分类编号：___ 单位代码：10065

密　　级：___ 学　　号：1410090003

天津师范大学

研究生学位论文

论文题目：基于 RNN 的端到端自然场景文本识别

学 生 姓 名：安兴乐 申请学位级别：硕士

申请专业名称：计算机应用技术

研 究 方 向：图像处理与模式识别

指导教师姓名：朱远平 专业技术职称：副教授

提交论文日期：2017 年 3 月

天津师范大学硕士学位论文原创声明

本人郑重声明:此处所提交的硕士学位论文《基于 RNN 的端对端场景文本识别方法的研究》，是本人在导师指导下，在天津师范大学攻读硕士学位期间独立进行研究工作所取得的成果。据本人所知，论文中除已注明部分外不包含他人已发表或撰写过的研究成果。对本文的研究工作做出重要贡献的个人和集体，均已在文中以明确方式注明。本声明的法律结果将完全由本人承担。

作者签名：　　日期：　年　月　日

天津师范大学硕士学位论文使用授权书

《基于 RNN 的端对端自然场景文本识别方法的研究》系本人在天津师范大学攻读硕士学位期间在导师指导下完成的硕士学位论文。本论文的研究成果归天津师范大学所有，本论文的研究内容不得以其他单位的名义发表。本人完全了解天津师范大学关于保存、使用学位论文的规定，同意学校保留并向有关部门送交论文的复印件和电子版本，允许论文被查阅和借阅，同意学校将论文加入《中国优秀博硕士学位论文全文数据库》和编入《中国知识资源总库》。本人授权天津师范大学，可以采用影印、缩印或其他复制手段保存论文，可以公布论文的全部或部分内容。

本学位论文属于（请在以下相应方框内打“ √");

保密□,在　　年解密后适用本授权书

不保密□

作者签名:　　日期:　　年　　月

导师签名：　　日期:　　年　　月

摘 要

随着移动智能设备的普及，图像的获取越来越便捷，提取并理解智能设备拍摄的图像中的信息具有重要的应用价值。文字由于包含了大量丰富的语义信息，对于计算机理解图像具有重要意义。文本识别作为计算机视觉领域基础性研究工作，自然场景的文本识别具有深远的研究意义和实用价值。在图像检索、人机交互、车牌识别等领域有着广泛的应用。自然场景由于复杂的背景、多样化的字体、以及随意的分布等问题，使得传统的光学字符识别技术（OCR）难以被广泛应用。自然场景中的文字检测与文本识别仍然存在很多需要解决的技术难点。

对单个字符的位置获取与特征提取是文本识别中最为关键的一步。由于噪音、模糊、遮挡、布局等的影响，字符检测与识别精确度受到很大制约。为解决这个问题，很多方法被提出，如自适应二值化算法、级联部件提取与直接字符检测等算法。虽然这些算法在特定场景下能够取得较好的效果，面对复杂的自然场景效果并不乐观。直接进行字符的检测与识别有着很大的困难。

得益于计算机硬件性能的提高，以及大数据获取越来越便捷，深度学习技术近年来取得了重大突破，在模式识别领域被广泛应用。本文借鉴图像检索、目标检测、自然语言处理等领域的进展，结合深度学习技术，提出了新的识别方法，本文主要工作如下：

（1）提出了基于RNN结合CNN的文本识别方法。利用CNN提取原始图像的抽象特征，然后将特征送入RNN中建模图像全局信息。针对传统的文本识别需要较深的专业背景知识，文本检测与识别等过于复杂等问题，该方法采用端到端训练，无需过多的背景知识。大大降低了训练难度。

（2）对验证码图片数据集和Google的街景门牌号码SVHN数据集上进行了两部分实验。第一部分实验验证了CNN结合RNN相对RNN与对文本识别的精度与收敛速度的提高；第二部分实验探索CNN结合RNN对文本识别的提高以及不同的滑动窗对识别率的影响。

关键字：自然场景；文本识别；深度学习；RNN；CNN

ABSTRACT

With the popularity of mobile smart devices, it becomes more and more convenient to acquire images. Thus it has high application value to extract and understand the information in the images captured by smart devices. Since text contains a great number of semantic information, it is significant important to extract the information included in the figures using computer. As a basic research work in the field of computer vision, text recognition in natural scenes has great research and practical value. It has been widely used in the fields of image retrieval, human-computer interaction and license plate recognition. Due to the complexity of background, diversity of fonts, and the random distribution of natural scenes, it makes the traditional optical character recognition technology (OCR) difficult to be widely used. Text detection and recognition in natural scenes still have many technical difficulties to be solved.

The most important step in text recognition is to obtain the position and extract the feature of single character. Due to the influence of noise, blur, occlusion and layout, the accuracy of character detection and recognition is greatly restricted. To address these issues, several approaches were proposed, which employed adaptive binarization, connected component extraction or direct character detection. These methods work well in certain cases, but are still far from producing all satisfactory results especially in complex natural scenes. There is great difficulty in direct character detection.

Thanks to the improvement of computer hardware performance and the convenience of large data acquisition, deep learning technology has made a great breakthrough in recent years, and is widely used in pattern recognition. Based on the progress of image retrieval, object detection and natural language processing, this paper proposes a new text recognition method combined with deep learning technology. As follows is the main work of this paper:

(1) This paper proposes a text recognition method based on convolutional neural networks (CNN) and recurrent neural networks (RNN). We use CNN to extract the abstract features of original image, and then feed the abstract features to RNN to model the global information. Our method is trained using an end to end method which researchers don’t need much background knowledge, while text detection and recognition is too complex and requires deeper professional background knowledge. This greatly reduces the difficulty of training.

(2) We conducted experiments on captcha image and the famous Google Street View House Number dataset. In the first part, we demonstrate the effectiveness and speed of convergence of CNN combined with RNN compared with RNN for text recognition. In the second part, we investigate the improvement of text recognition by RNN which combined with CNN, and the influence of different sliding windows on recognition rate.

Key words: natural scenes; text recognition; deep learning; recurrent neural networks (RNN) ; convolutional neural networks (CNN)

目录

第一章　绪论

1.1 文本识别的研究背景及意义

文本是人们重要的交流工具，如何让计算机来理解并处理人类的文字一直是计算机视觉与模式识别领域重要的课题。早在上个世纪初期光学字符识别（Optical Character Recognition, OCR）[1]技术就被提出，经过长期发展，OCR在工业领域的应用日臻成熟[3]。

然而随着智能移动设备走入千家万户，在智能移动上直接理解并处理文字信息必然有着巨大的应用需求。相对于多媒体文档中的常见的亮度、色彩、形状等底层信息，文字能够直接传递高级语义信息。合理地利用多媒体文档上的文字信息，能够极大的推动以下领域的发展：

（1）图像与视频检索。移动互联网时代图像与视频文件越来越多，如何对海量的图像与视频进行检索是一个重要课题。基于内容的图像和视频检索[4][7]是根据图像和视频的抽象信息，依据一定的相似性指标来进行检索。目前难点在于计算机对图像与视频的相似性判断是根据图像的底层视觉特征来进行的相似性判断。而人由于在生活中拥有大量的经验与知识，对图像的相似性判断是根据语义信息来进行的相似性判断，这就产生了人对图像所理解的“语义相似”与计算机理解的“视觉特征相似”的“语义鸿沟”问题。而图像或视频中所识别的文本信息，可以提供较为丰富的高层语义信息，从而可以根据这些语义信息来对图像或者视频进行检索。

（2）车牌识别。随着经济的发展与生活水平的提高，机动车数量越来越多。高效的车辆管理不仅可以缓解交通压力，而且对公共安全有重要意义。车牌往往就是唯一确定车辆身份的标志。现在往往在重要路口都有摄像头监控过往车辆的车牌，记录车辆的行驶信息，可以对违反交通法规的车辆记录在案，或者对可疑车辆及时报警。总之，车牌识别系统[8] [8] [8]对车辆管理具有重要意义，是模式识别与计算机视觉领域重要课题之一。

（3）无人驾驶。随着传感器与自动导航技术的发展，自动驾驶汽车[10]的也开始走入人们的视线。2012年5月Google的第一辆自动驾驶汽车获得自动驾驶许可证。自动驾驶可以有效的辅助甚至代替人类操作机动车量，能够为人类节省不少的驾驶时间，并且可以极大的减少能源消耗，降低交通事故的发生，让乘坐车辆更加安全。高速行驶车辆在复杂路况下识别道路指示牌和建筑标志中的文字也相当重要[5]。

1.2 文本识别技术发展

文本的分析与识别概念的提出是现在已经有了 70 多年的历史了，创建学科性领域类别的划分后，科学研究和实验检验都是以 OCR 为主要代表技术的[6]。第一个具有商用性能的 OCR 技术时出现在上世纪 50 年代，随着电子技术和物理成像技术日臻完善，OCR 成为了文档扫描处理和识别领域最为通用手段和效能较高的技术手段。最开始，OCR 被研究者设计作为早期的邮政和银行系统内部使用，OCR 被用作表格读取，支票解

析，邮政地址识别等技术。在 OCR 技术还不成熟的时候，国际商业机器巨头 IBM 公司就分别研发了 IBM1418 和 IBM1428 两个型号的当时具有较准确识别功能的文档图像识别解析设备；随后，美国政府又在 1965 年开始采用 OCR 技术的开发，在邮政编码识别和邮件自然分类上做出应用。日本邮政系统受到美国邮政 OCR 技术提高效率的启发也开始利用 OCR 技术进行信件分类；在 70 年代的意大利和德国也在本国的邮政运营系统上使用 OCR 技术革新；80 年代后期，日本利用自己电子技术优势，开发 OCR 技术用于专利文档分析和管理技术，而现阶段，商用方面的 OCR 各类衍生产品被广泛应用在文档处理和自动化办公上，这对削减人力成本，提高文件规范化和结构化文档印刷字符类的识别能力达到了 98%以上。

虽然 OCR 技术取得了可喜的表现，但是在当今信息社会和各种繁杂文本信息充斥的空间中，OCR 技术还有很高的可提升空间，OCR 对是被对象的限制也成为其技术上的最大局限性。随着这个行业的发展，现实需要为复杂自然场景条件下的文本识别技术提出了新的挑战。

第一、新型的数字图像采集设备的大量使用

早期的图像采集设备主要是成像仪器、扫描仪等，但是随着技术进步，制造能力的不断提升，图像数字采集设备的大量使用，例如 PDA、交通摄像头，手机成像摄像头、数码电子成像设备等。这些种类繁杂的图像数字采集设备，已经渗透到人类日常生活中的方方面面，而他们工作原理就和现代技术有差别，它们所能够采集到的有效图像数字成像，和扫描仪器所采集到的成像，从结构类型、规范性、操作性上都存在着很多的弊端和不利情况，这些原始设备所采集的数字图像可能不具有规范性和严格操作性，包括多种文字的混合搭配，文本的不规则旋转，图像的扭曲与形变，光照强度的不均匀等，已经无法满足于 OCR 对于输入的要求和限制，这也正是传统 OCR 处理的一些局限性的重要体现。

第二、IT 和 WEB 技术的发展

从最早期的文本识别要求的提出来看，文本的识别和分析较为规则和特定排列的数字组合，这个技术主要集中应用于邮政运营系统、银行支票系统，专利部门文件管理的内容上。但是现实中，自然条件下文本环境更加的复杂，伴随计算机技术迅猛发展，IT 和 WEB 技术利用，折让各种信息类型，信息内涵量级，交互方式更替等有了本质上的变化，多重的新型文本图像识别技术不断的涌现出来，这也是的文本分析和识别技术从处理能力、手段、对象的变化，以便适应当前的复杂变化。

1.2 自然场景文本识别难点分析

近几十年来文字识别研究得到了空前的发展，一大批科研工作者致力于这方面的研究，但是目前仍然不存在一种在所有自然场景中适用的实际方法来识别文本。如图 1-1 所示，光照不均匀，分辨率过低，视角的多变性，拍摄设备的移动都会导致图像模

糊，文字难以识别。传统应用于扫描文档的 OCR 引擎几乎无法正确的扫描识别出这样的文字。

图 1-1 自然场景下的文本图像示例

自然场景下的文本图像与传统的扫描文档相比，具有以下的特点：

1. 图像背景更加复杂。扫描文档的图像背景往往为单一颜色，对文本识别影响较小。自然场景中的图像则可能包含车辆、行人、楼宇建筑、花草树木、广告牌与交通灯等与文字没有任何相似性的元素。同时，背景元素对文本的遮挡和相似色彩的背景或前景也加大了识别难度。背景的光照、颜色、亮度等不确定，变化毫无规律。因此自然场景文本识别需要适应复杂的场景，对算法鲁棒性提出更高的要求。

2. 字符畸变严重。扫描文档的图像中字体多为印刷体，几何形变较小。自然场景中文本图像多由相机拍摄，由于拍摄视角的多变、拍摄设备的差异、透视效果的不同都可能导致图像的畸变。自然场景文本图像为了增加艺术效果，也会增加人为扭曲，旋转，倒影等效果。这无疑增大了字符识别的难度。

3. 文档结构复杂。传统文档图像中文本往往布局固定，文字为水平或垂直分布。而自然场景图像中文本分布随意，字符极有可能出现在文本图像的各个位置。对于传统文本识别，往往需要首先对字符进行检测，确定文本的位置后提取文本信息再进行识别。

为了获取对复杂图像中文本信息的识别和分析， 然后进行分类和检索等功能，文本信息需要能够直接和有效地描述图像的视觉特征，清晰地表达图像内的对象类型和给出的图像和场景的语义信息。同时，这类自然性的图像，需要更加丰富而复杂的图像匹配，其中包含的文本信息，与一般性的扫描类型的文档进行比较，其中获得了一些有价值的信息，表 1-1 就是对图像文本信息的属性进行归纳和对比。

从上表中可以看出来，这些针对于扫描类型的文本识别不同，其在复杂背景和自然拍摄图像文本识别和分析，困难性和复杂性是提高的。扫描文档中，在语言种类、字体、字号、排列方式、对比度等方面，结构化和规范化程度较高，这使得该类问题可以直接用于商业 OCR 进行处理。

现在越来越多的研究将机器学习方面的算法引入到场景文本识别领域来，取得了很好的效果，但是距离实用化还有一段距离。为取得更好的识别效果，训练算法需要大量的专业领域知识同时结合机器学习相关算法，对科研工作者提出较高的要求。

表 1-1 图像内文本属性对照[52]

属性		说明	图像/视频	扫描文档
几何属性	字符大小	字符规则性	不均匀	相对统一
	字体种类	字符形状	多样	相对统一
	排列分布	水平/垂直	存在	较多
		倾斜直线	存在	整体均匀
		曲线排布	存在	较少存在
		自由排列	存在	不存在
		视角变换	大量存在	不存在
		扭曲变形	大量存在	不存在
	字符间距	文本聚合程度	不均匀	相对均匀
颜色属性	色彩分布	图像色彩表现	文本内字符颜色存在不同	相对一致
光度属性	灰度分布	图像灰度表现	依赖设备，存在曝光，暗纹	均匀
边缘属性	字符边界	字符与背景视差	依赖设备，存在模糊，散焦	明显
背景属性	文本背景	文本与背景载体	复杂，丰富，多样	简单

综上所述，随着智能移动终端设备以及移动互联网的逐渐普及，多媒体文档中文字识别会越来越重要，而目前的自然场景文字识别技术依然不能满足现实需要，所以展开对文字识别的研究显得非常迫切。

1.3 自然场景文本识别的研究现状

1.3.1 传统文字识别研究

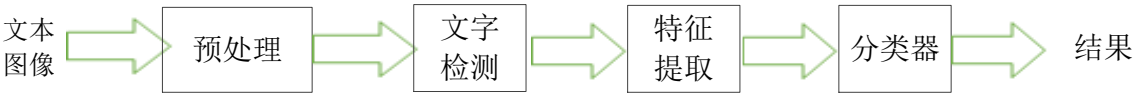


图 1-2 自然场景文本处理流程

传统的文本识别是通过对单个字符的检测然后对字符提取特征，将特征送入分类器训练以此得到合适的分类器，最终完成识别。如图所示，文字图像经过预处理后，检测到文本区域，提取出文本信息，之后提取人工设计的特征，送入分类器训练，经过大量训练后，得到识别结果。经典算法例如，Rodriguez[45]等人使用支持向量机（support vector machine, SVM）结合Fisher[50]向量处理的算法，通过提取Fisher特征，送入SVM训练来建模图像与文本的关系，最终来识别图像中的文本。同样类似的还有Bissacco[49]，他使用神经网络结合HOG特征来做字符分类器，对分割后的结果使用N元模型（N-gram）来做搜索，最终得到字符结果。

Neumann提取字符级联部件特征来定位与识别字符。Weinman将字符分割与识别结合的方法来进行识别。Shi结合DPM与条件随机场来构建部件模型来对场景文字进行识别。Neumann提出一种结合滑动窗与字符连接部件的方法对字符进行识别。

传统文字识别方法要求研究人员预先设计恰当的特征表达方法。但是自然场景文字极其复杂，设计能够表达文字信息的特征非常的困难，所以在自然场景文字识别中，传统的文字识别方法效果有限。

1.3.2 基于深度学习的文本识别

基于传统文字识别方法的局限，越来越多的研究开始利用深度学习的方式来进行识别。2013年，Alsharif[43]等人将卷积神经网络提取的特征送入隐马尔科夫（Hidden Markov Model, HMM）网络，来实现文字识别。Goodfellow [47]等人使用训练好的字符位置分类器来预测文本中字符数量，直接使用卷积神经网络进行字符识别，该方法对Google的街景门牌号码数据集和验证码数据集均取得了很好的效果。2016年，富士通研究所利用过分割技术结合卷积神经网络（Convolutional Neural Network, CNN）[26]首先分割出字符，然后对字符进行识别。完成了对自然场景文字的分割与识别。是目前自然场景文本识别领域效果最好的算法。

1.3.3 端到端研究方法的兴起

目前的方法还是有很多的问题：算法鲁棒性差，难以应对长字符串以及粘连字符；人工设计与提取特征往往不具有很好的通用性。虽然基于深度学习的方法取得了一定成绩，但是往往需要研究者有较高的专业背景知识，算法过多依靠研究者的经验来确定网络参数，技术过于复杂。

2015 年，Miao[51] 使用循环神经网路网络识别语音信息。只需要足够多的标记样本，就可以完成对语音的识别。网络结构简单，训练过程无需过多人工干预。Kai wang 提出了一种结合字符检测、词组检测与非极值抑制的方法来整合自然场景下文本的检测与识别流程。这种端到端（end to end）的方法逐渐成为人们的研究热点。该方法不需要研究者具有较强的背景知识就可以训练出很好的模型，取得很好的效果。

1.4 本文的研究目标与内容安排

本文针对上述的问题展开探索与研究，提出了基于深度学习的文本识别方法，以实现端到端的训练，无需专业领域的背景知识,简化了特征提取与分类器训练算法难度。

本文的主要研究内容如下：

1. 研究学习并总结了经典文本检测与识别方法。对不同方法的特点进行了分析。
2. 提出了结合CNN与RNN的新型网络结构，实现了端到端的文字识别。
3. 构建标准验证码数据集，对本文模型进行评估。

本文的内容与结构安排如下：

第一章：自然场景文本研究综述。介绍了自然场景文本识别的研究意义与研究背景；文本识别与自然场景文本识别的技术发展；以及传统文字识别与基于深度学习的文本识别的发展，分析了当前算法的一些不足并介绍了基于端到端研究方法的兴起。

第二章：深度神经网络概述。对人工神经网络（Artificial Neural Networks, ANN），递归神经网络(Recurrent Neural Network, RNN)与卷积神经网络（Convolutional Neural Networks, CNN）的基本理论进行简要的介绍。

第三章：基于CNN-RNN的文本识别方法。提出了CNN结合RNN的新型网络结构来的研究方法，以及训练网络时常用的防止过拟合的dropout技术，并学习研究了CTC解码技术。

第四章：自然场景文本识别实验。自然场景文本实验系统设计与常用的深度学习开发框架介绍，在验证码数据集和Google街景门分析了不同实验参数对识别准确度与收敛速度的影响。

第五章：全文总结与展望。对本文的工作进行了总结，并指出了CNN-RNN网络的不足，并提出对算法的改进，对未来要开展的工作作出展望。

第二章 深度神经网络概述

2.1 传统人工神经网络

人工神经网络（Artificial Neural Networks, ANN）或称作神经网络[15]，是模拟动物的神经网络的计算模型。这些计算模型通过模拟动物大脑的神经网络进行抽象计算，构建人工神经元，然后按照一定的拓扑结构来构建神经元之间的连接，来模拟动物的神经网络。20 世纪 40 年代，D.O.Hebb 首次提出了改变神经元连接强度的 Hebb 规则，打开了神经网络研究的大门。这时的神经网络实只有一个隐含层，且受限于当时计算机硬件资源，并没有预期的效果[11][13]。20 世纪 70 年代，Paul Werbos 提出了对以后有深远影响的反向传播算法(back—propagating，BP)[13]，但没有引起学界的关注。之后，Geoffrey Hinton 对 BP 算法再次进行了研究[14]，在 Nature 上发表了一篇对这个算法的详细研究，人们再次开始关注人工神经网络。

2.1.1 神经元结构

神经元是人工神经网络最重要的组成部分，它的构造非常简单。人工神经元和感知器类似，模拟动物的神经元特性，接受一组输入并产生输出。图 2-1 所示是一个最典型的神经网络模型，仅包含一个神经元。

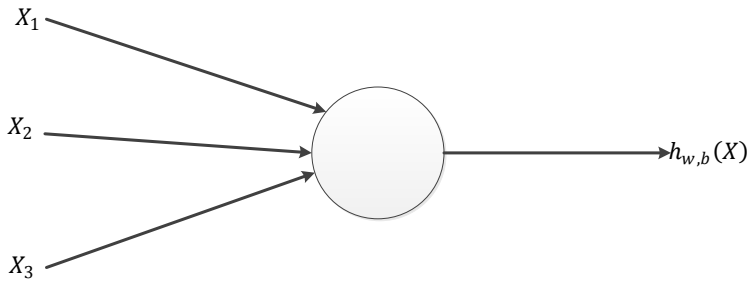


图 2-1 神经元结构

这个神经元的输入为X_1，X_2，X_3 输出为$h_{w,b}(X) = f(W^T X)$，f为激活函数，W为线性变换，b为偏置项。

2.1.2 常用激活函数

线性模型的表达能力往往有限，为了增强神经网络的表达能力，我们需要使用连续非线性激活函数。下面介绍三种神经网络中最常使用的激活函数。

logistic 函数

logistic 函数是一种 sigmoid 型函数，定义为：

$$\sigma(x) = \frac{1}{1+e^{x}} \tag{2.1}$$

logistic 函数将输入映射到（0，1）范围内。函数的特点和动物神经元类似，对一些输入会产生兴奋，对另外一些输入会产生抑制。当输入越大时，函数输出越接近 1；输入越小时，函数输出越接近 0.

tanh 函数

tanh 函数也是一种 sigmoid 型函数，定义为：

$$\tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \tag{2.2}$$

tanh 函数可以看作是放大并平移后的 logistic 函数：

$$\tanh(x) = 2\sigma(2x) - 1 \tag{2.3}$$

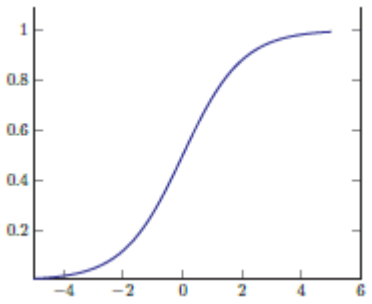
图 2-2 给出了 logistic 函数与 tanh 函数的形状。

线性修正单元

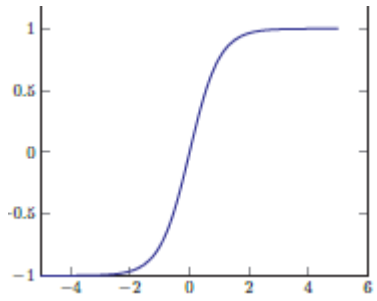
线性修正单元（rectified linear unit, ReLU）也称为 rectifier 函数[17][18]。在深层神经网络中普遍被使用。ReLU 形状如图 2-2 所示，是一个“斜坡”函数，定义为：

$$\text{rectifier}(x) = \begin{cases} x & x \geq 0 \\ 0 & x < 0 \end{cases} \tag{2.4}$$

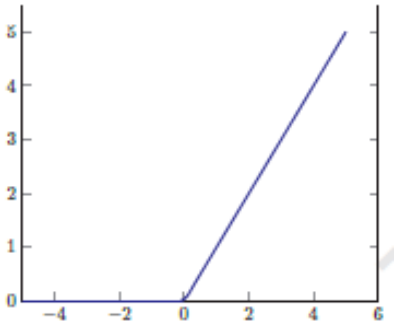
采用 ReLU 函数的神经网络在计算上更加高效。Relu 函数会使神经元的一部分输出为 0，使得网络的相对稀疏，可以减少参数的相互依存关系，有效缓解了过拟合问题。



(a) logistic 函数



(b) tanh 函数



(c) rectifier 函数

图 2-2 常用激活函数曲线

2.1.3 前馈神经网络

单层神经元通过训练只能得到一个线性分类面，其分类能力有限。动物大脑中神经细胞是相互连接在一起的，我们可以构建类似的神经网络结构，组成多层的人工神经网络。多层神经网络能够表示多个非线性分类面，分类能力大大增强。其中，前馈神经网络是最简单的多层神经网络[19]。

在前馈神经网络中，各神经元分属于不同的层。每一层的神经元接收上一层神经元输出的信号，并输出给下一层的神经元。整个网络中无反馈，信号从输入层向输出层单向流动，可以用一个有向无环图表示。如图 2-3 所示是一种广泛使用的前馈神经网络。它包含输入层 L1，两个隐藏层 L2,L3,以及输出层 L4。

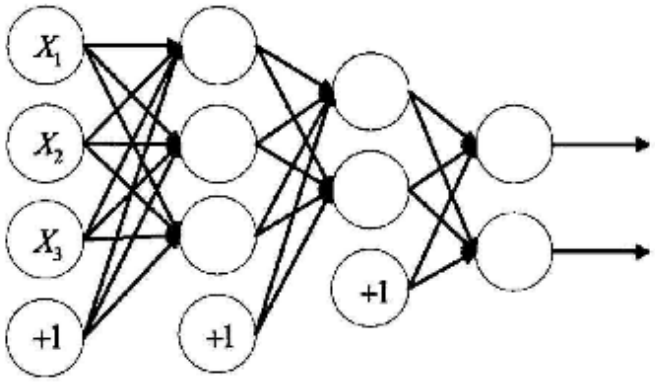


图 2-3　三层神经网络

我们使用下面的记号来描述前馈神经网络。l:　表示神经网络的层数；n^l:表示第L层神经元的个数；$f_l(\cdot)$:表示L层神经元的激活函数；$W^{(l)} \in R^{n^l \times n^{l-1}}$:表示$\mathrm{L}-1$层到第L层的权重矩阵；$b^{(l)} \in R^{n^l}$:表示$\mathrm{L}-1$层到第 L 层的偏置；$z^{(l)} \in R^{n^l}$:表示L层神经元的输入；$a^{(l)} \in R^{n^l}$:表示L层神经元的输出。我们可以得到前馈神经网络第L层神经元的输入为：

$$z^{(l)} = W^{(l)} \cdot a^{(l-1)} + b^{(l)} \qquad (2.5)$$

第 L 层的神经元输出为：

$$a^{(l)} = f_l(z^{(l)}) \qquad (2.6)$$

公式合并为：

$$a^{(l)} = f_l(W^{(l)} \cdot a^{(l-1)} + b^{(l)}) \qquad (2.7)$$

整个网络可以看作一个函数$\mathrm{f(x;W,b)}$。

2.1.4 反向传播算法

给定样本$(X^{(i)}, y^{(i)}, 1 \le \mathrm{i} \le \mathrm{N})$, 前馈神经网络的输出为$\mathrm{f(x|W,b)}$, 损失函数为：

$$\mathrm{J(W,b)} = \sum_{i=1}^{N} L\left(y^{(i)}, f\left(X^i \middle| W, b\right)\right)$$

$$= \sum_{i=1}^{N} J(W, b; X^{(i)}, y^{(i)}) \qquad (2.8)$$

采用梯度下降法来对损失函数最小化，来对参数进行更新：

$$W^{(l)} = W^{(l)} - \alpha \frac{dJ(W,b)}{dW^{(l)}} \tag{2.9}$$

$$b^{(l)} = b^{(l)} - \alpha \frac{dJ(W,b;X^{(i)},y^{(i)})}{db^{(l)}} \tag{2.10}$$

α为更新率。根据链式法则可得：

$$\frac{dJ(W,b;x,y)}{dw_{ij}^{(l)}} = tr\left(\left(\frac{dJ(W,b;x,y)}{dZ^{(l)}}\right)\frac{dZ^{(l)}}{dw_{ij}^{(l)}}\right) \tag{2.11}$$

对于第 L 层，我们定义误差项$\delta^{(l)} = \frac{dJ(W,b;x,y)}{dZ^{(l)}}$为损失函数关于第 L 层的神经元$Z^{(l)}$的偏导数。具体算法如下：

输入：训练集$(X^{(i)},y^{(i)})$, i=1, …, N, 迭代次数为 T

输出：W, b

For t = 1… T do

For i = 1…N do

计算前馈神经网络每一层的状态与激活值，直到最后一层；

计算反向传播中每一层的误差$\delta^{(l)}$；

更新参数:

$$W^{(l)} = W^{(l)} - \alpha \frac{dJ(W,b)}{dW^{(l)}};$$

$$b^{(l)} = b^{(l)} - \alpha \frac{dJ(W,b;X^{(i)},y^{(i)})}{db^{(l)}};$$

End

End

2.2 循环神经网络

2.2.1 简单循环神经网络

传统神经网络中每层之间的神经元是没有连接的，这种网络对于很多问题无能为力。循环神经网络（Recurrent Neural Network, RNN ）[20]，利用带自反馈的神经元

可以处理任意长度的输入序列。这种网络更加适合处理与序列有关的任务。它已被广泛应用于语音识别，自动翻译，聊天机器人等任务上。

一个简单循环神经网络按时序展开之后如图 2-4 所示：

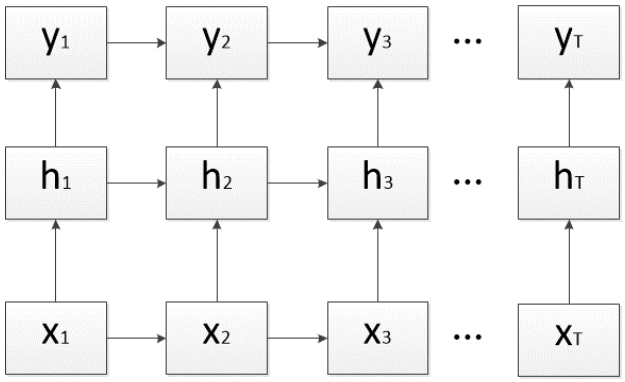


图 2-4　循环神经网络按时序展开

这种网络在神经元之间既有内部的反馈连接，又有各层之间的前馈连接。假设在时刻t时，输入为X_t，隐层状态为h_t。我们可以得到如下的表达形式：

$$h_t = f(Uh_t + WX_t + b) \tag{2.21}$$

f为激活函数，U和W为线性变换函数，b为偏置项。我们可以看到h_t不仅和当前时刻的输入有关，也和上一个时刻的隐层状态有关。

2.2.2 RNN 反向传播算法

循环神经网络的 参数训练可以通过随时间进行的反向传播算法[21] [21]来确定。假设循环神经网络在每个时刻 t 都有一个损失J_t。整个序列损失为 $J = \sum_{t=1}^{T} J_t$。序列的损失J关于U的梯度为：

$$\frac{\partial J}{\partial U} = \sum_{t=1}^{T} \frac{\partial J_t}{\partial U}$$

$$= \sum_{t=1}^{T} \frac{\partial h_t}{\partial U} \frac{\partial J_t}{\partial h_t} \tag{2.22}$$

根据链式法则，展开可得

$$\frac{\partial J}{\partial U} = \sum_{t=1}^{T} \sum_{k=1}^{t} \frac{\partial h_k}{\partial U} \frac{\partial h_t}{\partial h_k} \frac{\partial y_t}{\partial h_t} \frac{\partial J_t}{\partial y_t} \tag{2.23}$$

将$\frac{\partial h_t}{\partial h_k}$项展开，$\frac{\partial h_t}{\partial h_k} = \prod_{i=k+1}^{t} \frac{\partial h_i}{\partial h_{i-1}}$

$$= \prod_{i=k+1}^{t} U^T \, diag[f'(h_{i-1})] \quad (2.24)$$

令 $\eta = \prod_{i=k+1}^{t} U^T \, diag[f'(h_{i-1})]$。

将 η 代入公式（2.24），可得

$$\frac{\partial J}{\partial U} = \sum_{t=1}^{T} \sum_{k=1}^{t} \frac{\partial h_k}{\partial U} \frac{\partial y_t}{\partial h_t} \frac{\partial J_t}{\partial y_t} \eta^{t-k} \quad (2.25)$$

显然，当 η>1，$t-k \to \infty$时，$\eta^{t-k} \to \infty$，会出现梯度爆炸问题。而如果 η>1，$t-k \to \infty$时，$\eta^{t-k} \to 0$，会出现梯度消失。因此，在实际中简单循环神经网络只能学习到短周期的时序依赖关系，即长期依赖问题[23]。

2.2.3 长短时记忆网络（Long short term memory, LSTM）

为解决梯度爆炸问题和长期依赖问题，Hochreiter 和 Schmidhuber 引入了门机制（Gating Mechanism）来控制信息的累计速度，并选择性遗忘以前的信息。这就是长短时记忆神经网络（Long Short-Term Memory Neural Network, LSTM）[24]。在时刻t时候，记忆单元C_t记录了到时刻t为止的所有历史信息，受三个门单元控制：输入门i_t，遗忘门f_t和输出门O_t。LSTM 的更新方式为：

$$i_t = \sigma(w_t X_t + U_t h_{t-1} + V_i C_{t-1}), \quad (2.26)$$

$$f_t = \sigma\left(w_f X_t + U_f h_{t-1} + V_f C_{t-1}\right), \quad (2.27)$$

$$o_t = \sigma(w_o X_t + U_o h_{t-1} + V_o C_t), \quad (2.28)$$

$$\widetilde{C_t} = tanh(w_c X_t + U_c h_{t-1}), \quad (2.29)$$

$$C_t = f_t \odot C_{t-1} + i_t \odot \widetilde{C_t}, \quad (2.30)$$

$$h_t = o_t \odot \tanh(C_t), \quad (2.31)$$

这里σ是 logistic 函数，V_i，V_t，V_o是对角矩阵，X_t是当前时刻的输入。遗忘门f_t控制每一个内存单元需要遗忘的信息，输入门i_t 控制每一个内存单元记忆多少新的信息，输出门O_t控制每一个内存单元输出多少信息。

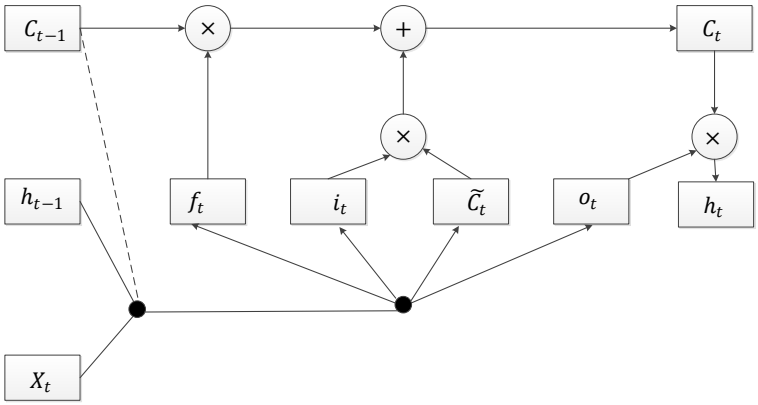


图 2-5 LSTM 结构图

LSTM 网络可以学到长周期的历史信息。LSTM 网络的计算机构可以用 图 2-5 来表示。

2.3 卷积神经网络

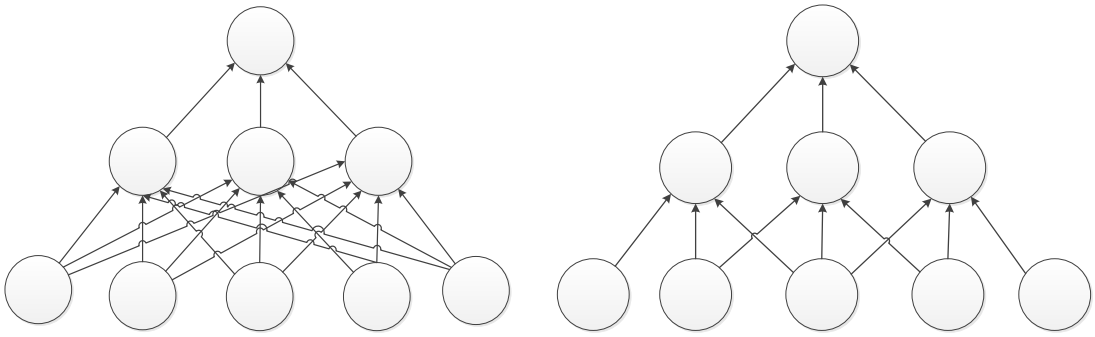
卷积神经网络（Convolutional Neural Networks，CNN）属于前馈神经网络。上个世纪六十年代，Hubel 和 Wiesel 对动物的视觉系统的研究中，发现大脑皮层的构造对于局部的视觉空间非常敏感，提出了局部感受野（receptive field）的感念。进而发展出了卷积神经网络（Convolutional Neural Networks，CNN）。卷积神经网络被广泛应用于计算机视觉与模式识别领域，是目前机器学习领域的研究热点。卷积神经网络通过局部感受野、全值共享和下采样实现了对输入图像的位移变化、尺度变化、形变变化的不变性。对于卷积神经网络第 L 层神经元的输入有如下定义：

$$X^l = f(W^{(l)} \otimes X^{(l-1)} + b^{(l)}) \tag{2.31}$$

这里$W^{(l)}$为第 L 层滤波器，$X^{(l-1)}$代表 L-1 层神经元的输入，$b^{(l)}$代表偏置矩阵。f 为激活函数。通常会使用多组滤波器来得到多组输出，以此来增强卷积层的表达能力。

2.3.1 局部连接

全连接网络中，层与层神经元结点之间是全连接的。而在卷积神经网络中，每一层的神经元节点只和相邻层较近的神经元节点相互连接。



a. 全连接网络　　b. 卷积神经网络

图 2-6 全连接与卷积神经网络连接方式示意图

在普通前馈神经网络中，每层神经元节点之间均相互连接。卷积神经网络中，最底层神经元只与它最相近的三个神经元连接。这样大大降低了神经网络的参数量。

2.3.2 权值共享

卷积层中每一个滤波器对输入数据进行卷机操作，提取图像特征。每个滤波器共享相同的参数。

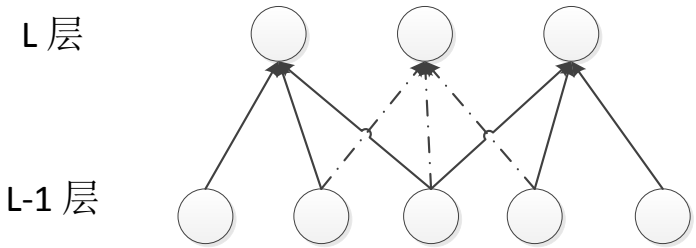


图 2-7 权值共享示意图

在上图中，L 层有三个神经元，与特定神经元相连接的连接线的权重是共享的。上图展示部分有三组共享权值的滤波器。共享权值使得训练时神经网络要学习的参数数量大大降低。滤波器用来提取局部区域的特征，相当于一个特征提取器。

2.3.3 下采样

卷积层共享权重虽然可以减少神经元之间连接的个数，但是经过卷积后的特征维数仍然很高。在卷积层后进行池化（Pooling）操作，可以降低特征维数，避免过拟合。设经过第 L 层卷积后的特征为X^l，我们将X^l分为 K 个区域$R_k, \mathrm{k} = 1, \ldots \mathrm{K}$,加上下采样函数$\mathrm{down}(\cdots)$后的神经元定义为：

$$X^l = f(W^{(l)} \otimes \mathrm{down}({R_k}^{(l-1)}) + b^{(l)}) \qquad (2.32)$$

其中，$W^{(l)}$和$b^{(l)}$分别为待训练的权重与偏置项。

下采样（Subsampling）函数一般取区域中的最大值（Maximum Pooling）或者平均值（Average Pooling）。

$$pooling_{max}(R_k) = \max_{i \in R_k} a_i \tag{2.33}$$

$$pooling_{avg}(R_k) = |\frac{1}{R_k}| \sum_{i \epsilon R_k} a_i \tag{2.34}$$

池化操作使得神经元对于较小的形态学改变能够保持不变性，并且可以使得感受野更大。

2.3.4 LeNet-5 示例

1989 年 LeCun 提出了经典的卷积神经网络模型 LeNet-5[38]，之后被美国多家银行用来识别支票上面的手写体数字。

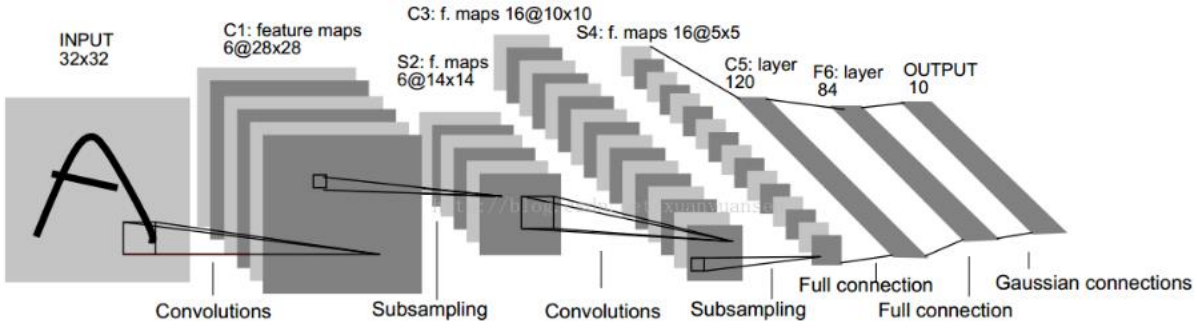


图 2-8 LeNet-5 网络结构

LeNet-5 一共有 8 层。第一层为输入层，输入图像为 32×32 像素的黑白图片。

C1 层为卷积层。卷积核为 5×5，一共 6 个卷积核。输出为 6 组 28×28 的 feature map. 该层神经元个数为 6× 784 =4704。每个 feature map 包括 5×5 的卷积核参数，以及一个偏置项，所以待训练参数为（5×5+1）×6=156。

S2 层为下采样层。对 C1 层卷积得到的 6 组 feature map 中 2×2 邻域进行下采样，取平均值。该层神经元个数为 14×14。根据公式 2.32 可知，可训练参数为 6×（1+1）=12 个。

C3 层同样为卷积层。上一层 S2 有 6 组 feature map，C3 层将上一层不同的 feature map 进行组合，再进行卷积操作。表 2-1 所示为 S2 层与 C3 层的连接关系。

表 2-1

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	√				√	√	√			√	√	√	√		√	√
1	√	√				√	√	√			√	√	√	√		√
2	√	√	√				√	√	√			√		√	√	√
3		√	√	√			√	√	√	√			√		√	√
4			√	√	√			√	√	√	√		√	√		√
5				√	√	√			√	√	√	√		√	√	√

由表 2-1 可知，C3 层第一个特征映射依赖于 S2 层前三个 feature map，C3 层第二个特征映射依赖于 S2 层第二、三、四个 feature map，依此类推。C3 层一共有 60 个卷积核，大小为 5×5。C3 层可训练参数为 $5\times5\times60+16=1516$。

S4 层又是下采样层。对 C3 层卷积得到的 16 组 feature map 中 2×2 邻域进行下采样。该层神经元为 5×5。可训练参数为 16×（4+1）= 2000 个。

C5 层为卷积层。每个 feature map 与上一层的全部 feature map 相连接。一共有 120×16=1920 个卷积核，卷积核大小为 5×5。可训练参数为 1920×25+120 = 48120 个。

F6 层为全连接层。该层将上一层的 feature map 全部连接，输出为 84 个神经元。可训练参数个数为 84×（120+1）= 10164 个。

最后一层为输出层。输出层由 10 个径向基函数（Radial Basis Function， RBF）组成，输出结果为 10 个数字的概率值。

2.4 本章小结

本文回顾并研究了传统人工神经网络的基本结构，常用激活函数的作用与前馈神经网络的前向算法，以及如何通过反向传播算法来训练网络参数。研究了循环神经网络如何对时序信息进行建模分类。针对传统的循环神经网络训练时容易产生梯度爆炸和梯度弥散的问题，研究学习了基于门机制的长短时记忆网络。对于目前研究热门卷积神经网络，详细介绍了卷积神经网络的作用机制以及降低计算量的方法，最后以经典网络模型 Lenet-5 为例讲解卷积神经网络的工作机制。

第三章 基于 CNN-RNN 的文本识别方法

3.1 问题分析

计算机对文本的识别是通过将图像信息转化为计算机可表示的信息并进行处理的过程。由于自然场景的文字及其复杂，现在的识别方法没有很好的鲁棒性，并且需要较深的专业领域知识，通用性较差。 同样的问题，在物体分类、图像分割等其他计算机视觉领域也存在。随着深度学习理论的发展，卷积神经网络在计算机视觉领域取得了不错的成绩。卷积神经网络直接输入图像像素信息，不需要人工设计特征，同时也大大减少了数据预处理的工作量。

虽然基于卷积神经网络[25]的自然场景文字识别取得了一定的成果，但是仍然存在一些不足。卷积神经网络针对单字的识别虽然较高，但是对于复杂的自然场景下的文本信息，却依然需要配合传统的字符定位与分割方法[26]，不能实现端到端的训练。而且当图像所含文本较长时，识别率会明显下降。

我们借鉴深度学习在语音识别上取得的成就[28]，将文本识别任务作为特殊的“语音信息”进行处理[27]。本章将会结合 CNN 与 RNN，提出一种新型网络结构，用于文本识别，以期能够实现端到端的自然场景文本识别。

3.2 方法概述

本文提出基于 CNN 结合 RNN 为基础框架的网络模型。

实验流程主要包括图像预处理，CNN 层特征提取，RNN 层网络预测，CTC 解码最终得到期望的识别结果。如图 3-1 所示，自然场景文本图像经过 CNN-RNN 网络模型便可以识别出文本中的字符。

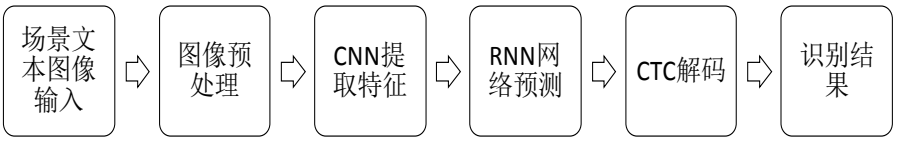


图 3-1　文本识别流程图

3.3 图像预处理

自然场景文本图像通常为彩色图像且图像的光照强度不同，文本的差异性较大。为此，我们采用图像预处理的方法将文本图像先经过图像大小归一化以及灰度归一化以及滑动窗提取图像切片。

灰度归一化一方面可以去除不同文本图像之间的颜色差异，另一方面也可以减少计算机的运算负载，且并不会影响文本图像中字符的识别精度。图像大小归一化则有利于后续神经网络的处理。

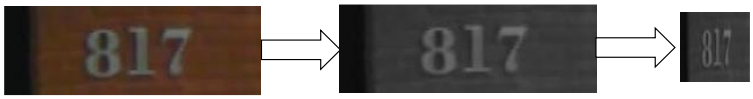


图 3-2 图像预处理

图像预处理示意图如图 3-2 所示,先将实际自然场景文本图像灰度归一化读取灰度图，再对图片调整大小，将图像大小归一化为 60× 60 像素。

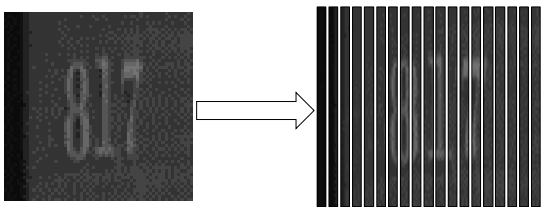


图 3-3 滑动窗提取图像切片

如图 3-3 所示。将图像归一化后，设定滑动窗大小为 5×60 像素，步长为 4 像素。这样会得到重叠为 1 个像素，大小为 5×60 像素的图像切片。

3.4 CNN 层

卷积神经网络（convolutional neural networks，CNN）层通过对图像的卷积提取局部特征，然后通过下采样减少网络计算量。图像完成预处理后，使用卷积神经网络层来对图像切片提取特征。这里设置卷积核大小分别为 3×3、5×5，步长均为 1。卷积后使用 Relu 函数进行激活，再经过池化层。CNN 层的输入为一批图片，由 2000× height × width × channels的张量组成，分别代表批大小、切片高度、切片宽度、切片通道数。

早期神经网络中人们通常使用 Sigmoid 以及 Tanh 函数作为神经元的激活函数。实验表明，数据在远离原点的区域变化不大，这会导致反向求导的时候梯度过快消失，使得训练变得困难。之后，因为 ReLU 函数更加接近生物神经细胞的激活模式，并且因为近似线性函数的优点，开始被应用于卷积神经网络。本文使用 ReLU 函数作为激活层的激活函数。

为减少计算量，图像切片经过卷积层、激活层后再次进行池化操作。本文使用最大化池化（Max pooling）:使用 2×2 的滑动窗进行池化，步长为 2，取滑动窗中的最大值输出。

3.5 RNN 层

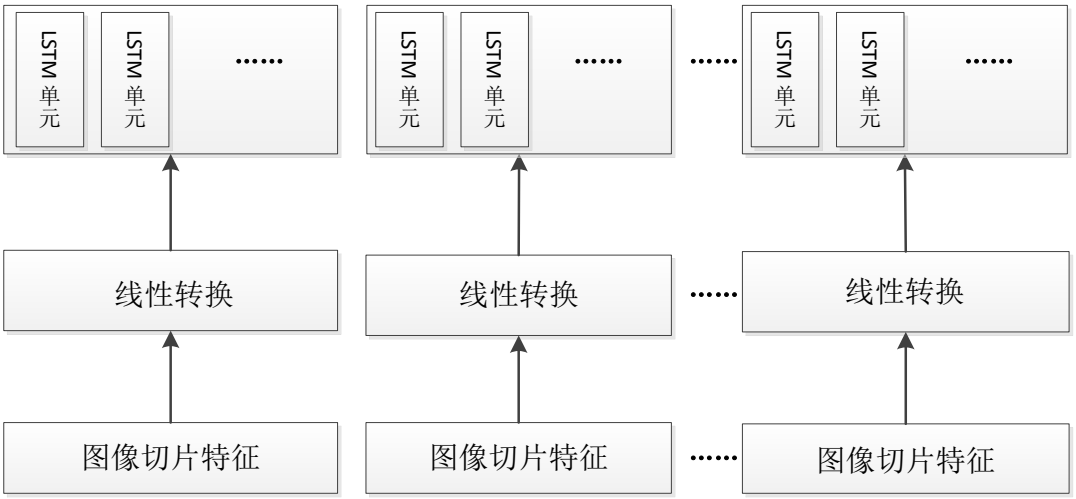
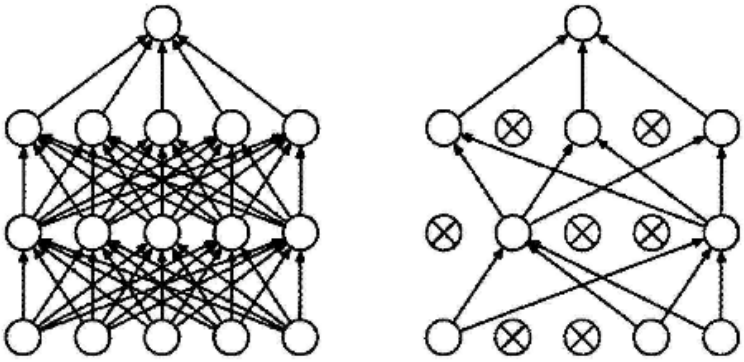


图 3-4 RNN 层处理流程

为解决反向传导时的梯度爆炸以及梯度弥散问题，本文中循环神经网络（Recurrent Neural Network, RNN）采用长短时记忆网络（Long short term memory, LSTM）。卷积层提取图像切片特征后，将特征数据送入 LSTM 网络。本文中每个 RNN 结构有一层隐含层，每层包含 100 个 LSTM 单元。特征数据首先经过一次线性转换，得到 1× 100的向量，以适应 LSTM 单元结构，然后送入下一层的网络。RNN 网络结构最终会输出预测的结果序列。

3.6 dropout

Dropout[34] 作为一种防止过拟合的方法，广泛被用在神经网络训练中。训练时，随机对全连接层直接连接移除。



(a) 不进行 dropout 的全连接网络　　(b) dropout 的全连接网络

图 4-5 dropout 示例

Wojciech Zaremba 在 2015 年将 dropout 应用于 RNN 网络[35]，取得了不错的效果。图 4-6 是 RNN 中 dropout 应用示例：

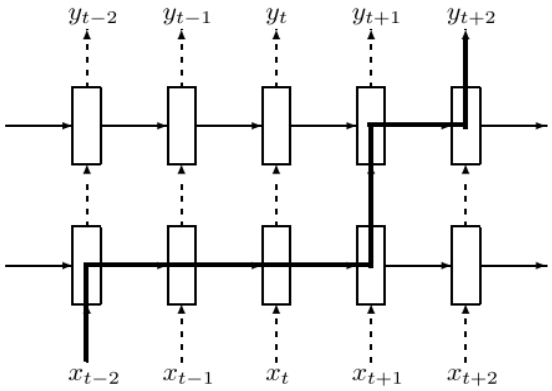


图 4-6 dropout 在 RNN 中应用

图中虚线代表应用 dropout 的连接。单步的层间连接进行一定概率的 dropout，序列的循环连接不使用 dropout。

3.7 CTC 解码

在 LSTM 网络中，如果序列的长度为 T，则会产生 T 个输出。这就需要将 LSTM 的输出解码为正确的类别。Connectionist Temporal Classification (CTC)[29]技术用来实现 LSTM 网络输出与数据标签对齐并计算 LSTM 网络训练过程中标签与 LSTM 网络输出之间的 Loss。

假设 S 代表训练集合，符合分布$D_{x\times z}$,输入空间 X=$(Rm)^*$,代表 m 维具有真实值的向量构成的序列的结合。目标空间 Z=L^*，代表标签 Label 的有限字符集合构成的序列的集合。S 中每个样本构成一个序列对(x,z)。目标序列 Z=$(z_1, z_2, \dots z_u)$的长度至多与输入序列 X=$(x_1, x_2, \dots x_u)$等长，即$\mathrm{U} \leq \mathrm{T}$。我们在这里利用 S 来训练一个时序分类器，对未知的输入序列分类，最小化任务对应的误差。

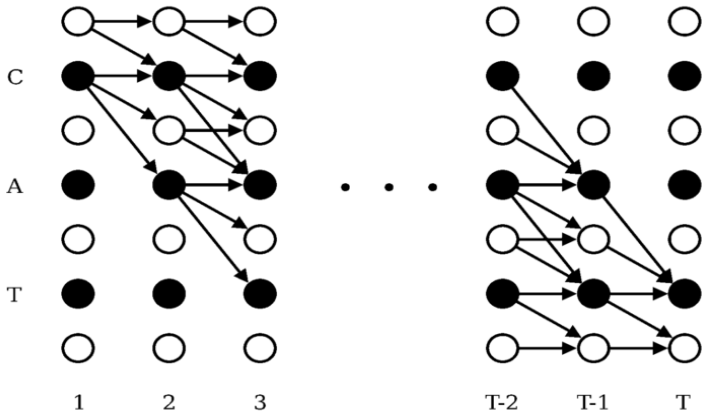


图 3-5 CTC 前向变量递归计算示意图

3.7.1 标签错误率

定义时序分类器 h 的标签错误率 LER（label error rate）为：分类结果与标签的平均编辑距离：

$$\mathrm{LER}(\mathrm{h},S') = \frac{1}{|S'|}\sum_{(x,z)\in S'}\frac{ED(h(x),Z)}{|Z|} \tag{3.1}$$

ED(p,q)表示 p 与 q 两个序列的编辑距离。

3.7.2 RNN 网络输出到标签

当输入为文本图像时，RNN 网络的输出为一系列标签的概率分布，对于英文文本图像来说，这些标签通常为 10 个阿拉伯数字和 26 个英文字母加一个 blank 标签，blank 标签的作用是分割每个标签。假设各个时刻 RNN 网络输出的标签概率互不影响，则 CTC 输出的概率就等于各个时刻输出的概率的乘积。即：

$$\mathrm{p}(\pi|\mathrm{x}) = \prod_{t=1}^{T} y^{t}_{\pi_t} \tag{3.2}$$

π 为满足目标标签的可能路径。

CTC 通过定义一个映射函数 F 来建立输出序列到标签序列的联系。映射函数 F 在数据标签中增加空白标签来分割标签之间的间隔，最后会把空白符号（blank）和预测出的重复符号消除。例如 F(233aac)=23ac;F(5_yy_y)=5yy。

映射函数 F 转化为目标标签的路径有很多中，最后输出的概率值即为所有这些路径的概率和：

$$\mathrm{p(L|X)} = \sum_{\pi \in F^{-1}(L)} p(\pi|\mathrm{X}) \tag{3.3}$$

CTC 使用一种动态编程的算法来对路径进行搜索，将所有可能路径分解为根据给定标签的前缀来计算。这里定义一个前向变量

$$\alpha(\mathrm{t,u}) = \sum_{\pi \in \mathrm{V(t,u)}} \prod_{i=1}^{t} y^{i}_{\pi_i} \tag{3.4}$$

这里的 V(t,u)代表所有可能路径的结合。由此可得出整个序列的概率为 T 时刻前向变量处在最后一个 blank 标签和倒数第二个非 blank 标签的概率和：

$$\mathrm{p(L|X)} = \alpha(T, U') + \alpha(\mathrm{T}, U' - 1) \tag{3.5}$$

前向变量的计算按照时间进行递归计算可得：

$$\alpha(t,u) = y^{t}_{l'_u} \sum_{i=f(u)}^{u} \alpha(t-1, i) \tag{3.6}$$

同理，这里定义一个后向变量$\beta(\mathrm{t,u})$,表示 t+1 时刻追加到$\alpha(\mathrm{t,u})$的所有路径输出结果为原始目标标签 L 的总概率，即：

$$\beta(\mathrm{t,u}) = \sum_{\pi \in \mathrm{W(t,u)}} \prod_{i=L}^{T-t} y^{t+i}_{\pi_i} \tag{3.7}$$

同样来递归计算后向变量，可得：

$$\beta(\mathrm{t,u}) = \sum_{i=u}^{g(u)} \beta(\mathrm{t+1, i})\, y^{t+1}_{l'_i} \tag{3.8}$$

根据前向后向变量，定义文本识别中的 CTC 损失函数为：

$$\begin{aligned} \mathrm{Loss(S)} &= -\ln \prod_{(x,z)\in S} p(z|x) \\ &= -\sum_{(x,z)\in S} \ln p(z|x) \end{aligned} \tag{3.9}$$

训练集为S，p(z|x)可以根据前向-后向变量来计算。对于任意时刻 t,可以计算所有位置的正确路径总概率：

$$\mathrm{p(z|x)} = \sum_{u=1}^{|z'|} \alpha(t,u)\beta(\mathrm{t,u}) \tag{3.10}$$

损失函数为:

$$\mathrm{Loss} = -\sum_{(x,z)\epsilon S} \log P(z|x)$$

$$= -\ln \sum_{u=1}^{|z'|} \alpha(t,u)\beta(t,u)$$

以上便是 CTC 解码以及训练过程。

3.8 本章小结

本章针对自然场景文本识别的难点以及传统识别方法的局限，提出了 CNN 结合 RNN 的新型网络结构。并详细介绍了本文使用的 CNN 网络以及 RNN 网络的参数意义以及详细的配置信息，以及训练网络时防止过拟合而广泛被应用的 dropout 技术，并介绍了用于 RNN 网络的 dropout 技术。本章还研究学习了对 RNN 网络输出进行解码的 CTC 解码技术。

第四章　自然场景文本识别实验

深度学习的训练要求计算机有较高的计算能力。2012 年 ImageNet 竞赛第一名 AlexNet 模型[30][31][31]，使用两块 GTX 580 GPU,训练时间长达六天，2014 年 ImageNet 竞赛第一名 Oxford VGGNet 模型[33]，使用 4 块 Titan Black GPU，训练时间更是长达 3 周。另外，神经网络所含的参数众多，如神经网络层数、神经元数量、规则化系数、激活函数等。训练时超参数如学习率，批尺寸（batch size）大小等也会明显影响收敛速度。

本章尝试使用一些策略来缓解这些问题，并探索不同的参数对实验的影响。

4.1 实验环境介绍

4.1.1 不同深度学习开发框架对比

深度学习技术近年来发展迅速，很多互联网企业与研究机构也都推出了自己的深度学习开发平台，将深度学习理论迅速转化为代码应用到开发实践当中，大大促进了深度学习领域的发展。

我们首先介绍当前最为常用的几大深度学习开发框架，并比较各个框架的优点与适用场景。

Caffe

Caffe（Convolutional Architecture for Fast Feature Embedding）[30]最早是由伯克利大学视觉与学习中心的贾扬清博士开发的基于 C++/CUDA 实现的卷积神经网络工具。2013 年开源后迅速在深度学习开发人员中普及开，是深度学习领域最为流行的开发框架。

Caffe 利用了 MKL, OpenBLAS, cuBLAS 等计算库，支持 CUDA 加速；提供了一整套工具集，可以用于模型训练、预测、微调、数据预处理，以及自动测试；完全开源，代码组织良好，可读性强。

但是 Caffe 不支持递归神经网络。用户如果自己定义新的网络架构，需要改动源代码并配合 ProtoBuf 描述。同时为了支持 GPU，需要自己手动实现一遍 CUDA 版的 forward、backward、gradient update。目前 Caffe 只支持单机多卡的并行计算，不支持多几多卡的分布式计算。

PyTorch

PyTorch[40]是 Facebook AI 实验室领衔开发深度学习工具包，源自较早的 torch 工具包。支持大部分的机器学习算法，是一个轻巧的框架。集成了各种计算加速库，如

Intel MKL、CuDNN 和 NCCL 来优化速度。PyTorch 基于 python，底层使用 C/CUDA 扩展模块实现，可已在 GPU 加速基础上实现张量；支持自动求导。

PyTorch 使用反向模式自动微分的技术，可以零延迟地改变网络的行为。相对于 Tensorflow、Theano、Caffe 等框架，后者需要事先构建一个神经网络，然后进行使用。PyTorch 是目前动态神经网络最快的实现技术。使得定义神经网络获得最高的灵活性及速度。

MXNet

MXNet[41]是分布式机器学习社区（Distributed Machine Learning Common, DMLC）开发的一个同时兼顾效率和灵活性的深度学习框架，支持多机多卡的分布式运行。MXNet 支持众多的语言绑定（C++/python/R/Go），并且支持混合式符号编程和命令式编程。其核心是一个动态的依赖调度，能够自动并行符号和命令的操作，提供了两种编程接口：N 维数组（ndarray）接口，类似于 Matlab 或 Python 中的 numpy 或 PyTorch 中的 tensor；符号（Symbolic）接口，可以快速的构建一个神经网络，实现自动求导。

TensorFlow

2015 年 Google 推出了人工智能学习系统 TensorFlow[42]。并在 2017 年 2 月 16 日发布了 TensorFlow1.0 正式版。TensorFlow 是分布式大规模机器学习框架，并且支持手机等移动设备的移植。随着 TensorFlow 的发展，以及对 CuDNN 等加速库的支持，TensorFlow 成为目前最为流行的深度学习框架之一。

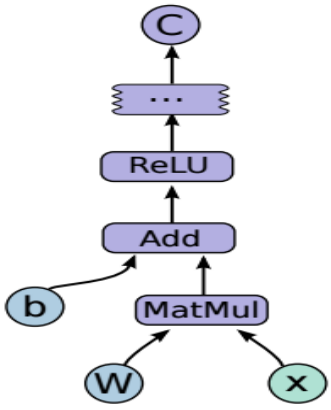


图 4-1 Tensorflow 计算图

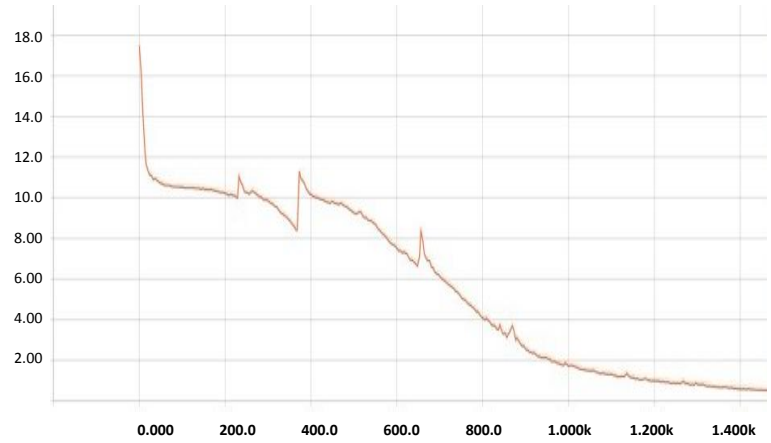
TensorFlow 已经被广泛应用于工业界，如小米、京东、Uber 等。TensorFlow 支持多种神经网络模型，可以轻松自定义网络结构。支持自动求导。采用数据流计算，其表达的数据流计算可以由有向图来表示。如何 4-1 所示，每个节点有一个或多个输入和零个或多个输出，表示一种操作的实例化。图中的叶子节点通常为常量或者变量，非叶子节点为一种操作，箭头所示为张量的流动方向。

除了上述的深度学习框架，还有一些框架可以尝试，这里不再一一列举。本文综合考虑多种框架，从易用性以及稳定性上考虑，选择 TensorFlow 作为本文的实验平台。

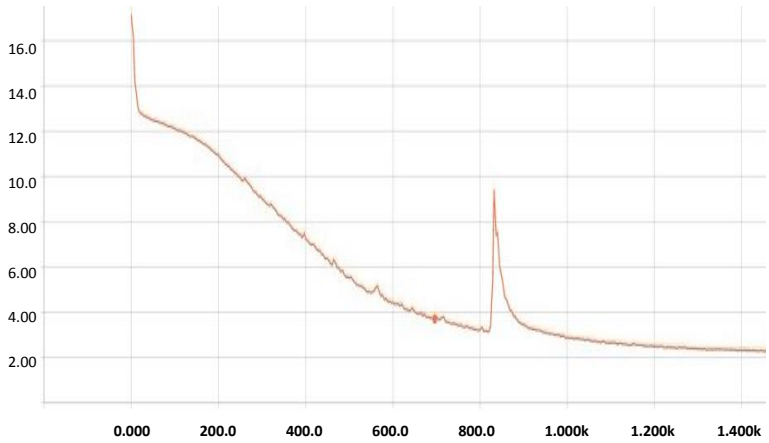
4.1.2 基于 GPU 的训练

GPU 相对 CPU 有更多的计算核心，每个核拥有相对较小的缓存，少而简单的数字逻辑运算但与。这使得 GPU 更加适合处理相对简单而重复的计算任务。目前主流的深度学习研究工作越来越多的使用 GPU 来对训练过程进行加速。训练过程中采用 批随机梯度下降中，每批样本的数量更具 GPU 显存大小进行合理设置，尽可能充分利用 GPU 显存。

如图 4-2，RNN 识别验证码图片时，对比 batchsize 为 2500 与 2700 的实验可以看到，batchsize 越大，Loss 值下降越快。意味着网络收敛更快。



(a) batchsize=2500



(b) batchsize=2700

图 4-2 收敛速度比较

本实验使用 Nvidia GTX 980ti GPU，深度学习框架 Tensorflow 基于 cudnn 加速。为减少 CPU 与 GPU 通讯耗时，一次性将全部训练数据读入内存，每一轮次迭代由内存送入 GPU 数据。这样大大增加了 GPU 利用率，减少资源浪费，加速训练过程。

表 4.1 实验环境

硬件配置	具体型号
CPU	Intel core i7-4790K
GPU	NVIDIA GTX980ti
内存	16GB
操作系统	Ubuntu 14.04
开发语言	Python
GPU 开发库	CUDA 5.1
图像处理库	Opencv 3.1
深度学习框架	Tensorflow 0.9

4.1.3 实验环境

实验使用 python 进行数据预处理，使用 tensorflow 框架[37]进行神经网络训练。具体实验环境如表 4.1 所示。

4.2 数据集

(a) Captcha 验证码图片 *(b) SVHN 数据*

图 4-3 数据集

本文使用了两种数据集，根据 python 验证码库 captcha 生成的验证码图片以及被广泛使用的 Google 街景门牌号数据集（SVHN）[36]。

Captcha 生成的验证码噪声、背景图与字符数均可以由用户决定，我们这里生成长度为 4 个与 3 个字符的验证码图片。SVHN 数据集是 Google 采集的真实世界的街景门牌号码数据集。SVHN 有 10 类数字，长度不等。其中 73257 幅图片用于训练，26032 幅图片用于测试，531131 幅图片是较容易的图片用于额外的补充数据。

4.3 实验流程

4.3.1 训练过程

这里以 Captcha 生成的验证码为例说明实验训练的流程。图 4-4 为 Captcha 生成的验证码。其中字符为“02384U”。由于该验证码集合所生成的图片大小一致，灰度一致，不需要进行灰度归一化的预处理操作。输入图片经过滑动窗切片、CNN 层处理，RNN 层处理，经过 CTC 层与样本标签计算损失值。其具体流程如下：

1、滑动窗切片：图片大小为 120×30 像素，滑动窗大小为 5×30 像素，步长为 4 个像素。经过所选用图片后，得到 29 列切片。如图 4-4 所示，数据集文本图片在输入端预处理后经过滑动窗分隔得到重叠的切片，之后进入集成网络。

2、CNN 层处理：卷积核大小为 5× 5，步长为 1。对于切片边缘，使用 0 填充来使得卷积后的特征图大小保持 5× 30。对于每一个切片，我们使用 8 个卷积核提取特征，加上偏置项后使用 Relu 函数激活，得到 8 个特征图。为了减小计算量，特征图经过最大化池化，池化大小为 2× 2，步长为 1。池化后特征图减小为 3× 15。图 4-4 红线框内展示了部分切片数据经过卷积层后的特征图。

3、RNN 层处理：CNN 层的对每个图像切片提取特征后，得到 8 个特征图。RNN 的输入为 8×3× 15。经过线性转换后，将 8 个特征图转换为 1× 100的向量以适应 RNN 单元的输入。RNN 层有一个隐层，每层单个 RNN 单元含有 100 个神经元。每个 RNN 单元产生 1*100 的向量，经过线性转换为各个 RNN 单元的分类结果。图 4-4 中展示了部分特征图经过 RNN 单元后得到的分类结果，其中包含空白标签。

4、CTC 解码：RNN 层输出的分类结果数有 29 个。通过与标签计算损失值 loss。利用批量梯度更新（stochastic gradient descent，SGD）来进行反向传播（Backpropagation algorithm, BP）来进行梯度更新。图 4-4 中 CTC 解码层得到 RNN 层的输出结果后，与真实标签计算损失值，通过反向传播来更新 RNN 层与 CNN 层的网络参数

4.3.2 测试过程

训练网络时，每迭代一定轮次（本实验中为 50000 次）进行一次测试，观察网络的训练情况。训练时只需要输入样本，经过 CNN 层提取特征，RNN 层预测结果序列，CTC 解码后，与真实标签比较，得出正确率。

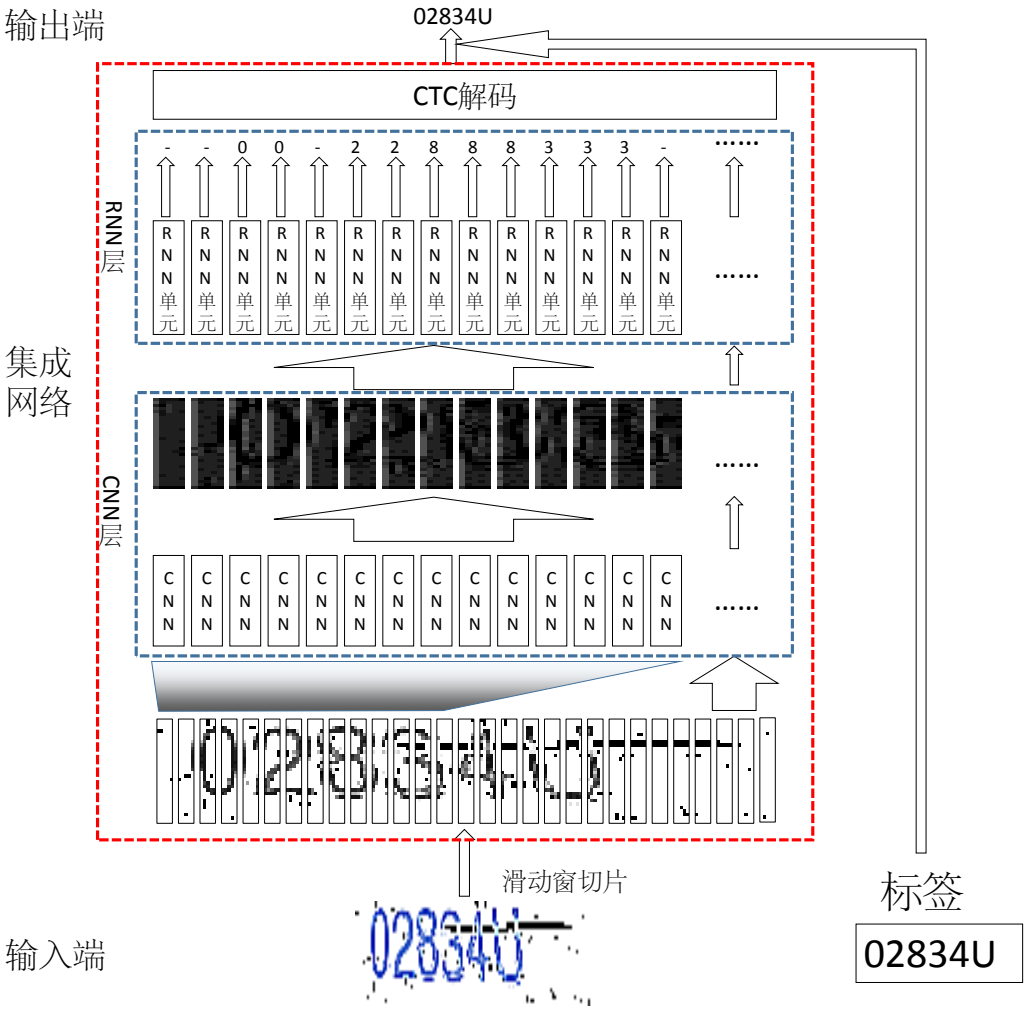


图 4-4 算法流程图

4.4 实验结果分析

4.4.1 CNN-RNN 算法与 RNN 算法收敛速度比较

本文首先对验证码数据集进行训练。针对验证码数据集，分别使用了 RNN 模型与本文提出的 CNN 结合 RNN 的新模型训练，均迭代 8400 次。损失值下降情况如图 4-5 所示。

图 4-5 中橘色表示 CNN-RNN 模型的 train-loss 曲线，淡蓝色表示传统 RNN 模型 train-loss 曲线。的的本文提出的 CNN-RNN 网络较单纯 RNN 网络收敛速度明显加快。最终 CNN-RNN 模型的正确率为 98.2%，RNN 模型的实验正确率为 92.5%。

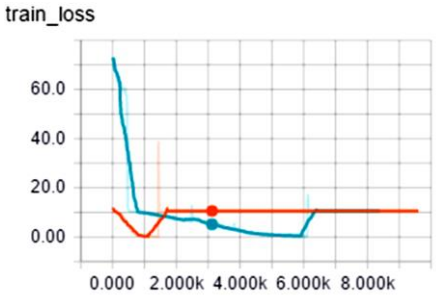


图 4-5 验证码数据集训练 Loss 曲线

该模型训练耗时 21 个小时。两组实验的精度大致接近，损失值下降情况如图 4-5 所示。

本文继续针对 SVHN 数据集进行训练。分析图像归一化后不同的大小、RNN 序列长度对结果的影响。

4.4.2 实验参数对结果影响

本文还研究了三种不同的实验参数设置下 CNN-RNN 的网络模型的 train-loss 曲线异同，实验采用 SVHN 数据集。实验设置 1 和 2 通过选用不同的滑动窗来获取不同数量的切片来适应各自的 RNN 网络的序列长度。实验设置 1 与实验设置 3 通过将图片归一化为不同的大小，来对比不同的归一化大小对实验的影响。详细的实验设置见表 4-2 所示。

表 4-2 实验参数设置

	参数设置 1	参数设置 2	参数设置 3
图像大小（像素）	40×81	40×81	40×39
滑动窗大小（像素*像素）	3×81	5×81	3×81
滑动窗步长（像素）	3	4	3
RNN:time_step	27	20	13
精度	0.90	0.88	0.93

采用较小的滑动窗可以获取较多的图像切片，从而使用更长的 RNN 序列。对比图 4-6 中的 a、b 两图可以发现，较小的滑动窗实验收敛速度更快，train-loss 曲线更加平滑。对比 4-6 中 a、c 两图，对图片进行合适的归一化也可以很大程度上提高收敛速度与最后模型的精确度。

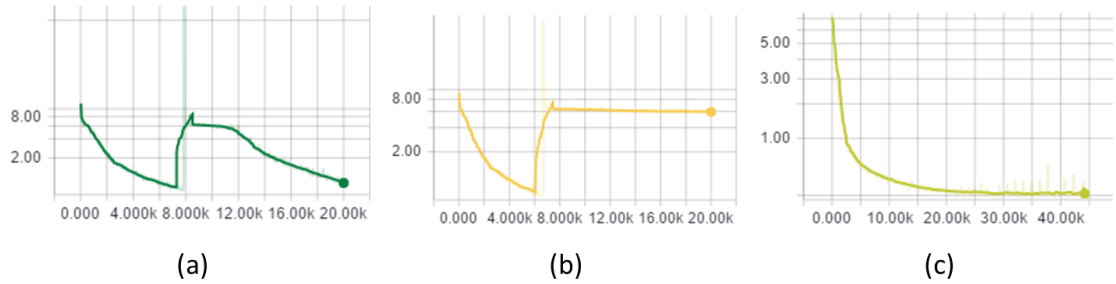


图 4-6 不同参数设置下的训练 loss 曲线

4.5 本章小结

本章主要简要介绍了常用的深度学习框架和自然场景文本识别实验的实验环境配置。并介绍了本章实验所用的实验数据集，以及训练网络模型的训练与测试过程。对验证码数据集验证了 CNN-RNN 网络对传统 RNN 网络的收敛速度以及准确度的提升。并在 google 街景门牌号码数据集上针对不同的滑动窗步长、RNN 序列长度、数据集图片归一化大小进行实验，分析了不同的网络超参数对实验精度的影响。

第五章　全文总结与展望

5.1 总结

近年来，深度学习技术成为模式识别领域的研究热点，并取得了惊人的成绩。文本识别作为模式识别与计算机视觉领域经典研究课题，已经取得了长足的发展。而随着各种智能设备的广泛应用，新型数字图像采集设备也在不断出现，比如手机摄像头、数码相机、交通摄像头等。移动互联网越来越普及，各种信息类型以及信息量激增。文本识别的应用场景在不断发展。自然场景下的文本识别含有丰富的语义信息，对于场景理解、人机交互以及自动化控制有着极其重要的意义。本文提出的使用 CNN 结合 RNN 的方法可以将一幅图像直接作为输入，将特征提取与分类器识别结合，实现了端到端的训练。

本文的工作主要包含以下几个方面：

（1）总结和概括了人工神经网络，深度学习和 RNN 的发展历史，分析了深度学习对人工神经网络的影响。

（2）介绍了 CNN 和 RNN 的算法思想和结构，并推导了 CNN 和 RNN 的训练过程。

（3）提出了基于 CNN 结合 RNN 的文本识别方法。利用 CNN 提取图像的高层语义特征，结合 RNN 捕获图像全局序列信息。最后利用 CTC 解码技术实现识别文本输出。只需要输入文本图片与标签即可完成对网络模型的训练，实现了对网络模型的端到端训练。

（4）利用设计的 CNN-RNN 网络，对验证码图像和 Google 街景门牌号码图像进行识别分类，并验证了 CNN-RNN 网络与传统 RNN 网络在收敛速度与识别精度上的不同。证实了 CNN-RNN 网络好于传统的 RNN 网络。

（5）通过对滑动窗大小、RNN 序列长度等的不同设置，研究不同的超参数对识别精度的影响。结果证明较小的切片和较长的 RNN 序列更能取得更好的实验准确率。

5.2 展望

深度学习经过近些年的发展，在人脸检测、物体检测、人脸识别、文本识别等领域被广泛应用。随着 GPU、分布式计算等技术的发展，以及大数据的便捷获取，深度学习会在更多领域得到应用。基于自然场景的文本识别是一个复杂的问题，现有的技术仍然不能够被大规模商业化应用。以后可能会在一下几个方面对算法进行改进。

（1）提升算法的鲁棒性。本文数据集相对现实场景的数据仍然是较小的。而更大规模的数据集往往更能训练出泛化能力更强的模型。

（2）本文中算法时间复杂度和空间复杂度都较高。训练时使用显存占用高达5.8GB，训练时间长达21小时。未来的研究会尝试减小计算量，使用预训练的CNN来提取特征，以期能够有更好的效果。

（3）中文字符类别相对英文类别更多，字符随意性更大，目前本文提出的CNN=RNN网络对中文手写体识别的识别精度不高。考虑通过加深CNN层数，使用其他的RNN网络比如GRU等来对中文手写体进行识别研究。

（4）本文中的CNN-RNN网络不仅可以在自然场景下的文本识别有较好的效果，还可以应用于手势识别、步态识别、天气预报与表情识别等任务中。接下来的研究考虑将本文的CNN-RNN网络应用于相关领域的研究。

参考文献

[1] HerbertF Schantz. History of OCR,Optical Character Recognition. Recognition Technologies Users Association, 1982.
[2] Shyang-Lih Chang,Li-Shien Chen,Yun-Chung Chung,and Sei-Wan Chen,Automatic License plate recognition. IEEE Transactions on Intelligent Transportation Systems,5(1):42-53,2004
[3] Gustav Tatwchek. Reading machine,December 31 1935. US Patent 2,026,329.
[4] Behzad Shahraray and David C Gibbon. Automated authoring of hypermedia documents of video programs. In ACM International Conference on Multimedia, pages 401-409,1995
[5] M. Merler, C. Galleguillos, and S. Belongie. Recognizing groceries in situation using in vitro training data. In SLAM,2007.
[6] T. de Campos, B. Babu, and M. Varma. Character recognition in natural images. In VISAPP, Feb. 2009.
[7] Toshio Sato, Takeo Kanade, Ellen K Hughes, and Michael A Smith. Video ocr for digital news archive. In IEEE International Workshop on Content-Based Access of Image and Video Database, pages 52-60,1998.
[8] 赵志宏，杨绍普，马增强. 基于CNNLeNet-5的车牌字符识别研究[J]. 系统仿真学报，2010，03：638. 641.
[9] Rodolfo Zunino and Stefano Rovetta. Vector quantization for license-plate location and image coding. IEEE Transactions on Industrial Electronics,47(1):159-167,2000.
[10] Ramos S, Gehrig S, Pinggera P, et al. Detecting Unexpected Obstacles for Self-Driving Cars: Fusing Deep Learning and Geometric Modeling[J]. arXiv preprint arXiv:1612.06573, 2016.
[11] F.Rosenblatt. The Perceptron: a probabilistic model for information storage and organization in the brain. Psychological review,vol 65,pp.386-404,1958.
[12] W.McCulloch and W.Pitts. A logical calculus of the ideas immanent in nervous activity,bulletin of Mathematical biophysics,vol.10,pp. 115-133,1943.
[13] Werbos P J. Beyond regression: new tools for prediction and analysis in the behavioral sciences[D]. Boston: Harvard University,1974.
[14] Rumelhart D, Hinton G, Williams R. Learning representations by back-propagating errors[J]. Nature,1986,323:533-536.
[15] Hopfield J J. Artificial neural networks[J]. IEEE Circuits and Devices Magazine, 1988, 4(5): 3-10.
[16] Glorot X, Bordes A, Bengio Y. Deep Sparse Rectifier Neural Networks[C]//Aistats. 2011, 15(106): 275.

[17] Nair V, Hinton G E. Rectified linear units improve restricted boltzmann machines[C]//Proceedings of the 27th international conference on machine learning (ICML-10). 2010: 807-814.
[18] Glorot X, Bordes A, Bengio Y. Domain adaptation for large-scale sentiment classification: A deep learning approach[C]//Proceedings of the 28th international conference on machine learning (ICML-11). 2011: 513-520.
[19] 徐春晖, 徐向东. 前馈型神经网络新学习算法的研究[J]. 清华大学学报: 自然科学版, 1999, 39(3): 1-3.
[20] Williams R J, Zipser D. A learning algorithm for continually running fully recurrent neural networks[J]. Neural computation, 1989, 1(2): 270-280.
[21] Werbos P J. Backpropagation through time: what it does and how to do it[J]. Proceedings of the IEEE, 1990, 78(10): 1550-1560.
[22] Williams R J, Peng J. An efficient gradient-based algorithm for on-line training of recurrent network trajectories[J]. Neural computation, 1990, 2(4): 490-501.
[23] Bengio Y, Simard P, Frasconi P. Learning long-term dependencies with gradient descent is difficult[J]. IEEE transactions on neural networks, 1994, 5(2): 157-166.
[24] Hochreiter S, Schmidhuber J. Long short-term memory[J]. Neural computation, 1997, 9(8): 1735-1780.
[25] Hubel D H, Wiesel T N. Receptive fields, binocular interaction and functional architecture in the cat’s visual cortex[J].The journal of physiology,1962,160(1):106.
[26] Chen L, Wang S, Fan W, et al. Cascading Training for Relaxation CNN on Handwritten Character Recognition[C]//Frontiers in Handwriting Recognition (ICFHR), 2016 15th International Conference on. IEEE, 2016: 162-167.
[27] Graves A. Supervised sequence labelling[M]//Supervised Sequence Labelling with Recurrent Neural Networks. Springer Berlin Heidelberg, 2012: 5-13.
[28] Graves A, Mohamed A, Hinton G. Speech recognition with deep recurrent neural networks[C]//Acoustics, speech and signal processing (icassp), 2013 ieee international conference on. IEEE, 2013: 6645-6649.
[29] Graves A, Fernández S, Gomez F, et al. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks[C]//Proceedings of the 23rd international conference on Machine learning. ACM, 2006: 369-376.
[30] Jia Y, Shelhamer E, Donahue J, et al. Caffe: Convolutional architecture for fast feature embedding[C]//Proceedings of the 22nd ACM international conference on Multimedia. ACM, 2014: 675-678.
[31] Bhaskar J, Patel A. Image Classification using Convolutional Neural Network[J].

[32] Krizhevsky A, Sutskever I, Hinton G E. Imagenet classification with deep convolutional neural networks[C]//Advances in neural information processing systems. 2012: 1097-1105.
[33] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition[J]. arXiv preprint arXiv:1409.1556, 2014.
[34] Srivastava N, Hinton G E, Krizhevsky A, et al. Dropout: a simple way to prevent neural networks from overfitting[J]. Journal of Machine Learning Research, 2014, 15(1): 1929-1958.
[35] Zaremba W, Sutskever I, Vinyals O. Recurrent neural network regularization[J]. arXiv preprint arXiv:1409.2329, 2014.
[36] Goodfellow I J, Bulatov Y, Ibarz J, et al. Multi-digit number recognition from street view imagery using deep convolutional neural networks[J]. arXiv preprint arXiv:1312.6082, 2013.
[37] Abadi M, Agarwal A, Barham P, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems[J]. arXiv preprint arXiv:1603.04467, 2016.
[38] LeCun, Yann, et al. "Comparison of learning algorithms for handwritten digit recognition." International conference on artificial neural networks. Vol. 60. 1995.
[39] Jia, Yangqing, et al. "Caffe: Convolutional architecture for fast feature embedding." Proceedings of the 22nd ACM international conference on Multimedia. ACM, 2014.
[40] Mitra, Bhaskar, Fernando Diaz, and Nick Craswell. "Luandri: a Clean Lua Interface to the Indri Search Engine." arXiv preprint arXiv:1702.05042 (2017).
[41] Chen, Tianqi, et al. "Mxnet: A flexible and efficient machine learning library for heterogeneous distributed systems." arXiv preprint arXiv:1512.01274 (2015).
[42] Abadi, Martín, et al. "Tensorflow: Large-scale machine learning on heterogeneous distributed systems." arXiv preprint arXiv:1603.04467 (2016).
[43] Mishra, Anand, Karteek Alahari, and C. V. Jawahar. "Scene text recognition using higher order language priors." BMVC 2012-23rd British Machine Vision Conference. BMVA, 2012.
[44] Zimmermann, Grant R., Joseph Lehar, and Curtis T. Keith. "Multi-target therapeutics: when the whole is greater than the sum of the parts." Drug discovery today 12.1 (2007): 34-42.
[45] Rodriguez-Serrano, Jose A., Florent Perronnin, and France Meylan. "Label embedding for text recognition." Proceedings of the British Machine Vision Conference. 2013.
[46] Almazán, Jon, et al. "Word spotting and recognition with embedded attributes." IEEE Transactions on Pattern Analysis and Machine Intelligence 36.12 (2014): 2552-2566.

[47] Matan, Ofer, et al. "Multi-digit recognition using a space displacement neural network." NIPS. 1991.
[48] Jaderberg, Max, et al. "Reading text in the wild with convolutional neural networks." International Journal of Computer Vision 116.1 (2016): 1-20.
[49] Bissacco, Alessandro, et al. "Photoocr: Reading text in uncontrolled conditions." Proceedings of the IEEE International Conference on Computer Vision. 2013.
[50] Perronnin, Florent, et al. "Large-scale image retrieval with compressed fisher vectors." Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on. IEEE, 2010.
[51] Miao Y, Gowayyed M, Metze F. EESEN: End-to-end speech recognition using deep RNN models and WFST-based decoding[C]//Automatic Speech Recognition and Understanding (ASRU), 2015 IEEE Workshop on. IEEE, 2015: 167-174.
[52] K.Jung,K.I.Kim and A.K.Jain,Text information extraction in images and Video:A survey[J],Pattern Recognition,2004,37(5):997-997

致 谢

时光如梭，转眼之间研究生生涯就要过去。回过头看一下三年来的点点滴滴，心中感到非常的充实。在科研与生活上，老师和同学们都给予了我很大的帮助，在这里我要对他们表示深深的感谢。

衷心地感谢我的导师朱远平副教授。感谢朱老师带领我走进科研的大门。我对数字图像处理领域一无所知的时候，是朱老师耐心地引导我；在我的科研受挫的时候，朱老师给我支持与帮助。正是在朱老师的悉心指导下，我才得以顺利完成本文。他求是严谨的治学态度、平易近人的性格对我影响很大。每次和朱老师的交流与讨论都会让我有新的领悟与发现，使我的科研可以顺利地进行下去。祝愿朱老师事业更上一层楼。

其次要感谢中国科学院自动化研究所复杂系统管理与控制国家重点实验室的王春恒老师和肖百桦老师对我的指导。在我学习与科研中，我的师兄祁成作帮助我一起不断地调试程序。一次次的谈心让我转换新的思路，帮助我走出困惑，特别的感谢他。

感谢我的姐姐和姐夫对我论文的指导，他们一次一次的帮我修改论文，查阅文献，帮助我润色论文。感谢我的妻子在我无数次熬夜工作的时候默默的陪伴我，在我情绪低落的时候悉心的照顾与支持，让我迈向下一个目标。你是我坚强的后盾，给了我奋斗的力量。

感谢我的同学张矿、张京杰、袁源、刘荣轩和张砚硕，正是有了你们，我的研究生生活充满了这么多的乐趣。

最后，感谢评审论文和出席本次答辩的诸位老师们在百忙之中提出宝贵的意见！