廣東工業大學
GUANGDONG UNIVERSITY OF TECHNOLOGY

硕士学位论文

（专业学位）

基于深度学习的脱机手写汉字识别的研究与应用

作　　者　　姓　　名：袁柱

导　　师　　姓　　名：徐海水

学科（专业）或领域名称：计算机技术

论　文　答　辩　年　月：二〇二〇年六月

分类号：　　学校代码：11845

UDC：　　密级：　　学　　号：2111705101

广东工业大学硕士学位论文

（工程硕士）

基于深度学习的脱机手写汉字识别的研究与应用

袁柱

导师姓名（职称）：徐海水 副教授

侯仰峰 高级工程师

学科(专业)或领域名称：计算机技术

学生所属学院：计算机学院

答辩委员会主席：韩国强

论文答辩日期：二〇二〇年六月二十一日

A Dissertation Submitted to Guangdong University of Technology
for the Degree of Master
（Master of Engineering）

Research and application of offline handwritten Chinese character recognition based on deep learning

Candidate: Zhu Yuan
Supervisor: Haishui Xu

June 2020
School of Computer Science and Technology
Guangdong University of Technology
Guangzhou, Guangdong, P. R. China, 510006

摘要

手写汉字识别是人机交互的一种重要形式，在票据处理、文件录入等领域具有很高的实用价值。但因汉字的字符数量众多、每个人书写风格各异等原因，手写汉字识别的准确率一直不高。近年来，深度学习在图像识别、目标检测等领域取得了突破性的进展，逐渐成为了模式识别问题中常用的方法。本文基于深度学习的方法，综合利用特征提取技术对脱机手写汉字的识别进行了研究，使其识别准确率得到提升。主要的研究工作如下：

（1）结合 CASIA-HWDB1.1 数据集的具体情况，基于 AlexNet、VGGNet 和 GoogLeNet 构建多个不同结构的卷积神经网络（CNN）模型，研究不同结构的 CNN 模型对脱机手写汉字的识别效果。实验结果表明，基于 VGG-11 结构的端到端 CNN 模型的识别准确率最高，能够达到94.5%。

（2）在 VGGNet 结构的基础上，设计了多个不同深度的卷积神经网络模型，研究卷积神经网络的深度对识别准确率的影响。实验结果表明，在模型小于 11 层的情况下，模型结构越深，识别准确率越高。

（3）提出了 Gabor 特征和 HOG 特征融合 CNN 模型的识别方法。虽然端到端的 CNN 模型能够取得不错的结果，但是作为一个黑盒子，CNN 在接受原始图像输入的时候，会忽略一些特定领域的信息。本文利用传统图像处理领域的 Gabor 特征提取和 HOG 特征提取技术，综合改进对训练数据集进行特征提取，并将提取到的特征图与原训练数据集融合在一起作为 CNN 的训练数据。实验表明，Gabor 特征图和 HOG 特征图对 CNN 模型的识别准确率分别提高了0.8%和0.6%。

（4）对于 CNN 模型训练速度慢和不易收敛等问题，研究了基于迁移学习的微调技术和批量归一化算法对模型训练的加速效果，分析了不同 Dropout 概率对模型识别准确率的影响。最后选择实验结果较好的模型与参数，基于 TensorFlow 框架实现一个对脱机手写汉字图像进行识别的测试系统。

关键字：深度学习；卷积神经网络；手写汉字识别；特征提取

ABSTRACT

Handwritten Chinese Character Recognition(HCCR) is an important form of human-computer interaction, which has high practical value in bill processing, document entry and other fields. However, due to the large number of Chinese characters and different writing styles of each person, the accuracy of offline handwritten Chinese characters is not good enough. In recent years, Deep Learning has made a breakthrough in image recognition, object detection and other fields, and has gradually become a common method in Pattern Recognition. Based on the method of Deep Learning, this paper studies the off-line handwritten Chinese character recognition by using the feature extraction technology, which improves the recognition accuracy. The main research work is as follows:

(1) Follow the specific situation of CASIA-HWDB1.1 dataset, several Convolutional Neural Network(CNN) models with different structures are constructed to study the recognition effect of different CNN models on offline handwritten Chinese characters based on AlexNet, VGGNet and GoogLeNet. The experimental results show that the end-to-end CNN model based on VGG-11 structure has the highest recognition accuracy of 94.5%.

(2) Based on the VGGNet structure, several CNN models with different depths are designed to study the influence of the depth of CNN on the recognition accuracy. The experimental results show that when the model is less than 11 layers, the deeper the model structure is, the higher the recognition accuracy is.

(3) The recognition method of CNN model based on the fusion of Gabor feature and HOG feature is proposed. Although the end-to-end CNN model can achieve a nice results, as a black box, CNN ignores some information in specific fields when receiving the original image input. In this paper, Gabor feature extraction and HOG feature extraction technology in the traditional image processing field are used to improve the training dataset, and the extracted feature map and the original training dataset are fused together as CNN’s training data. The experiments show that the recognition accuracy of Gabor extraction and HOG extraction is improved by 0.8% and 0.6% respectively.

(4) For the problems such as slow training speed and difficult convergence of CNN model, this paper studies the acceleration effect of fine-tuning technology based on Transfer Learning

and Batch Normalization algorithm on model training, and analyzes the influence of different dropout probability on model's accuracy. At last, a testing system for off-line handwritten Chinese character recognition is implemented based on Tensorflow framework.

Keywords: Deep Learning; Convolutional Neural Network; Handwritten Chinese character recognition; Feature extraction

目录

CONTENTS

第一章 绪论

1.1 研究背景与意义

1950 年，计算机科学的伟大先驱阿兰·图灵（Alan Turing）在其论文《Computing Machinery and Intelligence》中提出了著名的图灵测试（Turing Test）：“一个人在不接触到对方的情况下，通过与对方的一系列对话，如果在相当长的一段时间内，无法判断对方是人还是机器，那么就可以认为这个机器是智能的[1]。”图灵测试的提出，为人工智能（Artificial Intelligence，AI）的发展起到了一定的指导作用。在其后的几十年间，人工智能在经历了多个繁荣与低谷之后，基于机器学习（Machine Learning，ML）的方法因其在多个领域中表现出色，逐渐成为了人工智能的主流方向[2]。近十年来，基于神经网络（Neural Network）结构的深度学习（Deep Learning, DL），在强大计算能力以及庞大数据量的推动下，掀起了新的一波人工智能浪潮，在该浪潮的推动之下，AI 的应用正在渗入各行各业。从 2012 年到今天，仅仅是 8 年的时间，AI 的应用随处可见。从进入广大公司的人脸识别、语音识别，到医学界的癌症检测、药物发现，到交通上的自动驾驶和智慧城市，再到学校饭堂里面根据不同餐盘识别不同类别菜肴进而统计价格的自动结账机器等等，都在进一步地影响着大众的生活。2017 年，国务院发表了《新一代人工智能发展规划》，将人工智能的发展上升到国家的发展战略，意图使用人工智能技术自动化地处理众多繁琐的任务，提高人们的生产力，改善人们的日常生活。

文字作为人类历史中的伟大发明，从其诞生以来，一直记载着人类历史中的重大事件。而汉字，作为中国历史的重要载体，书写着整个中华民族的兴衰荣辱，是中华文化传承至今的最大功臣。作为整个地球上使用人数最多的文字，汉字遍布于世界各个角落，如何自动地识别汉字，一直以来都是人们关注和研究的热点问题。汉字的识别通常分为印刷体汉字识别和手写体汉字识别（Handwritten Chinese Character Recognition，HCCR）[3]。印刷体汉字识别由于字体和排版规范，相对而言较为简单。手写体汉字由于不同个体的书写习惯迥异、采集环境不同等原因，一直以来都是一个极具挑战性的任务。手写体汉字识别又分为联机手写汉字识别和脱机手写汉字识别两类[3]。联机手写汉字识别指的是对数字手写板或者触摸屏等物理输入设备的在线文字信号进行处理，脱机手写汉字识别是对离线的手写汉字图像进行

处理和识别。由于脱机手写汉字缺乏笔画顺序等信息，加上不同的图像采集设备在不同光照和不同的纸张上也会带来较多的干扰，因此，脱机手写汉字识别是个相对较难的识别问题。脱机手写汉字广泛存在于快递单据、邮件信封、支票等地方，对其进行采集和自动识别能够给人们的生活提供极大的便利。尤其对于偏远地区的中老年人来说，由于缺乏智能设备的使用，手写汉字的录用能够拉近他们与外面世界的距离。

1.2 国内外研究现状

1.2.1 手写汉字识别的研究现状

汉字的识别可以追溯到 20 世纪 60 年代，来自 IBM 的 Casey 和 Nagy 通过模板匹配的方法对一千个印刷汉字进行有效的识别，并在 1966 年发表相应的论文[4]，引起了各界对汉字识别的初步关注。1977 年，日本东芝公司开发了首个可以识别两千个不同类型的常用汉字系统[5]。1980 年，国标 GB2312-80 正式推出汉字字符集的标准，给汉字识别研究领域提供了数据集的参考。20 世纪 90 年代，随着图像处理和模式识别的发展，手写汉字识别也在逐渐进步当中。在此期间，传统的图像识别方法主要形成了图像预处理、特征提取和图像分类三个步骤[6]。图像预处理方面，归一化算法[7]的出现，为汉字的不规范找到了解决的方案。特征提取方面，有结构特征和统计特征两种形式，而来自统计特征中的梯度特征提取[8]和 Gabor 特征提取[9]表现最为优秀。在分类算法中，常用的有支持向量机（Support Vector Machine，SVM）[10]、隐马尔可夫模型（Hidden Markov Model，HMM）[11]、改进的二次判决函数（Modifled Quadratic Discriminant Function，MQDF）[12]等均在提取的特征当中，较好的将不同的手写汉字进行分类。但是在之后较长的一段时间内，基于“预处理-特征提取-分类器”的手写汉字识别并没有取得了较大的进展。深度学习的流行，给手写汉字识别领域带来了新的研究思路。2011 年和 2012 年 ICDAR（International Conference on Document Analysis and Recognition）手写汉字识别的比赛中，获胜者都是基于深度学习的方法。2013 年来自英国华威大学的 Graham 使用深度稀疏卷积神经网络的方法[13]，将联机手写汉字识别的准确率提高到了 97.39%，获得了当界联机手写汉字识别的冠军。自此之后，深度学习在手写汉字识别领域取得了主导的地位。2013 年，富士通公司的团队采用了改进的卷积神经网络模型[14]，取得了当届 ICDAR 脱机手写汉字识别的冠军，识别率高达 94.77%。

相比于传统的手写汉字识别方法，深度学习展现了强大的特征提取和分类能力，能够达到更好的识别效果。但是不同结构的卷积神经网络模型对手写汉字识别准确率的影响较大，如何构建高效的卷积神经网络模型往往需要进行多次实验的对比。同时，因为汉字字符数量过大，卷积神经网络模型往往较为复杂，在训练过程中很容易造成过拟合的现象，如何更有效地进行模型的训练和测试还需要不断地研究和改进。

1.2.2 深度学习在图像识别上的研究现状

深度学习作为神经网络模型新的发展方向，其源头可以追溯到 20 世纪 40 年代。1943 年，来自美国的心理学家 McCulloch 和数学家 Pitts 参考生物神经科学提出了单个神经元的 M-P 模型[15]，该模型为后续的神经网络研究打下了基础。1958 年，Frank 等人提出了具有学习能力的单层感知机（Perceptron）模型[16]，该模型简单易懂，通过最小化损失函数的方式来训练超平面，使得超平面能够将特征空间一分为二，从而实现分类的效果。但是感知机模型并不能够处理线性不可分的问题，连简单的“异或”基本逻辑都无法实现，导致在之后较长的时间内，感知机模型得不到重视。1986 年，Rumelhart 等人提出了一种利用误差传播算法进行训练的多层感知机模型[17]，多层感知机模型也叫前馈神经网络或者反向传播网络（Back Propagation Network），该网络结构解决了一些单层感知机所不能解决的问题，但是 BP 神经网络在增加网络的层数时很容易遇到过拟合、梯度弥散和局部最优等问题。1989 年，Yann LeCun 等人提出了卷积神经网络结构（Convolutional Neural Network，CNN）[18]，并成功应用于手写数字识别。该网络模型为日后深度学习的高速发展打下了坚实的基础，但由于网络的训练需要大量的人工干预，训练速度较慢，当时并未得到学术界的重视。2006 年，多伦多大学的 Hinton 教授在《Science》上发文介绍了一种名为深度置信网络（Deep Belief Network，DBN）的全新网络结构，论文表示多隐层的神经网络具有很优异的特征学习能力[19]。针对神经网络难于训练的问题，文章提出了逐层预训练的方式，从而掀起了深度学习的研究热潮。2012 年，Hinton 教授的学生 Krizhevsky 等在 ImageNet 图像识别大赛上设计了卷积神经网络 AlexNet[20]，以较大的优势赢得了当届图像识别赛事的冠军，从而使深度学习名扬天下。AlexNet 让人们认识到了卷积神经网络强大的特征提取和分类的能力，为图像识别领域开创了一条新的道路。此后，深度学习迎来了繁荣期。2014 年，牛津大学的 Simonyan 等

提出了 VGGNet[21]网络结构和 Google 公司 Szegedy 等提出的 GoogLeNet[22]网络模型分别获得了 2014 年 ImageNet 竞赛的前两名，VGGNet 和 GoogLeNet 两个不同的网络结构给业界构建卷积神经网络模型提供了新的思路。为了进一步加深网络的深度，何凯明等于 2016 年提出了深度残差网络（Residual Network）结构[23]，该网络结构通过残差块的结构，能够进一步加深神经网络的深度且有效地减缓了过拟合和梯度弥散等现象。

时至今日，深度学习的研究日新月异，多种神经网络结构纷纷被提出，更加优秀的优化算法、分布式训练方法以及越来越多优秀的深度学习框架也使得深度学习模型更加容易训练。在优化算法方面，在最小化损失函数（Loss Function）的目标下，梯度下降（Gradient Descent）算法是目前深度学习中最为常用的优化算法之一，小批量随机梯度下降算法采用 mini-batch 的方式进行一定的改进，同时保证了训练的速度和收敛的准确率，但是小批量随机梯度下降算法容易陷入鞍点和局部最优点。Momentum 算法[24]在小批量随机梯度算法上加入了动量的思想，能够加速模型的收敛速度以及减缓陷入局部最优的情况。AdaGrad 算法[25]、AdaDelta 算法[26]、RMSProp 算法[27]加入了自适应学习率的机制，使得在模型的训练过程中能够动态地调整模型的学习率，加快了模型的收敛速度。Adam 算法同时集合了动量机制和自适应学习率的机制，利用梯度的一阶矩估计和二阶矩估计动态调整学习率，使参数更加稳定[28]。还有更具鲁棒性的 Radam 算法[29]等，都在进一步高效地优化目标函数。而在工业界的努力之下，各种深度学习框架（如 PyTorch、TensorFlow、MxNet、Caffe 等）的出现，给各大研究人员以及工程师实现深度学习的实验及应用带来了极大的便利，使得深度学习应用走向寻常百姓家成为了可能。目前，深度学习技术已经在多个领域取得了巨大的成功，比如计算机视觉、人脸识别、自然语言处理、推荐系统等领域[30]。

1.2.3 手写汉字识别的研究难点

手写汉字识别是一个较为困难的问题，主要表现在以下三个方面[31]。

（1）汉字字符的类别繁多。与英语或者拉丁字母不同，汉字由古老的象形文字逐步演变而来，存在着数量众多的汉字字符。仅仅是 1980 年制定的国标编码 GB2312-80 就收录了 6763 个汉字字符，其中包含 3755 个常用的一级汉字和 3008 个二级汉字，远高于 26 个英文字母。而 2000 年发布的国标 GB18010 更是达到了两万

多个字符。如此庞大的类别，给汉字识别造成了极大的挑战。

（2）汉字字符存在大量的形近字。比如“甲-田-由-曲”、“大-太-犬”、“海-悔-诲”等，虽然在字符结构和笔画中相差不大，但却代表着完全不同类别的汉字，很容易造成困扰。

（3）每个人的书写风格不一。与打印的字体相比，手写汉字缺乏严格的规范性。中国有超过 13 亿的人口，每个独立个体的字迹都有着其自身的独特性，即使是同一个人，在不同的时间段或不同的环境之下书写的汉字在形状上也不是完全相同。汉字的基本笔画也因为个人的书写习惯，通常差异很大。

1.3 本文的研究内容和贡献

本文结合脱机手写汉字识别的研究现状，对于其识别准确率不高、深度学习模型难以训练等问题，主要的研究内容和贡献为：

（1）基于 AlexNet、VGG 和 GoogLeNet 设计和改进了五个不同深度、不同结构的 CNN 模型，研究不同深度不同结构的网络模型对手写汉字识别准确率的影响。

（2）提出了 Gabor 特征和 HOG 特征融合 CNN 的识别方法，探讨了传统图像处理领域的特征提取方法对 CNN 模型识别准确率的影响。实验表明，传统图像特征提取方法与 CNN 的结合，能够提高网络模型的识别效果。

（3）针对 CNN 模型容易陷入过拟合的情况，设计了多组实验探讨不同的 dropout 概率对 CNN 模型识别性能的影响，以及对过拟合现象的减缓作用。

（4）对于深度学习模型训练速度缓慢的特点，在模型的训练过程中先使用较小的数据集进行模型预训练，再用微调（fine-tuning）的技术对模型的参数进行初始化。同时，研究了批量归一化算法对模型训练的加速效果。

（5）通过实验的对比，选择了其中一个实验结果较好的卷积神经网络模型和对应的模型参数，实现了一个对输入手写汉字图像的识别系统。

1.4 本文章节安排

本文的结构分文为五章，每章的内容如下：

第一章：绪论。本章主要介绍了手写汉字识别的研究背景及其意义，描述了手写汉字识别和深度学习在图像识别领域的研究现状、手写汉字识别的研究难点以及本文的研究内容和章节安排。

第二章：深度学习的相关理论。本章主要介绍了与深度学习相关的基础理论，

包括单个神经元、前馈神经网络的数学模型和卷积神经网络的具体结构。同时介绍了反向传播算法以及在模型训练过程中经常用到的 dropout 技术和批量归一化算法。

第三章：基于卷积神经网络的手写汉字识别。本章首先介绍了本文所使用的数据集 CASIA-HWDB1.1 以及图像识别领域中常用到的数据预处理技术。其次构建了五个不同结构的卷积神经网络模型，研究不同的模型结构对手写汉字识别准确率的影响，最后在 TensorFlow 框架的基础上进行了相应的实验，并对实验结果进行了一定的分析。

第四章：特征提取融合 CNN 的手写汉字识别方法。本章首先介绍了两种传统图像处理领域的特征提取方法：Gabor 特征提取和 HOG 特征提取，并将其对训练数据集提取到的特征图与原训练图像合并输入到卷积神经网络模型中进行训练，研究 Gabor 特征和 HOG 特征对模型识别准确率的影响，最后探讨了不同 dropout 概率对识别准确率的影响以及批量归一化算法对模型训练的效果。

第五章：总结和展望。本章对本文的研究成果和不足进行总结，并展望一下深度学习和脱机手写汉字识别未来的研究方向。

第二章 深度学习的相关理论

2.1 引言

深度学习是机器学习的一个子问题，典型的深度学习就是一个很深层的神经网络。机器学习中的神经网络起源于人类大脑的神经系统，是为了模拟人类大脑神经系统而设计出来的一种计算模型。该模型通过多个节点的互相连接，并为每个连接赋予适当的权重，以对复杂关系的建模。在神经网络模型中，每个节点都相对应着一个特定的函数，来自其他节点的输入信号通过不同权重的组合，加上对应的激活函数，可以模拟出一个非常复杂的逻辑关系[32]。在理论情况下，模型的参数愈多，复杂度愈高，其表达能力就愈强。然而，越是复杂的模型，其训练难度越大，很多情况下得不到较好的训练结果，需要在模型的表达能力和模型的复杂度之间做出一定的权衡。“权值共享”是一个能够减少模型参数的有效方法，该方法中一组神经元使用相同的权值，在保证了模型表达能力的同时，使得模型更加容易训练。CNN 的构建就是采用这样的思路。

与动物视觉系统对物体的识别过程类似，神经网络模型对于图像的分析过程采用一种分层的结构。模型的浅层结构提取图像的边缘特征等基础信息，中间隐藏层通过组合各种边缘信息形成一定的角度和轮廓，而在往后的隐藏层中，轮廓和角度的组合便成了图像中物体的一部分，通过这种分层组合的方式，使得神经网络模型最终能够完成识别物体的任务。因此，在深度学习模型中，模型的层次关系对学习表达能力非常重要，而深度越深，模型参数越多，所需要的数据量也会越来越多。

2.2 传统前馈神经网络模型

2.2.1 神经元

神经元是组成复杂网络结构的最小单元。一个神经元接收来自n个不同神经元输入进来的信号，每个传入的信号都附带着一个相应的权值w_i，神经元对传递过来的信号与权重进行求和，并加上自身的偏置值，最后通过激活函数（activation function）处理，产生输出结果。单个神经元的模型如图 2-1 所示：

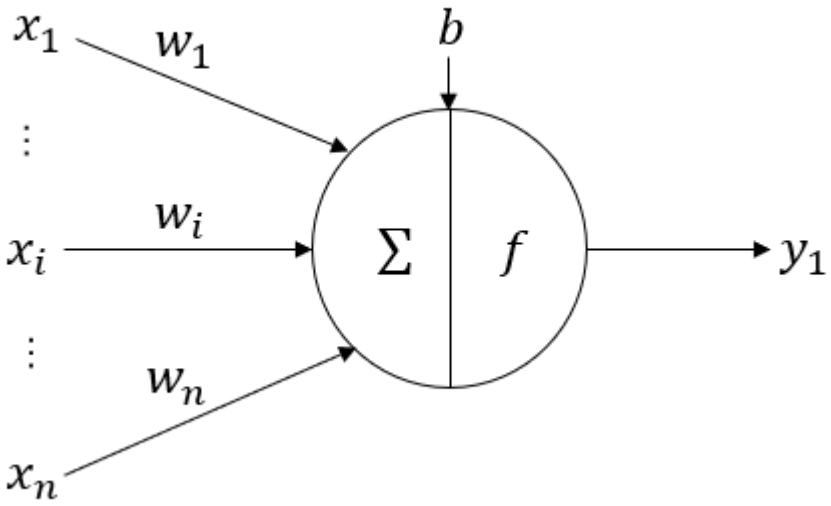


图 2-1 神经元的模型结构

Fig2-1 The model structure of Neuron

在神经元数学模型中，假设传入的信号为向量x，权值为w，偏置值为b，则神经元模型的输出结果为：

$$y = f\left(\sum_{i=1}^{n} w_i x_i + b\right) \tag{2.1}$$

其中f为激活函数（activation function）。常用的激活函数有：Sigmoid 函数和修正线性单元（Rectified Linear Unit, ReLU）函数。Sigmoid 的表达式为：

$$y = \frac{1}{1 + e^{-x}} \tag{2.2}$$

对应的函数曲线图如图 2-2 所示：

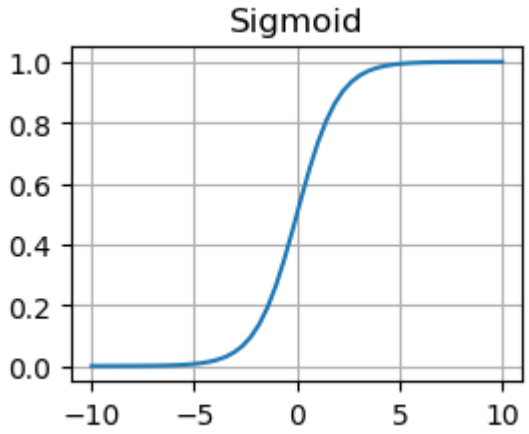


图 2-2 Sigmoid 函数曲线图

Fig2-2 The Sigmoid function graph

ReLU 函数的表达式为：

$$y = \begin{cases} 0, & x < 0 \\ x, & x \geq 0 \end{cases} \tag{2.3}$$

对应的函数曲线图如图 2-3 所示：

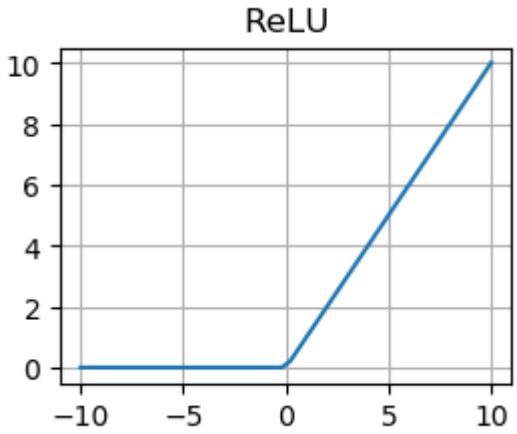


图 2-3 ReLU 函数曲线图

Fig2-3 The ReLU function graph

2. 2. 2 前馈神经网络

单个神经元组成的数学模型，其学习能力非常有限，只能处理一些简单的线性可分的问题。想要解决非线性可分等复杂问题，需要将多个神经元结合在一起，形成具有多个层级结构的前馈神经网络（Neural Network,NN）。该网络的示意图如图

2-4 所示。

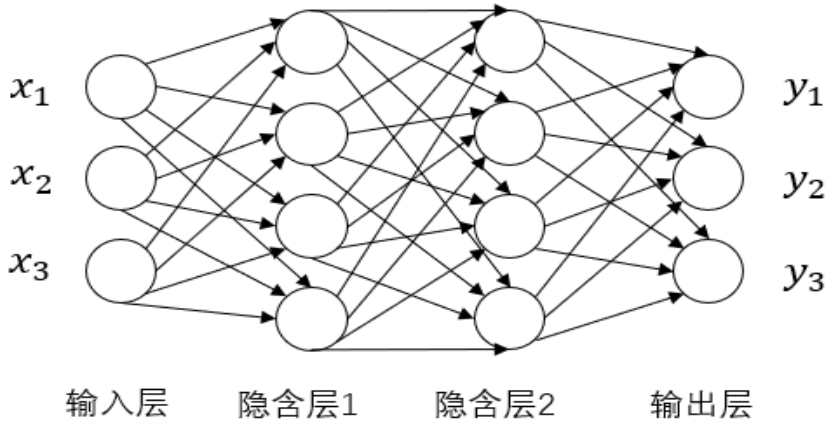


图 2-4 前馈神经网络示意图

Fig2-4 Diagram of feedforward neural network

前馈神经网络在结构上可分为输入层、隐含层和输出层[2]。输入层接受外界的输入信号，隐含层负责对输入信号进行学习和加工，输出层对神经网络所学习到的结果进行输出。在前馈神经网络结构中，相同层的神经元之间没有进行连接，而与相邻层的所有神经元进行连接，每个连接都对应了一个相应的权值。隐含层和输出层中每个神经元的输入为前一层所有神经元的输出值与对应权值乘积的和。设a_i^{l-1}为第$l-1$层第i个神经元的输出，w_{ij}^l为第$l-1$层第i个神经元与第l层第j个神经元的连接权值，b_j^l为第l层第j个神经元的偏置值，则第l层第j个神经元的输入值z_j^l为：

$$z_j^l = b_j^l + \sum_{i=1}^{k} a_i^{l-1} \times w_{ij}^l \tag{2.4}$$

该神经元的输出值a_j^l为：

$$a_j^l = f\left(z_j^l\right) \tag{2.5}$$

公式（2.4）中，k表示第$l-1$层神经元的个数。

在分类任务中，输出神经元的个数往往等于分类任务中类别的个数。为了提高网络的学习效率和方便数据的计算，在分类问题中，往往需要对输出层的输出结果进行 Softmax 函数处理。Softmax 函数公式为：

$$P_i = \frac{e^{a_i}}{\sum_{j=1}^{k} e^{a_j}} \tag{2.6}$$

其中，a_i为最后一层第i个神经元的输出，k表示最后一层神经元的个数。

Softmax 函数将最后一层每个神经元的输出值映射到$(0,1)$之间，并将每个神经元的输出值转化为该神经元对应类别的概率P_i。即对于输入信号x而言，前馈神经网络最后一层的每个神经元对应的输出结果为第i个分类的概率$P(Y=i \mid x)$。

2.2.3 反向传播

刚定义的前馈神经网络的每个权值都是随机的值，并不能够对输入数据进行正确的分类，需要对其进行反复的训练，以达到拟合实际情况的需求。目前为止，对前馈神经网络最为常用的训练方法是方向传播算法（Back Propagation）。反向传播算法由以下两种方向的计算组成：1）神经网络的前向传播计算出输入向量对应的结果。2）对比输出结果与真实标签之间的误差，并通过梯度下降的策略更新网络模型的权值。

对于训练样本(x,y)，其中$y=(y_1,y_2,\dots,y_k)$，假定神经网络的输出为$a_j^l=f(b_j^l+\sum_{i=1}^{k}a_i^{l-1}\times w_{ij}^l)$，则该神经网络模型在$(x,y)$上的损失函数可定义为：

$$E(w,b)=\frac{1}{2}\sum_{j=1}^{k}\left(y_j-a_j^l\right)^2 \tag{2.7}$$

神经网络模型的反向传播过程就是不断更改模型的参数w和偏置值b，逐渐最小化损失函数E。作为权值w和偏置值b的函数，根据梯度下降策略，逐步计算损失函数E和权值w与偏置值b的梯度，使其往负梯度的方向进行更新。即：

$$w_{ij}^l=w_{ij}^l-\eta\frac{\partial E}{\partial w_{ij}^l} \tag{2.8}$$

$$b_j^l=b_j^l-\eta\frac{\partial E}{\partial b_j^l} \tag{2.9}$$

其中，η为一个$(0,1)$的正数，称为学习率，控制着反向传播算法每一轮迭代对参数w和b的修改步长。

由链式求导法则可得：

$$\frac{\partial E}{\partial w_{ij}^l}=\frac{\partial E}{\partial a_j^l}\cdot\frac{\partial a_j^l}{\partial w_{ij}^l}=\frac{\partial E}{\partial a_j^l}\cdot\frac{\partial a_j^l}{\partial f}\cdot\frac{\partial f}{\partial w_{ij}^l} \tag{2.10}$$

式（2.10）中，f为激活函数。同理，

$$\frac{\partial E}{\partial b_j^l}=\frac{\partial E}{\partial a_j^l}\cdot\frac{\partial a_j^l}{\partial b_j^l}=\frac{\partial E}{\partial a_j^l}\cdot\frac{\partial a_j^l}{\partial f}\cdot\frac{\partial f}{\partial b_j^l} \tag{2.11}$$

在神经网络的训练过程中，从输出层到多个隐含层，再到输入层，不断计算损失函数对权值w和偏置值b的负梯度并进行相应的更新，直到计算出整个模型的最优解。

2.2.4 前馈神经网络存在的问题

前馈神经网络可以很好地拟合多个不同的函数。但是当前馈神经网络处理二维图像信息的时候，会存在两个显著的缺点。

1）参数过多。前馈神经网络在处理二维信息的时候，会将二维信息展开形成一维向量的形式。以输入图像大小是$100 \times 100 \times 3$为例，展开后的输入数据为$x = \{x_1, x_2, \dots, x_{30000}\}$，对应着输入神经元的个数为 30000，而第一个隐含层中每个神经元都会有 30000 个互相独立的连接，每个连接都对应着一个独立的权值参数。随着每个隐含层神经元数目的增长和隐含层个数的增长，参数的规模也会急剧增加，最终导致整个模型难以训练，非常容易造成过拟合的现象。

2）不具备局部不变性特征。在自然图像中，物体可能存在于图像中的每个角落，物体的大小也会因为图像的尺寸而发生改变，镜头的角度不同也会造成物体的方向不一。前馈神经网络在处理图像时，并不能提取出这些局部不变的特征，而是将不同位置的同一物体、不同尺寸的同一物体、不同角度的同一物体等作为不同物体进行处理，丢失了图像原有的语义信息。

2.3 卷积神经网络

卷积神经网络（Convolutional neural network，CNN）是一种具有局部连接、权值共享等特性的神经网络，广泛应用于图像处理领域。启发于生物学家对动物视觉皮层的研究，卷积神经网络使用“局部感受野”的方式来处理输入图像，即卷积层中的神经元只接受其覆盖区域内的信号[33]。这种特性可以极大地减少神经元之间连接权值的数量，降低模型的复杂度。由于 CNN 独特的结构，使得其在图像上具有一定程度的局部不变性，非常适合于图像识别等领域。

2.3.1 卷积神经网络的结构

卷积神经网络结构一般由卷积层、池化层和全连接层组成[18]。典型的 LeNet-5 结构如图 2-5 所示。

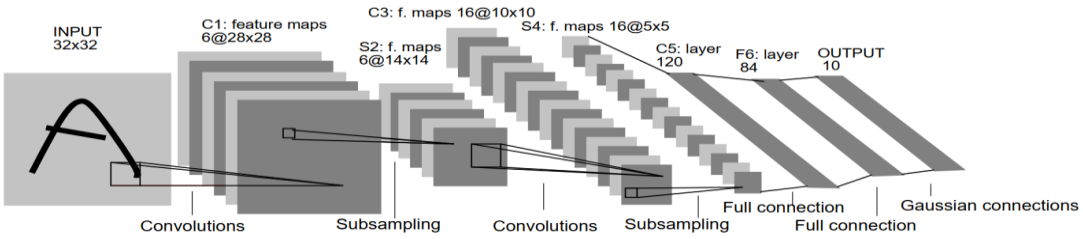


图 2-5 LeNet-5 结构示意图[18]

Fig2-5 LeNet-5 Structure diagram

卷积层的主要作用是利用多个卷积核提取输入图像中多个局部区域的信息，形成一系列的特征图（Feature Map）。特征图经过池化处理（也叫子采样处理），可以达到降维的效果，从而减少参数的数量。一个或多个卷积层加上一个池化层，可以抽象为一个卷积块。CNN 模型的前半部分，通常由多个卷积块依次堆叠组成。从输入数据往输出结果的方向，通常情况下卷积计算得到的单个特征图尺寸越来越小，而输出的特征图的数量越来越多。模型的后半部分由全连接神经网络组成，即是 2.2 节所介绍的前馈神经网络。

CNN 整个模型结构可抽象为图 2-6 所示[34]。M（M>=1）个卷积层和一个池化层组合形成一个卷积块。整个网络结构由 N 个连续的卷积块堆叠而成，其后接着 K 个全连接层，全连接层输出的特征向量往往需要通过分类器处理。常用的分类器有：Softmax、SVM、logistic 回归。

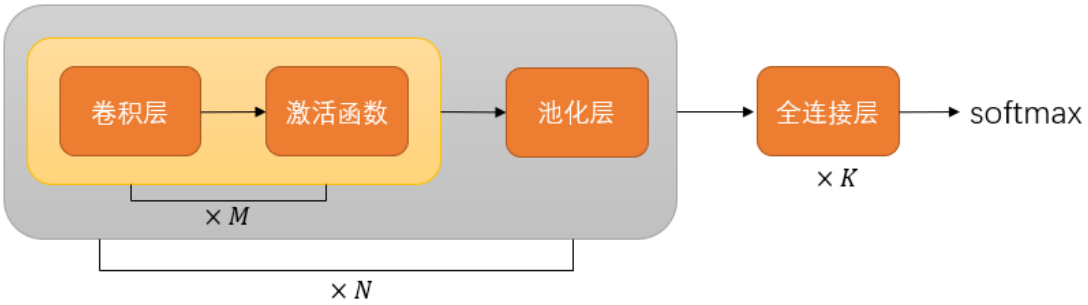


图 2-6 CNN 结构示意图

Fig2-6 A structure of CNN

在前馈神经网络中，如果第$l-1$层有$k-1$个神经元，第l层有k个神经元，则仅仅是两层之间的权值参数就有$(k-1)\times k$个。当k值较大或者层数较多时，整个模型

的权值参数就会呈爆炸性增长，从而导致训练的难度非常大，效率非常低。作为 CNN 的核心，卷积层的效果可以避免这种参数过多的情况。

卷积运算在 CNN 结构中也称为互相关（cross correlation）运算。通常情况下，卷积层的输入为一组特征图，$X \in R^{M\times N\times D}$，其中每一个特征图$x \in R^{M\times N}$。模型的参数卷积核通常为四维的张量$W \in R^{U\times V\times P\times D}$，其中一个切片为$w \in R^{U\times V}$二维矩阵。

假设输入数据的是一张二维的特征图$x \in R^{M\times N}$，模型定义的一个卷积核为$w \in R^{U\times V}$，则经过卷积计算的结果为：

$$z = \sum_{d=1}^{D} w \otimes x + b \tag{2.12}$$

式（2.12）中，$\otimes$表示卷积的操作，d表示卷积计算的区域。其中，

$$z_{ij} = \sum_{u=1}^{U}\sum_{v=1}^{V} w_{uv} x_{i+u-1,j+v-1} + b \tag{2.13}$$

式（2.13）中，b为偏置值，z_{ij}为该卷积核对所计算区域的输出。详情如图 2-7 所示。

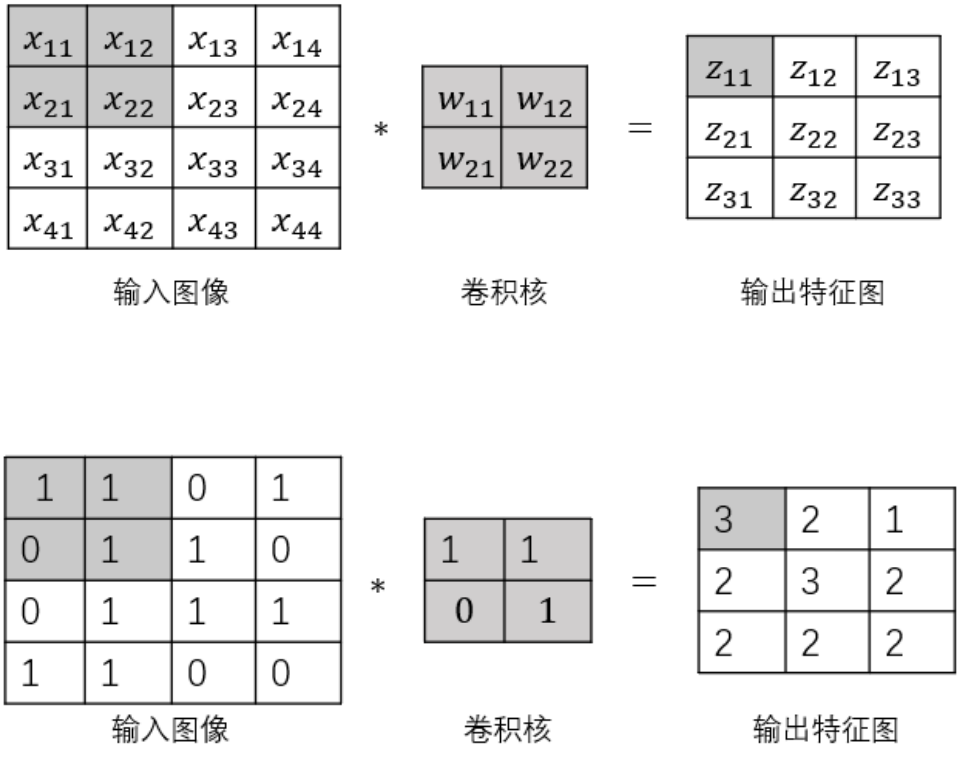


图 2-7 卷积计算的过程

Fig2-7 The process of convolution calculation

在图 2-7 中，卷积核作用于输入特征图的左上角位置，覆盖的区域为2×2，卷积计算的输出结果为：

$$z_{11} = w_{11}x_{11} + w_{12}x_{12} + w_{21}x_{21} + w_{22}x_{22} + b \tag{2.14}$$

类似的，卷积核按照一定的步长（stride）与输入特征图进行卷积的计算，得到该卷积层的输出特征图。

通常在卷积计算过后，需要对计算结果进行非线性激活处理，然后输入到下一层网络中。即将卷积计算的结果z输入到激活函数中，对应的公式为：

$$a = f(z) \tag{2.15}$$

公式（2.15）中，f为激活函数，常用的是 ReLU 激活函数。

整个卷积的计算过程，就是一个三维的输入特征图与四维的卷积核进行互相关计算，计算后的输出结果为一个全新的三维特征图。

在卷积计算的基础上，可以加入对输入特征图的零填充（zero padding）和卷积核的滑动步幅（stride）来增加卷积计算的多样性，使得卷积层能够更加灵活地提取特征。零填充即是给原输入特征图的四周分别添加内容为 0 的元素，使得输入特征图在外围多出一圈，避免了输入特征图的边界信息被忽略。滑动步幅指的是卷积核在输入特征图计算过程中移动的步数。图 2-8 展示了零填充为 1 的效果图。

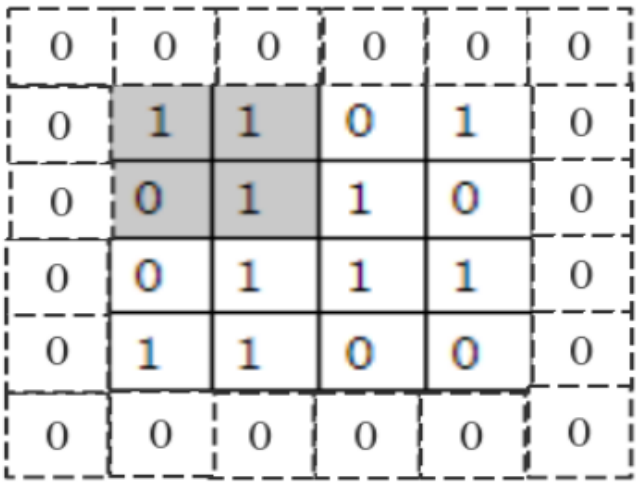


图 2-8 零填充的效果

Fig2-8 The effect of zero padding

在加入零填充和滑动步幅的卷积计算中，设原始的特征图尺寸为$n \times n$，卷积核尺寸为$k \times k$，填充的数量为p，卷积核的滑动步幅为s，则输出特征图的尺寸为：

$$\frac{(n-k+2p)}{s+1} \tag{2.16}$$

池化层（Pooling Layer）也叫做子采样层（Subsampling Layer），其主要的作用是进行特征的选择和特征的降维。池化层按照运算的过程可分为最大池化层（Max Pooling）和平均池化层(Average Pooling)。

对于最大池化层而言，池化的作用就是在输入特征图中某个区域，遍历挑选出该区域内的最大值。假设某个输入特征图$x \in R^{M\times N}$，池化的区域为R_{mn}，其中$1 \le m \le M$，$1 \le n \le N$，则对该区域的最大池化结果为：

$$y = \max(x_i) \tag{2.17}$$

式（2.17）中，x_i为区域R_{mn}中的值。

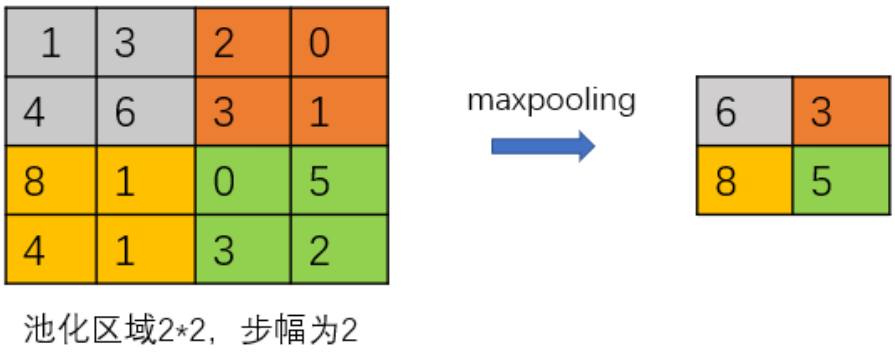


图 2-9 最大池化层的计算

Fig2-9 The calculation of maxpooling

对于平均池化层而言，输出值为区域R_{mn}内对应特征值的平均信息。

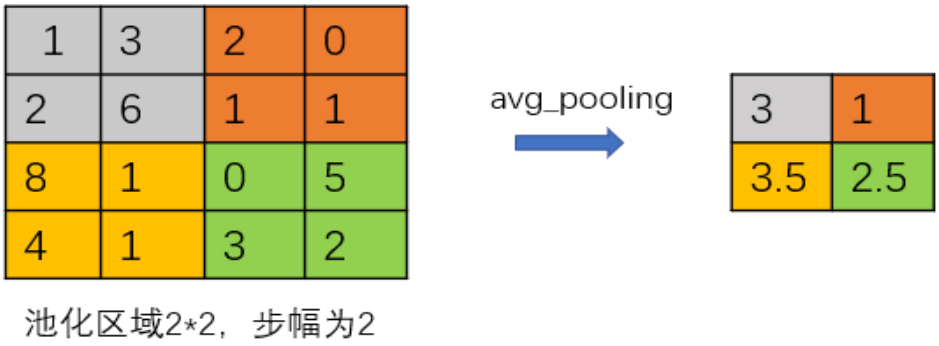


图 2-10 平均池化层的计算

Fig2-10 The calculation of average pooling

与卷积层类似，池化层的滑动步幅可以根据实际的情况做出相应的改变。但是过大的步幅会导致神经元的个数急剧降低，对应的特征信息也会有所损失。

2.3.2 卷积神经网络的特点

经过前面章节的介绍分析可得，与前馈神经网络相比，CNN 的主要特点是局部连接和参数共享[35]。

（1）局部连接。如图 2-11 所示，与全连接层相比，卷积层中卷积核的神经元与输入特征图仅仅是局部连接。假设M^l为第l层的神经元个数，M^{l-1}为第$l-1$层神经元的个数，K为卷积核的尺寸，则在全连接层中，总共需要$M^l \times M^{l-1}$ 个连接参数，而在卷积层中，连接参数为$M^l \times K$，一般情况下，K远远小于M^{l-1}。

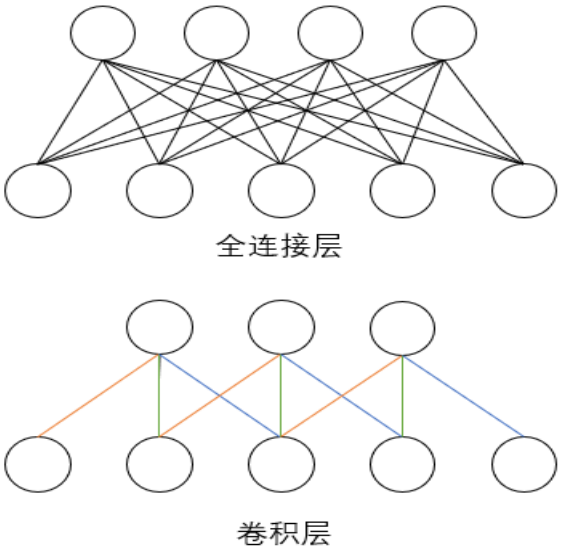


图 2-11 局部连接与参数共享的示意图

Fig2-11 Sparse connection and Weight sharing

（2）权值共享。在 CNN 中，作为参数的卷积核对于输入特征里面所有的神经元都是相同的。每个卷积核都重复地作用于多个感受野区域，这样地好处是一个卷积核能够提取到位于不同感受野中相同的特征，而不用定义多个不同的卷积核，避免造成过多的参数。

2.3.3 卷积神经网络的反向传播

与前馈神经网络的思想类似，CNN 的 BP 过程是通过最小化损失函数从而对卷

积核中的参数进行更新。在卷积神经网络中，后半部分的全连接层的输出通常进行Softmax 函数处理，处理之后的输出结果与真实标签进行对比。而在分类任务中，真实的标签通常进行 one-hot 编码处理，即对于一个类别数量为 K 的分类任务，ont-hot 编码使用维数为 K 的向量来表示所有的类别，其中每个类别对应于向量中的某个位置。例如，当 K=3 时，类别 1 对应的向量为[1,0,0]，类别 2 对应的向量为[0,1,0]，类别 3 对应的向量为[0,0,1]。

在 Softmax 函数处理输出结果和 one-hot 编码处理真实标签的情况下，常用的损失函数为交叉熵损失函数（cross-entropy），公式定义为：

$$L(w,b) = -\sum_{i=1}^{K} y_i \ln a_i \tag{2.18}$$

式（2.18）中，K代表最后一层输出神经元的个数，即数据集的类别个数，y_i代表当前输入样本对应的真实类别向量，a_i是第i个输出神经元对该样本的输出。在训练过程中，通过不断最小化损失函数来调整整个网络的参数，使得模型的输出与样本的真实数据之间的差距逐渐靠近，以达到拟合数据集的目的。对于卷积层而言，在进行反向传播的时候，假设当前层的误差信息为δ^l，定义为

$$\delta^l = \frac{\partial L(w,b)}{\partial z^l} = \frac{\partial L(w,b)}{\partial a^l} \cdot \frac{\partial a^l}{\partial z^l} \tag{2.19}$$

若第l层为卷积层，该层的特征净输入为：

$$z^l = \sum_{d=1}^{D} w^l \otimes x^{l-1} + b^l \tag{2.20}$$

则前一层的误差信息δ^{l-1}

$$\delta^{l-1} = \frac{\partial L(w,b)}{\partial z^{l-1}} = f'(z^{l-1}) \odot [\delta^l \otimes rot180(w^l)] \tag{2.21}$$

其中⊗为卷积计算，⊙为矩阵对应位置的乘积。在卷积神经网络进行反向传播时，通常是通过第l层的误差信息计算出第$l-1$层的误差信息，从而计算出损失函数对模型参数的偏导数。损失函数对权值和偏置值的偏导数分别为：

$$\frac{\partial L}{\partial w^l} = \frac{\partial L(w,b)}{\partial z^l} \cdot \frac{\partial z^l}{\partial w^l} = \delta^l \cdot \frac{\partial z^l}{\partial w^l} = a^{l-1} \otimes \delta^l \tag{2.22}$$

$$\frac{\partial L}{\partial b^l} = \sum_m \sum_n \delta^l_{mn} \tag{2.23}$$

对于池化层而言，由于没有需要训练的参数，因此只需要将误差传递给上一层即可。如果使用的最大池化层，需要将当前误差信息传播到池化区域最大值的位置；如果使用的是平均池化层，则需要将当前的误差信息平均传到池化区域中。

2.4 Dropout

在深度学习的训练过程中，因为模型参数过多或者数据集过少等因素，过拟合是一个很常见的现象。实用的减缓过拟合的方式有扩大训练数据集、Dropout（丢弃法）[36]等。

Dropout 的思想是在训练神经网络模型的过程中，随机丢弃一部分的神经元及其对应的连接。假设对于神经网络中的某一隐藏层来说，神经元的丢弃概率为p，那么该层中的神经元会有p的可能性会被丢弃，Dropout 的效果图如图 2-12 所示。在每次小批量的训练的时候，随机丢弃的神经元都不是相同的。输入数据经过前向传播的计算过程中，受抑制的神经元不参与计算，而在反向传播的过程中，在没被丢弃的神经元之间，经过梯度下降算法来更新对应的权重。而在下一个小批量的输入数据到来时，又会按照p的概率重新随机丢弃一部分的神经元。如此反复，直到模型训练完毕。Dropout 方法往往只使用在训练数据集中，在测试集测试的时候，通常不对隐藏层的神经元进行丢弃。

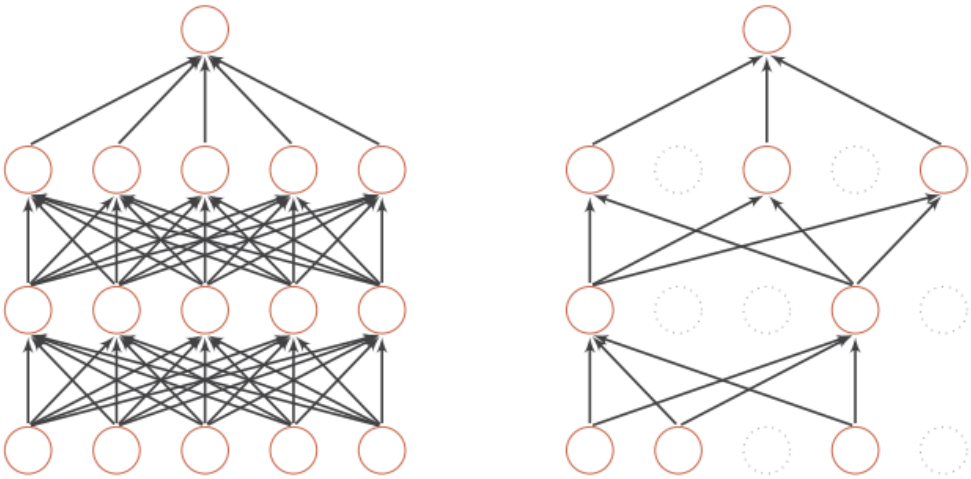


图 2-12 Dropout 的示意图[34]

Fig2-12 The diagram of dropout

从集成学习的角度来讲，Dropout 方法每次丢弃或者抑制概率为p的神经元，相当于从原始的神经网络模型中提取出一个子模型，而每次的迭代中，都约等于训练

一个与上一次迭代不一样的子模型，所有的这些子模型都会共享着原始的模型参数，最终，整个神经网络模型相当于集成了多个不同网络结构的组合模型。

2.5 批量归一化

批量归一化算法（Batch Normalization, BN）是由谷歌研究院 Sergey Ioffe 提出来的，该算法通过减轻神经网络训练时的内部协变量偏移[37]（Internal Covariate Shift），从而对模型的训练起到加速的作用。在模型的训练过程中，批量归一化算法利用小批量训练样本上的均值和方差，不断调整神经网络的中间输出，使整个神经网络在各层的中间输出数值更稳定，从而更容易训练。

假设神经网络中某层的输入是大小为 m 的小批量数据$X = \{x^1, \dots, x^m\}$，其中小批量X中任意的样本$x^i \in R^d, 1 \le i \le m$。对小批量$X$求均值和方差：

$$\mu_X = \frac{1}{m}\sum_{i=1}^{m} x^i \tag{2.24}$$

$$\sigma_X^2 = \frac{1}{m}\sum_{i=1}^{m}\left(x^i - \mu_X\right)^2 \tag{2.25}$$

接着标准化x^i中的每一维度，得到

$$\hat{x}^i = \frac{x^i - \mu_X}{\sqrt{\sigma_X^2 + \epsilon}} \tag{2.26}$$

其中ϵ是一个大于 0 很小的常数，是为了保证分母大于 0。

在标准化的基础上，BN 算法引入了两个能够学习的d维向量系数：拉伸系数γ和偏移系数β，它们与$\hat{x}^i$分别做按元素乘法和加法计算：

$$y^i = \gamma \odot \hat{x}^i + \beta \equiv BN_{\gamma,\beta}\left(x^i\right) \tag{2.27}$$

得到的y^i就是批量归一化算法对于输入x^i的输出。

在神经网络的训练中，批量归一化算法常常作为网络模型中的一层 BN 层，且通常应用在卷积层和全连接层中。在对卷积层做批量归一化的时候，BN 层通常加在卷积计算之后、应用激活函数之前。全连接层类似，一般加在非线性激活函数之前。

2.6 本章小结

本章介绍了深度学习的基本理论知识，从最简单的单个神经元计算模型开始，

介绍了两个常用的激活函数 sigmoid 函数和 ReLU 函数，从而引出了多个神经元结构组成的前馈神经网络模型，以及训练模型的 BP 算法，并描述了前馈神经网络模型在训练中可能遇到的问题，从其存在的问题中引出了功能更加强大的卷积神经网络结构。本章详细地介绍了 CNN 中卷积层和池化层的计算流程以及每层的作用，描述了 CNN 的一般结构形式以及 CNN 模型的特点，计算了卷积神经网络在反向传播算法中修改权值的过程，最后介绍了在模型训练过程中常用的 Dropout 技术以及批量归一化算法。

第三章 基于卷积神经网络的手写汉字识别

3.1 引言

传统的脱机手写汉字识别的流程与普通的图像识别流程基本相同，可分为以下步骤：输入样本、样本预处理、图像特征的提取、图像的分类，流程图如图 3-1 所示。在传统的手写汉字识别中，图像特征的提取是整个流程的重中之重，往往决定了识别的准确率。图像的分类通常由机器学习算法进行分类，如 SVM、Adaboost 等。

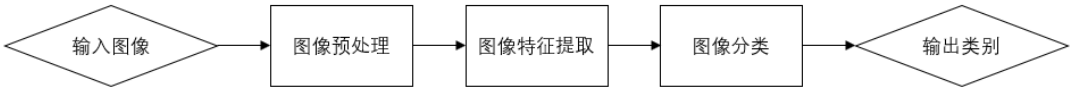


图 3-1 传统图像识别方法

Fig3-1 Traditional image recognition method

卷积神经网络的一大特点就是可以集特征提取和分类于一体，因此可以使用卷积神经网络实现端到端（end-to-end）的脱机手写汉字识别，从而省去繁琐易错的特征提取的流程。而基于 CNN 模型的图像识别方法的大致流程图如图 3-2 所示:

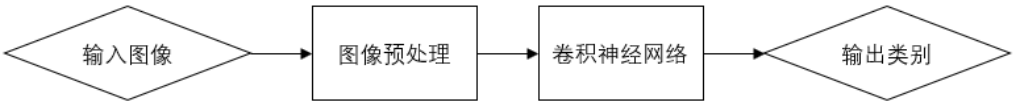


图 3-2 基于 CNN 的图像识别方法

Fig3-2 Image recognition based on CNN

对于输入的训练图像，进行简单的预处理，即可将其输入到 CNN 模型当中，经过大量的有监督学习后，整个网络模型最终能够识别出输入图像的各种特征，从而对新的输入图像进行判别和分类。

3.2 数据集与数据预处理

3.2.1 数据集

常用于脱机手写汉字识别的数据集有中国科学院发布的 CASIA-HWDB1.0-1.2[38]、北京邮电大学发布的 HCL2000[39]以及华南理工大学发布的 SCUT-COUCH[40]。本文选取中国科学院发布的 CASIA-HWDB1.1 用于训练神经网络模型。CASIA-HWDB1.1 数据集总共包含 3755 种类别的手写汉字，每个类型的汉字由 300 个不同的作者贡献而成。部分数据如图 3-1 所示。整个 CASIA-HWDB1.1 数据集的样本数目为 1121749，划分为 897758 张训练数据集和 223991 张测试数据集。其中训练数据集由 240 个不同的作者所写，测试数据集由 60 个不同的作者所写。CASIA-HWDB1.1 的信息如表 3-1 所示。

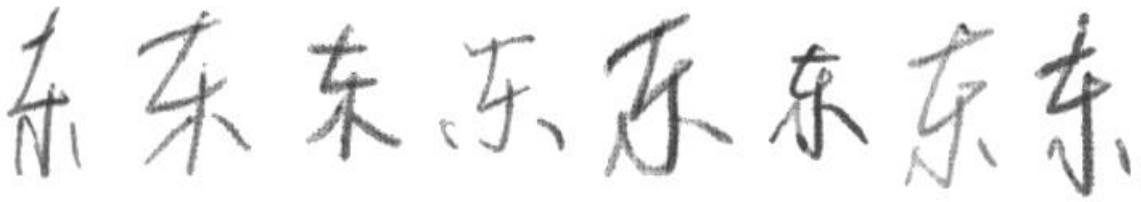


图 3-3 汉字“东”的部分训练数据

Fig3-3 Part of training data “东”

表 3-1 CASIA-HWDB1.1 数据集信息

Tab3-1 The information of CASIA-HWDB1.1 dataset

数据集	类别数目	样本数目		贡献人数	
		训练集	测试集	训练集	测试集
CASIA-HWDB1.1	3755	897758	223991	240	60

完整的 CASIA-HWDB1.1 数据集总共包含 1121749 张图片，即使是性能最佳的 GPU，完整训练一轮也要花费好几个小时甚至好几天。受限于计算资源，本文所使用的数据集仅仅是 CASIA-HWDB1.1 中的一个子集，其中包含 500 个不同类别的手写汉字，每个类别包含大约 240 张训练数据和 60 张测试数据。本文使用的手写汉字

数据集相关信息如表 3-2 所示。

表 3-2 本文所使用的数据集信息

Tab3-2 Dataset information used in this article

数据集	类别数目	样本数目		贡献人数	
		训练集	测试集	训练集	测试集
部分 HWDB1.1	500	120000	30000	240	60

3.2.2 数据预处理

（1）格式转换

CASIA-HWDB1.1 发布的数据为二进制格式的示例以及标签，因此预处理的第一步是将二进制格式转化为图像的格式，本文选取.png 的格式对图像进行编码和存储。转换后的图像的为像素值 255 的白色背景和 255 个灰度级（0-254）的前景像素。

（2）增强图像对比度

原数据集由 300 个不同的人贡献而成，由于采集光线和采集设备的不同，会造成不同的图像出现亮度不同、笔画厚重程度也不一样的情况。因此需要对原始的数据集图像进行对比度的增强。常用的对比度增强的方法有对比度拉伸法，采用线性函数的形式对原始图像的灰度值进行变换，以将重要的部分突出出来。

（3）尺寸归一化

CASIA-HWDB1.1 数据集中，所有的样本数据都不是同一个尺寸的。CNN 结构往往需要标准尺寸的输入图像。参考于 MNIST 手写数字数据集里面28×28的尺寸，本文将所用到的数据集扩大一倍，归一化为56×56规格的图像，同时在图像的外围添加了额外的四个空白像素，使得最终的图像尺寸为64×64。图 3-4 展示了图像对比度增强和尺寸归一化之后的效果图。

图 3-4 对比度增强和尺寸归一化的效果图

Fig3-4 Contrast enhancement and Size normalization

3.3 模型结构

CNN 与传统脱机手写汉字识别方法最大的不同在于可以实现端到端的识别过程。得益于众多研究人员的潜心研究，诞生了诸如 LeNet、AlexNet、VGG、GoogLeNet 等优秀的 CNN 模型。考虑到 LeNet 模型较为简单，特征提取能力较弱，本文分别基于 AlexNet、VGG 和 GoogLeNet，构建并改良了相应的模型，以进行手写汉字的识别。

（1）AlexNet-HCCR

第一个模型 Model_1 是基于 AlexNet 的网络结构。AlexNet 是第一个现代的深度 CNN 结构，2012 年，在 ImageNet 大赛中以较大的优势夺得了冠军。该网络模型使用了较多现代的卷积神经网络方法，比如使用了激活函数 ReLU，使用了 Dropout 技巧来减缓过拟合现象，使用 GPU 来进行并行计算等等。原始的 AlexNet 结构如图 3-5 所示。

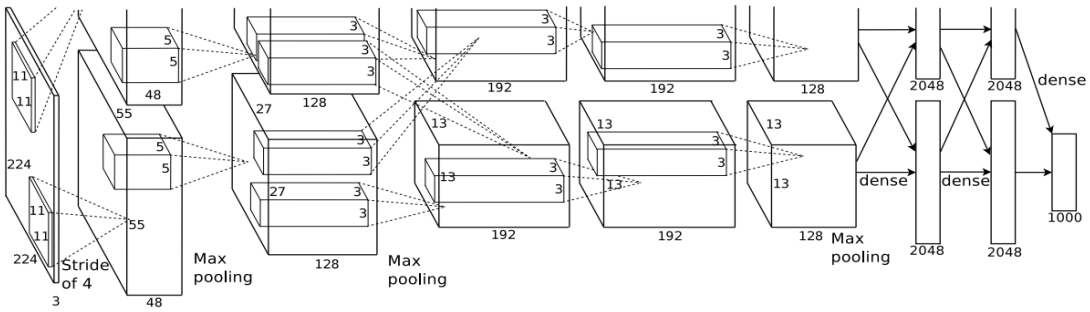


图 3-5 AlexNet 网络结构[20]

Fig3-5 The structure of AlexNet

AlexNet 处理的是$224 \times 224 \times 3$的 RGB 图像，模型总共包含 8 层变换，其中有五个卷积层、两个全连接隐藏层以及一个全连接输出层。因为 ImageNet 比赛上图像中物体占用更多的像素，AlexNet 第一个卷积层的卷积核大小为11×11，卷积核的数目为 96，第二个卷积层的卷积核尺寸为5×5，卷积核的数目为 256，后三个的卷积层的卷积核尺寸全都采用3×3的结构，卷积核的数量分别为 384、384 和 256。在第一、第二和第五个卷积层之后，AlexNet 都使用了池化区域为3×3，步幅为 2 的最大池化层。紧接着最后一个卷积层的是两个神经元个数为 4096 的全连接隐藏层，以及输出数目为 1000 的全连接输出层。

考虑到本文所使用的数据大小为$64 \times 64 \times 1$的黑白图像，基于 AlexNet，本文构建的第一个模型 Model_1 的第一个卷积层的卷积核大小为5×5，该层卷积核的数目为 96，卷积层后接上池化区域为2×2、步长为 2 的最大池化层；第二个卷积层的卷积核大小为3×3，卷积核数目为 256，该卷积层后连接着一个最大池化层。池化区域和步长与第一个最大池化层一样。第三、第四和第五个卷积层连在一起，每层的卷积核大小都是3×3，卷积核的数目依次是 384、384、256。第五层后面连接了一个最大池化层，最大池化层后连接着三个全连接层，神经元个数分别是 1024、1024 和 500。整个模型结构如图 3-6 所示。

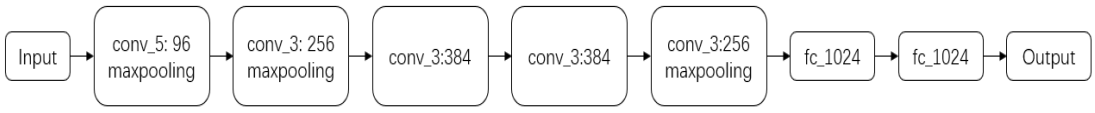


图 3-6 Model_1 的网络结构

Fig3-6 The network structure of Model_1

（2）VGG-HCCR

与 AlexNet 不同，VGG 采用连续数个3×3的卷积核来代替较大的卷积核。VGG 的论文显示，在卷积神经网络中，对于给定的感受野，采用堆叠的小卷积核通常优于偏大的卷积核，这是因为可以通过增加网络的深度来保证模型可以学习到更加复杂的模式，而且小卷积核的参数较少，学习的代价也会比较小。与此同时，多个小的卷积核在功能上可以代替较大的卷积核。例如，一个14×14的特征图，使用一个7×7大小的卷积核进行卷积计算，可以得到一个8×8的输出特征图，类似的，使用三个3×3的卷积核与14×14的特征图进行运算的时候，也同样可以得到一个8×8

的输出特征图。使用三个小的卷积核还能够增加模型的非线性关系，有利于提高模型的泛化能力。

在 VGG 结构中，整个模型主要由卷积层模块和全连接层模块堆叠而成，其中卷积层模块由多个 VGG 块堆叠而成，每个 VGG 块使用连续多个相同的填充（padding）为 1、卷积核大小为3×3的卷积层后接上一个步幅为 2、池化区域为2×2的最大池化层。VGG 块的结构如图 3-7 所示，其中n表示卷积核的数目。

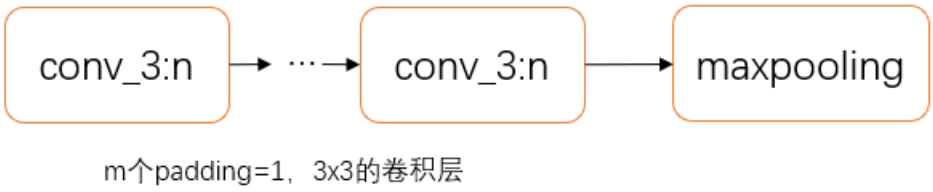


图 3-7 VGG 块的结构

Fig3-7 The structure of VGG block

为了探讨模型的深度与手写汉字识别准确率的关系，以及探讨卷积层中卷积核的数目对手写汉字识别准确率的影响，本文基于 VGG 的网络结构，设计了三个不同深度和宽度的模型，分别命名为 Model_2、Model_3 和 Model_4。Model_2、Model_3 以及 Model_4 的卷积模块都是由 VGG 块堆叠而成，不同的是彼此之间的卷积层数和卷积核的数目不一样。三个模型的结构如表 3-3 所示。

Model_2 是一个总共包含有九层的模型，其中六个卷积层和三个全连接层，在所有的卷积层中，卷积核统一设计为3×3的尺寸，六层卷积层中卷积核的数量依次为：64、128、256、256、512、512。在第一、第二、第四、第六层卷积层的之后，连接了相同结构步幅为 2、池化区域为2×2的最大池化层。其后三层全连接层的输出个数分别为 1024、1024、500。

与 Model_2 相比，Model_3 是一个总共包含 11 层的模型，其中有八个卷积层和三个全连接层。Model_3 中三层全连接层与 Model_2 的三层全连接层结构一样，八层的卷积层的卷积核大小都统一设计为3×3的结构，每层卷积核的个数依次为：64、64、128、128、256、256、512、512。在第二、四、六和第八层卷积层之后，紧接着的是与 Model_2 相同的最大池化层。

Model_4 的整体结构与 Model_3 的结构基本相似，不同之处在于每层卷积层的卷积核个数不一样。与 Model_3 相比，Model_4 每层卷积层的卷积核个数都略大一

些。Model_4 八层卷积层的卷积个数分别为 80、80、160、160、320、320、640、640。

表 3-3 Model_2 到 Model_4 的结构图

Tab3-3 The structure from Model_2 to Model_4

Model_2	Model_3	Model_4
9 层	11 层	11 层
输入数据（64 × 64 × 1的灰度图像）		
conv_3: 64	conv_3: 64 conv_3: 64	conv_3: 80 conv_3: 80
maxpooling		
conv_3: 128	conv_3: 128 conv_3: 128	conv_3: 160 conv_3: 160
maxpooling		
conv_3: 256 conv_3: 256	conv_3: 256 conv_3: 256	conv_3: 320 conv_3: 320
maxpooling		
conv_3: 512 conv_3: 512	conv_3: 512 conv_3: 512	conv_3: 640 conv_3: 640
maxpooling		
FC: 1024		
FC: 1024		
FC: 500		
Softmax		

（3）GoogLeNet-HCCR

在 AlexNet 和 VGG 网络模型中，每一层卷积核大小通常是固定的，而在 CNN 模型当中，如何设计卷积核的大小是一个非常关键的问题。受《Network In Network》[41]的影响，GoogLeNet 提出了一种叫做 Inception 模块的结构，使得在一个卷积层中包含多个不同大小的卷积操作。在 Inception 模块中，一个卷积层内同时具有 1 × 1、3 × 3、5 × 5的卷积核，通过合适的填充（padding），输入特征图在不同大小

的卷积核操作下，能够保持输出单个特征图的尺寸一致。而 Inception 模块则将多个不同大小卷积核得到的特征图拼接在一起，作为整个 Inception 模块的输出。Inception 模块如图 3-8 所示。

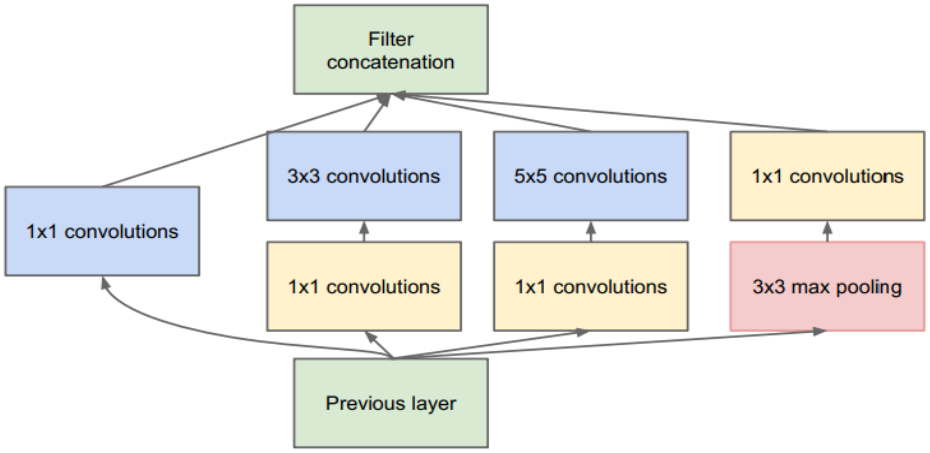


图 3-8 Inception 模块[22]

Fig3-8 The Inception Module

由图 3-8 可知，Inception 模块包含四条并行的线路，分别为1×1的卷积、3×3的卷积、5×5的卷积和3×3的最大池化层，用于提取不同空间尺寸的信息。在3×3、5×5的卷积层之前和3×3最大池化层之后，使用1×1的卷积操作来减少通道的个数，以降低模型的复杂度。

原始的 GoogLeNet 包含 9 个 Inception 模块。考虑到本文所使用数据集的实际情况以及受限的计算资源，本文基于 GooLeNet 设计的 Model_5 的主体由 4 个 Inception 模块组成。主体结构如图 3-9 所示。

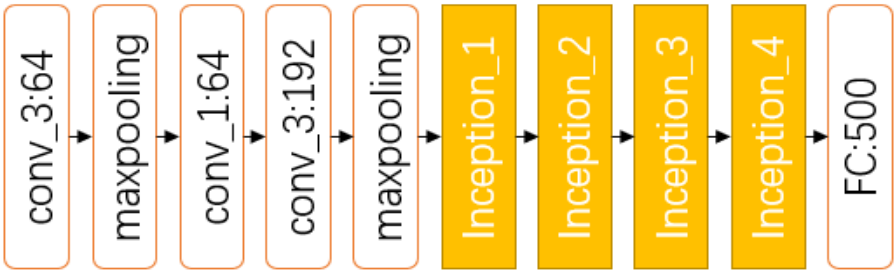


图 3-9 Model_5 的主体结构

Fig3-9 The main structure of Model_5

在 Model_5 中，第一个 Inception 模块中第一条线路1×1的通道数为 64，第二

条线路中1×1和3×3的通道数分别为 96 和 128，第三条线路中1×1和5×5的通道数分别为 16 和 32，第四条线路1×1的通道个数为 32。第二个 Inception 模块中第一条线路1×1的通道数被设计为 128，第二条线路中1×1和3×3的通道个数为 128 和 192，第三条线路中1×1和5×5的通道个数为 32 和 96，第四条线路中1×1的通道个数为 64。第 3 个 Inception 模块中，第一条线路中 1×1 的通道个数为 256，第二条线路中 1×1、3×3 的通道个数为 112 和 256，第三条线路中 1×1、5×5 的通道个数为 32 和 128，第四条线路中1×1的通道个数为 64。第 4 个 Inception 模块中，第一条线路中1×1的通道个数为 384，第二条线路中1×1、3×3的通道个数为 64 和 96，第三条线路中1×1、5×5的通道个数为 96 和 96，第四条线路中1×1的通道个数为 128。四个 Inception 模块中的最大池化层都设计为区域为3×3、步长为 1。

在神经网络当中，为了给模型添加非线性关系，往往在神经元的输出之前加入激活函数。较为常用的激活函数有 sigmoid 函数和 tanh 函数，但是随着网络模型的深度越来越大，sigmoid 函数和 tanh 函数在进行反向传播的时候，经常会出现梯度消失等问题。Alex Krizhevsky 在 AlexNet 中引入了 ReLU 激活函数，在一定程度上减缓了神经网络中梯度消失等问题，加速模型的收敛速度。与前两个函数相比，ReLU 函数具有很好的稀疏性，使得大约50%的神经元会处于激活的状态。这与生物神经网络中神经元的兴奋状态很相似。此外，由于 ReLU 函数更为简单，在计算上也更为高效。如图 3-10 所示，ReLU 函数在模型的训练过程中可以大大的提高模型收敛的速度。因此，本文设计的所有模型当中，隐藏层所有神经元的激活函数都使用 ReLU。

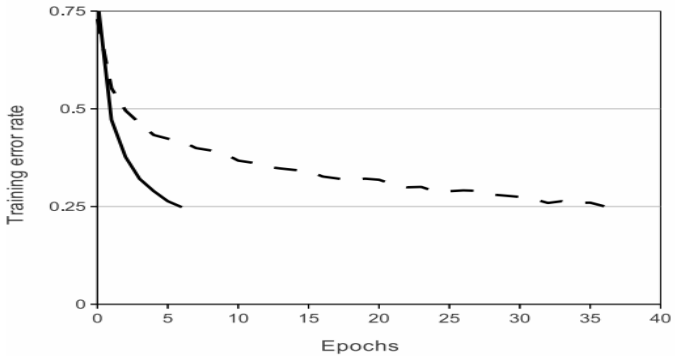


图 3-10 ReLU 与 sigmoid 函数的对比[20]

Fig3-10 The comparison between ReLU and sigmoid functions

3.4 实验结果与分析

3.4.1 实验环境

本文的实验环境是 TESRA 超算网络中心，由于平台信息的不公开，用户无法查看 CPU、GPU 和 RAM 等具体参数。使用的深度学习框架为谷歌开源的 TensorFlow[42]，版本为 1.14，编程语言为 Python3.6，数据集为本文选取的 500 类部分 HWDB1.1。

3.4.2 模型的训练

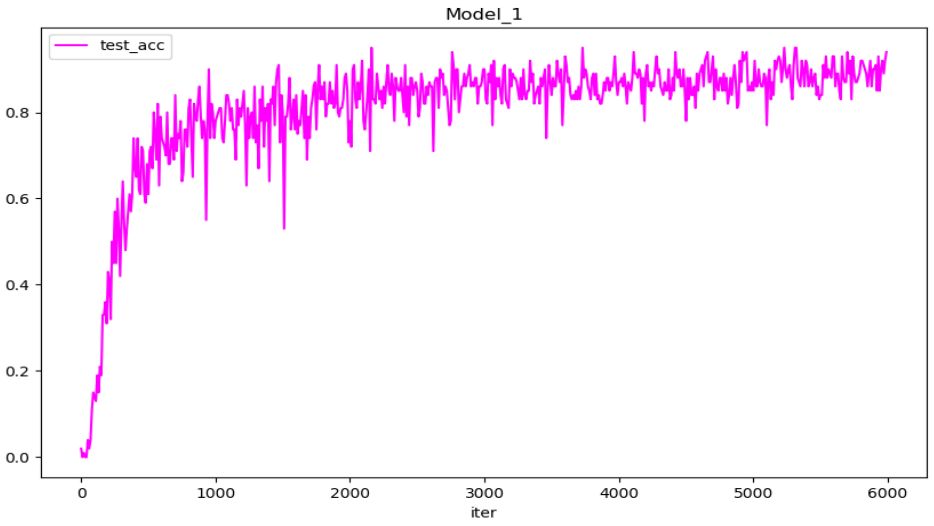
对于 3.3 节定义的五个 CNN 模型，在最后一层输出层之后，都进行 softmax 函数处理，所定义的损失函数均为交叉熵损失函数。模型的训练优化算法是 Adam 算法，最初的学习率为 0.002，学习率衰减指数为 0.97，每经过 1000 次的迭代，学习率衰减一次。batch_size 设置为 100，在训练过程中，同时加入了 L2 正则化和 dropout 方法来减缓过拟合的情况，dropout 的概率设计为50%。

为了加快卷积神经网络的训练速度，本文使用有监督的预训练方法，首先对 CASIA_HWDB1.1 数据集中的随机 10 个类别的汉字作为训练集，用于训练 3.3 节中定义的五个模型，其中每个模型最后一层神经元的个数定义为 10，当整个模型达到一个较好的识别准确率的时候，将这个模型的参数作为新网络的初始化参数，而不是使用完全随机的初始化参数。在网络参数初始化完成后，使用本文所用到的 500 类别的数据集，模型的输出神经元个数也相应地改成 500。这种预训练的方法在深度学习领域叫做微调（fine tuning）[43]。因为对于所有的图像来说，完整的图像都是由相似的边缘、纹理、形状等通过组合而成，而在 CNN 当中，处于较浅层的卷积层所提取的特征往往都是图像的基本纹理和一些相似的边角特征，这些特征对于较大的数据集也行之有效。

3.4.3 实验的结果

对于 500 个类别数据集，本次实验分别使用了 3.3 节设计的 5 个网络模型来进行训练和测试。

（1）模型一对应的结果为：



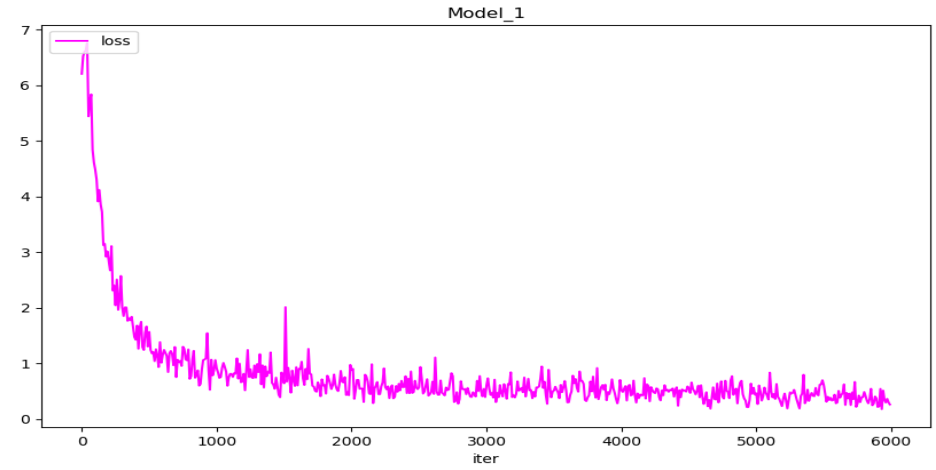


图 3-11 Model_1 测试识别准确率和 loss 函数变化图

Fig3-11 The accuracy and loss function curve of Model_1

如图 3-11 所示，具有 5 个卷积层和 3 个全连接层的 Model_1 在将近 2000 次的迭代之后，Model_1 接近收敛，整个模型在测试数据集中的识别准确率约为 89.5%。

（2）模型二对应的实验结果为：

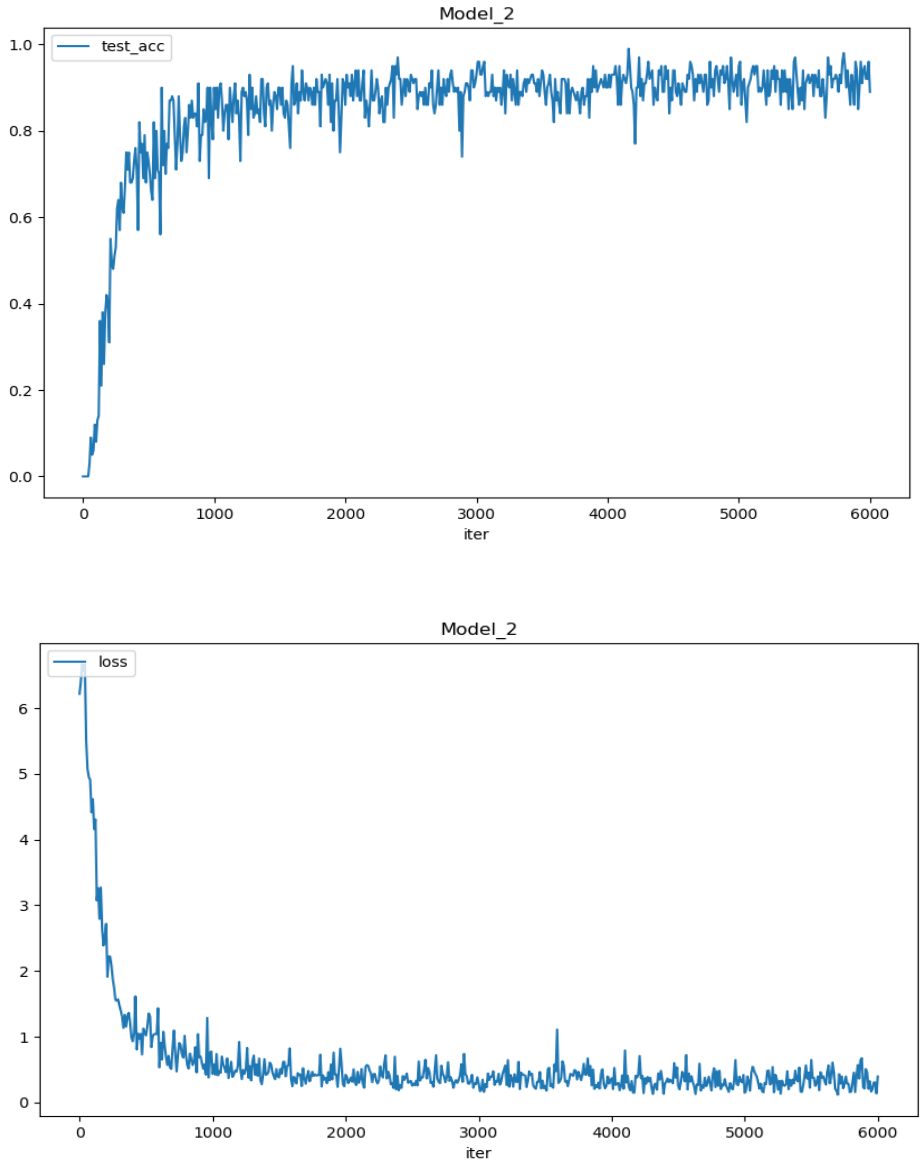


图 3-12 Model_2 的识别准确率和 loss 函数变化图

Fig3-12 The accuracy and loss function curve of Model_2

图 3-12 显示了，具有 6 个卷积层和 3 个全连接层的 Model_2 在迭代了将近 1600 次之后，模型达到了收敛的状态，整个模型的识别准确率为92.7%。注意到，虽然 Model_2 的层数比 Model_1 的层数要多，但是由于 Model_2 中使用的卷积核大小均为3×3的结构，参数要比 Model_1 中5×5的卷积核要少，因此该模型的收敛速度比 Model_1 的收敛速度更快，识别准确率也比 Model_1 要高。

（3）模型三对应的实验结果为：

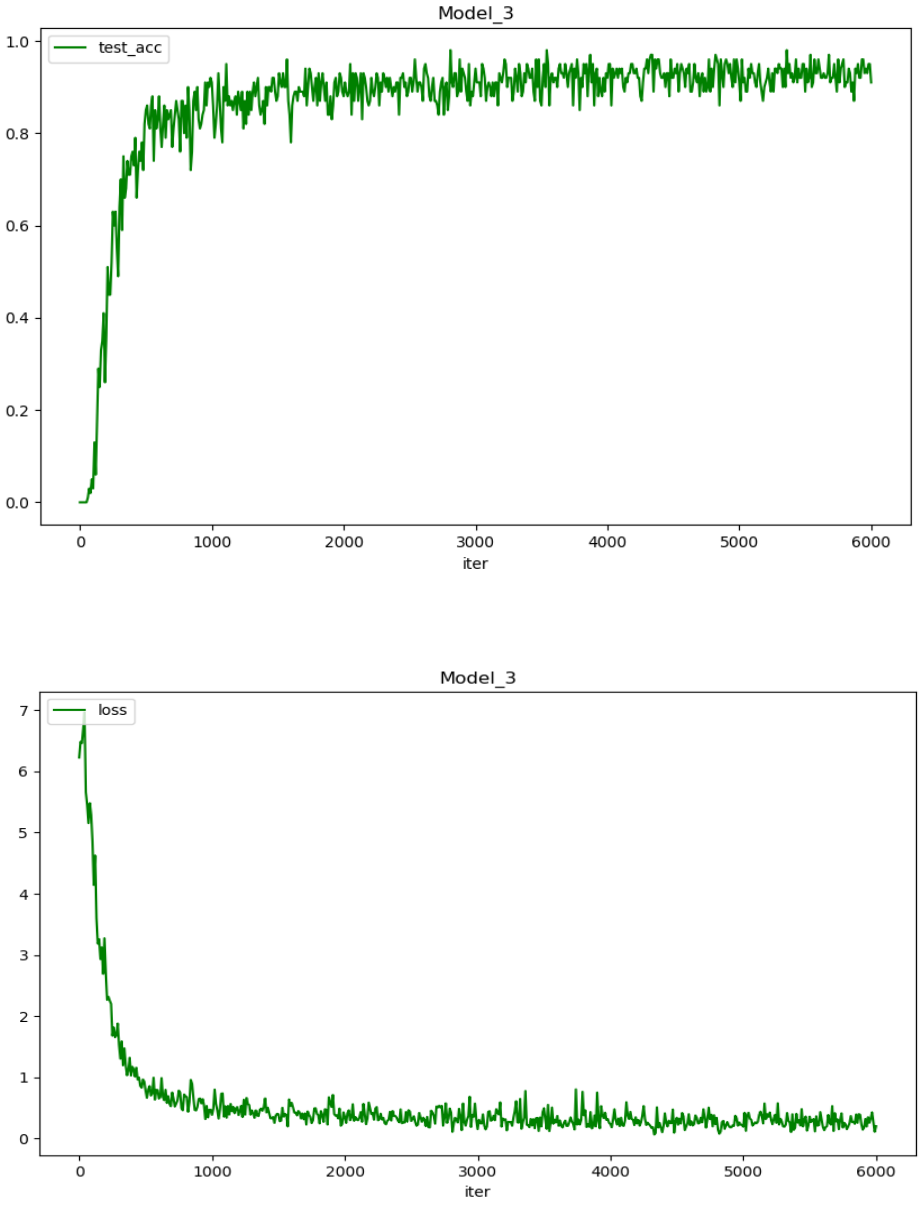
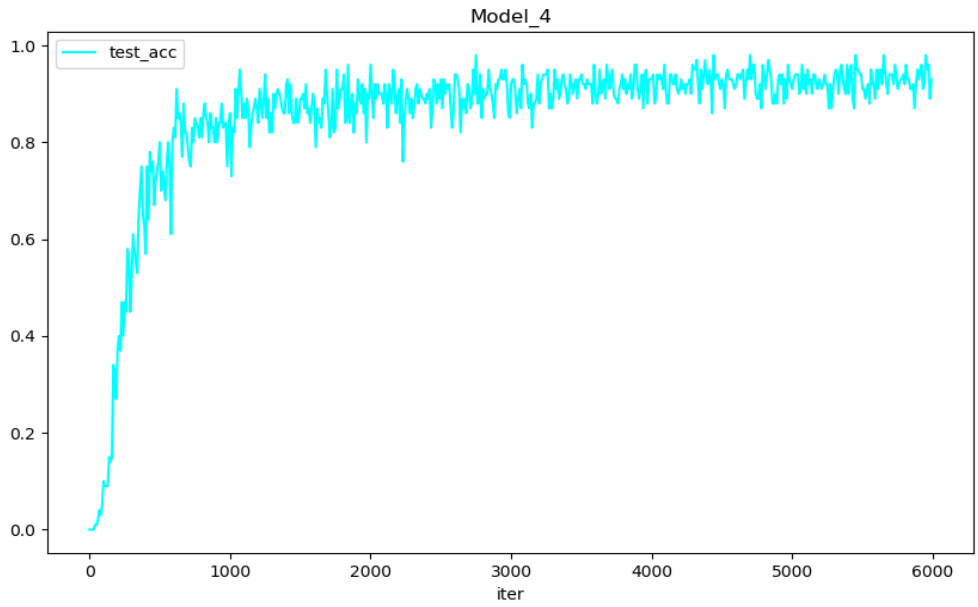


图 3-13 Model_3 的识别准确率和 loss 函数变化图

Fig3-13 The accuracy and loss function curve of Model_3

图 3-13 显示了，具有 8 个卷积层和 3 个全连接层的 Model_3，在将近 1800 次的迭代之后，模型达到了收敛，此时模型在测试数据集上的识别准确率约为94.3%。虽然迭代速度较 Model_2 慢了一些，但是多加了两层卷积层的 Model_3，识别准确率提高了1.6%。

（4）模型四对应的实验结果为：



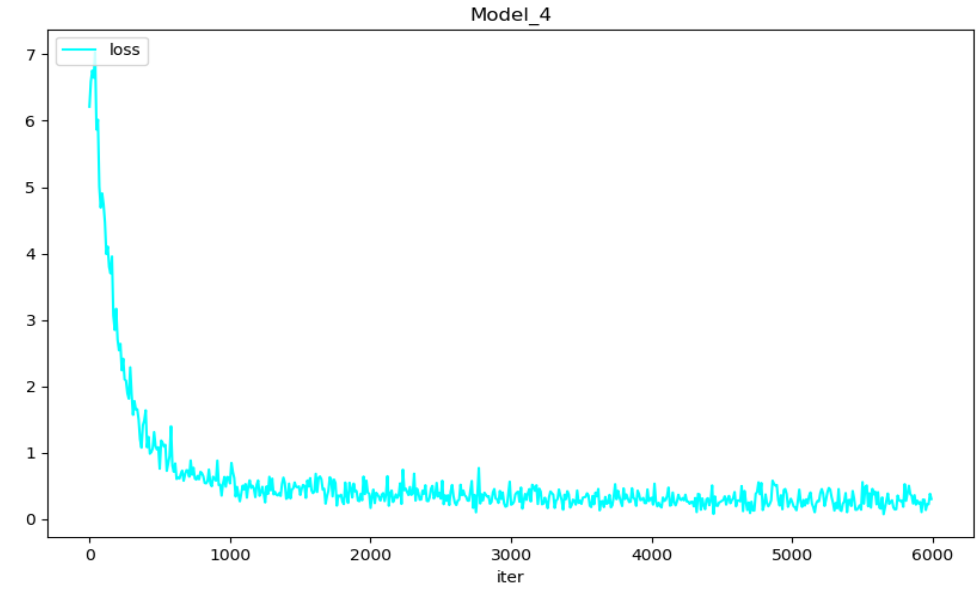


图 3-14 Model_4 的识别准确率和 loss 函数变化图

Fig3-14 The accuracy and loss function curve of Model_4

如图 3-14 所示，具有 8 层卷积层和 3 层全连接层的 Model_4，在经过接近 2000 次迭代之后，模型达到了收敛的状态。此时，模型的识别准确率约为94.5%。Model_4 中卷积层的层数和全连接层的层数与 Model_3 中的一样，不同的是，Model_4 每层的卷积核的数量都多于 Model_3 中卷积核的数量，取得的效果是识别准确率提高了 0.2%，但模型的准确率的有比较大的抖动，收敛速度偏慢。实验结果显示，虽然卷

积核的数量能够提高 Model_3 的识别准确率，但是当卷积核的个数达到一定程度之后，增大卷积核的数量对模型准确率提高的幅度并不大，反而会增多模型的参数，使得模型更加难以训练，模型的收敛速度也会变慢。

（5）模型五对应的实验结果为：

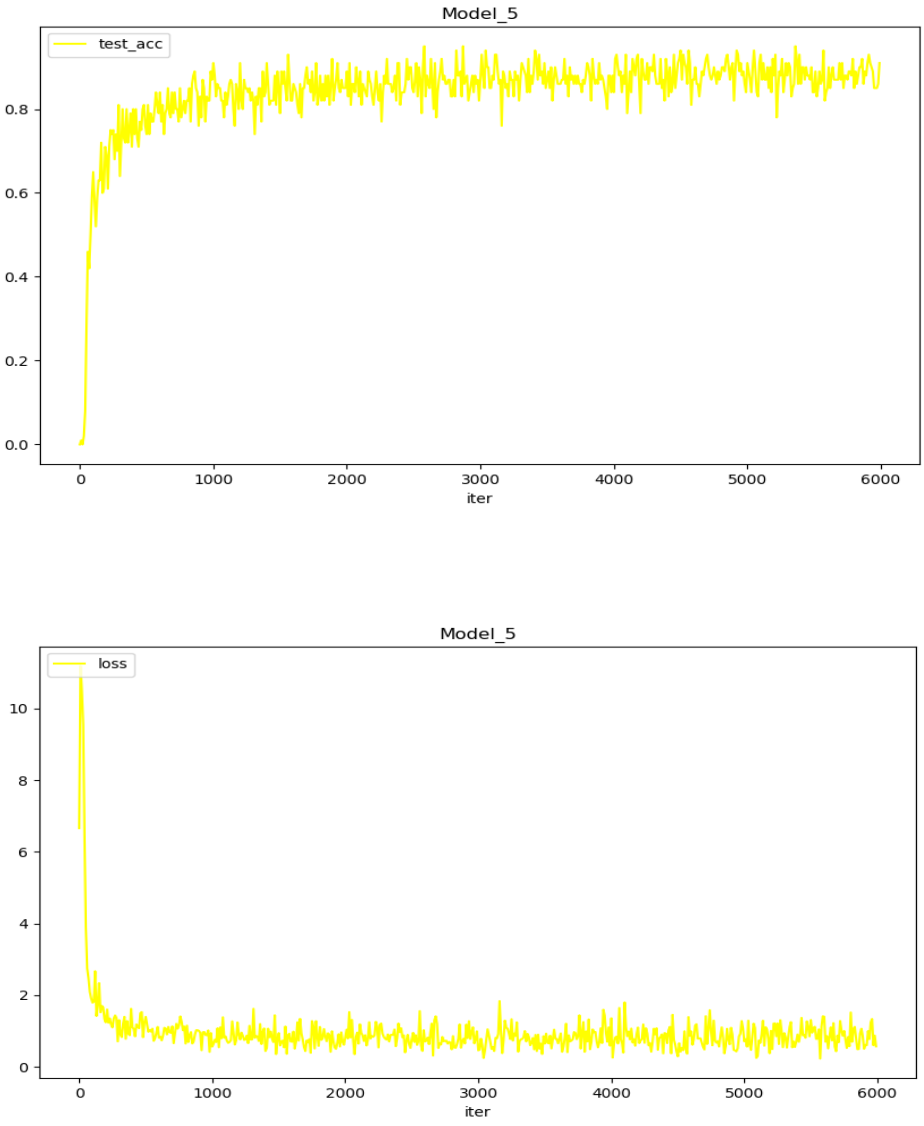


图 3-15 Model_5 的识别准确率和 loss 函数变化图

Fig3-15 The accuracy and loss function curve of Model_5

如图 3-15 所示，具有 4 个 Inception 模块的 Model_5 在经过解决 1000 次的迭代之后，模型达到了收敛的状态。此时，模型的准确率约为89.8%。

五个模型对应的识别准确率如表 3-4 所示。

表 3-4 五个模型在测试集中的识别准确率

Tab3-4 Recognition accuracy corresponding to five models

模型	识别准确率
Model_1	89.5%
Model_2	92.7%
Model_3	94.3%
Model_4	94.5%
Model_5	89.8%

3.4.4 结果的分析

在深度学习中，通常情况下，模型的表达能力与网络的深度成正比。然而，模型的参数数量与模型的深度成正相关，随着模型的深度越深，参数也会成倍增加。在缺乏足够的训练数据集和足够的计算资源的时候，深度较深的网络模型通常难以拟合所需要的分类器，很容易造成过拟合的现象。因此，需要根据所使用的数据集的实际情况，选择恰当的网络模型。

2.3 节中基于 AlexNet、VGG、GoogLeNet 设计的五个不同结构的 CNN 模型，其中效果最好的是三个基于 VGG 的 Model_2、Model_3、Model_4。对比 Model_2 和 Model_3 发现，Model_2 具有 6 层的卷积层，Model_3 具有 8 层的卷积层，在本文的实验中，Model_3 比 Model_2 的识别准确率高了1.6%。这个结果表明，在设计恰当的网络结构和良好的训练方法当中，较深的卷积神经网络模型的识别准确率往往更高。

与 Model_3 类似，Model_4 也具有 8 个的卷积层和 3 个的全连接层。不同的是，Model_4 的卷积层中的卷积核的个数比 Model_3 的要多得多，相应的，识别准确率也比 Model_3 要高0.2%。但是，如图 3-13 和 3-14 所示，Model_4 在训练时间和收敛速度上均不如 Model_3。这表明了卷积核的数量，在达到某个阈值之后，单纯增加卷积核的数量对识别准确率并无较大的裨益，反而会增加训练的难度。

3.5 本章小结

本章首先介绍了传统手写汉字识别和基于 CNN 模型的手写汉字识别在识别流程上的区别，其次介绍了图像识别领域中常用的数据预处理方法，包括图像对比度

的增强和尺寸的归一化。再次基于 500 类别的 CASIA-HWDB1.1 数据集设计了五个具有不同深度不同结构的 CNN 模型，并使用 CASIA-HWDB1.1 部分数据集对这五个 CNN 模型进行训练和测试。最后给出了模型的训练结果以及对结果的分析。

第四章 特征提取融合卷积神经网络的识别方法

4.1 引言

CNN 集特征提取和特征分类于一身，在很多计算机视觉领域，往往可以设计为一个端到端的结构，将原始的图像作为模型结构的输入进行训练和分类。然而，卷积神经网络到目前为止，还无法用严格的数学定义来证明其工作原理，往往都被当作一个黑箱子来使用。因此，在手写汉字识别中，CNN 很有可能没有办法学习到一些特定领域的有效先验信息[44]。因此在本章中，本文将研究传统的特征提取方法与 CNN 互相融合的识别方法。

传统的图像识别中，特征的提取在很多时候比分类器更重要，特征提取算法的好坏，往往直接影响了整个模型的识别准确率。对于汉字而言，特征提取的算法主要有两种[45]：结构特征提取和统计特征提取。结构特征提取一般基于文字的笔划和形状，根据人类对文字识别的惯性，对文字的整体结构进行特征的提取。统计特征的提取通常是通过某种数学的变换，从原始的空间映射到另一个空间中，然后统计新的数据空间中所得到的某些特征。图像的结构特征的提取较为困难，而统计特征的提取往往抗干扰性较好，因此本文选择使用统计特征提取。例如 Gabor 特征提取[46]和 HOG 特征提取[47]。

Gabor 变换广泛应用于图像预处理方面，研究表明，Gabor 变换在手写汉字识别领域具有较好的效果[48]。方向梯度直方图 HOG（Histogram of oriented gradient）在计算机视觉领域一直被认为是一个很好的图像特征。在本章将会研究 Gabor 变换以及 HOG 对手写汉字识别准确率的影响。

4.2 特征提取

4.2.1 Gabor 特征

Gabor 滤波器在图像处理等领域有着广泛的应用。Gabor 滤波器的频率和方向与动物的视觉系统较为相似，对图像的边缘敏感，能够提供良好的方向选择和尺度选择，特别适用于图像纹理的表达和分离。对于手写汉字而言，识别汉字的过程可以近似于识别整个汉字的纹理，因此手写汉字识别与 Gabor 滤波器之间具有一定的联系。

Gabor 变换可看作是一种特殊的傅里叶变换。在信号处理领域，傅里叶变换可以将时域中的信号转换到频域，并在频域中对信号进行相应的处理，提取时域中不容易提取到的特征。而 Gabor 变换是一种在特定时间窗口内的傅里叶变换，可以通过在空间域内不同尺寸、不同方向上提取相应的特征。在空间域中，二维的 Gabor 变换可以看作是一个由高斯核函数调制的正弦平面波，可以在空间域内的不同方向和不同大小对相关的特征进行提取。多方向的 Gabor 变换可定义为：

$$F(x,y;k,\vartheta_k) = I(x,y) * G(x,y;k,\vartheta_k) \tag{4.1}$$

其中，$I(x,y)$代表着输入的图像，$G(x,y;k,\vartheta_k)$代表 Gabor 滤波器。Gabor 滤波器的详情可表示为：

$$G_1(x,y) = \frac{k^2}{\sigma^2}\exp\left[-\frac{k^2(x^2+y^2)}{2\sigma^2}\right] \tag{4.2}$$

令$R = kxcos\vartheta_k + kysin\vartheta_k$，则 Gabor 变换的复数形式可表现为：

$$G(x,y;k,\vartheta_k) = G_1(x,y)\left[\cos R - \exp\left(-\frac{\sigma^2}{2}\right)\right] + iG_1(x,y)sinR \tag{4.3}$$

其中，$k = \frac{2\pi}{l}$，$\sigma = \pi$，$\vartheta_k = \frac{\pi k}{M}$（$k = 0,1,\dots,M-1$）。$l$和$\vartheta_k$分别表示波长和 Gabor 滤波器的方向。

4.2.2 方向梯度直方图

方向梯度直方图（Histogram of Oriented Gridients，HOG）是目前计算机视觉和模式识别领域中非常常见的一种描述图像局部纹理的特征。在一张图像当中，局部物体的形状和轮廓可以被梯度或者边缘的方向密度很好地描述，这是因为较大的梯度往往存在于某物体的边缘，而统计梯度的信息，可以较好地描述出图像中物体的大致轮廓[49]。HOG 就是通过计算和统计图像中某个区域的梯度方向直方图来形成该图像的特征。HOG 特征最初由法国研究者 Dalal 在 2005 年的 CVPR 上提出，该算法通过计算和统计图像某个区域的梯度方向直方图来形成相应的特征，从而结合 SVM 来进行行人检测[47]。

HOG 特征提取的步骤如下：

（1）图像预处理。主要包括伽马校正和灰度化处理。伽马校正用于提高图像的对比度，较少光线对实验的影响。灰度化是为了计算简单，将彩色的 RGB 图片变成

黑白的灰度图。在本文的数据集中，可以忽略这一步骤。

（2）计算每个像素点在该位置的梯度，包括方向和大小。对于图片中某个位置的像素点$H(x,y)$，该点x轴方向梯度的大小和y轴方向梯度的大小分别定义为：

$$g_x = \sqrt{(\mathrm{H(x+1,y)} - \mathrm{H(x-1,y)})^2} \tag{4.4}$$

$$g_y = \sqrt{(\mathrm{H(x,y+1)} - \mathrm{H(x,y-1)})^2} \tag{4.5}$$

其中，$H(\mathrm{x-1,y})$和$H(\mathrm{x+1,y})$分别代表该像素点水平方向左右两侧的像素值，$H(\mathrm{x,y+1})$和$H(\mathrm{x,y-1})$分别表示当前像素点垂直方向上下两个像素值。总的梯度强度为：

$$g = \sqrt{g_x^2 + g_y^2} \tag{4.6}$$

同时，当前像素点的梯度方向可定义为：

$$\theta = arctan\frac{g_x}{g_y} \tag{4.7}$$

其中，梯度的方向往往取其绝对值，因此梯度的方向的取值范围是$[0\sim\pi]$。例如，对于图 4-1 中点 A 来说，该点的梯度大小和方向的计算表达式为：

$$g_x = \sqrt{(30-20)^2} = 10 \tag{4.8}$$

$$g_y = \sqrt{(32-64)^2} = 32 \tag{4.9}$$

$$g = \sqrt{10^2 + 32^2} \tag{4.10}$$

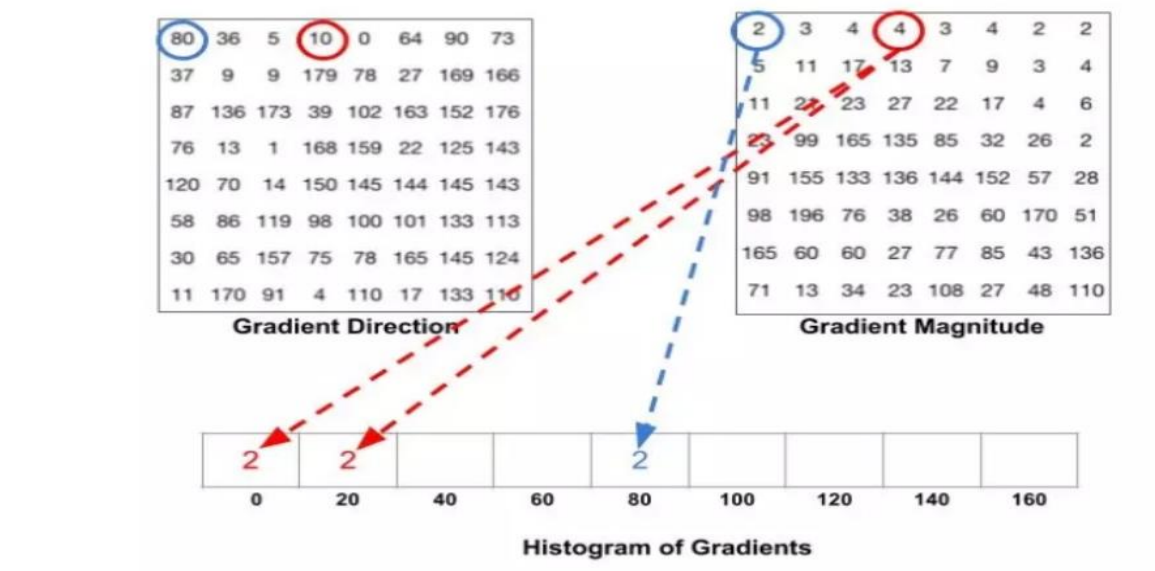
$$\theta = arctan\frac{10}{32} \tag{4.11}$$

	32	
20	A	30
	64	

图 4-1 像素点 A 梯度的计算

Fig4-1 Calculation of gradient of pixels A

（3）统计梯度直方图。在经过步骤 2 的计算之后，每个像素点都会有两个值：梯度的大小和梯度的方向。HOG 特征通常会在原图像中选择一个特定大小的区域 cell，用来统计梯度方向直方图。例如，常用的 cell 的大小设置为8×8。在一个8×8的 cell 里面，经过步骤 2 的计算，得到了一个梯度大小的矩阵和一个梯度方向的矩阵。因为梯度方向的取值为$[0\sim\pi]$，通常将梯度的方向分为 9 等分，每一等分用一个 bin 来表示，每个 bin 代表着 cell 中在该方向的梯度。HOG 根据 cell 中梯度方向矩阵中每个值，依次将梯度大小矩阵的值映射到 9 个等分中。映射完成后，便可得到该 cell 的梯度直方图，而得到的梯度直方图可以由一个 9 维向量表示。如图 4-2 所示，对于某个8×8的 cell 来说，得到了两个关于梯度的矩阵，每个矩阵中对应点的值为梯度的方向和梯度的大小，HOG 通过梯度的方向来统计处于某个方向的梯度值。图 4-2 的左上角的像素点对应的方向为80°，大小为 2，则将 2 映射到80°方向的直方图中。同理，对于第一行第四列梯度方向为10°大小为 4 的像素点，将像素点的大小按比例映射到对应的方向直方图中。当所有的像素点映射完毕后，便可得到了属于该 cell 的梯度特征直方图。梯度直方图如图 4-3 所示。



4-2 梯度映射过程

Fig4-2 Gradient mapping process

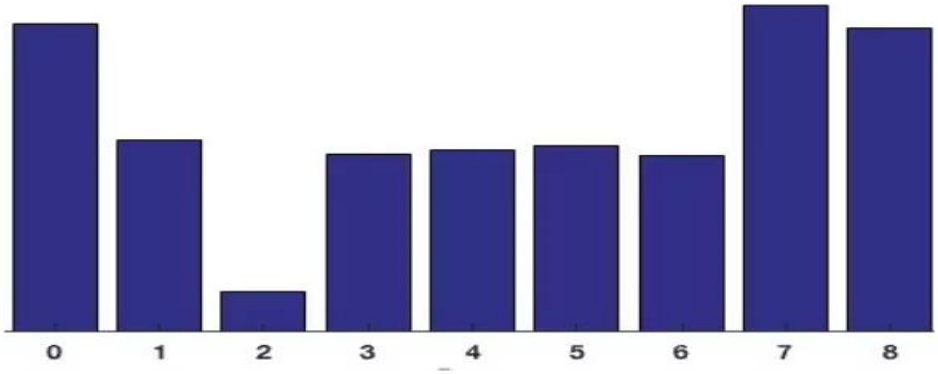


图 4-3 梯度直方图

Fig4-3 Histogram of Oriented Gridients

（4）对 block 进行归一化处理。一个 block 由多个 cell 组成，每个 cell 可以由一个向量表示。归一化的方法是让向量中每一个值与该向量的模长相除，目的是为了降低光照对实验的影响。

（5）HOG 特征的描述。将图像中所有 block 的 HOG 特征串联在一起，便组成了该图像对应的 HOG 特征描述。

4. 3 实验的优化与对比

在第三章中，构建了五个端到端的手写汉字识别的模型，每个模型直接对输入的图像进行特征提取和分类，都达到了较好的结果。本章将在原有的端到端的识别系统上，研究融合 Gabor 特征提取以及 HOG 特征提取对模型准确率的影响。同时探讨 dropout 法对实验性能的影响以及批量归一化算法对模型训练的加速效果。

对比 3.3 节中设计的五个模型，本章选择识别准确率较高且收敛速度较快的 Model_3，并在此基础上进行了适当的优化和改造，形成了一个 11 层的 CNN 模型，命名为 CNN_11。模型的整体结构如图 4-4 所示。本章实验将在此 CNN_11 的基础上，进一步探索上述的多个问题。

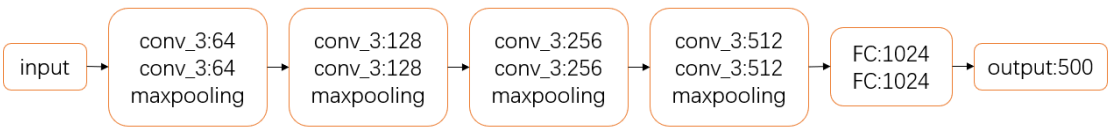


图 4-4 CNN_11 的整体模型结构

Fig4-4 The structure of CNN_11

4.3.1 统计特征对识别准确率的影响

统计特征融合 CNN 的识别方法，在流程上与 CNN 的端到端识别方法较为不同，首先需要对输入的训练图像进行 Gabor 特征和 HOG 特征的提取，然后将得到的特征图与原输入训练图像一起输入到 CNN 模型当中进行训练。方法的流程图如图 4-5 所示

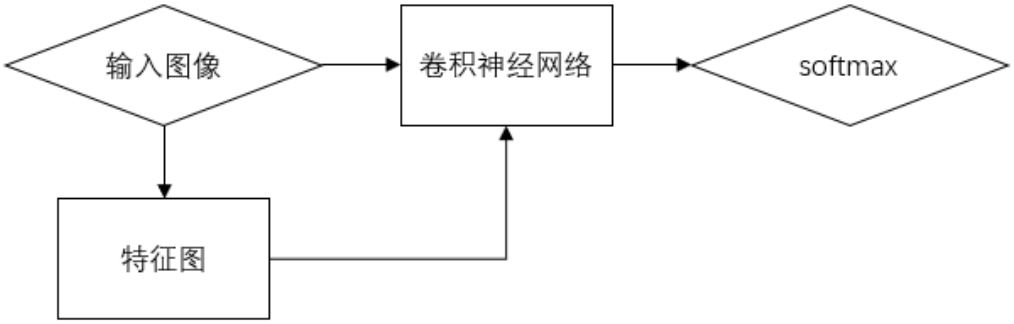


图 4-5 统计特征融合 CNN_11 的实验流程图

Fig4-5 The process of Statistical feature fusion CNN_11

（1）Gabor 特征提取

在 Gabor 滤波器当中，滤波器的方向和尺寸决定了所滤波的位置。在进行一定的实验之后，本文选择提取八个方向的 Gabor 特征，分别代表着0、$\frac{\pi}{8}$、$\frac{\pi}{4}$、$\frac{3\pi}{8}$、$\frac{\pi}{2}$、$\frac{5\pi}{8}$、$\frac{3\pi}{4}$、$\frac{7\pi}{8}$八个方向。而 Gabor 滤波核的尺寸设置为9×9，波长设置为$4\sqrt{4}$。对输入图像进行 Gabor 特征提取之后的特征图如图 4-6 所示。

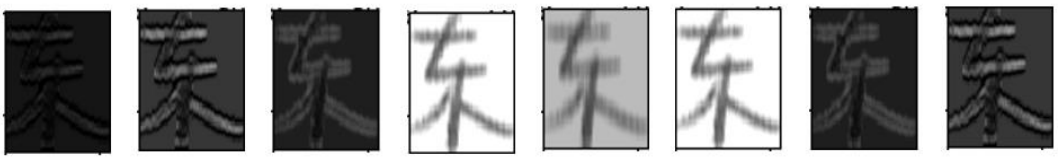


图 4-6 八方向 Gabor 特征图

Fig4-6 8-direction Gabor feature map

（2）HOG 特征提取

与 Gabor 特征提取类似，在模型的训练之前，首先对输入数据进行 HOG 特征的提取。考虑到本文所使用数据集的情况，HOG 特征提取的参数与 HOG 原论文的

建议值并不一样。在进行一定的实验之后发现，梯度的方向分为 8 个等分和 9 个等分所提取的 HOG 特征都相对理想。实验中发现，对 HOG 特征提取结果影响较大的参数是 cell 的大小。在 HOG 的论文中，作者在行人检测上做了相应的实验后，发现 cell 的尺寸为8×8是最好的结果。然而，对于本文的数据集而言，64×64图像的尺寸相对较小，8×8大小的 cell 所提取出来的 HOG 特征并不理想。经过多组的实验发现，cell 的大小为2×2时，所提取出来的 HOG 特征最能描述原图像的特征信息。图 4-7 展示了 cell 在2×2、4×4、6×6 和 8×8所对应的 HOG 特征图。

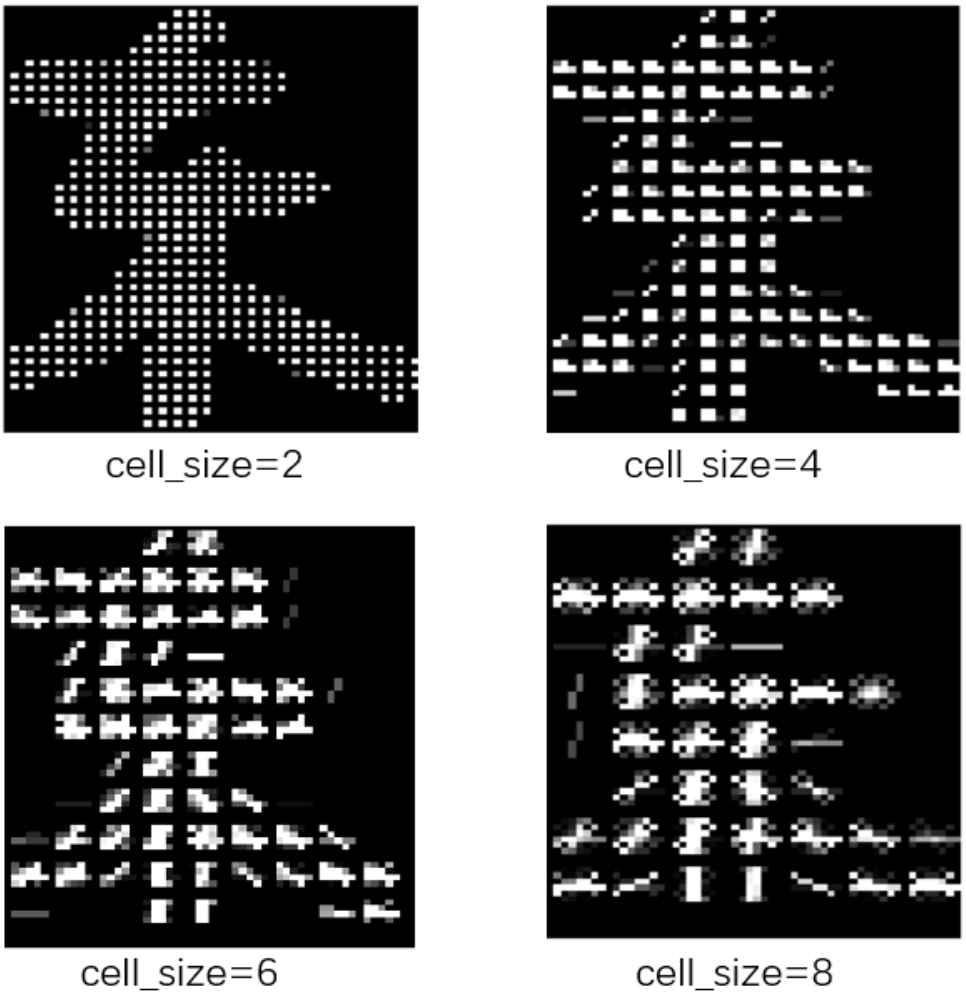


图 4-7 不同 cell 尺寸提取的 HOG 特征图

Fig4-7 HOG feature map from different cell-size

在本文中，HOG 特征提取的参数依次为：cell 的尺寸为2×2，梯度的直方图通道数目为 8，每个 block 所包含的 cell 大小为2×2。

对于提取得到的 Gabor 特征图以及 HOG 特征图，将与原输入图像合并一起输

入到 CNN_11 中进行模型的训练。所得到的测试结果如表 4-1 所示。

表 4-1 CNN_11 在不同输入数据的识别准确率

Tab4-1 The accuracy of CNN_11 in different input data

数据集	识别准确率
原图	94.3%
原图+Gabor 特征图	95.1%
原图+HOG 特征图	94.9%

由表 4-1 的实验结果可知，在添加了 Gabor 特征图和 HOG 特征图之后，模型 CNN_11 的识别准确率分别提升了0.8%和 0.6%，这表明了传统的特征提取算法融合深度学习模型，可以有效地提升模型的识别性能。

4.3.2 Dropout 对实验的影响

时至今日，深度学习的模型深度逐渐在增加，结构也越来越复杂，每个深度学习都伴随着大量的参数。在训练过程中，常常会发生模型结构在训练集上表现得很好，在测试数据集上的准确率却一般的情况，这个就是过拟合的现象。而 dropout 方法，是一个减缓过拟合常用的技术，通常应用在全连接层上。对于丢弃神经元的概率，如果丢弃的概率太小，则与不使用 dropout 方法无异，而丢弃的概率太大，网络模型很有可能得不到充分的训练，达到的效果很有可能会更差。本节将使用 CNN_11 的网络结构，通过改变 dropout 概率，来研究其对识别率的影响 。

对于模型 CNN_11，本实验对所有全连接隐藏层做不同的 dropout 概率处理，丢弃的概率在分别为 0.2、0.3、0.5、0.7 和 0.8，实验结果如表 4-2 所示。在丢弃了20%和30%全连接隐藏层神经元的情况下，整个模型还保留着大量的神经元，CNN_11 在训练数据集上的识别率和在测试数据集上的识别率相差较大，训练识别率比测试识别率要高得多，有明显的过拟合现象。在丢弃概率为 0.5 的情况下，训练识别率和测试识别率分别为95.3%和94.3%，CNN_11 在训练数据集和测试数据集中都达到了较好的识别效果。而在丢弃概率为 0.7 和 0.8 的情况下，不管是在训练数据集中，还是在测试数据集中，虽然识别率相差不大，但都没能达到理想的结果，识别率仅仅在90%左右。原因是丢弃的神经元过多，使得模型得不到充分的训练。

表 4-2 不同 dropout 概率对应的识别准确率

Tab4-2 The accuracy corresponding to different dropout probabilities

丢弃概率	Train_accuracy	Test_accuracy
0.2	97.8%	88.9%
0.3	97.7%	91.9%
0.5	95.3%	94.3%
0.7	90.9%	90.4%
0.8	87.6%	89.3%

4.3.3 BN 对模型训练的影响

在神经网络的训练过程中，由于机器中 RAM 等硬件设施的限制，往往不能将所有的数据同时加载到模型中进行训练，通常情况下采取小批量 mini-batch 的训练方式。BN 算法通过计算 mini-batch 训练数据的均值和方差，动态地调整整个网络模型的中间输出，使得下一层的输入数据的均值和方差都在一定的范围内，减轻了模型对初始化参数的依赖，使得神经网络在各层的中间输出值更为稳定，从而使得网络模型的训练速度更快。

在 CNN 中，通常对卷积层和全连接隐藏层做批量归一化处理。对于卷积层而言，批量归一化通常当作一个独立的层置于卷积计算和激活函数之间。增加批量归一化方法的卷积层结构如图 4-8 所示。同理，在全连接隐藏层中，BN 层通常也会置于激活函数之前。

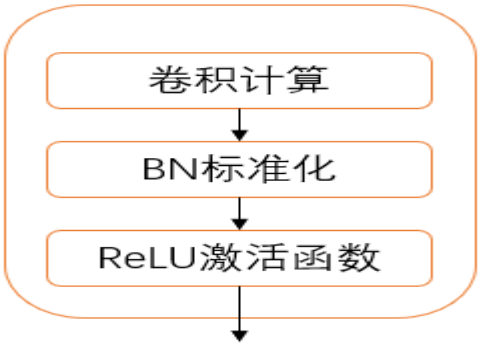


图 4-8 添加 BN 的卷积层内部结构

Fig4-8 Internal structure of Convolution layer with BN

在本节中，将研究 BN 层对 CNN_11 模型训练的影响，分别做了两次不同的实验，实验一中 CNN_11 不做批量归一化处理，实验二中 CNN_11 将除了输出层，其他层中都增加了 BN 层的结构。实验效果如图 4-9 所示。

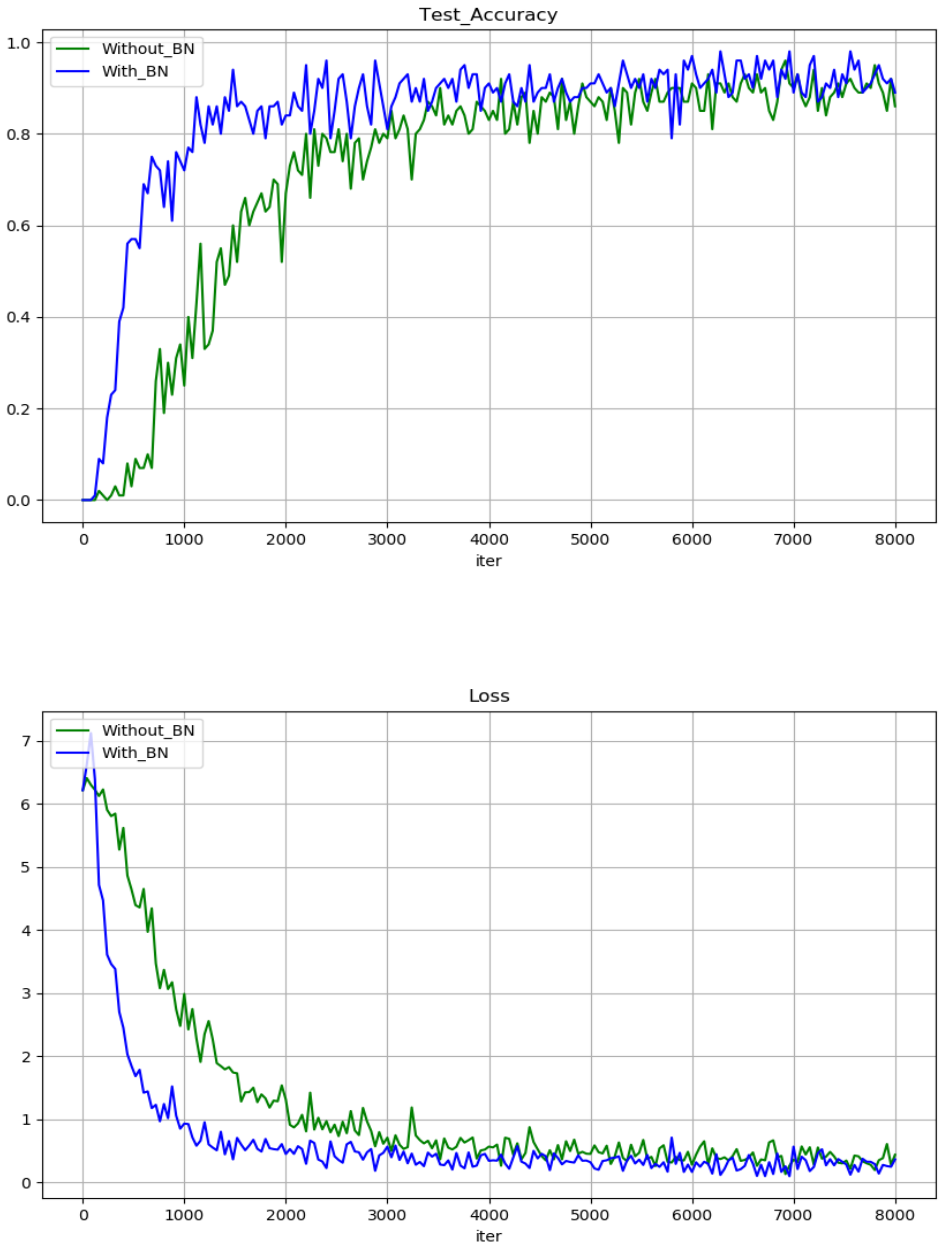


图 4-9 BN 对比实验结果

Fig4-9 Experimental result with BN

从图 4-9 中可以看出，加入了 BN 层后，CNN_11 模型在迭代了大约 2000 次之后就可以达到收敛，而没有加入 BN 层的模型，需要经过将近 3500 轮的迭代才能够达到收敛。同时，加入 BN 层的 CNN_11 数值变换更加稳定。实验表明，加入 BN

层的 CNN 模型能够更快收敛，同时抖动的幅度更小。

4.4 脱机手写汉字识别系统的实现

对于定义的网络模型 CNN_11 与训练良好的模型参数，利用 TensorFlow 框架提供的 API，本文将其保存于本地磁盘中，以便进行手写汉字识别系统的搭建[50]。HCCR 系统最主要的两个组件是 CNN 模型和模型对应的结构参数。在进行手写汉字识别之前，需要构建网络模型以及给模型加载对应的参数。网络模型结构可以手动构建与前面章节相同的 CNN_11 的结构，也可以从本地磁盘的 meta 文件中读取，本文选择手动构建与保存模型一摸一样的网络，然后对该网络进行参数的加载。加载参数完毕后，便可对输入的手写汉字图像进行计算和识别。整个识别流程如图 4-10 所示。对于输入的手写汉字图像，经过训练良好的网络模型的计算后，在模型的输出层得到了 softmax 函数计算的结果，取输出层神经元中概率最大值所对应的类别作为本系统的识别结果，通过与标签的映射关系，即可得到了输入图像的最终结果。

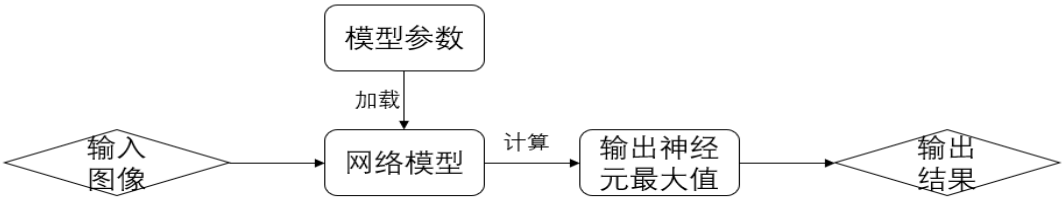


图 4-10 HCCR 系统的流程图

Fig4-10 The process of HCCR system

图 4-11 中的五张图像分别代表了随机挑选的汉字“万”、“世”、“东”、“乘”和“会”，将这五张图像分别输入到本文的 HCCR 系统中，得到的输出结果如图 4-12 所示。输出结果表明，该 HCCR 系统能够顺利地完成手写汉字识别的任务。

图 4-11 输入测试图像“玩”、“世”、“东”、“乘”、“会”

Fig4-11 Test image “玩”,“世”,“东”,“乘”,“会”

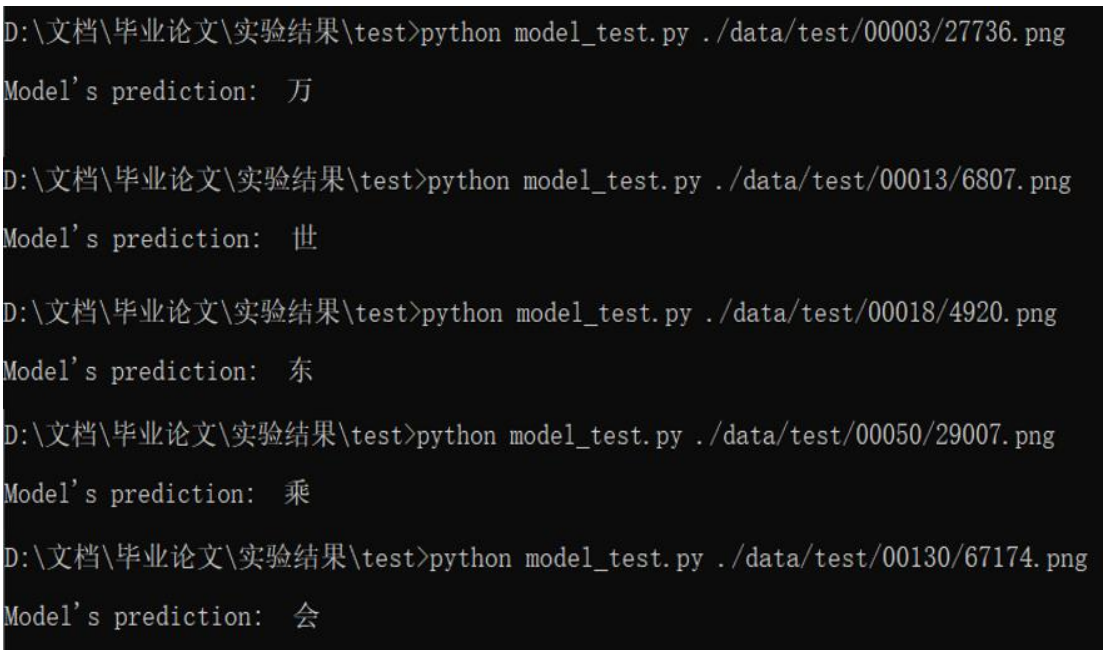


图 4-12 识别结果

Fig4-12 Recognition results

4.5 本章小结

本章首先介绍了 Gabor 和 HOG 两种特征提取的算法，提出了特征提取融合 CNN 模型的训练方式。其次对第三章的实验进行优化和改进，将 Gabor 特征图和 HOG 特征图与原图像一起训练卷积神经网络模型，分析和对比特征提取对实验结果的影响。再次探讨了不同的 dropout 概率对模型识别准确率的影响以及批量归一化算法对模型训练过程中的影响，最后实现了一个简单易用的手写汉字识别系统。

总结和展望

本文工作的总结

由于汉字类别众多、手写随意性较大等原因，手写汉字识别一直以来都没有得到较好地解决。CNN 近年来在图像识别领域内大放异彩，其良好的特征提取和图像分类的能力，给图像识别带来了强有力的技术。本文在 CNN 模型的基础上，针对脱机手写汉字识别做了如下的研究工作：

（1）查阅了大量的文献，调研了手写汉字识别以及深度学习的国内外研究现状，理清了手写汉字识别的难点。

（2）研究了前馈神经网络和 CNN 的理论基础和结构模型，包括正向传播和反向传播的计算过程以及模型参数更新的策略，并研究了对网络模型性能和训练收敛速度影响较大的激活函数、dropout 概率以及批量归一化算法。

（3）针对 CASIA-HWDB1.1 数据集的具体情况，设计了五个不同深度不同结构的卷积神经网络模型，分析和研究了不同深度不同结构的 CNN 模型对识别准确率的作用。

（4）研究了传统图像识别领域中常用的特征提取方法，包括 Gabor 特征提取和 HOG 特征提取，并将其融合到 CNN 模型当中。通过实验表明，Gabor 特征和 HOG 特征与原图像作为 CNN 模型的输入，能够提升 CNN 模型的识别性能。

（5）对于卷积神经网络难于训练的问题，研究了模型训练过程中的微调技术，以进行网络结构参数的初始化，并在网络的训练过程中研究不同的 dropout 概率对模型准确率的影响以及批量归一化算法对模型训练的效果。最后设计出了一个简易的脱机手写汉字识别系统。

对未来的展望

本文通过 CNN 模型在脱机手写汉字识别中取得了较好的识别效果。受到自身水平和计算资源等条件的限制，本文依然存在许多不足的地方，在日后的研究工作中还需要在以下方面进一步研究和改进：

（1）增大手写汉字识别类别的个数。受限于计算资源，本文无法完全使用 CASIA-HWDB1.1 全部的数据集，无法完全训练一个能够识别 3755 个一级汉字的模型，更不用说 6763 个类别的二级汉字。在未来，计算资源允许的情况下，应该考虑

对更大的训练数据集进行训练和测试。

（2）提高模型的训练速度。深度学习的模型往往伴随着大量的参数，需要庞大的数据集和很强的计算能力。如何快速地训练出一个效果良好的模型，是一个亟待解决的问题，需要在网络结构和相关优化算法上面进一步的研究。

（3）进一步深入研究整行手写汉字的识别。本研究主要针对单个手写字符的识别，但是在日常生活中，汉字往往都是以一整行或者多行的形式出现，如何较好地识别整行的手写汉字，是一个极具实用价值的研究。常用的思路有两种，一种是通过将整行手写汉字切分为单个字符，分别使用 CNN 依次对单个手写字符进行识别，最后重组识别结果；另外一种是 CNN 结合循环神经网络（RNN）的 CRNN 识别方法，通过在 CNN 中加入 RNN 的记忆功能，实现端到端的纯深度学习的识别方法。CRNN 使用卷积层提取输入图像的特征信息，并使用 RNN 中的双向 LSTM 进一步提取图像卷积特征中的序列特征，对序列特征进行预测，最后通过 CTC 的 loss 计算方法，完成一整行手写汉字的识别。但是截至今日，对于一整行的手写汉字而言，还没有报道出较好的识别效果，仍然需要进一步的研究。

参考文献

[1] Turing A M . Computing Machinery and Intelligence[M]// Computers and Thought. American Association for Artificial Intelligence, 1950.

[2] 周志华. 机器学习[M]. 清华大学出版社, 2016:10-15.

[3] 孙华,张航.汉字识别方法综述[J].计算机工程,2010,36(20):194-197.

[4] Casey R , Nagy G . Recognition of Printed Chinese Characters[J]. IEEE Transactions on Electronic Computers, 1966, 15(1):91-101.

[5] Yasuda M , Yamada H , Saito T . An improved correlation method for character recognition. A proposal of reciprocal feature extraction[J]. Systems, computers, controls, 1984, 15(4): 29-38.

[6] 金连文,钟卓耀,杨钊,杨维信,谢泽澄,孙俊.深度学习在手写汉字识别中的应用综述[J].自动化学报,2016,42(08):1125-1141.

[7] Liu C L, Marukawa K. Pseudo two-dimensional shape normalization methods for handwritten Chinese character recognition. Pattern Recognition, 2005, 38(12): 2242-2255.

[8] Liu C L. Normalization-cooperated gradient feature extraction for handwritten character recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2007, 29(8): 1465-1469.

[9] Ge Y, Huo Q, Feng Z D. Offline recognition of handwritten Chinese characters using Gabor features, CDHMM modeling and MCE training. In: Proceedings of the 2002 IEEE International Conference on Acoustics, Speech, and Signal Processing. Orlando, FL, USA: IEEE, 2002. I-1053-I-1056.

[10]Mangasarian O L , Musicant D R . Data Discrimination via Nonlinear Generalized Support Vector Machines[J], 1970.

[11]Kim H J , Kim K H , Kim S K , et al. On-line recognition of handwritten chinese characters based on hidden markov models[J]. Pattern Recognition, 1997, 30(9):1489-1500.

[12]Liu C L , Sako H , Fujisawa H . Discriminative Learning Quadratic Discriminant Function for Handwriting Recognition[J]. IEEE Transactions on Neural Networks, 2004, 15(2):430-444.

[13]Graham B. Spatially-sparse convolutional neural networks. arXiv: 1409.6070, 2014.

[14]Yin F , Wang Q F , Zhang X Y , et al. ICDAR 2013 Chinese Handwriting Recognition Competition[C]// International Conference on Document Analysis & Recognition. IEEE Computer Society, 2013.

[15]Mcculloch W S , Pitts W . A Logical Calculus of the Ideas Immanent in Nervous Activity[J]. biol math biophys, 1943.

[16]Rosenblatt, F. The Perceptron: A Probabilistic Model for Information Storage and Organization in The Brain[J]. Psychological Review, 1958, 65:386-408.

[17]Rumelhart D E , Hinton G E , Williams R J . Learning representations by back-propagating errors[J]. nature, 1986, 323(6088):533-536.

[18]Lecun Y , Bottou L . Gradient-based learning applied to document recognition[J]. Proceedings of the IEEE, 1998, 86(11):P.2278-2324.

[19]Hinton G , Salakhutdinov R . Reducing the dimensionality of data with neural networks[J]. Science, 2006, 313(5786):p. 504-507.

[20]Krizhevsky A , Sutskever I , Hinton G . ImageNet Classification with Deep Convolutional Neural Networks[J]. Advances in neural information processing systems, 2012, 25(2).

[21]Simonyan K , Zisserman A . Very Deep Convolutional Networks for Large-Scale Image Recognition[J]. Computer ence, 2014.

[22]Szegedy C , Liu W , Jia Y , et al. Going Deeper with Convolutions[J]. 2014.

[23]He K , Zhang X , Ren S , et al. Deep Residual Learning for Image Recognition[C]// 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2016.

[24]Qian N . On the momentum term in gradient descent learning algorithms[J]. Neural Networks, 1999, 12(1):145-151.

[25]Duchi J , Hazan E , Singer Y . Adaptive Subgradient Methods for Online Learning and Stochastic Optimization[J]. Journal of Machine Learning Research, 2011, 12(7):257-269.

[26]Zeiler, M. D . Adadelta: an adaptive learning rate method[J]. computer science, 2012.

[27]Tieleman, T., & Hinton, G. (2012). Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. COURSERA: Neural networks for machine learning, 4(2), 26-31.

[28]Kingma D , Ba J . Adam: A Method for Stochastic Optimization[J]. Computer Science, 2014.

[29]Liu L , Jiang H , He P , et al. On the Variance of the Adaptive Learning Rate and Beyond[J]. 2019.

[30]Lecun Y , Bengio Y , Hinton G . Deep learning[J]. nature, 2015, 521(7553):436.

[31]赵继印, 郑蕊蕊, 吴宝春,等. 脱机手写体汉字识别综述[J]. 电子学报, 2010, 038(002):405-415.

[32]焦李成,杨淑媛,刘芳,王士刚,冯志玺.神经网络七十年:回顾与展望[J].计算机学报,2016,39(08):1697-1716.

[33]周飞燕,金林鹏,董军.卷积神经网络研究综述[J].计算机学报,2017,40(06):1229-1251.

[34]邱锡鹏. 神经网络与深度学习[M]. 机械工业出版，2020:111-125.

[35]Goodfellow I , Bengio Y, Courville A. 深度学习[M]. 人民邮电出版社, 2017:201-220.

[36]Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov. Dropout: a simple way to prevent neural networks from overftting. JMLR. 2014.

[37]Ioffe S , Szegedy C . Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift [J]. 2015.

[38]Liu C L , Yin F , Wang D H , et al. Online and offline handwritten Chinese character recognition: Benchmarking on new databases[J]. Pattern Recognition, 2013, 46(1):155---162.

[39]Zhang H , Guo J , Chen G , et al. HCL2000 - A Large-scale Handwritten Chinese Character Database for Handwritten Character Recognition[C]//2009 10th

International Conference on Document Analysis and Recognition. IEEE Computer Society, 2009.

[40]Jin L , Gao Y , Liu G , et al. SCUT-COUCH2009-a comprehensive online unconstrained Chinese handwriting database and benchmark evaluation[J]. International Journal on Document Analysis & Recognition, 2011, 14(1):53-64.

[41]Lin M , Chen Q , Yan S . Network In Network[C]// ICLR. 2014.

[42]TensorFlow [Online], available: https://github.com/tensorflow/tensorflow, May 11, 2016.

[43]Erhan D , Bengio Y , Courville A , et al. Why Does Unsupervised Pre-training Help Deep Learning?[J]. Journal of Machine Learning Research, 2010, 11(3):625-660.

[44]Zhong Z , Jin L , Xie Z . High Performance Offline Handwritten Chinese Character Recognition Using GoogLeNet and Directional Feature Maps[J]. 2015.

[45]何志国,曹玉东.脱机手写体汉字识别综述[J].计算机工程,2008(15):201-204.

[46]Yong G, Huo Q and Feng Z.-D, Offline recognition of handwritten Chinese character using Gabor features, CDHMM modeling and MCE training, ICASSP, pp.1053-1056, 2002.

[47]Dalal N , Triggs B . Histograms of Oriented Gradients for Human Detection[C]// IEEE Computer Society Conference on Computer Vision & Pattern Recognition. IEEE, 2005.

[48]Ding K , Liu Z , Jin L , et al. A comparative study of gabor feature and gradient feature for handwritten chinese character recognition[C]// 2007 International Conference on Wavelet Analysis and Pattern Recognition. IEEE, 2008.

[49]傅红普, 邹北骥. 方向梯度直方图及其扩展[J]. 计算机工程, 2013(05):218-223.

[50]李金洪. 深度学习之 TensorFlow 入门、原理与进阶实战[M]. 北京：机械工业出版社. 2018:151-180.

攻读学位期间发表的论文

[1] 袁柱.一种基于卷积神经网络的小篆识别方法[J]. 现代计算机.

致谢

时光匆匆，转眼间研究生三年的学习生涯即将过去。在这三年里，有过很多的欢乐，也遇到不少的挫折，感谢一路陪伴我成长的老师、同学和亲朋好友们。

首先，感谢我的指导老师徐海水老师，感谢徐老师在这三年的时间里对我学习和生活上的帮助。在学习上，徐老师总是耐心地帮我解决学习过程中所遇到的问题，在论文完成的过程中给了我很多的指导和建议。徐老师治学严谨的态度和对其他领域最新研究成果的关注，是我们学习的好榜样。在与徐老师的交谈中，总能锻炼到自己的思维逻辑能力和对问题的思考方式。

感谢实验室师兄师姐一直以来对我的引导和帮助，尤其是宋怦阳师兄和龙可师兄对我科研和工作提出的建议和帮助。感谢三位室友一直以来对我的照顾，感谢陈锦芸同学一直以来的陪伴和在撰写论文期间的互相监督。

感谢家人一直以来对我的大力支持，从研究生入学开始，一直都在背后给我加油鼓劲，在这次疫情期间，感谢爸妈给我提供了良好的环境，让我能够安心完成毕业论文。

感谢广东工业大学！从物理学院到计算机学院，从本科生到研究生，在这七年无比美好的时光里，让我能够在这个平台上展示自我，让我能够增进自己的阅历，接触到了工程技术的魅力，体会到了学术前沿的思想。

最后感谢论文审阅的老师们和专家们，感谢你们抽出宝贵的时间对我论文的指导和建议。