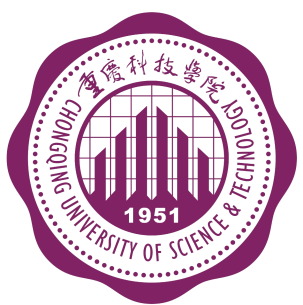
重庆科技学院



**毕业设计（论文）**

题 目  基于无监督学习神经网络数据

降维自编码器设计与实现

院（系） 电气与信息工程学院

专业班级 物联网2014-02

学生姓名 张亚兵 学号 2014443802

指导教师 向毅 职称 教授

评阅教师 职称

2018 年 6 月 1 日

**学生毕业设计（论文）原创性声明**

本人以信誉声明：所呈交的毕业设计（论文）是在导师的指导下进行的设计（研究）工作及取得的成果，设计（论文）中引用他（她）人的文献、数据、图件、资料均已明确标注出，论文中的结论和结果为本人独立完成，不包含他人成果及为获得重庆科技学院或其它教育机构的学位或证书而使用其材料。与我一同工作的同志对本设计（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。

毕业设计（论文）作者（签字）：

年 月 日

# 摘 要

随着信息技术的进步，当今各个领域的数据规模和复杂度都在飞速增加，在大数据时代人们希望能够获得更多的数据，来挖掘出数据后面隐藏的客观规律。数据量越来越大，便造成了数据维度越来越来越高，在高维数据中数据冗余现象严重，加大了数据分析的难度，同时极大增加了数据计算量。近年来随着机器学习的发展为数据降维算法提供了许多新的解决方案。自编码器是属于无监督的机器学习方法，当自编码器引入稀疏的特性时，自编码器可以通过无监督学习得到数据特征，将高维度的数据映射到低维空间，降维后的数据具有良好的可分析性，同时自编码器在神经网络的预训练也有非常好的效果。

本课题基于tensorflow机器学习平台对mnist数据进行降维，主要工作如下：

1.设计实现了基于神经网络的数据降维自编码器，该自编码器在mnist数据的学习和重建取得了很好的效果，在数据降维到10%左右的时候，可以保持99%以上的重建准确率。

2.自编码器可以用于神经网络的预训练，本课题使用自编码器预训练之后在mnist识别方面原神经网络取得了更好的效果。

3.移植tensorflow和opencv到android手机端，完成了基于手机的手写数字识别。

**关键词：自编码器 数据降维 机器学习 神经网络 android**

# ABSTRACT

With the development of information technology, the data scale and complexity are in all areas of rapid increase, in the era of big data people want to be able to get more data, to dig out the data hiding behind the objective laws. Amount of data is more and more big, then caused the data become more and more to the higher dimensions, data redundancy in the high latitudes data is serious, caused difficulties to data analysis, greatly increases the amount of calculation data at the same time. In recent years, with the development of machine learning, many new solutions have been provided for the data reduction algorithm. Since the encoder belongs to unsupervised machine learning method, when the encoder is introduced into sparse feature, the encoder can get data through unsupervised learning feature, map the high dimensional data to low dimensional space, dimension reduction after the data has good analytical, at the same time the encoder in advance of the neural network training also has a very good effect.

This topic is based on the tensorflow machine learning platform to reduce the data of mnist. The main tasks are as follows:

1. Design and implementation based on neural network data dimension reduction since the encoder, the mnist data of study and reconstruction since the encoder has achieved good results, at the time of data dimension reduction to about 10%, can keep more than 99% of the reconstruction accuracy.

2. Since the encoder can be used for pre-training of neural network, this topic has achieved better results in mnist recognition after using the self-encoder pre-training.

3. Transplanting tensorflow and opencv to the android phone end, and completing the shouxie1 digital identification based on the phone.

**Key words: self-encoder; Data dimension reduction;Machine learning; The neural network;Androi**

# 目 录

[摘 要 I](#_Toc515702983)

[ABSTRACT II](#_Toc515702984)

[目 录 III](#_Toc515702985)

[1 绪 论 - 1 -](#_Toc515702986)

[1.1 课题背景及研究意义 - 1 -](#_Toc515702987)

[1.2 国内外研究现状 - 1 -](#_Toc515702988)

[1.3 课题研究内容 - 3 -](#_Toc515702989)

[1.4 论文组织结构 - 3 -](#_Toc515702990)

[2 相关技术和平台 - 4 -](#_Toc515702991)

[2.1 深度学习概述 - 4 -](#_Toc515702992)

[2.2 Tensorflow机器学习平台 - 4 -](#_Toc515702993)

[2.3 自编码器简介 - 5 -](#_Toc515702994)

[3 理论基础 - 8 -](#_Toc515702995)

[3.1 神经网络 - 8 -](#_Toc515702996)

[3.2 损失函数与学习率 - 9 -](#_Toc515702997)

[3.3 反向传播和梯度下降 - 10 -](#_Toc515702998)

[4 稀疏自编码器实现 - 12 -](#_Toc515702999)

[4.1 MNIST数据集 - 12 -](#_Toc515703000)

[4.2稀疏自编码器网络设计 - 12 -](#_Toc515703001)

[4.2.1 稀疏自编码器网络结构 - 13 -](#_Toc515703002)

[4.2.2 激活函数选择 - 15 -](#_Toc515703003)

[4.2.3 神经网络参数初始化 - 16 -](#_Toc515703004)

[4.3 自编码器和PCA算法的结果对比 - 17 -](#_Toc515703005)

[4.4 预训练与微调 - 18 -](#_Toc515703006)

[5 手机端数字识别实现 - 20 -](#_Toc515703007)

[5.1 系统设计方案 - 20 -](#_Toc515703008)

[5.2 tensorflow移植 - 20 -](#_Toc515703009)

[5.3 图像预处理 - 21 -](#_Toc515703010)

[5.3.1 opencv计算机视觉库 - 21 -](#_Toc515703011)

[5.3.2 图像预处理流程及主要算法 - 22 -](#_Toc515703012)

[5.3 总结 - 24 -](#_Toc515703013)

[6 总结与展望 - 25 -](#_Toc515703014)

[参考文献 - 26 -](#_Toc515703015)

[致谢 - 27 -](#_Toc515703016)

# 1 绪 论

## 1.1 课题背景及研究意义

当今世界正在飞速的向大数据方向发展，各个领域的数据量和数据规模都在以惊人的速度增长。这种数据规模的增长给各个行业的发展带来了巨大的机遇和变革，人们挖掘更多更广的数据，希望能从数据分析中得到隐藏在纷繁杂乱表面下事件和现象的客观价值规律。将同一事物的信息尽可能详细的汇总后起来，这就导致对事物的表述更加复杂，原本单一的事物表述迈向高维化。

高维数据包含了大量对事物的表述，同时也包含了大量的冗余并隐藏了重要关系的相关性，这给后期的数据处理带来了巨大的挑战。几个预测变量可能仅能反映数据某一方面的特征而不能建立全面的特征。以当下盛行的机器学习为例，为了提高机器学习模式识别的正确率，在训练模型时需要采集数据量巨大的数据特征集，这个数据特征集一般需要十万样本以上，（样本数据大小对机器学习的影响）如果单个样本图片的分辨率为60X80，在实际处理该类型的输入图像时输入维数便可达4800，数据维数过高会对深度神经网络的运行效率产生巨大影响，同时还会带来训练模型过拟合的问题。数据降维可以有效解决高维数据带来的种种不便，在经过对有效信息的提取综合及无用信息的摈弃之后，一张60x80的图片可以被降到600维甚至更低。

## 1.2 国内外研究现状

数据降维主要通过一些映射或者变换将样本数据同高维空间映射到低维的子空间。因为在优化过程中需要优化数据的结构去除数据中冗余的部分，所以优化后会获得一个高维数据的有效的低维表现形式。数据降维广泛应用于分类、可视化、通信、和高纬度数据存贮过程。目前主要的数据降维方法如图1.1所示。

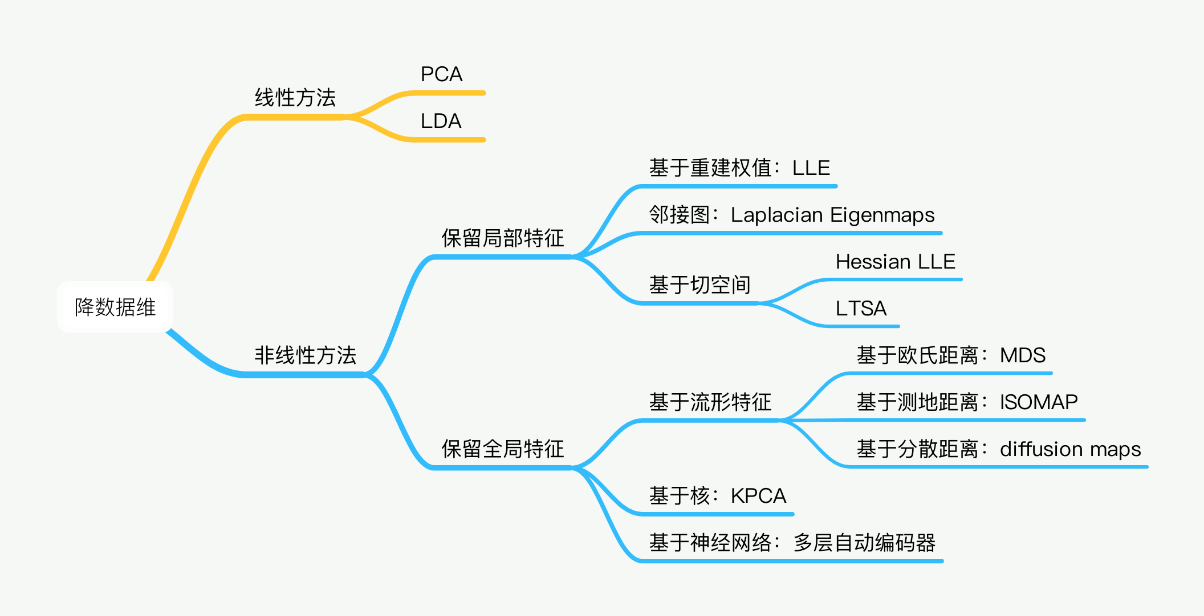


图1.1 主要数据降维方法示意图

数据降维方法可简单分为线性方法和非线性方法，其中经典的降维算法有是以主成分分析PCA为代表的线性投影法该方法将会在之后的章节进行对比分析，其他的算法还有以核主成分分析KPCA为代表的核方法、以等距映射算法ISOMAP为代表的流形学习算法。

PCA（Principal Component Analysis）是目前应用最广泛的一种经典无监督线性降维算法，其基本思想为通过最小化数据的重建误差去寻找一组正交变换基作为线性变换矩阵，并保持数据分布方差较大的特性来，近似地表示原始数据，从而达到数据降维的目的[11]。但是PCA算法构建的协方差矩阵大小是与样本的维度成正比的，因此计算非常高维的数据集的特征向量是非常困难的[11]。

PCA数据降维方法的复杂度较高，并且算法的目标太明确，降维后的数据中缺少次要信息且在实际处理中线性投影法不能解决非线性数据降维问题，而核方法、流形学习等非线性方法计算复杂度高，并且由于隐含层较少，只能处理一些简单特征，在复杂问题表现能力有限。为了解决上述问题，G.E.Hinton等提出了一种用于自动提取特征的自编码深度学习方法，构建一个具多隐藏层的深度学习模型，这个模型通过对训练数据进行编码解码学习，来提取能数据中的高级抽象特征，这个方法保证在重构误差最小策略下实现数据降维。

降维算法可以在一定程度上减小对高维数据进行降维，同时挖掘事物中隐含的特征和结构。当下的机器学习方法可以分为监督学习方法和无监督学习方法，监督学习算法如SVM，决策树，bosting，DNN，CNN等在开始训练前要事先给数据准备好标签。而当下机器学习需要巨量的样本数据，这样就导致很多时候标注数据集成本太高难以找到合适的训练数据。而无监督学习不需要事先准备样本，而是直接学习数据特征对数据进行聚类，因此无监督学习算法的适用性更广，自编码器就是基于的无监督学习实现的。

## 1.3 课题研究内容

本课题基于无监督学习的自编码器进在mnist数据集上进行数据降维的研究，主要研究内容包括以下几个方面：

1.基于无监督学习自编码器的设计、实现、原理，自编码器降为效果与经典的PCA算法的对比研究；

2.研究神经网络参数设置、激活函数、权重初始化等因素对自编码器性能的影响，并从其中找出性能较高的方案；

3.研究基于神经网络的的mnist手写字体识别，把训练得到的神经网络保存为模型文件。

3.研究自编码器与神经网络预训练和微调的方法，使用训练好的自编码器模型重构出部分mnist手写字体识别神经网络的参数，与没有经过参数调整的mnist手写字体识别神经网络进行对比；

4.将tensorflow和opencv移植到android手机上，开发出一个手机app，app可以根据opencv做图像预处理从黑纸白字的纸上分割出手写数字，使用tensorflow接口调用训练好的mnist手写数字识别模型来识别手机摄像头捕捉到的图片。

## 1.4 论文组织结构

本课题基于无监督学习的神经网络设计实现了对mnist数据进行降维的自编码器，并此基础上研究了自编码器在神经网络在神经网络训练和微调中的作用，基于android app设计实现了手机端手写字体的识别。

第一章主要介绍数据降维的意义，对目前主要的数据降维算法进行了介绍和简单对比，介绍课题研究内容以及论文组织结构。

第二章介绍本课题进行的技术基础以及实现平台，主要包括深度学习的概念、tensorflow机器学习平台、自编码器的原理。

第三章介绍神经网络的技术基础，主要有以下三个概念：前馈神经网络、损失函数、神经网络的反向传播。

第四章介绍自编码器的实现，包括自编码器网络的实现，以及网络结构、激活函数、全职初始化对整个神经网络的影响，最后介绍了自编码器和PCA算法的对比以及在预训练和微调中的作用。

第五章介绍在android手机平台实现手写字体识别的流程，包括tensorflow和opencv的移植，以及简单的图像预处理操作。

# 2 相关技术和平台

## 2.1 深度学习概述

如今，机器学习已经成为了人工智能领域中的一个重要的组成部分，自机器学习不断改进发展至今，已经广泛的应用于解决实际问题上面，在图像识别、语音识别、自动驾驶等人工智能领域取得了巨大成功。目前对于深度学习有一个比较为大众接受的定义为：“如果对于某类任务T和性能度量P，一个计算机程序被认为可以从经验E中学习是指，经过经验E改进后，他在任务T上由由性能度量P衡量的性能有所提升”[7]。

深度学习可以学习更加复杂的特征，相比较于其他的学习方法，在保证大量的训练数据的情况下，深度学习拥有更强的学习能力，这使得深度学习在很多方面取得重大突破，在图像分类、语音识别领域甚至达到了超越人类的的效果。

深度学习算法可以分为监督学习和无监督学习两种，监督学习在的训练样本带有标签，无监督学习的在训练过程中没有标签[15]。“大数据”时代使机器学习更加容易获得数据，深度学习一般需要5000个以上的标注样本才能达到比较不错的效果，则需要至少1000万个标注样本的数据集。目前的数据标注在很大程度上仍然依赖于人工，因此数据集的准备部分将会产生巨大的费用，因此无监督学习和半监督学习算法成为机器学习研究的主要侧重点，本课题中的自编码器正是无监督学习中的典型代表。

## 2.2 Tensorflow机器学习平台

Tensorflow是由Google开源的机器学习平台，在该平台上可以快速实现神经网络的搭建。本文课题中的神经网络全部是基于tensorflow平台实现。

TensorFlow支持C++和Python的接口，基于tensorflow提供的库可以快速搭建起神经网络，支持自由的算法表达，可以实现各种机器学习算法，且支持自动求导不需要通过反向传播求解梯度。Tensorflow可以在分布式的CPU和GPU上运行，且目前已经发布了可以在手机和嵌入式设备上运行的tensorflow lite版。图2.1为tensorflow架构，Tensorflow的底层使用C API进行编写，因此拥有广泛的移植性，可以在快速部署在各种设备上。

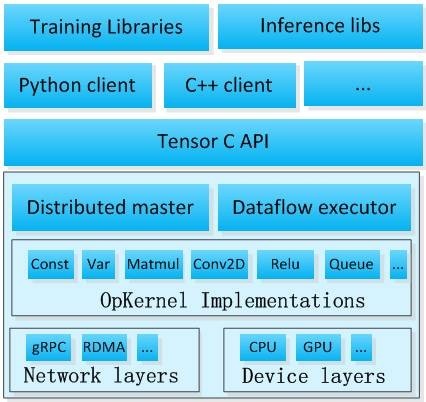


图2.1 tensorflow架构

## 2.3 自编码器简介

自编码器（autoencoder）是属于无监督神经网络，在实现上有全连接和卷积两种方式，在训练中使用输入数据作为模型评判的标准。自编码器根据结构的不同有欠完备自编码器、稀疏自编码器、去噪自编码器等[7]。

稀疏自编码器可以学习数据的潜在的联系和规律，建立抽象特征，可以用于分类这样的任务。自编码器被应用于数据降维可以有效的学习到物体特征。

自编码器属于深度学习框架的一种，有多个非线性的隐藏层组成，隐藏层之间相互连接由低层传递到高层，可以学习数据的规律和特征。深度学习中高阶表示中输入数据结构信息，拥有更强的学习能力。

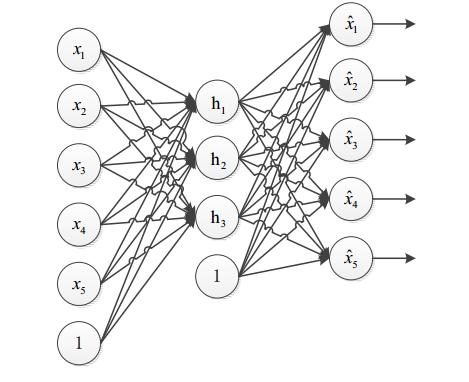
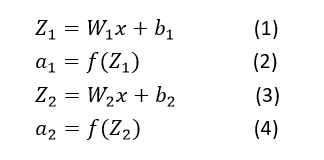


图2.2 简单自编码器结构图

一个最简单的自编码器如图2.2所示，该自编码器由三层组成从左到右依次为输入层、隐藏层、输出层，从输入层到隐藏层的转换相当于编码的过程，从隐藏层到输出层相当于一个解码的过程。自编码器是一个试图还原其输入的系统，自编码器的输入层维度和输出层维度相等，隐藏层的维度小于输入层和输出层，在训练中使输出层尽可能的等于输入层（实际上并不相等），当输出完全等于输入时则成为实现了对数据的无损编码，中间的隐藏层便是对数据特征的抽象学习。自编码器的编码和解码过程可由公式(1)至(4)表示其中a1为编码结果，a2为解码结果：



自编码器原理上属于典型的BP神经网络，是一种按照误差逆向传播算法训练的多层前馈神经网络[7]，在训练过程中可以通过反向传播来不断优化自身，调整自身参数，以达到最优的结果。使用自编码器对数据进行训练提取数据特征，自编码器的隐藏层可以看作输入数据的低维形式，即输入数据的特征，在训练的过程中通过不断调整自编码器隐藏层的参数和结构当输入与输出最接近时，自编码器的性能最好，自编码器的结构可以让自编码器在确保重构误差最小策略下对数据进行降维。

自编码器属于无监督学习的一种，可以有效的对数据进行降维，提取数据的高维特征。经过降维后的数据进行训练可以得到更好的鲁棒性，同时使用自编码器等无监督学习方法预训练整个神经网络的初始权值已经是当下DNN、CNN等主流机器学习算法的重要步骤。在2006年以前，对深度神经网络的训练方法为：先随机初始整个网络的权重，之后在训练过程中使用反向传播函数来优化权重。这样训练的神经网络泛化能力很差，结果往往难以得到预期，根据研究表明（文献）主要由以下三个原因：1.监督学习的数据集过小；2.局部极值和全局极值；3.梯度弥散问题。该问题被 “逐层贪婪的无监督学习算法”解决，训练一个以受限玻尔兹曼机为初始模型的深度自编码器，取得了很好的数据降维效果。对于一个监督学习的神经网络，先由无监督学习的特征检测模块对网络中的权重进行无监督的预训练，将训练得到的权值用来初始化整个神经网络，之后开始监督学习的神经网络的训练[12]，对预训练的权值进行优化，可以获得更好的训练模型。预训练可以得到比随机初始化更低的误差，同时可以让模型学到抽象级别的不同特征，抽象的级别随网络层的增高而加深。自编码器由于更加容易理解的理论基础，被广泛的应用于深度学习和数据降维中。

# 3 理论基础

## 3.1 神经网络

机器学习中的神经网络模型受人神经科学的启发，每个节点与人神经元的构造非常相似。目前使用比较广泛的一种定义为“神经网络是由具有适应性的简单单元组成的广泛并行连接的网络，他的组织能够模拟生物神经神经系统对真实的世界物体做出交互反应”[8]。图3.1是神经元模型，这个神经元的输出y的计算公式为：

在这个神经元中，神经元可以接受n个输入，这些输入通过带有权值的边进行传递，将输入加权求可以得到一个总值，将总值通过阈值化处理之后通过“激活函数”处理产生输出，来模拟一个神经元的工作过程。最后神经元输出一个数值y到下一层神经网络的神经元中，神经网络的在训练过中不断的优化神经元边上的参数，以达到更优的效果。



图3.1 神经元模型

多层的神经网络如图3.2所示，将许多的神经元按指定的结构连接起来，就可以得到我们机器学习中的神经网络。每一层神经网络可以有许多的神经元，如果神经元的运算是线性的，那么整个网络便是线性的，否则便是非线性。同一层的神经元之间彼此不连，每层神经网络之和下一层的网络互联，，作为下一层的输入，这样的神经网络通常称为“前馈神经网络”。前馈神经网络的最后一层是输出层，代表神经网络对输入数据的预测结果。以Mnist的前馈神经网络为例，该神经网络的输出层是一个对十个数字概率预测的十维数组，数组中的每一个节点对应一个类别的预测概率。

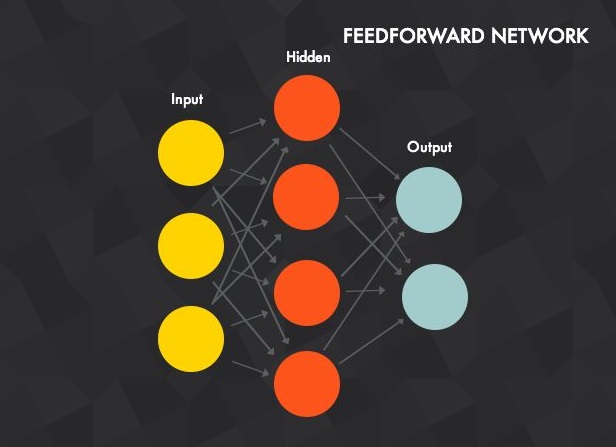


图3.2 简单神经网络结构

## 3.2 损失函数与学习率

前馈神经网络的每一次运算都是对输入目标的一次预测，想要让神经网络的输入值更加接近真实值，就要不断比对预测结果与真实值之间的差异，以便对神经网络的权值进行调整，起这种功能的函数便是损失函数（loss function or objective function），这里用loss来代指神经网络的损失。在神经网络中损失函数的输出值越大表示预测值与实际值的差异越大，在神经网络的训练过程便是要尽量缩小loss的过程。在本课题中用到的损失函数为reduce\_mean损失函数和softmax\_cross\_entropy损失函数。

softmax\_cross\_entropy损失函数由softmax回归函数和cross\_entropy交叉熵函数两部分组成。softmax将神经网络的输出变成一个概率分布，用概率分布来表示为不同类别的概率分别是多大，如果一个神经网络的原始输出为，那么经过Softmax回归之后的输出可以表示为：

使用softmax函数对神经网络的结果进行回归之后，使用交叉熵来计算该结果与正确结果之间的预计。交叉熵是是信息论中的一个概念，在这里他用来评判两个概率分布之间的距离，给定两个概率分布p和q，则以q来表示p的交叉熵为：

其中p代表正确答案，这个正确答案通过我们的标签进行提供，q代表预测值的概率分布，使用交叉熵来计算q和p之间的距离便是神经网络的loss。由于手写字体图像的像素值范围在0~1之间所以可以推得使用reduce\_mean函数计算loss的时候loss等于自编码器重构图像平均到每个像素点的错误率。

## 3.3 反向传播和梯度下降

得到损失函数之后，要调整网络的参数来使loss尽可能的小，这里便要用到反向传播算法，反向传播算法是目前最为经典的算法，在许多网络中都被广泛使用。

假设有一个函数y = f(x)，我们使用这个函数的导数f’(x)来表示f(x)在点x处的斜率，根据倒数我们便可以知道如何更改x来改善y，使y达到最小。在神经网络里面，我们每次移动指定步长的x来使得神经网络的损失函数最小，移动的步长便是神经网络的学习率，这种方法叫做梯度下降。在神经网络中输入是多维的，要找到整个神经网络损失函数最小的全局最优解，所以神经网络引入梯度来表示相对一个向量求导的导数：f，f的倒数包含包含所有偏导数的向量，记为。梯度的第i个元素是f关于的偏导数。为了最小化f，我们计算f的方向导数得到f下降得最快的方向，与倒数的概念一样，这里我们要计算f的方向导数：

其中是u与梯度的夹角，在负梯度上移动可以减小f，这被就是最速下降法(method of steepest descent)或梯度下降(gradient descent)。梯度下降建议新的点为：

传统的神经网络训练效果与神经网络的参数初始化有很大关系，存在极大的随机性。

在神经网络的训练中需要设置神神经网络的学习率，对应到反向传播算法中就是每次移动的步长，即每次根据计算梯度向某个方向移动。在训练过程中一般使用衰减的学习率，或者指定每层神经网络的学习率，在刚开始训练时使用较大步长来移动到最优的结果区域，随着迭代层数的加深逐渐减小学习率，向最优的结果靠近。在梯度下降的过程中，需要寻找loss的全局最优点，但在实际训练中会陷入图3.3所示的局部最优和全局最优的问题。

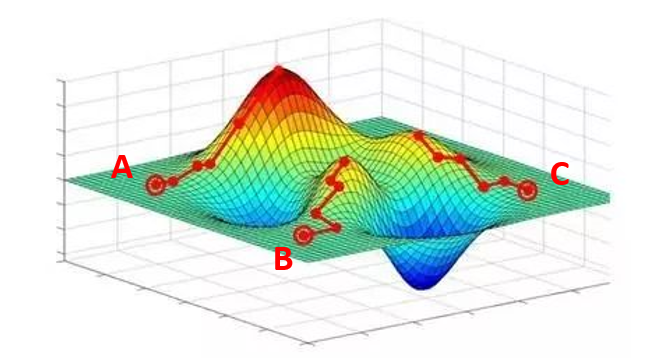


图3.3 局部最优与全局最优

假设在某个神经网络的训练中需要寻找全局最大值，在神经网络的参数初始化时，由于参数初始化时有很大的随机性，所以初始化结果可能位于A、B、C三类点。当参数初始化为A类点时，神经网络可以沿着梯度很快收敛到全局最答点，但是如果参数初始化在B类点或者C类点时，神经网络便会沿着梯度收敛到局部最答点。在实际的神经网络训练中，可能会涉及到上百万个参数的初始化，想要在参数初始化时达到最优的A类点几乎时不可能的，更多的时候神经网络会收敛到B类点或者C类点这样的局部最优点。

在训练神经网络学习复杂特征以解决现实中的某个问题时，由于神经网络在参数的初始化方面有极大的随机性，会对之后的训练结果产生极大的影响。所以早期的深层神经网络的优化一直是难以解决的问题，制约了整个神经网络的发展，最后这个问题被无监督的学习模型所解决，这部分将在本文4.4小节介绍。

# 4 稀疏自编码器实现

## 4.1 MNIST数据集

MNIST 数据集由来自 250 个不同人手写的数字构成，它包含70000张手写数字的灰度图片，其中每一张图片包含28 x28 个像素点，对应的数据维度为[28，28，1]，图4.1是一些MNIST数据集的样本图片。MNIST数据集的标签为0~9的十个阿拉伯数字，在训练中可以拆分为55000张训练数据集、5000张的测试数据集、10000张的测试集。

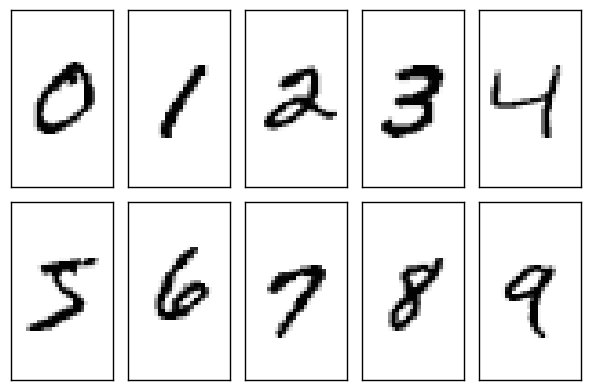


图4.1 mnist数据集示例图片

在训练过程中，通过对数据进行学习调整，优化网络参数，最后训练出神经网络模型；验证集对训练出来的模型调整分类器的参数，确定神经网络机构以及控制模型复杂度，优化模型等；测试集对训练的神经网络参数做准确性验证，确定模型在某一场景下的准确率。

## 4.2稀疏自编码器网络设计

如本文第3小节所述，神经网络的训练过程可以图4.2所示的三步来完成，1.前向传播获得神经网络的预测结果；2.计算loss得到预测结果与真实结果之间的距离；3.通过反向传播来优化神经网络中的参数，向梯度中的更优解方向进行优化。



图4.2 神经网络训练过程图

本文2.2小节简单介绍了自编码器的概念、性质、应用等特点，自编码器的特殊性体现在输入层维度和输出层维度相同，在训练过程中是的自编码器的输入尽可能的等于输出，自编码器的隐藏层单元数要小于输入输出层的维度而且整个神经网络是非线性的，这样在训练过程中自编码器就必须要去学习高维数据的低纬表示，这样在经过很多轮迭代之后自编码器可以很好的学习到数据特征。

在神经网络的训练过程中，自编码器最后结果的准确率影受神经网络结构、激活函数的选择、神经网络参数初始化方法等参数的影响。

### 4.2.1 稀疏自编码器网络结构

由于MNIST数据集相对来说较为简单，所以整个神经网络的结构设置上不能过深，同时自编码器中间节点的数据维度越低，自编码器还原出的图片越粗糙。

图4.3是本文稀疏自编码的网络结构，该网络的输入层可以填入[None, 784]大小的数据，由于Mnist数据集是28x28的图片，对应的数据维度为784。接下来的w2、w3、w4层相当于编码层，经过784-256-128-80三层神经网络的学习，w4层便是自编码器学习到的数据特征。w4，w5，w6相当于解码的过程，将w4层学习到的图片特征进行还原，重构出原始图像。由于整个网络结构是非线性的，所以编码和解码的过程其实是不可逆的，不能能由编码的过程反推到解码的方法，自编码器学习的是整幅图像数据特征的低维表达，使得网络的输出尽可能的等于输入，但是实际上无损编码是不存在的，自编码器不能完全还原回图片。

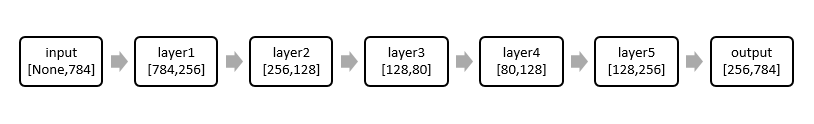


图4.3 自编码器网络结构

神经网络的图结构是影响神经网络的重构数据的准确率的重要因素之一，图4.4是不同神经网络的loss对比，3.2小节介绍过使用reduce\_mean函数计算自编码器网络的loss，其loss等同于准确率。可以看到浅层神经网络的收敛速度更快，但是准确率受中间隐藏层节点的影响在97%~96%之间波动，中间层设置为80层的神经网络收敛速度要慢一些，但是准确率更高最后准确率收敛到99.5%左右。

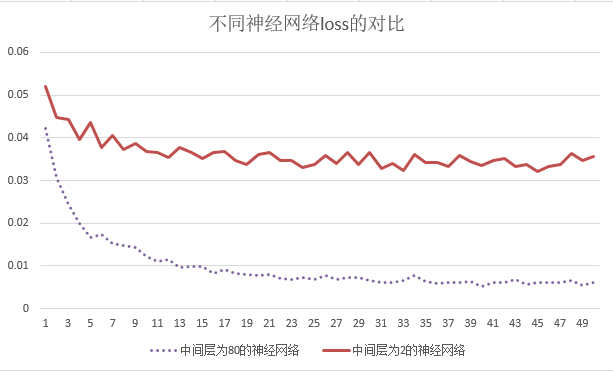


图 4.4 不同神经网络的loss对比

图4.5为神经网络中间层设定为2层时在二维坐标系中的打印结果

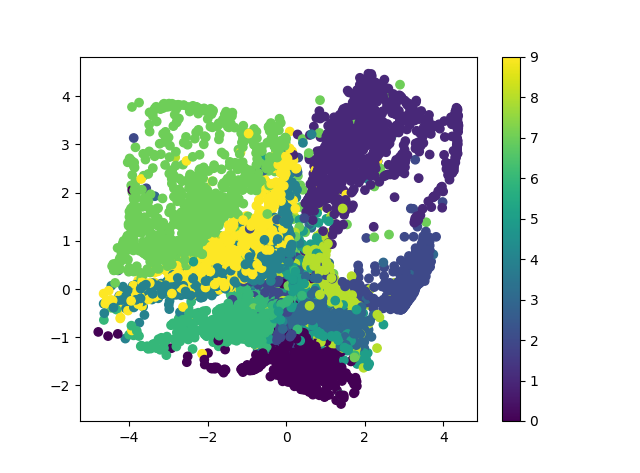


图4.5 二维自编码器聚类结果

可以看到即使神经网络的中间结点设定为2维整个神经网络的计算结果依然是聚类的。0~9十个不同的数字分为十个不同的颜色，根据神经网络中间层2维的数据内容，将测试数据集的图片放入到坐标系中。由于中可以得到，即使是在2维的神经网络中，神经网络依然可以学习到数据的部分特征，绘制出的坐标图在形式上依然是聚类的，但是准确率偏低，有许多的图片被分类到错误的类别中去。由此可以得出结论，在MNIST数据集上面神经网络可以很好的学习到数据的特征，将784维的图片用80维的数据来表示的时候，图片重构的结果准确度在99.5%，平均到每张图片有4个左右的像素点错误，图4.6为前一行选取了10张测试图片，后一行选取了一张自编码器重构出的图片，肉眼几乎难以发现两张图片存在的差别，而图4.7为准确率在96%时重构的出图片，可以明显看到重给出的图片存在许多的噪声。

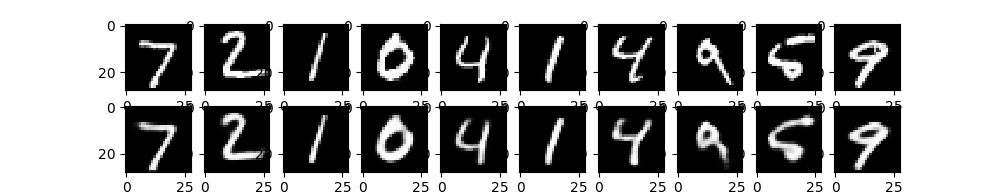


图4.6 重构效果图1

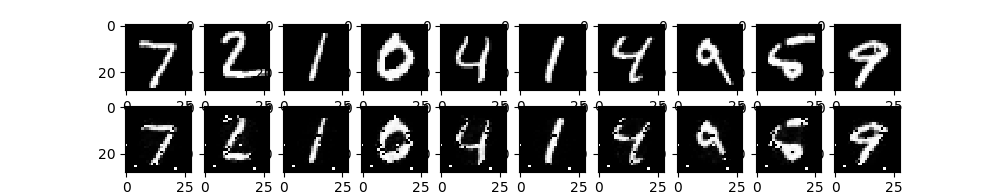


图4.7 重构效果图2

### 4.2.2 激活函数选择

在神经网络中激活函数主要用来将线性的网络变为非线性。在线性的神经网络中，模型的输出y和输入x满足以下关系：

那么线性模型最大的特点是任意线性模型的组合仍然还是线性模型，而在前向传播的算法中如果只有线性变换，根据线性变换的性质，多层的线性网络都用一个单层的神经网络来表达，二者等价。线性模型能够解决的问题是有限的，如果一个模型可以可以通过直线来划分平面比如最简单的回归问题，那么线性模型也可以用来解决这种问题。然而在现实世界中，在深度学习中需要解决许多现实世界中的复杂问题，这些复杂问题至少是无法通过直线或者高维空间的平面划分的，那么要解决现实世界中的复杂问题，便需要激活函数将线性的神经网络转化为非线性的神经网络。

本课题选取了目前最常用的两种激活函数sigmod激活函数和relu激活函数来作比较。Sigmoid函数是一种有界的可微实函数，函数曲线上的任一点对应的导数值均为正值，函数具有“S”形状，其函数公式如下：

Relu函数把所有的负激活（negative activation）都变为零。这个操作增加模型乃至整个网络的非线性特征，对卷积层的表达效果不产生影响，可以用来解决因梯度消失造成的神经网络学习收敛慢的问题，其公式如下：

由于MNIST的像素范围比较小，Sigmod在应对较小的输入和输出的时候会产生比较好的结果。

图4.8为使用Sigmod和Relu激活函数的准确率对比，可以观察到Relu函数收敛速度很快，很快就收敛到98%附近，可以看出在处理小数值的网络结构时Sigmod函数取得的效果更好准确率依然是99.5%。

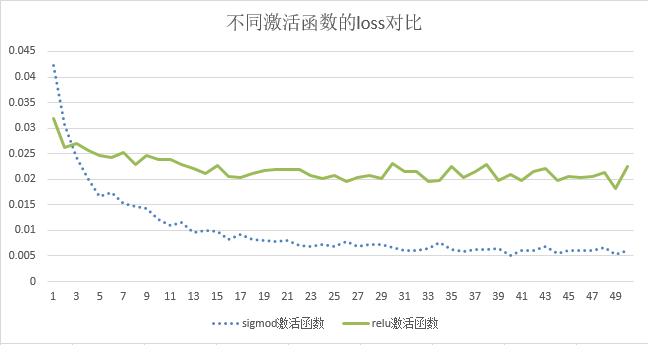


图4.8 不同激活函数loss对比

### 4.2.3 神经网络参数初始化

自编码器是一个典型的BP网络，由于BP网络在梯度下降进行优化的时候要受到局部最大值和全局最大值的影响，所以神经网络在训练时产生的最终结构有非常大的随机性，初始点能够决定算法是否收敛，有的初始点十分不稳定会是该算法遭遇数值困难，因此神经网络参数的初始化对会对整个网络的准确率产生影响。

通常情况下，可以对神经网络参数随机初始化参数。在参数随机初始化的过程中，较大的初始权重可以有助于避免在每层线性成分的前向或反向传播中丢失信号，但是如果初始权重太大，那么会在前向传播或反向传播中产生爆炸的值。较大的权重会产生使激活化饱和的值，导致饱和单元梯度的丢失，以上这些因素都决定了权重的理想初始大小，因此关于如何初始化权重正则化和优化有着不同的观点，优化观点建议应该足够大以成功传播信息，正则化希望初始权重小一些[8]。

图4.9是不同初始化参数方法的loss对比，在本课题中权重初始化从截断的正太分布中初始化权重，与完全随机的初始化权重相比准确率更高。

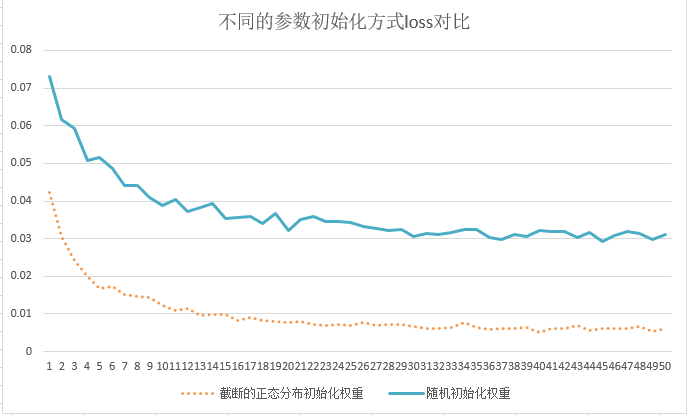


图4.9 不同的参数初始化方式的loss对比

## 4.3 自编码器和PCA算法的结果对比

在本文1.2小节中简单的介绍了PCA算法，使用受限玻尔兹曼机训练了一个神经网络并与PCA算法的处理结果或做出对比[1]。对比如图4.10所示，左边为PCA的结果，右边是自编码器的结果。自编码网络784−400−200−100−50−25−6的编码网络和一个6−25−50−100−200−400−784的解码网络组成。编码层的6个结点都是线性的，其他结点都是逻辑的（二进制的）[1]。该网络用20,000张图片训练，并用10,000张新图片测试[1]。这个自编码器将784维的数据映射到6维，并几乎完美重建了图片，比PCA给出的重建效果好得多[1]。

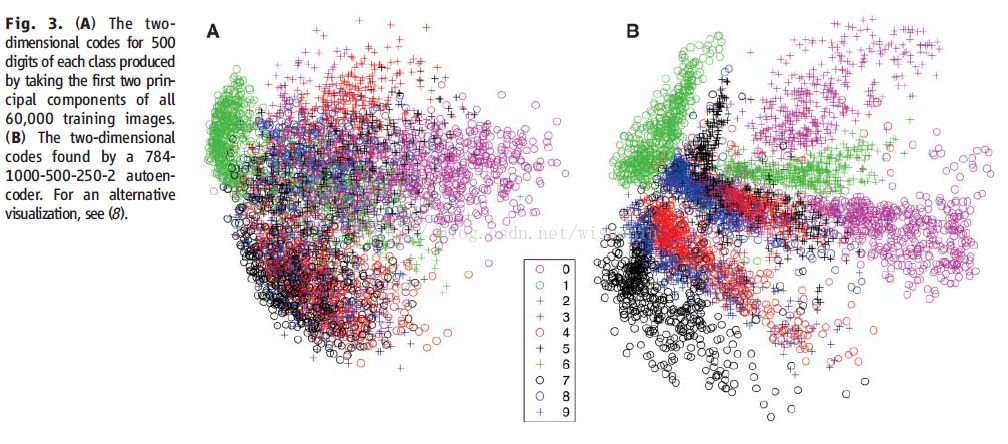


图4.10 自编码器与PCA的对比

## 4.4 预训练与微调

在早期的神经网络训练中，如果在神经网络参数完全随机的情况下，那么训练结果也有很强的随机性。BP算法被证明在浅层神经网络中是有效的，基于梯度下降可以找到全局最优点，而对于更深层的神经网络效果不佳。在深层神经网络中通常难以解决全局最优和局部最有问题，而复杂的问题往往需要对对象做更高级的抽象，浅层网路难以达到理想效果，这个时候就必须依赖深层网络这就导致BP算法难以在实际应用中取得理想效果。

2006年，Hinton等人提出用深度信念网络（DNB）的无监督学习算法，解决深度学习模型优化困难的问题。深度信念网络是以受限玻尔兹曼机为基本单元搭建的信任网络，采用了逐层初始化和整体反馈的方法，成功克服了深层网络难以训练的难题[15]。基于无监督学习技术提出贪婪逐层预训练学习过程初始化深度学习模型的参数，对神经网络进行预训练，预训练可以学习到目标网络的特征，之后将与训练网络的参数直接用于深层网络的参数初始化，堆栈各层神经网络转换为深度监督前馈神经网络，用梯度下降进行微调，便可以获得较高的准确率[14]。

图4.11为Mnist手写字体字体识别的网络结构图，图中输入层以及中间的隐藏层layer1~layer3与自编码器网络的前几层完全一样，当自编码器模型训练完成后便可以把自编码前四层的参数直接用到mnist手写字体识别的神经网络中，完成参数初始化的过程。与自编码器神经网络不同的是，这里的激活函数使用Relu激活函数，最后输出层的维度是十代表0~9是个数字的预测结构，以及最后的损失函数使用softmax\_cross\_entropy激活函数。



图4.11 Mnist手写字体识别网络

Mnist手写数字识别神经网络的目标是识别出一副28x28图像中的0~9十个数字，该神经网络属于监督学习，在Mnist的数据集中为每张图提供一个标签对应图像中的数字，在训练过程中神经网络计算预测结果与标签的loss，之后来优化神经网络，在全连接的神经网络中，目前对mnist数据识别的准确率最高为98%，卷积神经网络则可以达到99%。图4.12完全随机初始化变量的神经网络与部分使用稀疏自编码器网络参数的神经网络在MNIST手写字体识别效果上的对比，可以看到自编码器优化之后神经网络收敛的速度更快，同时取得的结果更优。

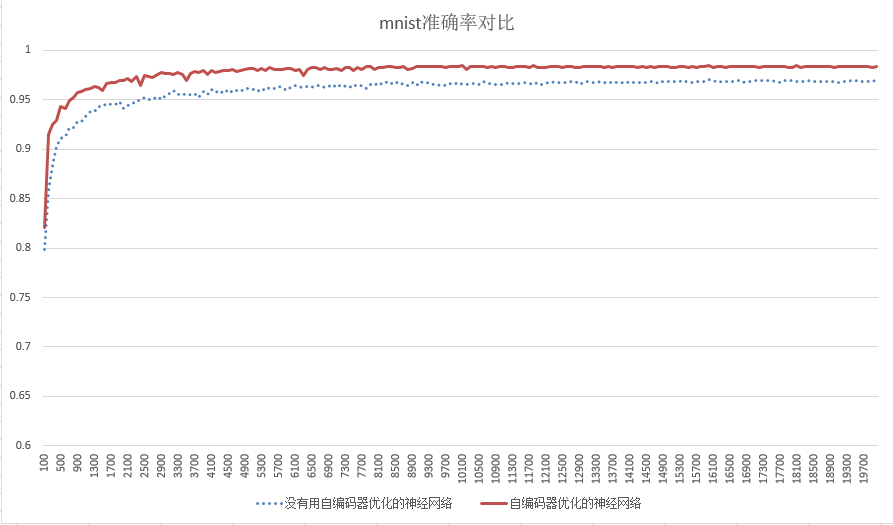


图4.12 两种神经网络准确率对比

# 5 手机端数字识别实现

## 5.1 系统设计方案

在之前的章节本文介绍了如何训练自编码器，并用自编码器的神经网络参数优化MINST手写字体识别模型，在完成了模型优化之后生成的模型就可以对手写的字体进行识别，因为模型的输入图象必须满足一下两个条件：1.单通道图像；2.图像大小必须是28x28。为了实现对任意手写的字体进行识别，选择用android 手机app作为系统实现平台，系统结构如图5.1所示。

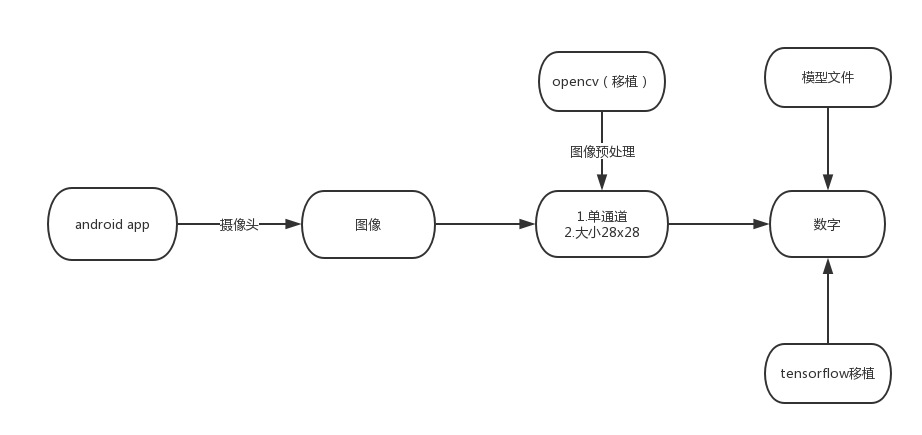


图5.1 android系统流程图

数字识别APP的识别环境是白底黑字的手写数字，app直接从手机摄像头获取图像数据，之后由移植到android平台上的opencv计算机视觉库对图像进行预处理，从前景中分割出数字图像，将进行了预处理之后的图像经过移植的tensorflow接口填充进入手写字体识别的模型中，由模型预测对当前数字做出预测并输出到屏幕上。

## 5.2 tensorflow移植

移植tensorflow需要将编译tensorflow的源码得到tensoerflow接口文件，之后使用android JNI开发的方式将编译tensorflow得到的接口库文件导入android工程。JNI是Java和C/C++相互通信的接口，本文2.2介绍过tensorflow的架构，tensorflow的底层架构是基于C API的因此具有很强的移植性，可以移植到各个平台，图5.2是android架构图

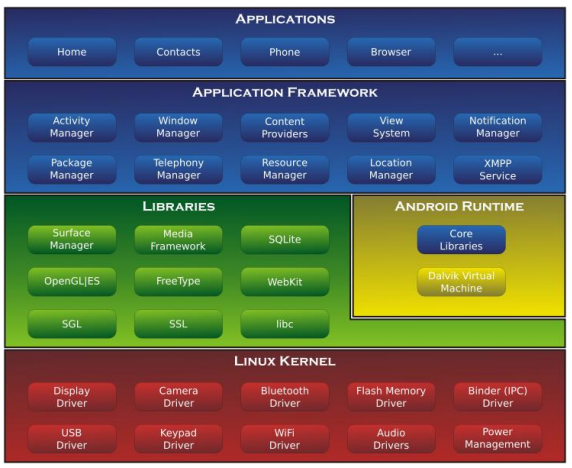


图5.2 android架构图

android系统架构大体上可以分为四层：1.Linux内核；2.系统运行库层；3.应用框架层；4.应用层。由于在系统运行库这一层中Android系统的核心库使用了大量的C/C++库，Android运行时库也在这一层，如Dalvik虚拟机就是Android运行时库中的一部分。JNI可以直接使用本地代码库(即C/C++代码)，并通过 Dalvik 虚拟机与应用层和应用框架层进行交互，因此通过JNI的方式移植tensorflow接口库到android工程里面，就可以在代码中调用训练好的模型文件对手写字体图像进行预测。

## 5.3 图像预处理

### 5.3.1 opencv计算机视觉库

近年来，数字图像处理发展迅速，实用价值高，应用范围极为广泛，现已应用于军事技术、政府部门和医疗卫生等多种领域[17]。随着数字图像处理的日益广泛。OpenCV是一个基于BSD许可（开源）发行的跨平台计算机视觉库，可以运行在Linux、Windows、Android和Mac OS操作系统上，它由轻量级而且高效由一系列 C 函数和少量 C++ 类构成，同时提供了Python、Ruby、MATLAB等语言的接口，实现了图像处理和计算机视觉方面的很多通用算法[19]。

在数字图像中每一副图像都是一个像素点的矩阵，图像的每一行每一列各有多少个像素点便是我们通常所说的分辨率。图像的通道数则表示每一个像素点可以有几个分量，比如RGB的三通道模式，每个像素点由三个字节组成，这三个字节分别代表红、绿、蓝三个颜色，即一幅图像可以有中不同颜色的表现方式。在本位5.1小节提到的灰度图则代表单通道图像，一个像素点只用一个字节来表示，0为黑色，255为白色。

由于移植的tensorflow模型需要将目标图像调整到指定大小和通道，以及手机摄像头拍摄到的图片是不能直接使用的，需要对图片做预处理将数字从图片中分割出来，然后转换为指定格式才能填充入模型进行识别。所以同样需要使用android JNI开发来把opencv移植到android端。

### 5.3.2 图像预处理流程及主要算法

图像预处理的流程如图5.3所示：图像阈值处理将彩色的图片变为二值图像（黑色和白色）；腐蚀和膨胀操作用来消去图像中比较小的噪声，同时连通像素点；针对图像中每一个存在的前景寻找到它的轮廓，拟合出轮廓的外接矩形，并将这个外接矩形从整张图像中分割出来；经过以上几个步骤之后图像变成单通道，之后只需要将图片大小调整至28x28就可以直接放入神经网络中进行预测。

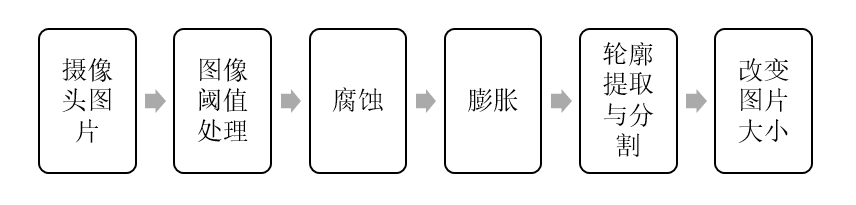


图5.3 图像预处理流程

图像阈值化操作就是去除掉一些低于或高于某个设定值的像素，其基本思想是，给定一个数组和阈值，小于阈值便置为0大于阈值便置为最大，本系统使用反二进制阈值化对图像进行处理，其处理方式为:

进行了阈值化处理之后的效果如图5.4所示，可以看出此时的图片与Mnist的图片在颜色分布上已经是一样的了。

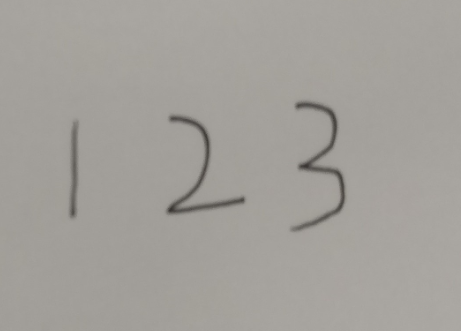
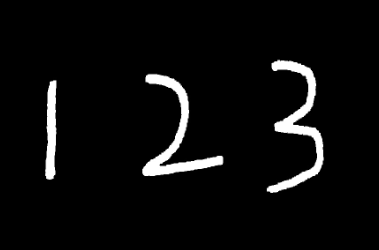
 

图5.4 阈值化处理效果图

膨胀操作将图像与指定大小的核进行卷积，这是一个是求局部最大值的操作，核与图像卷积，即计算核覆盖区域的像素点最大值，并把这个最大值复制给参考点制定的像素，这样会使图像中的高亮区域逐渐增长[18]。腐蚀是膨胀的饭操作，腐蚀操作同样是与核卷积，不同的是腐蚀操作计算的是区域像素的最小值，计算被吓覆盖的区域像素的最小值[18]。腐蚀能够消除细的凸起，可以用来消除图像中的噪声，切保证图像内较大的区域仍然存在，而如果想要连通图像的区域则使用膨胀操作[18]，其公式如下：

图5.5腐蚀和膨胀处理之后的图像，可以看到进行操作之后的数字的像素值比操作之前要更强。

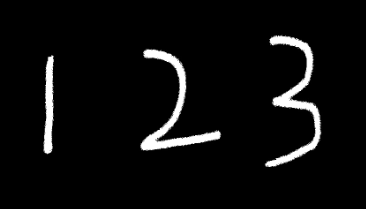
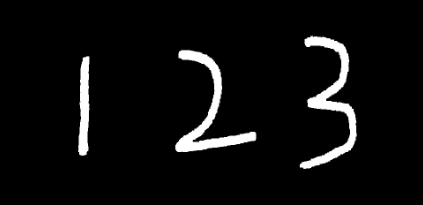
 

图5.5 腐蚀和膨胀效果图

在二值化图像中查找轮廓其实只需要沿着每个数字的白色外边框就可以描绘出图像的轮廓信息，之后根据每个边框的位置信息，拟合出外边框的外接矩形，之后将序列的外接矩形从图像中分割提取出来，就可以得到每个数字，将每个数字的图像大小变换为28x28大小的图片就可以直接放到神经网络中进行识别，图5.6是对腐蚀和膨胀之后图像的轮廓的寻找结果。

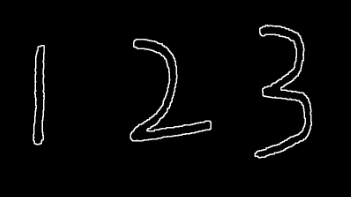


图5.6 轮廓寻找结果

## 5.3 总结

通过上述方法，在不考虑Android手机端计算资源的情况下，可以将大部分基于Tensorflow的模型部署到Android。手机摄像头负责图像采集，移植到Andoird手机上的Tensorflow接口负责调用神经网络模型处理图片，基本上所有的模型对输入图片的格式都要要求，例如手写字体识别神经网络，要求图片是28x28大小的单通道图像，调用opencv对图像进行预处理将图片转换为指定的输入格式。图5.7左是手机端进行手写数字识别的效果图，图5.7右则为移植了物体检测模型之后的效果图（模型来自于Google）。

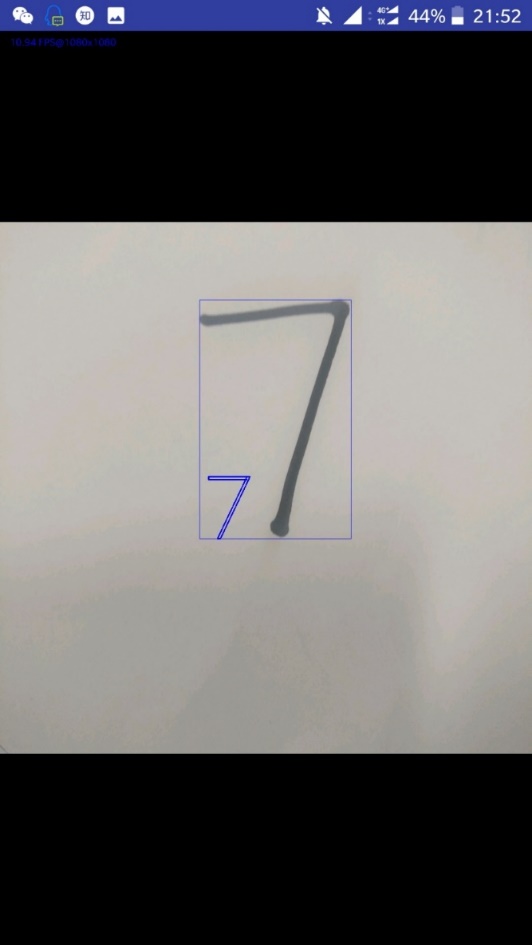
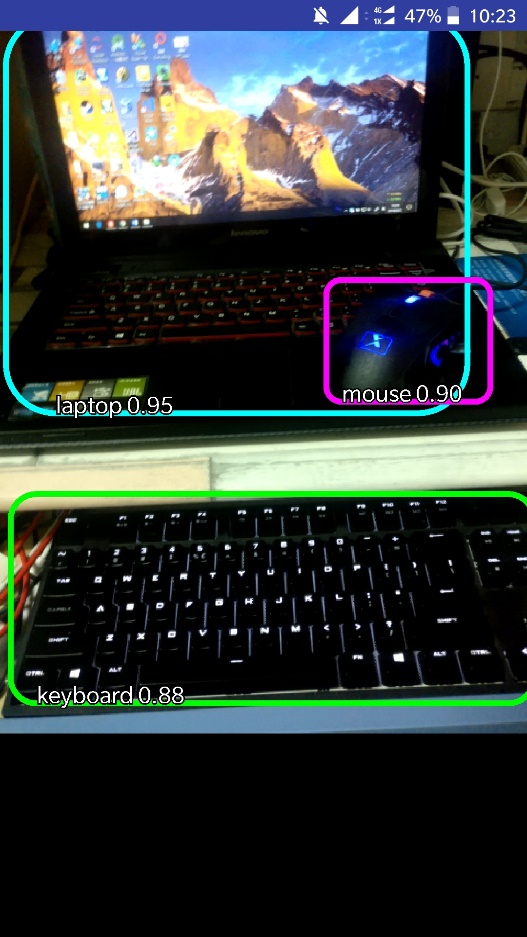
 

图5.7 识别效果图

# 6 总结与展望

本次课题实现了基于全连接神经网络的数据降维自编码器，在数据降维率和精度上都超过了开题报告时的预期，但是本次课题没有对卷积和反卷积进行研究，未能基于卷积神经网络的实现自编码器，卷积神经网络对于特征的提取与学习要更优于全连接的神经网络，如果能拓展到基于卷积的自编码器那么在性能方面将会再度提升，这是本次课题的不足之处。由于本次课题的数据训练部分全部在我的笔记本电脑上实现计算资源有限，对于降维的数据只能选择mnist数据集，同时训练的对比结果也不是很充足。本次神经网络训练的训练都是基于非常经典的参数初始化设置，在训练中收敛速度非常快，由于参数设置全部随机或者初始值全部为定值的神经网络训练耗时太长，也未能加入对比。在手机app进行手写字体识别的环节，由于全连接神经网络和卷积神经网络对图像的角度旋转均不能很好的处理，所以对角度旋转超过一定角度的字体不能进行识别，同时图像预处理环节的效果有待提升，app可以进行进一步的优化提升准确率。

以上是关于本次课题的总结，更多的是关于本次课题未能实现的部分。希望以后能有更多的机会从事深度学习和人工智能方面的研究，同时也希望能有机会对此次课题中不足的机会进行完善。

# 参考文献

[1]G. E. Hinton and R. R. Salakhutdinov.Reducing the Dimensionality of Data with Neural Networks[J].Science,2006,313,(5786).

[2]Asja Fischer (19) (20) Christian Igel (20).An Introduction to Restricted Boltzmann Machines[J].Lecture Notes in Computer Science，2012,7441,(1).

[3]Unsupervised Multi-modal Learning[J].Lecture Notes in Computer Science,2015,9091,(1).

[4]J. A. Gómez-Ruiz, J. Mu?oz-Perez;M. A. García-Bernal.Spicules for Unsupervised Learning[J].Lecture Notes in Computer Science,2009,5517，(1).

[5]Anush Sankaran , Mayank Vatsa, Richa Singh , Angshul Majumdar.Group sparse autoencoder[J].Image and Vision Computing,2017,60,(Complete).

[6]Michael Kampffmeyer, Sigurd Løkse, Filippo M. Bianchi, Robert Jenssen, Lorenzo Livi.Deep Kernelized Autoencoders[J].Lecture Notes in Computer Science,2017,10269,(1).

[7]BENGIO Y,Lèarning deep architectures for AI［J］.Foundations and Trends in Machine Learning，2009，2(1):1-127．

[8]赵申剑等译.Deep Learning.人民邮电出版社,2017.8

[9]李越.OpenCV应用现状综述[J].工业控制计算机,2017,30(07):123-124+126.

[10]甘炎灵. 高维数据降维及其在图像识别中的应用[D].华中师范大学,2017.

[11]黄少立. 基于稀疏表示的降维算法研究[D]. 西北农林科技大学, 2011.

[12]鲁亚平. 面向深度网络的自编码器研究[D].苏州大学,2016.

[13]岳永鹏. 深度无监督学习算法研究[D].西南石油大学,2015.

[14]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(07):1921-1930+1942.

[15]殷瑞刚,魏帅,李晗,于洪.深度学习中的无监督学习方法综述[J].计算机系统应用,2016,25(08):1-7.

[16]贾文娟,张煜东.自编码器理论与方法综述[J].计算机系统应用,2018(05):1-9.

[17]秦小文,温志芳,乔维维.基于OpenCV的图像处理[J].电子测试,2011(07):39-41.

[18]于仕琪,刘瑞桢.学习opencv.清华大学出版社.2009.10

[19] https://baike.baidu.com/item/opencv/10320623

# 致谢

这个论文从选题到后来的算法实现，对我来说完全是一个全新的领域，开始的时候我并没有十足的把握能够完成这个课题，但是到最后一步一步走来我学到了许多机器学习方面的知识收获很大。再次感谢我的指导老师向毅老师在论文期间提供的指导和帮助。至此我完成了我的毕业论文，大学四年的生活划上了句号，这四年来我经历了许多也学到了许多，在此感谢在我大学四年中给我提供过帮助的老师和同学们，感谢学校对我的培养，感谢一起走过大学的人们。