法，随机移除主要类的样本(具有更多样本的类)，直到类相对平衡，取得了很好的效果。陈晔欣在研究基于三维卷积神经网络时，使用了5折交叉验证的方法，避免了样本不均衡问题对实验的干扰[17]。

本次研究中也面临着这个问题，并且从采样的角度和改进的交叉验证角度来解决这个问题，取得了一定的效果。

1. **CNN和经典机器学习算法的结合**

V. Tong等人提出了一种基于卷积神经网络、基于NetFlow和基于数据包特征的分类方法[10]。提出的方法有两个主要的分类阶段。第一阶段使用随机森林和基于netflow的特征来检测谷歌环聊服务。第二阶段采用卷积神经网络，结合基于NetFlow的特征，基于数据包的特征和一些替代方案，将网络流分为视频流，Google播放音乐流或文件传输流。实验表明可以检测到五种基于QUIC的服务，准确度很高（约99％）。

因为时间的原因和自身的学习水平，本次研究没有涉及这一方面。

1. **CNN和其他深度神经网络的结合**

M.Lopez-Martin等人将CNN和RNN结合起来，用于网络流量分类任务。网络输入数据的两个维度分别为作者选取的流统计特征（源端口号、目的端口号、包方向、负载字节数、TCP窗口大小、到达间隔等）和数据包序列号[5]。他们在组合网络的研究中，保持CNN的输出中特征向量维度不变，其他两个维度平坦化，作为RNN需要的时间维度，这些数据共同构成的矩阵作为输入，提高了准确率。

这个方面在本次研究中也是没有涉及的，因为时间和精力等一些原因。

综上所述，本文分析了影响网络流量分类过程中卷积神经网络分类性能差异的主要原因，并且计划了要做的工作。本次研究中考虑到实际情况（时间，自身知识水平等条件），仅仅涉及前三部分的研究。在实验中本文选取原始数据包作为特征，调试好自己的网络架构（卷积神经网络维度，网络深度以及卷积核数目和大小等），然后尝试了采样方法和交叉验证方法来解决数据不平衡问题。

## 1.3 主要工作

本文描述了网络流量分类领域的现状，阐述了传统网络流量分类技术的局限性，列举了近年来一些学者利用卷积神经网络进行网络流量分类的工作，并且总结出了基于卷积神经网络的网络流量分类性能差异的主要原因。

根据自身水平和其他条件，本次研究中工作为：首先对原始流量数据进行一系列的预处理，使其能够符合卷积神经网络模型的输入标准。然后使用深度学习框架Tensorflow完成模型的搭建，调整参数和网络架构，直至对混合流量分类时可以顺利识别10种常规流量，10种加密流量以及2种恶意流量，并且综合准确率达到90%以上。最后面对数据不平衡问题，尝试采样方法以及交叉验证方法，进一步了提高准确率。

## 1.4 论文组织结构

第一章是绪论，主要有此研究的背景和意义，现今研究情况，研究的主要工作以及论文的组织结构。

第二章为理论基础，主要有实验数据集的介绍，主流深度学习框架的对比和选用，卷积神经网络的基本原理以及实验平台和评估指标。

第三章中是本次研究设计的基于卷积神经网络的流量分类系统，包含系统框架设计，卷积神经模型的设计，三大系统模块（预处理模块，训练模块和测试模块）以及实验结果与分析。

在第四章本文描述了对整个毕业设计进行的反思总结以及未来可以改进的方向。

# 第2章 实验基础

本章节主要介绍实验中所涉及的一些基础知识，包括数据集，深度学习框架，卷积神经网络以及实验平台和评价指标。

## 2.1 实验数据集

数据集一般可以由两个方面来获得。一是实验者自己捕获数据。如Zhou H等人使用Microsoft Network Monitor对校园网流量进行数据采集得到网络样本，打上标签使用[16]。二是使用网络上公开的数据集。如Lotfollahi等人在卷积神经网络和堆叠自动编码器神经网络结合的研究中使用了ISCXVPN2016数据集[6]。陈晔欣在研究基于三维卷积神经网络的网络流量分类中使用了USTC-TFC2016数据集[17]。

实验者自己捕获数据时可以根据自身需要捕获到适合自己的数据，但是通常还要进行去噪，规范化等等一系列的后续处理，需要一定的时间和精力。使用网络上公开的数据集大都没有了噪音，可以直接简单的处理之后用于自己的研究中，不过可能和自己的实验不太匹配。

此次研究中考虑到研究程度没有前辈们那么深入，并不需要特别的数据集，公开数据集已经可以满足此次研究的需要，所以此次研究采用的数据集为公开数据集。本次实验用到两个公开数据集：ISCXVPN2016数据集和USTC-TFC2016数据集。

### **2.1.1** ISCXVPN2016数据集

本次研究中使用的第一个数据集为ISCXVPN2016数据集。此数据集通过使用Wireshark等工具抓取流量生成数据。

整个数据集大小为27.5G，整个数据集有Chat,VPN-Chat,P2P,VPN-P2P等14个类别。此数据集用来支持研究中的模型训练以及样本不均衡问题的解决方案的测试。

在本次实验中本文取其中的10个类别进行研究,各个类别包含非VPN和VPN两个类别，如ICQ类别流量包含VPN-ICQ类别流量和NOVPN-ICQ类别流量。每个类别最多采用5000个数据样本。数据集的详细信息见表2.1。

可以看到aim流量类别的非VPN数据样本数目是4099个，aim流量类别的VPN数据样本数目为2455个，icq流量类别的非VPN数据样本数目是3476个。这些都是小类流量，整个数据集中除了这3种流量，都是可以采样至5000个的。在本文的后面两部分实验，会从采样的角度和交叉验证的角度来解决这个问题，即样本不平衡问题。

表2.1 ISCXVPN2016数据集

|  |  |  |
| --- | --- | --- |
| 名称 | 非VPN数据样本数目 | VPN数据样本数目 |
| aim | 4099 | 2455 |
| facebook | >5000 | >5000 |
| email | >5000 | >5000 |
| netflix | >5000 | >5000 |
| hangouts | >5000 | >5000 |
| icq | 3476 | >5000 |
| youtube | >5000 | >5000 |
| skype | >5000 | >5000 |
| vimeo | >5000 | >5000 |
| spotify | >5000 | >5000 |

### **2.1.2** USTC-TFC2016数据集

此数据集是Wei Wang等人研究恶意软件流量分类时创建的[13]。一部分是十种网络攻击流量，另一部分是使用专业网络流量仿真工具IXIABPS工具仿真的十种常规流量。整个数据集的大小为3.7GB。在本次研究中，抽取部分数据完成常规流量和恶意流量的分类。

在本次实验中取整个数据集中的8个正常流量类别和2个恶意流量类别（zeus，virut）进行研究，每个类别最多采用5000个数据样本。数据集的详细信息见表2.2。

表2.2 USTC-TFC2016数据集

| 名称 | 数据样本数目 |
| --- | --- |
| smb | >5000 |
| mysql | >5000 |
| ftp | >5000 |
| zeus | >5000 |
| virut | >5000 |
| weibo | >5000 |
| worldofwarcraft | >5000 |
| facetime | >5000 |
| bittorrent | >5000 |
| skype | >5000 |

### **2.1.3** 数据流的结构

续表（2.2）

只有充分了解数据集中数据流的构成之后，才可以对它进行一些预处理操作，处理为下一步卷积神经网络分类时所需要的数据。数据集数据为pcap和pcapng两种格式。

如图2-1，pcap文件分为pcap文件头（24B），pcap数据包头（16B）+pcap数据包，pcap数据包头（16B）+pcap数据包，pcap数据包头（16B）+pcap数据包...



图2-1 pacap文件格式图

1. **pcap文件头解析**

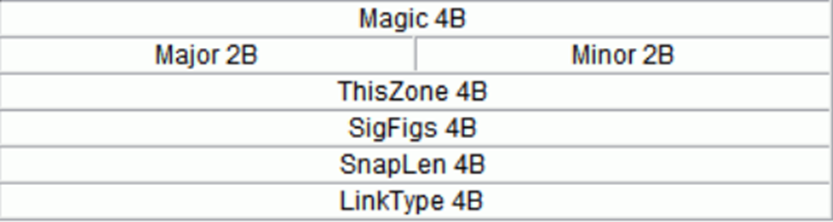


图2-2 pcap文件头信息

如图2-2，Magic：4B，含有标识文件的信息以及字节的顺序信息。Major：2B，文件主要版本号。Minor：2B，文件次要版本号。ThisZone：4B，时区。SigFigs：4B，时间戳精度。SnapLen：4B，最大存储长度。LinkType：4B，链路类型。

1. **pcap数据包头解析**

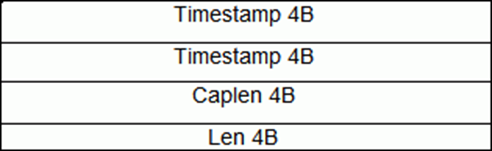


图2-3 pcap数据包头信息

如图2-3，Timestamp：4B，时间戳高位。Timestamp：4B，时间戳低位。Caplen：4B，当前数据区长度。Len：4B，离线数据长度。

1. **pcap数据包**

数据包就是常见的数据帧。在实验中不涉及它的结构，所以不再多做介绍。

pcapng是pcap的进阶，拥有更多的优势：支持从多个接口捕获；改进的时间戳分辨率；将注释直接嵌入到捕获文件中；存储在捕获文件中的其他元数据；可扩展格式等等。除此之外，其他都是和pcap文件是一样的。

## 2.2 深度学习框架

深度学习框架对深度学习的发展来说意义重大。从前，主流的深度学习框架都来自国外机构或公司研发，从最开始蒙特利尔大学与伯克利大学推出的 Theano、Caffe 框架，到现在谷歌维护的 TensorFlow、Facebook 推出的 PyTorch，深度学习产业在前人的积累上得以快速发展。而现在国内的浪潮已经掀起，2016 年，百度发布深度学习开源框架 PaddlePaddle（飞桨）。2020 年3月有三个深度学习开源框架发布，华为开源了自主研发的 MindSpore，旷视开源深度学习框架 MegEngine（天元），清华大学发布基于元算子和动态编译的深度学习框架 Jittor（计图）。

每个深度学习框架都有自己的特点：Theano擅长处理多维数组，通常用来和其它深度学习库结合起来进行数据处理。它可以支持规模很大的神经网络算法的运行，理解容易，但是由于大型模型的编译时间较长，扩展性不高和不支持多个GPU的并行运算等原因，逐渐被淘汰。

Caffe是一个卷积神经网络框架，把Matlab执行快速卷积网络的方式带到C和C++，同时带有Python接口。Caffe在AI界非常知名，项目在GitHub上有高达30.1k的star数。Caffe在诞生之后被许多大型科技公司采用。Caffe适用于微调现有网络，训练模型不用自己写代码。但是对循环网络不友好，对于庞大的网络（GoogLeNet和ResNet）相当的笨重，没有商业支持，发展缓慢，适用性也比较差。

Facebook于2017年1月开放了Python版本的Torch，称为Pytorch。它有着很多优点：许多容易组合的模块化的组件，很多预先训练的模型等。但是也有一些缺点：通常需要编写自己的训练代码（即插即用），没有商业支持，参差不齐的文档（开发文档质量太差）。

2015年11月谷歌（Google）出品，基于Python和C++编写。GitHub上最热，谷歌搜索最多，使用人数最多（本人也是其中之一），大多数网上招聘工作描述中也提到了它。2019年3月已发布最新的TensorFlow2.0 版本。TensorFlow拥有大量的开发者，有详细的说明文档，可查询资料多，自带tensorboard可视化工具，能够让用户实时监控观察训练过程，支持多GPU、分布式训练，跨平台运行能力强。但是它也有一些不足的地方，例如速度较慢，内存占用较大，已预定性的模型不多，不易工具化。

综合考虑，鉴于tensorflow强大的生态，完善的功能，良好的性能和活跃的社区等，本次实验采用的深度学习框架为tensorflow。

## 2.3 卷积神经网络

卷积神经网络通过卷积操作和池化操作学习输入特征的局部模式。随着网络层数的增加，卷积神经网络对这些局部模式不断地进行组合、抽象，最终学习到高级特征。

最早的卷积神经网络是由LeCun等人在1989年提出的[4]。随着深度学习的蓬勃发展，后面有了更加优秀的多层叠加网络。例如数字识别网络LeNet-5[2]。在这之后，AlexNet，GoogLeNet，VGG，ResNet纷纷涌现出来。

### **2.3.1** 主要结构

虽然卷积神经网络不断在发展，但是卷积神经网络的构成元件都是差不多的，都是由几个重要的神经网络层组合而成，不同的卷积神经网络可能有着深度，卷积核大小，滑动步长等等差异。图2-4为数字识别网络结构图。

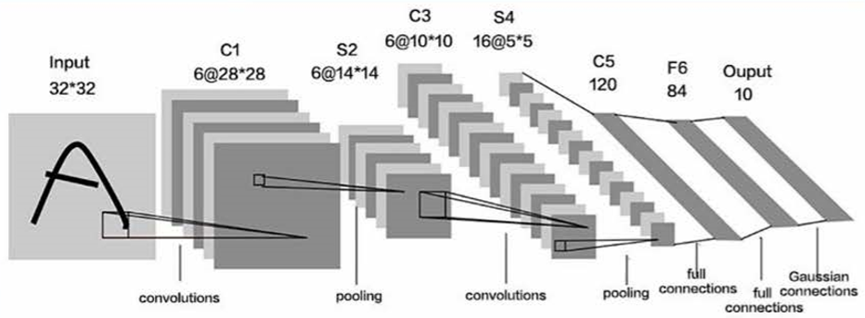


图2-4 数字识别网络结构图

如图2-4，可以看出卷积神经网络由输入层，卷积层，池化层，全连接层四种类型的层次组合而成的。

输入层：输入层作为整个神经网络的输入。在卷积神经网络的图像处理过程中，看作图片像素矩阵。三维矩阵的长宽和深度，分别表示图像的大小和色彩通道。一般图像有两种：一种是彩色图片，通道数为3，另一种是灰度图片，通道数为1。此次研究中用的是灰度图片。

卷积层：卷积层中每个神经元的输入只是上的神经网络的一部分（局部连接），通过卷积核卷积，然后加上激活函数的转化，作为下层输入的数据。卷积操作后会形成更加抽象和突出的特征。通常来说卷积层处理的节点矩阵深度会增加，由一个平面转化为几个平面。

池化层：池化层的作用是提取重要有效的特征，通过池化操作可以缩小全连接层中神经元的个数，使得神经网络中参数大大减少。神经网络对于矩阵的深度没有影响，但是矩阵的大小会有一定的缩小。

全连接层:在经过几个卷积层和几个池化层的处理，在卷积神经网络后面的几层一般使用全连接层，用来铺平得出分类结果。最后一层的全连接网络层负责分类。常用的分类函数为softmax (多分类)或sigmoid(二分类)。

### **2.3.2** 核心操作（卷积操作和池化操作）

1. **卷积操作**

在卷积神经网络卷积过程中，每个卷积核依次与输入矩阵中的一个子块进行逐点相乘并求和，如图2-5卷积操作第一步卷积图。

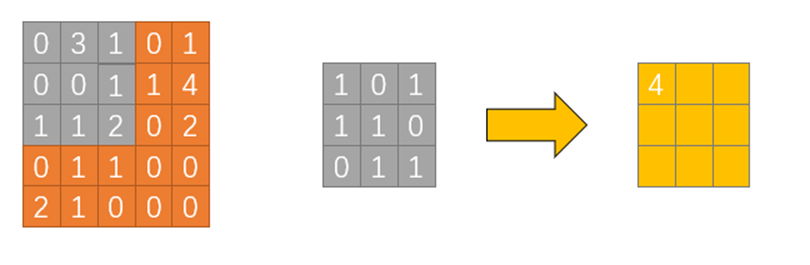
****

图2-5 卷积操作第一步卷积图

如图2-5所示，第一步先对左上角区域进行点积，得出的结果依次写入一个新的矩阵，新的矩阵的大小是由卷积的步长决定的，即输入矩阵子块的移动规则，有X轴上的步长以及Y轴上的步长，在图中卷积的X轴和Y轴的步长都为1，所以结果矩阵的大小为3×3。结果如图2-6卷积操作最后一步卷积图。

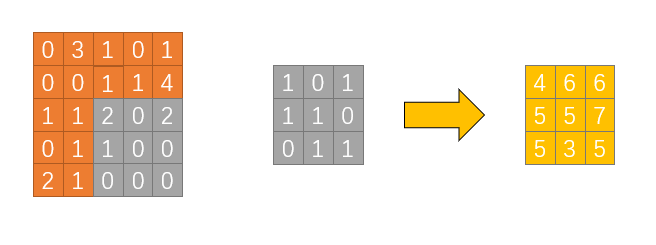


图2-6 卷积操作最后一步卷积图

如图2-6，卷积操作完成后，形成一个新的矩阵，然后经过激活函数处理作为下一层的输入数据。

在卷积操作中，有一些决定性的参数：第一个是卷积核的尺寸，即感受野的大小，通常指卷积核的长和宽，其深度通常默认与输入特征的深度相同。第二个是卷积核步长：即卷积核在长度方向和宽度方向上每次移动的距离，步长和卷积核大小共同控制着卷积后输出特征的尺寸。第三个是卷积核数量：卷积核的数量对应卷积核输出特征的深度，每个卷积核的输出为一个通道，多个卷积核的输出进行堆叠，形成一个特征立方体。

1. **池化操作**

在卷积神经网络池化过程中，上层矩阵经过池化操作，提炼出重要特征，形成新的矩阵输入到下一层，减少了神经网络中的不必要的参数。目前有三种池化方法。第一种是mean-pooling（平均池化）。第二种是max-pooling（最大池化）。第三种是minimal-pooling（最小池化）。如图2-7为平均池化图，如图2-8为最大池化图，如图2-9为最小池化图。

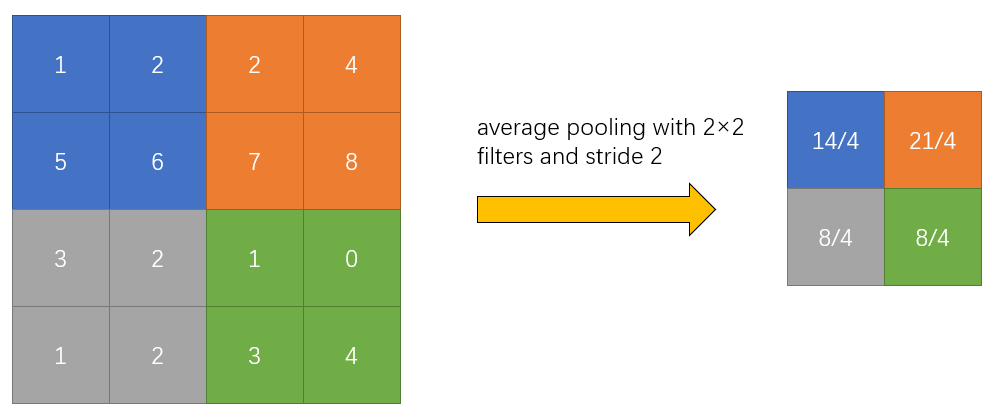


图2-7 平均池化图

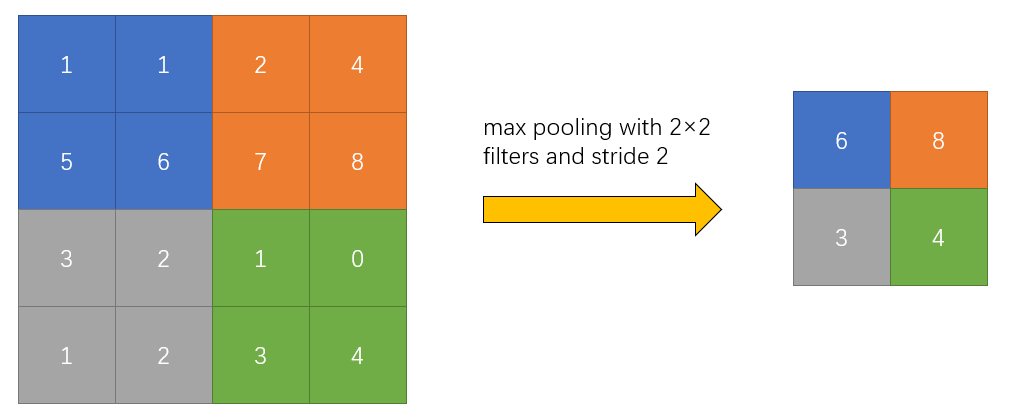


图2-8 最大池化图

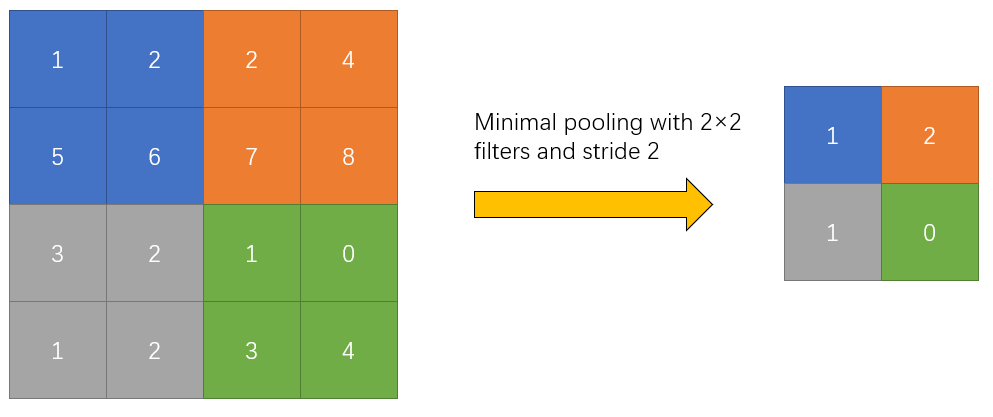


图2-9 最小池化图

如图2-7所示，平均池化是对输入矩阵的一个子块所有特征点求平均值，结果逐个填入新矩阵，步长为2，生成2×2的新矩阵。如图2-8所示，最大池化是对输入矩阵的一个子块所有特征点取最大值，结果逐个填入新矩阵，步长为2，生成2×2的新矩阵。如图2-9所示，最小池化是对输入矩阵的一个子块所有特征点选取一个最小值，结果逐个填入新矩阵，步长为2，生成2×2的新矩阵。

对于最小池化，池化过程往往会丢失明显的特征，例如一个子块的最大值为8，最小值为0，最后选取0填入新的矩阵，那么特征点为8的这个很突出的特征就丢失了，对整个模型的学习造成了很大的负面影响，所以现在常用的卷积神经网络都弃用了这个方法。

对于平均池化，池化过程中往往关注于中间值。例如在人脸识别的过程中，经过平均池化的图片背景信息往往更加突出，而不是人脸部分，对于特征纹理是不敏感的，这个方法在图像识别分类领域应用的也是不多的。

最大池化是目前为止学者们采用最多的一种池化策略，它注重于事物的突出特征，对图像边缘和纹理特征十分的敏感。在学习重要特征的同时，去掉冗余的信息，降低网络的复杂度，防止了过度拟合。

综上所述，鉴于最大池化策略的优秀，本次研究中采用了最大池化策略。

### **2.3.3** 优点和不足

卷积神经网络作为一个优秀的深度学习模型，了解它的优点和不足之后，然后才能更好的使用它。

1. **卷积神经网络的优点**

卷积神经网络有着局部连接，权值共享等优势。

（1）局部连接

在传统的神经网络结构中，神经元之间的连接是全连接的。即n-1层的神经元与n层的全部神经元都进行连接。但是在卷积神经网络中，n-1层与n 层的部分神经元连接。

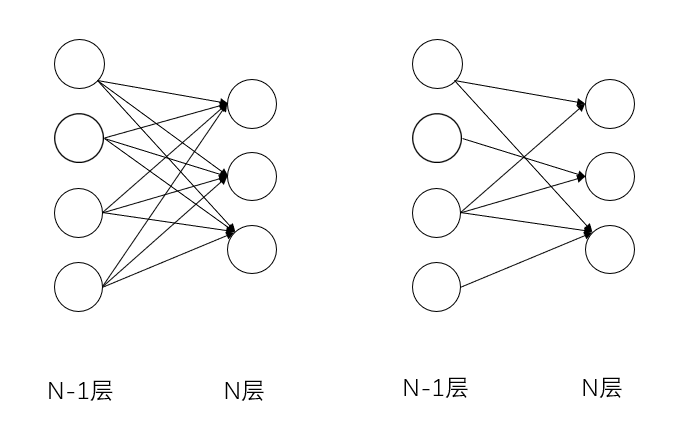


图2-10 全连接与局部连接对比图

图2-10显示了全连接和局部连接不同的地方，左半部分为全连接示意图，由图可以看出前一层与后一层神经元之间都有着边的连接，每条边都有着相应的参数，所以全连接方式的参数是很多的。右半部分是局部连接示意图，由图中可以看出局部连接存在着很少的边，参数数目降低了很多。对比左右部分的图可以明显发现连接数减少，相应的参数也会随之减少。

（2）权值共享

在卷积神经网络中，卷积层中的卷积核类似于滑动窗口，在整个输入矩阵中按照特定的步长进行滑动，包括X轴方向以及Y轴方向。卷积得到输入矩阵的特征矩阵，这个特征矩阵就是卷积层提取出来的局部特征。在整个卷积神经网络的训练过程中，包含权值参数的卷积核也会更新，直至训练结束。

权值共享就是整个输入矩阵在使用同一个卷积核内的参数。假如在一次卷积过程中，卷积核使用了9次，卷积核大小为4×4×1，那么非权值共享下的参数就有144个，而权值共享下的参数只有16个。因此权值共享策略大大减少了卷积核中参数数目。

1. **卷积神经网络的不足**

（1）卷积神经网络的优势在于特征的高效提取，但分类部分的全连接层含有大部分参数量，对训练效率有较大影响。

（2）卷积神经网络不记忆物体的位置和方向，它通过查看整个图像，然后检查图像中是否存在某些组件来进行预测（没有组件特征之间的关系），如果组件特征提取的不明显，准确率会大大下降。

（3）为了避免局部最优，卷积神经网络对初始参数和超参数的确定需要很多工作，因此，卷积神经网络的一个缺点是它们需要根据当前问题进行初始化，成本很高，也需要该领域的一些知识。

（4）学习成本很高，涉及到很多复杂的数学知识，对于小白还是要有一定的时间才能掌握使用，作者刚开始学的时候也是觉得很难理解。

综上所述，卷积神经网络是十分强大的，虽然有些许不足，但是作为网络流量分类的模型是足以胜任的。

## 2.4 实验环境以及评价标准

1. **实验环境**

表2.3为本次实验硬件平台表。

表2.3 实验硬件平台表

|  |  |
| --- | --- |
| 操作系统 | Window10 |
| 处理器 | AMD A10-8700P |
| 内存 | 8192MB RAM |
| 显卡 | AMD Radeon R6 Graphics |
| 硬盘 | 500G三星固态 |

表2-3 硬件环境表

表2-3为实验所使用的硬件环境，软件坏境为Anaconda3，python 3.7， Tensorflow 2.1.0，numpy 1.18.1，imageio 2.6.1，opencv3.4.5.20等。

1. **评价标准**

基于所有数据集的累计真正（TP），假正（FP），真负（TN）和假负（FN），以准确率，精准度，召回率和F1得分作为评价指标。如表2.4为数据集结果统计规则表。

表2.4 数据集结果统计规则表

|  |  |  |
| --- | --- | --- |
|  | 分类结果 | 真实标签 |
| TP | 正例 | 正例 |
| FP | 正例 | 负例 |
| TN | 负例 | 负例 |
| FN | 负例 | 正例 |

最终评价指标为准确率（Accuracy，ACC），精准度（Precision），召回率（Recall）以及F1得分（F1-score）。这些指标的详细信息如公式2-1，公式2-2，公式2-3，公式2-4所述。

（2-1）

（2-2）

（2-3）

（2-4）

# 第3章 基于卷积神经网络的流量分类系统

此章节介绍本文设计的基于卷积神经网络的流量分类系统。包括设计的系统框架和卷积神经网络模型，三大系统模块（预处理模块，训练模块和测试模块）的构成以及最终实验结果的分析与总结。

## 3.1 系统框架

如图3-1为此次研究中基于卷积神经网络的流量分类系统框架。

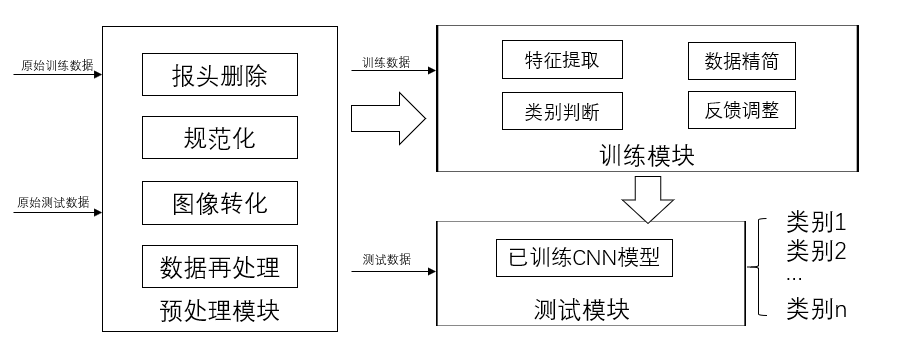


图3-1 流量分类系统框架

如图3-1中所示，流量数据输入到预处理模块中，经过报头删除，规范化，图像转化以及数据再处理四个步骤输出为卷积神经网络可以直接使用的数据。对于训练数据，将它输入到训练模块中，经过特征提取，数据精简，类别判断以及反馈调整训练出卷积神经网络模型。对于测试数据，将它输入到含有训练完成的卷积神经模型的测试模块，根据分类结果，完成对系统的评估。

## 3.2 卷积神经网络模型

如图3-2为本次研究中使用的卷积神经网络模型，虽然本次研究中涉及二分类实验，十分类实验以及二十分类实验，但是这些分类实验使用的模型除了最后一层不同（二分类最后一层为两个神经元，十分类为十个神经元，二十分类为二十个神经元），其他都是一致的，图3-2展示的为10分类的卷积神经网络模型。

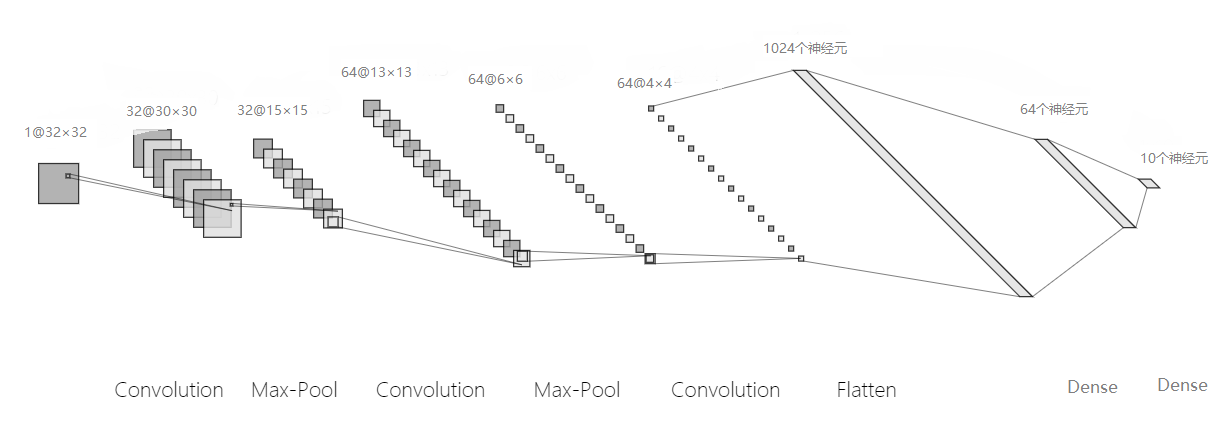


图3-2 卷积神经网络模型

如图3-2所示，整个模型共有八个层次。依次是卷积层，池化层，卷积层，池化层，卷积层，展平层，全连接层，全连接层。

第一个卷积层输入数据为32×32×1，代表一个预处理模块处理好的灰度图片。共有32个卷积核，大小为3×3，x轴和y轴步长都为 1，非填充模式，结果使用ReLU激活函数进行转化，生成32个30×30的平面。

第一个池化层接收上层数据，为最大池化层，池化层大小为2×2，x轴和y轴步长都为2，生成32个15×15的矩阵。

第二个卷积层接收上层数据，共64个卷积核，大小为 3×3， x轴和y轴步长都为1，结果进行ReLU函数激活，生成64个13×13的矩阵。

第二个池化层接收上层数据，为最大池化层，池化层的大小为2×2，x轴和y轴的步长都为2，生成64个6×6的矩阵。

第三个卷积层接收上层数据，共64个卷积核，大小为 3×3， x轴和y轴步长都为1，结果进行ReLU函数激活，生成64个4×4的矩阵。

展平层接受上层数据，将64个矩阵展平降维，输出为1024个神经元。

第一个全连接层接受展平层的输出数据，处理为64个神经元。

末尾的层是全连接层，使用10个神经元完成10分类。

## 3.3 预处理模块

通常情况下原始的流量数据不能直接用于卷积神经网络的输入，所以要进行数据预处理。在此模块，原始数据包数据经过预处理输出为卷积神经网络标准输入数据。预处理阶段有四个主要步骤，包括报头删除，规范化，图像转化和数据再处理。

1. **报头删除**

数据链路报头包含物理链路信息，这对于在网络中转发帧是必不可少的，但对于应用程序标识或流量特征任务来说却不是有效信息。因此，在预处理阶段，首先移除以太网报头。即在读取流量数据时，只提取pcap数据包的信息。在一条pcap数据中，除去pcap文件头的24B信息和pcap数据包头的16B信息。

1. **规范化**

卷积神经网络需要相同的输入长度，而数据集中的数据包长度则从50到1392字节不等。因此，数据集将在零填充步骤中添加一些零值，以使每个数据包的长度相似。在本次研究中，取数据包的前1024B，所有对于少于1024B的进行零填充，多于1024B的只取前1024B。

1. **图像转化**

此方法借鉴刘金来等人在数据预处理阶段的方法，将数据包中一字节八位的二进制数据转化成为0到255的数值，正好对应灰度图片中像素点的数值**错误!未找到引用源。**。然后按照先后顺序排列为一维序列数据，再将一维序列数据利用numpy工具包转化为大小为32×32的二维数组。最后利用imageio工具包的imwrite函数将二维数组转化为灰度图片并保存。

1. **数据再处理**

此次研究共涉及3组实验部分，每一组实验部分根据不同的需要做出针对的数据再处理工作。第一组部分为基础五组实验，第二部分为采样实验，第三部分为交叉验证实验。

（1）基础五组实验部分

在基础的五组实验中（VPN和非VPN二分类实验，VPN十分类实验，非VPN十分类实验，VPN和非VPN二十分类实验，常规流量和恶意流量十分类实验），训练集，验证集以及测试集比例为8:1:1，选取样本至5000个，此组实验目的是调试好卷积神经网络模型，初步得出10种流量类型，10种加密流量类型以及2种恶意流量的分类结果，F1得分在90%以上，其他不多做处理。

（2）采样三组实验部分

在采样三组实验中，为了研究原始样本，使用欠采样方法处理的样本，使用过采样方法处理的样本以及使用欠采样和过采样结合的方法处理的样本对样本不均衡问题的影响，做出以下处理：

第一组为初始对照实验，不做处理。

第二组欠采样实验中，以非VPN10分类为例，为了达到数据平衡，仅对大的类别进行欠采样，小类不处理。因为由表2.1可以知道icq的类别的样本最少，为3476个，所以都是每类采取3476个样本。

第三组过采样实验中，方法是仅对小类进行过采样，因为数据集中youtobe（视频流量）和netflix（视频流量）等数据量充足的流量类别数据包非常的多，如果仅对小类进行过采样，使他们和大类样本数目达到一样的数目，那么所需时间是非常大的，而且其实没有必要使用大类的所有数据包，仅仅使用一部分就足够模型的训练学习了。故此种方法从理论上弃用了。

第四组欠采样和过采样结合实验中，这种方法是对大类进行欠采样，保证模型的充分学习的同时精简大类样本，然后对小类过采样，与大类的样本数目保持一致，使得小类流量可以充分学习，每类采取5000个样本。

按照这样的设定，此实验部分共有三组实验：对照实验，欠采样实验以及欠采样和过采样结合实验。

（3）交叉验证实验部分

在交叉验证实验中，为了解决数据不平衡问题，本文从使小类别的流量数据充分利用的角度来尝试着解决这个问题。例如在本次实验中，数据集被分为三部分。训练集，验证集，测试集的比例大约为8:1:1。原本验证集的作用是确定人工调参的方向，避免模型的过拟合问题，确保模型的可泛化性。它是不参加模型的训练的，那么这部分数据在十分紧张的小类流量面前是很可惜的。所以想办法把它利用起来，但是又不能影响模型的可泛化性，也要避免过拟合。

常用的交叉验证方法是把数据集划分为有序的K等份，共有K次独立不相干实验，准确率取K次实验的平均值。对于某K次实验，第K份数据为测试集，其余部分为训练集。如图3-3为常用的5折交叉验证方法图。

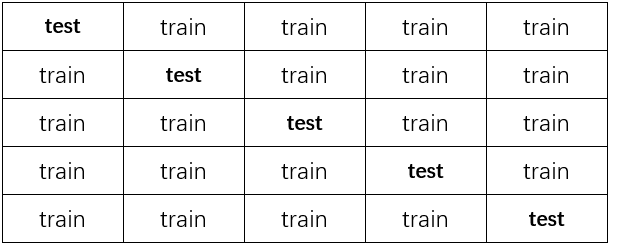


图3-3 常用的5折交叉验证图

这种方法的目的一般有两个。一个是评估模型，取K次评价指标值的平均。另一个是根据平均误差选出K个模型中误差最小的模型作为最终模型。交叉验证可以在较少的数据集中学习到更多信息，这个是和作者的需求相符合的，对于小类流量会十分的友好。

本次研究中，借鉴交叉验证的思想，针对本文的研究加以改进。对于整个数据集，首先选出十分之一保留为测试集，接着把其余的数据集分为五个部分，训练过程中共有五次大的迭代，第一次大的迭代第一部分为验证集，其余部分为训练集，第二次大的迭代继承第一次大的迭代的模型，把第二部分确定为验证集，其余部分为训练集…第五次迭代继承第四次迭代的模型，确定第五个部分为验证集，其余部分为训练集。如图3-4为修改后的五折交叉验证方法图。

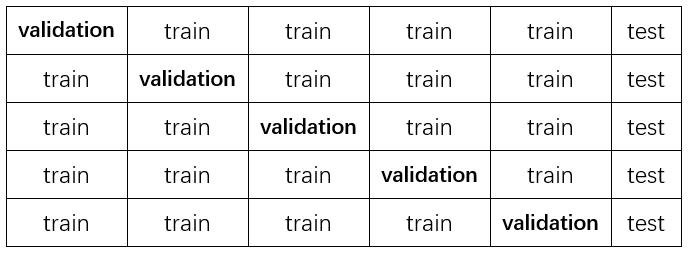


图3-4 改进五折交叉验证方法图

这样就可以对小类数据充分的利用，而且对于模型首先划出的测试集也是未知的，提高准确率的同时也保证模型的可泛化性，避免过拟合问题。

交叉实验有三组实验，第一组为初始不做处理的对照实验。第二组为测试集保证对于模型是未知的，训练集和验证集五折交叉的实验，第三组训练集和验证集十折交叉，其他与第二组保持一致的实验。

## 3.4 训练模块与测试模块

本小节为系统模块的训练模块和测试模块。训练模块的工作为输入处理之后的数据，使得模型不断的训练，充分学习样本的特征。测试模块对已完成的模型做出评估。

### **3.4.1** 训练模块

训练模块主要完成的工作分为四部分。特征提取，数据精简，类别判断以及反馈调整。

特征提取的工作由卷积神经网络的卷积层负责，将输入矩阵的各个局部分别按照特定的步长等规则，与含有参数的小矩阵（卷积核）进行卷积，卷积的结果交给激励函数进行转化，激励函数的结果即为该层的最终输出。

数据精简工作中，采用权值共享，稀疏连接，池化等策略将数据和特征精简，仅关心那些重要有效的数据特征。

类别判断工作中，最后通过全连接层的铺平分类，最后由输出层（最后一层全连接层）根据已有知识判断类别。

反馈调整工作中，网络会对比神经网络最终的类别预测结果和训练数据的真实标签，按照一定规则来做出逆向的反馈，对网络中的参数进行调整，使得分类性能更棒。

### **3.4.2** 测试模块

测试模块的作用是对训练好的模型进行评估。测试模块使用训练模块最终确定的各个参数，在未知的测试集上进行测试。相比于训练模块，测试模块中数据是一个正向的传播，没有误差反馈和网络调整，所以各个参数也不再发生变化，每个数据样本只利用一次。

## 3.5 实验结果以及分析

本小节呈现了此次研究的最终实验结果，然后对结果进行分析总结。第一部分实验为基础五组实验，目的是调整卷积神经网络的模型架构以及其他实验设置，使得系统能够正常分类出10种常规流量，10种VPN加密流量以及2种恶意流量。并且使这五组实验的分类准确率达到90%以上。

在第一部分实验结束后在10种常规流量中分类实验中遇到了数据不平衡问题，也是本文要解决的问题。例如在此次实验中aim流量样本数为4099个，icq流量样本数为3476个，而其他8种流量样本十分充足，采样到了5000个。在网络训练过程中，少数类别样本个数较少，模型对这些类别特征没有充分学习，所以会导致误判，结果就是本来是少数类别的流量会判给其他类别，总体分类的准确率，精度，召回率等都会有一定程度的降低，在接下来的两部分实验中，本文就是要解决这个问题，从数据不平衡的角度提高分类的各个指标。

第二部分为采样三组实验，在第一部分实验确定的模型架构等基础之上，针对数据不平衡问题，从采样的角度来在数据预处理模块做出改变，测试3种采样方案对准确率的影响。

第三部分为交叉验证三组实验，面对数据不平衡问题，从数据充分利用的角度，借鉴交叉验证的思想，测试修改的五折交叉方法和十折交叉方法对准确率的影响。

### **3.5.1** 基础五组实验

本次实验使用ISCXVPN2016公开数据集，基于卷积神经网络对网络流量分类，验证卷积神经网络在流量分类的可行性，调整模型参数，提高准确率。训练集，验证集以及测试集比例为8:1:1。训练集用来训练模型更新参数。验证集来观测分类效果，手动调整模型。最后的测试集对模型进行评估测试。

1. **VPN与非VPN二分类实验**

本次实验数据为表2.1中10种常规流量和对应的10种VPN加密流量，每种流量采取1000个样本，共20000个样本。实验进行20次迭代。此实验的目的是使得模型可以预测出两种类型的流量：常规流量和VPN加密流量。

图3-3为每次迭代的训练准确率以及验证准确率变化图，表3.1为20次迭代之后测试集在训练完成的模型上测试的结果表，包括准确率，精度，召回率以及F1得分。

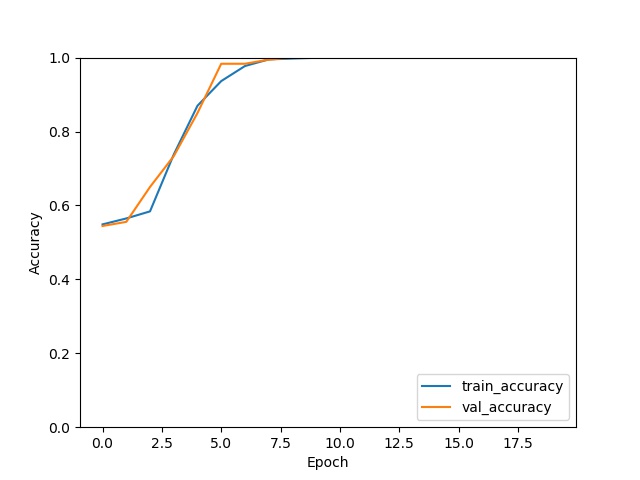
****

图3-3 二分类实验每次迭代的训练准确率以及验证准确率变化图

表3.1 二分类实验测试集测试结果表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| category | precision | recall | f1-score | accuracy | support |
| vpn\_flow | 100% | 100% | 100% |  | 100 |
| flow | 100% | 100% | 100% |  | 100 |
| avg | 100% | 100% | 100% | 100% | 200 |
| weighted avg | 100% | 100% | 100% |  | 200 |

由图3-3可知在第10次迭代后分类准确率趋于平稳。训练准确率和验证准确率最终达到100%。由表3.1可知在对于模型未知的测试集中，测试样本支持个数都为100个，两种流量类型的精度，召回率，F1得分都是100%，综合准确率为100%。实验的结果非常的棒。

1. **VPN10分类实验**

此次实验数据为表2.1中10种加密流量，每种流量最多采取5000个样本。实验进行50次迭代。此实验的目的是使得模型可以预测出10种加密类型的流量。

图3-4为VPN10分类每次迭代的训练准确率以及验证准确率变化图，表3.2为50次迭代之后测试集在训练完成的模型上测试的结果表，包括准确率，精度，召回率以及F1得分。

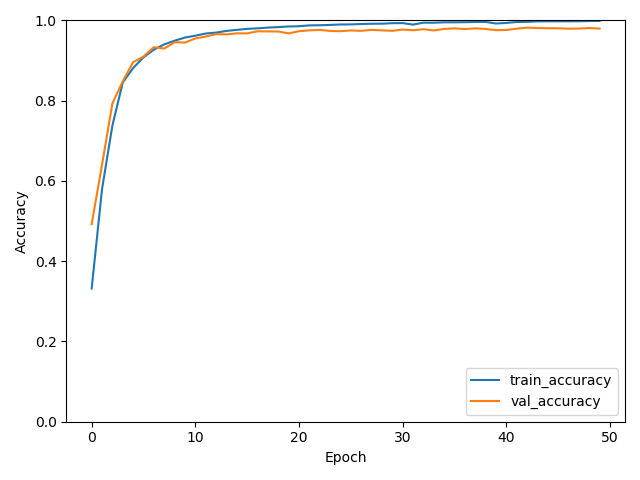
****

图3-4 VPN10分类实验每次迭代的训练准确率以及验证准确率变化图

表3.2 VPN10分类测试集测试结果表

| category | precision | recall | f1-score | accuracy | support |
| --- | --- | --- | --- | --- | --- |
| vpn\_hangouts | 96% | 98% | 97% |  | 500 |
| vpn\_vimeo | 99% | 100% | 100% |  | 500 |
| vpn\_youtube | 98% | 99% | 99% |  | 500 |
| vpn\_icq | 97% | 98% | 97% |  | 500 |
| vpn\_netflix | 98% | 97% | 97% |  | 500 |
| vpn\_email | 98% | 98% | 98% |  | 500 |
| vpn\_aim | 98% | 91% | 94% |  | 275 |
| vpn\_skype | 99% | 98% | 98% |  | 500 |
| vpn\_spotify | 97% | 96% | 97% |  | 500 |
| avg | 98% | 97% | 98% | 98% | 200 |
| weighted avg | 98% | 98% | 98% |  | 200 |

由图3-4可知在第40次迭代后分类准确率不再发生大的变化，训练准确率在40次迭代后稳定在99%，验证准确率在40次迭代之后稳定在98%。由表3.2可知综合准确率在98%，F1平均得分为98%，加权得分为98%。实验的结果也是非常不错的。

1. **非VPN10分类实验**

此次实验数据为表2.1中10种常规非加密流量，每种流量最多采取5000个样本。实验进行50次迭代。此实验的目的是使得模型可以预测出10种常规非加密流量。

图3-5为非VPN10分类卷积神经网络每次迭代的训练准确率以及验证准确率变化图，表3.3为50次迭代之后测试集在训练完成的模型上测试的结果表，包括准确率，精度，召回率以及F1得分。

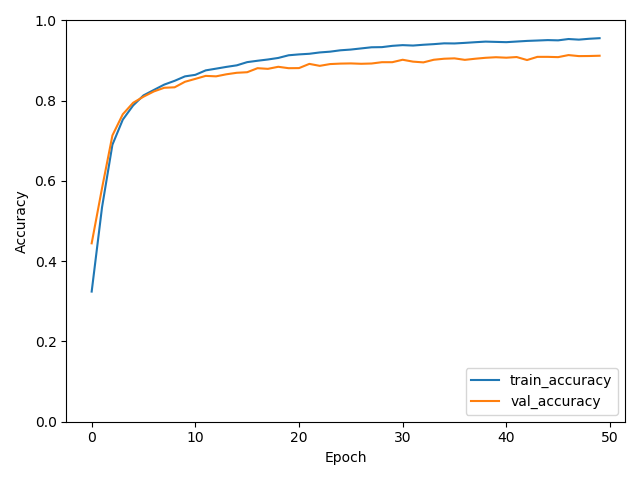
****

图3-5 非VPN10分类实验卷积神经网络每次迭代的训练准确率以及验证准确率变化图

表3.3 非VPN10分类测试结果表

| category | precision | recall | f1-score | accuracy | support |
| --- | --- | --- | --- | --- | --- |
| aim | 74% | 85% | 79% |  | 409 |
| vimeo | 96% | 91% | 93% |  | 500 |
| spotify | 88% | 93% | 90% |  | 500 |
| icq | 80% | 65% | 71% |  | 347 |
| hangouts | 96% | 97% | 96% |  | 500 |
| email | 99% | 99% | 99% |  | 500 |
| facebook | 98% | 95% | 97% |  | 500 |
| skype | 97% | 99% | 98% |  | 500 |
| netflix | 94% | 96% | 95% |  | 500 |
| youtube | 96% | 95% | 96% |  | 500 |
| avg | 92% | 91% | 91% | 92% | 4756 |
| weighted avg | 92% | 92% | 92% |  | 4756 |

由图3-5可知在第45次迭代后分类准确率趋于平稳，训练准确率在45次迭代后稳定在95%，验证准确率在45次迭代之后稳定在91%。模型有着轻微的过拟合现象，即训练准确率曲线和验证准确率曲线Y轴坐标越来越大。

续表（3.3）

由表3.3可知在对于模型未知的测试集中，aim流量测试样本支持个数为275，icq流量测试样本为347个，其他流量测试样本都为500个。分类综合准确率在92%，相比于二分类实验的100%以及VPN10分类实验的98%，综合准确率虽然在90%以上，但是还是不够高的，F1得分等指标值也是不如前两次实验的，那么就来分析一下原因。

根据实验结果表小类流量aim和icq的F1得分分别为79%和71%，远远不如其他流量（都在90%以上）。在表2.1 中（ISCXVPN2016数据集实验数据表）可以了解到这两种流量和其他流量的差异在于样本的数量，aim为4099个样本，icq为3476个样本，其他流量样本为5000个。那么这正是本文在国内外研究现状中提出的数据不平衡问题，即在网络训练过程中少数类别样本数目不够，模型对这些类别特征没有充分学习，所以会导致误判，结果就是本来是少数类别的流量会判给其他类别，分类的准确率，精度，召回率等都会有一定程度的降低。在第二部分实验以及第三部分实验本文将从采样角度和数据充分利用的角度来解决这个问题。

1. **VPN与非VPN二十分类实验**

此次实验数据为表2.1中20种流量（10种常见的流量以及对应的10种加密流量），每种流量最多采取5000个样本。实验进行50次迭代。此实验的目的是使得模型可以预测出10种常规非加密流量以及10种VPN加密流量。

图3-6为混合流量二十分类卷积神经网络每次迭代的训练准确率以及验证准确率变化图，表3.4和表3.5为50次迭代之后测试集在训练完成的模型上测试的结果表，包括准确率，精度，召回率以及F1得分。

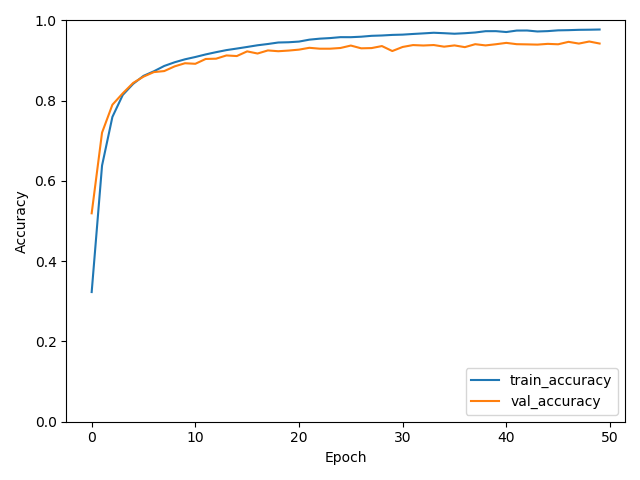
****

图3-6 混合流量20分类实验卷积神经网络每次迭代的训练准确率以及验证准确率变化图

表3.4 混合流量20分类测试集测试第一部分结果表

| category | precision | recall | f1-score | accuracy | support |
| --- | --- | --- | --- | --- | --- |
| aim | 77% | 70% | 73% |  | 409 |
| youtube | 95% | 98% | 96% |  | 500 |
| email | 98% | 99% | 99% |  | 500 |
| vimeo | 92% | 92% | 92% |  | 500 |
| vpn\_netflix | 94% | 97% | 96% |  | 500 |
| skype | 97% | 97% | 97% |  | 500 |
| vpn\_hangouts | 96% | 96% | 96% |  | 500 |
| netflix | 97% | 95% | 96% |  | 500 |

表3.5 混合流量20分类测试集测试第二部分结果表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| category | precision | recall | f1-score | accuracy | support |
| vpn\_icq | 97% | 98% | 97% |  | 500 |
| facebook | 98% | 95% | 96% |  | 500 |
| vpn\_facebook | 96% | 95% | 95% |  | 500 |
| vpn\_aim | 97% | 93% | 95% |  | 245 |
| hangouts | 97% | 97% | 97% |  | 500 |
| vpn\_youtube | 99% | 98% | 98% |  | 500 |
| vpn\_spotify | 96% | 93% | 94% |  | 500 |
| spotify | 92% | 90% | 91% |  | 500 |
| vpn\_vimeo | 98% | 99% | 99% |  | 500 |
| vpn\_skype | 99% | 98% | 98% |  | 500 |
| icq | 69% | 79% | 74% |  | 347 |
| vpn\_email | 95% | 97% | 96% |  | 500 |
| avg | 94% | 94% | 94% | 94% | 9501 |
| weighted avg | 94% | 94% | 94% |  | 9501 |

由图3-6可知训练准确率在45次迭代后稳定在97%，验证准确率在40次迭代之后稳定在94%。模型有着非常轻微的过拟合现象，即训练准确率曲线和验证准确率曲线Y轴坐标微微有些差异。

由表3.2可知在对于模型未知的测试集中，aim流量测试样本支持个数为275，icq流量测试样本为347个，vpn\_aim流量测试样本支持个数为245，其他流量测试样本都为500个。分类综合准确率在94%，不及二分类实验（100%）以及VPN10分类实验（98%），但是超过了非VPN10实验（92%）。

根据实验结果表小类流量aim和icq的F1得分分别为73%和74%，远远不如其他流量（都在90%以上）。在非VPN10分类实验中发现了这个问题，本文分析了原因，在这个实验中再次发现了这个问题，即数据不平衡问题，小类分类的效果并不可观。在第二部分实验以及第三部分实验本文将从采样角度和数据充分利用的角度来解决这个问题。

1. **USTC-TFC2016数据集十分类实验**

此次实验数据为表2.2中10种流量（8种常见的流量以及2种恶意流量），每种流量最多采取5000个样本。实验进行20次迭代。此实验的目的是使得模型可以预测出8种常规流量以及2种恶意流量。

图3-7为USTC-TFC2016数据集十分类每次迭代的训练准确率以及验证准确率变化图。

表3.5为20次迭代之后测试集在训练完成的模型上测试的结果表，包括准确率，精度，召回率以及F1得分。

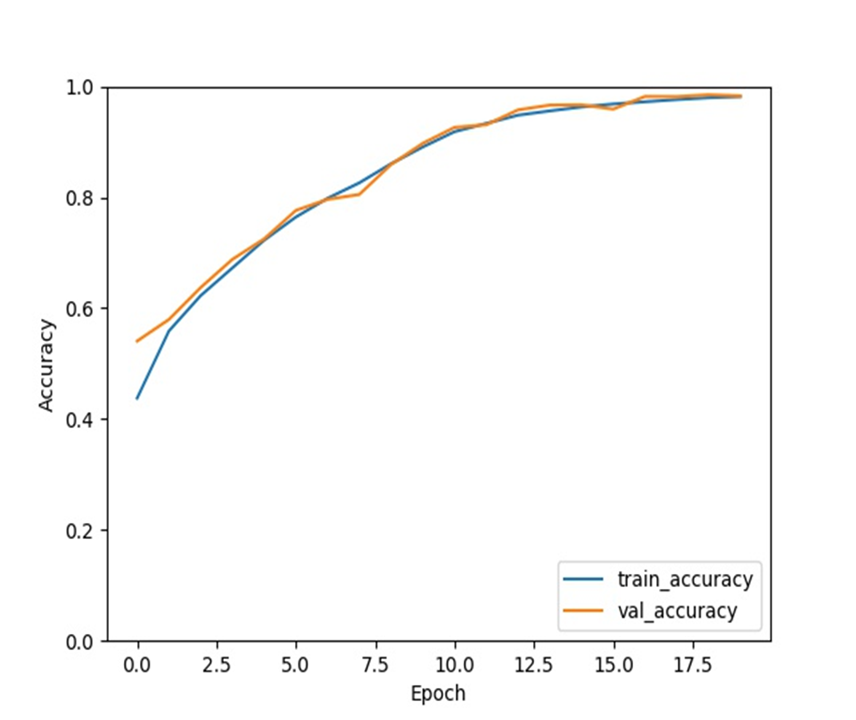


图3-7 十分类每次迭代的训练准确率以及验证准确率变化图

表3.5 测试结果表

| category | precision | recall | f1-score | accuracy | support |
| --- | --- | --- | --- | --- | --- |
| weibo | 95% | 100% | 98% |  | 500 |
| worldofwarcraft | 96% | 95% | 96% |  | 500 |
| skype | 95% | 99% | 97% |  | 500 |
| facetime | 100% | 100% | 100% |  | 500 |
| virut | 99% | 98% | 99% |  | 500 |
| smb | 100% | 92% | 96% |  | 500 |
| ftp | 96% | 96% | 96% |  | 500 |
| mysql | 97% | 95% | 96% |  | 500 |
| zeus | 98% | 99% | 99% |  | 500 |
| bittorrent | 100% | 100% | 100% |  | 500 |
| avg | 98% | 98% | 98% | 98% | 5000 |
| weighted avg | 98% | 98% | 98% |  | 5000 |

由图3-7可知训练准确率在18次迭代后稳定在98%，验证准确率在18次迭代之后稳定在98%。由表3.5可知virut，zeus两种恶意流量F1得分都为99%，综合准确率为98%，F1得分为98%，此次实验完成了对混合流量识别出常规流量和恶意流量的工作，结果是很不错的。

### **3.5.2** 采样三组实验

本次实验为了研究不平衡数据的不同处理方法对准确率的影响，先在ISCXVPN2016数据集选取10个非VPN流量类别，从样本采样的角度来进行实验，并对3种采样方法对比：仅过采样，仅欠采样，过采样和欠采样结合。

3种方法目的是一样的，即使得各个流量类别的训练样本数目保持一致。仅过采样方法是样本数目都处理为样本最多的流量类别的数目。仅欠采样方法是样本数目都处理为样本最少的流量类别的数目。过采样和欠采样结合方法是在样本最多的流量类别的数目和样本最少的流量类别的数目之间探索出一个合适的样本数目。

对于第一种仅过采样，在第一部分实验中50000份数据就可以达到98%的准确率，所以实际上没有必要采取那么多的样本，一定量的样本是可以满足模型的训练学习的，而且对于视频流量（Netflix，Youtobe等）的样本数目可能是几百万，对于小类就要复制几百次，复制次数到达一定程度时也是对于模型分类的提高没有多大的效果的。在此次研究中，作者的硬件设施也是不支持的，所以综合分析，这个方法被排除了。

1. **原始对照实验**

对于第一种实验不做任何处理，作为对照实验，与基础五组实验的非VPN10分类实验保持一致。

1. **仅欠采样**

在仅欠采样实验中，保证每个类别的训练集和验证集样本数目一致，每类样本数目采取至最小类别流量的样本数目，最小类别流量为3476个。其他条件与原始对照实验保持一致。

1. **过采样和欠采样结合**

在过采样和欠采样结合实验中，每个类别的训练集和验证集样本数目也是一样的，按照训练集，验证集，测试集的比例为8:1:1的原则。那么每个类别的训练集和验证集共有4500个样本，对大类流量采取样本至4500个，对小类流量（aim，icq）采用随机复制扩充至4500个样本。测试集不做处理，保证数据的原始分布。其他条件与原始对照实验保持一致。

在实验结果分析时，抽取前三组每个实验的aim类别流量F1得分，icq类别流量F1得分,综合F1得分，综合准确率四个指标进行比较。如图3-8为前三组采样对比实验结果图。

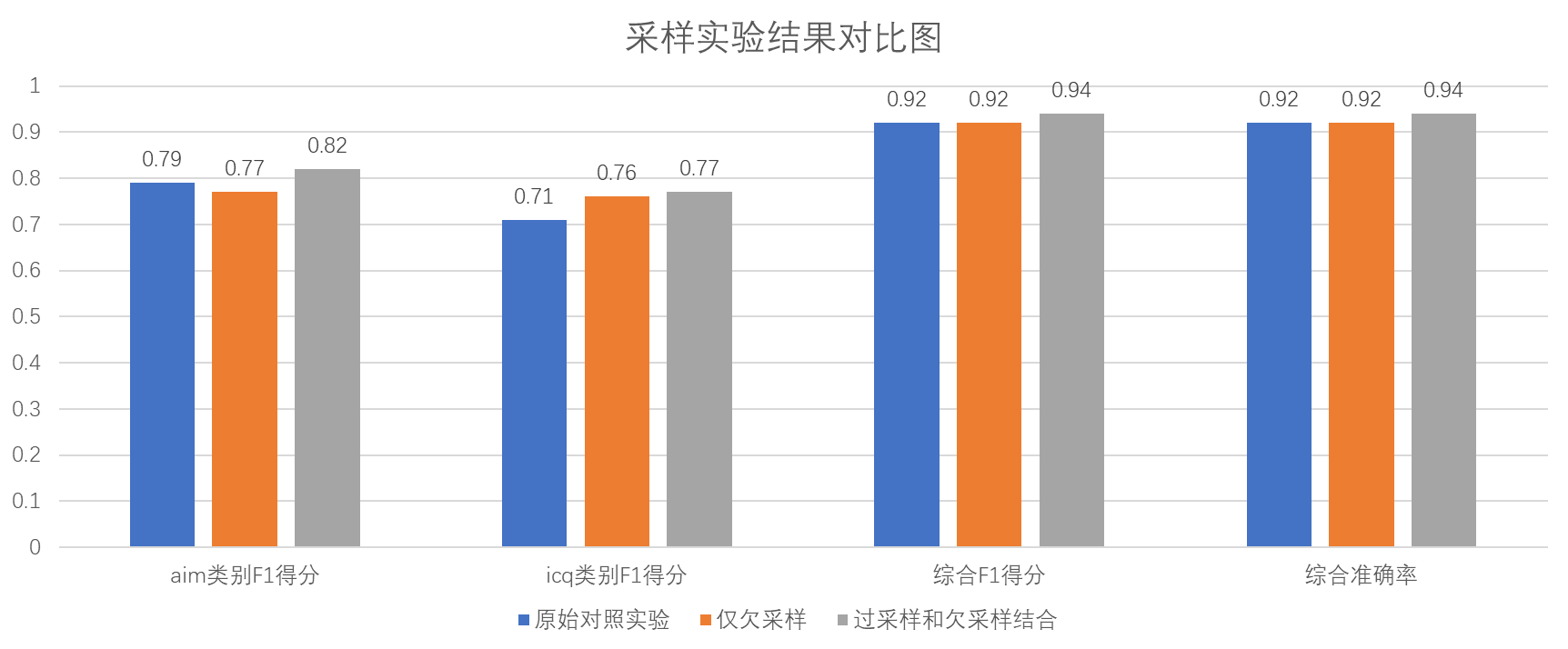


图3-8 前三组采样对比实验结果图

如图3-8所示，仅欠采样实验与原始对照实验相比相差并不大，虽然保证了样本数目一致，但是各项指标没有明显的增加，原因是仅欠采样的样本数目减少，影响了特征的充分学习。过采样和欠采样结合的实验各项指标都有所增加，综合F1得分和综合准确率增加了两个百分点。本次的实验确定了最优的采样方法，即过采样和欠采样结合的方法。

### **3.5.3** 交叉验证三组实验

本次验证交叉验证方法借鉴经典交叉验证思想，但是针对本次实验又加以改变，使得分类系统可以充分利用数据集，减少数据不平衡时对准确率的影响，提高准确率的同时避免过拟合。数据集选取10个非VPN流量类别，根据对照实验处理为三组，第一组为训练集，验证集和测试集大致为8:1:1的固定的数据集，第二组实验数据集中首先拿出十分之一的数据集保留为测试集，然后其余部分为训练集和验证集，训练集和验证集遵循五折交叉验证的原则，测试集仍然保证对于模型是未知的。第三组实验数据集中同第二组相比训练集和验证集遵循十折交叉验证的原则，其他与第二组保持一致。

1. **原始对照实验**

对于第一个实验不做任何处理，作为对照实验，与基础五组实验的非VPN10分类实验保持一致。

1. **五折交叉实验**

五折交叉实验的改变在数据输入上，首先拿出十分之一的数据作为测试集，测试集不参与模型的训练，保证对于模型它是未知的。然后对剩下的五分之四（训练集和验证集）做交叉验证，每轮10次迭代，继承上一轮的模型，共50次迭代。与对照实验的50次迭代一致，其他条件和第一个对照实验保持一致。

1. **十折交叉实验**

对于十折交叉实验，首先拿出十分之一的数据保留为测试集，然后对剩下的十分之九（训练集和验证集）做交叉验证，每轮5次迭代，继承上一轮的模型，共50次迭代，总的与对照实验一致，其他条件和第一个对照实验保持一致。

在实验结果分析时，抽取三组每个实验的aim类别流量F1得分，icq类别流量F1得分，综合F1得分，综合准确率四个指标进行比较。如图3-9为三组对比交叉实验结果图。

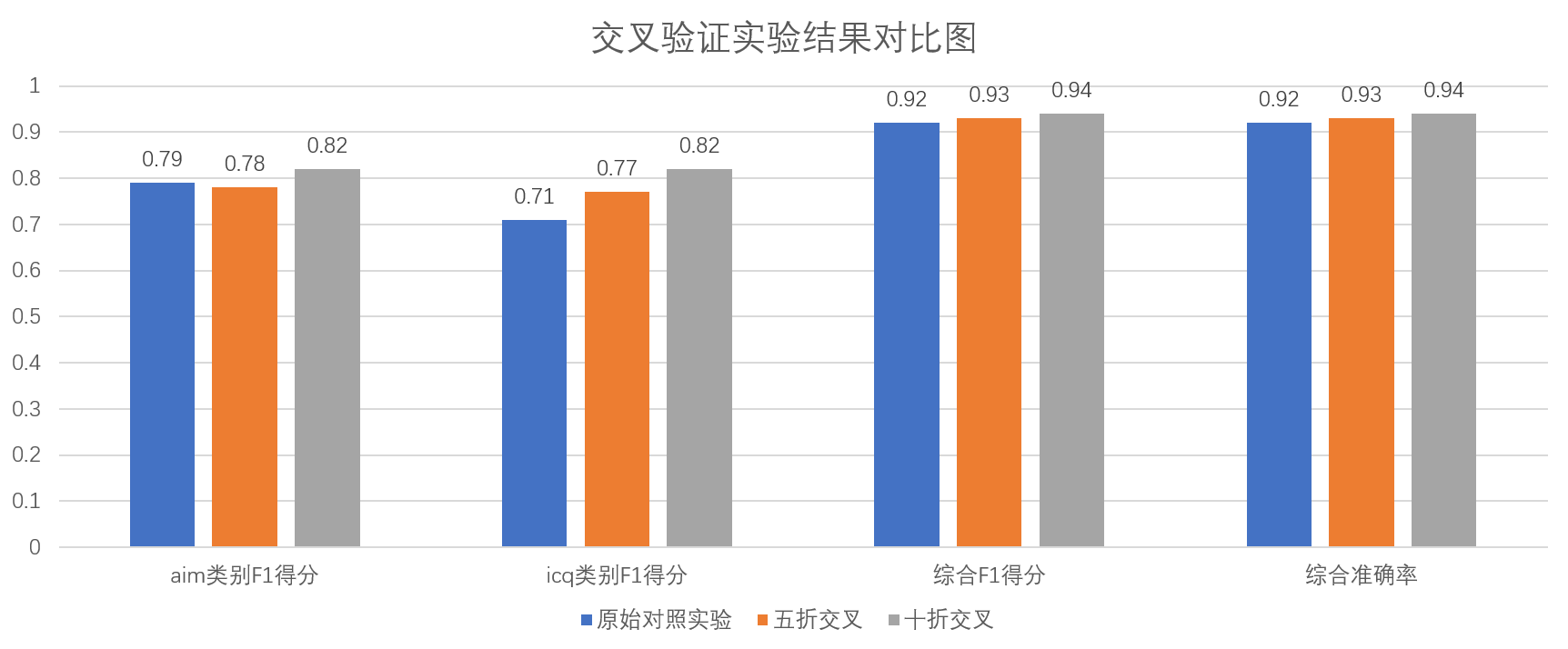


图3-9 前三组对比实验结果图

如图3-9所示，十折交叉实验各个指标值都超过了五折交叉实验的指标值，十折交叉实验相比于原始对照实验，aim类别流量F1得分提升了3个百分点，icq类别F1得分提升了11个百分点，综合得分提升了2个百分点，综合准确率提升了2个百分点。实验证明了提出的改进交叉验证的有效性。

### **3.5.4** 实验总结

在本次研究中，首先做了基础的分类实验，模型可以分出10种常规流量和对应的10种VPN加密流量，还有8种常规流量和2种恶意流量，并且分类准确率可以达到90%以上。然后从采样角度解决数据不均衡问题，通过几个方法的比较，最终的过采样和欠采样结合方法比未经采样处理原始实验的F1得分和准确率都提升了2个百分点。最后从数据充分利用的角度尝试了五折交叉实验和十折交叉实验，最优的十折交叉方法相对于原始对照实验的F1得分和准确率也是都提升了2个百分点，显示了此方法的可靠性。

实际上在实验中还有一个有趣的地方，就是在仅欠采样的实验中，每个类别的样本数目都为3476个，但是小类流量（aim，icq）的F1得分在80%左右，其他大类流量F1得分都在90%以上。在其他实验中，小类F1得分在各种处理下都有所提升，但是同样还是和大类有着显著差异。

为什么样本个数一样的同时小类分类效果依然和大类有这么大的差别呢？数据充分利用之后还是不能完全弥补它们的差距呢？

后来本文提出了一个可能的原因，可能因为小类别流量不仅仅是数据包数目少，每个数据包里面的数据内容也是很少的，在数据预处理模块本文选取特征的手段是原始数据包数据，那么从一开始小类别流量和大类别流量在样本数目和样本内容上都有着显著的差异，仅仅弥补样本数目是不够的。

回过头来观察小类图片和大类图片，发现确实小类的图片大部分都是黑色居多，大类图片大部分都是白色居多。这个现象是因为在数据预处理阶段取数据包的前1024个字节，不够的进行零填充，小类的数据包内容很少，大都进行了零填充，0在灰度图片意味着黑色。于是这就验证了本文的猜想，使用原始数据包数据作为特征时，数据不平衡的问题要分解为两个问题，一个是样本数目，一个是样本内容。但是样本内容是一开始就确定了的，无法改变。使用原始数据包数据作为特征面对数据不平衡问题时，有着无法弥补的先天缺陷。

本文所做的采样和交叉验证实验从样本个数的角度出发，但是样本内容是无法改变的，要想改变样本内容，就要更换选取特征的手段，不再使用原始数据包数据作为特征，而是那些对于大类和小类没有显著差异的特征，这样新的特征没有容量上的差异，再加上从样本个数角度的处理，相信更进一步的解决数据不平衡问题。不过由于时间原因，此次研究中没有再进行探索。

# 第4章 结 论

在本章节，第一部分为总结本次研究中所做的工作。第二部分针对数据不平衡问题，提出未来可以继续前进的方向。

## 4.1 工作总结

文章中描述了网络流量分类领域的现状，阐述了传统网络流量分类技术的局限性，列举了近年来一些学者利用卷积神经网络进行网络流量分类的工作，总结出了基于卷积神经网络的网络流量分类性能差异的主要原因。

在研究中本文选取原始流量数据的数据包作为数据来源，进行一系列的数据预处理，保证数据能够输入到卷积神经网络模型。然后使用深度学习框架Tensorflow完成模型的搭建，调整超参数和网络架构，直至模型对混合流量分类时可以分成10种常规流量以及对应的10种加密流量，还有2种恶意流量，并且准确率达到90%以上。最后面对数据不平衡问题，尝试了采样方法以及改进的交叉验证方法，F1得分和准确率都在原来的基础上提升了2个百分点。

## 4.2 未来展望

在面对数据不平衡问题时，本文尝试了采样方法和改进的交叉验证方法，取得了一定的效果，但是小类流量的准确率和F1得分还是和大类有着差距，经过思考和查阅资料，提出了几个可以尝试的方向。

选取时间序列等流特征代替数据包数据作为数据来源。在此次研究中，通过实验发现小类流量的差异和大类流量的差异在两个方面，一个是数据包的内容，一个是数据包的数目。如果选取数据来源时使用大类流量和小类流量没有差异的流特征，那么就避开了小类流量和大类流量的样本内容的差异，只剩下了样本的数目差异。这个方法的关键在于选取哪些小类流量和大类流量差异不大的流特征。

对于小类流量使用更加严格的惩罚因子。面对小类流量误判犯错时，给予它相比于大类流量更加严格的惩罚，加快小类流量的学习进程[7]。这个需要对神经网络有着深入的了解。

对于小类流量加入人工构造数据。例如SMOTE(Synthetic Minority Over-sampling Technique)[1]。SMOTE是过采样算法，它可以人工生成数据集中从未出现过新的小类样本。具体就是使用K近邻在少数类样本之间插值生成新的样本。这个方法难点在需要一些工作找寻合适的K值。

加入机器学习算法。在这个方面前人做出了尝试，例如V. Tong等人提出了一种基于卷积神经网络、基于NetFlow和基于数据包特征的分类方法[10]。第一阶段使用随机森林检测两种流量。第二阶段将第一阶段剩下的流量采用卷积神经网络等手段分为其他三种流量。最后的实验结果表明可以分类出五种基于QUIC的流量，准确度很高（约99％）。

不过这种办法还是很耗时耗力的。在实践中会遇到几个比较难的问题，第一是首先要找到对小类别流量数据友好的机器学习算法，这中间要尝试很多机器学习算法，要做很多工作，第二是找到合适的算法之后，面对数据集的迁移变化，可能这个合适的算法就会变得不再适用，接着又回到了第一个问题，所以还是挺耗费时间和精力的，但是一个不错的尝试方向。

# 参考文献

1. Han H, Wang W, Mao B, et al. Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning[C]. International Conference on Intelligent Computing, 2005: 878-887.
2. Lecun Y L , Bottou L , Bengio Y , et al. Gradient-Based Learning Applied to Document Recognition[J]. Proceedings of the IEEE, 1998, 86(11):2278-2324.
3. Kornycky J, Abdul-Hameed O, Kondoz A, et al. Radio frequency traffic classification over WLAN[J]. IEEE/ACM Transactions on Networking, 2016, 25(1): 56-68.
4. Lecun Y, Boser B E, Denker J S, et al. Handwritten Digit Recognition with a Back-Propagation Network[C]. neural information processing systems, 1989: 396-404.
5. Lopezmartin M, Carro B, Sanchezesguevillas A, et al. Network Traffic Classifier With Convolutional and Recurrent Neural Networks for Internet of Things[J]. IEEE Access, 2017, 5: 18042-18050.
6. Lotfollahi M, Siavoshani M J, Zade R S H, et al. Deep Packet: A Novel Approach For Encrypted Traffic Classification Using Deep Learning[C]. soft computing, 2020: 1999-2012.
7. Pulgar F J, Rivera A J, Charte F, et al. On the Impact of Imbalanced Data in Convolutional Neural Networks Performance[C]. hybrid artificial intelligence systems, 2017: 220-232.
8. Rezaei S, Liu X. How to achieve high classification accuracy with just a few labels: A semi-supervised approach using sampled packets[J]. arXiv preprint arXiv:1812.09761, 2018.
9. Rezaei S, Liu X. Deep Learning for Encrypted Traffic Classification: An Overview[J]. IEEE Communications Magazine, 2019, 57(5): 76-81.
10. Tong V, Tran H A, Souihi S, et al. A novel QUIC traffic classifier based on convolutional neural networks[C]. 2018 IEEE Global Communications Conference (GLOBECOM), 2018: 1-6.
11. Wang W, Zhu M, Wang J, et al. End-to-end encrypted traffic classification with one-dimensional convolution neural networks[C]. intelligence and security informatics, 2017: 43-48.
12. Wang Z, Dong Y, Mao S, et al. Internet multimedia traffic classification from QoS perspective using semi-supervised dictionary learning models[J]. China Communications, 2017, 14(10): 202-218.
13. Wei W, Ming Z, Xuewen Z, et al. Malware traffic classification using convolutional neural network for representation learning[J], 2017: 712-717.
14. Kong L, Huang G, Wu K, et al. Identification of Abnormal Network Traffic Using Support Vector Machine[C]. parallel and distributed computing: applications and technologies, 2017: 288-292.
15. Zhang J, Chen X, Xiang Y, et al. Robust network traffic classification[J]. IEEE/ACM transactions on networking, 2014, 23(4): 1257-1270.
16. Zhou H, Wang Y, Lei X, et al. A Method of Improved CNN Traffic Classification[J], 2017: 177-181.
17. 陈晔欣. 基于深度学习的网络流量分类技术研究[D]. 北京邮电大学, 2019.
18. 刘金来. 深度学习模型在网络流量分类中的应用研究[D].哈尔滨理工大学,2018.
19. 柏骏, 夏靖波, 吴吉祥, et al. 实时网络流量分类研究综述[J]. 计算机科学, 2013(09):14-21.

# 

# **附 录**

## 附录A: 主要源程序

//流量分类

datapath = "E:\\DLResult\\AllTo20\\TransformImage\\train"

testpath = "E:\\DLResult\\AllTo20\\TransformImage\\test"

modelpath = "E:\\DLResult\\AllTo20\\20200519Result"

pltpath = "E:\\DLResult\\AllTo20\\20200519Result"

labeldict = {}

def create\_model():

model = tf.keras.models.Sequential([

keras.layers.Conv2D(32, (3, 3), activation='relu', input\_shape=(32, 32, 1)),

keras.layers.MaxPooling2D((2, 2)),

keras.layers.Conv2D(64, (3, 3), activation='relu'),

keras.layers.MaxPooling2D((2, 2)),

keras.layers.Conv2D(64, (3, 3), activation='relu'),

keras.layers.Flatten(),

keras.layers.Dense(64, activation='relu'),

keras.layers.Dense(20)

])

model.compile(optimizer=tf.keras.optimizers.Adam(learning\_rate=0.001),

loss=tf.keras.losses.CategoricalCrossentropy(from\_logits=True),

metrics=['accuracy'])

return model

train\_images, train\_labels = deal\_image(datapath)

test\_images, test\_labels = deal\_image(testpath)

# print(train\_images,train\_labels)

# print(test\_images,test\_labels)

model = create\_model()

model.summary()

history = model.fit(train\_images, train\_labels, batch\_size=320, verbose=2, epochs=50,

validation\_split=0.1)

plt.plot(history.history['accuracy'], label='train\_accuracy')

plt.plot(history.history['val\_accuracy'], label='val\_accuracy')

plt.xlabel('Epoch')

plt.ylabel('Accuracy')

plt.ylim([0, 1])

plt.legend(loc='lower right')

plt.savefig(pltpath+"\\"+"result.jpg")

plt.show()

test\_loss, test\_acc = model.evaluate(test\_images, test\_labels, verbose=0)

print(test\_loss, test\_acc)

Y\_test = np.argmax(test\_labels, axis=1)

y\_pred = model.predict\_classes(test\_images)

print(classification\_report(Y\_test, y\_pred))

model.save(modelpath+"\\"+"my\_model.h5")