

東南大學

毕业设计(论文)报告

题 目 基于卷积神经网络的一般物体识别

自动化 院（系） 自动化 专业

学 号 08013334

学生姓名 于容谦

指导教师 李新德

起止日期 2016.12.20-2017.6.5

设计地点 中心楼 401

东南大学毕业（设计）论文

独创性声明

本人声明所呈交的毕业（设计）论文是我个人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得东南大学或其它教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示了谢意。

论文作者签名：_____日期：_____

东南大学毕业（设计）论文使用

授权声明

东南大学有权保留本人所送交毕业（设计）论文的复印件和电子文档，可以采用影印、缩印或其他复制手段保存论文。本人电子文档的内容和纸质论文的内容相一致。除在保密期内的保密论文和在技术保护期限内的论文外，允许论文被查阅和借阅，可以公布（包括以电子信息形式刊登）论文的全部或中、英文摘要等部分内容。论文的公布（包括以电子信息形式刊登）授权东南大学教务处办理。

论文作者签名：_____导师签名：_____日期：_____

基于卷积神经网络的一般物体识别

08013334 于容谦

指导教师 李新德

摘要

一般物体识别（General Object Recognition, GOR）是计算机视觉（Computer Version）中的一个特别重要的研究方向，其期望计算机像我们人类一样，无论在任何环境下，对于任意的物体都有较强，正确率较高的检测与识别能力。这篇论文介绍了一种采用卷积神经网络的对于一般物体的识别方法。近年来，卷积神经网络越来越成为物体识别的主流方法。本文在计算机视觉（computer version）领域，结合深度学习（deep learning）理论，首先在网络上采集数据库，搭建简单的小型数据库，共 10 类物体。基于卷积神经网络的原理,结合实现卷积神经网络的 MATLAB 实用工具箱-Mat Conv Net，搭建 LeNet-5 卷积神经网络框架，实现一般物体的识别。卷积神经网络算法对于多种形式的变形都具有高度不变的性质，比如图像的倾斜变形与平移变形，这种算法属于人工神经网络的一部分，在语音辨别与图像处理领域表现极为突出。通过此次研究，证明在数据库比较小的情况下，这种方法是行之有效的。同时，Mat Conv Net 也可以通过图形处理器（GPU）的加速来提高效率。完成了卷积神经网络功能实现，并能对检测物体实现较高的识别率（90%）。

关键词：一般物体识别；人工神经网络；卷积神经网络；线性和非线性滤波；机器学习；深度学习；MATLAB；Mat Conv Net；图形处理器加速。

General Object Recognition based on Convolution Neural Networks

08013334 YU RONGQIAN

Supervised by LI XINDE

Abstract

General Object Recognition (GOR) is a particularly important research direction in Computer Vision, which expects computers to be as strong as human beings, in any environment, for any object with higher detection and recognition capabilities. This paper presents a method for recognizing general objects using convolution neural networks. In recent years, convolution neural networks are increasingly known as the main methods of object recognition. In the field of computer vision, combined with the deep learning theory, we first collect the database on the network, build a simple small database, a total of 10 categories of objects. Based on the principle of convolution neural network, combined with the realization of convolution neural network MATLAB practical toolbox -Mat Conv Net, build LeNet-5 convolution neural network framework, to achieve the identification of the general object. Convolution neural network algorithm is highly invariant for various forms of deformation, such as the image of the tilt deformation and translation deformation, this algorithm is part of the artificial neural network, in the field of speech recognition and image processing is extremely prominent. Through this study, it is proved that this method is very effective when the database is relatively small. At the same time, Mat Conv Net can also be through the graphics processor (GPU) to accelerate to improve efficiency. Completed the convolution neural network function, and can detect the object to achieve a higher recognition rate (90%).

Key words: general object recognition; artificial neural network; convolution neural network; linear and nonlinear filtering; machine learning; depth learning; MATLAB; Mat Conv Net; graphics processor acceleration.

目 录

摘要.....	I
Abstract.....	II
第一章 介绍.....	1
1.1 研究背景与意义.....	1
1.2 国内外研究现状.....	1
第二章 物体识别过程.....	4
2.1 数据预处理（data preprocessing）.....	4
2.2 特征提取.....	5
2.3 分类器.....	5
第三章 卷积神经网络.....	7
3.1 发展历史.....	7
3.2 整体网络模型结构.....	7
3.3 前向传输计算.....	10
3.3.1 输入层（Input layer）.....	10
3.3.2 卷积层（Convolutional layer）.....	10
3.3.3 下采样层（Subsampling layer, Pooling layer）.....	11
3.4 反向传输调整权重.....	11
3.4.1 输出层残差.....	12
3.4.2 卷积层的残差（当下一层为采样层时）.....	12
3.5 卷积神经网络训练过程.....	12
3.5.1 向前传播阶段：.....	13
3.5.2 向后传播阶段.....	13
第四章 Mat Conv Net 模型.....	16
4.1 Mat Conv Net 介绍.....	16
4.2 Mat Conv Net 框架搭建.....	16
第五章 程序设计与实现.....	19
5.1 程序流程设计.....	19
5.2 程序编写.....	19
5.3 程序结果.....	27
5.3.1 不同训练结果对比.....	27
5.3.2 每个类型物体识别率对比.....	28
5.3.3 是否采用图形处理器加速对比.....	28
第六章 全文总结与结论.....	30
致谢.....	32
参考文献.....	33

第一章 介绍

1.1 研究背景与意义

一般物体识别 (General Object Recognition) 是计算机视觉 (Computer Vision) 中的一个特别重要的研究方向, 其期望计算机像我们人类一样, 无论在任何环境下, 对于任意的物体都有较强, 正确率较高的检测与识别能力。这项技术要求在给定训练样本的前提下, 机器能够自主对样本进行有效、全面而具体的学习, 也就是机器学习 (machine learning), 并在观察相同类别物体时, 对于新的结构与外观不同的物体时, 能够做出准确的判断。在科学技术不断发展的当今社会, 一般物体识别在高科技等多个方面起着越来越重要的任务。有许多国内外的知名学者正在从事此方面的研究。因此可以说, 此项技术处在一个快速发展的初级阶段, 而在传统的特定物体识别 (Instance Recognition) 领域, 如人脸识别, 发展时间长, 目前已经有较为成熟的解决办法。但设计一套适用于一般环境中的一般物体的识别系统仍然具有很大的难度, 其通用且稳定的理论与算法框架还未出现, 其中影响的重要因子之一就是复杂多变的背景, 有时提供虚假有效信息, 有时则遮挡有效信息, 严重影响了机器的有效学习与识别。^{[1][3][13]}

而伴随着深度学习技术发展步与成长, 卷积神经网络 (Convolutional Neural Network, CNN), 在近阶段, 在图像识别领域取得了巨大的突破与进展, 与此同时, 在世界范围内, 各大机器学习的实验室也慢慢开始关注这个领域。

卷积神经网络算法有很大的优点, 其对于多种形式的变形都具有高度不变的性质, 比如图像的倾斜变形与平移变形, 这种算法属于人工神经网络的一个分支, 在语音识别与图像处理领域表现极为突出, 其具备类似于生物神经网络 (BNN, Biological Neural Networks) 的权值共享网络结构。网络的输入是图形、图像、语音、视频等信息, 而在传统的神经网络中, 需要进行复杂的特征提取工作, 并要进行数据的重建过程。当输入为多维图像时, 卷积神经网络的优点更加突出, 可以大幅度减少时间成本。^[2]

1.2 国内外研究现状

继承平移变形等高度不变特性与生物神经网络的思想, 诞生了卷积神经网络。对于手写体数据库 (MINST), 当今最高识别率的算法就是来源于卷积神经网络。而远在 1998 年, 纽约大学终身教授、卷积神经网络之父-Yan LeCun 结合反向传播 (Back Propagation) 算法与梯度下降 (gradient decent) 优化算法, 第一次对于手写体数据库训练卷积神经网络, 效果极佳, 在当时引起轰动, 也第一次将卷积神经网络应用于机器学习领域。但由于当时的科技发展水平与处理器硬件的限制, 加上理论分析繁琐复杂, 网络的训练的方法缺乏技巧与经验, 卷积神经网络的研究一度停滞不前, 无法取得进步。直到近年来, 随着深度学习的发展与进步, 卷

积神经网络的身影又一次展现在众多研究人员的面前，成为了当下深度学习领域的热门。^[1]

在人工神经网络的研究的基础上，在 2006 年，Geoffrey Hinton 与其深度学习小团队共同研究，改进了传统的神经网络，使其更接近大脑的工作方式，从而更好地模拟大脑，进而第一次提出了深度学习的概念。上文提到，深度学习理论近年来在机器学习领域占领了越来越重要的地位，与传统的学习方法相比较，笔者认为，深度学习有两点明显的优势：

- (1) 更适用于大数据：在当今这个信息时代，海量数据处理迫在眉睫，过时信息等于无效信息。传统机器学习方法已经不能满足信息的即时获取、即时处理与即时发布，但深度学习能够明显提高学习效率。
- (2) 端到端机器学习系统（end-to-end learning system）：如上文提到，“网络的输入是图形、图像、语音、视频等信息”，可以通过深度学习直接将输入转化成文本信息，这是传统神经网络模型（如循环神经网络）所不能实现的。

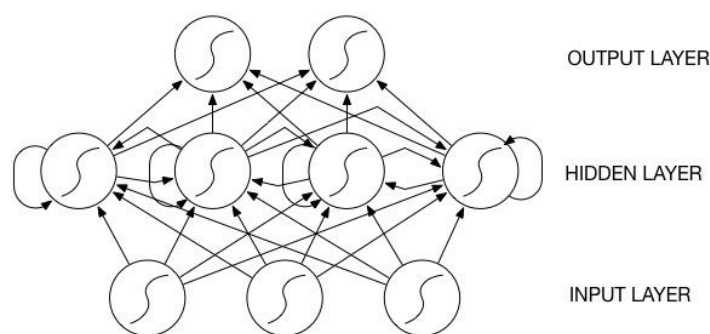


图 1.1 循环神经网络（RNN）图示

卷积神经网络可谓是深度学习的代表性网络，所以具备了深度学习的众多优点，而在物体识别领域，相比传统方式，有了更加长足深远的进步。

在科技最前沿，目前国际上公认的规范数据库竞赛是 ILSVRC（ImageNet Large Scale Visual Recognition Challenge），自 2012 年到 2015 年的冠军分别是：Alexnet、R-CNN、Googlenet、MSR，这些网络无一例外全部采用的是卷积神经网络。同时，众所周知，目前机器视觉领域三大世界顶级会议为：IEEE 国际计算机视觉与模式识别会议（CVPR, Conference on Computer Vision and Pattern Recognition）、国际计算机视觉会议（ICCV, International Conference of Computer Vision）与欧洲计算机视觉国际会议（ECCV, European Conference on Computer Vision），据不完全统计，在物体识别领域，三大会议的录用论文有将近百分之八十是基于深度学习领域的，而这些录用论文采用的网络结构，卷积神经网络占绝对的主导地位，这就说明，卷积神经网络无疑是目前最适合做物体识别算法的架构。可以这么说，由于卷积神经网络的普及，在未来，物体识别领域，仍然具有

无限可能。

毫无疑问，当今世界各大高科技公司也在深度学习领域展开激烈竞争，各种深度学习框架如雨后春笋一样层出不穷。百度（Baidu）、谷歌（Google）与脸谱网（Facebook）等大型互联网公司、企业都在研究开发属于自己的深度学习库。而对于我们普通开发者或者卷积神经网络的普通爱好者来说，有一个工具大大加速了我们学习的速度，那就是 Mat Conv Net。

说到 Mat Conv Net，它其实就是 MATLAB 的一个工具箱，如果想利用该工具进行开发。任何开发者都可以在其官网上下载到。众所周知，MATLAB 是深受计算机视觉领域研究人员喜爱的开发环境之一，伴随着卷积神经网络的兴起，同时为计算机视觉开发者提供友好高效的开发环境，基于这些动机，MATLAB 平台推出了 Mat Conv Net，其中公开了很多简单且易于调用的卷积神经网络构建块，如卷积（convolution）、归一化（normalization）、池化（pooling）等指令。利用这些命令块进行组合与扩展，可以很容易构建卷积神经网络架构。不同于往常采用低级语言（如 C 语言）编写卷积神经网络，Mat Conv Net 大大简化了构建网络的过程，十分有利于刚接触卷积神经网络的新人的学习与研究。同时，MATLAB 本身可以支持图形处理器（GPU, Graphics Processing Unit）进行加速，Mat Conv Net 也不例外。这就意味着，无论系统多复杂，或者将来需要添加新的命令块，都可以通过图形处理器加速来保证运算效率。在第三节，我会详细介绍 Mat Conv Net 的安装与调试，并利用 Mat Conv Net 搭建卷积神经网络。^{[2][6]}



图 1.2 Mat Conv Net: CNNs for MATLAB

此篇文章是按照如下组织的：卷积神经网络及识别算法的细节在第 2 节与第 3 节进行介绍，第 4 节在介绍 Mat Conv Net 模型，第 5 节采用 MATLAB 编写程序实现目标，观点与结论在第 6 节中给出。

第二章 物体识别过程

实现物体识别的方法有很多，具体的实现过程也是多种多样，在这诸多的方法中，却有一些环节是必不可少的，它们是实现物体识别过程中最为核心的部分：对数据进行预处理，提取特征，分类方法，具体流程见图 2.1。

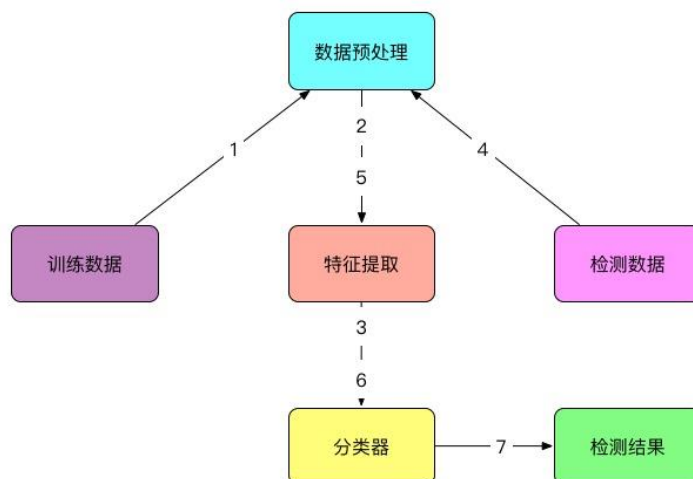


图 2.1 物体识别过程

2.1 数据预处理（data preprocessing）

在所有计算机视觉算法中，预处理几乎都是首先进行的步骤，目的是尽最大可能在不改变图像本身所具备的信息的基础上，让每张图像表现出来的外观特性（比如颜色信息，明暗信息，尺寸信息等）尽可能保持一致，对数据进行预处理，是指对原始的数据进行处理，通过对数据进行一定的操作，使得数据能够更优地为后续的操作分析做准备。在这个过程中，实际上是对输入数据进行优化，当对某一个算法或者某一类数据库进行实验的时候，往往需要对数据进行预处理。

对原始数据的预处理往往与要处理的问题与采样设备相关，以车牌识别为例，如果要对车牌进行识别，就要将车牌上的字母与数字信息提取出来，然后对其进行分割识别，当完成了这些操作之后，才能对数字与字母进行识别，可见数据预处理的必要性与重要性。而有的数据自身所带的特征已经具有理想的性质，就不必要进行数据预处理，这种情况不是很常见。

在数据预处理的领域，此操作的必要性有两点：

- （1）高质量的数据挖掘工作需要同样高质量的数据进行支持；
- （2）如果不对数据进行预先处理，那么原始的数据不光不能起到效果，而且还会对训练过程造成很大影响。比如原始数据往往会有残缺的数据、比较高的噪声值等等，如果将未经处理的这些数据输入到系统中去，会有很大的干扰。

在数据预处理领域，可以采用的方法可以分为以下几类：

- (1) 降噪操作：删除原始数据中的噪声；
- (2) 提取特定内容：针对某些特定要求提取某些特定内容；
- (3) 采样：在原始数据中采样代表子集；
- (4) 标准化：用于特定输入要求，往往为了满足后续算法。

2.2 特征提取

在物体识别的过程中，特征提取可谓是最为重要的部分，同时也是最难控制的一步。在分类器的性能方面，会主要取决于原始样本与特征之间映射性能的好坏，并且，特征维数也间接决定了分类器的设计。不良特征往往会导致错误识别或者遗漏识别，而良好特征会保证系统将不同类别的数据区分开来，会显著提高识别效率，为之后的处理节省了时间。一般来说，对于一副图像，它的内部信息通常由颜色信息、纹理信息、形状特征以及空间关系所构成，结合这些特征，在已有神经网络中加入采用交替布置的卷积层与采样层，然后加上全连接层，来进行特征提取。经过每一层的处理后，提取出来的特征也是不同的，在网络的较低的层上，提取出的特征往往比较简单，维数也比较少，一般称为全局特征，比如轮廓特征；随着层数的加深，提取出的特征维数也会逐渐升高，一般称为局部特征。

一般来说，图像的特征包括以下四种：

- (1) 直观性特征：主要是指几何特征，该特征的特点是较为稳定，当人脸的姿态发生变化时，特征的变化较小，同时，光照的影响也比较小，但测量精度往往比较低，而且不容易被提取出来；
- (2) 灰度统计特征：指图像的灰度值特征；
- (3) 变换系数特征：指图像的变换过程中所经过的处理，往往将变换的程度称为变换系数；
- (4) 代数特征：指根据统计学习的方法提取出来的特征，具有很高的识别精度。提取代数特征的方法分为两种，线性投影特征提取与非线性投影特征提取方法。这也是目前主流的特征与特征提取方法。

2.3 分类器

这里将介绍一下 softmax 分类器。在机器学习的进程中，通常会通过已经训练好的分类器对新数据进行分类。在介绍逻辑回归之前首先介绍一下线性回归。

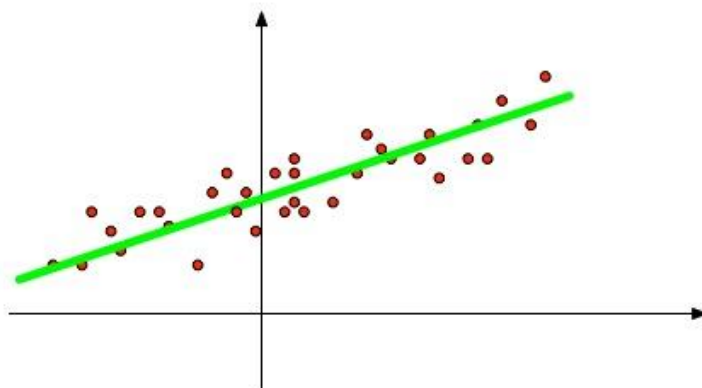


图 2.2 单变量线性回归模型

上图 2.2 展示了一种单变量线性回归模型。其中红色的圆点代表的是单变量的空间分布，可以看出此单变量在空间内呈线性分布，所以我们可以用公式 2-1 进行线性拟合。

$$\hat{y} = wx + b \quad (\text{式 2-1})$$

而逻辑回归的概念，简单来看，就是在线性回归的基础上加上了一个 sigmoid 函数。

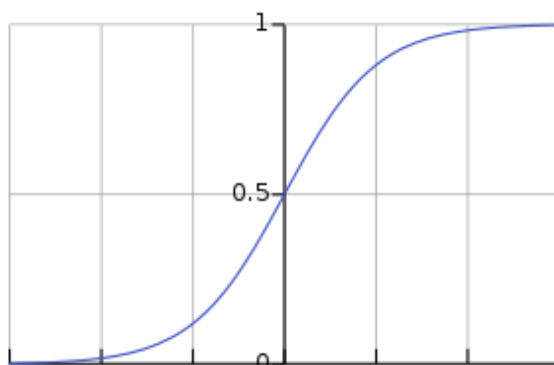


图 2.3 sigmoid 函数

softmax 回归是对于多类别情况的逻辑回归，将 sigmoid 函数作为逻辑回归的假设函数 $h_{\theta}(x)$ ：

$$h_{\theta}(x) = \frac{1}{1 + \exp(-\theta^T x)} \quad (\text{式 2-2})$$

sigmoid 函数作为激活函数经常出现在神经网络当中，如图 2.3 所示，它的值域在 0 到 1 之间， θ 是逻辑回归函数的参数， $y^{(i)}$ 是类别，优化的过程就是找到合适 θ 的值得过程，这个过程一般通过定义代价函数以及最小化代价函数完成，代价函数为 $J(\theta)$ ：

$$J(\theta) = -\frac{1}{m} \left| \sum_{i=1}^m y^{(i)} \log h_{\theta}(x^{(i)}) + (1 - y^{(i)}) \log(1 - h_{\theta}(x^{(i)})) \right| \quad (\text{式 2-3})$$

最小化 $J(\theta)$ 时，常采用梯度下降的算法，在不断迭代的过程中更新 θ 的值。

第三章 卷积神经网络

3.1 发展历史

神经网络（Neural Network）的历史可以追溯到 1962 年，在对猫的视觉皮层细胞的早期研究中，Hubel 和 Wiesel 两位学者提出了感受野（receptive field）的概念，根据上述概念，在 1984 年，一名叫做福岛（Kunihiko Fukushima）的日本学者又提出了神经认知机（neocognitron）的概念，这是在感受野的概念自提出之后，第一次应用于人工神经网络（ANN, Artificial Neural Network），也可以看作是卷积神经网络的第一个网络实现，主要应用的方向是手写体数字的识别。之后便是由在之前提到的，卷积神经网络之父 Yan LeCun，构建了第一个严格意义上的卷积神经网络架构。而有他提出的网络架构也被称为 LeNet-5 网络架构，现在来看，LeNet-5 仍然是一个最典型的卷积神经网络架构，由三个大的部分组成：卷积层、池化层、全连接层。LeNet-5 简单来说，工作原理是这样：卷积层可与池化层进行合作，组成若干个卷积组，由卷积组一层一层进行特征的提取，而这一步，可以说是受感受野的启发。然后经由全连接层，完成最后的分类。[24]-[28]

从那之后，深度学习研究一度沉寂，直到 2012 年，AlexNet 在当年的 ImageNet 图像分类竞赛上一举成名。一般情况下，imagenet 图像有大约 1000 个可能的类别，而在每幅图像上，电脑可以同时进行 5 次预测结果，无论哪一次结果预测准确，都算预测准确，反之，如果全部 5 次结果都是错误的，就算失败，这种分类错误率就叫 top5 错误率（top-5 error rate）。而 AlexNet 将 top-5 错误率降低了足足十个百分点，从那之后卷积神经网络重新回到人们的视野。

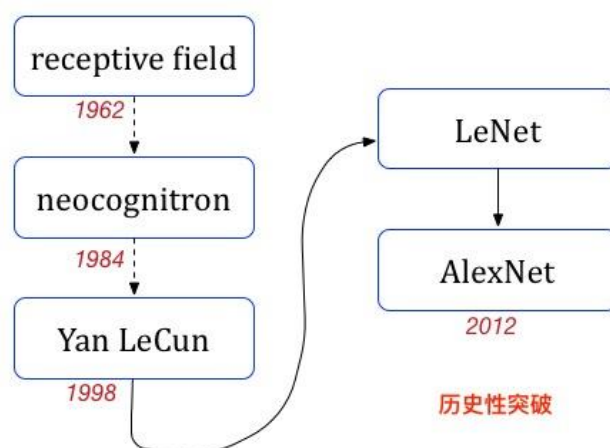


图 3.1 卷积神经网络发展历史

3.2 整体网络模型结构

随着深度学习技术的进步与发展，结合卷积神经网络，图像识别领域在近阶段取得了巨大的突破，世界上各大机器学习实验室也慢慢开始关注。

卷积神经网络算法对于多种形式的变形都具有高度不变的性质，比如图像的倾斜变形与平移变形，这种算法属于人工神经网络的一部分，在语音辨别与图像处理领域表现极为突出。卷积神经网络算法具备权值共享网络结构，这种特殊的网络结构类似于生物神经网络。网络的输入是图形、图像、语音、视频等信息，而在传统的神经网络中，需要进行复杂的特征提取工作，并要进行数据的重建过程。当输入为多维图像时，卷积神经网络的优点更加突出，可以大幅度减少时间成本。

传统的网络识别模型是采用一个由人工设计的特征提取器（feature extractor）收集输入数据的有效信息，同时屏蔽其他信息，从而得到一组特征向量（feature vector），这些特征向量就可以输入到一个可以加以训练的分类器（classifier），从而实现分类。卷积神经网络的网络模型有很大不同，这种网络结构组合了特征提取与分类识别，可以直接输入原始图像，特征提取过程隐藏在网络之中，直接输出分类结果。

卷积神经网络是对于反向传播神经网络（BPNN, back propagation neural network, 如图 3.2）的进一步改进，共同特点是：

- （1）对计算输出值进行前向传播；
- （2）对权重与偏置的调整进行反向传播。

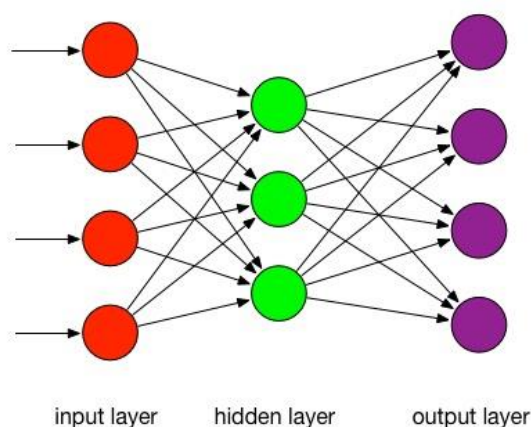


图 3.2 反向传播神经网络

而最大的不同点在于，卷积神经网络神经单元的感知域是上层的部分神经单元而非全部神经单元，也就是相邻层的神经单元是部分连接，而标准的反向传播神经网络相邻层的神经单元是全连接。

卷积神经网络的架构有以下三个公认的重要方面：

- （1）局部区域感知：在标准的反向传播神经网络中，所有的像素点都是混乱的，像素点之间的联系并没有被利用起来；而局部感知可以发现图片上的一条线，或者是一个圆点，这些都是数据的局部特征，再由这些局部基本特征进行组合，就可以构成整体视觉。
- （2）权重共享机制：如下图所示，网络图中有若干个图（map），网络的每

一层都是由若干个图所组成，每个图都包含若干个神经单元，这些神经单元都公用同一卷积核（convolution kernel），这里卷积核也可以理解为权重（weight），它往往代表一个特征，打个比方来说，如上文提到，如果把图片上的一条线作为一个卷积核，那么利用卷积计算的知识，把卷积核与整个图片进行卷积，卷积值大的区域是一条线的概率就大，而其他位置是一条线的可能性就比较小。当我们用固定大小的权重去匹配图像时，与卷积运算特别相似。而对于标准的反向传播神经网络，其卷积核却不是特定的特征，而是这一层的所有权重，也叫做整体感知。权重共享的机制可以有效减少训练参数。

- (3) 空间或时间上的采样：当某个特征被找到时，我们关心的是与其他的特征的相对位置而不是具体位置，举一个例子，识别“好”字，当我们确定了“女”字后，只需要知道当在“女”字的右边是“子”字，就可以确定是“好”字，至于这个字在整个图像的哪个位置，对于识别来说，是没有必要的，所以，对于变形图片或者是扭曲图片，这种忽略具体位置的识别策略明显优于传统识别方法。所以要进行采样，要忽略特征的具体位置。

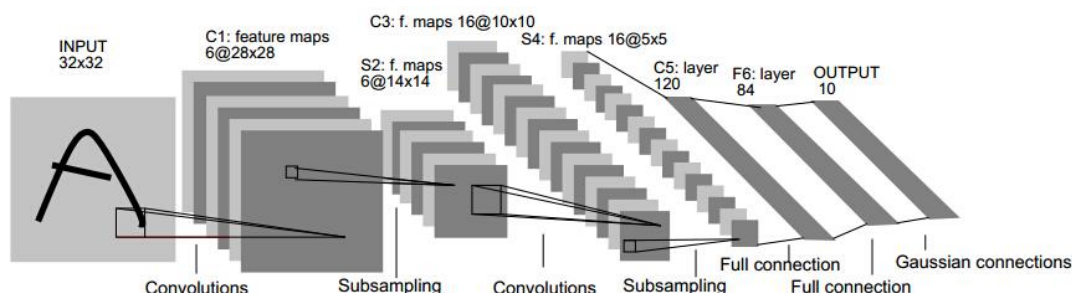


图 3.3 LeNet-5 卷积神经网络网络图^[1]

当存在输入数据存在空间上的扭曲时，比如图像数据的变形，或者是输入数据在时间序列上的变化时，卷积神经网络的以上三个特点可以让识别保持很强的鲁棒性与稳定性。卷积神经网络的一般思路是交替设置下采样层跟卷积层。通过卷积层，可以将特征提取出之后，通过一定的组合，进而描述图片的特征。上图 3.3 是 LeNet-5 卷积神经网络的示例。

以上是实现卷积神经网络的基本思想，但实际情况有很多版本，这里实现的版本中，采样层没有偏置，也没有权重，只是对卷积层进行采样，采用 MINST 作为测试数据集，网络结构如图 3.4。

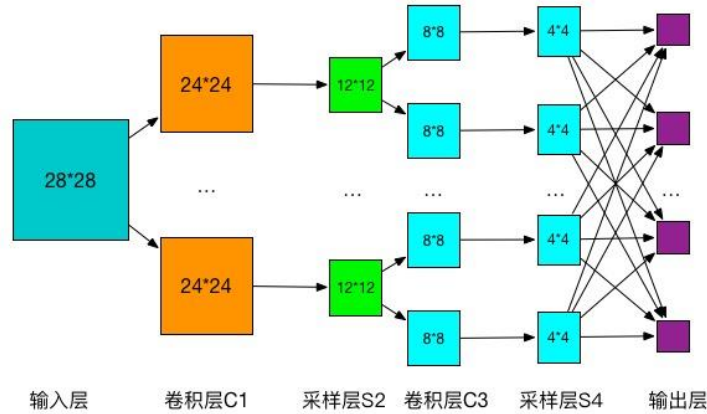


图 3.4 LeNet 卷积神经网络 minst 示意图

3.3 前向传输计算

3.3.1 输入层（Input layer）

输入层不是一个值，而是一个输入向量，为了减少系统的复杂度，一般会使用灰度图像，当输入为 RGB 彩色图像时，需要提前进行归一化处理，如果归一化到[0,1]，需要采用 sigmoid 激活函数，当然也可以采用 tanh 正切三角函数，二者的区别在于，当输入在[-1,1]之间的时候，sigmoid 函数值变化十分敏感，当接近区间边界或者超出区间边界的时候，会进入饱和状态，也就是失去所需要的敏感性能，这样就会大大影响网络预测的精度；而 tanh 正切三角函数的输入有很好的非线性单调上升性与非线性单调下降性，十分适用于后面将要提到的、本系统所采用的梯度下降算法，同时，容错率更高。但是，相比 sigmoid 激活函数，饱和期相对延迟。

图片的尺寸就是向量的大小，由图 3.4 可知，输入为一个 28*28 的矩阵。

3.3.2 卷积层（Convolutional layer）

由图 3.4 可知，卷积层 C1 的输入来自于输入层，而卷积层 C3 的输入来自于采样层 S1，所以，一般来说，卷积层的输入来源于两个部分，一个是输入层，一个是采样层。在权值共享部分提到，在卷积层，每一个图（map）都有一个大小相等的卷积核，在 LeNet 中，输入层到卷积层的卷积核尺寸为 5*5。为了更直观的展示，由图 2.5 进行示例。为了简单说明，只采用 4*4 的 map 与 2*2 的卷积核，进行卷积运算之后，我们可以得到一个 3*3 的特征矩阵，计算过程如式 3-1，

$$(4 - 2 + 1) \times (4 - 2 + 1) = 3 \times 3 \quad (\text{式 3-1})$$

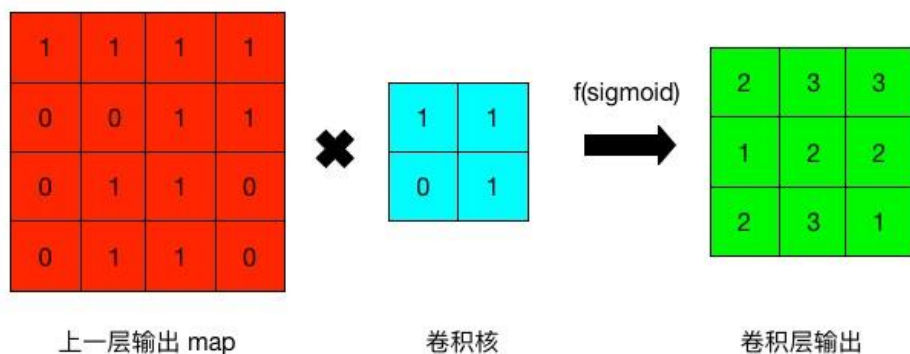


图 3.5 卷积核图解

这就相当于卷积核在上一层的输出 map 上遍历一遍，所以最后的特征 map 是卷积运算之后再将对元素累加，然后加上一个偏置 (bias)，最后进行 sigmoid 运算得到，公式如式 2-2，假设上一层输出为 M_j ，卷积核为 K_j ，输出为 M_{2j} ，

$$M_{2j} = \text{sigmoid}(\text{sum}(M_{ij} \times k_{ij}) + \text{bias}) \quad (\text{式 3-2})$$

特别需要留意的一点是，初始化过程中会决定卷积层的图的个数，但是在卷积层中，图的大小取决于两个方面，卷积核与前层输入的图的尺寸，具体的关系如式 2-3 所示，假设卷积核为 k 阶方阵，上一层输入的 map 为 n 阶方阵，那么卷积层的 map 的大小就是 c 阶方阵，

$$c \times c = (n - k + 1) \times (n - k + 1) \quad (\text{式 3-3})$$

3.3.3 下采样层 (Subsampling layer, Pooling layer)

下采样层的具体任务就是对前层的图进行采样，也就是在相邻的小区域中采用聚合统计 (Aggregation statistics)，经过这样的处理之后，可以有效提取有用信息，降低计算量，详细过程如图 3-6，区域大小为 $\text{scale} \times \text{scale}$ ，这里示例中采用的小区域的均值，卷积核的大小是 2×2 ，卷积核内所有元素的值都是 $1/4$ ，而在有的实现当中，情况会有所改变，有时候采用某区域最大值。需要注意的是，卷积的计算窗口是有重叠的，就是如图 3.6 深颜色标注区域，但所使用的计算窗口却是没有重叠的，所以需要去掉重叠部分，从而得到采样层的输出：

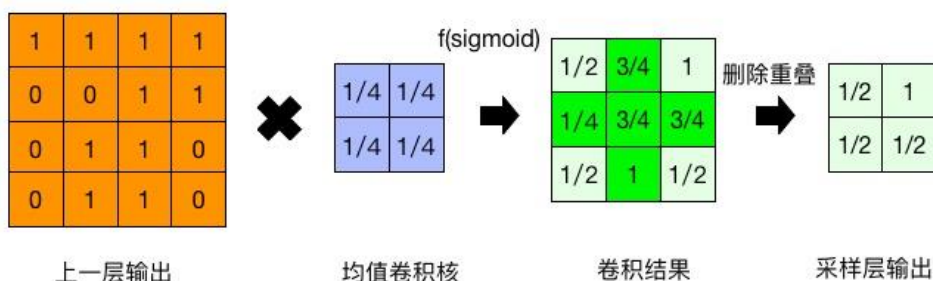


图 3.6 采样层图解

3.4 反向传输调整权重

类似于标准反向传播神经网络的反向传输过程，卷积神经网络也是通过最小化残差（Minimum residual）调整权重与偏置，这同样是卷积神经网络最繁杂的地方之一，复杂的结构决定了结构处理方式也不同，同时，由于权值共享的存在，计算残差的任务并不简单。

3.4.1 输出层残差

类似于标准反向传播神经网络，在卷积神经网络中，输出层的残差的计算方式与中间层不同，输出层的残差计算如式 3-4，误差值是由输出值与类的标准值得到的：

$$\delta_i^{(n_i)} = \frac{\partial}{\partial z_i^{(n_i)}} \frac{1}{2} \|y - h_{w,b}(x)\|^2 = -(y_i - a_i^{(n_i)}) \cdot f'(z_i^{n_i}) \quad (\text{式 3-4})$$

3.4.2 卷积层的残差（当下一层为采样层时）

首先假定已经通过计算，得出采样层残差，现在需要上一层卷积层残差，由图 3-4 可以看到，采样层 S2（S4）的 map 的大小是卷积层 C1（C3）的 1/4，计算过程如式 2-5，

$$\frac{1}{4} = \frac{1}{scale \times scale} \quad \text{when } scale = 2 \quad (\text{式 3-5})$$

同时，可以看到，这两层的 map 的个数是相同的，为了使采样层残差与上一层输出的 map 保持一致的维度，将采样层残差与大小为 $scale * scale$ 的单位矩阵进行 Kronecker product（克罗内克积）运算，以此操作来进行扩充。如图 3.7 展示拓展过程与计算过程。

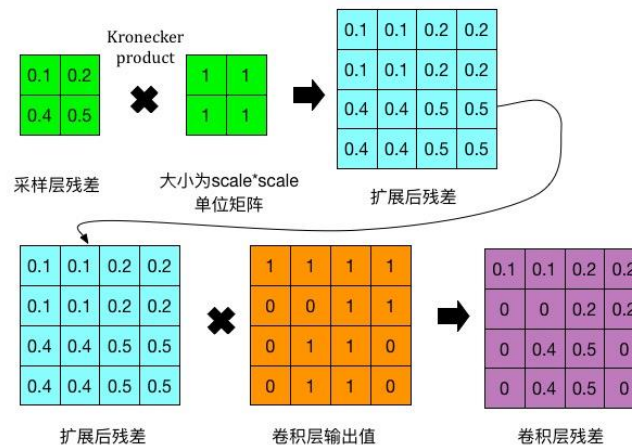


图 3.7 扩展及计算卷积层残差

3.5 卷积神经网络训练过程

在训练之前，首先明确两个概念，有指导学习与无指导学习，在模式识别（Pattern Recognition）领域，目前的主流方向是有指导学习，但是在聚类分析

（Cluster analysis）上，更常用的则是无指导的学习方法。

在有指导的学习方式当中，样本的类别是已知的，自然分布倾向已经不是划分空间的标准，而是根据同类样本的分部跟异类样本的分离度，来确定分类边界，将不同类型的样本划分出来。这个学习过程是漫长又复杂的，需要不断调整边界位置。

卷积神经网络能够做到，在没有任何输入输出对的数学表达式的条件下，学习海量映射关系，只需要用已有的模型训练卷积神经网络，映射能力是网络本身具有的。

所以卷积神经网络采用有指导学习方法，由类似于输入向量与理想输出向量的向量对构成其样本集，这些样本都可以从网上即时运行的系统中采集而来。训练之前，系统所有的权值都需要给定一个随机的数值，这个随机数值被称为“小随机数”，如果权值都全部相等，网络无法进行训练，不同的随机数可以保证卷积神经网络的正常的训练过程。同时，小随机数还可以避免网络因为权值太大，从而进入饱和状态，使训练失败。

卷积神经网络训练算法包括四步，这四步可以分为一下两个阶段：

3.5.1 向前传播阶段：

- （1） 随机选取单个样本 (X, Y_p) ，将其输入到网络当中；
- （2） 计算对应的实际输出，记为 O_p 。

在这个阶段，信息经过层层变动，从输入层到达输出层。网络在完成训练后，此进程是正常运行时执行的。在此过程中，由式（3-6）计算：

$$O_p = F_n (... (F_2 (F_1 (XpW^{(1)})W^{(2)}) ...)W^{(n)}) \quad (\text{式 3-6})$$

3.5.2 向后传播阶段

- （1） 计算实际输出 O_p 与理想输出 Y_p 的差值；
- （2） 采用上文提到的极小化误差法，调整权值。

向后传播阶段的工作需要严格控制精度，这里将网络中第 p 个数据的误差定义为 E_p ，那么，整个数据集的误差可见式 3-7，

$$E_{all} = \sum E_p \quad (\text{式 3-7})$$

总误差的具体算式 3-8，

$$E_p = \frac{1}{2} \sum_{j=1}^m (y_{pj} - o_{pj})^2 \quad (\text{式 3-8})$$

因为这个阶段传播方向与正常传播方向相反，所以叫做反向传播阶段。有时候这个阶段也被称为误差传播阶段，因为正向传播时，只能计算输出层的误差，而其他误差需要输出层误差反向计算得出。接下来将展开介绍卷积神经网络训练的过程。按表格 3-1 定义各变量：

表 3-1 卷积神经网络训练变量定义

字母	代表的含义
N	输入的单元数
L	中间层的单元数
M	输出层的单元数
X	网络的输入
H	中间层输出量
Y	网络的实际输出量
D	目标输出量
V_{ij}	输出单元到隐含层单元的权值
W_{jk}	隐含层到输出层的权值
θ_k	输出层单元的阈值
φ_j	隐含层单元的阈值

我们可以得到，中间层单元输出如下式 3-9，

$$h_j = f(\sum_{l=1}^{N-1} V_{lj}x_l + \varphi_j) \quad (\text{式 3-9})$$

输出层单元的输出如下式 3-10，

$$y_k = f(\sum_{j=1}^{L-1} W_{jk}h_j + \theta_k) \quad (\text{式 3-10})$$

激励函数为 S 型函数如下式 3-11，

$$f(x) = \frac{1}{1+e^{-kx}} \quad (\text{式 3-11})$$

网络的训练可按照如下过程：

- (1) 确定训练样本集，从中随机地选择 500 个数据作为训练集。将 V_{ij} 、

W_{jk} 、 θ_k 、 φ_j 初始化为小随机数（接近 0），同时给定精度控制参数 ε 与学习率 α 一个初始值；

- (2) 由训练集输入 X 到网络中，设定 D 值；
- (3) 利用式 3-9，得到 H，再利用式 3-10，得到 Y；
- (4) 比较 y_k 与 d_k ，由式 3-12 计算输出误差（M 个）；

$$\delta^k = (d_k - y_k) y_k(1 - y_k) \quad (\text{式 3-12})$$

由式 3-13 计算中间层误差（L 个）；

$$\delta_j = h_j(1 - h_j) \sum_{k=0}^{M-1} \delta_k W_{jk} \quad (\text{式 3-13})$$

- (5) 根据式 3-14 至 3-17 计算权值与阈值变化；

$$\Delta W_{jk}(n) = \frac{\alpha}{1+L} * (\Delta W_{jk}(n-1) + 1) * \delta_k * h_j \quad (\text{式 3-14})$$

$$\Delta V_{ij}(n) = \frac{\alpha}{1+N} * (\Delta V_{ij}(n-1) + 1) * \delta_k * h_j \quad (\text{式 3-15})$$

$$\Delta\theta_k(n) = \frac{\alpha}{1+L} * (\Delta\theta_k(n-1) + 1) * \delta_k \quad (\text{式 3-16})$$

$$\Delta\phi_j(n) = \frac{\alpha}{1+L} * (\Delta\phi_j(n-1) + 1) * \delta_j \quad (\text{式 3-17})$$

(6) 根据式 3-18 至 3-21 调整权值与阈值；

$$W_{jk}(n+1) = W_{jk}(n) + \Delta W_{jk}(n) \quad (\text{式 3-18})$$

$$V_{ij}(n+1) = V_{ij}(n) + \Delta V_{ij}(n) \quad (\text{式 3-19})$$

$$\theta_k(n+1) = \theta_k(n) + \Delta\theta_k(n) \quad (\text{式 3-20})$$

$$\phi_j(n+1) = \phi_j(n) + \Delta\phi_j(n) \quad (\text{式 3-21})$$

(7) 每当 k 值变为 M 之后，需要根据式 3-22 来判断总误差，若满足误差要求，继续进行下一步；若不满足，返回第三步；

$$E = \frac{1}{2} \sum_{k=0}^{M-1} (d_k - y_k)^2 \quad (\text{式 3-22})$$

(8) 结束训练，保存权值与阈值，以便后续训练随时调用。

前向传播与反向传播过程的网络流程图见图 3.7。

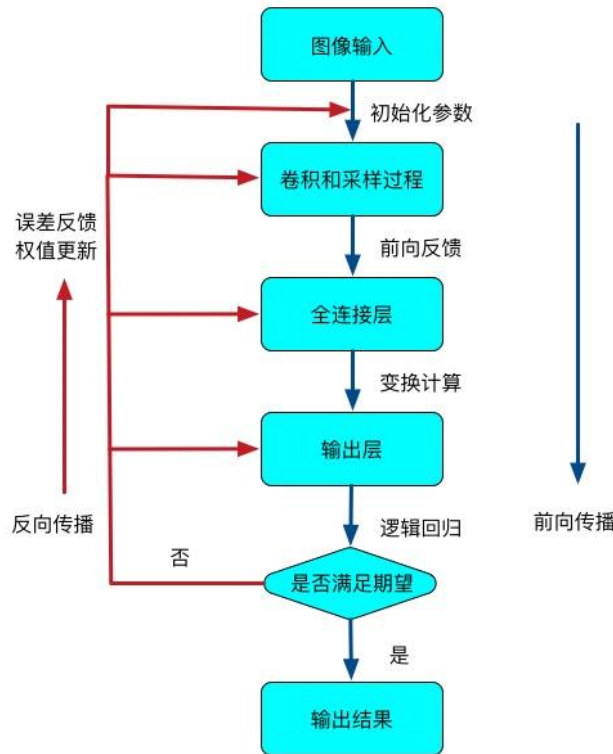


图 3.7 前向传播与反向传播过程的网络流程图

第四章 Mat Conv Net 模型

4.1 Mat Conv Net 介绍

Mat Conv Net 是 MATLAB 工具箱，用于计算机视觉应用，实现卷积神经网络的。上文提到，卷积神经网络在机器学习方面起到的作用越来越显著，特别是图像的理解，在慢慢取代传统方法。

现如今市面上有很多许多机器学习以及卷积神经网络的开源库。其中一些比较受欢迎的有：CudaConvNet, Torch, Theano, Caffe 等。这些库都普遍有上百位活跃的支持者和很多用户支持。所以，人们不禁要问，创建这个库的理由是什么？^[8]

开头提到，MATLAB 的便利性是关键因素，利用集成好的命令块，可以快速搭建卷积神经网络。Mat Conv Net 其实就是 MATLAB 的一个工具箱，任何用户都可以在 MATLAB 网站上得到。众所周知，MATLAB 是深受计算机视觉领域研究人员喜爱的开发环境之一，伴随着卷积神经网络的兴起，同时为计算机视觉开发者提供友好高效的开发环境，基于这些动机，MATLAB 平台推出了 Mat Conv Net，其中公开了很多简单且易于调用的卷积神经网络构建块，如卷积（convolution）、归一化（normalization）、池化（pooling）等指令。利用这些命令块进行组合与扩展，可以很容易构建卷积神经网络架构。不同于往常采用低级语言（如 C 语言）编写卷积神经网络，Mat Conv Net 大大简化了构建网络的过程，十分有利于刚接触卷积神经网络的新人的学习与研究。同时，MATLAB 本身可以支持图形处理器（GPU, Graphics Processing Unit）进行加速，Mat Conv Net 也不例外。这就意味着，无论系统多复杂，或者将来需要添加新的命令块，都可以通过图形处理器加速来保证运算效率。同时，Mat Conv Net 可以轻松应对如 AlexNet 这样大型卷积神经网络模型，用户在学习的过程中还可以 Mat Conv Net 主页上下载主流卷积神经网络的预培训版本。虽然有如此强大功能，Mat Conv Net 对 MATLAB 用户是完全免费的，而且使用和安装非常方便。MATLAB 和与之兼容的 C++ 编译器就可以实现，如果想使用 CUDA 加速的话，只需要配置相应的图像处理器。

4.2 Mat Conv Net 框架搭建

Mat Conv Net 易于安装和使用。图 4.1 演示了卷积神经网络搭建，以及对图像分类的简单尝试。

此例子首先下载（导入）Mat Conv Net，然后进行编译调试，接着下载（或导入）卷积神经网络模型，最后进行图像分类判断。

```

% install and compile MatConvNet (run once)
untar(['http://www.vlfeat.org/matconvnet/download/' ...
    'matconvnet-1.0-beta12.tar.gz']);
cd matconvnet-1.0-beta12
run matlab/vl_compilenn

% download a pre-trained CNN from the web (run once)
urlwrite(...
    'http://www.vlfeat.org/matconvnet/models/imagenet-vgg-f.mat', ...
    'imagenet-vgg-f.mat');

% setup MatConvNet
run matlab/vl_setupnn

% load the pre-trained CNN
net = load('imagenet-vgg-f.mat');

% load and preprocess an image
im = imread('peppers.png');
im_ = imresize(single(im), net.meta.normalization.imageSize(1:2));
im_ = im_ - net.meta.normalization.averageImage;

% run the CNN
res = vl_simplenn(net, im_);

% show the classification result
scores = squeeze(gather(res(end).x));
[bestScore, best] = max(scores);
figure(1); clf; imagesc(im);
title(sprintf('%s (%d), score %.3f', ...
    net.classes.description{best}, best, bestScore));

```

bell pepper (946), score 0.704



图 4.1 Mat Conv Net 框架示例

vl_simplenn: 网络的输入包括在网络上下下载的网络模型和预处理图像，输出结果 Res，进行分类。这也被成为卷积神经网络的“构建块”。可以在 Mat Conv Net 中修改其封装代码。

在运行程序之前，需要按照如上图所示搭建框架，以方便调用各种“构建块”，在本次毕业设计中，开发了相关程序，在搭建框架时，首先打开程序：

点击“初始化 Mat Conv Net”，待出现“set up successfully”进行下一步；



图 4.2 初始化程序

单击“xcode 调试”，待出现“compile successfully”关闭程序，表示 Mat Conv Net 已经搭建好，如图 3.3 所示，表示可以调用相关函数。



图 4.3 搭建 Mat Conv Net 成功

至此，Mat Conv Net 框架已搭建完成，需要注意的一点是，如果用户采用 windows 系统，需要安装与 MATLAB 版本兼容的 Visual Studio 版本进行联合调试；如果是非 windows 系统，如 OSX 系统，需要安装与 MATLAB 版本兼容的 Xcode 版本进行联合调试，其他系统同理。具体兼容与匹配请参考 Mat Conv Net 官方文件。

关于 CUDA 调用图形处理器加速，需要注意的是，用户所用计算机的图形处理器需要满足 CUDA 的官方要求，否则无法进行加速。

第五章 程序设计与实现

5.1 程序流程设计

因为神经网络的两个大的步骤为训练网络和测试网络，所以相应的，系统有两大主要流程。

图 5.1 为卷积神经网络的训练流程图示。具体步骤如下：在训练开始之前，当我们完成程序的编写，首先需要把程序调试好，程序准备就绪好，需要准备训练所用的训练集以及相应的标签列表，如果采用国际通用的数据库，如 MINST 手写体训练库，就不需要自己准备标签；一般情况下，都用自己准备的数据库，那么就需要为所有图片打上标签，这个工作虽然繁琐但是不可省略，可以参照下文 5.2 的提示打标签，然后，设置网络的参数，例如损失函数值、学习率、最大迭代次数等等，在本次设计中，有对应的程序可以进行操作，即按照下图 5.3 进行修改。并且进行相应的预处理，如标准化，二值化的处理。一切准备就绪之后，就可以进行网络的训练了。

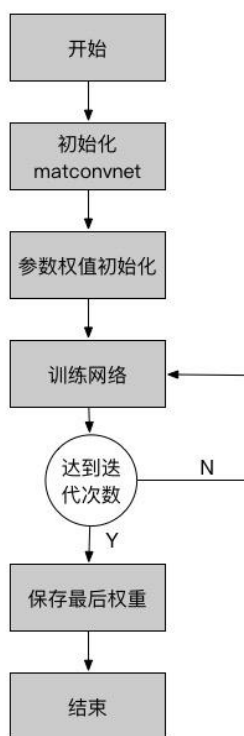


图 5.1 训练流程图

随着训练的进行，会不断产生权重文件，而在训练的过程中，网络会不断检查当前的迭代次数，如果还没有达到预设的最高迭代次数，就会继续当前的训练；当迭代次数达到预期最大时，训练就会终止，接着生成最终的权重文件，训练结束。这个权重文件将会用来作为网络的测试。

5.2 程序编写

此次程序采用的是经典的 LeNet-5 网络，上文提到，这时一种很典型的卷积神经网络，在当年美国的各大银行中，就是用这种方法来识别银行支票上的数字，在要求精度极高的商用领域都能得到广泛应用，可见 LeNet-5 网络的可靠性是非常高的。

主要包括输入层、输出层、两个卷积层、两个下采样层（池化层）和一个全连接层，共七层网络结构，在每一层上（除去输入层），都含有可以训练的连接权重（即参数）。具体的输入输出关系见表 5-1。

表 5-1 LeNet-5 网络输入输出关系

各层级	输入输出关系
输入层（input）	直接输入图片，像素为 $32*32$
卷积层（C1）	采用 $5*5$ 的卷积核，生成 6 个特征 maps，参数为 $(6*5*5+6)=156$ 个
下采样层（S2）	$2*2$ 个像素相加，之后乘参数，加偏执，参数为 $2*6=12$ 个
卷积层（C3）	采用 $5*5$ 的卷积核，生成 16 个特征 maps，每一个特征 maps 都与 S2 特征有关，详细见表 5-2
下采样层（S4）	操作与 S2 类似，参数个数为 $16*2=32$ 个
卷积层（C5）	与上一层（S4）全连接，参数为 $(5*5*16*120)=48120$ 个
全连接层（F6）	与上一层（C5）全连接，参数为 $(120*84+84)=10164$ 个
输出层（output）	与上一层（F6）输出全连接

表 5-2 LeNet-5 网络 C3 层特征 maps 与 S2 层特征对应关系

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

采用的样本为网络上随机挑选的商品图片，因为在网络商城中的上架物体背景变化较多，而且为主流物体模型，训练集为 1000 张图片，一共分为 10 类物体，每一类物体 200 张图片作为测试集，在迭代 10 次的情况下，正确率能够

达到 90%。程序的编写参考 Mat Conv Net 提供的第三方库。关于程序的参数解释如下：

`opts.train.learningRate`: 学习速率；

`opts.train.batchsize`: 一次输入卷积神经网络的样本数；

`opts.train.numSubBatches` : 每一个 batch 被分成的份数；

`opts.train.gpus`: 是否调用图形处理器；

`vl_argparse(opts, varargin)`: 把 `varargin` 中所有的参数属性和值放入 `opts` 中；

`prepareDINet(net,opts)`: 对导入的预先训练的样本进行处理；

`cnn_image_setup_data(varargin)`: 对数据格式进行准备，Mat Conv Net 对训练的数据有格式要求，通常情况下是生成 `imdb` 结构体保存起来，目的是为了将训练集与测试集的图片名称与标签提取出来，并做映射关系，统一的数据格式方便调用 Mat Conv Net 的相关函数对数据进行处理；

`cnn_train_dag(net, imdb, getBatchFn(opts, net.meta))`: 直接调用 `cnn_train_dag` 的应用程序编程接口（Application Programming Interface, API）对整个集合进行训练；

`cnn_imagenet_deploy(net)`: 将训练好的网络保存。

结合卷积神经网络算法、LeNet-5 网络结构与 Mat Conv Net 架构，在 MATLAB 环境下，编写一般物体识别程序，程序的列表如下图 5.2 所示：

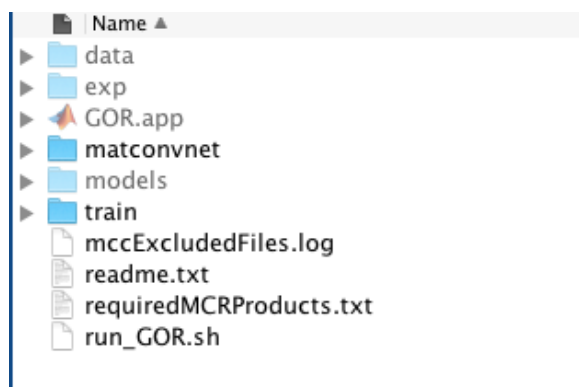


图 5.2 程序列表

文件夹内容及功能见图 5-3:

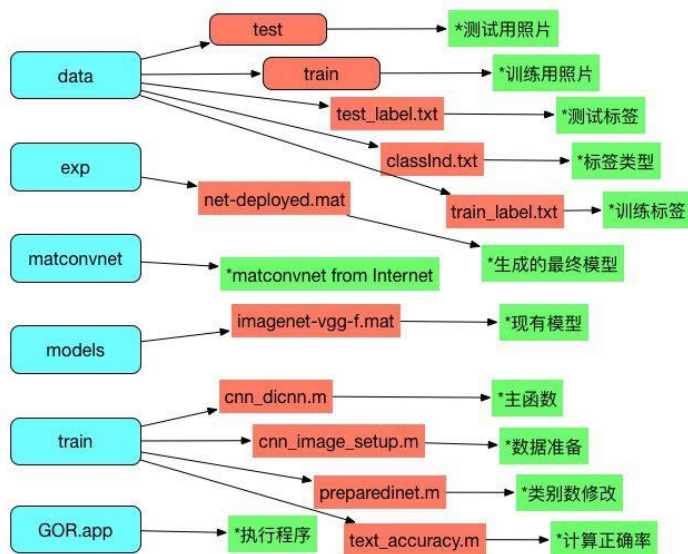


图 5-3 程序各文件功能

运行程序 GOR.app:

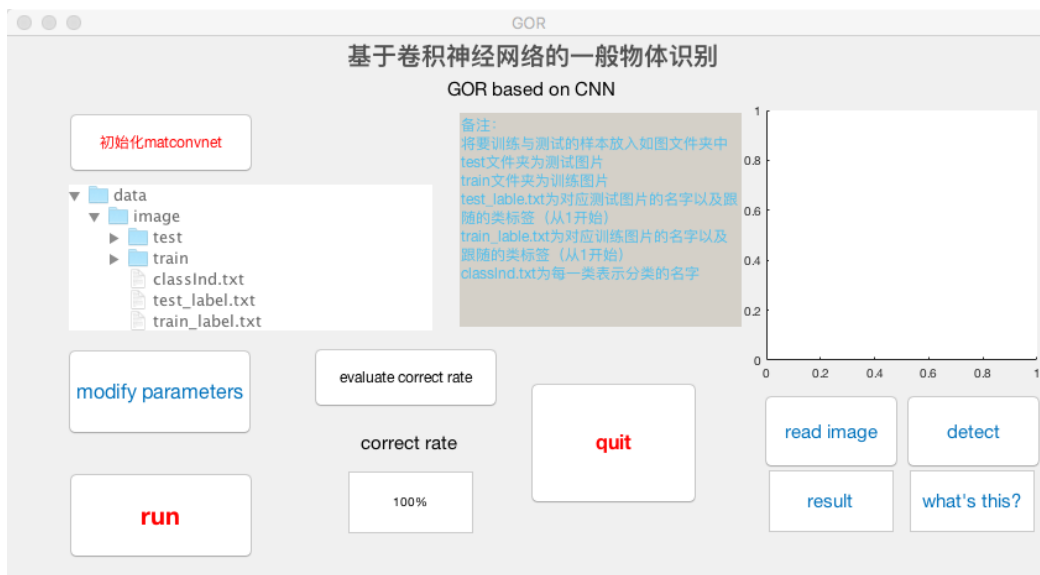


图 5.3 主程序界面

首先将要训练与测试的样本放入如图 5.2 文件目录下中:

test 文件夹为测试图片;

train 文件夹为训练图片, 共有 10 类物体, 部分样本图片如下图;



怀表 类别:watch



运动裤 类别:trousers



发卡 类别:hairpin



拉杆箱 类别:drawbar_box



椅子 类别:chair



电视墙 类别:tea_table



酸奶 类别:yoghourt



AVON



香水 类别:perfume

白桦树



D'ARPEGE



暗语



化妆品 类别:makeup_bottle



鞋子 类别:shoes

图 5.4 10 种类别的部分样本图片

test_label.txt 为对应测试图片的名字以及跟随的类标签（从 1 开始）；

train_label.txt 为对应训练图片的名字以及跟随的类标签（从 1 开始）；

classInd.txt 为每一类表示分类的名字。

```
1 watch
2 hairpin
3 trousers
4 drawbar_box
5 chair
6 tea_table
7 yoghourt
8 perfume
9 makeup_bottle
10 shoes
```

图 5.5 分类名示例（10 类）

由于我是在训练好的模型上继续训练，所以需要有一个预训练模型，在图 5.2 文件地址下建立一个 models 文件夹，然后把 imagenet-vgg-f.mat 放入到 models 里面。这里我使用的是 vgg-f 的 model，可以通过官方网站直接下载到。

接着就是网络训练。相关源程序在 train 文件夹中。

在网络训练之前，可以调整一些基础参数，如果要修改其他参数，在可视化界面中无法修改，需要修改源程序，可视化界面提供学习率的修改（默认为 0.001）、迭代次数的修改（默认 20 次）以及是否开启图形计算器加速（默认不开启），然后点击“modify parameters”按钮，如图 5.6 所示：

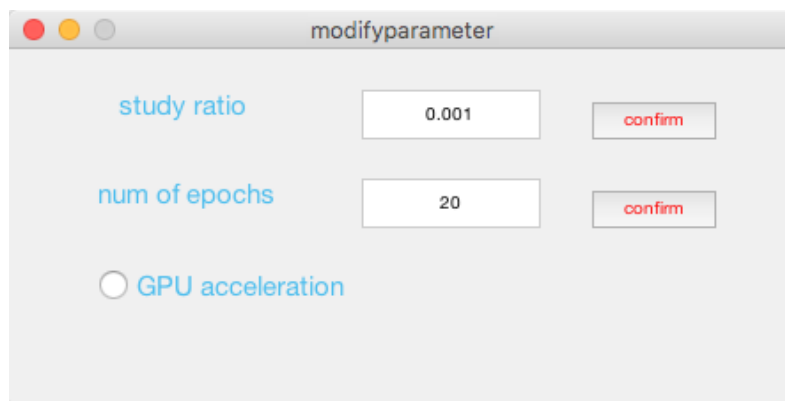


图 5.6 参数调整界面

然后就可以运行程序，程序完成后，根据可视化界面引导可以即时计算匹配率。

当迭代次数太多（50 次以上），如果不采取图形处理器加速的话，耗费的时间比较长，所以默认迭代次数设定为 20 次，而且，程序会保存每一次迭代之后训练的结果，如果中途中断程序，当下次开启程序时，程序会寻找上一次最后完成迭代次数的下一次继续进行，这样有很大的好处，如果发现设置迭代次数太多，想中途停止，那么可以停止程序，将程序的迭代次数改为当前次数，再运行程序，程序会在训练完当前次数后停止，这样可能提高效率；或者由于客观因素，比如停电或者电脑崩溃等，不能继续训练，当再次打开程序时，程序会检索上次训练结果，而不至于浪费时间重新训练，这一点是基于人性化考虑而设计的。下图就是程序正常运行时的过程。

```
set up successfully
train: epoch 01: 1/125: 1.4 (1.4) Hz objective: 2.303 toplerr: 0.750 topSerr: 0.500
train: epoch 01: 2/125: 1.8 (2.7) Hz objective: 2.301 toplerr: 0.750 topSerr: 0.375
train: epoch 01: 3/125: 2.1 (2.7) Hz objective: 2.302 toplerr: 0.792 topSerr: 0.417
train: epoch 01: 4/125: 2.2 (2.8) Hz objective: 2.290 toplerr: 0.750 topSerr: 0.406
train: epoch 01: 5/125: 2.8 (2.8) Hz objective: 2.282 toplerr: 0.675 topSerr: 0.400
train: epoch 01: 6/125: 2.8 (2.6) Hz objective: 2.275 toplerr: 0.646 topSerr: 0.396
train: epoch 01: 7/125: 2.7 (2.4) Hz objective: 2.266 toplerr: 0.643 topSerr: 0.357
train: epoch 01: 8/125: 2.7 (2.5) Hz objective: 2.264 toplerr: 0.625 topSerr: 0.359
train: epoch 01: 9/125: 2.7 (2.6) Hz objective: 2.251 toplerr: 0.611 topSerr: 0.347
train: epoch 01: 10/125: 2.6 (2.5) Hz objective: 2.233 toplerr: 0.587 topSerr: 0.325
train: epoch 01: 11/125: 2.6 (2.5) Hz objective: 2.225 toplerr: 0.602 topSerr: 0.318
train: epoch 01: 12/125: 2.6 (2.7) Hz objective: 2.191 toplerr: 0.583 topSerr: 0.292
train: epoch 01: 13/125: 2.6 (2.5) Hz objective: 2.183 toplerr: 0.596 topSerr: 0.279
```

5.7 正常训练过程

图 5.3 的右边部分为物体识别部分，可以将任意图片上传，进行识别，此次程序只对 10 类物体进行训练，比如将测试集中的鞋子照片上传，进行识别，操作如图 5.8。

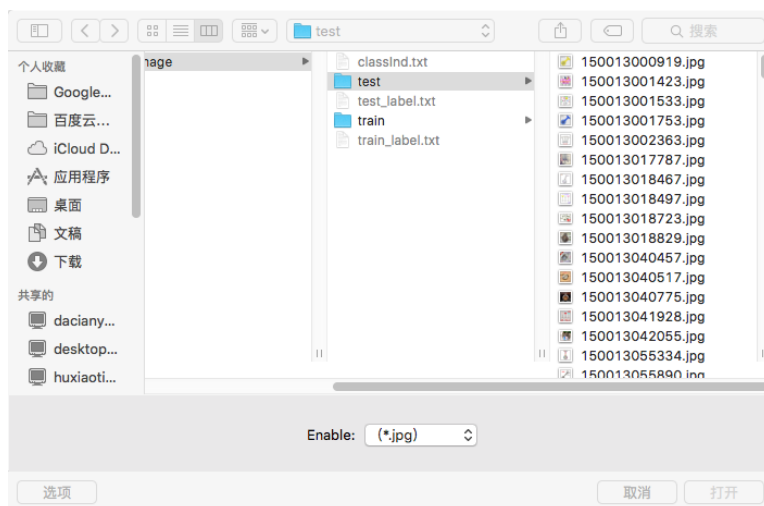


图 5.8 选择照片

选择照片之后单击“detect”按钮，等待结果。如图 5.9。

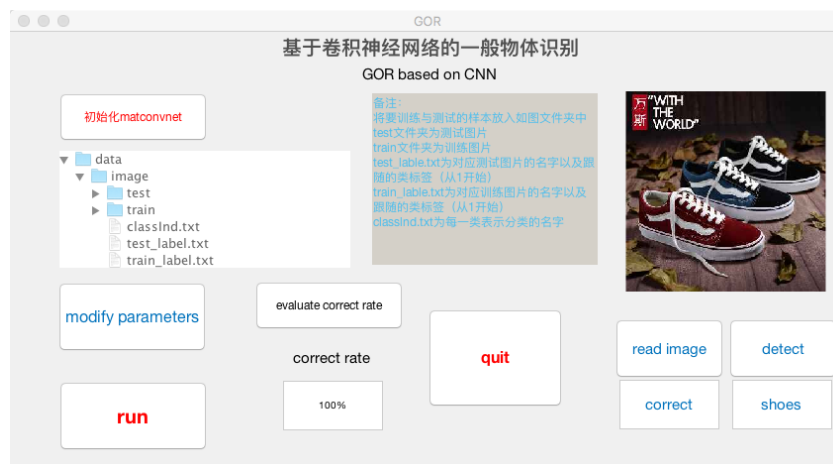


图 5.9 鞋子识别结果

可以看到，系统显示识别正确（correct），图中所示物体为鞋子（shoes）。

5.3 程序结果

5.3.1 不同训练结果对比

设定学习率相同，均为 0.001，分别迭代 1 次与迭代 10 次，比较正确率。

Train 为训练集，val 为验证集，objective 为损失函数。

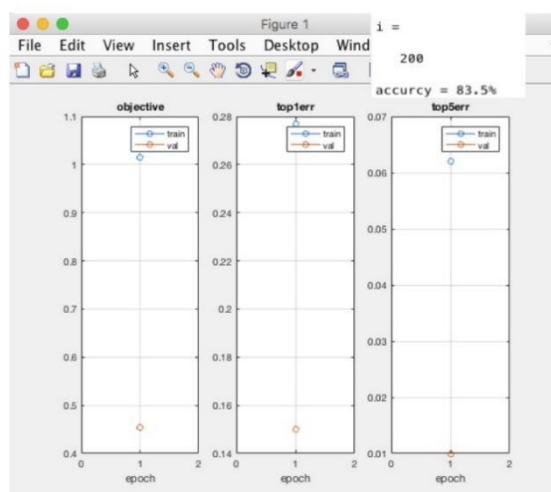


图 5.8 迭代 1 次与相应正确率（0.835）

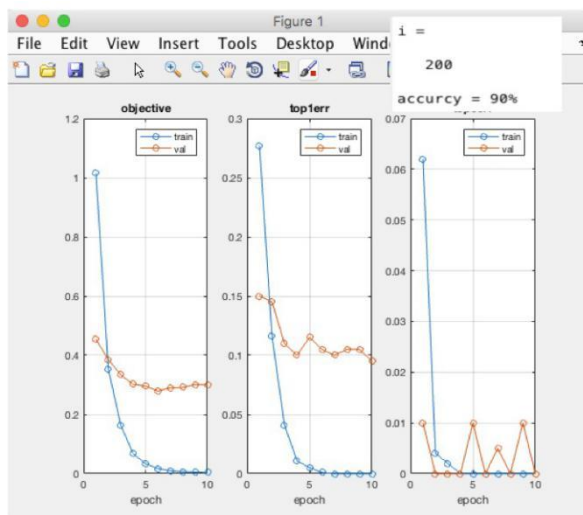


图 5.9 训练 10 次与相应正确率（0.900）

可以看到，迭代 10 次之后，top-5 错误率已经基本收敛。

5.3.2 每个类型物体识别率对比

识别库中含有每一类物体 20 张，在迭代 1 次与 10 次后，识别率如图 5.10

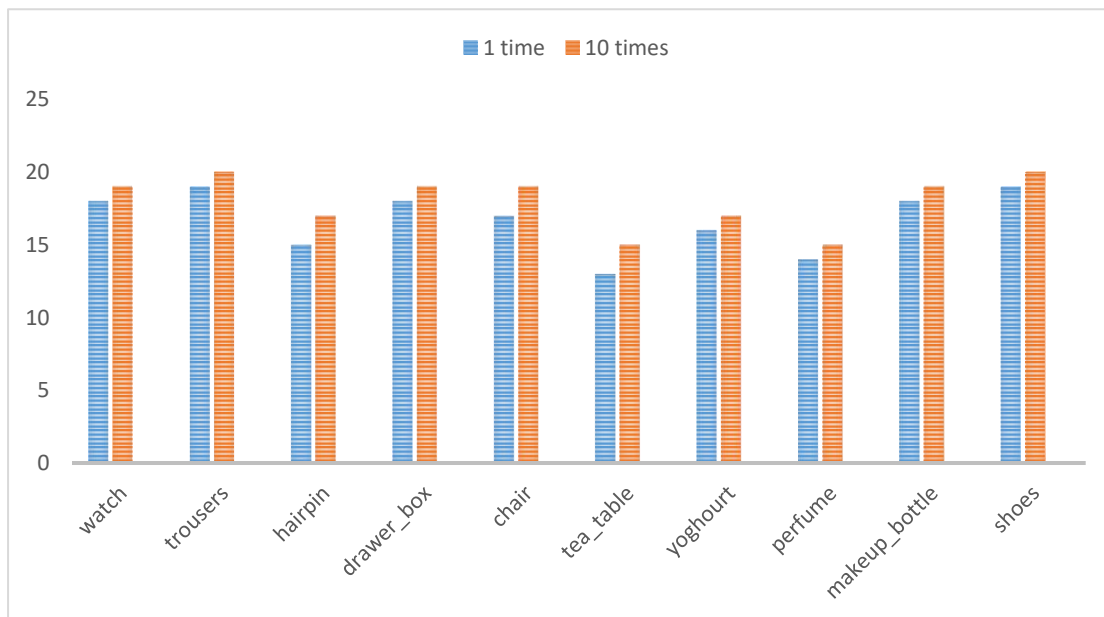


图 5.10 10 类物体的正确识别个数

由上图可以看到，当迭代 10 次时，所有物体的识别率都有相应的提升，平均识别率从 87.5% 上升到 90%。在这 10 类物体当中，茶几的识别率最低，原因在于背景较为复杂，造成较大干扰；而且茶几摆放组合方式较多，训练样本比较少，导致识别率较低。香水与头绳的识别率较低因为种类繁多并且类内区别度比较大，像鞋子、手表与裤子由于样式较为单一，识别率接近 100%。

5.3.3 是否采用图形处理器加速对比

未采用图形处理器加速之后，处理速度如下图，

```

set up successfully
train: epoch 01: 1/125: 1.4 (1.4) Hz objective: 2.303 toplerr: 0.750 top5err: 0.500
train: epoch 01: 2/125: 1.8 (2.7) Hz objective: 2.301 toplerr: 0.750 top5err: 0.375
train: epoch 01: 3/125: 2.1 (2.7) Hz objective: 2.302 toplerr: 0.792 top5err: 0.417
train: epoch 01: 4/125: 2.2 (2.8) Hz objective: 2.290 toplerr: 0.750 top5err: 0.406
train: epoch 01: 5/125: 2.8 (2.8) Hz objective: 2.282 toplerr: 0.675 top5err: 0.400
train: epoch 01: 6/125: 2.8 (2.6) Hz objective: 2.275 toplerr: 0.646 top5err: 0.396
train: epoch 01: 7/125: 2.7 (2.4) Hz objective: 2.266 toplerr: 0.643 top5err: 0.357
train: epoch 01: 8/125: 2.7 (2.5) Hz objective: 2.264 toplerr: 0.625 top5err: 0.359
train: epoch 01: 9/125: 2.7 (2.6) Hz objective: 2.251 toplerr: 0.611 top5err: 0.347
train: epoch 01: 10/125: 2.6 (2.5) Hz objective: 2.233 toplerr: 0.587 top5err: 0.325
train: epoch 01: 11/125: 2.6 (2.5) Hz objective: 2.225 toplerr: 0.602 top5err: 0.318
train: epoch 01: 12/125: 2.6 (2.7) Hz objective: 2.191 toplerr: 0.583 top5err: 0.292
train: epoch 01: 13/125: 2.6 (2.5) Hz objective: 2.183 toplerr: 0.596 top5err: 0.279

```

图 5.11 无 GPU 加速训练过程

采用图形处理器加速之后，处理速度如下图，

```

train: epoch 01: 2/125: 2.4 (2.7) Hz objective: 2.302 toplerr: 0.875 top5err: 0.417
train: epoch 01: 3/125: 2.8 (2.9) Hz objective: 2.303 toplerr: 0.875 top5err: 0.417
train: epoch 01: 4/125: 2.8 (3.1) Hz objective: 2.292 toplerr: 0.813 top5err: 0.417
train: epoch 01: 5/125: 2.9 (3.1) Hz objective: 2.285 toplerr: 0.750 top5err: 0.417
train: epoch 01: 6/125: 2.9 (3.1) Hz objective: 2.278 toplerr: 0.708 top5err: 0.417
train: epoch 01: 7/125: 2.9 (3.0) Hz objective: 2.268 toplerr: 0.696 top5err: 0.417

```

图 5.12 有 GPU 加速训练过程

通过对比，可以发现，采用图形处理器加速之后，处理速度显著提升（由 1.8hz 上升至 2.8hz），迭代一次的平均时间由 200s 以上降低至 40s 以内。

第六章 全文总结与结论

本文是作者作为本科生对卷积神经网络的第一次深入尝试,借助于 Mat Conv Net 强大的功能,独立完成对卷积神经网络 LeNet-5 结构的搭建,同时手动采集 1200 张、共 10 类物体的图片数据,完成对训练集与测试集的搭建,笔者对卷积神经网络有了全面深刻的理解,没有在创新性上进行突破,但是在传统的卷积神经网络的学习中,我认为注意理解以下几点可以更好地理解卷积神经网络。

在数据输入层,需要理解归一化,去均值,PCA 降维。在卷积层中,可以将每个神经元看成一个滤波器,随着窗口滑动,滤波器对局部数据进行计算,还有参数共享机制。了解各个激励函数的特性,如 sigmoid, ReLU(The Rectified Linear Unit/修正线性单元)等。在池化层中可以将其理解为对图像进行压缩。

物体识别在近几年逐渐成为模式识别科研领域的热门,不光在实验室,在日常生活中,对于物体识别的需求与应用更是与日俱增,随着电脑硬件的飞速发展与科研人员的刻苦钻研,促使着物体识别也在不断进步。随着社会的发展,经济的进步,对生活的需求也越来越高,这也使着这项技术向着更复杂,更精密的方向发展,更需要我们广大科研人员开阔创新,走在科技的前沿,以满足当代社会对高新科技的不断需求。

本文的最终的实现比较简单基础,并不能满足更高等级问题的需求。如何针对实际特定问题、特定图像训练库设计更有效的网络结构,目前仍是一个难题,在改善网络方面,无论从应用上还是从理论上,我们还有很多工作要完成,我觉得可能的研究方向有以下两点:

- (1) 在理论上,如何在算法中利用深度模型的选择性、稀疏性,如何设计算法保证收敛与稳定性;
- (2) 在应用上,针对海量数据、更深层次与结构的网络,如何保持高效。如何融合数值优化与并行计算等方法。

随着在以上两方面的深入发展,我相信卷积神经网络在图像理解中将会得到更好的进步。在未来,卷积神经网络会在很多方面有上升空间。

(1) 多输入方向的卷积神经网络

传统的卷积神经网络在目前的模式识别领域取得的巨大进步是有目共睹的,但是仍然是传统的方法,采集输入集,通过特征提取,最后通过分类器分类,但有的时候,往往需要图像清晰度,对视频进行即时处理,就需要连续多张图像综合全面分析才能够得到可靠的结果,当前,采用具备互补特性的高级图像传感器是很多进行视觉分析的系统的方法,可以显著提高系统的鲁棒性。因此,多输入的卷积神经网络具有很深的研究价值与应用价值。由于不同卷积层代表不同层的数据特征,所以建立不同层级的信息融合层来实现不同图像系统信息融

合是可行的。

(2) 改善泛化能力方向

在日常实验的过程中,当我们要对系统进行训练时,往往需要对我们人工采集的样本集进行标记,众所周知,如果想提高卷积神经网络的识别准确度,必然需要大量的标记过的样本进行训练,都在实践中,海量样本的手工标记显然是不现实的,只也是制约卷积神经网络精度的一个重要原因。如果有半监督形式的卷积神经网络可以在一定程度上解决这个问题,也就是通常所说的网络的泛化能力。与此同时。也可以利用最优化核函数等方法,来提高网络的泛化能力。

(3) 仿生视觉模型

卷积神经网络就是启发于生物视觉模型,但是其实卷积神经网络的结构与真实的视觉系统还是有很大的差距,我么你可以把卷积核看成视觉细胞,在卷积神经网络中,卷积核在训练过程中,内部的各个元素是互相独立的,真正的视觉模型是将卷积核的相互联系起来,发挥卷积核的又是,这样,在隐含层就可以采集到更多的特征,对图像的理解与表示就更加真实,利用卷积神经网络的卷积层可以提取图像的局部信息特征,在解决图像融合问题上,这是非常关键的。

(4) 高层输出与底层输出的交互

卷积神经网络中的各个节点层都是有不一样的作用,而每层的输出也同时表示各层的图形特征,如今的卷积神经网络往往更在乎识别结果,却忽视了不同层级的图像特征,正确利用这些特征可以有效提高系统的功能。高层信息与底层信息的交互可以充分利用卷积神经网络的纵向多层结构的特点,所以,研究卷积神经网络的各个层级之间的关系,并且制定高层输出与底层输出的交互计算,能够显著提高图像处理的品质。

(5) 向其他领域拓展

近几年,卷积神经网络在图像处理、物体识别与目标检测等方向已经获得了很大的成就,同时,在其他领域,卷积神经网络还没有被很好地利用,在 2015 年国际机器学习大会上,专家们就指出,未来,卷积神经网络一定会在广泛的行业中得到应用。

深度学习给机器学习带来新的潮流,各行各业,尤其在各大公司、各大高校,,都收到了密切地关注,而卷积神经网络绝对是当今深度学习的宠儿,在一般物体识别领域,如果能在理论算法、建模与工程实现方面得到突破,一定能够推动深度学习的长远发展。[4][10]

致谢

在此论文搁笔之时、行将完成之际，首先感谢东南大学，“止于至善”这四个字教会了我太多，严谨务实，是我作为一个东大人对自己的严格要求；勇于探索，是我作为一名科研人员对自己的激励鞭策。感谢东南大学自动化学院四年来对我的栽培，回首这四年，时光荏苒，愉快充实这四个字可以说概括了我的本科生生涯，感谢我的每一位老师，感谢你们传道授业，你们的忠告我会牢记在心里，没有你们的悉心教导就没有我的今天，感谢在这四年里朝夕相处的同学，社团的同事，是你们丰富了我的大学生活，你们将是我一生的财富，感谢一直默默支持我的几位舍友，谢谢你们。

我最要衷心感谢的是我的导师，李新德教授。感激您在本科这四年对我的谆谆教导。在学习生活中，您的国际化视野总能为了指明前进的方向，您渊博的知识影响着我、教育着我，让我不断进取，让我明白学无止境；在论文的写作中，选题时，您能充分尊重我的意见，开题时，您能不厌其烦地解答我的问题，一次次批改开题报告，写作过程中，您能精确地指出我的诸多问题，您认真负责的精神令人敬佩；在日常生活中，您和蔼可亲，让我在这个城市真正感受到家的温暖，您在传授知识的同事，能教给我许许多多人格哲理与在社会中生存的道理，我将终身受益，千言万语汇成一句话：谢谢您，李教授！

再次感谢自动化学院所有的老师与同学，是你们的帮助让我能拥有一个美好并值得怀念的大学本科生活！

最后感谢一直来全力支持我的父母，没有你们的帮助就没有我的今天，感谢你们的爱与支持！

人生路漫漫，其修远兮！我会一直铭记给过我帮助的你们，我的本科生的学习即将结束，不说再见，因为我的研究生生活才刚刚开始，我的科研道路也刚刚迈出了微小的一步，新的征程即将开启，我也准备好了！

再次感谢大家！

参考文献

- [1] 杨眷玉. 基于卷积神经网络的物体识别研究与实现[D]: [博士学位论文] 电子科技大学, 2016.
- [2] 田渊栋. 基于特征组合的一般物体识别相关算法研究[D]: [博士学位论文] 上海交通大学, 2007.
- [3] 李新德, 刘苗苗, 徐叶帆, 等. 一种基于 2D 和 3D SIFT 特征级融合的一般物体识别算法[J]. 电子学报, 2015, 43(11):2277 - 2283.
- [4] 王茜, 张海仙. 深度学习框架 Caffe 在图像分类中的应用[J]. 现代计算机, 2016(5):72 - 75.
- [5] 刘建伟, 刘媛, 罗雄麟. 深度学习研究进展[J]. 计算机应用研究, 2014, 31(7):1921 - 1930.
- [6] 艾泳宏 .MAT CONV NET 介绍 .[EB/OL]. http://blog.sina.com.cn/s/blog_92cd3a1c0102x1ch.html, 2016-10-27
- [7] 九尾 . Theano 学习笔记 .[EB/OL]. http://blog.sina.com.cn/s/blog_6553da670102winw.html, 2016-09-13
- [8] liccv. 几种深度学习框架的分析与比较. [EB/OL]. <http://blog.csdn.net/liccv/article/details/52526506>, 2016-10-27
- [9] 常亮, 邓小明, 周明全, 等. 图像理解中的卷积神经网络[J]. Acta Automatica Sinica, 2016, 42(9):1300 - 1312.
- [10] 余凯, 贾磊, 陈雨强, 等. 深度学习的昨天、今天和明天[J]. 计算机研究与发展, 2013, 50(9):1799 - 1804.
- [11] 吴正文. 卷积神经网络在图像分类中的应用研究[D]: [博士学位论文] 电子科技大学, 2015.
- [12] 冈萨雷斯. 数字图像处理的 MATLAB 实现[M] 清华大学出版社, 2013.
- [13] 李新德, 张晓, 朱博. 基于立体视觉的一般物体识别方法[J]. 东南大学学报(自然科学版), 2013, 43(4):711 - 716.
- [14] 王添翼. 基于卷积网络的三维模型特征提取[D]: [博士学位论文] 吉林大学, 2006.
- [15] 李葆青. 基于卷积神经网络的模式分类器[J]. 大连大学学报, 2003(2):19 - 23.
- [16] SimonHaykin, Haykin, 叶世伟, 等. 神经网络原理[M] 机械工业出版社, 2004.
- [17] 肖柏旭. 基于卷积网络的人脸检测的研究与实现[D]: [博士学位论文] 华北电力大学(河北), 2007.

- [18] 陆璐. 卷积神经网络的研究及其在车牌识别系统中的应用[D]: [博士学位论文] 合肥工业大学, 2006.
- [19] 顾佳玲, 彭宏京. 增长式卷积神经网络及其在人脸检测中的应用[J]. 系统仿真学报, 2009, 21(8):2441 - 2445.
- [20] 赵志宏, 杨绍普, 马增强. 基于卷积神经网络 LeNet-5 的车牌字符识别研究[J]. 系统仿真学报, 2010,22(3):638 - 641.
- [21] Hildebrandt T H. Optimal training of thresholded linear correlation classifiers[J]. IEEE Transactions on Neural Networks, 1991, 2(6):577 - 588.
- [22] Neubauer C. Shape, position and size invariant visual pattern recognition based on principles of neocognitron and perceptron[J]. 1992.
- [23] Lovell D, Tsoi A C, Downs T, et al. Comments on 'Optimal training of thresholded linear correlation classifiers' [with reply]. [J]. IEEE Transactions on Neural Networks, 1993, 4(2):367 - 369.
- [24] Fukushima K. Analysis of the process of visual pattern recognition by the neocognitron[J]. Neural Networks,1989, 2(6):413 - 420.
- [25] Van O A, Nienhuis B. Pattern recognition in the neocognitron is improved by neuronal adaptation.[J]. Biological Cybernetics, 1993, 70(1):47 - 53.
- [26] Nebauer C. Evaluation of convolutional neural networks for visual recognition[J]. IEEE Transactions on Neural Networks, 1998, 9(4):685 - 96.
- {27} Cottrell, Garrison W, Metcalfe, et al. EMPATH: face, emotion, and gender recognition using holons[C]. In:Advances in Neural Information Processing Systems. 1990. 564 - 571.
- [28] Fukushima K. Neocognitron: A hierarchical neural network capable of visual pattern recognition[J]. Neural Networks, 1988, 1(2):119 - 130.