

单位代码: 10293 密 级:

南京邮电大学
硕 士 学 位 论 文



论文题目: 基于神经网络的短时交通流量预测研究

学 号	<u>1010061627</u>
姓 名	<u>吴凯</u>
导 师	<u>周西峰 副教授</u>
学 科 专 业	<u>控制理论与控制工程</u>
研 究 方 向	<u>计算机控制与系统集成</u>
申请学位类别	<u>工学硕士</u>
论文提交日期	<u>二〇一三年六月</u>

Study of short-term traffic flow prediction based on neural network

Thesis Submitted to Nanjing University of Posts and
Telecommunications for the Degree of
Master of Engineering



By

WU Kai

Supervisor: Prof. Zhou Xifeng

June 2013

南京邮电大学学位论文原创性声明

本人声明所呈交的学位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，也不包含为获得南京邮电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。

本人学位论文及涉及相关资料若有不实，愿意承担一切相关的法律责任。

研究生签名：_____ 日期：_____

南京邮电大学学位论文使用授权声明

本人授权南京邮电大学可以保留并向国家有关部门或机构送交论文的复印件和电子文档；允许论文被查阅和借阅；可以将学位论文的全部或部分内容编入有关数据库进行检索；可以采用影印、缩印或扫描等复制手段保存、汇编本学位论文。本文电子文档的内容和纸质论文的内容相一致。论文的公布（包括刊登）授权南京邮电大学研究生院办理。

涉密学位论文在解密后适用本授权书。

研究生签名：_____ 导师签名：_____ 日期：_____

摘要

智能交通系统（ITS）是目前缓解城市交通压力、减少环境污染的一种有效、新型的方法。交通流量预测是智能交通系统的重要组成部分，其中短时交通流量预测是交通流量预测中主要的研究内容，及时获取短时交通流量预测的信息有助于保障交叉路口的畅通。因此如何获取准确的短时交通流量预测信息是保证交通有效运行的关键。

根据城市路况交通流量高度的非线性和不确定性等特点以及以往交通流量预测的方法，本文在对神经网络研究的基础上建立了基于 BP 神经网络的短时交通流量预测模型，通过仿真实验验证了梯度下降法的不足，使用 LM 学习算法作为 BP 神经网络的训练算法，并通过仿真实例验证了该算法的可行性。

为了克服 BP 神经网络因初始参数选取不当而造成易陷入局部极小点的缺点，将智能优化算法引入到优化 BP 神经网络的初始参数中，分别介绍了粒子群优化算法和遗传算法的基本原理以及它们的改进算法，并利用这些智能算法优化 BP 神经网络的初始权值和阈值，在此基础上分别建立了基于 PSO-BP、MPSO-BP、IAGA-BP 神经网络的短时交通流量预测模型，仿真结果表明基于智能算法优化后的神经网络的短时交通流量预测模型预测精度得到了提高。

最后对文章中提到的预测算法作了分析总结，并将遗传算法的思想引入到粒子群算法中，提出了 APSO 算法，该算法结合了粒子群算法和遗传算法的优点，在此基础上建立了基于 APSO-BP 神经网络的短时交通流量预测模型，仿真结果表明该预测模型具有更高的预测精度。

关键词： 交通流量，预测，神经网络，粒子群优化算法，遗传算法

Abstract

Abstract: Intelligent Transportation Systems (ITS) is an effective and a new way to ease urban traffic pressure and reduces environmental pollution. Traffic flow forecasting is an important part of ITS, and the short-term traffic forecasting is the main study content of the traffic flow forecasting. It can ensure the smooth flow of the intersection by accepting the short-term traffic forecast information timely. Therefore, how to obtain the accurate short-term traffic forecast information is the key to ensure the effective operation of the traffic.

According to the highly nonlinear and uncertain characteristics of urban road traffic and the methods of the previous traffic flow forecasting, this paper establishes the short-term traffic flow prediction model based on the research of BP neural network, training the BP neural network by the LM learning algorithm, and then forecasts the city intersection traffic flow, the simulation example demonstrates the feasibility of the algorithm.

In order to overcome the shortcomings of the BP neural network that is easy to fall into local minimum points caused by the improper selection of the initial parameters, this paper optimizes the initial parameters of the BP neural network by the intelligent optimization algorithm, and introduces the basic principles of the particle swarm optimization and genetic algorithm and their improved algorithms, then optimizes the initial weights and thresholds of BP neural network by the intelligent algorithms, and establishes the short-term traffic flow prediction model based on PSO-BP, MPSO-BP, IAGA-BP neural network forecasting model on this basis. After the intelligent algorithms optimize the neural network, the simulation shows that the prediction accuracy is improved.

Finally this paper makes a summary of the prediction algorithm, and introduces the idea of genetic algorithm to the particle swarm optimization, proposes the APSO algorithm which combines the advantages of the PSO and the genetic algorithm, then establishes the short-term traffic flow prediction model based on APSO-BP, and the simulation shows that the prediction model has higher prediction accuracy.

Key words: traffic flow, prediction, neural network, particle swarm optimization, genetic algorithm

目录

第一章 绪论	1
1.1 课题的研究背景和意义	1
1.2 国内外 ITS 研究现状	1
1.2.1 智能交通系统发展概述	1
1.2.2 交通流量预测的发展	2
1.3 本文的工作内容	3
1.3.1 本文的研究工作	3
1.3.2 章节安排	4
第二章 人工神经网络的基本理论	5
2.1 人工神经网络研究的发展历史	5
2.2 人工神经网络基础知识	6
2.2.1 人工神经网络模型	6
2.2.2 神经网络的训练和学习	9
2.3 BP 神经网络的基本理论	10
2.3.1 BP 神经网络概述	11
2.3.2 BP 学习算法	12
2.4 本章小结	15
第三章 BP 神经网络在短时交通流量预测中的应用	16
3.1 基于 BP 神经网络交通流量预测模型的建立	16
3.1.1 输入层数据的选择	16
3.1.2 BP 神经网络预测模型分析	18
3.2 基于 BP 神经网络预测模型仿真实例	21
3.2.1 传统 BP 算法的不足及改进	21
3.2.2 基于 LM 算法 BP 神经网络的交通流量预测结果及分析	26
3.3 本章小结	30
第四章 智能算法优化 BP 神经网络在交通流量预测中的应用	31
4.1 粒子群算法及其改进	31
4.1.1 基本粒子群算法	31
4.1.2 粒子群算法的缺陷及其改进	33
4.1.3 MPSO-BP 神经网络混合算法	35
4.2 基于粒子群神经网络短时交通流量预测结果分析	36
4.3 遗传算法在交通流量预测中的应用	39
4.3.1 基本遗传算法	39
4.3.2 基本遗传算法的不足及改进	43
4.4 IAGA-BP 神经网络复合预测算法	46
4.5 基于 IAGA-BP 神经网络短时交通量预测结果分析	46
4.6 本章小结	49
第五章 APSO-BP 神经网络交通流量预测模型	50
5.1 自适应变异算子的引入	50
5.2 基于 APSO-BP 神经网络短时交通流量预测结果分析	50
5.3 预测算法分析总结	52
5.4 本章小结	53
第六章 总结与展望	54

参考文献 56

附录 1 攻读硕士学位期间撰写的论文 58

致谢 59

第一章 绪论

1.1 课题的研究背景和意义

随着国民经济和我国基础设施建设的快速发展,智能交通系统(ITS)、交通诱导和交通规划已经逐渐成为交通领域研究的热门^[1]。智能交通技术是有效解决当前交通问题的一种综合的多学科交叉技术。智能交通技术起源于美国,目前已经在各国得到了广泛的应用和发展。但对于这项技术来说,准确的交通流量预测无疑是实现它的前提和基础,因此怎样准确的获得预测信息已成为这方面研究的一个热点^[2]。

城市交叉路口是智能交通预测系统中的主要研究对象,根据预测的时间跨度可分为短期交通流量预测和长期的交通流量预测。其中短时交通流量预测一般被认为时间跨度不超过 15 分钟^[3],它也是 ITS 领域主要的研究内容,及时获取短时交通流量预测的信息有助于保障交叉路口的畅通,交叉路口的畅通是保障城市交通有效运作的前提和关键,同时短时交通流量预测信息能够有效的帮助交通管理从静态向智能方向转变,从而实现 ITS 的动态优化,保证道路交通的畅通^[4-6]。

交叉路口的流量具有高度非线性和不确定性等特点^[7],并且具有较强的时间相关性,这些特点给交通流量预测带来了困难,再加上随机干扰因素的影响,使得流量预测的不确定性和不规律性更加明显。传统的交通流量预测方法包括时间序列法、趋势预测法、ARIMA 模型法^[8]已经不能满足实践中越来越高的精度要求,经典的纯数学方法也很难建立起标准的数学模型来取得满意的结果,因此必须寻求新的预测方法。随着计算机技术的发展以及交通流量预测研究本身的深入,一些精度更高的智能方法不断涌现,这其中包括小波分析、灰色预测、混沌分析以及人工神经网络等方法^[9]。神经网络具有其极强的非线性映射能力,在非线性信息预测的领域取得了广泛的应用。近年来,一些国内外学者已经将其应用于交通流量的预测^[10-12]。

1.2 国内外 ITS 研究现状

1.2.1 智能交通系统发展概述

智能交通系统(Intelligent Traffic System)简称 ITS 起源于上个世纪 60 年代的美国,此

后世界各地都相继开展了有关这方面的研究。随着国民经济的高速发展,日益突出的交通问题和能源问题使得各国纷纷成立各自相应的机构,并制定相应的 ITS 标准和开发计划,如美国成立的社会组织 Intelligent Traffic System of America(ITS AMERICA),欧洲 19 个国家投资的 EUREKA 项目以及日本的交通智能协会 (VERTIS) 等。

我国的 ITS 发展相对于发达国家起步较晚,但是我们综合我国的基本国情积极参加了各方面的学术会议,积极吸收国外的先进经验。1994 年在法国巴黎召开的第一届世界 ITS 大会,揭开了中国 ITS 发展的序幕。随着国内研究人员的不断努力,我国在 ITS 领域的研究得到了长足的进步和发展。在 2007 年 10 月,第十四届世界 ITS 大会在北京召开,这是我国第一次举办世界 ITS 大会,也证明了我国在世界 ITS 领域地位的提高。

1.2.2 交通流量预测的发展

为了保证 ITS 有效的运作,准确的交通流量预测信息是其实现的前提和关键,国内外对其研究一直都很活跃。早期的交通流量预测是为交通控制系统服务的,根据预测原理的不同,预测的算法可以分为三类,第一类是以传统的数学和物理的方法为基础,建立一个合理的数学预测模型,可称为传统预测模型;第二类主要以现代的科学技术(如模糊技术、混沌技术和神经网络技术)为研究手段的预测模型,称为智能预测模型;第三类是组合的预测的方法,合理的将不同的预测模型组合起来,利用不同模型的优点组合出适当的预测模型来达到提高预测精度的目的^[13]。

传统的数学预测模型是利用历史的交通流量数据,假定未来的交通流量数据与过去的数据具有一定的相关性,这样就可以找到用历史数据来表示未来时刻的数据。基于数学分析的主要有自回归模型 (Autoregressive AR)、滑动平均模型 (Moving Autoregressive MA)、自回归滑动平均模型 (Autoregressive Moving Average Model ARMA) 以及卡尔曼滤波模型 (Kalman Filtering Model) 等。在国外, Nancy L.Nihan 和 Kjell Holmeslan 利用一个路段 4 年的交通流量数据对该路段的交通流量进行预测,预测精度较高,但是该方法不仅要求较多的历史数据,对使用者的数学知识水平以及建模能力和技巧都要求较高。基于现代科学技术的智能预测方法不是用数学方程来描述预测数据和历史数据之间的关系。而是建立起一个本身的预测结构,凭借自身的“记忆”、“学习”和“经验”的分析总结来达到准确预测的目的^[14-16]。

这类方法的优点在于具有动态的学习和分析能力,在复杂的非线性、非平稳随机过程的领域中具有很好的效果,常见的智能预测算法有神经网络模型和非参数回归模型等。Nancy L.Nihan 和 Gary A.Davis 利用无参数回归模型预测交通流量,在某些情况下比基于数学分析方

法的结果要好, Michael J.Demestky 和 Brain L.Smith 利用神经网络的方法来进行预测, 预测的结果没有时滞现象且具有较高的预测精度。组合预测 (combination forecasting) 是根据各种预测方法的优点将不同的模型组合起来, 以合适的方法得出复合模型的预测结果, 以达到提高预测精度的目的。Maschavan der Voort 等人将神经网络模型和 ARIMA 模型有效的结合起来, 使得 ARIMA 模型得到了更好的推广和移植, 但对组合预测模型的研究还不是很多, 一方面因为交通流量数据的高度非线性, 且有关这方面的理论还不是很充分, 还处于发展阶段。但是就初步应用的结果来看, 组合预测理论无论在精确度, 鲁棒性上来说都是单项预测所不能及的。因此我国的 ITS 系统在日后的发展中将在组合预测方面的研究更加的广泛和深入。

总而言之, 非线性预测方向的理论研究在交通流量的预测中已经占着重要的地位, 交通流量中的随机性、非线性在预测中越来越得到重视。而神经网络的优点在于只利用观察到的数据, 可以从训练过程中通过学习来抽取和逼近隐含的输入输出非线性关系, 且不需要精确的数学表达式, 预测精度较高, 适应于交通流量及交通参数的预测, 在 ITS 中有着广阔的应用前景。

1.3 本文的工作内容

1.3.1 本文的研究工作

本文简要介绍了国内外的智能交通系统的研究和发展现状, 确立了交叉路口的交通流量预测在 ITS 中的重要地位, 并就目前几种常用的交通流量预测算法的优缺点进行了评价。考虑到交通流量的非线性和随机性以及神经网络在非线性系统预测中的优势, 本文应用人工神经网络技术对短时的交通流量进行预测, 并通过分析比较给出常规神经网络优化算法的不足, 并将智能优化算法应用到优化神经网络的初始参数, 使预测精度得到进一步的提高。

由于交通流量的随机性和非线性, 国内外的一些参考文献在对神经网络在交通流量预测中的应用研究多在理论阶段, 本文在总结国内外研究和分析结果的基础上, 选取合适的神经网络模型结构以及结合智能优化算法优化神经网络初始参数, 主要的研究内容有:

(1) 介绍人工神经网络的基本情况, 特别研究 BP 神经网络的基本原理及其在交通流量预测中的应用;

(2) 分析传统 BP 神经网络在交通流量预测中的不足, 提出与智能优化算法相结合的复合预测方法, 分别研究了粒子群算法和遗传算法的基本原理及其改进算法, 建立了相应的预测模型预测交通流量;

(3) 综合比较了各种预测模型的优缺点, 提出将遗传算法的思想运用到粒子群算法中, 将带有遗传思想的粒子群算法优化 BP 神经网络, 最后利用该模型对算例进行了预测, 验证算法可行性。

1.3.2 章节安排

本文的主要目的是研究交通流量的预测方法, 以提高交通流量预测精度为主要目的, 将神经网络以及智能的优化算法相结合运用到交通流量预测中, 全文总共分为 6 个部分, 其主要工作内容包括:

第一章: 总结了国内外的 ITS 的研究现状, 并确立了交通流量预测在 ITS 研究中的重要性, 明确了课题研究的目的和意义, 最后确定本文具体研究工作。

第二章: 主要介绍人工神经网络的基本理论, 发展概况, 着重阐述了 BP 神经网络的结构、数学表达式以及常用的训练算法。

第三章: 介绍了 BP 神经网络在交通流量预测中的应用, 研究了基于 BP 神经网络预测模型的建立, 通过仿真实验验证了传统梯度下降法的不足。

第四章: 研究了智能优化算法优化 BP 神经网络的初始参数在交通流量预测中的应用, 重点介绍了粒子群算法和遗传算法的基本原理及其改进。

第五章: 对文中提到的预测算法的预测结果进行了比较, 最终提出将遗传算法的思想应用到粒子群算法中去, 用带有遗传思想的粒子群算法优化 BP 神经网络来达到提高预测精度的目的, 通过仿真验证其可行性。

第六章: 总结和展望部分, 对本论文中提到的算法的优缺点进行总结和归纳, 并对下一步的研究工作提出设想。

第二章 人工神经网络的基本理论

人工神经网络^[17-19] (Artificial Neural Networks) 简称为 ANNS, 也称为神经网络 (NNS) 或者称为连接模型 (Connection Model), 它主要是模仿动物的神经行为特征, 能进行并行和分布式信息处理的数学模型。它是通过调节内部节点之间复杂的连接关系来进行信息的处理。神经网络前后经历了 60 多年曲折的发展, 在广大的领域发挥着越来越大的作用。

2.1 人工神经网络研究的发展历史

人工神经网络起源于 20 世纪 40 年代, 在半个多世纪的发展历程里, 它经历了一条兴起到衰落、再由衰落到兴起的曲折过程。

1943 年, 心理学家 W•McCulloch 和数理逻辑学家 W•Pitts 通过研究神经元的基本特征提出了神经元的数学模型。该模型一直对这一领域的研究起着指导和影响作用, 因此他们二人也被称为神经网络研究领域的先驱。

1945 年, 冯•诺依曼领导设计了存储式的电子计算机, 并取得了成功, 这标志着电子计算机时代的开端。1948 年, 他在研究中比较了存储式计算机结构与人脑结构的区别, 并提出了以简单神经元作为基本单元构造自动网络结构, 冯•诺依曼虽然引导的是普通计算机的发展, 但他对人工神经网络的发展也起着推动作用。

50 年代末, F•Rosenblatt 设计了一种多层神经网络称为“感知机”, 这是首次将人工神经网络从理论阶段上升到工程实践中的一次尝试, 这引领了首次神经网络研究的热潮。当时, 世界上许多实验室纷纷仿制感知器并将其应用在各个领域。但这一波研究热潮没有持续很长时间就有许多研究者放弃了这一研究, 这主要是因为当时数字计算机的发展越来越火热, 使许多人误以为数字计算机能够取代神经网络解决人工智能、模式识别等一系列问题, 从而不重视对感知机的研究; 同时由于电子水平技术的落后, 而且当时主要的电子元件是电子管和晶体管, 用来制作的神经网络体积大且价格昂贵, 要完成真实的神经网络规模几乎是完全不可能的; 最后在 1968 年出版的一本名为《感知机》的著作中提出了感知机功能的缺陷, 它甚至不能解决如异或这类基本的问题, 而且当时神经网络的理论在多层网络中还很有缺陷, 一直找不到合适的计算方法, 至此, 人工神经网络在 60 年代末的研究陷入了低潮。

在 60 年代初期, Widrow 提出了一种自适应线性元件网络, 并在此基础上发展了多层的自适应网络。虽然这种模型并未以神经网络命名, 但实际上也是人工神经网络的一种。

由于感知机性能的缺陷,使得神经网络模型的研究沉寂了很长的时间。直到 80 年代初期,数字和模拟混合的超大规模集成电路得到了快速提升并付诸于应用,另外,在若干领域数字计算机的应用受到了限制。这预示着人工神经网络的发展时机已经成熟,之后美国物理学家 Hopfield 发表了两篇关于人工神经网络研究的论文引起巨大反响,这也使得人们重新认识到了神经网络的重要性。随后,大批的专家和学者围绕 Hopfield 神经网络开展了进一步的研究。

从神经网络复杂曲折的发展历史以及应用前景不难看出,神经网络的发展前景具有强大的生命力,虽然当前神经网络的智能水平不高,许多数学理论以及实际方面的应用还没有得到很好的解决,但是随着人们在智能学科领域的深入研究以及对人类大脑信息处理机制的不断认知,人工神经网络必然会在智能科学领域有着更加广阔的应用空间。

2.2 人工神经网络基础知识

2.2.1 人工神经网络模型

人的大脑由多达 10^{11} 个不同种类的神经元(神经细胞)组成,神经元的主要功能是传输信息。信息在一个神经元上是以电脉冲的形式传输的,这种电脉冲称为动作电位。一个神经元产生动作电位的最大次数约等于 500 次/s。

人工神经网络的第一个数学模型 MP 模型由 McCulloch 和 Pitts 建立,该模型的基本工作原理为:神经细胞的工作方式为兴奋或者是抑制。根据这一原理,McCulloch 和 Pitts 在神经元模型中采用了硬极限函数,这种方式后来也被多层感知机、离散 Hopfield 神经网络等所采纳。神经元中突触的活动强度决定了信号之间的连接强度,因此在 MP 模型中,神经元之间突触的活动强度由具体的固定实数即权值来表示。同时,模型中的神经元可以从大量的其他神经元接收信息,来产生相应的兴奋和冲动;在其他条件不变的前提下,只要兴奋达到了阈值之上,就会产生动作电位。相反,如果这个刺激总和低于阈值,就不会产生任何反应。图 2.1 所示为 MP 模型示意图。图中, x_1, x_2, \dots, x_n 为神经元的输入; w_1, w_2, \dots, w_3 为神经元的连接权值; T 为神经元的阈值; y 为神经元对应的输出。其中神经元的输出采用二值函数的表示方式,即:

$$y = \begin{cases} 1, & \sum_{i=1}^n w_i x_i \geq T \\ 0, & \sum_{i=1}^n w_i x_i < T \end{cases} \quad (2.1)$$

式中， x_i 代表了神经元第 i 个输入； y 则表示神经元相应的输出； w_i 表示神经元第 i 个输入权值，而 T 则代表相应的阈值； n 为输入个数。

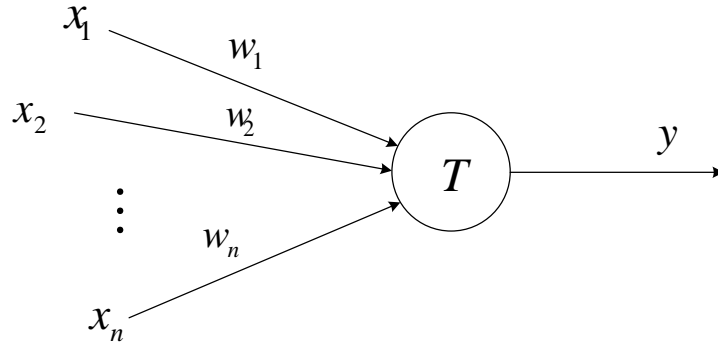


图 2.1 MP 模型

单个 MP 神经元模型可以进行或、与、或非、与非等二值逻辑的计算，但不能进行异或运算。该模型只要通过简单的计算就能产生复杂的行为，这曾引起很大反响，但由于它结构固定，权值阈值无法自行调节，即是静态的神经元。所以 MP 模型缺乏最关键的学习能力要素。

由于 MP 模型的权值和阈值不能自主动态的学习，在实际应用中受到很大的限制，因此需要灵活度复杂性更高的神经元模型。如图 2.2 所示，与 MP 模型一样是一个具有 n 个输入的通用神经元模型。其中， $\mathbf{x}=(x_1, x_2, \dots, x_n)^T$ 为神经元输入， $\mathbf{w}=(w_1, w_2, \dots, w_n)^T$ 为动态可调的权值， θ 表示偏移信号，用来模拟神经元的兴奋阈值， $u(\bullet)$ 和 $f(\bullet)$ 分别作为神经元的基函数和激活函数，其中基函数 $u(\bullet)$ 是一个多输入单输出函数 $u = u(x, w, \theta)$ ，激活函数 $f(\bullet)$ 的作用是对基函数输出 u 进行压缩： $y = f(\bullet)$ 。下面介绍常用的基函数及激活函数的类型。

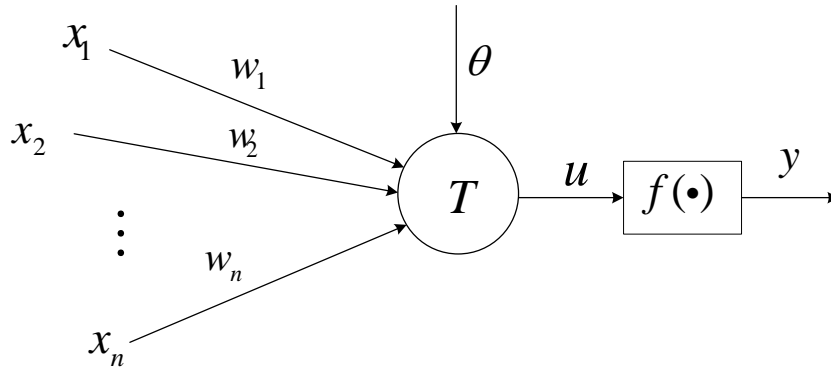


图 2.2 通用神经元模型

大多数的神经网络都以线性函数作为基函数的表达方式，如 BP 神经网络、RBF 神经网络等。当使用线性函数时，基函数的输出 u 为输入值和阈值的加权和，即

$$u = \sum_{i=1}^n w_i x_i - \theta = x^T w - \theta \quad (2.2)$$

该基函数的形状在多维空间中表现为一个超平面。激活函数也称为神经元函数、挤压函数或活化函数，它是人工神经元中的一个重要概念，它的主要作用是通过函数对输入输出进行范围变化，其中 Sigmoid 函数也叫 S 形函数，是目前人工神经网络常用的激活函数。S 形函数是一个严格单调递增的光滑曲线。对数正切函数为 S 形函数中的一种形式，其函数表达式为

$$y = f(u) = \frac{1}{1 + e^{-\lambda u}} \quad (2.3)$$

式中，参数 λ 称作 Sigmoid 函数的增益，为 S 形函数的斜率参数，通过改变此参数可以得到不同斜率的 S 形函数。对数正切函数也叫单极性 Sigmoid 函数，该函数是可微的且取值在 0 到 1 的区间内连续变化。图 2.3 给出了当 $\lambda=1$ 时对数正切函数的函数图形。方程 (2.3) 定义的 S 形激活函数的值域为 0 到 1。有时，激活函数的值域需要从 -1~1 变化，且关于原点是奇对称的。因此，可以采用双曲正切 S 形激活函数（也叫双极性 Sigmoid 函数）。

$$y = f(u) = \tanh(\lambda u) = \frac{e^{\lambda u} - e^{-\lambda u}}{e^{\lambda u} + e^{-\lambda u}} \quad (2.4)$$

同样， λ 为该函数的斜率参数，通过改变 λ 的值就可以得到不同斜率的 S 形函数。图 2.4 给出了当 $\lambda=1$ 时双曲正切函数的函数图形。

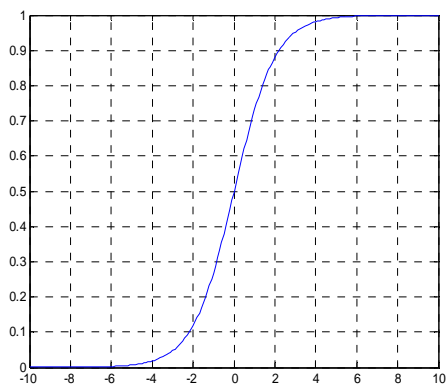


图 2.3 单极性 Sigmoid 函数

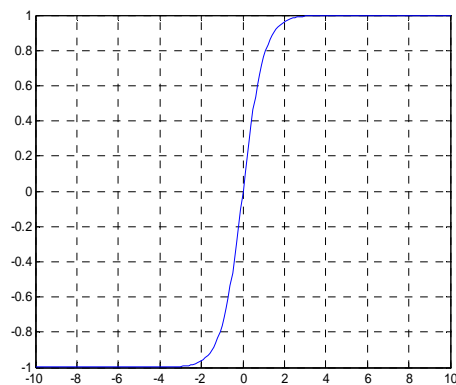


图 2.4 双极性 Sigmoid 函数

2.2.2 神经网络的训练和学习

生物神经系统能够从周围的环境进行学习，所以能够适应环境，而强大的学习能力也是人工神经网络重要的特点。必须要对神经网络进行训练才能实现它对某种功能的操作，就是要它学会做这件事，并且神经网络具有记忆功能，能把这些记忆存储在网络权值中。实际上就是权值矩阵根据外部激励的变化来做自适应的变化。数学表达式为：

$$\frac{dw}{dt} \neq 0 \quad (2.5)$$

神经网络的学习或训练的实质是变动的加权，所以它的学习和训练是混用的。训练的本意就是加入输入的向量并按照预定好的规则来调节自身的权值，训练过程中，网络的权值都要收敛到一个定值以保证每个输入产生一个要求的输出。这个预先的规则就是训练算法。一般的训练算法分为两类：有指导的训练和无指导的训练。常见的学习规则有：

(1) Hebb 学习规则

1949 年，神经生物学家 Hebb 在研究中提出了有关神经网络学习机理中的“突触修正”的假说，即当神经元的突触膜前后电位同时为正时，则突触导通性加强，相反当神经元的突触膜前后电位正负相反时，突触导通性减弱。根据这种定义假设的权值调整方法称为 Hebb 学习规则。该规则代表了一种无导师、前馈的学习方法，一般用于自组织网络中。根据 Hebb 学习规则，假定神经元当前输入 $\mathbf{x}=(x_1, x_2, \dots, x_n)^T$ ，输出 $y=f(\mathbf{w}(t)^T \mathbf{x})$ ，则权值矢量 $\mathbf{w}(t)$ 的调节量 $\Delta \mathbf{w}(t)$ 为：

$$\Delta \mathbf{w}(t) = y\mathbf{x} \quad (2.6)$$

其中 $\Delta \mathbf{w}(t)$ 可以理解为样本 \mathbf{x} 对当前权值的影响度。

(2) δ 学习规则

δ 学习规则是最常用的神经网络学习算法，被称为最速下降法或梯度法，是一种有导师的学习算法。假设神经元权值的修正目标为极小化标量函数 $F(\mathbf{w})$ ，神经元当前的权值为 $\mathbf{w}(t)$ ，且下一时刻的权值调整公式为 (2.7) 所示，式中 $\Delta\mathbf{w}(t)$ 为当前时刻修正方向。

$$\mathbf{w}(t+1) = \mathbf{w}(t) + \eta\Delta\mathbf{w}(t) \quad (2.7)$$

每次期望修正均有：

$$F(\mathbf{w}(t+1)) < F(\mathbf{w}(t)) \quad (2.8)$$

接下来对 $F(\mathbf{w}(t+1))$ 进行一阶泰勒展开，得

$$F(\mathbf{w}(t+1)) = F(\mathbf{w}(t) + \eta\Delta\mathbf{w}(t)) \approx F(\mathbf{w}(t)) + \eta \mathbf{g}(t)^T \Delta\mathbf{w}(t) \quad (2.9)$$

式中， $\mathbf{g}(t) = \nabla F(\mathbf{w}(t)) \Big|_{\mathbf{w}=\mathbf{w}(t)}$ 表示 $F(\mathbf{w})$ 在 $\mathbf{w} = \mathbf{w}(t)$ 时的梯度矢量。显然当

$$\Delta\mathbf{w}(t) = -\mathbf{g}(t) \quad (2.10)$$

此时式 (2.9) 的右边第 2 项就会小于零，式 (2.8) 就会满足，这是梯度法的基本原理。为了得到目标函数的梯度值， δ 学习规则的基函数一般取线性函数，连续可微的 Sigmoid 函数一般作为激活函数。神经元权值调节 δ 学习规则目的是通过训练权值 \mathbf{w} 使得训练样本 (\mathbf{x}, d) 的输出误差达到最小。

$$E = \frac{1}{2}(d - y)^2 = \frac{1}{2}[d - f(\mathbf{w}^T \mathbf{x})]^2 \quad (2.11)$$

计算梯度矢量如 (2.12) 所示，并令 $\Delta\mathbf{w}(t) = -\nabla E_{\mathbf{w}}$ 即能得到式 (2.13) 所示权值修正公式。

$$\nabla E_{\mathbf{w}} = -(d - y) f'(\mathbf{w}^T \mathbf{x}) \mathbf{x} \quad (2.12)$$

$$\Delta\mathbf{w}(t) = (d - y) f'(\mathbf{w}^T \mathbf{x}) \mathbf{x} \quad (2.13)$$

神经元初始权值一般为零左右的随机值， δ 学习规则作为最广泛的学习规则常用于感知器及单层神经网络中。

2.3 BP 神经网络的基本理论

BP 神经网络是最重要的也是应用最广泛的神经网络模型之一，在模式识别、函数逼近、

人工智能预测等方面都有着广泛的应用，并取得了一定的效果^[20]。

2.3.1 BP 神经网络概述

BP 神经网络的特点是多层前馈，即信号由前至后传播，而误差的传播方向则是相反的。在输入信号的传播过程中，是由输入层至隐含层最后到输出层逐层传递，同时下一层神经元的状态受到上一层神经元的影响，输出层的输出没有达到期望值就会将误差进行反向传播，并通过误差来调整自身的权值和阈值，从而使网络的输出逼近期望值。图 2.5 为 BP 神经网络的拓扑结构。

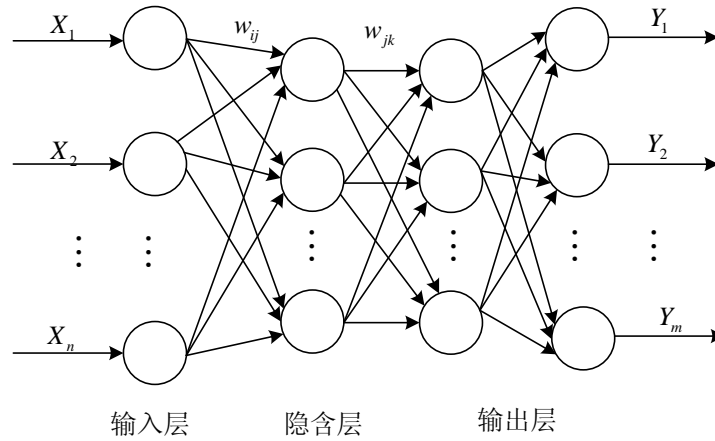


图 2.5 BP 神经网络结构图

图中， X_1, X_2, \dots, X_n 作为 BP 神经网络的输入值， Y_1, Y_2, \dots, Y_m 为神经网络的输出值， w_{ij} 和 w_{jk} 为神经网络的权值。BP 神经网络本身可以看作一个非线性函数，网络的输入输出作为函数本身的自变量与因变量。当网络输入输出节点数分别为 n 和 m 时，代表的就是网络从 n 个自变量到 m 个因变量之间的函数映射关系。假定网络的输入矢量为 $\mathbf{x}=(x_1, x_2, \dots, x_n)^T$ ，第一隐层有 n_1 个神经元对应的输出为 $\mathbf{x}'=(x'_1, x'_2, \dots, x'_{n_1})^T$ ，第二隐层有 n_2 个神经元对应的输出为 $\mathbf{x}''=(x''_1, x''_2, \dots, x''_{n_2})^T$ ，输出层有 m 个神经元对应为 $\mathbf{y}=(y_1, y_2, \dots, y_m)^T$ 。假设输入层到第一隐层的权值阈值为 w_{ij} 和 θ_j ，第一隐层到第二隐层的权值阈值为 w'_{jk} 和 θ'_k ，第二隐层到输出层的权值阈值为 w''_{kl} 和 θ'_l ，得到各层神经元输出为：

$$\begin{cases} x_j' = f(\sum_{i=1}^n w_{ij}' x_i - \theta_j'), & j=1,2,3,\dots,n_1 \\ x_k'' = f(\sum_{j=1}^{n_1} w_{jk}' x_j' - \theta_k''), & k=1,2,3,\dots,n_2 \\ y_l = f(\sum_{k=1}^{n_2} w_{kl}'' x_k'' - \theta_l''), & l=1,2,3,\dots,m \end{cases} \quad (2.14)$$

由此可得出它完成了 n 维到 m 维空间的映射。

2.3.2 BP 学习算法

BP 学习算法的准则是基于最小均方的规则，假设一个样本（假定为第 P 个样本）输入网络产生相应的输出，均方误差（MSE）为各输出单元的误差平方和，即：

$$E^{(p)} = \frac{1}{2} \sum_{l=0}^{m-1} (d_l^{(p)} - y_l^{(p)})^2 \quad (2.15)$$

所有的样本输入之后总的误差为：

$$E_T = \sum_{p=1}^P E^{(p)} = \frac{1}{2} \sum_{p=1}^P \sum_{l=0}^{m-1} (d_l^{(p)} - y_l^{(p)})^2 \quad (2.16)$$

设 w_{sp} 为网络中的一个权值，则由梯度下降法得到批处理方式下的权值修正公式为：

$$\Delta w_{sp} = -\frac{\partial E_T}{\partial w_{sp}} \quad (2.17)$$

并假定此时 S 函数的增益 $\lambda=1$ 。对于输出层有：

$$w_{kl}''(t+1) = w_{kl}''(t) - \eta \frac{\partial E_T}{\partial w_{kl}''} \quad (2.18)$$

式中， t 为迭代次数， $\frac{\partial E_T}{\partial w_{kl}''}$ 的求解采用 δ 学习准则，则：

$$\begin{aligned} \frac{\partial E_T}{\partial w_{kl}''} &= \sum_{p=1}^P \frac{\partial E^{(p)}}{\partial w_{kl}''} = \sum_{p=1}^P \frac{\partial E^{(p)}}{\partial y_l^{(p)}} \frac{\partial y_l^{(p)}}{\partial u_l^{(p)}} \frac{\partial u_l^{(p)}}{\partial w_{kl}''} = -\sum_{p=1}^P (d_l^{(p)} - y_l^{(p)}) f'(u_l^{(p)}) x_k^{(p)} \\ &= -\sum_{p=1}^P (d_l^{(p)} - y_l^{(p)}) y_l^{(p)} (1 - y_l^{(p)}) x_k^{(p)} = -\sum_{p=1}^P \delta_{kl}^{(p)} x_k^{(p)} \end{aligned}$$

式中， $\delta_{kl}^{(p)} = (d_l^{(p)} - y_l^{(p)}) y_l^{(p)} (1 - y_l^{(p)})$ ，因此有：

$$w_{kl}''(t+1) = w_{kl}''(t) + \eta \sum_{p=1}^P \delta_{kl}^{(p)} x_k^{(p)} \quad (2.19)$$

对于中间隐含层有：

$$w'_{jk}(t+1) = w'_{jk}(t) - \eta \frac{\partial E_T}{\partial w'_{jk}} \quad (2.20)$$

$\frac{\partial E_T}{\partial w'_{jk}}$ 的求解准则同样采用 δ 学习准则，即：

$$\begin{aligned} \frac{\partial E_T}{\partial w'_{jk}} &= \sum_{p=1}^P \frac{\partial E^{(p)}}{\partial w'_{jk}} = \sum_{p=1}^P \sum_{l=0}^{m-1} \frac{\partial E^{(p)}}{\partial y_l^{(p)}} \frac{\partial y_l^{(p)}}{\partial u_l^{(p)}} \frac{\partial u_l^{(p)}}{\partial x_k^{(p)}} \frac{\partial x_k^{(p)}}{\partial u_k^{(p)}} \frac{\partial u_k^{(p)}}{\partial w'_{jk}} \\ &= - \sum_{p=1}^P \sum_{l=0}^{m-1} (d_l^{(p)} - y_l^{(p)}) f'(u_l^{(p)}) w''_{kl} x_k^{(p)} (1 - x_k^{(p)}) x_j^{(p)} \\ &= - \sum_{p=1}^P \sum_{l=0}^{m-1} \delta_{kl}^{(p)} w''_{kl} x_k^{(p)} (1 - x_k^{(p)}) x_j^{(p)} = - \sum_{p=1}^P \delta_{jk}^{(p)} x_j^{(p)} \end{aligned}$$

在式中， $\delta_{jk}^{(p)} = \sum_{l=0}^{m-1} \delta_{kl}^{(p)} w''_{kl} x_k^{(p)} (1 - x_k^{(p)})$ ，因此有：

$$w'_{jk}(t+1) = w'_{jk}(t) + \eta \sum_{p=1}^P \delta_{jk}^{(p)} x_j^{(p)} \quad (2.21)$$

同理我们可以得到第一隐层的权值修正公式为：

$$w_{ij}(t+1) = w_{ij}(t) + \eta \sum_{p=1}^P \delta_{ij}^{(p)} x_i^{(p)} \quad (2.22)$$

式中， $\delta_{ij}^{(p)} = \sum_{k=0}^{n_2} \delta_{jk}^{(p)} w'_{jk} x_j^{(p)} (1 - x_j^{(p)})$ 。推导的过程中的学习分为两个阶段，第一阶段是由

前向后计算各层的输出，下一阶段反过来由误差反向传播来进行权值修正。由此可得算法的主要步骤为：

(1) 权值的随机初始化。 $w_{sp} = \text{Random}(\bullet)$ ， sp 为 ij 、 jk 或 kl 。

(2) 依此输入 P 个学习样本，假设当前输入的是第 p 个样本。

(3) 根据网络结构依次计算出各层的输出， x'_j ， x''_k 以及 y_l ， $l=0,1,2,\dots,m-1$ ； $k=0,1,2,\dots,n_2$ ；

$j=0,1,2,\dots,n_1$ 。

(4) 分别求各层的反传误差。

$$\delta_{kl}^{(p)} = (d_l^{(p)} - y_l^{(p)}) y_l^{(p)} (1 - y_l^{(p)}), \quad l=0,1,2,\dots,m-1$$

$$\delta_{jk}^{(p)} = \sum_{l=0}^{m-1} \delta_{kl}^{(p)} w''_{kl} x_k^{(p)} (1 - x_k^{(p)}), \quad k=0,1,2,\dots,n_2$$

$$\delta_{ij}^{(p)} = \sum_{k=0}^{n_2} \delta_{jk}^{(p)} w'_{jk} x_j^{(p)} (1 - x_j^{(p)}), \quad j=0,1,2,\dots,n_1$$

并记下当前各个 $x_k^{(p)}$, $x_j^{(p)}$ 及 $x_i^{(p)}$ 的值。

(5) 记录已经学习的样本个数 p , 若 $p < P$, 则转入步骤 (2) 继续计算, 否则转入步骤 (6)

(6) 由权值修正公式修正各自的权值。

(7) 按照新的权值公式计算 x'_j , x''_k 以及 y_l 和 E_T , 若每个 p 和 l 均满足 $|d_l^{(p)} - y_l^{(p)}| < \varepsilon$ (或者 $E_T < \varepsilon$) 或达到了最大的学习次数, 则学习终止, 否则继续转入步骤 (2) 新一轮新的学习。

BP 算法的流程如下图所示:

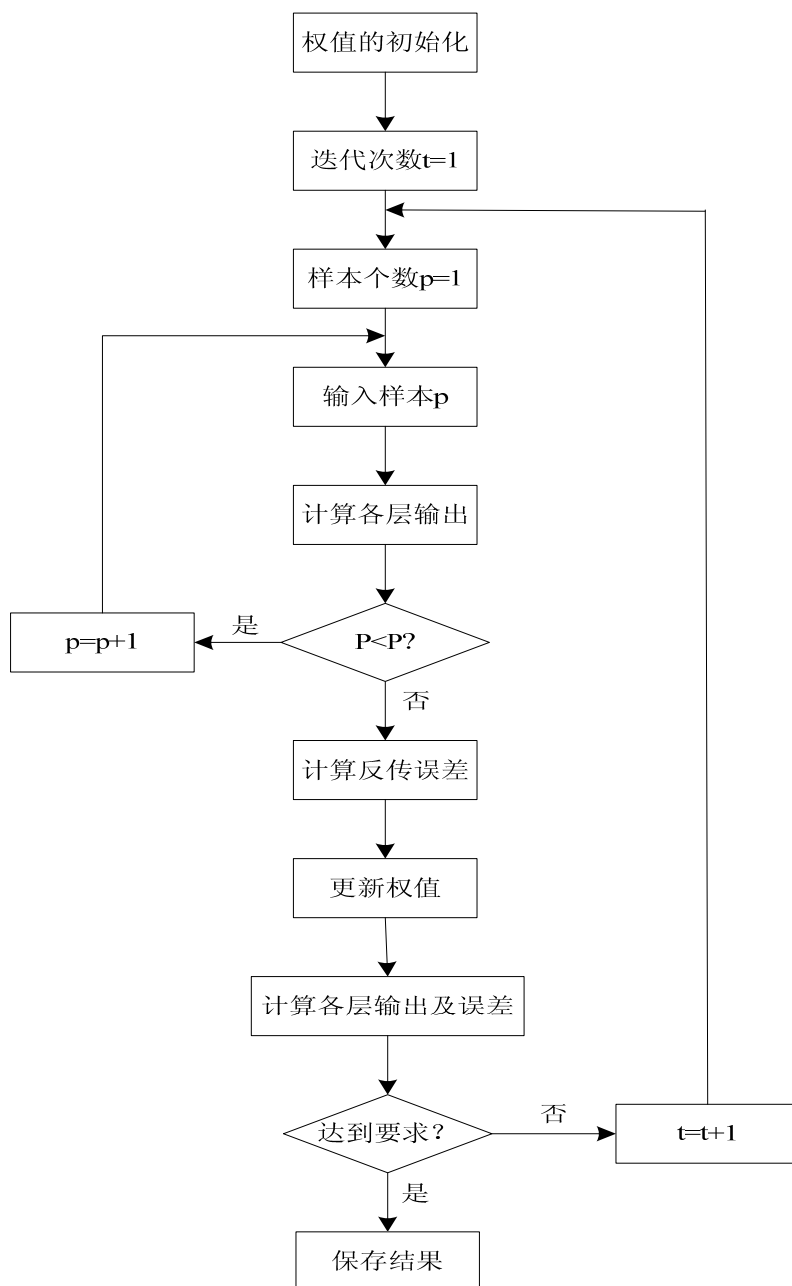


图 2.6 BP 算法流程图

2.4 本章小结

本章首先对神经网络发展的历史做了简要的介绍，并对人工神经网络的几个模型和训练学习算法做了详细的论述，着重介绍了 BP 神经网络的基本结构，并对反向传播误差的 BP 学习算法做了详细的描述。

第三章 BP 神经网络在短时交通流量预测中的应用

交通流量预测是指以施工、道路网的建设对交通的影响以及以给道路用户提高服务为宗旨的，依此进行的对交通流量、拥挤情况、行程安排等情况。过去的交通流量预测大都是依靠交通管理者的经验来进行的，但随着科学技术的发展以及计算机技术的普及，目前以及未来的预测越来越依靠计算机采取新的智能算法来进行。神经网络作为一种新的数学建模方法，因其强大的识别非线性系统的能力而被较多的用于预测系统中。本章的内容主要讨论怎样用 BP 神经网络构造神经网络预测模型。

3.1 基于 BP 神经网络交通流量预测模型的建立

3.1.1 输入层数据的选择

短时交通流量是一组复杂的非线性数据的集合，但在时空上是一个渐进具有规律性的过程。本文以城市典型交叉路口的交通流量为研究对象，各个相邻的路段之间必然存在着一定的相关性，同时各个路段每天的交通流量在时间上来说具有分布的统计规律性，一般可以从时间上和空间上进行分析。从时间上来说，路段每天的高峰和低峰的分布具有一定的平稳性，一般的高峰有早高峰和晚高峰两种，同时路口的流量也与该路段前几个时间段的流量有关。从空间的角度来看，路段出口的流量也必然受到上游路段交通流量数据的影响^[21]。

根据上述的特点，考虑到输出交通流量在空间和时间上的变化规律，输入信息在空间上来自相邻上下游路段的交通流量数据，时间上选取本路口前几个时刻段的数据流量，把这些检测点的数据作为输入变量。图 3.1 为本文研究的典型交叉路口示意图， q_1 、 q_2 、 q_3 、以及 Q_t 分别表示了上游交叉口 t 时刻北方位、西方位、南方位、以及下游路口的交通流量。

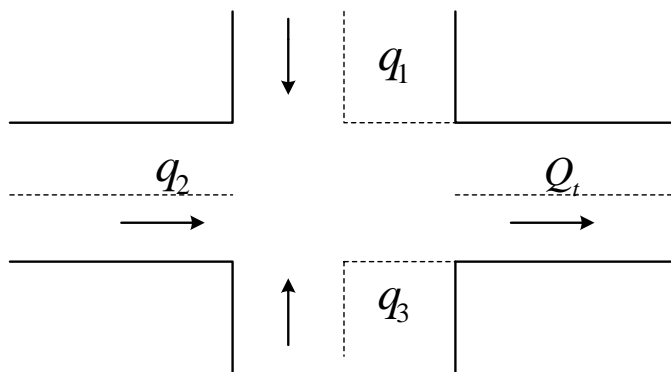


图 3.1 典型交叉路口交通流量示意图

从图中我们可以看出, t 时刻下游的交通流量 Q_t 显然有三部分组成, q_1 中的左转流量、 q_2 中的直行流量以及 q_3 中的右转流量, 每个方向中可能有少量反方向的车辆, 但数量相比下很小, 这里可忽略不计。因此在 t 时刻 Q_t 主要由 q_1, q_2, q_3 所组成, 可见从空间上来看, 下游路口的流量与上游不同方向路口之间的必然相关性。同时在预测下游该路段的流量时, 该路段的前几个时刻的流量 (Q_{t-1}, Q_{t-2}, \dots) 也有着时间上的联系。因此这就确认了预测下游路段流量所需要的输入数据流量, 这样就可以利用路口的前几个时间段的交通流量数据以及该路口上游路口的流量数据来完成对指定路口数据流量的预测, 即: $Q_{t+1}=f(q_1, q_2, q_3, Q_t, Q_{t-1}, Q_{t-2}, \dots)$ 。在确定了基本的相关要素后, 可以考虑不同的输入方案作为神经网络的输入, 从中选择比较理想的方案。在本文中, 考虑到神经网络的结构, 太多的输入量可能造成网络的复杂, 运算时间较长以及降低了网络的学习效率, 采用了 $(q_1, q_2, q_3, Q_t, Q_{t-1}, Q_{t-2}, Q_{t-3})$ 作为输入量来进行预测, 即采用了上游的路口流量以及本段路口的前三个时间段和当前时刻的流量总共 7 个单元来预测下一个时刻的流量。

现在对采集到的预测路口的交通流量数据进行可预测性分析, 每隔 15 分钟记录一次该时间内的交通流量, 总共记录一天 92 组的交通流量数据。

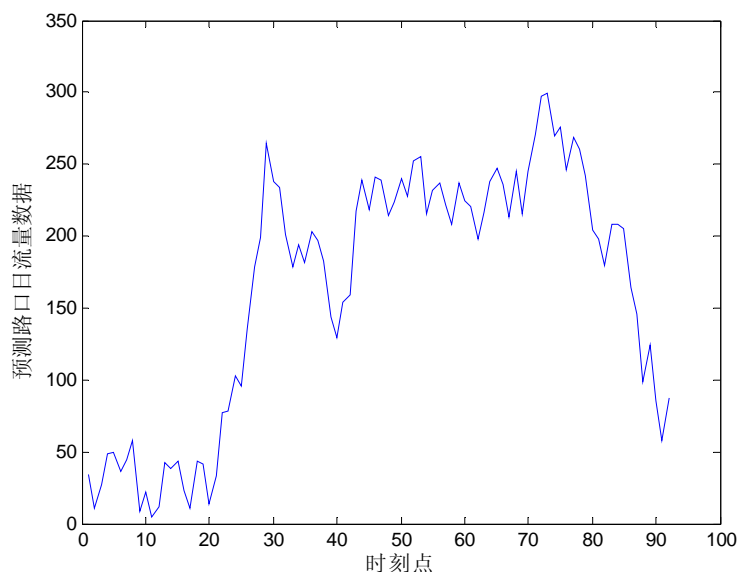


图 3.2 预测路口日原始流量数据

从图 3.2 可以看出预测路口各个时段的数据都完整地采样到, 没有遗漏的数据, 也没用异常变化的数据出现, 说明采样到的数据完整性较好。所采样到的数据不用进行修复和去噪处理, 可以用于预测。接下来利用功率谱的分析方法对采集到的数据进行可预测性分析。对于时间序列的功率谱图情况, 有以下结论:

- (1) 谱图中如果具有单个或多个尖峰, 则对应于周期或拟周期运动。

(2) 谱图中不存在明显的峰值或峰值连成一片, 则对应于混沌。

对流量数据 $x_1, x_2 \cdots x_{92}$ 根据傅里叶变换公式:

$$a_k = \frac{1}{N} \sum_{i=1}^N x_i \cos\left(\frac{i\pi k}{N}\right) \quad (3.1)$$

$$b_k = \frac{1}{N} \sum_{i=1}^N x_i \sin\left(\frac{i\pi k}{N}\right) \quad (3.2)$$

之后计算 $p'_k = a_k^2 + b_k^2$, 由多组 $\{x_i\}$ 计算得到一批 $\{p'_k\}$, 根据公式 $p(i) = 20 * \ln(p(i))$, 求得功率谱, 如下图所示:

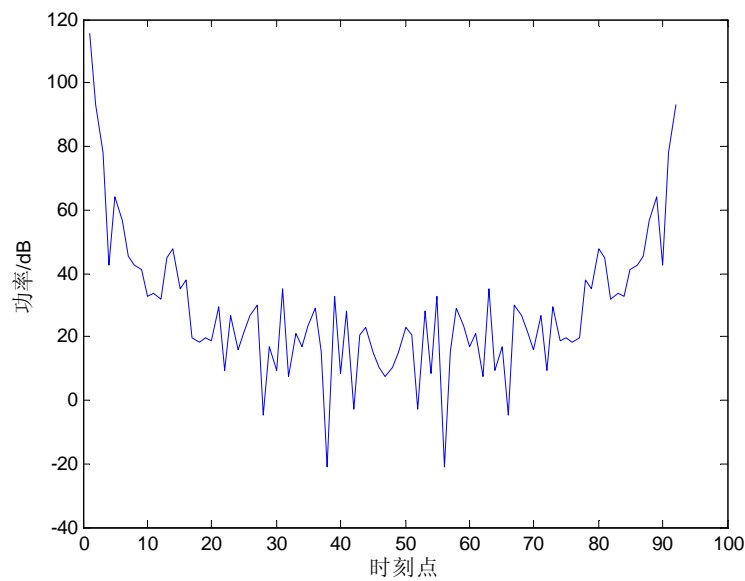


图 3.3 预测路口预测日交通流量功率谱

由图可知: 预测日的交通流量功率谱结构很复杂, 不存在单个或多个峰值; 系统呈现了一定的混乱随机状态, 可用于短时交通流量预测。

3.1.2 BP 神经网络预测模型分析

BP 网络学习算法对本身网络结构参数的选取非常敏感, 不同的网络结构针对不同的实际问题有着很大的差别。一般而言, 网络本身的结构越复杂, 处理非线性问题的能力就会越强, 但这也使得训练时间变长, 相反如果网络结构过于简单, 那么网络训练将难于收敛或者不收敛。

神经网络的拓扑结构由网络的输入层、隐含层以及输出层和神经元组成, 输入和输出的神经元主要根据需求解的问题以及数据的表示方式决定, 因此在 BP 网络结构设计中, 重

点在于隐含层层数以及隐含层神经元个数的确定。同时在预测模型中，对数据的预处理以及传输函数的选取也对解决分析问题起着关键的作用。

(1) BP 网络层数的确定

如上所描述，神经网络的输入输出层数目是由需要解决的实际问题以及数据表示方式综合决定，本文所要解决的是关于短时交通流量的预测问题，由上一节输入层数据的分析我们选取 7 个神经元作为网络的输入层，因为要预测下游路口某时刻的交通流量，因此输出的神经元数目取 1 个即要预测的路口流量就可。

对于 BP 神经网络而言，有一个很重要的定理存在，就是在闭区间内，任何一个连续的函数都可以用单隐层的 BP 神经网络来逼近，再由上面的分析并考虑实际的情况，如果将隐层数目取两层或以上，将会使网络的结构变得更加复杂，训练时间自然也会大大增加，因此本文在交通流量预测中隐层的数目就采用一层隐层。

隐层神经元数目的选取一直是一个复杂的问题，往往是根据实际的问题由设计者根据经验以及多次的实验来确定，因而也不存在一个合理的数学表达式来描述它。隐层单元的数目一般与求解的问题、输入输出单元的数目都有着直接的联系。隐层单元的数目过多会导致学习时间的延长、预测结果也不一定最好，同时也可能导致容错性较差、对以前没看过的样本识别力变差，这意味着存在着一个最为合适的隐层单元数。通过前人的经验总结，一般可以用以下三个公式来作为选择隐层单元数时的参考公式。

$$n_1 = \log_2 n \quad (3.3)$$

其中， n 为输入网络神经元数。

$$n_1 = \sqrt{n+m} + a \quad (3.4)$$

其中， m 为网络输出神经元个数， n 为输入网络神经元数， a 为[1,10]之间随机的常数。

$$\sum_{i=0}^n C_{n_i}^i > k \quad (3.5)$$

其中， n_1 为隐层单元数， k 为样本的数目， n 为输入网络神经元数。若 $n_1 < i$ ， $C_{n_i}^i = 0$ 。

还有一种方法来确定隐层神经元的数目。首先使用可变的隐层单元数，或者使用足够多的隐层单元，通过训练踢出不起作用的隐单元，直到不能收缩为止。同样也可以反过来，一开始使用较少的神经元，学习到一定次数后再逐步增加隐层单元的数目，一直到达到比较合理的隐层单元数目为止。

本文根据要解决的问题的复杂程度以及确定好的神经网络输入输出单元数，综合了前人

的经验公式，选取 14 个隐层单元数目。至此也就确定了本文预测模型所需的神经网络拓扑结构，建立了一个 7-14-1 的预测网络结构，即输入层有 7 个节点，隐含层有 14 个节点，输出层有 1 个节点，得到的网络输出即为预测时刻的预测值。

(2) 数据的预处理

当将实际采集到的数据输入神经网络时，有的时候不是直接使用这些数据，而是要进行必要的数据处理。对实际采集的数据进行处理主要有两方面的原因：

- 1) 神经网络隐含层所选取激励函数的作用范围，要避免神经元的饱和。
- 2) 数据本身特点也会影响模型的适应度。

鉴于以上两个原因，一般对输入数据的预处理也可以分为两种：一种就是归一化处理，将数据选取到神经网络隐层激励函数的作用范围内；另一种是对数据进行变换，例如进行平滑处理，数据的修补等，来修正去除一些不规则的数据，产生典型的信号数据。

本文根据实际情况在下面的仿真分析中均采用将数据归一化的方法。一般 BP 神经网络在处理函数逼近以及预测等非线性数据时会采用 Sigmoid 类（单极性和双极性）函数，由于本文中要处理的是交通流量数据，该类数据都是正实常数，故在此将数据处理为区间[0,1]之间的数据，归一化方法有多种方式，这里采用如下公式进行变化：

$$\hat{x} = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \quad (3.6)$$

式中， x 为原始的交通流量数据， x_{\min} 和 x_{\max} 分别为原始数据中的最小值和最大值， \hat{x} 为归一化后的数据。对数据进行训练和预测后，在输出层采用相对应的数据变换方式得到原来的真实值，与其对应的数据逆变式为：

$$x = \hat{x} * (x_{\max} - x_{\min}) + x_{\min} \quad (3.7)$$

(3) 激励函数的选取

在本文 2.2.1 节中已经对神经元常用的激励函数 $f(u)$ 进行了分析，激励函数 $f(u)$ 的作用是将非线性函数无限域的空间转换到有限域的空间输出，根据不同的实际应用选取合适的激励函数，通常选取 Sigmoid 类函数。该类函数所表现出来的非线性特性使得 BP 神经网络的训练建立了从输入到输出的非线性映射，能够很好的表达交通流量数据的客观现象。

在对数据进行归一化处理后，对 BP 网络的隐层采用如图 2.4 所示的双极性函数，该图形是以 (0,0) 为对称中心，映射到的输出范围为 (-1,1)，根据实际情况在输出层采用如图 2.3 所示的单极性函数，该图形是以 (0,0) 为对称中心，最后映射到 (0,1) 区间。

(4) 网络初始值的选取

由于系统的非线性，网络初始权值和阈值的选取对于网络的学习能否达到全局最小值以及是否能够收敛都有着很大的关系，这其中有一个重要的要求：初始权值在输入累加的时候使每个神经元的状态接近于零，若初始权值选取过大，加权后的输入值就会落到激励函数的饱和区，使得 $f'(u)$ 变得非常小，而在权值修正的过程中，因为 δ 与 $f'(u)$ 成正比，所以 δ 就会趋向于零，致使 $\delta \Delta w \rightarrow 0$ ，那么网络的调节功能就会停顿^[22]。所以为了保证加权后神经元的输出值接近零，神经元的取值的都取比较小的随机值，一般在 $(-1, 1)$ 之间，同样对输入样本也做归一化处理，使得样本中比较大的输入依然能落在传递函数大的地方。

(5) 学习速率的选取

学习速率决定着每次迭代过程中权值的变化量。一般来说，学习速率越大，相应的训练时间就会越少，但是，如果学习速率过大会造成学习过程的不稳定；小的学习速率学习速度过慢，可能会导致收敛速度过慢，不过这样可以保证网络的稳定性，因此一般情况下会选择较小的学习速率，学习速率的范围一般在 $0.01-0.8$ 之间。每一个具体的网络都会存在与之相对应的合适的学习速率，但对于复杂的非线性网络，在不同的学习阶段可能要不同的学习速率，因此在这种情况下，合适的方法是采用自适应的学习速率减少找寻学习速率的次数和训练的时间，从而使得预测系统在不同的阶段有不同大小的学习速率。

3.2 基于 BP 神经网络预测模型仿真实例

通过上面两小节分析，本文建立了基于 BP 神经网络的交通流量预测模型，为了验证分析，本文以某城市图 3.1 所示的交叉路口 4 天的交通流量数据，每隔 15 分钟记录一次该时间内的交通流量，一共记录 4 组 384 个时间点数据，用前 3 天的数据作为 BP 神经网络的训练样本。用训练好的神经网络模型预测第 4 天的交通流量。

3.2.1 传统 BP 算法的不足及改进

通过 3.1 节的分析，基于 BP 神经网络在城市典型路口交通流量预测算法的具体实现步骤如下：

(1) 确定神经网络各层神经元的个数：由上述分析，本文确定的神经网络输入层为 7，隐含层由经验公式选取为 14，输出层即要预测的交通路口流量的神经元个数取 1；

(2) 输入采集到的训练样本和测试样本，这里要对所有采集的样本按照式 (3.6) 进行归一化处理；

(3) 选择激励函数即网络隐含层和输出层的传递函数以及训练函数、训练迭代次数、最大允许误差等。这里分别选取 `tansig` 和 `logsig` 作为传递函数；

(4) 利用神经网络对训练样本进行训练，再由测试样本进行仿真输出，计算出预测值和真实值之间的各种误差。

根据以上的分析，在 MATLAB 环境中，BP 神经网络的交通流量预测的流程图如下图 3.4 所示：

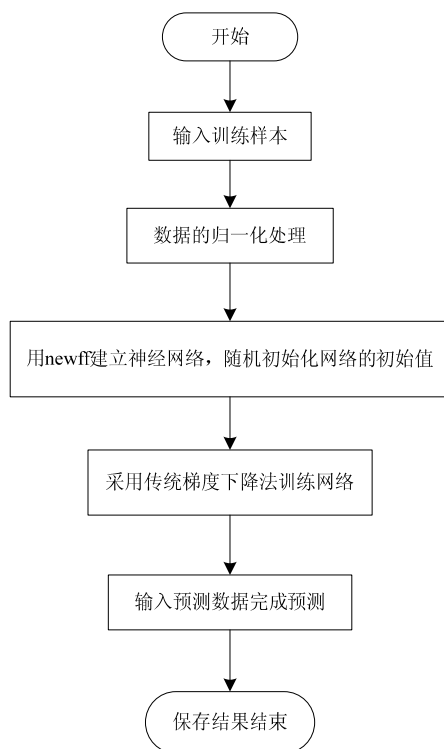


图 3.4 BP 神经网络交通流量预测流程图

在这里使用传统的 BP 梯度下降法来训练网络，设定最大允许的误差为 1×10^{-4} ，最大的迭代次数为 1000，图 3.5 为传统梯度下降法的训练误差图。

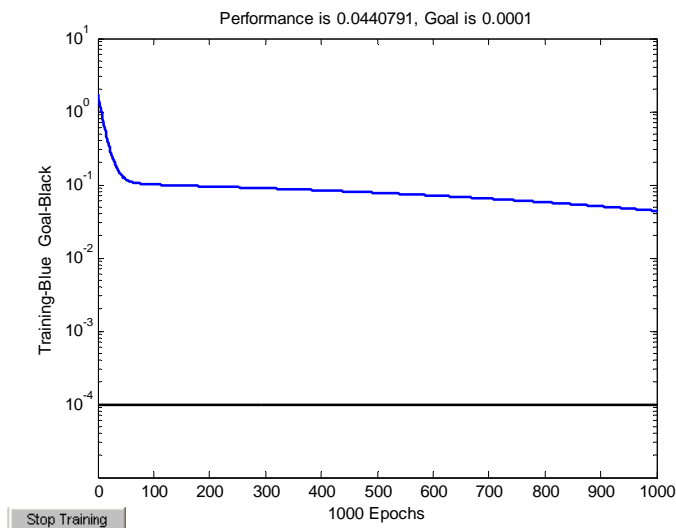


图 3.5 梯度下降法训练误差

从训练误差图中我们可以看出，传统的 BP 梯度下降算法在前期的学习有较快的收敛速度，到了后期收敛速度越来越慢，以至于到了一定阶段学习能力几乎下降为零。这主要是因为梯度下降法的训练从某一个起点开始到达误差的最小值，从而不同的起始点就可能导致不同的极小值，就有可能得到不同的最优解。同时梯度下降法还有着“喜新厌旧”的现象，就是在训练的过程中有遗忘旧样本的趋势，这使得训练后期样本的多样性缺失，学习能力趋向于静止。

评价神经网络学习算法的标准有很多种。例如学习的时间，训练所到的精度等，总的来说就是能否在大范围的变化时到达预定的精度。通常的观点是 BP 学习算法有易陷入局部极小点且收敛速度较慢，训练的网络泛化能力不够等缺点。前人在提高算法的收敛速度以及避免陷入局部极小点方面进行了研究并提出很多方法。现将介绍一些主要的方法。

(1) 附加动量法

附加动量法是 Rumelhart 等于 1986 年提出的一种改进 BP 网络训练时间的方法，且保证学习过程的稳定性。该方法在每一次权值的调节上加了一个正比于上次加权变化量的值，这就要求带有记忆功能，在每一次调节后记住该调节量以便下一次加权调节中使用。具体调节公式如下：

$$w(t+1) = w(t) - (1-\alpha)\eta \frac{\partial E_T}{\partial w(t)} + \alpha \Delta w(t) \quad (3.8)$$

式中， $\Delta w(t) = w(t) - w(t-1)$ ； α 为调节的动量因子，一般为 0.95。从式中可以看出在权值修正过程中加入了上一次权值修正的方向记忆。

附加动量法同时考虑了误差在梯度上的作用和误差曲面变化上的影响，它的作用是避免

网络陷入浅的局部极小点，利用附加动量值来滑过这些值。它的实质是通过动量因子来传递最后一次权值变化的影响，并且增加动量有促使权值调节向误差曲面底部的平均方向变化的作用，当网络权值进入误差曲面底部平坦区时有 $\Delta w(t+1) \approx \Delta w(t)$ ，即梯度将会很小，从而防止了 $\Delta w(t+1) = 0$ 的出现，能使网络跳出局部极小值。由附加动量法设计原则，当权值的修正在误差中导致太大的增长结果时新的权值应该不被采用，同时应该使动量停止下来以避免网络进入较大的误差曲面；当新的误差变化率超过了事先预定好的最大误差变化率时，也应该取消当前计算的权值变化。由这些原则使得在附加动量法训练的过程中要有判定条件加以判定，以正确使用修正公式，其判定条件如下：

$$\alpha = \begin{cases} 0, & SSE(t) > 1.04SSE(t-1) \\ 0.95, & SSE(t) < SSE(t-1) \\ \alpha, & \text{其他} \end{cases} \quad (3.9)$$

式中， SSE 为网络的输出误差平方和。

在附加动量的作用下，当网络的训练陷于局部极小值后，能产生继续向前的正向斜率运动来跳出较浅的峰值，最终进入全局极小值。当在附加动量的作用下达到一定的高度后（即产生了 $SSE(t) > 1.04SSE(t-1)$ ）自动返回，这样来回摆动最终停留在最小点上。但该方法也有缺点，它对初始权值的选取有要求，必须使其取值在误差曲线上的位置所在误差下降方向与误差最小值方向一致。如果下降方向不一致，附加动量法就会失效，使得训练结果再次落入局部极小值。

（2）牛顿法

常规的 BP 梯度下降法在权值修正时只用了一阶导数的信息，即误差函数对权值的梯度，如果采用二阶导数进行权值调整（牛顿法）可以提高收敛速度。假设网络权值修正的目标是极小化误差函数 $E(w)$ ，当前网络权值为 $w(t)$ ，权值修正量为 $\Delta w(t)$ ，则下一时刻的权值 $w(t+1) = w(t) + \Delta w(t)$ 。对 $E(w(t+1))$ 进行二阶泰勒展开有：

$$E(w(t+1)) \approx E(w(t)) + g^T(t)\Delta w(t) + \frac{1}{2}\Delta w^T(t)A^T(t)\Delta w(t) \quad (3.10)$$

式中， $g(t)$ 是 $E(w)$ 的梯度向量；方阵 $A(t)$ 为 $E(w)$ 的 Hessian 阵（二阶导数），即元素值为 $E(w)$

对各权值的二阶导数， $A_{ij}(t) = \frac{\partial^2 E(w)}{\partial w_i \partial w_j}$ 。权值修正后的误差函数的变化量为：

$$\Delta E(t) = g^T(t)\Delta w(t) + \frac{1}{2}\Delta w^T(t)A^T(t)\Delta w(t) \quad (3.11)$$

希望通过变动 $\Delta w(t)$ 使式 (3.10) 达到最小, 显然当满足:

$$\Delta w(t) = -A^{-1}(t)g(t) \quad (3.12)$$

$\Delta E(t)$ 取得最小值。牛顿算法收敛速度快, 但计算量太大不易实现。

(3) 拟牛顿法

牛顿法的主要思想是基于二阶泰勒级数的快速优化方法, 其基本原理已如上述介绍, 为了克服牛顿法的计算量太大的弊端, 人们提出在算法中使用 Hessian 矩阵的近似值作为替代, 即拟牛顿法, 修正值被看成梯度的函数。常见的拟牛顿算法有 DFP(Davidon-Fletcher-Powell) 算法和 BFGS(Broyden-Fletcher-Goldfarb-Shanno) 算法, 目前 BFGS 算法被公认为最好的拟牛顿算法。

(4) Levenberg-Marquard(LM)法

LM 算法与拟牛顿法一样, 为了避免计算 Hessian 矩阵而采用近似的二阶训练速率代替, Hessian 矩阵的近似为: $H = J^T J$, 梯度近似为 $g = J^T e$ 。则 LM 算法的权值调整公式如下:

$$x(k+1) = x(k) - [J^T J + \mu I]^{-1} J^T e \quad (3.13)$$

式中, J 为误差对权值微分的雅克比矩阵, e 为误差向量, I 为单位阵, μ 为常数, 其中 μ 为自适应调整系数, $\mu \rightarrow \infty$ 时上式为最速下降法, $\mu \rightarrow 0$ 时上式为牛顿法, 即为近似的 Hessian 矩阵。

除了上述四个常用的改进的 BP 学习算法, 还有不少学者提出的改进方法。比如弹性 BP 算法、改进误差函数法、样条权函数算法和代数算法以及竞争 BP 算法等。这些方案都是以加快网络的收敛速度和避免算法陷入局部极小点为目的。对于一个具体的问题, 采用哪种学习算法训练网络, 使得网络收敛快、训练误差小, 这本身也是难以确定的。这是因为问题的复杂程度、训练样本集的个数以及网络参数的选取和网络自身的用途等都影响着算法的选取, 本文也是根据前人总结的经验和需要解决的问题来选取合适的算法。

LM 算法在处理函数逼近问题上收敛速度快, 算法精度较高。如果要处理的问题是非线性函数逼近或者是预测问题, 使用 LM 算法具有比较好的效果。另外, LM 算法在处理模式识别问题上效果较差。

拟牛顿算法和 LM 算法类似, 所需存储空间小于 LM 算法, 但拟牛顿算法与牛顿算法一样, 算法的运算量随着网络规模大小成几何增长, 因此对于规模稍大的网络使用附加动量法或者任何其他改进梯度算法可能要好, 规模较小的用拟牛顿算法中 BFGS 则更有效。而对于

附加动量法对初始值的选取很重要，建议多用几次初始值粗略的训练几次才能找到合适的初始位置。

本文要进行的是对交通流量的预测，隶属于非线性函数逼近的问题，再通过对几种常用的改进 BP 算法的分析和总结前人的实践经验后，本文采用 LM 训练函数 `trainlm` 作为对网络最终的训练算法。

3.2.2 基于 LM 算法 BP 神经网络的交通流量预测结果及分析

(1) 预测性能评价指标

交通流量预测精度的好坏对智能交通系统的控制有着直接的影响，通过对不同预测性能指标的评价可以得出预测精度的高低。一般评价预测性能是通过实际输出与预测输出之间的差异来判断，但是它们之间的误差表达方式有很多种，而且其中某一种表达方式不能很好的判断该预测模型的好坏。从实践中我们知道，一种模型的最大绝对误差低于另外一种，而其平均绝对误差可能高于另一种，所以我们应该通过不同的误差表达式来对预测模型的优缺点进行综合评价，下面介绍几种常用的误差表达式：

1) 均方误差 (Mean Square Error)

$$MSE = \frac{1}{N} \sqrt{\sum_{i=1}^N (x_i - x_i')^2} \quad (3.14)$$

2) 平均绝对误差 (Mean Absolute Error)

$$MAE = \frac{1}{N} \sum_{i=1}^N |x_i - x_i'| \quad (3.15)$$

3) 平均绝对百分比误差 (Mean Absolute Percent Error)

$$MAPE = \frac{1}{N} \sum_{i=1}^N \left| \frac{x_i - x_i'}{x_i} \right| \quad (3.16)$$

4) 相对误差 (Relative Error)

$$RERR = \frac{x_i' - x_i}{x_i} \quad (3.17)$$

这些公式中，交通流量时间序列为 x_1, x_2, \dots, x_N ，预测输出和实际输出为 x_i' 和 x_i 。

(2) 基于 LM-BP 神经网络的短时交通流量预测流程

在 MATLAB 环境中，下图 3.6 为 LM 算法训练 BP 神经网络的短时交通流量预测流程图：

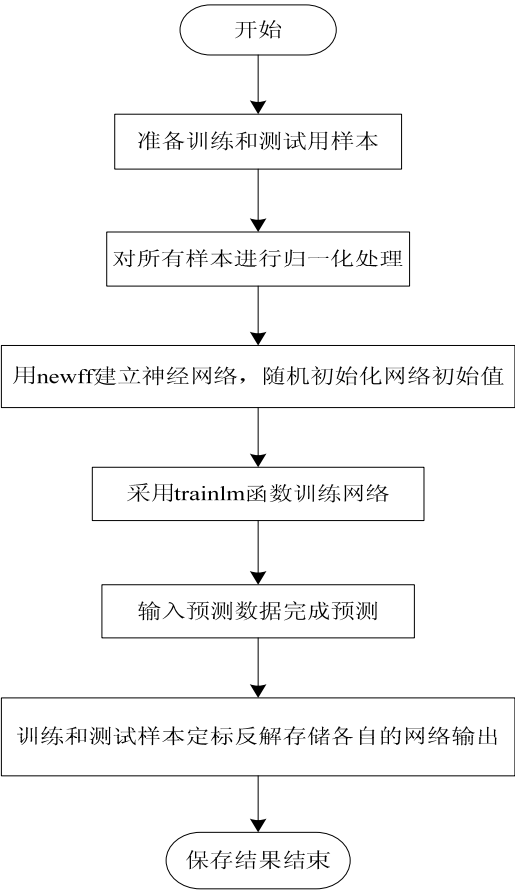


图 3.6 LM 算法训练 BP 神经网络预测流程图

(3) 预测结果及分析

采用 LM 算法训练 BP 神经网络进行交通流量预测的环境与传统 BP 算法预测环境相同，依然设定最大允许的误差为 1×10^{-4} ，最大的迭代次数为 1000，图 3.7 为 LM 算法的训练误差图。

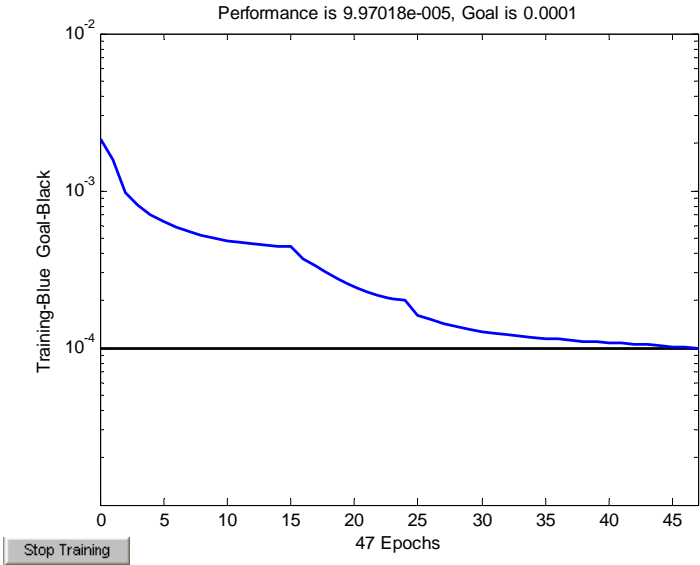


图 3.7 LM 算法训练误差图

从图中可以看出，LM 算法在 47 步后就能达到训练目标，与图 3.5 相比可以看出 LM 训练算法的优越性，传统 BP 训练算法在 1000 步后仍然不能达到训练目标。

接下来作出基于 LM 算法 BP 神经网络的短时交通流量实际结果与预测结果的对比曲线，如图 3.8 所示，为了便于分析比较，表 3.1 列出了交通流量高峰期 8:00~11:00 和 17:00~20:00 时间段的预测对比情况。从图中可以看出，LM 算法训练 BP 神经网络的交通流量预测的结果与实际的交通流量有大致的变化趋势，也验证了采用 LM 算法进行训练的可行性。

仿真输出的均方误差(MSE)为 10.7482；平均绝对误差(MAE)为 28.0769；平均绝对百分比误差(MAPE)为 15.356%；从表 3.1 可以得出，在交通流量高峰期相对误差(RERR)最大的为 28%；最小的为-0.4%；平均相对误差为 12.7%；另外在所有的预测数据中 54.5%的预测相对误差在 10%以内；28.5%的预测误差在 5%以内。

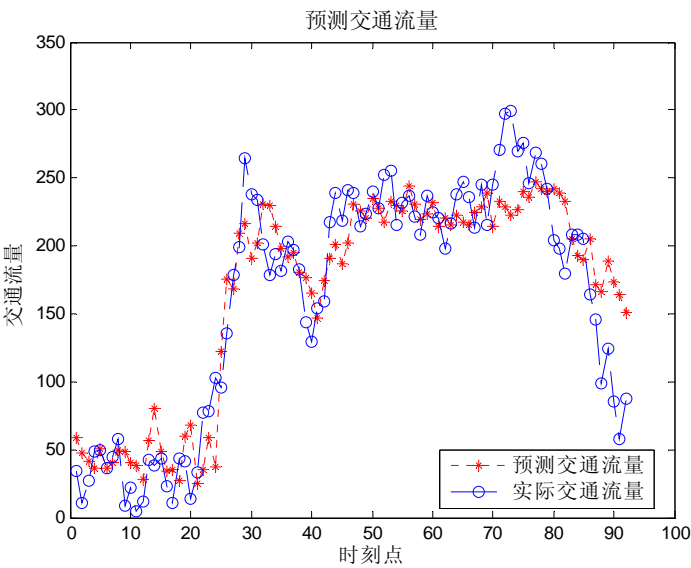


图 3.8 基于 LM-BP 预测网络的预测曲线图

表 3.1 基于 LM-BP 预测网络交通流量预测结果

时间	预测交通流量	真实交通流量	相对误差
8:00	230	179	0.28
8:15	214	194	0.1
8:30	198	182	0.08
8:45	192	203	-0.05
9:00	195	197	-0.01
9:15	181	183	-0.01

9:30	177	144	0.23
9:45	165	129	0.28
10:00	147	154	-0.05
10:15	175	159	0.1
10:30	191	217	-0.12
10:45	202	239	-0.15
11:00	187	218	-0.14
17:00	229	215	0.07
17:15	239	245	-0.02
17:30	215	271	-0.2
17:45	233	297	-0.21
18:00	228	299	-0.24
18:15	222	270	-0.18
18:30	226	276	-0.18
18:45	240	246	-0.02
19:00	236	269	-0.12
19:15	247	260	-0.05
19:30	241	242	-0.004
19:45	240	204	0.18
20:00	242	198	0.22

通过对网络学习速率以及对 BP 学习算法的改进，在此基础上建立的 LM-BP 神经网络的交通流量预测模型，通过仿真实验验证了该方法的可行性。但这些改进方法都会受到网络初始参数的影响，且算法都是以梯度下降法为前提，不同的起始点会导致不同的最优值，因此算法易陷入局部极小值且不稳定。这里可以考虑结合智能优化算法和神经网络的优点，利用智能算法优化网络的初始权值和阈值，用 LM 算法对优化后的网络进行进一步的训练，此时网络仅在已接近最优解的基础上训练，从而可以有效提高网络的训练速度和精度。在下章将介绍使用智能优化算法对 BP 网络的初始参数进行优化。

3.3 本章小结

本章详细介绍了 BP 神经网络进行短时交通流量预测的方法，对网络输入数据选择、数据的可预测性分析、数据的处理、激励函数的选取以及预测网络结构的确定作了详细的分析。通过仿真指出了传统 BP 梯度下降法的不足，并通过分析比较最终采用 LM 算法训练 BP 神经网络，通过仿真实验验证了该方法的可行性，指出了改进 BP 算法受到网络初始参数的影响，且算法都是以梯度下降法为前提，算法易陷入局部极小值。

第四章 智能算法优化 BP 神经网络在交通流量预测中的应用

智能优化算法主要是从自然界出发,通过模拟自然界中各种群寻优的现象出发,以群体智能为基础求解复杂问题最优解的一类数学优化算法,目前主要的智能优化算法有粒子群算法、遗传算法、蚁群算法和模拟退火算法等^[23-25],本章主要介绍粒子群算法和遗传算法在优化 BP 神经网络中的应用。

4.1 粒子群算法及其改进

4.1.1 基本粒子群算法

粒子群算法是基于群体智能寻优的演化算法,其思想来源于人工生命和演化理论。通过对鸟群的飞行研究发现,当一个区域只有一块食物且一群鸟在这个区域内随机进行搜索,那么找到这块食物最快速有效的方法就是搜索当前离这块食物最近的鸟群周围区域。鸟群当前只是追踪着它们有限数量的邻居,但最终搜索的结果是整个鸟群在某个中心的控制下向目标移动,也就是说复杂的行为表现在简单规则的相互作用上^[26-28]。

粒子群算法就是从自然界中种群群体活动的特性中得到启发,并将这个思想用于求解优化问题中,1995 年 Eberhart 和 Kennedy 利用这一特性提出了粒子群优化算法(PSO)。在粒子群算法中,每个优化问题的潜在解可以看成搜索空间中的一个点,称为“粒子”。粒子以一定的速度在空间中飞行,具体的速度以本身的经验和其他粒子的经验进行动态的调整。所有的粒子都有一个被目标函数所确定的适应值(fitness value),以及自己到目前为止所发现的最好位置(particle best,pbest)和当前位置,这个位置就是目前粒子自己的经验。同时,粒子还明确到目前为止整个种群中所有的粒子发现的最好位置(global best,gbest),gbest 为 pbest 中的最好值,这个值为其他粒子的飞行经验。整个优化搜索的过程就是由一群随机初始化的粒子以迭代的方式不断更新自身的速度和位置,最终找到最优解和最优位置,粒子群算法有着迭代计算和群体智能的双重特点。

在基本粒子群优化算法中,由 N 个粒子组成一个群落,其中每个粒子的位置代表了优化问题在 M 维搜索空间中的潜在解,算法通过粒子之间的相互作用对搜索空间进行寻优从而发现最优解。

假设在 M 维的搜索空间中有 N 个粒子组成群落,其中第 n 个粒子表示为 M 维向量

$x_n = (x_{n1}, x_{n2} \dots x_{nM})^T$, $n=1, 2 \dots N$, 即当前第 n 个粒子在空间中的位置。该位置是优化问题中的一个潜在解, 将 x_n 带入目标函数求出适应值, 判断适应值的大小来衡量其是否为最优解。该粒子的速度也为 M 维的向量: $v_n = (v_{n1}, v_{n2} \dots v_{nm})^T$ 。记第 n 个粒子在第 k 次迭代时的最优位置为: $pb_n = (pb_{n1}, pb_{n2} \dots pb_{nm})^T$, 整个种群此时的最优位置为: $gb = (gb_1, gb_2 \dots gb_M)^T$, 粒子在确定了上述的两个位置后, 就会依照下面两个公式来更新自身的速度和位置:

$$v_{nm}^{k+1} = v_{nm}^k + c_1 r_1^k (pb_{nm}^k - x_{nm}^k) + c_2 r_2^k (gb_m^k - x_{nm}^k) \quad (4.1)$$

$$x_{nm}^{k+1} = x_{nm}^k + v_{nm}^{k+1} \quad (4.2)$$

式中, $m=1, 2 \dots M$; c_1 和 c_2 为学习因子或加速常数; r_1 、 r_2 为(0,1)之间的随机数; k 为当前的迭代次数; v_{nm}^k 和 x_{nm}^k 分别为第 n 个粒子在第 k 次迭代中第 m 维的速度和位置; pb_{nm}^k 和 gb_m^k 分别为第 n 个粒子在第 m 维的个体极值和群体在第 m 中的全局极值。从粒子的迭代过程中可以看出, c_1 调节的是粒子向自身最好位置飞行的步长, c_2 调节的是向群体最好位置飞行的步长。为了降低粒子飞行过程中离开搜索空间的可能性, 将搜索空间 v_{nm} 限定与一定范围, 即 $v_{nm} \in [-v_{\max}, v_{\max}]$, 通常设定 $v_{\max} = kx_{\max}$, $0 \leq k \leq 1$ 。

基本粒子群优化算法的优点在于不需要太多的参数, 易于操作, 但算法中少量的参数直接影响着算法的性能以及收敛性。由于理论方面的研究还不完善, 大部分参数的选取都是依靠前人的经验总结^[29-30], 主要的参数设置如下:

(1) 粒子数目。一般的粒子数取 20~40。粒子的数目越多, 搜索空间的范围就越大, 发现全局最优解就更容易, 但算法运行的时间就越长。

(2) 粒子长度。与神经网络的权值和阈值个数有关。本文是对 BP 神经网络进行优化, 粒子长度为:

$$n = D_1 \times D_2 + D_2 \times C + C \quad (4.3)$$

式中, D_1 、 D_2 、 C 分别为网络输入层、隐含层以及输出层神经元个数。

(3) 粒子最大速率。粒子最大速率决定了粒子在一次飞行中所迈出的最大步长, 速率过大会导致粒子飞出寻优范围内, 因此要限制粒子的飞行速率。一般设置 $v_{\max} = kx_{\max}$ 中的 $k=0.5$ 。

(4) 加速常数。学习因子 c_1 和 c_2 分别表示了个体认知和社会知识对粒子的影响度, 一般令 $c_1=c_2=2$, 也有认为 c_1 应与 c_2 不等。

(5) 算法终止条件。一般智能优化算法的终止条件可以设为达到最大迭代次数或者是设定某个误差准则。

(6) 适应度函数。粒子群算法中的适应度函数一般直接选取目标函数。根据本文要解决的问题可以选取根据训练样本的输入得到的输出与实际值计算的均方误差 (MSE) 作为评价每个粒子的适应度函数。

确定了基本粒子群算法的各个参数后, 基本粒子群算法的流程可如下图所示:

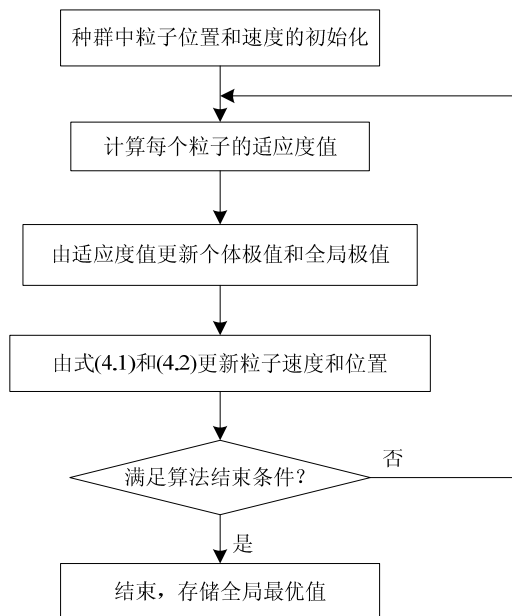


图 4.1 基本粒子群优化算法流程图

4.1.2 粒子群算法的缺陷及其改进

粒子群优化算法在对相对较小的群体中具有易于实现、需要调节的参数少、收敛速度快等优点, 对内存以及 CPU 的速度都没有很高的要求, 而且它并不依靠目标函数的梯度信息只需要函数值, 粒子群算法在解决全局优化问题上有着广泛的应用。

粒子群优化算法虽有各种优势, 但也存在着自身的局限性, 主要表现为:

(1) 粒子群算法是一种概率算法, 在数学上缺乏完整规范的理论基础, 难以证明它们的正确和可靠性, 在全局收敛工作上也缺乏研究。

(2) 粒子群算法的寻优过程同参数的选取有较大的关系, 通常对于不同的问题选取不同的参数。一般对参数选取的规律都有一个定性上的认识, 依据前人总结的经验。

(3) 粒子群算法往往有早熟收敛的问题, 在种群没到全局极小点时就聚焦在一点不动, 这个点往往是局部极小点, 也就是说早熟收敛不能保证算法已经收敛到全局最优。

(4) 粒子群优化算法在接近最优解时的收敛速度比较缓慢, 这是种群缺乏多样性的表现。大量粒子群的改进大都集中在避免算法陷入局部极小值、提高搜索精度上。

针对粒子群算法的缺点，研究者们提出了种种的改进。为了更好的控制粒子的探测和开发能力，这里的探测是指让粒子以较大的程度偏离原先的寻优轨迹，开发是指粒子在原先轨道中的细部搜索，Shi 等人提出将惯性权重^[31] w 引入式(4.1)中，形成如下的标准粒子群算法：

$$v_{nm}^{k+1} = wv_{nm}^k + c_1r_1^k(pb_{nm}^k - x_{nm}^k) + c_2r_1^k(gb_m^k - x_{nm}^k) \quad (4.4)$$

通过实验研究了惯性权重 w 的作用，较大的 w 有利于粒子跳出局部寻优而进行全局寻优；较小的 w 则有利于局部寻优^[32-33]。这样标准的粒子群算法中就有三个权重因子分别为：惯性权重 w 、加速常数 c_1 和 c_2 。在式（4.4）中，若 $c_1=c_2=0$ ，粒子就会以当前的速度飞行直到边界，这会使得搜索空间变的有限，很难得到最优解。若 $w=0$ ，那么粒子的速度只取决于当前和历史的最好位置，而本身没有记忆性。若 $c_1=0$ ，粒子不具备认知能力，就是“只有社会”的模型，这使得算法的收敛速度更加的快，但更容易陷入局部极小点。相反若 $c_2=0$ ，则粒子就是“只有认知”的模型，整个个体没有交互，很难找到最优解。

惯性权重的选取一直是标准粒子群算法中重要的研究工作。较大的惯性权重使得粒子具有较大的速度和探索能力，较小的惯性权重使得粒子具有较强的局部开发能力，因此惯性权重必须合理选取以平衡探索和开发能力。一般惯性权重分为固定和时变两种，固定的惯性权重使得算法一直具有相同的探索和开发能力，而时变的权重使得粒子在优化时期不同阶段具有不同的探索和开发能力。文献[34]提出了一种递减惯性权重的取值方法^[34]，惯性权重是迭代次数的函数，实验设置了 w 从 0.9 到 0.4 线性递减，使得算法在前期具有较大的全局探索能力，到了后期具有较好的局部搜索能力，并在文献中提出了惯性权重的变化公式如下：

$$w(k) = w_{\max} - \frac{w_{\max} - w_{\min}}{k_{\max}} \times k \quad (4.5)$$

式中， k 代表当前的迭代次数， k_{\max} 为最大的迭代次数， w_{\max} 、 w_{\min} 为最大和最小的权重因子。这种动态调节惯性权重的方法加快了收敛速度，提高了算法的性能。

在 Clerc 等人的研究中引入了收缩因子^[35]的概念，该方法描述了一种选择惯性权重 w 、加速系数 c_1 、 c_2 的方法。相关的方程式如下：

$$v_{nm}^{k+1} = K[v_{nm}^k + \varphi_1r_1^k(pb_{nm}^k - x_{nm}^k) + \varphi_2r_1^k(gb_m^k - x_{nm}^k)] \quad (4.6)$$

$$K = \frac{2}{2 - \varphi - \sqrt{\varphi^2 - 4\varphi}} \quad (4.7)$$

式中， $\varphi = \varphi_1 + \varphi_2$ ， $\varphi > 4$ 。其中 K 是一种受 c_1 和 c_2 限制的 w ，文献中计算得出 K 固定取值 0.73。

实验证明, 收缩因子的引入可以提高算法的收敛率, 在使用收缩因子时对算法进行设定, 如设参数 $v_{max}=x_{max}$, 这样在收敛性能和搜索性能方面都会有好的效果。

本文根据这两种方法的优点, 同时采用惯性权重法和收缩因子法的改进粒子群算法 (MPSO), 将粒子的更新公式变为如下所示:

$$v_{nm}^{k+1} = K(wv_{nm}^k + c_1r_1^k(pb_{nm}^k - x_{nm}^k) + c_2r_1^k(gb_m^k - x_{nm}^k)) \quad (4.8)$$

$$x_{nm}^{k+1} = x_{nm}^k + v_{nm}^{k+1} \quad (4.9)$$

在基本的粒子群算法流程中按照式(4.8)和式(4.9)来更新粒子的速度和位置。

4.1.3 MPSO-BP 神经网络混合算法

粒子群算法作为新兴的智能算法, 通过群体之间的合作和竞争来指导粒子的寻优, 收敛速度快、全局搜索能力强, 不需要依靠目标函数的梯度信息, 克服了 BP 神经网络学习算法的固有缺陷。将 PSO 算法与神经网络相结合, 考虑到了粒子群算法超强的全局搜索能力以及 LM 学习算法的快速局部收敛能力, 具体过程如下: 首先使用改进的粒子群算法训练预测网络的权值阈值组合, 直到满足算法结束条件, 在此基础上, 使用改进的 BP 学习算法 LM 算法对网络进行进一步的训练, 此时网络仅在接近网络最优解的基础上进行寻优, 可以有效提高网络的训练速度和精度。MPSO-BP 混合算法的具体流程如下:

- 1) 建立基于 BP 神经网络的短时交通流量预测模型, 确定网络拓扑结构并输入归一化后的训练样本;
- 2) 初始化种群的位置和速度, 确定粒子个数 N 、惯性权重 w 以及收缩因子 K 、加速常数 c_1 、 c_2 以及 gb 和 pb_n 的初始值;
- 3) 将种群映射为网络的权值和阈值, 作为 BP 网络的初始参数值, 由训练样本的输入求出网络的输出并由实际的输出计算出均方误差 (MSE)。将 MSE 作为衡量每个粒子的适应度值函数并求出每个粒子的个体极值;
- 4) 评价每个粒子的个体极值, 从中选择最优的个体极值作为这次迭代的全局极值 gb , 记录该最优值的粒子序号, 该极值作为 BP 预测网络下次迭代的最优参数;
- 5) 判断 gb 和当前的迭代次数是否满足 MPSO 结束条件, 满足的话则退出寻优转入步骤 7);
- 6) 根据公式 (4.8) 和 (4.9) 更新每个粒子的速度和位置并计算出每个粒子的适应度值, 若该值优于当前的值, 则将此次的 pb_n 作为当前的个体极值。将所有粒子中个体极值

最优的与当前全局最优值相比，若优于则将 pb_n 设置为该粒子的位置，记录下该粒子的位置。并根据式（4.5）更新惯性权重，再转入步骤 3）；

- 7) 根据得到的最优粒子位置作为 BP 网络的初始值，采用 LM 算法对优化后的 BP 网络进行二次训练，输入测试样本进行预测。

4.2 基于粒子群神经网络短时交通流量预测结果分析

本文基于粒子群神经网络的预测模型与 BP 神经网络短时交通流量预测模型相同，在训练的过程中 MPSO-BP 网络的参数分别设置为：惯性权重 $w_{\max}=0.9$ 、 $w_{\min}=0.4$ ，加速常数 $c_1=2$ 、 $c_2=1.8$ ，粒子个数 $N=40$ ，允许的最大迭代次数 $k_{\max}=200$ 。LM 算法训练设定最大允许的误差为 1×10^{-4} ，最大的迭代次数为 1000。为了便于分析比较，分别采用了 PSO-BP 算法和 MPSO-BP 算法对预测网络进行参数寻优，其适应度函数值变化对比如下图所示：

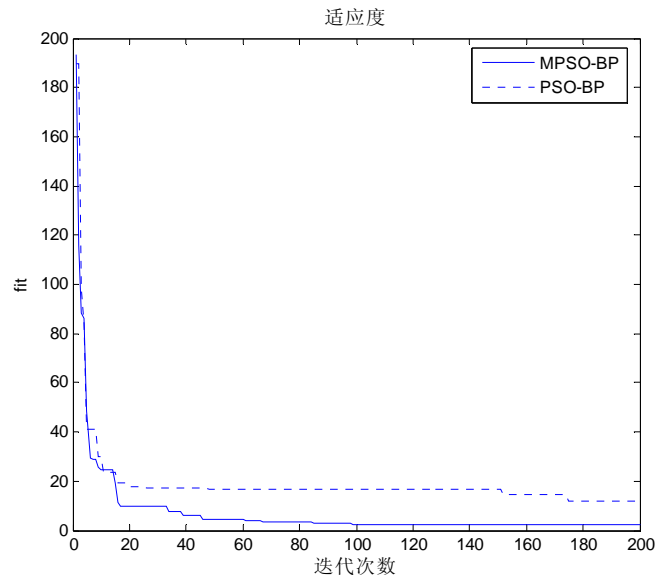


图 4.2 MPSO-BP 和 PSO-BP 粒子迭代过程

从图中可以看出，改进后的粒子群算法的寻优精度和速度都得到了提高。在 MPSO 算法对预测网络进行初步寻优确定了预测网络的初始参数后，用 LM 算法对 BP 网络进一步的训练，训练曲线图如下所示：

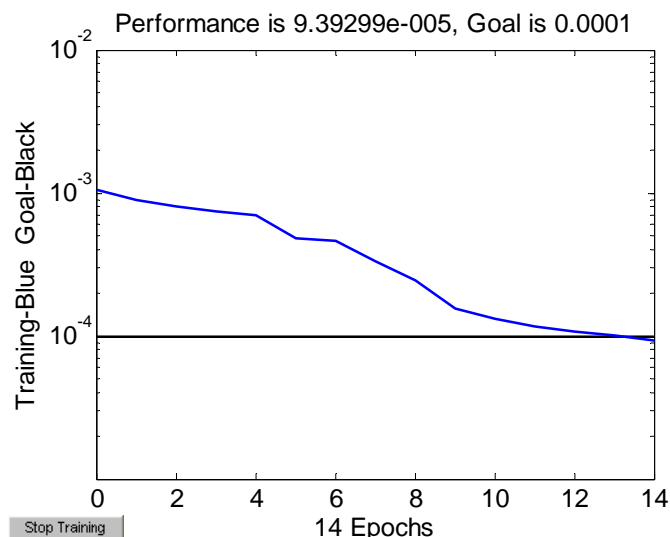


图 4.3 LM 算法训练误差图

此时网络已经在接近最优解的基础上被训练，与图 3.7 所示的单独使用 LM 算法训练网络相比，训练的速度得到了提高。

训练好网络后，输入测试样本进行短时交通流量的预测，图 4.4 为实际流量与预测流量的对比图，从图中可以看出，预测的交通流量曲线与实际交通量的曲线拟合程度比 LM-BP 神经网络预测模型要高。

实验的仿真结果为均方误差 $MSE=9.7372$ ；平均绝对误差 $MAE=25.0568$ ；平均绝对百分比误差 $MAPE=14.158\%$ ；仿真的各项指标均优于 LM-BP 神经网络。表 4.1 为交通流量高峰期 8:00~11:00 和 17:00~20:00 时间段的预测对比情况，从表中可以看出在交通流量高峰期相对误差 RERR 最大的为 29%；最小的为 -0.5%；平均相对误差为 12.290%；在高峰期预测的平均误差要比 LM-BP 预测网络要低。另外在所有的预测数据中 72.5% 的预测相对误差在 10% 以内；40.5% 的预测误差在 5% 以内。

为了便于分析比较，本文在相同的条件下以基本的粒子群算法 PSO-BP 优化神经网络，实验的结果为： $MSE=10.2385$ ； $MAE=27.9841$ ； $MAPE=14.982\%$ ，在所有的预测数据中 60% 的预测相对误差在 10% 以内；40% 的预测误差在 5% 以内。可见通过粒子群优化算法优化神经网络的初始参数可以提高网络的预测精度，改进后的粒子群优化算法在相同的控制精度下效果更好。

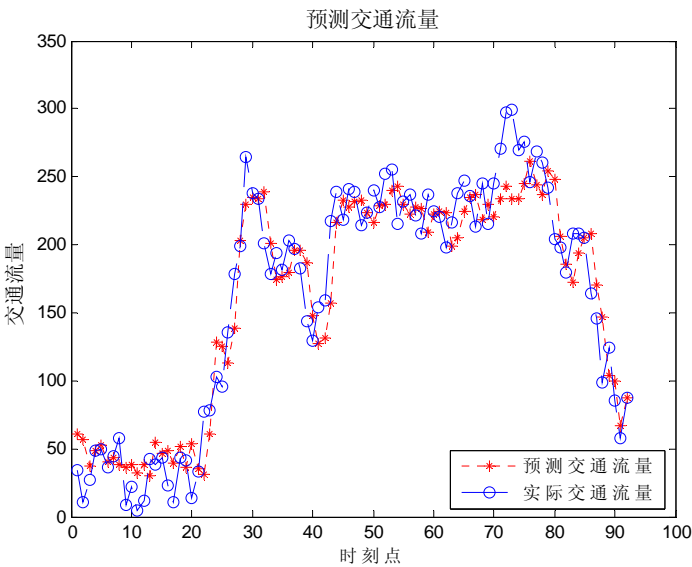


图 4.4 基于 MPSO-BP 预测网络的预测曲线图

表 4.1 基于 MPSO-BP 预测网络交通量预测结果

时间	预测交通流量	真实交通流量	相对误差
8:00	201	179	0.12
8:15	175	194	-0.1
8:30	177	182	-0.03
8:45	179	203	-0.1
9:00	196	197	-0.005
9:15	196	183	0.07
9:30	186	144	0.29
9:45	148	129	0.15
10:00	128	154	-0.17
10:15	132	159	-0.17
10:30	157	217	-0.28
10:45	217	239	-0.09
11:00	233	218	0.07
17:00	230	215	0.07
17:15	220	245	-0.1
17:30	234	271	-0.14
17:45	243	297	-0.18

18:00	234	299	-0.22
18:15	243	270	-0.1
18:30	234	276	-0.15
18:45	234	246	-0.05
19:00	245	269	-0.09
19:15	262	260	0.008
19:30	244	242	0.008
19:45	237	204	0.16
20:00	254	198	0.28

4.3 遗传算法在交通流量预测中的应用

遗传算法是一种模拟生物进化的计算思想，作为一种新型的全局优化算法，以其鲁棒性好、简单通用、并且适用于并行处理等优点，奠定了它在计算智能中的地位^[36]。本节通过对遗传算法基本原理的研究，尝试将遗传算法训练 BP 网络的初始参数，提高预测的精度。

4.3.1 基本遗传算法

遗传算法是遗传学和现代计算机科学相结合的产物，因此在遗传算法中常用到物种进化的基本用语。下表为遗传学和遗传算法中所对应的基本用语。

表 4.2 遗传学和遗传算法中所用的基本用语对照表

遗传学术语	遗传算法注释
染色体(Chromosome)	对解的编码
基因(Gene)	解中每一分量的特征
个体(Individual)	解
种群(Population)	个体的集合
适应性(Fitness)	适应度值
复制(Reproduction)	由适应度函数值选取的一组解
交叉(Mutation)	根据交叉算子产生新解的过程
基因型(Genotype)	基因组合的模型，为染色体的内部表现
表现型(Phenotype)	由基因型形成的个体

编码(Coding)	由问题空间向遗传算法空间的映射
解码(Decoding)	由遗传算法空间向问题空间的映射

在遗传算法中有三个基本的遗传算子：选择、交叉和变异，它们的特点如下^[37-39]：

第一：遗传算法中的操作都是随机进行的，但这种随机是一种高效有向的搜索。

第二：遗传操作的结果与遗传算子选取概率是密切相关的。

第三：遗传算子的操作与具体的实际问题有关，它们的操作和个体的编码方式直接相关，主要的编码方式有二进制编码和实数编码。本文采用实数编码，在解码的时候直接把基因值赋给预测网络的参数。

(1) 选择算子

选择(Selection)是在种群中挑选优质的个体并淘汰劣质个体的过程。选择的操作是建立在对种群个体适应度的评估上，其基本思想是个体适应度较大的个体被选中的概率较大。目前常用的选择方法有：适应度比例方法、最佳个体保留方法、期望值方法等。其中适应度比例方法也称赌轮选择或 Monte Carlo 选择，是目前遗传算法中最基本也是最常用的方法。

设种群的规模为 n ，其中，第 i 个个体适应度值为 f_i ，则个体被选中的概率为：

$$p_{si} = \frac{f_i}{\sum_{j=1}^n f_j} \quad (4.10)$$

选择概率 p_{si} 反应的是第 i 个个体的适应度在整个个体适应度总和中的比例。若个体的适应度越大，被选中的概率就越高。适应度的比例选择过程描述如下：

1) 计算群体中每个个体的适应度值 f_i ， $i=1,2,\dots,n$ ；

2) 计算群体中所有个体适应度值和 $F = \sum_{j=1}^n f_j$ ；

3) 计算个体 k 被选择的概率 $p_k = \frac{f_k}{\sum_{j=1}^n f_j}$ ， $k=1,2,\dots,n$ ；

4) 计算个体 k 的累积概率 $q_k = \sum_{j=1}^k p_j$ ， $k=1,2,\dots,n$ ；

在选择的过程中旋转 M 次，每一次都是按下列步骤选择一个个体加入到新的群体中：

a) 在 $[0,1]$ 区间内产生均匀分布的伪随机数 r ；

b) 如果 $r \leq q_1$ ，则选择第 1 个个体，否则选择第 k 个个体，其中 $2 \leq k \leq n$ ，使得 $q_{k-1} \leq r \leq q_k$

成立；

重复上述步骤 a) 和 b) n 次。

表面上看来个体的选择是随机的，但实际上的选择是根据邻近的两个适应度累计值的差值 $\Delta S_i = S_i - S_{i-1} = f_i$ 进行的，其中 f_i 为第 i 个个体的适应度。若 f_i 越大， ΔS_i 的距离就越大，那么随机落在这个区域的可能性就越大，即第 i 个个体被选中的概率就越大，下图描述了这种情形。图中的指针是固定的，外圈圆环可随意转动，环中每段对应适应度大小。从统计学上说，适应度大的个体被选中的概率大，反之被选中的机会小。这样处理增加了个体的多样性，以便执行交叉和变异操作。可见赌轮盘法既体现了“适者生存”的原则，又保持了种群的多样性。

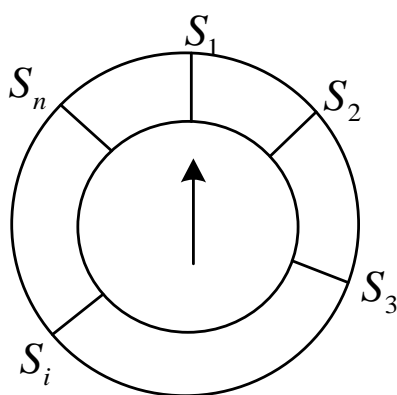


图 4.5 赌轮盘法方法原理

(2) 交叉算子

选择的过程中，按照适应度函数的优劣来选择合适的父代种群后，存入库中再进行随机的配对选择，并按一定的概率在两两之间进行交叉，也就是个体之间的信息交换，即生物在自然界中进化时染色体的交叉重组。交叉操作是按一定概率 p_c 进行的，较大的概率可增强遗传算法全局搜索的能力，但这样遭到破坏的可能性也较大；反之遗传算法的搜索能力就会变弱可能会陷入迟钝。一般的 p_c 选择区间为 0.6~1。

交叉操作是遗传算法中的核心内容，在进行选择操作后再以合适的概率进行交叉操作可能会产生品种优良个体，这些优良个体会作为整个搜索空间中的新起点。通过交叉的操作使得遗传算法的搜索能力得到了进一步的飞跃，使得一开始以随机构成的种群能形成新的优良个体。目前常用的交叉方法有基于二进制编码的交叉算子以及基于实数编码的交叉算子。其中，基于二进制编码的交叉算子包括单点交叉、二点交叉、一致交叉、均匀交叉等；基于实数编码的交叉算子方法有线性交叉、中间交叉等。下面介绍单点交叉的计算方法。

1) 对种群个体两两随机配对。假定种群规模为 N ，则共有 $\lfloor N/2 \rfloor$ 对互相配对的个体组合；

2) 对每一对互相配对的个体随机设置某一基因之后的位置为交叉点, 假定染色体长度为 λ , 则共有 $\lambda-1$ 个可能交叉点位置;

3) 对每一对配对的个体, 以设定的概率 P_c 在设定的位置相互交互个体的基因信息, 产生两个新的个体。

接下来给出一个单点交叉的例子。

个体 A 0011|11000 \rightarrow 001100111 新个体 A'

个体 B 1010|00111 \rightarrow 101011000 新个体 B'

本例中的交叉点位置设在第 4 和第 5 个基因座之间。交叉的时候该点后的个体部分串相互交换。于是, 个体 A 中的第 1 到第 4 个基因与个体 B 的第 5 到第 9 个基因位置构成新个体 A', 同理可得新个体 B'。

(3) 变异算子

变异算子是通过改变种群个体串中某一些基因座上的基因值, 对于由字符集 $\{0,1\}$ 组成的二值串来说, 变异的具体操作就是将相应的基因值取反, 即 $1 \rightarrow 0$ 或是 $0 \rightarrow 1$ 。变异算子的具体操作是先随机确定群体中所有个体的基因座, 然后以预先设定的变异概率 p_m 来对这些基因座的基因位置进行变异。变异的目的是可加强算法的局部搜索能力和保持种群的多样性。变异的概率应该取较小值, 否则算法在接近最优解时可能会因变异而遭到破坏, 通常变异的概率取 0.001~0.1。

目前常用的变异方法有: 基本位变异、逆转变异、均匀变异等。其中, 基本位变异是先在个体串中随机一个或多个挑选基因位, 再以变异概率 P_m 对这些基因信息做变动。下面给出 $\{0,1\}$ 二值串中基本位变异操作的例子。

个体 A 10 010 110 $\xrightarrow{\text{变异}}$ 10110010 个体 A'

在遗传算法的操作中, 交叉算子和变异算子分别作为主要算子和辅助算子, 遗传算法正是通过调节交叉算子和遗传算子来平衡算法的全局和局部的搜索能力。在基本的遗传算法中, 采用二进制编码方式, 使用固定长度的二进制符号来表示个体, 其基因由字符集 $\{0,1\}$ 组成, 选择操作使用适应度比例方法; 交叉操作采用单点操作; 遗传操作采用基本位变异; 交叉概率和变异概率都采用固定值。

基本遗传算法在运行前需要设定 4 个参数:

1) M : 群体的规模, 一般取 20~100;

2) G : 遗传运算的迭代次数, 一般取 100~500;

3) P_c : 交叉概率, 一般取 0.4~0.99;

4) P_m : 变异概率, 一般取 0.001~0.1。

下图 4.6 为一个完整的基本遗传算法的流程图:

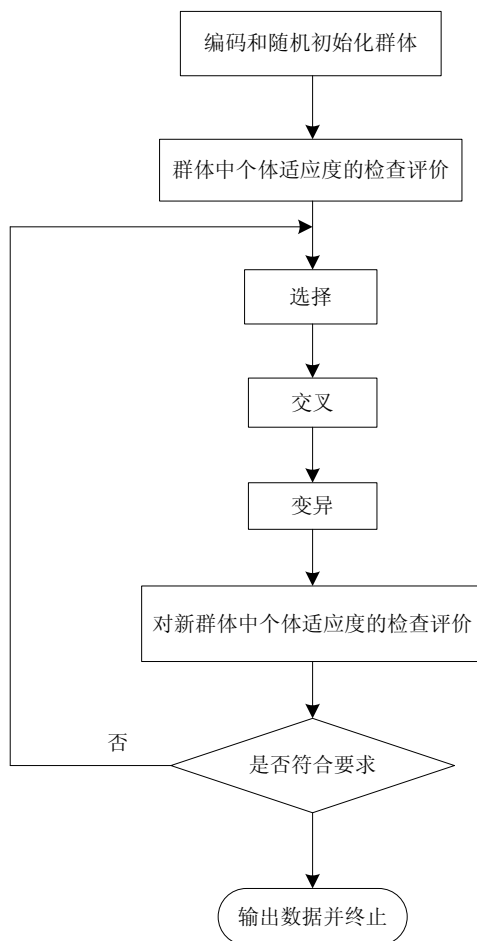


图 4.6 基本遗传算法流程图

4.3.2 基本遗传算法的不足及改进

遗传算法在理论上可以借鉴生物进化理论和遗传学理论来形成一整套完善的理论体系,但在实际应用中还存在很多问题等待去解决和完善,基本遗传算法的主要不足有^[40-42]:

1) 早熟收敛问题。遗传算法是采取个体适应值的大小来确定解的好坏,因此当某个个体的适应度值过大,这个个体的基因在整个种群就会扩大,这样导致了群体过早的失去了样本多样性并陷入局部最优解。

2) 局部搜索能力。遗传算法具有较强的全局搜索能力,但局部搜索能力不足,这使得算法在后期,收敛速度缓慢甚至无法收敛到最优解。

3) 遗传算子的无方向性。在遗传算法的三个操作算子中, 选择算子是保证找到近优解的唯一手段, 但交叉算子和变异算子并不能保证所引入的个体都是优良的。

针对遗传算法的改进, 主要包括对遗传的选择算子、交叉算子和变异算子的改进, 这其中包括自适应遗传算法、微量遗传算法和免疫遗传算法^[43]等。

在遗传算法的操作中, 交叉操作和遗传操作是遗传操作中的重要算子。交叉算子的作用是组合不同个体之间有用的信息来产生新的个体, 它可以大大的提高算法的搜索速度, 但若 p_c 过大, 越多的新个体会被引入, 但这些新个体可能会影响种群的进化速度。交叉操作在遗传算法中是起主要作用, 合理的选取会加大算法的收敛速度, 因此 p_c 常取 0.6~1, 变异操作的作用是保持群体中个体的多样性, 它主要用于增加种群的多样化, 是起辅助的作用。低的变异率可以防止在搜索过程中的任何位置永久保持不变, 这样保障了算法可以搜索空间中的每一个点; 高的变异率则趋向于随机搜索。因此应保证 p_m 取较小值, 常取 0.001~0.1。而在基本的遗传算法中, 交叉概率和变异概率都是给定的固定值, 在实际应用中必须通过不断的实验来调整这两个参数, 才能达到满意的效果。因此在这两个参数调整的基础上提出了自适应遗传算法(adaptive genetic algorithm, AGA)。

自适应遗传算法的改进主要集中在以下两个方面:

1) 由适应度值的大小及迭代次数来设计自适应的交叉和迭代概率公式。一般的调节公式遵循以下原则: 当个体适应值小于平均适应值即个体较差时, 应给予较大的交叉概率和较小的变异概率, 相反当个体适应值大于平均适应值即个体较好时, 此时当迭代次数越接近最大迭代次数时, 个体交叉概率应越小而变异概率要增大。

2) 精英保留策略。在进行交叉和变异操作后, 如果个体适应度值增大就保留这个变化, 反之保留原来的个体。

自适应遗传算法除了基本遗传算法的优点外, 还有以下特点:

- 1) 较强的全局搜索能力, 避免了早熟;
- 2) 较强的局部搜索能力, 避免了锯齿;
- 3) 使得遗传算法的进化具有方法性, 收敛能力加强。

在基本遗传算法的改进上, 自适应遗传算法的实施步骤如下:

- 1) 确定要求解问题的目标函数, 并把目标函数转化为非负的适应度函数, 而且是在任意情况下其值希望越大越好;
- 2) 编码: 将要求解问题中的数据表示成遗传空间中的基因型的串结构;
- 3) 随机初始种群: 随机产生 N 个初始串结构, 每一个串结构构成一个个体, N 个个体组

成种群，遗传算法以这 N 个串结构为迭代初始点，并设置初始迭代次数 $t=0$ ，最大迭代次数 G ；

4) 由适应度函数计算群体中每个个体的适应度值；

5) 选择：对每个个体的适应度值进行复制并形成 M 个临时的集合 C 。其中，适应度低的个体会被剔除，适应度高的个体被复制到下一代；

6) 交叉：由配对原则和自适应的交叉概率对集合 C 中的个体进行交叉操作，形成后代集合 C' ；

7) 变异：由变异原则和自适应的变异概率对集合 C' 中个体部分信息进行变异操作，以形成新的集合 C'' ；

群体在进行选择、交叉和变异后得到下一代；

8) 判断是否停止，若 $t \leq G$ ，则 $t=t+1$ ，并转至 4)；否则输出个体的最优解保存并结束。

文献[44]根据适应度值提出了动态的调整交叉和变异概率的 AGA^[44]，调节公式如下：

$$p_c = \begin{cases} \beta_1(f_{\max} - f') / (f_{\max} - \bar{f}) & f' \geq \bar{f} \\ \beta_2 & f' < \bar{f} \end{cases} \quad (4.11)$$

$$p_m = \begin{cases} \lambda_1(f_{\max} - f) / (f_{\max} - \bar{f}) & f \geq \bar{f} \\ \lambda_2 & f < \bar{f} \end{cases} \quad (4.12)$$

式中： p_c 、 p_m 分别表示为交叉概率和变异概率， β_1 、 β_2 为交叉概率的调整系数， λ_1 、 λ_2 为变异概率的调整系数， f 、 f_{\max} 、 \bar{f} 分别为群体中个体适应度值、最大值、平均值， f' 为双亲适应度中的较大者。通过对交叉概率和变异概率的动态定义改善了种群的全局和局部搜索能力，但此算法在个体适应度值等于或接近最大适应度值时， p_c 、 p_m 等于或接近零值，这对进化初期不利，因此本文采用文献[45]中改进的自适应遗传算法 IAGA 优化 BP 神经网络，调节公式如下^[45]：

$$p_c = \begin{cases} p_{c1} - \frac{(p_{c1} - p_{c2})(f' - f_{avg})}{f_{\max} - f_{avg}} & f' \geq f_{avg} \\ p_{c1} & f' < f_{avg} \end{cases} \quad (4.13)$$

$$p_m = \begin{cases} p_{m1} - \frac{(p_{m1} - p_{m2})(f - f_{avg})}{f_{\max} - f_{avg}} & f \geq f_{avg} \\ p_{m1} & f < f_{avg} \end{cases} \quad (4.14)$$

式中： p_{c1} 、 p_{c2} 、 p_{m1} 、 p_{m2} 分别为交叉概率和变异概率的调整系数，IAGA 使得个体交叉和变异概率不为 0，提高到了 p_{c2} 和 p_{m2} ，可使算法跳出局部最优解。

4.4 IAGA-BP 神经网络复合预测算法

遗传算法和神经网络都是将生物学原理应用于科学研究的成果，它们都有较强的解决问题的能力，越多的研究者都尝试将遗传算法的优点和神经网络的优点结合起来。本节尝试将遗传算法和 BP 神经网络结合起来，利用改进的自适应遗传算法优化 BP 神经网络的初始参数，然后再使用 LM 算法对网络进一步的训练至收敛，充分发挥遗传算法的全局搜索能力和神经网络的映射能力，IAGA-BP 复合预测算法步骤如下所示：

- (1) 建立 BP 神经网络的预测模型，输入归一化后的训练样本，确定网络的拓扑结构；
- (2) 初始化种群，确定种群的规模 N 、最大的进化代数 G 、初始的交叉概率 p_c 和变异概率 p_m 以及交叉和变异概率的调整系数 p_{c1} 、 p_{c2} 、 p_{m1} 、 p_{m2} ；
- (3) 将种群映射为网络的权值和阈值，对应于 BP 网络的初始参数，把预测输出和期望输出之间的误差绝对值和 E 作为个体适应度值 F ，由于适应度值越小越好，故适应度函数取 $f(i)=1/F$ ；
- (4) 选择：根据适应度函数计算每个种群中每个个体的适应度值，适应度值低的个体从种群中去除，适应度高的个体将被保留到下一代。本文采取公式 (4.10) 所表示的概率；
- (5) 交叉：本文采用实数交叉法，第 k 个染色体 a_k 和第 l 个染色体 a_l 在 j 位的交叉操作为： $a_{kj}=a_{kj}(1-b)+a_{lj}b, a_{lj}=a_{lj}(1-b)+a_{kj}b$ ，式中 b 是 $[0,1]$ 间的随机数。交叉概率采用公式 (4.13) 所示的自适应调整规则；
- (6) 变异：利用公式 (4.14) 的变异概率产生新的个体，采用基本位变异的方法；
- (7) 将得到的新个体引入种群，计算适应度值，判断是否满足最大迭代次数或要求的精度，结束优化，否则转向 (3)；
- (8) 根据得到的最优个体值作为 BP 网络的初值，采用 LM 算法对优化后的 BP 网络进行二次训练，输入测试样本进行预测。

4.5 基于 IAGA-BP 神经网络短时交通量预测结果分析

本文基于 IAGA-BP 神经网络的预测模型与 BP 神经网络短时交通流量预测模型相同，训练过程中 IAGA-BP 网络的初始参数分别设置为： $N=40$ 、 $p_{c1}=0.9$ 、 $p_{c2}=0.6$ 、 $p_{m1}=0.1$ 、 $p_{m2}=0.01$ ，初始的交叉概率 $p_c=0.4$ 、变异概率 $p_m=0.05$ ，最大迭代次数 $k_{max}=200$ ，LM 算法训练设定最大允许的误差为 1×10^{-4} ，最大的迭代次数为 1000。为了便于分析比较，同时采用了 GA-BP 算法和 AGA-BP 算法对网络进行参数寻优，其适应度值的变化如图 4.7 所示。从图中可以看出，

改进后的自适应遗传算法在训练速度和精度都有所提高。在 IAGA 算法对预测网络的初始参数进行初步寻优后，用 LM 算法对网络进行进一步的训练，LM 算法训练 BP 网络的训练曲线如图 4.8 所示，由图可知，在初步优化后网络的收敛速度比 LM-BP 网络的收敛速度要快。

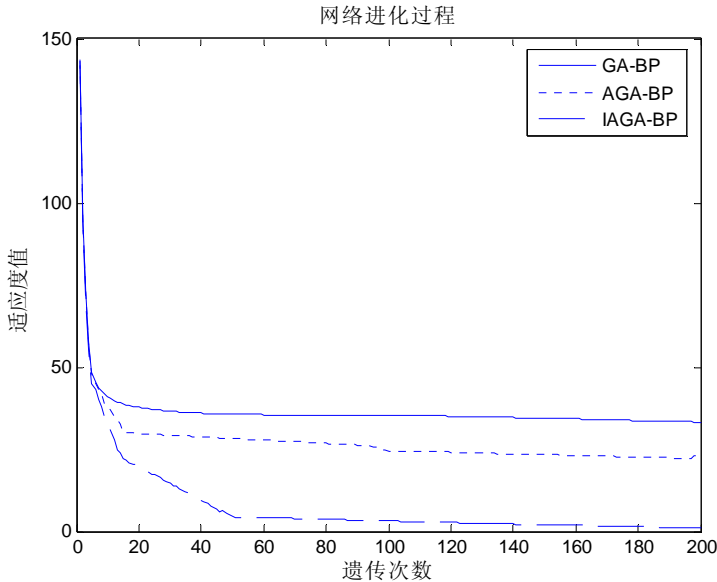


图 4.7 GA-BP、AGA-BP 和 IAGA-BP 的训练过程

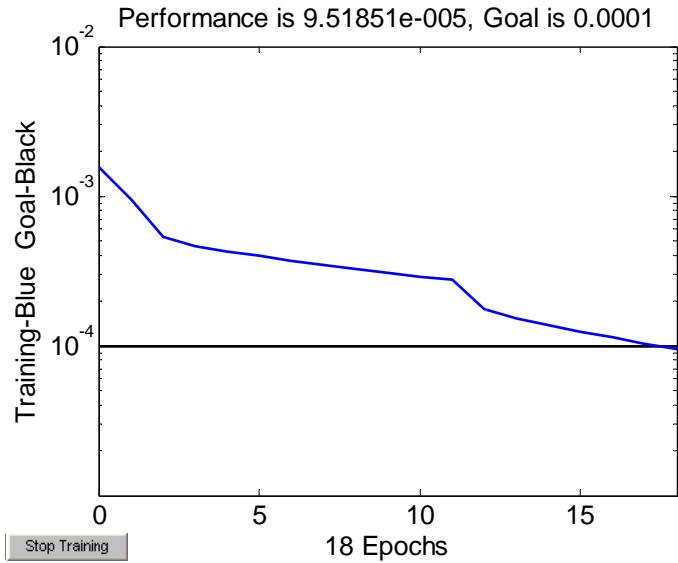


图 4.8 LM 学习算法训练误差图

用训练好的 BP 神经网络预测短时交通流量，图 4.9 为预测结果和实际交通流量的对比图。为了便于分析比较，表 4.3 列出了交通流量高峰期 8:00~11:00 和 17:00~20:00 时间段的预测对比情况。

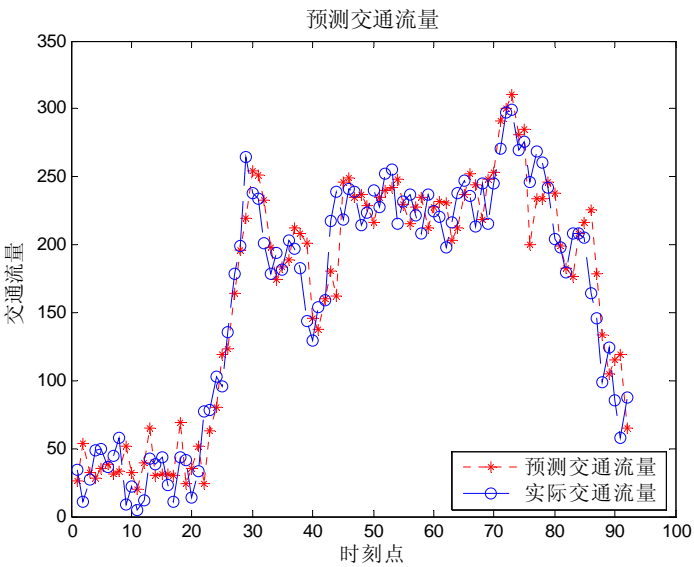


图 4.9 基于 IAGA-BP 预测网络的预测曲线图

表 4.3 基于 IAGA-BP 预测网络交通量预测结果

时间	预测交通流量	真实交通流量	相对误差
8:00	198	179	0.1
8:15	174	194	-0.1
8:30	184	182	0.01
8:45	188	203	-0.07
9:00	212	197	0.08
9:15	208	183	0.14
9:30	201	144	0.4
9:45	146	129	0.13
10:00	138	154	-0.1
10:15	161	159	0.01
10:30	247	217	0.14
10:45	249	239	0.04
11:00	250	218	0.15
17:00	248	215	0.15
17:15	253	245	0.03
17:30	290	271	0.07

17:45	300	297	0.01
18:00	310	299	0.04
18:15	280	270	0.04
18:30	285	276	0.03
18:45	200	246	-0.19
19:00	234	269	-0.13
19:15	234	260	-0.1
19:30	246	242	0.008
19:45	250	204	0.23
20:00	199	198	0.005

实验的仿真结果均方误差(MSE)为 10.4682；平均绝对误差(MAE)为 28.0130；平均绝对百分比误差(MAPE)为 15.043%，可见 IAGA-BP 神经网络的预测结果与 PSO-BP 网络相差不大，但由于优化了 BP 网络的初始权值和阈值，该模型的预测结果要优于 LM-BP 网络的预测模型。从表中可以看出在交通流量高峰期相对误差 RERR 最大的为 40%；最小的为 0.5%；平均相对误差为 9.63%；在高峰期预测的平均误差要比 MPSO-BP 预测网络要低。另外在所有的预测数据中 60%的预测相对误差在 10%以内；37.5%的预测误差在 5%以内。

4.6 本章小结

本章介绍了智能算法优化神经网络在短时交通流量预测中的应用，主要分为两个部分：第一部分详细介绍了粒子群算法的基本原理及其改进，并利用改进的粒子群算法优化 BP 神经网络的初始参数；第二部分详细介绍了遗传算法的基本原理及其改进，并用改进的自适应遗传算法优化预测网络的初始权值和阈值，在此基础上对短时交通流量进行了预测。

从预测的结果上来看，通过智能算法优化 BP 神经网络的初始参数，并在此基础上建立预测模型，充分结合智能算法和神经网络的优点，使预测的精度得到了提高。

第五章 APSO-BP 神经网络交通流量预测模型

5.1 自适应变异算子的引入

通过上一章的仿真结果来看,改进的粒子群算法(MPSO)总体上的各项指标优于其他的预测模型。本文的粒子群优化算法采取的是实数编码,并没有遗传算法中的选择、交叉、变异等操作,算法结构简单,运行速度快。粒子群算法在优化的过程中所有粒子都向最优解的方向飞去,粒子会逐渐同一化,群体的多样性会逐渐丧失并导致后期的收敛速度变慢^[46-48]。

假设在 N 个粒子组成的群体中,在第 k 次迭代时第 n 个粒子的个体最优值为 pb_n ,此时群体最优值为 gb 。在 MPSO 优化的过程中引入自适应的变异算子 λ , 变异算子可定义如下:

$$\lambda = gb / \bar{p} \quad (5.1)$$

$$\bar{p} = \frac{1}{N} \sum_{n=1}^N pb_n \quad (5.2)$$

实验证明,粒子群算法无论是早熟收敛还是全局收敛,群体中的粒子都会出现“聚焦”现象,此时全局极值就会是局部最优解, λ 的值就会趋近 1,算法便会陷入停滞状态。

自适应粒子群算法正是将遗传算法中的变异算子引入到算法中,通过判断 λ 的值来衡量粒子是否变异的依据。这里可假定 $\lambda=0.9$,当 $\lambda>0.9$ 时,则重新初始化粒子。自适应粒子群算法(APSO)是建立在改进的粒子群算法(MPSO)的基础上,简单的步骤可如下:

(1) 随机初始化种群中粒子的位置和速度;

(2) 将种群映射为网络的权值和阈值,利用预测网络的输入输出的均方误差作为评价每个粒子的适应度函数,得出每个粒子的个体极值并评价。从中选取最好的个体极值作为这次迭代的全局极值 gb ,判断 gb 和当前的迭代次数是否满足 MPSO 结束条件,满足的话退出寻优;

(3) 根据公式(4.8)(4.9)更新粒子的速度和位置,计算并更新每个粒子的适应度值,根据记录的值计算出变异算子 λ 的值,判断 λ 是否大于 0.9,若大于则重新初始化粒子,并根据式(4.5)更新惯性权重,再转入步骤(2)。

5.2 基于 APSO-BP 神经网络短时交通流量预测结果分析

自适应变异的粒子群算法在 MPSO 算法的基础上引入了自适应的变异算子,使得算法既

不会漫无目的地增加种群的规模，也可以有效避免粒子群算法中早熟收敛的问题。

将 APSO 算法优化神经网络得到的最优粒子作为 BP 网络的初始值，使用 LM 算法对优化后的 BP 网络进行训练，并输入测试样本进行预测，预测模型和参数的设置与 MPSO-BP 神经网络短时交通流量预测模型相同。图 5.1 为预测网络的预测曲线对比图。表 5.1 列出了交通流量高峰期 8:00~11:00 和 17:00~20:00 时间段的预测对比情况。

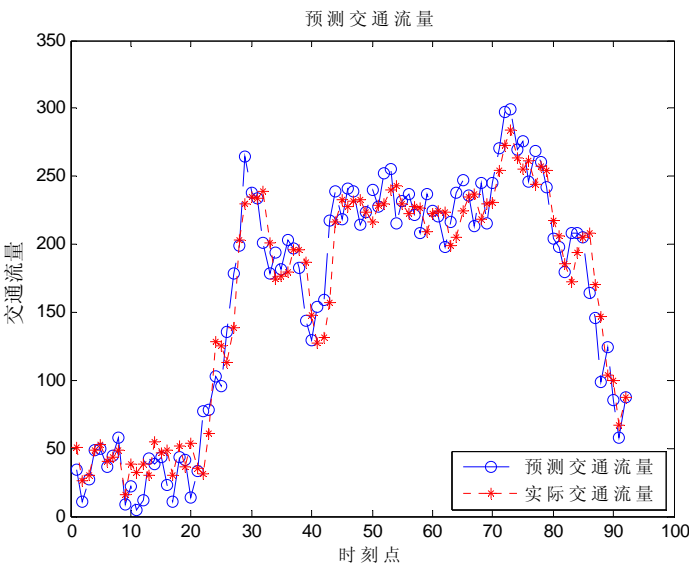


图 5.1 基于 APSO-BP 预测网络的预测曲线图

表 5.1 基于 APSO-BP 预测网络交通流量预测结果

时间	预测交通流量	真实交通流量	相对误差
8:00	190	179	0.06
8:15	171	194	-0.12
8:30	178	182	-0.02
8:45	190	203	-0.06
9:00	210	197	0.07
9:15	205	183	0.12
9:30	180	144	0.25
9:45	140	129	0.09
10:00	140	154	-0.04
10:15	165	159	0.04

10:30	240	217	0.11
10:45	245	239	0.03
11:00	245	218	0.12
17:00	230	215	0.07
17:15	238	245	-0.03
17:30	290	271	0.07
17:45	302	297	0.02
18:00	300	299	0.003
18:15	285	270	0.02
18:30	270	276	-0.02
18:45	230	246	-0.07
19:00	240	269	-0.1
19:15	240	260	-0.08
19:30	245	242	0.01
19:45	230	204	0.13
20:00	199	198	0.005

实验的仿真结果均方误差(MSE)为 8.9723；平均绝对误差(MAE)为 23.1254；平均绝对百分比误差(MAPE)为 8.923%，可见 APSO-BP 神经网络的预测结果在各方面的指标均优于其他算法；从表中可以看出在交通量高峰期相对误差 RERR 最大的为 25%；最小的为 0.5%；平均相对误差为 6.76%；在高峰期预测的平均误差相比其它预测算法都要低。另外在所有的预测数据中 75%的预测相对误差在 10%以内；52.5%的预测误差在 5%以内。

5.3 预测算法分析总结

本文通过对神经网络和智能优化算法的介绍，分别建立了 LM-BP、PSO-BP、MPSO-BP、IAGA-BP 以及 APSO-BP 的短时交通量预测模型，并分别对交通量进行了预测，表 5.2 列出了各个预测算法的性能指标：

表 5.2 预测算法性能指标对比

误差 算法	MSE	MAE	MAPE (%)	高峰期平 均相对误 差 (%)	10%以内 误差 (%)	5%以内误 差 (%)
LM-BP	10.7482	28.0769	15.356	12.7	54.5	28.5
PSO-BP	10.2385	27.9841	14.982	12.5	60	40
MPSO-BP	9.7372	25.0568	14.158	12.29	72.5	40.5
IAGA-BP	10.4682	28.0130	15.043	9.63	60	37.5
APSO-BP	8.9723	23.1254	8.923	6.76	75	52.5

从表中可以看出，通过智能优化算法优化 BP 网络的初始权值和阈值，之后使用 LM 学习算法对优化后的网络进行训练，在此基础上建立的预测模型的预测精度均比 LM-BP 预测模型的要高，这是因为智能优化算法使得预测网络在全局范围内搜索最优值，提高了 LM 算法训练的速度和精度，从而提高了网络的预测精度。

在改进的粒子群算法中引入了遗传算法中的自适应变异算子的思想，在此基础上建立了 APSO-BP 神经网络的预测模型，从表中可以看出，该预测模型的性能指标均优于其他预测算法，这是因为改进的粒子群算法同时采用了惯性权重和收缩因子，使算法能很好的平衡全局和局部搜索能力，同时采取了遗传算法中的变异思想，既不会漫无目的地增加粒子群的规模，又可以丰富种群的多样性，从而可以避免粒子群算法中的早熟收敛问题。

5.4 本章小结

本章将变异思想引入到粒子群优化算法中，建立了基于 APSO-BP 神经网络的预测模型，并通过仿真实验得出了相应的预测指标，然后将本文中所提到的预测算法的性能指标作了一个比较，通过对比验证了本章所提出的 APSO-BP 预测算法的可靠性。

第六章 总结与展望

随着汽车总量的快速增长和交通网络的迅速发展，越来越严重的交通问题使得人们对交通管理越来越重视，智能交通系统的快速发展使得城市的交通问题得到合理的解决。而准确的交通流量预测信息是其实现的前提和基础。本课题主要对短时交通流量预测这一热点问题进行了研究，针对交通流量高度的非线性和不确定性等特点，在国内外研究的基础上，应用新型的神经网络技术作为建立预测模型的基础，全文主要的研究工作和结论如下：

(1) 对人工神经网络的发展历史作了简要的介绍，重点研究了 BP 神经网络的基本原理。通过分析明确了 BP 神经网络在处理非线性问题上的强大能力，适用于对交通流量进行预测。

(2) 详细研究了 BP 神经网络交通流量预测模型的建立，其中包括输入数据的选择、数据的可预测性分析、网络结构和参数的设计以及梯度下降法的研究。通过仿真实验验证了传统 BP 学习算法的不足，在此基础上，通过对训练算法的研究，建立了基于 LM-BP 神经网络的短时交通流量预测模型，采用 LM 学习算法对神经网络进行训练，并通过仿真验证了算法的可行性。

(3) 为了避免神经网络因初始值选取不当而易陷入局部极小点、收敛速度慢的缺点，研究了如何将智能算法与 BP 神经网络结合起来。其中分别研究了粒子群优化算法的基本原理及其改进、遗传算法的基本原理及其改进，将这些智能优化算法优化 BP 神经网络的初始参数，采用 LM 学习算法对优化后的网络进行训练，由于智能优化算法具有很好的全局搜索能力，在此基础上分别建立了基于 PSO-BP、MPSO-BP、IAGA-BP 神经网络短时交通流量预测模型，通过仿真实验表明智能优化算法优化后的神经网络预测模型可以有效的解决网络易陷入局部极小点的问题，预测的精度得到了提高。

(4) 通过对粒子群优化算法和遗传算法的原理研究，提出了在粒子群算法中引入遗传算法中的变异算子，将粒子群优化算法的全局搜索能力和遗传变异的局部调控能力的优点结合起来，并在此基础上建立了基于 APSO-BP 神经网络的短时交通流量预测模型，仿真结果表明该预测模型具有更高的预测精度。

课题的研究以短时交通流量的预测为对象，虽然得出了一些成果，但仍然存在着许多不足之处，以后还需要进一步的研究。具体的工作和需要解决的问题有以下几点：

(1) 样本数据的选取对神经网络预测模型的学习性能影响较大。神经网络训练需要较多的基本数据，但过多的数据会使神经网络的学习变得复杂，因此如何选取合理的数据样本是

神经网络预测方法工作的一个重点。

(2) 如何确立合适的神经网络预测模型, 其中包括隐含层结点的选取、网络初始参数的确定以及激励函数的选取等, 本文都是依靠前人总结的经验, 这在以后都需要进一步的研究。

(3) 将智能优化算法与神经网络有效地结合起来, 充分发挥智能优化算法和神经网络的优势, 由于粒子群优化算法和遗传算法都有着各自的缺点, 如何设计智能优化算法与 BP 神经网络相结合的预测模型, 也是今后需要研究的问题。

(4) 随着对神经网络模型的进一步的研究, 出现了越来越多新兴的神经网络模型。如小波神经网络、混沌神经网络^[49-50]等, 将这些模型用于城市交叉路口交通流量预测中, 无疑为交通流量的预测提供了丰富的方法。

参考文献

- [1] 郇洪江,宫宁生,胡斌.改进的 BP 神经网络在交通流量预测中应用[J].微电子学与计算机,2010, 27(1):106-108.
- [2] 朱文兴,龙艳萍,贾磊.基于 RBF 神经网络的交通流量预测算法[J]. 山东大学学报,2007, 37(4):23-27.
- [3] MATLAB 中文论坛, MATLAB 神经网络 30 个案例分析[M]. 北京: 北京航空航天大学出版社, 2010.
- [4] 胡佩锋. 交通流量短时预测方法研究[D]. 北京交通大学硕士学位论文, 2006:33-38.
- [5] Haibo Chen, Susan Grant-Muller. Use of sequential learning for short-term traffic flow forecasting[J]. Transportation Research Part C.9(2001):319-336.
- [6] 犹勇. 组合预测方法在城市交通流量预测中的运用[J]. 重庆工商大学学报(自然科学版), 2007, 24(2):122-125.
- [7] 张敬磊,王晓原.基于非线性组合模型的交通流预测方法[J]. 计算机工程, 2010, 36(5): 202-204.
- [8] 孟维伟.基于神经网络的交通量预测技术研究[D].南京理工大学硕士学位论文,2006.5.
- [9] 田雨波.混合神经网络技术[M]. 北京: 科学出版社, 2009.
- [10] 飞思科技产品研发中心,神经网络理论与MATLAB7实现[M]. 北京: 电子工业出版社, 2005.
- [11] 叶嫣,吕智林. 基于粒子群优化的神经网络短时交通流量预测[J]. 计算机工程与设计, 2009, 30(18):4296-4298.
- [12] 申慧,刘知贵,李春菊.基于 BP 神经网络的交通流量预测设计[J]. 西南科技大学学报, 2008, 23(2):72-75.
- [13] 尚宁,覃明贵,王亚琴.基于 BP 神经网络的路口短时交通流量预测方法[J]. 计算机应用与软件,2006, 23(2):32-33.
- [14] 承向军,杜鹏等.单路口交通信号智能控制方法[J]. 系统工程学报, 2005, 20(1):23-29.
- [15] 王燕妮,樊养余.改进 BP 神经网络的自适应预测算法[J]. 计算机工程与应用,2010, 46(17): 23-26.
- [16] 黄席樾,向长城等. 现代智能算法理论及应用[M]. 北京:科学出版社,2009.
- [17] Chenghai Yu, Jie Chen, Jianlong Xu, Weiqiang Xu. Traffic Prediction in Telecommunications Networks: A Combined Forecast Method Based on Adaptive Genetic Algorithm[C]. 2009 International Conference on Intelligent Human-Machine Systems and Cybernetics, 2009
- [18] Yang Zuyuan, Huang Xiyue, Yin Lisheng, Liu Hongfei. GA Based Neural Network for Short-term Traffic Flow Prediction in Urban Signalized Arterials[C]. Computational Intelligence and Security, 2006 International Conference on, 2006
- [19] Quek C, Pasquier M, Lim B B S. Pop-Traffic: A Novel Fuzzy Neural Approach to Road Traffic Analysis and Prediction[J]. IEEE Transactions on Intelligent Transportation Systems, 2006, 7(2):133-146.
- [20] ZHANG Ya-ping, PEI Yu-long. Research on traffic flow forecasting model based on cusp catastrophe theory[J]. Journal of Harbin Institute of Technology(New Series). 2004,11(1):1-5.
- [21] 李婧瑜,李歧强,侯海燕,杨立才.基于遗传算法的小波神经网络交通流预测[J]. 山东大学学报(工学版),2007, 37(2):109-112.
- [22] Poli R. Analysis of the publications on the applications of particle swarm optimization[J]. Journal of Artificial Evolution and Applications, 2008, 8(2):4.
- [23] Robinson J, Rahmat-Samii Y. Particle swarm optimization in electromagnetics[J]. IEEE Trans. on Antennas and Propagation, 2004, 52(2):397-407.
- [24] Whitely D, Starkweather T, Bogart C. Genetic algorithms and neural networks: Optimizing connections and connectivity[J]. Parallel Computation, 1990, 14(3):347-361.
- [25] Hussein Dia. An object-oriented neural network approach to short-term traffic forecasting[C]. EurOpean Journal of Operational Research, 131(2001):253-261.
- [26] Kennedy J, Eberhart R C. Particle swarm optimization[C]. IEEE International Conference on Neural

- Networks, 1995:1942-1948.
- [27] Zheng Y L, Ma L H, Zhang L Y, et al. On the convergence analysis and parameter selection in particle swarm optimization[C]. Proceedings of the Second International Conference on Machine Learning and Cybernetics, 2003:1802-1807.
- [28] Lin C, Feng Q Y. The standard particle swarm optimization algorithm convergence analysis and parameter selection[C]. Proceedings of Third International Conference on Natural Computation, 2007.
- [29] 田雨波,朱人杰,薛权祥. 粒子群优化算法中惯性权重的研究进展[J]. 计算机工程与应用, 2008, 44(23):39-41.
- [30] 韩江洪,李正荣,魏振春. 一种自适应粒子群优化算法及其仿真研究[J]. 系统仿真学报, 2006, 18(10):2969-2971.
- [31] Y. Shi, and R. C. Eberhart, (1999). Empirical study of particle swarm optimization[J]. In Proc.1999 Congress on Evolutionary Computation, 1945- 1950. Piscataway, NJ: IEEE Service Center.
- [32] 江涛,张玉芳等. 一种改进的粒子群算法在 BP 网络中的应用研究[J]. 计算机科学, 2006, 33(9):164-165.
- [33] 何佳,陈智慧. 综合改进的粒子群神经网络算法[J]. 计算机工程与设计, 2008, 29(11):2890-2892.
- [34] Kennedy J, Eberhart R C. Particle Swarm Optimization[C]. Piscataway:Proceedings of the IEEE Conference on Evolutionary Computation, 2001:101-106.
- [35] M. Clerc and J. Kennedy. The particle swarm explosion, stability, and convergence in a multidimensional complex space[J]. IEEE Trans. Evolutionary Computation, vol. 6 , no.1,pp. 58-73, 2002.
- [36] 曹琛荔,孙旋.基于神经网络与遗传算法的交通流量预测研究[J]. 仪表技术, 2011, 01:22-24.
- [37] 任海艳,陈飞翔.自适应遗传算法的改进及其在曲线化简中的应用[J]. 计算机工程与应用, 2012, 48(11):152-155.
- [38] 陈超. 自适应遗传算法的改进研究及其应用[D],华南理工大学硕士学位论文, 2011.5.
- [39] Bramlete M F. Initialization,Mutation and Selection Methods in Genetic Algorithms for Function Optimization[J]. Proc ICGA4, 1991.100-107.
- [40] 姚亚夫,刘侃.基于遗传神经网络的交通流量预测研究[J]. 公路与汽运, 2007.
- [41] Shiliang Sun, Changshui Zhang, Member. A Bayesian Network Approach to Traffic Flow Forecasting[J]. IEEE Transactions on Intelligent Transportation Systems. 2006,7(1):124-132.
- [42] Feng jin, Shiliang Sun. Neural Network Multitask Learning for Traffic Flow Forecasting[C]. International Joint Conference on Neural Networks. 2008,1899:1902.
- [43] Shiliang Sun, Changshui Zhang, Member. The Selective Random Subspace Predictor for Traffic Flow Forecasting[J]. IEEE Transactions on Intelligent Transportation Systems.2007,8(2):367-373.
- [44] 杨超,王志伟.经 GA 优化的 WNN 在交通流预测中的应用[J]. 计算机工程, 2011, 37(14): 149-151.
- [45] 任子武,伞冶.自适应遗传算法的改进及在系统辨识中应用研究[J]. 系统仿真学报,2006,18(1): 41-44.
- [46] 吕振肃,候志荣.自适应变异的粒子群优化算法[J]. 电子学报, 2004, 33(3):416-419.
- [47] Stacey A, Jancic M, Grund Y I. Particle swarm optimization with mutation[J]. The 2003 Congress on Evolutionary Computation, Canbella, 2003:1425-1430.
- [48] 梁军,程灿.改进的粒子群算法[J]. 计算机工程与设计, 2008, 29(11):2893-2896.
- [49] DEL YON B, JUDIL SKY A, BENVENSISE A. Accuracy analysis for wavelet approximations[J]. IEEE Trans on Neural Network, 1995, 6(2):332-348.
- [50] 曾建潮,介婧,崔志华. 粒子群算法[M]. 北京: 科学出版社, 2004.

附录 1 攻读硕士学位期间撰写的论文

(1) 吴凯, 周西峰, 郭前岗, 基于粒子群神经网络的负荷预测方法研究, 电测与仪表, 已发表

致谢

时光飞逝，三年研究生时光转瞬即逝，在此向所有关心、指导帮助我的老师、同学和亲人以衷心的感谢！

首先我要感谢我的导师周西峰副教授和郭前岗教授。他们渊博的学识、严谨的处事原则、乐观开朗的生活态度，无时无刻向我传授着做人处世的道理和做学问的方法。从课题的开题到撰写论文的每一个详细步骤。两位导师都给与我无私的指导和帮助，提出了许多宝贵性的意见，正是两位导师不断的鞭策和鼓励，使我能够顺利的完成了论文的撰写工作。再次向两位导师表示感谢，希望他们身体健康、幸福快乐！

其次我要感谢我们 202 教研室的兄弟姐妹们，在学习和生活上都给予了我支持和帮助，需要特别感谢我的室友王严、朱文文、黎敏，感谢他们陪我度过难忘的三年研究生时光。

最后我将感谢我的家人，是他们无私的奉献和支持，才使得我能够乐观开朗的面对生活的困难，谢谢他们！