

基于 BP 神经网络的手写数字识别技术

目录

第一章 综述.....	1
1.1 研究背景	1
1.2 国内外研究现状	1
1.3 论文结构简介	2
第二章 手写数字识别方法及原理.....	2
2.1 手写数字识别方法及难点	2
2.2 手写数字图像的预处理	4
2.3 手写数字的特征提取	
第三章 基于 BP 神经网络的手写数字识别设计方案	5
3.1 问题分析	6
3.2 人工神经网络简介	7
3.3 BP 神经网络原理	8
3.4 系统设计方案及流程图	10
第四章 系统的实现与结果分析	11
4.1 MATLAB 简介	11
4.2 MATLAB 程序设计	11
4.3 MATLAB GUI 界面的设计	14
4.4 结果分析与改进	18
第五章 基于 Matlab 的遗传算法优化 BP 神经网络.....	19
5.1 模型建立	19
5.2 基本思想	20
5.2.1 求解全局最优解.....	20
5.2.2 算法实现.....	20
5.3 Matlab 实现.....	22
5.4 调参方案与结果分析.....	25
5.5 优化方案总结与待改进问题.....	27
第六章 总结与展望.....	28
6.1 实验总结.....	28
6.2 未来展望.....	28

第一章 综述

1.1 研究背景

数字是人们日常生活中信息交流的重要载体，但是由于手写习惯的差异，因此手写数字的形式因人而异。大量不同书写习惯的数字就给人们的肉眼识别带来极大的负担，故基于计算视觉的手写数字识别技术引起国内外研究学者的关注，如何实现高效、快速的识别是手写数字识别技术的关键。

阿拉伯数字作为广泛应用于各个国家的数学符号，是手写数字识别技术中关注的热点与主要研究对象。事实上，手写数字识别技术是字符识别技术的分支之一，其在邮政编码、赛事牌号、福利彩票、银行票据和排队等号中具有极大的应用价值，同时其识别也是图像处理、模式识别中的一个重要研究热点。

随着世界经济的迅猛发展，人们每天都需要处理大量的票据、账簿和快递单号等。在这些文字载体上，无一例外的都存在一个共同特征—具有大量的阿拉伯数字，并且这些阿拉伯数字均代表着一定的有效信息。因此如何利用手写数字识别技术快速、准确的提取载体中数字具有重大的意义与应用前景。同时在利用该技术进行数字识别时需要解决由于书写不规范带来的数字识别难题，这就需要在数字识别过程进行一定的预处理并且在提取特征的过程中能够实现提取不同书写形式的统一特征点。总而言之，数字手写识别技术在经济、商业服务领域中具有广阔的应用价值与良好的应用前景。

1.2 国内外研究现状

模式识别的概念于 1950 年首次被提出，并在 20 世纪 70 年代和 80 年代成为信息科学和人工智能的重要组成部分。主要用于图像分析处理，语音识别，声音分类，通信，计算机辅助诊断，数据挖掘等。其目的是通过一定的技术手段实现描述和解释某一现象，在后续实现该现象类型的识别。由于模式识别的研究内容是基于机器的事物学习及识别判断能力，故其引起了国内外研究学者的广泛关注。

手写字符识别技术，作为模式识别领域的一个传统研究分支，它的宗旨是利用生活中常见的计算机等设备智能识别纸张上数字，减轻人们的工作负担。目前，对于手写字符识别技术的研究层出不穷，目前阶段该技术根据输入设备的不同主要分为两大类：离线手写字符识别技术和在线手写字符识别技术。

离线字符识别，也称为光学字符识别（OCR），可根据识别字体分为打印识别和手写识别。国外对字符识别技术的研究较早开始，美国科学家亨德利于 20 世纪 30 年代左右提出了利用光学技术识别人物的想法。到 20 世纪 50 年代，由于印刷产业的发展，字符识别技术开始受到广大研究学者的广泛关注，成为模式识别领域的一个热点。随着研究学者们的深入研究，基于 OCR 原理的简单字符识别系统初现端倪，例如六七十年代来自日本的 Toshiba 公司研究开发了一种基于 OCR 原理的邮政编码识别系统，该系统不仅能实现邮政编码数字的识别，还能自动进行分拣工作。国内开始研究该技术的时间相对来说比较迟，但是国内研究人员的深入研究使 OCR 在中国的发展得到了很大的发展，研究过程经历了简单单一字符的识别到复杂混合字体的识别历程。而且，现有的系统都可以支持简体中文和繁体中文字符的识别，解决了识别多字体混合字符字体的问题，汉字识别率可以达到 98% 以上。此外，中国目前的 OCR 识别技术的商业应用还相当成熟。

在线手写字符识别是字符识别的另一个主要分支，该技术的初次出现极大的激发了国内外学者对

于联机手写字符识别的热情。1966 年，来自麻省理工学院的 J. Liu 发表了一篇关于在线中文手写识别的论文，该论文引发了一波在线手写字符识别研究。随着计算机技术的不断发展和相关理论的不完善，商业实用化逐步成为该技术的一种大力发展方向。1981 年，IBM 发明了一种可以有效识别汉字字符的在线识别系统。该系统基于中文笔画的识别和确定特征，可以达到 79.9% 的识别率。1988 年，中国研究学者刘迎建提出了一种基于笔段的在线手写汉字识别技术。该识别方法能够达到 95% 以上的识别率。目前，在线手写字符识别技术已经日臻成熟，各种不同的商业化软件如同雨后春笋层出不穷。国内知名软件开发商包括汉王公司，清华紫光文通公司和摩托罗拉公司。这一切都表明了一件事情：中国的手写字符识别技术正在蓬勃发展，手写识别系统已逐步商业化，在实用市场占有一席之地。

1.3 论文结构简介

在前人研究的基础上，提出了一种基于 BP 神经网络的手写数字识别方案，旨在实现在线手写数字识别技术。并利用 MATLAB 软件实现了该方案的代码，建立了相应的 GUI 界面，并针对基础神经网络给出了遗传算法的优化方案。经验证，所设计的识别系统具有准确、高效、快速的特点。本文的主要工作内容和主要章节如下所示：

第一章介绍了本文研究的目的及背景，阐述研究意义。接着，阐述国内外研究现状，介绍全文框架；

第二章主要介绍手写数字识别方法。首先，介绍了各种识别方法的原理。其次介绍图像预处理与特征提取，这两个步骤是识别方法中的前提与关键；

第三章首先分析了手写数字识别中的关键问题。然后，阐述了 BP 神经网络的原理和优点，提出了一种基于 BP 神经网络的手写数字识别方案。

在第三章的基础上，第四章采用 MATLAB 软件编写相应程序步骤的程序，实现了基于 BP 神经网络的手写数字识别技术。并进一步制作了相应的 GUI 界面，实现了良好的人机交互。最后结合实验不足，给出可行的改进方案。

在第四章的基础上，第五章针对 BP 神经网络普遍存在的不足之处，采用 GA 遗传算法优化 BP 神经网络的初始权值和阈值，并调整遗传算法的重要参数，对实验结果进行分析，提出改进方案。

第六章对全文的工作内容进行了总结，并对未来技术的发展给出了展望。

第二章 手写数字识别方法及原理

2.1 手写数字识别方法及难点

手写数字识别作为现代社会中备受关注的问题，其研究始于 1929 年至今已经有 90 年的历史。虽然手写数字识别的理论方法是无穷无尽的，但他们的研究结果仍然存在一定的局限性。该局限性是由于理论研究和运算等多方面的影响因素共同作用的结果。首先是理论研究方法，受计算机视觉研究机制的限制，手写数字识别理论仍有进一步发展的空间和潜力。其次实在技术支持层面，目前现有的手写数字识别方式均是以计算机作为运算载体。然而计算机虽然具有强大的运算能力与存储能力，但是其识别能力相较于人类大脑具有很大差距。例如，由于个人书写习惯导致的数字差异会对计算机的识别结果产生较大的影响，而对人脑来说其识别能力不会受到显著影响。综上所述，手写数字识别技术的理论基础具有一定的一般类型。如下面的流程图所示，手写数字识别技术的原理分为四个主要步

骤：图像采集，图像预处理，数字特征提取和最终识别匹配。

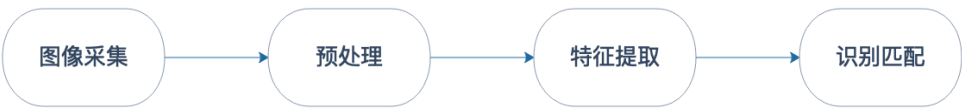


图 2-1 手写数字识别原理图

图片的采集是指利用手机或者数字拍摄设备将纸张或者其余载体上的数字拍摄下来，并利用其余软件读取图片信息。预处理的存在则是为了提取图片中有数字区域的关键区域，避免背景、污渍或者其余噪声信息对后续特征提取的影响。通常，图像的预处理操作主要包括图像的灰度化，图像的二值化，边缘检测和图像的腐蚀扩展。特征提取是整个系统中最关键的步骤之一。特征提取的质量将直接影响后期的数字识别率。它的含义是通过一系列特殊手段，例如测量或者其余处理手段得到数字信息的特征点，方便后期进行相应的数字匹配。识别则是根据测试图片的特征与所建立的特征提取库中的信息进行对比识别，找到最相似的数字，并将其输出。

手写数字识别虽然只是手写字符识别中的一大重要分支，虽然字符结构单一，但是其应用价值不容小觑。在前述的应用背景分析中，已经给出数字识别应用的领域及其应用价值。因此，对于它的研究值得研究学者们进行深入探究。综合来讲，无论数字的范围有多广，其组成部分万变不离其宗，均是由 0-9 这十个数字构成。它们的笔画简易，但是其识别仍然存在一定的难点。虽然目前对于手写数字识别的算法研究层出不穷，但是，在实际应用过程中，仍然存在较大的误码率。这些误识率的存在均是由于个人的书写差异所导致的。由于阿拉伯数字结构简单，在不同文化背景、职业领域以及书写习惯的影响下，人们的手写体数字具有变形多、规则复杂的特点。另外，阿拉伯数字中有些数字之间的区别度不够明显，因此不同的书写特征给手写数字识别引入更大的困难。除此之外，笔画的书写顺序以及练笔数字也对手写数字识别带来了负面影响。为了节省时间，很多人在书写数字的时候都不会按部就班一笔一划的进行书写，而且由于个人习惯的不同，其书写笔画也存在一定的差距。连笔字的书写可能引入冗余字段而缺少数字之间的连贯笔画，其数字的畸变也不可避免，这就为数字的结构识别带来极大的难度与挑战。总而言之，不论是基于哪种基元的结构识别亦或者是笔画识别，上述手写数字的种种因素是造成目前手写数字识别率产生不理想现象的主要原因，因此需要进一步发展手写数字识别方法，提高手写数字的识别率。

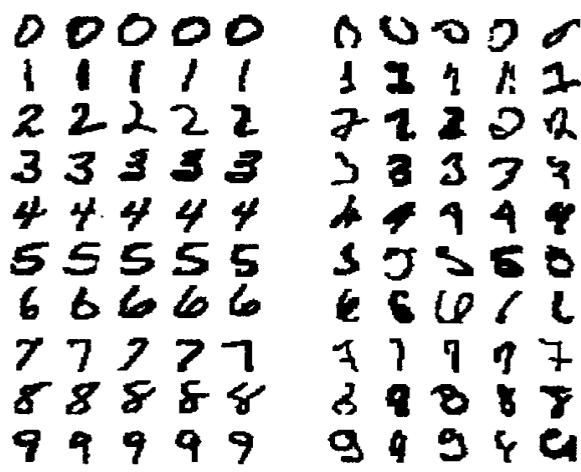


图 2-2 不同类型的手写体数字

2.2 手写数字图像的预处理

预处理作为手写数字识别中的关键一步，其作用是为了去除原始拍摄图片中所包含的噪声或者特征提取中冗余的无效信息，得到计算机可以进行计算处理的二进制矩阵。为后续的特征提取创造良好的条件，提高后续特征提取的准确度，从而达到整体系统识别率提升的目的。这就要求在预处理过程中在尽可能消除无效信息的条件下，最大化保留图片中的数字特点与特征。

在手写数字图片预处理的过程中，主要包括图片的平滑去噪、二值化以及归一化等关键步骤。下图给出图片预处理的基本流程。



图 2-3 图片预处理的基本步骤

平滑去噪的目的是为了消除由于光照、书写形式等外界条件对原始数字图片的不利影响。在图片进行平滑去噪的过程中，一般采用空域滤波器来实现该目的。空域滤波器根据内置作用机制的不同，主要分为线性滤波器和非线性滤波器两大类。线性滤波器通常是对图片进行频域变化的分析，基于傅里叶变换来实现对图片的滤波降噪，常见的线性滤波器有均值滤波器等；而非线性滤波器则是对空间邻域进行处理，例如高斯滤波器、中值滤波器等。下面给出了具有噪声情况下的高斯滤波后的图片，经过对比可以发现，滤波器的存在可以有效消除噪声对图片质量的影响，有利于突出图片本身所具有的特征。

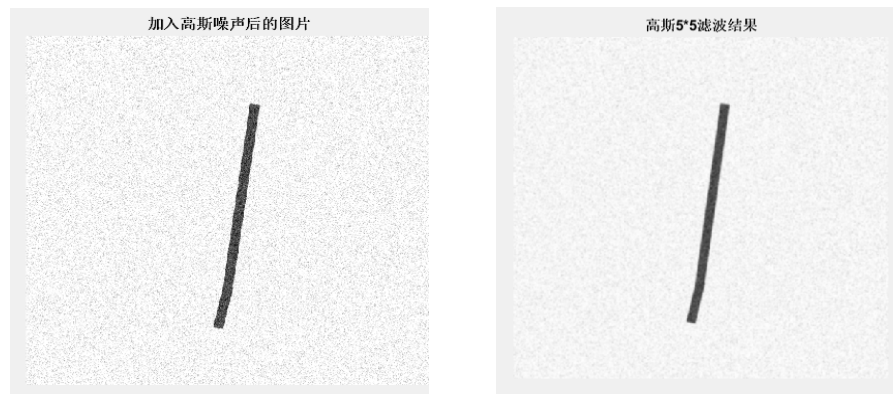


图 2- 4 图片平滑滤波结果对比图

图片的二值化则是根据一定的标准将原始图片的彩色矩阵变化为只有黑色两种颜色矩阵。数字图像二值化中关键的一点就是在于阈值的选择。二值化算法主要包括整体阈值二值化方法，局部阈值二值化方法和动态阈值二值化方法。下面给出了本设计中的二值化结果图。

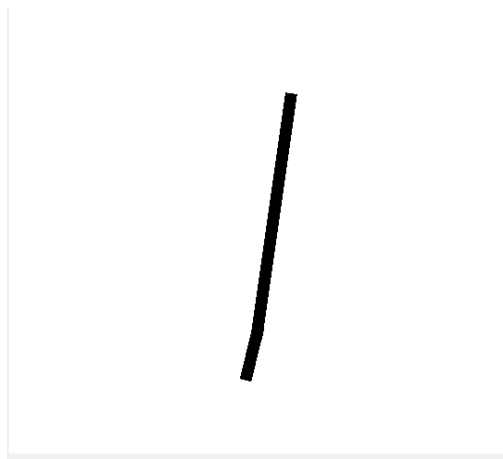


图 2- 5 图片二值化结果图

图片的归一化则是为了方便解决由于书写数字大小所带来的负面影响，将其归一化为同一个模板大小，模板的高度和宽度具有一致性，方便后期所需要进行的特征提取和数字识别步骤。在 MATLAB 软件中，可以简单的调用 `imresize` 函数实现图片大小的调整，使所有图片具有相同的尺寸大小。

2.3 手写数字的特征提取

手写数字的特征提取是设计系统中的关键步骤，特征提取的准确性对识别率有很大影响。特征提取，顾名思义，其目的就是为了找到某种变换，将巨大的数字图片的矩阵转化为包含大部分模式识别信息的小的特征空间，这样不仅能降低识别的复杂度，同时能够提升运算的速度。而且，通过给定训练数据的特征的准确分类，可以得到可靠的分类规则，有利于后期快速准确的实现不同手写数字的识别。

在手写数字识别研究中，常用的特征提取手段分为两大类：结构特征和统计特征。这两种特征提取方法各有千秋：结构特征可以准确的描述数字的结构，对于不同的形式的统一数字，其结构保持不变，具有一致性，可以有效避免草体或是连笔字对特征的影响；而统计特征的优点则是能够有效提出背景噪声带来的特征提取误差，这是由于其大量数据的统计特性所决定的。

本文采用的是基于结构特征提取的模式识别手写数字识别方法，当完成数字的结构特征步骤后，为了实现良好的识别能力，因此建立了一套数字标准形态模板。考虑到数字的变形较多，因此，在本文中构成了具有以下五套模板的匹配库。

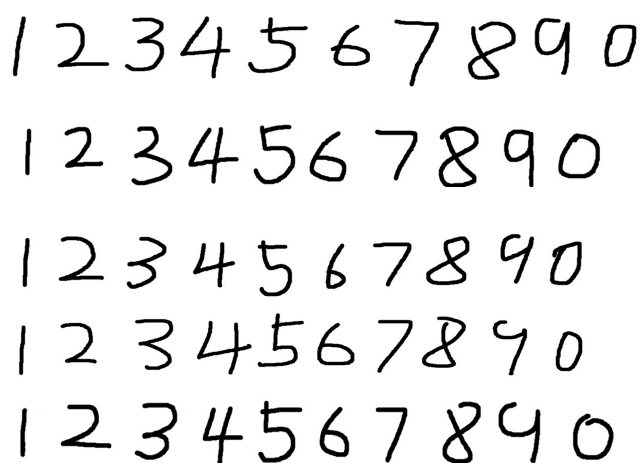


图 2-6 本文构造的数字模板

第三章 基于 BP 神经网络的手写数字识别设计方案

3.1 问题分析

手写数字识别作为一种模式识别方法，具有很强的应用前景和当代社会的广阔市场。然而，在神经网络识别过程中，最重要并且最具有挑战性的步骤就是特征矩阵的提取。在前述的分析中已经提到，由于书写习惯及连笔字的存在，使得数字特征提取具有极大的难度，并且这些问题的存在会使得提取的特征矩阵具有一定的误差。另外，在特征匹配过程中，由于特征矩阵的匹配是非线性映射，如何采用合适的手写数字匹配识别方法是手写数字识别系统中需要解决的另一个难点。因此，在本设计的过程中，需要着重解决上述存在的两大问题，在综合考虑分析后，本文采用基于结构特征提取的 BP 神经网络手写数字识别方案。

结构特征提取，顾名思义，是对数字的结构特征进行提取。结构特征可以准确的描述数字的结构，对于不同的形式的统一数字，其结构保持不变，具有一致性，可以有效避免草体或是连笔字对特征的影响，能够有效解决前述的特征点匹配的关键问题。此外，在前一章中已经提及，在本文的设计过程中，建立了基于结构特征提取的匹配库，方便后期的数字特征匹配。

由于特征匹配过程是一个典型的非线性映射，故在本设计的过程中，考虑采用 BP 神经网络的方法，进行相应的数字识别。BP 神经网络作为最广泛使用的神经网络方法之一，其强大的学习能力以及快速高效的计算能力得到广大研究学者的肯定。

因此，本文的执行机构采用 BP 神经网络训练识别，通过该网络实现手写数字的识别目的。

3.2 人工神经网络简介

人工神经网络是指以人脑神经网络为模拟目标，通过相关理论抽象简化的信息处理系统。类似于人脑神经网络结构，人工神经网络也是一个具有强自适应特性的非线性动态系统，由大量相互连接的神经元结构组成。人工神经网络根据神经元之间的互联方式的不同，也有不同的类型，如 BP 人工神经网络，RBF 人工神经网络，CNN 深度学习神经网络。

人工神经网络的学习方法也是一种模拟的人脑学习方法。人工神经网络在通过具有某些内在标准的数据训练之后可以具有辨别力。人工神经网络同样具有大量的神经元结构，其功能实现也是依赖于这些基础结构所实现。这些处理单元我们把它称作人工神经元。人工神经网络可以看做是由无数个神经元和相应的权值构成的结构，该结构可以通过一个简单的有向图所描述。、手写数字识别手写数字识别在有向图中，人工神经元是生物神经元的模拟，而定向弧是轴突—突触—树突对的模拟。人工神经元结构如下图所示：

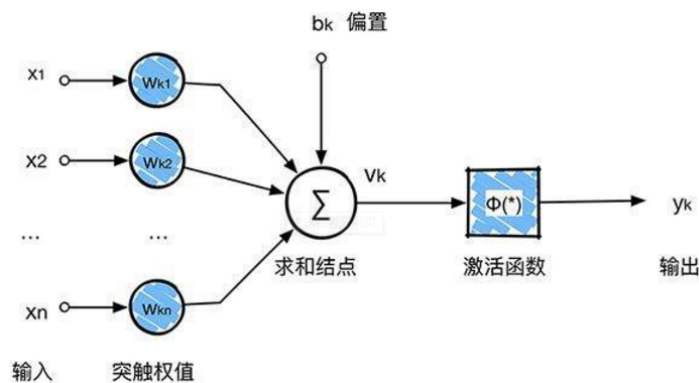


图 3- 1 人工神经元结构图

神经网络从两个方面模拟大脑：

- (1)神经网络获取的知识是从外界环境中学习得来的。
- (2)内部神经元的连接强度，即突触权值，用于储存获取的知识。

神经网络系统包括由大量神经元连接形成的拓扑，其可以处理人脑的不同部分之间的信息传输。根据这些大神经元的数量和它们之间的连接，人脑可以接收输入信息，并且非线性并行映射处理由分布式并行处理神经元执行，从而实现复杂的信息处理和推理任务。

在人工神经网络学习形式、工作过程以及学习准则这三点决定着神经网络的能力。

一般的神经网络中，其学习形式为：在构建神经网络时，首先要明确该网络的激活函数和传递函数。两种函数一旦确定，后期就没法进行更改。在后续步骤中，如果要对输出结果进行修正，只有改变输入和神经网络层数等指标实现。

工作过程为：神经网络的工作过程包括训练学习和判断识别两个过程。在学习过程中，每个神经元执行规则学习，调整权重参数，并执行非线性映射关系拟合以实现训练准确性。在识别阶段，利用已经建立好的学习规则，根据输入进行输出预测。

人工神经网络的学习标准可以根据不同的规则分为联想学习和非联想学习，监督学习和无监督学

习。下面将给出几种典型的学习准则：

误差修正型规则：这是一种有监督的学习方法。根据计算得到的误差对网络进行调整，实现良好的网络建立。

竞争型规则：这是一种无监督的学习过程，仅根据提供的学习样本进行网络的建立，没有预期的输出作为参考数据。

随机型规则：将随机和概率论的思想融入到神经网络中，根据输出的误差进行网络参数调整，实现网络的建立。

3.3 BP 神经网络原理

BP 神经网络作为一种人工神经网络，根据其学习标准其可以视为一种典型的监督学习算法。它的主要工作方式为：首先给定学习数据进行训练，根据反馈的误差通过反向传播算法不断的调整神经元的权值，使得神经网络的输出值与给定的学习样本输出尽可能接近。当输出和期望输出间的差异满足设定的误差值时，BP 神经网络训练结束。此时，通过保存网络的权重和偏差，可以建立起一个完整的神经网络结构。

BP 神经网络由输入层，隐式层，输出层和层之间的节点组成。在 BP 神经网络中，传播过程分为前向传播和反向传播。下面将给出这两种传播模式的详细过程。

1、正向传播过程

✧ 输入层

输入值通常是样本的每个分量的输入值，输出值通常等于输入值。

✧ 隐含层

对于节点 j ，其输入值 net_j 为其前一层各节点输出值 O_i 的加权和：

$$net_j = \sum_i W_{ij} O_i \quad (3.1)$$

$$\text{输出值 } O_i = f_s(net_i) \quad (3.2)$$

$$f_s(net_j) = \frac{1}{1 + e^{-(net_j - \theta_j)/\lambda}} \quad (3.3)$$

式中 $f_s()$ 称为激励函数。一般采用 sigmoid 函数；

式中 θ_j 称为阈值， λ 称为温度系数。

隐藏层可以是一个或多个层，但随着隐藏层的增加，网络的复杂性也会增加。学习速度减慢，网络的推导能力也可能下降，即所谓的“过度拟合”现象。网络隐藏层和节点数的选择是 BP 网络学习算法中要解决的问题。

✧ 输出层

输出层的输入 net_k 与输出 O_k 具有与隐含层结构相似性，分别为：

$$net_k = \sum_i W_{ki} O_i \quad (3.4)$$

$$O_k = f_x(net_k) \quad (3.5)$$

2、反向传播过程

BP 学习算法的误差函数通常定义为实际输出和预期输出的均方误差和。从输入到输出的非线性映射是通过最小化误差函数来不断改变权重所实现的。

让网络输入模式样本为 $x_p = \{x_{pi}\}$ ，预期输出为 $\{t_{pk}\}$ ，均方误差函数 E_p 为：

$$E_p = \frac{1}{2} \sum_k (t_{pk} - O_{pk})^2 \quad (3.6)$$

而对于所有的学习样本，系统的均方误差为：

$$E_p = \frac{1}{2P} \sum_p \sum_k (t_{pk} - O_{pk})^2 \quad (3.7)$$

为了建立良好的学习规则，通常要对权值进行修正。通常，经典的 BP 学习算法使用最快的梯度下降法来调整权重，其调整量的公式如下所示：

$$W_{kj} = -\eta \frac{\partial E}{\partial W_{kj}} \quad (3.8)$$

式中 $0 < \eta < 1$ 称为学习率。

从(3.8)式可以导出下列连接权重的校正公式

$$W_{kj} = \eta \delta_k O_j \quad (3.9)$$

对于输出节点：

$$\delta_k = (t_k - O_k) O_k (1 - O_k) \quad (3.10)$$

对于隐节点：

$$\delta_j = O_j(1 - O_j) \sum_k \delta_k W_{kj} \quad (3.11)$$

现阶段，BP 神经网络无论是在理论方面还是建模方面都非常成熟，具有强大的学习能力和判断能力。在构建网络过程中，中间层的数量和每层中的神经元的数量可以根据具体条件任意设置，并且性能根据结构而不同。因此，本文在分析手写数字识别问题的基础上，选择 BP 神经网络作为手写数字识别的基本方法。

3.4 系统设计方案及流程图

如前一章分析所述，本文重点讨论了手写数字识别的难点。在整体的系统中，其具体过程为：首先，收集必要的各种手写数字图片作为训练样本；在获取手写数字图片之后，对图像进行预处理。图片预处理的作用为消除背景噪声或者环境或识别数字目标的影响，提高后期特征提取的准确度。在这种设计中，主要的预处理方法是对图像进行平滑去噪，灰度化，二值化和图像归一化。在图像预处理之后，下一步是提取图像的特征。本文中所采用的特征提取方案为基于结构特征的数字特征提取，该方法的优点是能够避免数字书写差异对识别结果的影响。如图 2-6 所示，建立了五种数字模板库。最后一部分是基于 BP 神经网络的数字识别。其基本流程图如下所示：

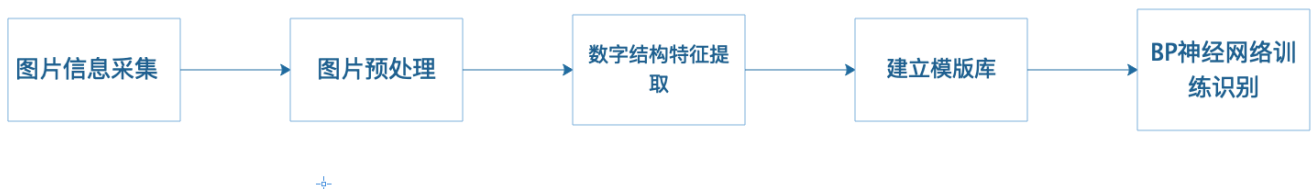


图 3-2 基本流程图

在系统流程确定的情况下，需要考虑方案实现的具体软件平台，经过多方调研后，本次设计采用 MATLAB 软件作为软件平台，MATLAB 作为一款矩阵处理软件，具有强大的图片处理功能及神经网络设计功能。其内置的函数为操作者提供了极为方便的程序设计参考，避免了重复编写函数的时间耗费，在下一章中，将详细介绍相应的方案设计 MATLAB 代码。

第四章 系统的实现与结果分析

4.1 MATLAB 简介

在前面章节的分析中已经说明，本次设计的所有代码均基于 MATLAB 平台进行编写。使用 MATLAB 软件编写这个手写数字识别程序的主要优点如下：(1) MATLAB 软件具有成熟的工具箱和库函数。同时这也避免了重复编程的缺点，为用户节省了时间，解决了用户程序优化的问题；(2) 强大的图形处理功能。作为一种数学计算软件，MATLAB 的图形和图像处理功能不容小觑。MATLAB 处理图形图像时一般是基于矩阵的处理，而且 MATLAB 中具有完善成熟的各种图像变换内置函数，能够及时有效的完成对图形图像的处理。此外，其强大的可视化功能可以及时显示处理后的图像。该系列功能将为实现手写识别系统奠定坚实的基础。(3) 简单的人际交互界面设计功能。MATLAB 不仅是一个编程软件其内置 GUI 功能也为编程用户提供了人机交互环境设计功能。GUI 界面设计简单，易于塑造。

综上所述，本文针对手写数字识别系统设计问题，结合 MATLAB 软件的优势，拟采用以下方案进行程序编写：

1. 利用 MATLAB 内置图像处理函数，完成图片的读入以及平滑去噪、二值化等预处理功能，方便后期图片的特征提取以及后期的识别；
2. 基于字体结构的特征提取，实现标准模板库的建立；
3. 利用 MATLAB 的 BP 神经网络完善的内置函数（newff 等函数）实现 BP 神经网络的参数设置、构建，完成相应的识别功能，给出识别结果；
4. 基于 MATLAB 的 GUI 界面设计功能，设计手写数字识别系统的 GUI 界面，完成人机交互环境功能的实现。

4.2 MATLAB 程序设计

在前面章节以及本章的第一节的分析中，已经提出手写数字识别系统的设计方案，绘制出相应的流程图，并选用 MATLAB 软件作为功能实现平台。因此，在本节中，针对上述流程方案提出基于 MATLAB 对的分区功能实现方案，并编写相应分区功能的程序。

首先需要解决的是图片的读入及显示问题，在 MATLAB 中 `imread` 函数和 `imshow` 函数为解决该问题提供了一个简洁明快的方式。其调用格式为 `A=imread (filename)`，其作用为跟据文件名 `filename` 读取灰图像，返回的数组 `A` 包含图像数据的矩阵信息。另外图像的显示可以通过 `imshow (A)` 来实现。因此，在设计中的子程序 `rec.m` 采用 `f=imread(path);` 的语句进行读入图片。

在读入图片以后是要进行图片的预处理阶段，由于本次选择的图片已经利用高斯平滑滤波器进行滤波，其中 `fspecial` 函数用以产生高斯平滑滤波器，`imfilter` 函数执行过滤操作，因此它不包含在编程中。具体的高斯滤波器效果如下图所示：

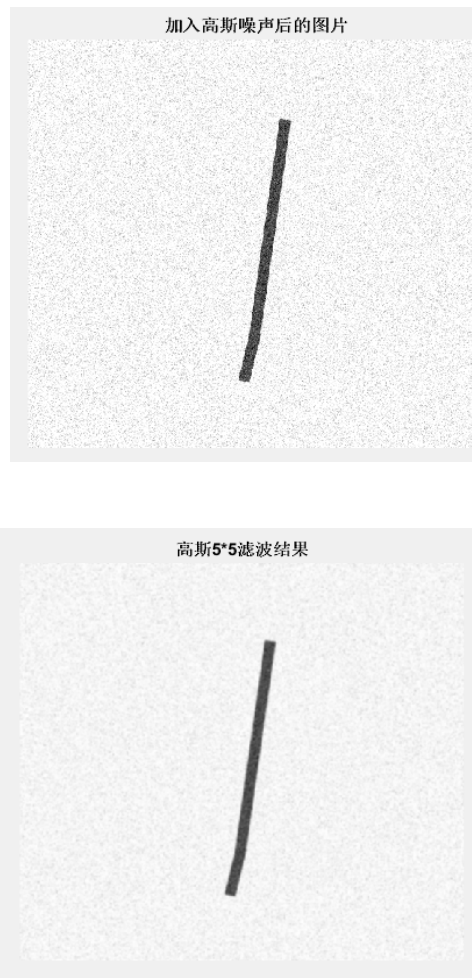


图 4-1 图片滤波结果

除了图像预处理过程之外，还存在图像的二值化操作。因此在程序设计过程中引入了判断机制，进行图片的二值化处理。具体处理程序如下图所示：

```
if size(img,3)>1
    img=rgb2gray(img);
end
thresh = graythresh(img);
img = 1 - im2bw(img,thresh);
```

图 4-2 图片的二值化程序

当完成图像的预处理时，需要进行的下一步就是提取各个数字所独有的特征并构造相应的模板库。特征提取的作用是为了提取手写数字的简化结构特征，方便后期的手写数字识别，提升运算速度。由于程序设计的原因，图片特征提取过程并不是在同一个.m 文件中完成的，因此仅给出本文中的特征提取部分程序，在附录中将给出本设计的所有程序。部分特征提取程序如下图所示，其中 P 矩阵就是所提取的数字特征矩阵。

```

1 function PP=handwritten(f)
2 global cnt1;
3 ff=cutting(f);
4 out=zeros([35 10]);
5 for i=1:cnt1
6     for j=1:7
7         for k=1:5
8             ft((j-1)*5+k)=sum(sum(ff((j*10-9:j*10),(k*10-9:k*10),i)));
9         end
10    end
11    ft=ft'/100;
12    out(:,i)=ft;
13 end
14 PP=out(:,1:10);

```

图 4- 3 图片特征提取程序

在特征提取结束后，后面是进行神经网络的训练以及识别过程。在 MATLAB 中内置有相应的 BP 神经网络训练函数。本次构建的前向 BP 神经网络是利用 newff 函数实现的，另外在 MATLAB 中可以对神经网络的参数进行设置，具体设置函数及含义分别为：

```

net.trainParam.show= ； %显示训练结果的间隔步数
net.trainParam.epochs= ； %最大训练步数
net.trainParam.goal= ； %训练目标误差
net.trainParam.mu = ； %学习系数的初始值，Marquardt 调整参数
net.trainParam.mu_dec= ； %学习系数的下降因子
net.trainParam.mu_inc= ； %学习系数的上升因子
net.trainParam.mu_max= ； %学习系数的最大值
net.trainParam.min_grad=； %训练中最小允许梯度值

```

在本次设计中所利用的训练及识别程序如下所示：

```

function net = edu_train(inputn,outputn,tt)
if tt==0
    net=newrb(inputn,outputn,0,4,7,1);
else
    S1=7;
    S2=1;
    net=newff(minmax(P),[S1 S2],{'transig' 'purelin'},'traingd');
    net.LW{2,1}=net.LW{2,1}*0.01;
    net.b{2}=net.b{2}*0.01;
    net.performFcn='sse';
    %网络进化参数
    net.trainParam.goal=0.05;
    net.trainParam.show=20;
    net.trainParam.epochs=20;
    net.trainParam.mc=0.85;
    net.trainParam.lr=0.0001;
    %Training
    [net,tr]=train(net,inputn,outputn);
    % %an = sim (net,inputn);
    % %error = sum(abs(an-outputn));
end

```

图 4- 4 BP 神经网络训练程序

```

112 ff=getframe(handles.axes1);
113 PP=handwritten(ff.cdata);
114 b=sim(net,PP);
115 num=[];
116 for i=1:cnt1
117     temp=0;
118     for j=1:10
119         if b(j,i)>temp
120             num(i)=j;
121             if num(i)==10
122                 num(i)=0;
123             end
124             temp=b(j,i);
125         end
126     end
127 end
128 num_fin=0;|

```

图 4- 5 BP 神经网络识别程序

到目前为止，手写数字识别系统的相应模块的 MATLAB 程序已经完成。为了方便操作及显示，在下一节中建立了相应的 GUI 界面，并对结果进行测试。

4.3 MATLAB GUI 界面的设计

MATLAB 为广大编程者提供了一个方便简洁的人机交互环境设计界面。GUIDE 是一个界面设计工具集，提供界面，属性和回调函数的外观。当设计完成后，MATLAB 不仅会有一个界面信息的 fig 文件，也会生成对应的 m 文件，用于回调函数的编写，帮助其完成功能。

在本节中，将建立手写数字识别系统的 GUI 界面，并完成程序识别功能的测试。

首先，GUI 界面的建立过程需要进行相应的框架绘制。利用 GUI 界面工具栏内的各个控件，建立如下图所示的 GUI 框架界面。其中 Start 按钮、Recognize 按钮以及 clear 按钮设置为只有当 BP 神经网络训练结束后才可用的状态。Start 按钮的功能是允许用户开始进行操作，axes 控件的存在则是为了方便用户进行数字的手写输入，Recognize 按钮则是实现相应的数字识别功能，clear 按钮则是实现清屏功能，为下一次手写数字的输入提供空白的 axes 界面。

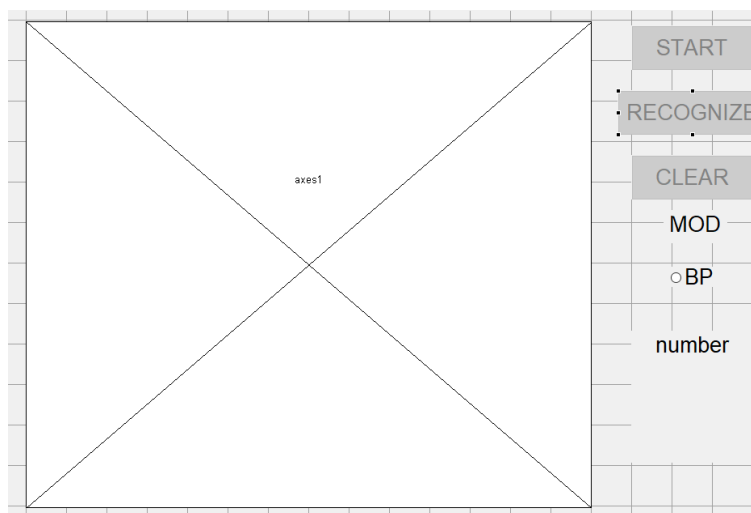


图 4- 6 GUI 界面布局图

接下来，便是对各个控件进行相应的回调函数编写。本次设计中，为了方便程序设计，除 fig 文件对应的 m 文件外，还编写了几个子程序。下面将给出总体函数图以及各个控件的回调函数。

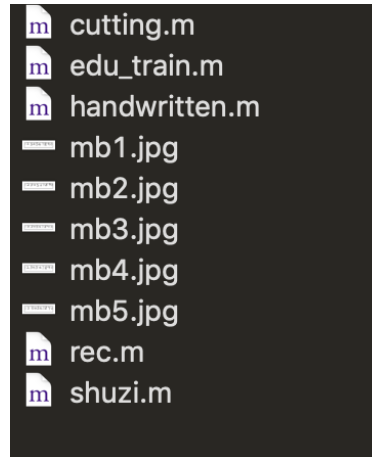


图 4- 7 手写数字识别系统程序列表

Start 按钮：

```
global start;
start=1;
set(handles.pushbutton2,'Enable','on');
set(handles.pushbutton3,'Enable','on');
```

Recognize 按钮：

```
global tt net cnt1;
ff=getframe(handles.axes1);
PP=handwritten(ff.cdata);
b=sim(net,PP);
num=[];
for i=1:cnt1
    temp=0;
    for j=1:10
        if b(j,i)>temp
            num(i)=j;
        if num(i)==10
            num(i)=0;
        end
        temp=b(j,i);
    end
end
```

```

end

end

end
num_fin=0;
for i=1:cnt1
    num_fin=num(i)*10^(cnt1-i)+num_fin;
end
set(handles.text1,'String',num_fin);
guidata(hObject, handles);
clear 按钮:
cla(handles.axes1);
set(handles.text1,'string','')
guidata(hObject, handles);
BP 控件:
global tt net;
set(handles.radioButton1,'value',1);
% set(handles.radioButton2,'value',0);
if(get(handles.radioButton1,'value'))
    tt=1;
    set(handles.pushbutton1,'Enable','on')
    net=rec(tt);
end

```

编程完成后，需要对程序进行测试。首先是神经网络的训练过程。如下图所示，所设计的 BP 神经网络在达到设置的最大步长 500 之前停止了训练过程，其误差如图所示。

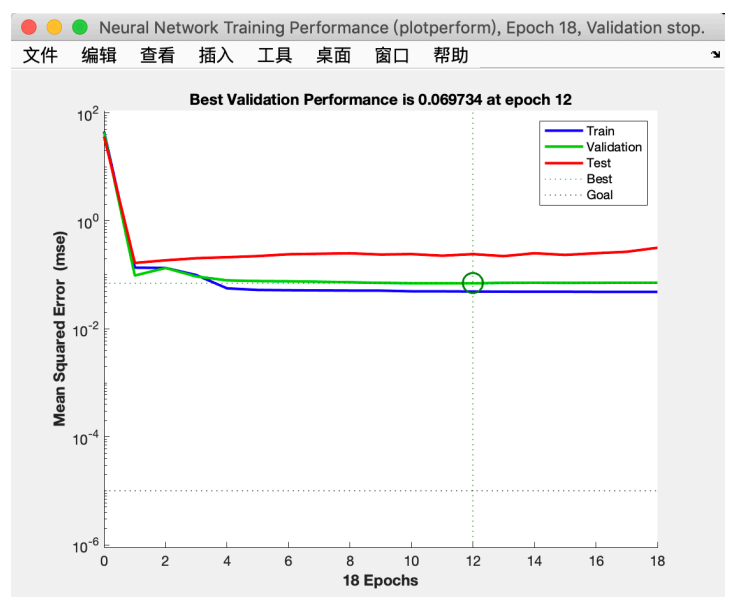
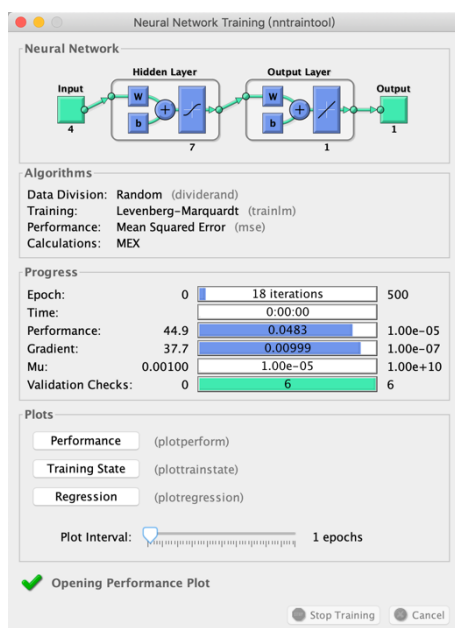
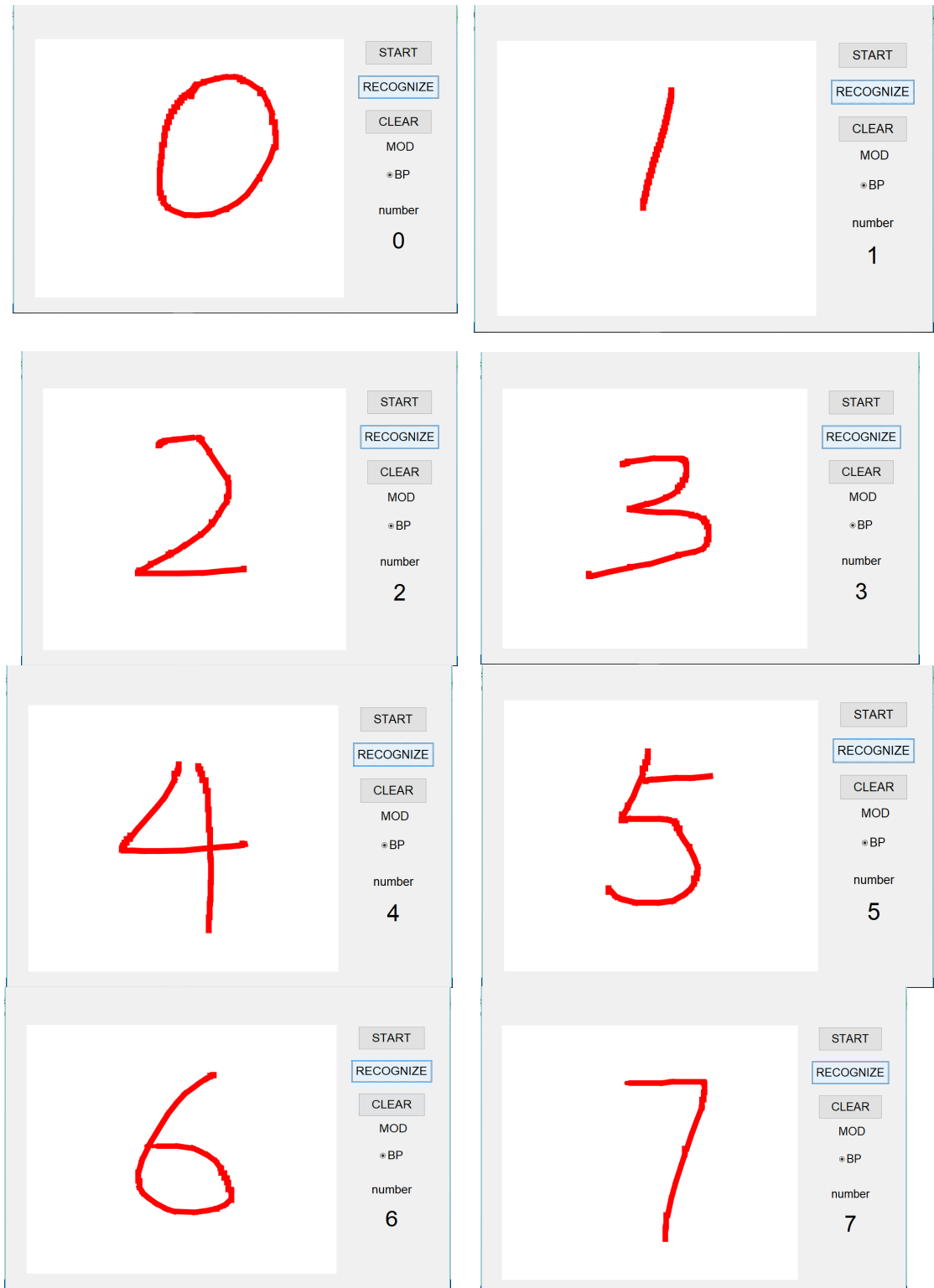


图 4- 8 神经网络训练结果图

接下来是对其识别能力的测试。神经网络训练结束后，按下 Start 按钮，在左侧的坐标轴中用鼠标分别输入 0-9 的 10 个阿拉伯数据，并单击 Recognize 按钮进行识别。得到具体结果如下图所示：



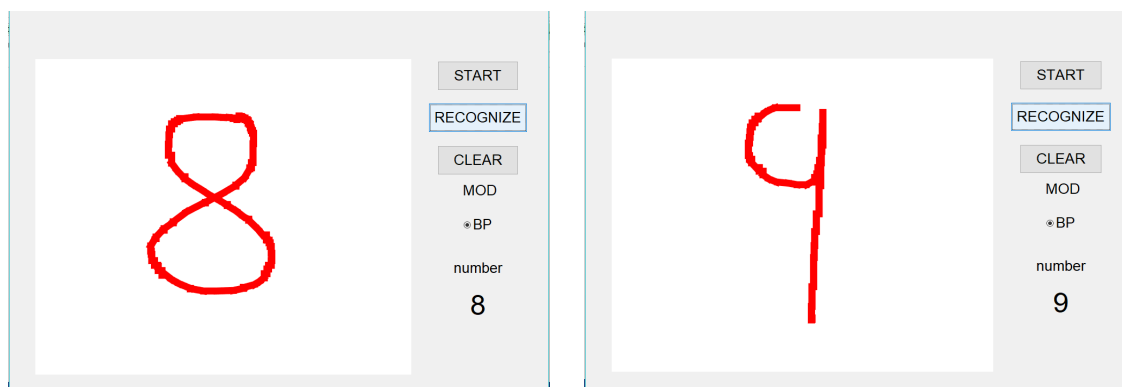


图 4- 9 识别结果

从识别结果图来看，所设计手写数字识别系统能够准确识别各个数字的结果，同时书写方式或者瑕疵的存在不会影响最后识别结果的准确率。例如图中 9 的识别图，虽然手写中数字 9 中存在笔画不连贯的空白，但是识别结果仍然准确无误，这也符合之前所说明的结构特征提取方法。

综上所述，本文所设计的手写数字识别系统能够高效、准确的识别 0-9 这 10 个阿拉伯数字，达到了手写数字识别的目标。

4.4 结果分析与改进

由网络训练结果图可以看出，本实验手写数字 shiBP 网络的泛化评估值(validation check)较小，说明 BP 网络的泛化能力一般，仍然具有很大的改进空间。

影响泛化能力的因素中，主要有训练样本的质量和数量，网络结构以及问题本身的复杂程度。针对 BP 神经网络，我根据本次实验设计存在的不足之处，总结出以下提高泛化能力的方法：

- 1) 对现有样本的一些处理,如在样本中加入噪声训练,对样本进行聚类,或选择抽样进行主动 (Active)学习,来寻求泛化能力的提高；
- 2) 正则化方法(这是一种在结构确定后提高泛化能力的方法,即在目标函数项中增加一项惩罚项或正则项)。
- 3) 第 2 种和第 3 种方法的结合:先进行结构优化,待网络结构确定后再采用正则化方法。
- 4) 第 1 种与其它方法的结合。
- 5) 经过查阅相关资料，改善泛化能力的方法还有双 BP 网络，Marquardt 算法、两阶学习算法，权值平滑法以及分布瓶颈法进行训练等

第五章 基于 Matlab 的遗传算法优化 BP 神经网络

本实验中，虽然 BP 神经网络在手写数字识别过程中表现良好，但不可忽视的是，BP 神经网络的权值和阈值都是随机初始化的，这样容易导致拟合效果不稳定，况且由于使用梯度下降的方式训练 BP 网络的权值和阈值，容易导致训练陷入局部最优中。所以本实验采用 GA 遗传算法，对基于 BP 神经网络的手写数字识别方法中，神经网络的初始权值和阈值进行优化。

5.1 模型建立

遗传算法优化 BP 神经网络算法流程如下流程图所示，分为 BP 神经网络结构确定，遗传算法优化，和 BP 神经网络预测 3 个部分。

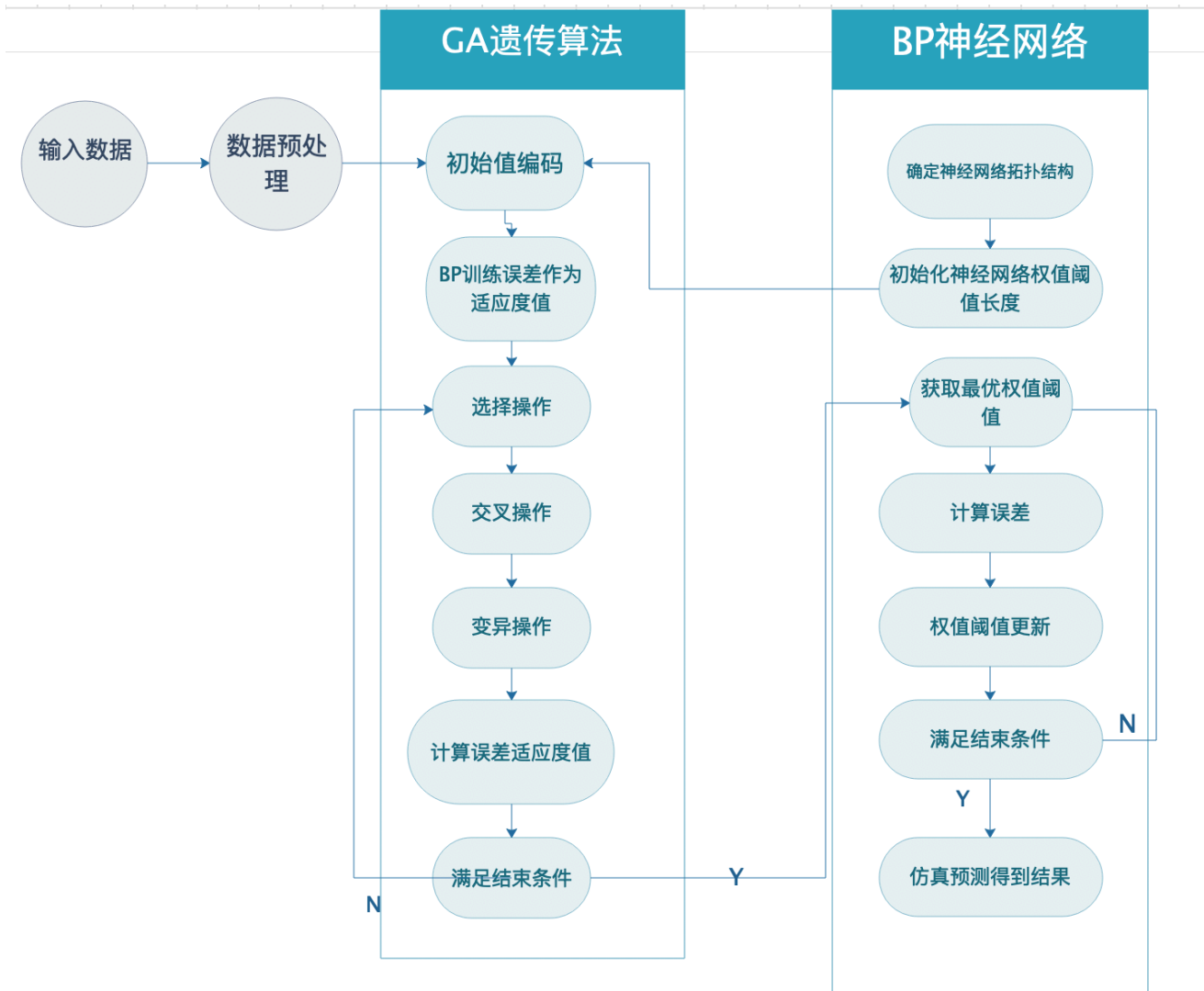


图 5-1 模型流程图

✧ BP 神经网络确定部分

根据输入输出参数个数确定 BP 神经网络结构，进而确定遗传算法个体长度。

✧ 遗传算法优化部分

使用遗传算法优化 BP 神经网络的权值和阈值，种群中的每个个体都包含了一个网络所有权值和阈值，个体通过适应度函数计算个体适应度值，遗传算法通过选择，交叉和变异操作找到最优适应度值对应的个体。

✧ BP 神经网络预测部分

用遗传算法得到最优个体对网络初始权值和阈值赋值，网络经训练后预测函数输出。

本次实验，设置有 4 个输入参数，1 个输出参数，所以设置的 BP 神经网络结构为 4-7-1，即输入层

有 4 个节点，中间层有 7 个节点，输出层有 1 个节点，共有 35 个权值，8 个阈值，所以遗传算法个体编码长度为 $35 + 8 = 43$ 。从 200 组输入输出数据中，随机选择 190 组作为训练数据。把训练数据预测误差绝对值和作为个体适应度，个体适应度值越小，该个体越优。

5.2 基本思想

用个体代表BP网络的初始权值和阈值，网络初始化后得到的预测误差作为个体的适应度值，GA算法通过模拟自然选择过程中的选择，交叉，变异操作寻找最优个体，即最优的BP神经网络初始权值。

GA算法思想中，种群，个体，染色体，基因概念非常具体，因此我着重对这四个概念进行了理解，并给出了如下个人阐述。

1. **种群**是全部个体的总和，信息量上而言，是包含全体成员的信息。因此种群通过逻辑联想可以定义为一个结构体，该结构体中包含个体信息。
2. **个体**可以理解为具体问题的一个解决方案，由于一个问题可以对应多种解决方案，因此应该有多个不同个体。
3. 多个**染色体**组成一个个体，因此，染色体对应到BP神经网络上，由4部分构成：输入层到隐含层之间的连接权值（一个二维矩阵），隐含层到输出层之间的连接权值（一个二维矩阵），隐含层神经元阈值（一维矩阵），输出层神经元阈值（一维矩阵）
4. **基因**即神经元与神经元之间的连接权值（一个数值）。

5.2.1 求解全局最优解

GA算法常用适应性函数（*fitness function*）来衡量这个解决方案的优劣，因此，一个基因组到其解的适应度形成一个映射，亦可以视为一个多元函数。获取最佳适应度评价就可以转化为求解最优解过程。

5.2.2 算法实现

遗传算法优化BP神经网络要素包括种群初始化，适应度函数，选择操作，交叉操作和变异操作。

1) 种群初始化

个体编码方法为实数编码，每个个体均为一个实数串，由输入层与中间层连接权值，中间层阈值，中间层与输出层连接权值，以及输出层阈值4部分组成。

个体包含神经网络全部权值和阈值，在网络结构已知情况下，个体即一个结构，权值，阈值

确定的神经网络。

2) 适应度函数

个体得到BP神经网络的初始权值和阈值，经过BP神经网络训练，将预测输出和期望输出之间的误差绝对值之和E作为个体适应度值F，计算公式为：

$$F = k \left(\sum_{i=1}^n \text{abs}(y_i - o_i) \right) \quad (5.2)$$

y_i : BP神经网络第*i*个结点的期望输出;
 o_i : 第*i*个结点的预测输出;

本次实验中，上式系数k被规定为1

3) 选择操作

基于适应度比例选择策略，使用轮盘赌法，设表示含有个个体的群体，设 $\{a_1, a_2, \dots, a_n\}$ 表示含有*n*个个体的群体，其中每个个体的适应度值分别为 $f(a_i), i=1, 2, \dots, n$ ，则个体 a_i 被选择的概率为

$$p(a_i) = \frac{f(a_i)}{\sum_{j=1}^n f(a_j)} \quad (5.3)$$

$$f\left(a_i = \frac{k}{F_i}\right) \quad (5.4)$$

K为系数，本实验中，令k=1。

4) 交叉操作

由于个体采用实数编码，所以交叉操作方法采用实数交叉法，第k个染色体ak和第m个染色体am在j位的交叉操作方法如下：

$$\begin{cases} a_{kj} = a_{kj}(1-b) + a_{mj}b \\ a_{mj} = a_{mj}(1-b) + a_{kj}b \end{cases} \quad (5.5)$$

式中，b是[0,1]之间的随机数。

5) 变异操作

选取第*i*个个体的第*j*个基因

进行如下变异操作

$$a_{ij} = f \left\{ \begin{array}{l} a_{ij} + (a_{\max} - a_{ij}) * f(g), r \geq 0.5 \\ a_{ij} - (a_{ij} - a_{\min}) * f(g), \text{else} \end{array} \right\}$$
$$f(g) = r' \left(1 - \frac{g}{G_{\max}} \right)^2 \quad (5.6)$$

式中，下标max代表基因组上界；下标min代表基因组下界；g为当前迭代次数；Gmax为最大进化次数；r，r' 为[0,1] 间的随机数。

5.3 Matlab实现

1. 染色体编码

```
1 function ret=Code(lenchrom,bound)
2 %本函数将变量编码成染色体，用于随机初始化一个种群
3 % lenchrom   input : 染色体长度
4 % bound       input : 变量的取值范围
5 % ret         output: 染色体的编码值
6 flag=0;
7 while flag==0
8 pick=rand(1,length(lenchrom));
9 ret=bound(:,1)+(bound(:,2)-bound(:,1)).*pick; %线性插值，编码结果以实数向量存入ret中
10 flag=test(lenchrom,bound,ret); %检验染色体的可行性
11 end
12
```

图 5-8 染色体编码程序

2. 适应度函数

```
1 %该函数用来计算适应度值
2 %x      input      个体
3 %inputnum input      输入层节点数
4 %outputnum input    隐含层节点数
5 %net     input      网络
6 %inputn  input      训练输入数据
7 %outputn input      训练输出数据
8 %error   output     个体适应度值
9 % error函数用来计算适应度值
10 function error = fun(x,inputnum,hiddennum,outputnum,net,inputn,outputn)
11 %网络权值赋值
12 net.iw{1,1}=reshape(w1,hiddennum,inputnum);
13 net.lw{2,1}=reshape(w2,outputnum,hiddennum);
14 net.b{1}=reshape(B1,hiddennum,1);
15 net.b{2}=B2;
16 %网络进化参数
17 if tt==0
18 net=newrb(inputn,outputn,0,4,7,1);
19 else
20 S1=7;
21 S2=1;
22 net=newff(minmax(P),[S1 S2],{'transig' 'purelin'},'traingd');
23 net.LW{2,1}=net.LW{2,1}*0.01;
24 net.b{2}=net.b{2}*0.01;
25 net.performFcn='sse';
26 net.trainParam.goal=0.05;
27 net.trainParam.show=100;
28 net.trainParam.epochs=20;
29 net.trainParam.mc=0.85;
30 net.trainParam.lr=0.0001;
31 %网络训练
32 net=train(net,inputn,outputn);
33 an = sim(net,inputn); %预测
34 error = error=sum(abs(an-outputn));
35 end
36
```

图 5-9 计算适应度函数程序

3. 选择操作

```
function ret=select(individuals,sizepop)
% 本函数对每一代种群中的染色体进行选择, 以进行后面的交叉和变异
% individuals input : 种群信息
% sizepop      input : 种群规模
% ret          output : 经过选择后的种群

%根据个体适应度值进行排序
fitness1=10./individuals.fitness;

sumfitness=sum(fitness1); %个体的适应度倒数之和
sumf=fitness1./sumfitness; %各个个体的概率
index=[]; %空矩阵
for i=1:sizepop %转sizepop次轮盘
pick=rand;%rand为生成一个[0, 1]的随机数, 相当于rand ()
while pick==0 %%生成0时重新随机
pick=rand;
end
for j=1:sizepop
pick=pick-sumf(j); %%随机数与个体概率之差
if pick<0 %随机数比个体的概率小, 则将该个体加入新种群。(概率越大的个体被选中的几率越大)
index=[index j]; %[index j]为将j个体加入新种群
break; %寻找落入的区间, 此次转轮盘选中了染色体i, 注意: 在转sizepop次轮盘的过程中, 有可能会重复选择某些染色体
end
end
end

individuals.chrom=individuals.chrom(index,:);

%individuals.chrom为种群中个体 (index, :) 中index为一维矩阵, 表示里面数字的行数, 种群的个体变为原来种群的某些个体

individuals.fitness=individuals.fitness(index); %选择出的新个体的适应度为新种群适应度

ret=individuals;
```

图 5-10 选择操作程序

4. 交叉操作 (部分关键代码截图)

```
1 function ret=Cross(pcross,lenschrom,chrom,sizepop,bound)
2 %本函数完成交叉操作
3 % pcross      input : 交叉概率
4 % lenschrom   input : 染色体的长度
5 % chrom       input : 染色体群
6 % sizepop     input : 种群规模
7 % ret         output : 交叉后的染色体
8 for i=1:sizepop %每一轮for循环中, 可能会进行一次交叉操作, 染色体是随机选择的, 交叉位置也是随机选择的, %但该轮for循环中是否进行交叉操作
% 则由交叉概率决定 (continue控制)
9 % 随机选择两个染色体进行交叉
10 pick=rand(1,2); %生成1*2的矩阵, 大小为[0, 1]的随机数
11 while prod(pick)==0 %计算矩阵pick的连乘积 (所有数相乘) %%如果矩阵pick中有元素为0, 则重新随机
12 pick=rand(1,2);
13 end
14 index=ceil(pick.*sizepop); %pick翻倍, ceil为取最小的正整数 (比元素大)
15 % 交叉概率决定是否进行交叉
16 pick=rand; %生成一个[0, 1]的随机数
17 while pick==0 %生成一个非0的随机数
18 pick=rand;
19 end
20 if pick>pcross %若生成的随机数大于交叉概率, 则此轮循环不进行交叉
21 continue;
22 end
23 flag=0;
24 while flag==0
25 % 随机选择交叉位
26 pick=rand; %生成一个[0, 1]的随机数
27 while pick==0 %生成一个非0的随机数
28 pick=rand;
29 end
```

图 5-11 交叉操作程序

5. 变异操作(部分关键代码截图)

```

for i=1:sizepop          %每一轮for循环中，可能会进行一次变异操作，染色体是随机选择的，变异位置也是随机选择的，
                        %但该轮for循环中是否进行变异操作则由变异概率决定（continue控制）
                        % 随机选择一个染色体进行变异
pick=rand;              %rand为生成一个[0, 1]的随机数，相当于rand ()
while pick==0           %生成0时重新随机
    pick=rand;
end
index=ceil(pick*sizepop); % 确定第几个染色体 一个个体是否可以变异多次
% 变异概率决定该轮循环是否进行变异
pick=rand;
if pick>pmutation       % 若生成的随机数大于变异概率，则此轮循环不进行变异
    continue;
end
flag=0;
while flag==0
    % 变异位置
    pick=rand;
    while pick==0
        pick=rand;
    end
    pos=ceil(pick*sum(lenchrom)); %随机选择了染色体变异的位置，即选择了第pos个变量进行变异
    %%变异操作
    pick=rand; %变异开始
    fg=(rand*(1-num/maxgen))^2; %%变异公式maxgen为最大迭代次数（最大变异次数）

    if pick>0.5
        chrom(index,pos)=chrom(index,pos)+(chrom(index,pos)-bound(pos,2))*fg;%bound(pos,2)为该个体（染色体）基因的上届
        %修改过index，因为选择哪个个体变异
    else
        chrom(index,pos)=chrom(index,pos)-(chrom(index,pos)-bound(pos,1))*fg;%bound(pos,1)为该个体（染色体）基因的下届
    end %变异结束

```

图 5-12 变异操作程序

5.4 调参方案及结果分析

遗传算法的参数中交叉概率和变异概率的选择是影响遗传算法行为和性能的关键所在，直接影响算法的收敛性。

交叉概率越大，新个体产生的速度就越快。然而，交叉概率过大时遗传模式被破坏的可能性也越大，使得具有高适应度的个体结构很快就会被破坏；但是如果交叉概率过小，会使搜索过程缓慢，以至停滞不前。

变异概率取值过小的情况下，就不易产生新的个体结构；如果其取值过大，那么遗传算法就变成了纯粹的随机搜索算法。

因此，本实验进行了对交叉概率和变异概率反复调整。

1) 交叉概率对实验结果的影响及分析

交叉概率的范围一般在[0.5,0.9]区间内选择，因此，针对本实验数据，控制突变概率在0.001情况下，主要选择交叉概率 0.5，0.8，0.9时的训练结果进行分析。

i. 网络泛化能力(validation check)

	pcross=0.5	pcross=0.8	pcross=0.9
best validation	0.069577	0.11335	0.061299

(表5.1)

从表5.1可以看出，泛化能力在交叉概率为0.5时表现一般，交叉概率处于0.8时表现最好，而在概率为0.9时又骤然下降。

分析以上结果，可以得出交叉概率不宜过小，否则样本的适应度函数变化较小，对神经网络泛化能力没有明显改善；而交叉概率选择也不宜过大，否则将可能破坏高适应度表现的个体，降低神经网络的泛化能力。

ii. 网络梯度下降速率比较

	pcross=0.5	pcross=0.8	pcross=0.9
gradient descent	0.0383	0.0317	0.0154

(表5.2)

从表5.2可以看出，梯度下降速率随交叉概率增加而下降，由此给出分析与解释，随交叉概率增加，神经网络的权值和阈值更新频率增加，因此梯度下降速率逐渐减小。

2) 变异概率对实验结果的影响及分析

变异概率影响了GA遗传算法中的终止代数，而最佳适应度都大致在[32,33]区间内；

下面选择在交叉概率为0.8情况下，变异概率在0.001，0.01以及0.1时的终止代数，仍以表格方式呈现。

	pmutation=0.001	pmutation=0.01	pmutation=0.1
终止代数	34	54	98

(表 5.3)

分析表 5.3 结果，可以得知变异概率增大可以显著影响终止代数值，进而可以大幅度提高神经网络阈值和权值更新的频率，增加得到全局最优解的可能性，该实验结果与对 GA 遗传算法优化 BP 神经网络权值阈值选择的设想完全一致。

3) 最佳训练结果展示与分析

大撒上，经过反复实验，最终，在交叉概率为 0.8，变异概率为 0.1 时，GA 遗传算法优化 BP 神经网络的训练结果表现最佳，展示如下：

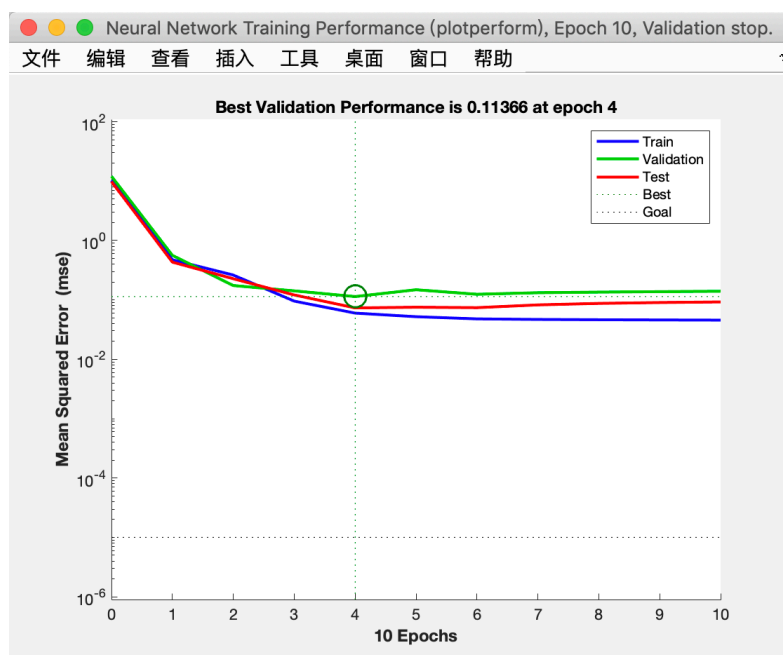
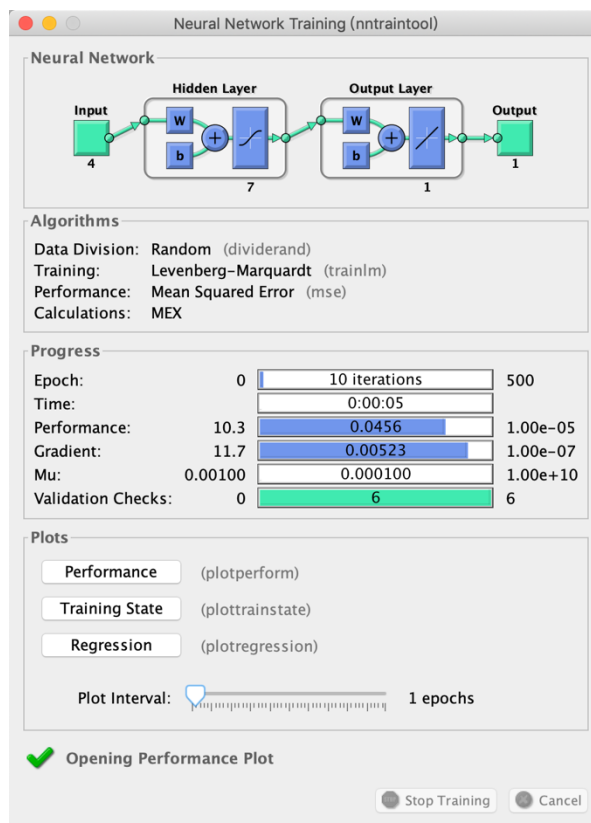


图 5-13 GA 算法优化 BP 神经网络结果

比较图 5.13 与单独使用 BP 神经网络训练结果，可以看出得到全局最优解的迭代次数明显减少；且神经网络训练结果的模型泛化能力显著增强，该优化结果符合优化设计预期。

5.5 优化方案总结与待改进问题

GA 遗传算法由于受交叉概率和变异概率的影响很大，在针对不同的优化问题时，需要反复实验

来确定交叉和变异概率，且一般很难找到适应于每个问题的最佳值。正如本实验所示，当算法优化指标不止一个时，反复实验所需要的横向以及纵向对比，十分依赖于手动调整参数，因此也耗费了实验者大量的时间成本。

基于以上 GA 遗传算法优化神经网络存在的局限性问题，应该设想一种自适应的遗传算法，使得交叉概率和变异概率能够随适应度自动改变。自适应策略如下所述：

- a) 当种群个体适应度趋于一致或趋于局部最优时，使交叉概率和变异概率二者增加，当群体适应度比较分散时，使交叉概率和变异概率减少。
- b) 对于适应度值高于群体平均适应度的个体，对应于较低的交叉概率和变异概率，使该个体得以保护进入下一代；低于平均适应度的个体，相对应于较高的交叉概率和变异概率，使该个体被淘汰掉。

综上所述，本实验中，使用 GA 遗传算法优化基于 BP 神经网络的手写数字识别模型得到了较好的优化结果；但是所采用的仍然是最基本的优化策略，在面对大型工程项目时难以投入使用；因此，在以后的学习过程中，我希望能够通过更多的文献和实验，来改进目前较为基础的优化算法。

第六章 总结与展望

6.1 实验总结

手写数字识别技术是字符识别技术的一个分支，在邮政编码，福利抽奖，钞票和排队等方面具有很大的应用价值。同时，它的识别也是图像处理和模式识别的重要研究热点。本文基于 BP 神经网络，提出了一种手写数字识别的设计方案。经验证，所设计的识别方案具有准确、高效的特点。本文的主要工作内容如下所示：

1. 提出了一种基于 BP 神经网络的手写数字识别方案；
2. 基于 MATLAB 软件实现了设计方案的功能，并进行了验证；
3. 实现了 GUI 界面的设计并对其进行功能验证。
4. 利用遗传算法对 BP 神经网络初始权值和阈值进行优化，得到了符合预期的优化结果。

6.2 未来展望

虽然本文针对手写数字识别提出了一种高效、稳定的基于 BP 神经网络的识别方案，但是本文的研究仍然具有一定的局限性，数字识别的模板库数据还需要进一步扩充。在分析本文工作内容的基础上，今后可以开展的工作如下：

1. 建立更为广泛的数据库；
2. 实现脱机与在线的数字识别综合系统设计。

