时间序列预测模型及其算法研究

模式识别与智能系统专业

研究生：罗凤斐　　导　师：古钟璧

在自然科学和社会科学各领域中，大量决策问题离不开预测，预测是决策的基础。解决预测问题的最有力的方法是发现、揭示给定动态过程或现象背后的规律。实际中，有关事物的信息经常是不完全的，有关的理论也是不完善的，人们对事物的了解仅限于观测数据，即时间序列，因此只能利用现有的历史数据构造模型，进而预测未来。

本文首先介绍了时间序列预测目前采用的方法与各种预测模型，介绍了这些方法与模型的性质特征，总结出目前的方法与模型在处理非线性系统的预测方面的特点与需要改进的一些方面；然后介绍了近些年兴起的神经网络尤其是前馈神经网络（BP 神经网络）在非线性预测模型方面的应用及表现出来的良好的特性与优势，同时也指出 BP 神经网络预测模型及其在训练算法方面的一些固有缺陷：BP 神经网络是一种静态的网络，没有时间处理能力，所以不能进行时间序列模式的识别；标准 BP 算法收敛速度慢、易落入局部极小点。针对这些缺陷，本文提出采用线性 AR 预测模型良好的对时间序列的模式识别能力来进行识别，构造出具有代表性的样本训练神经网络；同时在分析了标准遗传算法的主要缺点的基础上，对标准遗传算法（SGA）进行了改进设计，然后用改进的遗传算法（IGA）来优化网络的初始权阈值，最后采用 L-M（Levenberg-Marquardt）BP 算法来进行网络的训练与寻优。这样，既运用了 AR 模型良好的模型辨识能力，又发挥了神经网络良好的非线性映射能力，还综合了遗传算法的全局优化能力，从而使构造的预测模型更加实用、特性更好。

本文将所构造的模型与所设计的算法应用到都江堰灌区渠首岷江上游来水预测中，并通过大量的实验与仿真，实验结果表明，本文所设计的模型与算法

是可行的，且预测效果比传统的线性预测模型以及未经过算法改进的神经网络预测模型都好，从而验证了本模型与算法的可行性、有效性和优越性。此外，本模型及算法为时间序列的预测提供了一种预测的方法，具有较好的参考价值。

关键词：时间序列预测　非线性自回归神经网络 LMBP 算法 标准遗传算法（SGA）　改进遗传算法（IGA）

Study on the Model and Algorithm of Time Series Prediction

Pattern Recognition and Intelligent System

Graduate: Luo Fengman **Tutor:** Gu Zhongbi

In each field of natural science and social science, a great deal of decision problems can not get away from the prediction which is the foundation of the decision. The best method to settle down prediction problems is to detect and find out the law in the dynamic state process or phenomenon. In natural, the needed information is usually insufficient, also the relative theories. The people's understanding to the thing is limited by the observed data ,namely time series. So we can make use of the existing history data to establish a model to predict the future.

Firstly ,in this paper, introduces the current ways and models that the time series prediction adopt and the character and summarizes the aspects that the current ways and model are needed to be improved in the nonlinear system prediction. Secondly, introduces the good character and advantages of the artificial neural networks especially feed forward neural network (BP Neural Network) in the application of the nonlinear prediction models. Then, points out the weaknesses which exists in the BP Neural Network prediction model and its training algorithm: The BP neural network is a kind of static network that have no processing ability of time so that it can't identify the time series model. The standard BP algorithm is slowly constringency and is easy falling into the local minimum points. Aiming at these weaknesses, the paper presents a good identify ability of the AR prediction model which is used to identify the time series and set up the representative training samples in the BP Neural Network. Meanwhile, based on the analyzing the main

defects of the Standard Genetic Algorithm (SGA),designing Improved Genetic Algorithm (IGA) and using it to optimize initial weights of neural network ,and at last, adopting L- M(Levenberg- Marquardt) BP algorithm to training BP neural network to find the best optimized point. Thus, by using of AR prediction model's recognition capability, BP Neural Network prediction model's good nonlinear reflection ability and genetic algorithm' s overall optimize ability, the model that this paper constructs is better practical and has better qualities.

Applying the designed prediction model and algorithm to the prediction of the Ming river's incoming water, head of Dujiangyan irrigation areas, and a great deal of experiments and simulations show that the model and its algorithm is feasible, the prediction result is better than the traditional linear prediction ways and neural network prediction ways without improved algorithm. So, the model and its algorithm are feasible , useful and ascendant. In addition, this model and its algorithm provide a kind of method for time series prediction and have the better reference value.

Keywords: Time Series Prediction, Non_linear Autoregressive Neural Network, Levenberg- Marquardt Back-Propagation(LMBP) Algorithm, Standard Genetic Algorithm(SGA), Improved Genetic Algorithm(IGA)

1 绪论

1.1 课题提出及意义[1][2][3][4][5][6]

对于预测对象的所有已知资料只有对象随时间变化的历史资料的场合，时间序列分析预测法是系统预测的最基本的方法。近年来，时间序列建模及预测一直是学术研究和实际应用领域的研究热点。如自然领域的河水来水量预测、流域降水量预测、太阳黑子数预测等，社会领域中城市交通量预测、某一地区人口增长量预测、医院门诊量预测等，经济领域中的股市价格预测、国民收入预测、产品价格预测等。根据对许多有关时间序列预测方面的资料的分析和研究，人们逐渐掌握了一些建模及预测的基本规律。有两种建立时间序列预测模型的方法:如果数据间满足线性关系，可以用传统的建模及预测方法，如自回归(AR),滑动均值(MA)以及混合自回归滑动均值(ARIMA),即假设一组时间序列的未来值线性相关于其历史值。反之，若数据关系是非线性的，针对某些特殊情形，采用非线性建模的方法，但有时候仍旧用线性模型来近似拟合非线性时间序列，但这种方法是比较粗糙的。由于线性模型易于建立和运行，也便于理解和解释，故人们通常也用线性预测方法来解决现实世界中的许多问题。对于线性时间序列，传统的时间序列预测方法是很有效的。然而，现实世界中也存在许多非线性的问题，再用传统的线性预测方法不能很好的处理复杂的数据关系，从而导致预测精度不高。为了改进对于非线性系统的预测，人们已经提出许多非线性时间序列建模的方法。比如：双线性模型、阈自回归模型(TAR),滑动变换自回归模型（STAR）[20]等，这些时间序列模型在解决某些非线性问题时是有用的，但这些非线性缺乏普遍性。20 世纪 80 年代神经网络的崛起，为解决非线性的问题提供了一条良好的途径。因此，从 90 年代开始，神经网络在非线性领域的研究与应用非常广泛。人工神经网络用于预测的最大优点在于它不需要设计任何数学模型，只要通过过去的经验对历史数据的训练和学习，网络就能够“模拟”并“记忆”输入变量和输出变量之间的任何复杂的“函数”关系，处理各种模糊的、非线性的含存“噪声”的数据，并通过“联想”来实现预测。但是，神经网络算法易陷入局部最小点的固有缺陷，会导致预测结果的不稳定

与影响预测精度的提高。因此，对神经网络算法的研究也是目前的一个研究热点。随着遗传算法作为具有系统优化、适应和学习的高性能计算和建模方法的研究渐趋成熟，遗传算法为解决全局优化问题，提供了比较理想的方法。因此，将遗传算法用来优化神经网络也已成为目前研究的热点。如何结合线性和非线性时间序列预测模型的优点，建立实用性和通用性更好的时间序列预测的方法和模型，在理论上和实用上都具有非常重要的意义。

1.2 研究现状[4][5]

自从1970年Box和Jenkins的著作《时间序列分析、预测和控制》问世以来，逐渐形成了一整套时间序列模拟、估计、建模、预测和控制的理论和方法，在动态数据的处理分析、复杂信息的加工提取、预测未来和在线控制等方面显示出传统的数理统计静态处理手段无可比拟的优越性。这本著作奠定了时间序列分析方法在科研、经济、社会等各领域中旺盛的生命力。

按系统本身的性质划分，有线性系统和非线性系统。相应的有线性模型和非线性模型。线性模型如自回归模型（AR）、滑动平均模型（MA）、自回归滑动平均模型（ARMA）等，线性模型概念清晰、发展已比较成熟，国内外已有许多用线性模型进行预测的实例。国内应用方面，袁振洲[41]在分析铁路货源货流数据的内在规律及其时间序列特性的基础上，提出应用自回归积分移动平均（ARIMA(p,d,q)）模型预测其发展趋势的理论依据，采用ARIMA方法对1989～1994年北京局铁路的货运总量煤炭进行了预测，通过对各种方法预测结果的综合比较和分析，ARIMA（p，d，q）模型的预测效果令人满意。徐峰[42]等提出采用时间序列AR模型进行震动趋势预测，将现场测得的非平稳振动序列通过ARIMA模型和标准化处理，转化成标准正态平稳时间序列，模型参数估计使用了方法简单、参数估计无偏、精度高的最小二乘法，通过现场实测数据进行验证，计算结果表明AR模型能够很好地拟合振动信号时间序列并取得了一定的预测精度，可以达到预测要求。然而，实际的系统线性是相对的，非线性是绝对的。当非线性因素的影响较小，或在某一范围内影响较小时，可以用线性模型来描述，但当这种描述或逼近得不到令人满意的结果时，就要应用非线性时序模型，或其它的任何非线性方法。非线性预测方法主要指两个方面，一是对非

线性物理系统进行背景分析和研究，针对不同的非线性特征采用相应的非线性模型;二是采用一些非线性的迭代、学习模型来拟合比较复杂的时间序列数据。显然，非线性现象最好用非线性方法来描述。非线性模型与线性模型比较，其使用范围要广泛的多。因此，目前对预测的研究热点就集中在非线性模型和方法上面。

随着人工智能技术的发展，人工神经网络(Artificial Neural Networks, ANN)已得到了日益广泛的研究和应用，它的广阔前景在于其良好的预测性和实用性，其发展潜力巨大。基于神经网络的时间序列预测方法是神经网络近几年形成的一个分支，正得到快速发展。90 年代中期，美国 Frontier Financial 公司利用神经网络来对股票市场进行预测，再由专家给出相应的建议等。国内也有神经网络在预测分析领域中实际应用的一些探讨。杨红卫等[43]建立了基于神经网络的毫米波雨衰减预测模型，并将结果与 CCIR 模型进行分析比较，说明此模型能够降低平均误差和均方差；吴微和陈维强[44]将反向传播神经网络(BP)应用于沪市综合指数涨跌的预测，表明人工神经网络应用于股票市场的预测是可行的，有着良好的前景。在神经网络趋势预测的研究与应用方面，施式亮等[45]指出传统的预测方法存在的缺陷，应用神经网络建立了时间序列的矿井安全性预测模型，提高了安全预测的精度。

神经网络模型的训练问题实际上是一种优化问题，即寻找最优的连接权值，神经网络尽快收敛。而人工智能另一学科遗传算法正是一种非常好的全局优化算法，因此，随着遗传算法的发展，它与神经网络结合用来优化神经网络的收敛速度和精度有着广阔的应用前景。基于遗传算法的神经网络时序预测模型也在逐步的探索中。

1.3 人工神经网络与遗传算法发展概述[7]~[13]

本文采用线性理论中的 AR 模型分析方法与非线性模型中的神经网络模型相结合，并采用遗传算法对模型的算法进行优化来进行时间序列的建模与优化。由于 AR 模型发展比较成熟，因此主要对神经网络和遗传算法的发展进行概要介绍。

自从 1943 年心理学家 McCulloch 和数学家 Pitts 提出 MP 神经元模型以来，

神经网络的研究已有60多年的历史，走过了一条曲折而不平衡的发展道路，几经兴衰。作为人工智能的神经网络系统的研究则是从20世纪50年代末60年代初开始的。1957年，Rosenblatt提出了感知器（Perceptron）模型，试图模拟动物和人脑的感知和学习能力，并提出了引入隐层处理元件的三层感知器的概念。1960年，Widrow和Hoff引入了最小均方差（Least Mean－Square，LMS）算法，并用他系统阐明了自适应线性元件（Adaline）模型及提出了一种有效的学习方法Widrow-Hoff学习规则。从而在60年代，掀起了神经网络研究的第一次热潮。1969年M. Miskey和S. Papert发表了名为“perception”的专著，他们在专著中指出，简单的感知器的功能是有限的，它无法解决线性不可分的两类样本的分类问题。它使整个70年代神经网络的研究处于低潮。直到80年代初期，1982年和1984年美国物理学家J.J.Hopfield发表了两篇神经网络的文章，提出了Hopfield网络，指出利用该网络可以求解联想记忆和优化计算的问题；1986年D.E.Rumellhart和J.L.Mcclelland等人提出了多层前馈网络的反向传播算法（Back Propagation），简称BP网络或BP算法，该算法解决了感知器不能解决的问题，Hopfield网络和BP网络及算法的提出，第二次掀起了一股竞相研究开发神经网络的热潮。如今，神经网络的研究进入一个新的发展时期，它的应用研究几乎覆盖了所有的领域，吸引了很多国家的科学家、研究机构及企业界人士等，大量的有关神经网络机理、模型、算法特性分析，以及在各方面应用的学术论文象雨后春笋般在报刊杂志上和许多国际学术会议中涌现，神经网络以及建立在神经网络原理基础上的神经计算机成为当代高科技领域中方兴未艾的竞争焦点。

为了借鉴自然界生物进化中自然选择和自然遗传机制来达到自学习与优化的目的，遗传算法最早是由美国Holland教授于1962年提出，试图从生物进化机理中发展出适合于现实世界复杂优化问题的模拟进化算法（simulated evolutionary optimization）。随后，Holland在1975年出版了遗传算法方面的经典著作“Adaptation in Natural and Artificial Systems”，该书系统的阐述了遗传算法的基本理论，发展了一整套模拟生物自适应系统的理论，标志着遗传算法作为计算智能的一门分支科学的正式诞生。同年，De Jong在他的博士论文“An analysis of the behavior of a class of genetic adaptive systems”中对Holland的模式做了大量严格的计算实验，得出了明确的结论，

建立了著名的 De Jone 五函数测试平台，对遗传算法的六种方案的性能进行了详细的实验和分析，为后继者提供了研究范例，并为以后的广泛应用奠定了坚实基础，其成果成为可遗传算法发展过程中的一个重要里程碑。从那以后，遗传算法逐渐发展成为一种通过模拟自然进化过程解决最优化问题的计算模型，适合用于解决科学研究和工程实际所遇到的各种搜索和优化问题。近几年来，遗传算法已经有了很大的发展，并不断地向其他学科和领域渗透，有关遗传算法、神经网络的结合与应用研究已经成为智能系统领域一个十分引人注目的研究动向，从而构成了一种新型的智能系统整体优化的结构形式。

1.4 主要工作和内容安排

如前面所述，时间序列分析方法很多，线性和非线性型模型都是可行的，理论上非线性模型的预测精度高于线性模型的预测精度。对于较为复杂的系统，由于其具有很强的非线性特性，常常会遇到预测精度不够理想的情况，这也是实际预测中的难题。提高时间序列的预测精度，也是国内外从事此领域的科研工作者不断追求的。因此本文根据线性理论和非线性理论，采用线性自回归模型和 BP（Back Propagation，简称 BP）神经网络模型相结合构建预测模型，然后采用 BP 训练算法、遗传算法（Genetic Algorithms 简称 GA）相结合，提出了一种混合训练方法：对标准遗传算法进行改进，提高遗传算法的寻优效率；对标准 BP 算法加以改进，提高神经网络的训练和收敛速度。然后用改进遗传算法优化确定 BP 神经网络的初始权阈值，确定一个较好的搜索空间，然后再由改进 BP 算法在这个解空间里对网络进行训练学习至收敛。从而实现三者的优势互补，发挥自回归模型的辨识能力、自回归神经网络的广泛映射能力和遗传算法的全局搜索能力。并将基于改进遗传算法和改进 BP 算法的非线性自回归 BP 神经网络预测模型应用于都江堰灌区渠首岷江上游来水预测中，并通过 MATLAB 环境仿真来验证该模型的可行性与实用性。

本文研究工作的主要研究内容包括：

第一章介绍了课题背景和意义，阐述了该课题国内外的发展和现状，并综述了人工神经网络和遗传算法的发展概况，在此基础上，提出了本文所做的主要工作。

第二章本章系统学习和介绍了自回归（AR）模型、人工神经网络和遗传算法的基础理论，从而为时间序列预测的模型建立及算法优化提供了理论依据。

第三章详细介绍了时间序列的AR（p）预测模型的辨识方法和BP神经网络预测模型的结构确定方式选取，并基于AR模型的辨识能力和BP神经网络模型的非线性映射能力，建立了时间序列的非线性自回归 BP 神经网络预测模型（NARBP预测模型）。

第四章分析了所建立的NARBP预测模型在算法方面需要研究的方面；从搜寻到全局最优点的角度，提出了采用遗传算法和BP算法的结合算法，并对标准遗传算法（SGA算法）进行改进，设计了改进遗传算法（IGA算法），最后给出了IGA算法和LMBP算法相结合的优化算法对上章所构建的预测模型进行算法优化的步骤。

第五章结合本人在研究生期间所作课题：都江堰灌区闸口岷江上游来水预测项目，介绍所构建的IGA-NARLMBP预测模型在河水来水预测中的应用。

第六章对本文所做的工作做了总结，并指出进一步需要研究的方向与工作。

1.5 本章小结

本章阐述了时间序列预测模型和方法在国内外的发展和现状，并综述了人工神经网络和遗传算法的发展概况，在此基础上，提出了本文所做的主要工作。

2 理论基础

2.1 引言

传统的时间序列分析与预测理论以线性自回归（AR）模型和线性自回归滑动平均（ARMA）模型为基础，对线性系统有较好的效果，但不适合于非线性系统的时间序列建模与预测。而神经网络技术的出现，为非线性系统的时间序列建模与预测提供了新的方法。而遗传算法为神经网络的全局寻优提供了良好的途径。本章介绍利用自回归（AR）和神经网络进行非线性系统的时间序列建模与利用遗传算法进行模型算法优化的基础理论。

2.2 线性 ARMA 模型和最优预测器[2][3]

时间序列的预测问题涉及到利用过去观测到的观测值预测当前与未来观测值的随机模型构造问题，在统计学中常常使用具有随机输入的线性随机差分方程来描述。目前最重要，也是最常用的是线性自回归滑动平均（ARMA）法。

ARMA 法，是以美国统计学家 Geogre E.P.Box 和英国统计学家 Gwilym M.Jenkins 的名字命名的一种时间序列预测方法。其预测模型分为：自回归模型（简称 AR 模型）、滑动平均模型（简称 MA 模型）和自回归滑动平均混合模型（简称 ARMA 模型）。它的基本思想是：将预测对象随时间推移而形成的数据序列视为一个随机序列，即除去个别的因偶然原因引起的观测值外，时间序列是一组依赖于时间 t 的随机变量。这组随机变量所具有的依存关系或自相关性表征了预测对象发展的延续性，而这种自相关性一旦被相应的数学模型描述出来，就可以从时间序列的过去值及现在值来预测其未来值。

2.2.1 线性 ARMA（p,q）模型

用于预测的最一般模型是线性 ARMA（p,q）模型

$$Y_t = r + \sum_{i=1}^{p} \varphi_i Y_{t-i} + \sum_{j=1}^{q} \theta_j e_{t-j} + e_t \tag{2.1}$$

其中，p,q 为模型阶次，φ_i 为模型参数。设 $E(e_t \mid Y_{t-1}, Y_{t-2}, \ldots) = 0$，即 e_t 为 0 均值，独立且同分布，与过去的 Y_t 无关的随机变量。在整个过程中 e_t 有有限方差 σ^2。r 为截断值，通常对一个零均值的随机过程 Y_t，截断 r 为 0。

2.2.2 最优预测器

预测理论的中心是最小均方误差意义下的最优预测。对于给定的无穷多个过去观测值，如果条件均值存在，最小均方误差预测器就是条件均值，即

$$Y = E(\hat{Y}_t \mid Y_{t-1}, Y_{t-2}, \cdots) \tag{2.2}$$

在实际应用中，人们仅能得到有限的过去观测值 $Y_{t-1}, Y_{t-1}, \cdots, Y_1$，此时，最小均方误差预测器为

$$\hat{Y} = E(Y_t \mid Y_{t-1}, Y_{t-1}, \cdots, Y_1) \tag{2.3}$$

2.2.3 最优 AR（p）预测器

由于 AR 建模比 MA 和 ARMA 建模在计算上要简单，而且一个 MA 或 ARMA 模型可以用一个足够高阶的 AR 模型来近似，因此，AR 模型是使用的最广泛的线性预测模型。

AR（p）模型是 ARMA（p，q）模型的特殊情况

$$Y_t = \Phi_1 Y_{t-1} + \Phi_2 Y_{t-2} + \cdots + \Phi_p Y_{t-p} + e_t \tag{2.4}$$

相应的最优预测器为

$$\hat{Y}_t = \Phi_1 Y_{t-1} + \Phi_2 Y_{t-2} + \cdots + \Phi_p Y_{t-p} \tag{2.5}$$

2.3 人工神经网络[5][7][18]

2.3.1 人工神经网络基本概念

人工神经网络(Artificial Neural Networks,缩写ANN)是对人脑若干基本特征通过数学方法进行的抽象和模拟，是一种模仿人脑结构及其功能的非线性信息处理系统[15]。人工神经网络具有高速的大规模并行处理特性，高维的非线性动力特性，高度的容错性和鲁棒性，信息存储的分布特性，自组织、自适应、自学习的非定常性，联想记忆的非定常性等。

随着人工神经网络理论研究的不断成熟，神经网络的发展进入了一个新时期，应用研究也得到迅速发展，其应用领域包括：建模、时间序列分析、模式识别和控制等，并在不断的拓展。进入20世纪90年代以来，人工神经网络作为新学科、新方法和新技术，在自然科学和社会科学各个领域得到了广泛的应用，取得了许多丰硕的成果。神经网络研究也随之得到飞速发展，各种网络结构和算法系统应运而生，逐渐形成了较为完善的人工神经网络理论体系。人工神经网络作为一种数学算法体系，已经解决了许多实际问题，它的生命力也恰好在于其广泛的应用价值。

2.3.2 人工神经网络的模型

人工神经网络是由大量处理单元广泛互连而成的网络，是人脑的抽象、简化、模拟，反映人脑的基本特征。一般说来，作为神经元模型应具备三个要素：

1、具有一组突触或连接，常用w表示神经元i和神经元j之间的连接强度，或称之为权值。与人脑神经元不同，人工神经元权值的取值可在负值与正值之间。

2、具有反映生物神经元时空整合功能的输入信号累加器。

3、具有一个激励函数用于限制神经元输出。激励函数将输出信号压缩（限制）在一个允许范围内，使其成为有限值，通常，神经元输出的扩充范围在[0,1]或[-1,1]闭区间。

最早的神经元模型是1943年由美国心理学家McCulloch和数学家Pitts提出的MP模型。而现在最为常用的神经元模型如图2.1所示，其中$x_1, x_2, \cdots, x_n$为从其它神经元传来的输入信号，θ_j为神经元的阈值，w_{ji}为神经元i到神经元j的连接权值，$f(\cdot)$为神经元的传递函数或激励函数。该神经元模型的输入输出关系为

$$s_j = \sum_{i=1}^{n} w_{ji} x_i - \theta_j = \sum_{i=0}^{n} w_{ji} x_i (x_0 = \theta_j, w_{j0} = -1) \tag{2.6}$$

$$y_j = f(s_j) \tag{2.7}$$

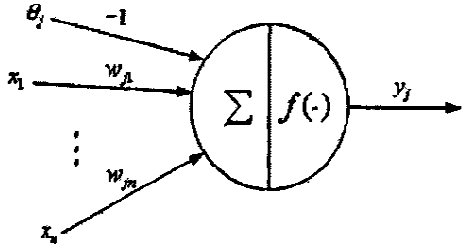

图2.1 人工神经元模型

激励函数是一个神经元及网络的核心。神经网络解决问题的能力与功效除了与网络结构有关，在很大程度上取决于网络所采用的激励函数。激励函数的基本作用有：①控制输入对输出的激活作用；②对输入、输出进行函数转换；③将输出幅度限制在一定范围内，如[0,1]或[-1,1]等。常用的基本激励函数有阈值函数、分段线性函数和Sigmoid函数。

2.3.3 人工神经网络的分类

可以从不同的角度对人工神经网络进行分类，如：

1、按拓扑结构分类：前馈网络、反馈网络和混合网络；

2、按学习方式分类：有监督学习和无监督学习；

3、按连接方式分类：线性关联网络和非线性关联网络；

4、按应用对象分类：模式识别、预测预报、方案优化、智能控制和智能决

策：

5、按神经网络算法的不同分类：Back Propagation（BP）网络、Hopfield网络、Gauss机等。

2.3.4 人工神经网络的运行过程

对于各种具有不同拓扑结构的人工神经网络而言，其运行过程大致上可以分成两个阶段：第一，网络的学习（或称之为训练）；第二，网络的联想。

1、学习过程

网络的学习过程就是通过不断地向网络输入一些样本模式，网络遵循一定的学习规则，来不断地调整网络的各层权重，使网络的输入和输出以一定精度向给定的样本模式逼近。学习的实质就是网络通过输入和输出信息识别存在于样本信息间的内在规律。

学习的方法可以分为两种：一种为有监督学习(Supervised Learning)或称有教师学习；另一种为无监督学习(Unsupervised Learning)或称无教师学习。

(1)有监督学习

有监督学习是给定一组输入、输出作为样本模式，网络根据学习规则不断地调整网络的权重，使网络的实际输出与样本的期望输出之间的误差降低到允许的精度。有监督学习常用来进行系统规律的映射。

(2)无监督学习

无监督学习的样本模式中不包含期望输出，网络完全按照样本所提供的某些统计规律来调整网络自身，以获得输入信息的某些固有特征，如聚类和统计上的分布规律。无监督学习常用于信息的分类。

不同拓扑结构的神经网络，甚至相同结构的神经网络，因其学习规则的不同，神经网络的功能各异。

2、学习规则

网络中的每一个神经元都是一个自适应单元，它的权重是可以调整的。权重调整方法的不同，便产生了各种各样的学习规则。

图2-2描述了网络中神经元节点j的权重Wj的调整情况。

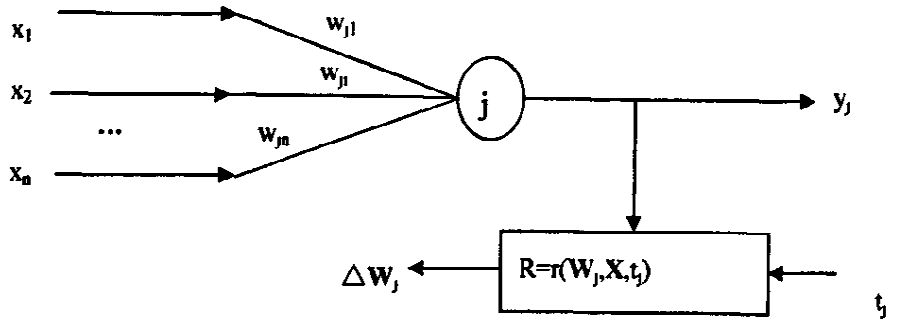

图 2-2 神经元节点权重的调整情况

学习信号 r 定义为:

$$R=r(W_j,\ X,\ t_j) \tag{2.8}$$

式中 t_j 代表示教信号。

$\triangle W_j$ 表示第 j 个神经元节点的权重增量，可定义为输入向量 X、学习信号 r 以及学习率 η 三者之积的形式:

$$\triangle W_j=\eta r(W_j,\ X,\ t_j)X^T \tag{2.9}$$

第 j 个神经元节点在下一迭代中对应的权重可表示为:

$$W_j(t+1)=W_j(t)+\triangle W_j(t) \tag{2.10}$$

3、网络的联想

完成训练的神经网络已经将学习样本的信息存储于网络之中，这样的网络已经具有映射功能。通过某种方式，使已经存储在网络中的有关信息从输出层复现出来的过程，称为网络的联想。不同网络的联想方式也不同。前馈联想：输入的激励模式只需一次性通过权重矩阵，就可以产生所需的响应，即输出模式；反馈联想：输入的激励模式通过网络产生的响应，作为下一次激励再输入网络，这样依次循环，直到输入、输出均不再变化为止。

2.3.7 BP 神经网络及其学习算法

人工神经网络的模型众多，从 20 世纪 40 年代开始，先后提出了数百种人

工神经网络模型，不同的模型有不同的学习算法。本文仅介绍前馈神经网络中的BP网络（Back Propagation Network）及其BP算法。

BP 神经网络(Back-Propagation Neutral Network) 通常是指基于误差反向传播算法(BP 算法)的多层前向神经网络，采用有教师的训练方式。它是D.E.Rumelhart 和 J.L.McCelland 及其研究小组在 1986 年研究并设计出来的。BP 网络能学习和存贮大量的输入-输出模式映射关系，而无需事前揭示描述这种映射关系的数学方程。由 Kolmogorov 定理[19](给定任一连续函数 $f:R^m \to R^n, y=f(x)$,这里 R 是闭区间[0,1]， f 可以精确地用一个三层前向网络实现逼近。)和 BP 定理[11]（给定任意 ε 和 L_2 在范数下: $f:[0,1]^m \to R^n$,存在一个三层 BP 网络,它可以在任意 ε 平方误差精度内逼近 f。）知:三层的 BP 网络可以以任意的精度逼近任意的非线性函数。BP 网络的学习规则是使用梯度下降法，通过反向传播来不断调整网络的权值和阈值，使网络的误差平方和最小。BP 神经网络模型拓扑结构包括输入层（input）、隐层(hide layer)和输出层(output layer)。BP 算法的学习过程，由正向传播和反向传播组成。在正向传播过程中，输入信息先向前传播到隐节点，经过激励函数后，再把隐节点的输出信息传播到输出节点，最后给出输出结果。每一层神经元的状态只影响下一层神经元的状态。如果在输出层不能得到期望的输出，则进入反向传播，将误差信号沿原来的连接通路返回，通过修改各层神经元的权值，使误差信号最小。BP 算法主要是解决多层神经网络的网络训练问题。 对于多层神经网络的输出层，很自然的可以用理想输出和实际输出之间的差异来进行训练，而对于隐含层，只能知道隐含层的输出，而无法知道隐含层的理想输出，如果没有合适的方法，就无法进行监督型学习。 BP 算法的发明，很好的解决了这个问题，由已知的输出层理想输出，推导出隐含层的学习法则。另外，BP 网络之间的关联信息分布式存储于各个连接权中，由于神经网络中连接权的个数很多，个别神经元的损坏只对输入输出关系产生较小的影响，这样，在缺失样本与参数的情况下也能保证较稳定的输出，因此，BP 网络还显示出了较好的容错能力[20]。

设一个有四层的神经网络，其中输入输出各一层，两层隐藏层。 如图 2-3:

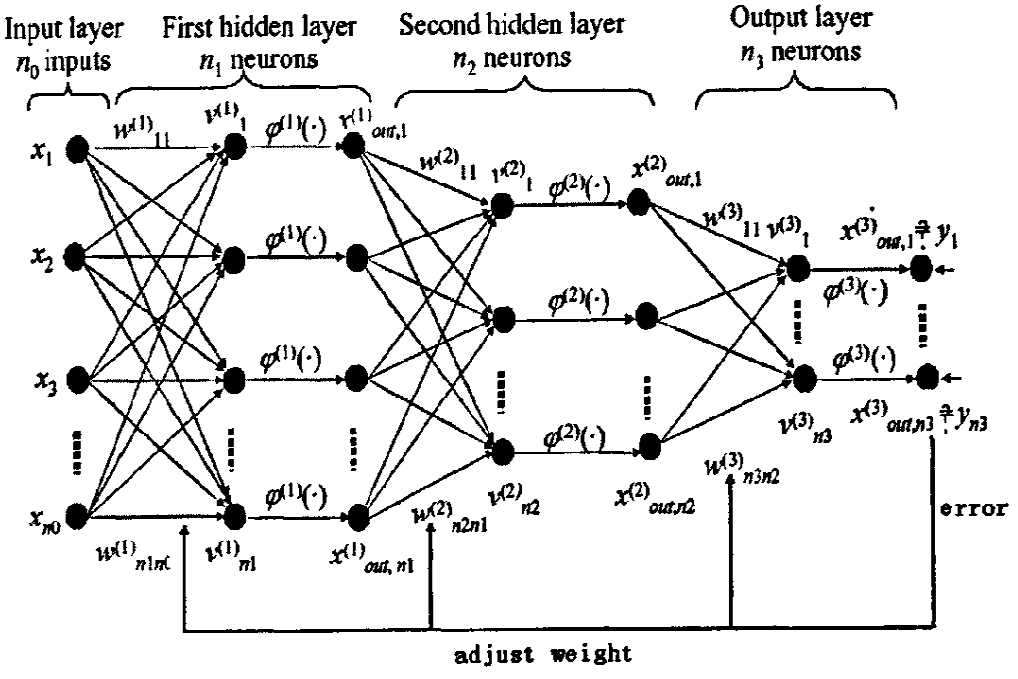

图 2-3 BP 神经网络模型

由图 2—3 的网络模型可见，设各层的激励函数分别为 f_1, f_2, f_3，w 为连接权值符号，φ 为阈值符号，则在正向传播算法中，第一个隐层第 j 个神经元节点的输出为：

$$x_{out,j}^{(1)} = f_1[\sum_{i=1}^{n0} w_{ij} x_i + \varphi^{(1)}(.)] \tag{2.11}$$

第二个隐层第 m 个神经元节点的输出为：

$$x_{out,m}^{(2)} = f_2[\sum_{j=1}^{n1} w_{jm} x^{(1)}{}_{out,j} + \varphi^{(2)}(.)] \tag{2.12}$$

输出层第 k 个神经元节点的输出为：

$$x_{out,k}^{(3)} = f_3[\sum_{m=1}^{n2} w_{mk} x^{(2)}{}_{out,m} + \varphi^{(3)}(.)] \tag{2.13}$$

表示网络的即时误差的能量方程是：

$$E_q=\frac{1}{2}(y_q-x_{out}^{(3)})^T(y_q-x_{out}^{(3)})=\frac{1}{2}\sum_{j=1}^{n_3}(y_{qj}-x_{out,j}^{(3)})^2=\frac{1}{2}\sum_{j=1}^{n_3}e_{qj}^2 \tag{2.14}$$

其中 y_q 表示对于第 q 个模式的网络理想输出，$x_{out}^{(3)}$ 是网络的实际输出。

在反向传播算法中，网络权值的学习规则是：

$$\Delta w_{ji}^{(s)}=-\eta^{(s)}\frac{\partial E_q}{\partial w_{ji}^{(s)}} \tag{2.15}$$

s=1，2，3，4 代表相应的网络层，$\eta^{(s)}>0$ 是学习效率参数。

则对于输出层(输出层的神经元 j)：

$$\Delta w_{ji}^{(3)}=-\eta^{(3)}\frac{\partial E_q}{\partial w_{ji}^{(3)}}=-\eta^{(3)}\frac{\partial E_q}{\partial v_j^{(3)}}\frac{\partial v_j^{(3)}}{\partial w_{ji}^{(3)}}(\text{链式法则}) \tag{2.16}$$

其中 $$\frac{\partial v_j^{(3)}}{\partial w_{ji}^{(3)}}=\frac{\partial}{\partial w_{ji}^{(3)}}(\sum_{i=0}^{n_2}w_{ji}^{(3)}x_{out,i}^{(2)})=x_{out,i}^{(2)}$$

$$\frac{\partial E_q}{\partial v_j^{(3)}}=\frac{\partial}{\partial v_j^{(3)}}\{\frac{1}{2}\sum_{j=1}^{n_3}[y_{qj}-\varphi(v_j^{(3)})]^2\}=-[y_{qj}-\varphi(v_j^{(3)})]\varphi'(v_j^{(3)})\equiv-\delta_j^{(3)}$$

φ' 表示 φ 的一阶导数，$\delta_j^{(3)}$ 定义为第 j 个神经元的局部梯度。因此对于输出层的权值学习法则是：

$$\Delta w_{ji}^{(3)}(k+1)=w_{ji}^{(3)}(k)+\eta^{(3)}\delta_j^{(3)}x_{out,i}^{(2)} \tag{2.17}$$

对于隐藏层(神经元 k 属于输出层，神经元 j 属于隐藏层)

$$\Delta w_{ji}^{(2)}=-\eta^{(2)}\frac{\partial E_q}{\partial w_{ji}^{(2)}}=-\eta^{(2)}\frac{\partial E_q}{\partial v_j^{(2)}}\frac{\partial v_j^{(2)}}{\partial w_{ji}^{(2)}} \tag{2.18}$$

其中 $$\frac{\partial v_j^{(2)}}{\partial w_{ji}^{(2)}}=\frac{\partial}{\partial w_{ji}^{(2)}}(\sum_{i=0}^{n_1}w_{ji}^{(2)}x_{out,i}^{(1)})=x_{out,i}^{(1)}$$

$$\frac{\partial E_q}{\partial v_j^{(2)}}=\frac{\partial}{\partial x_{out,j}^{(2)}}\{\frac{1}{2}\sum_{k=1}^{n_3}[y_{qk}-x_{out,k}^{(3)}]^2\}\frac{\partial x_{out,j}^{(2)}}{\partial v_j^{(2)}}$$

$$=\frac{\partial}{\partial x_{out,j}^{(2)}}\{\frac{1}{2}\sum_{k=1}^{n_3}[y_{qk}-\varphi(\sum_{j=0}^{n_2}w_{kj}^{(3)}x_{out,j}^{(2)})]^2\}\frac{\partial x_{out,j}^{(2)}}{\partial v_j^{(2)}}$$

$$=-[\sum_{k=1}^{n_3}(y_{qk}-x_{out,k}^{(3)})\varphi'(v_k^{(3)})w_{kj}^{(3)}]\varphi'(v_j^{(2)})$$

$$=-(\sum_{k=1}^{n_3}\delta_k^{(3)}w_{kj}^{(3)})\varphi'(v_j^{(2)})$$

$$=-\delta_j^{(2)}$$

因此，对于隐藏层，我们有：

$$\Delta w_{ji}^{(2)}(k+1)=w_{ji}^{(2)}(k)+\eta^{(2)}\delta_j^{(2)}x_{out,i}^{(1)} \tag{2.19}$$

总的来说：

$$w_{ji}^{(s)}(k+1)=w_{ji}^{(s)}(k)+\eta^{(s)}\delta_j^{(s)}x_{out,i}^{(s-1)} \tag{2.20}$$

对于输出层来说：

$$\delta_j^{(s)}=(y_{qj}-x_{out,j}^{(s)})\varphi'(v_j^{(s)}) \tag{2.21}$$

对隐藏层来说：

$$\delta_j^{(s)}=(\sum_{k=1}^{n_{s+1}}\delta_k^{(s+1)}w_{kj}^{(s+1)})\varphi'(v_j^{(s)}) \tag{2.22}$$

BP 算法的实现步骤可归纳如下：

第一步：初始化权值和阈值；

第二步：对样本信息进行训练：

Step1. 给出输入信息向量 X 和目标向量 Y,

Step2. 对输入 X 进行标准化,

Step3. 计算隐含层和输出层的实际输出,

Step4. 求目标向量与实际输出的偏差,

Step5. 若误差在要求内到 Step10,

Step6. 计算隐含层单元误差,

Step7. 求误差梯度,

Step8. 对权值和阈值进行修改，将误差信号沿原来的连接通路返回，通过

修改各层神经元的权值，逐次地向输入层传播去进行计算。再经过正向传播过程，这两个过程反复进行使得误差信号最小，

Step9. 回到 Step3,

Step10. 计算全部误差是否满足要求，若满足则学习结束，否则回到 Step6;

第三步：输入信息向量 X,经仿真计算预测输出向量 Y。

从以上的理论和分析中可以总结出，基于 BP 算法的神经网络具有理论上的完善性和广泛的实用性。

2.3 遗传算法[5][7][16][17]

2.3.1 遗传算法基本概念与特点

遗传算法[12][13](Genetic Algorithm,简称 GA)的概念最早由美国的 Holland 教授在 1975 年提出，是一种模仿生物界的遗传和进化过程而建立起来的一种自适应概率性搜索算法和非导数的优化算法，体现着“生存竞争、优胜劣汰、适者生存”的竞争机制。自从它发展起来，一直是优化领域的研究热点，并且正广泛的应用到各个领域，如模式识别、神经网络、工业控制、智能规划等。特别对许多用传统技术难以解决的复杂优化问题，遗传算法提供了一条行之有效的途径。大量的应用结果已经证明遗传算法具有极强的计算能力。经过多年的发展，它已经成为一种切实可行的、鲁棒性强的优化技术和搜索方法。

遗传算法作为一种快捷、简便、容错性强的算法，与其它寻优算法相比，具有以下特点：

1、遗传算法是对解集的编码进行运算，而不是对解集本身进行运算；

2、遗传算法的搜索始于解的一个种群，而不是某些单个解；

3、遗传算法只用适应度函数来评价解的优劣；

4、遗传算法采用的是概率搜索，而不是路径搜索。

遗传算法具有如下优点：

1、广泛的适用性。遗传算法是模拟生物界而构造的一种自然算法，以概率为主要手段，不涉及复杂的数学知识，亦不关心问题本身的内在规律。因此，遗传算法可以处理任意复杂的目标函数和约束条件。

2、全局优化。由于遗传算法不采用路径搜索，而采用概率搜索，所以是概率意义上的全局搜索。因此，解决的问题无论是否为凸性的，理论上都能获得最优解，避免落入局部极小点。

2.3.2 遗传算法的基本思想和构成要素

2.3.2.1 遗传算法的基本思想与结构

遗传算法的基本思想是从一组随机产生的初始解，即“种群”P（t），开始进行搜索，种群的每一个个体，即问题的一个解，称为“染色体”；遗传算法通过染色体的“适应值”来评价染色体的好坏，适应值大的染色体被选择的几率高，相反，适应值小的染色体被选择的可能性小，被选择的染色体进入下一代；下一代的染色体通过交叉和变异等遗传操作，产生新的染色体，即“后代”C（t）；经过若干代之后，算法收敛于最好的染色体，该染色体就是问题的最优解或近优解。其工作流程如图 2－4 所示。

遗传算法的一般结构可以描述如下：

```
begin
    t←0
    初始化 P（t）
    评价 P（t）
    while（终止条件不满足）do
    begin
        重组 P（t）以产生 C（t）
        评价 C（t）
        从 P（t）和 C（t）中选择 P（t+1）
        t←t+1
    end
end
```

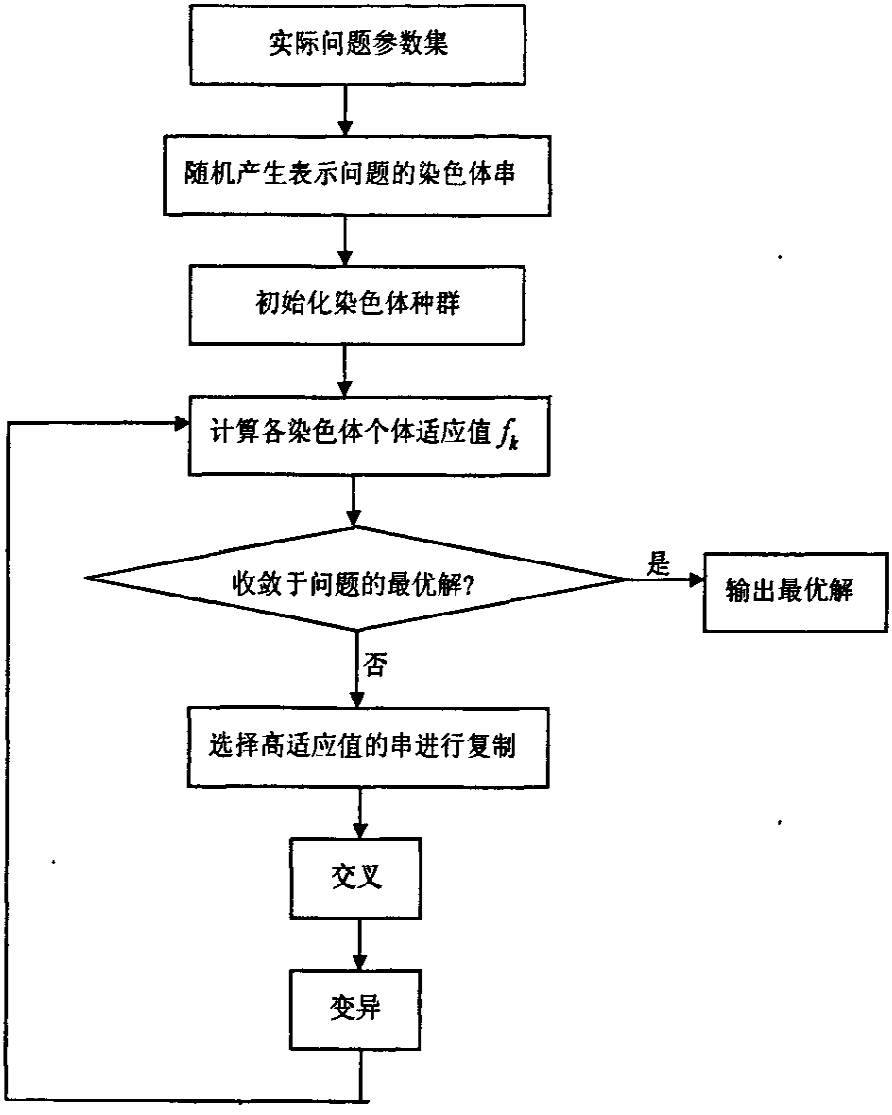

图 2—4 遗传算法的工作流程图

2.3.2.2 遗传算法的构成要素

1. 编码

编码是指将可行解从其解空间转到遗传算法所能处理的搜索空间的转换操作。遗传算法中进化过程是建立在编码机制基础上的，编码对于算法的性能和

效率影响很大

2. 初始化参数的生成

遗传算法的初始化参数有种群规模、交叉率、变异率、进化代数。

3. 适应度函数

适应度函数是用来区分群体中个体好坏的标准，是算法进化过程的驱动力。在具体应用中，适应度函数的设计要结合求解问题本身的要求而定。需要强调的是，适应度函数评估是选择操作的依据，因此适应度函数的选取至关重要，直接影响到遗传算法的收敛速度以及能否找到最优解。

4. 基本遗传操作

遗传算法一般都包括三个基本操作：选择（Selection，或复制Reproduction）、交叉（Crossover，或重组Recombination）和变异（Mutation）。它们构成了遗传算法具备强大搜索能力的核心，是模拟自然选择以及遗传过程中发生的繁殖、杂交和突变现象的主要载体。其中选择环节是优胜劣汰的体现；交叉是一个随机化的重组算子，它使得解空间扩大；变异是一个局部置换算子，它可以防止收敛于局部最优解。

1)选择

遗传算法的基本原理就是达尔文的自然选择原理，选择是遗传算法的推动力，选择是从群体中选择优胜的个体，淘汰劣质个体的操作，就是根据适应度的大小，决定每个个体被选中用来进行繁殖下一代的机率。适应度高的个体，就有较大的几率被选中，能产生较多的后代；而适应度低的个体，则被选中的概率较小，产生的后代自然就较少；甚至有些适应度特别低的个体被淘汰掉。

2)交叉

所谓交叉运算，是指对两个相互配对的染色体按某种方式相互交换其部分基因，从而形成两个新的个体，就是对被选中用来进行繁殖的两个个体，按照一定的概率 p_c（称为交叉率，在产生后代的过程中，被选来用于进行交换的个体数与群体大小的比值），决定它们是否进行交换。遗传算法的交叉操作可视为对生物遗传过程中基因重组的直接模拟。通过交叉，遗传算法的搜索能力得以飞跃提高。基本的交叉算子主要有：单点交叉、两点交叉和多点交叉等。单点交换是最基本的交换算子，其中基因码的交换点位置随机而定，两个父代染色体在交换点处进行互换。在双点交换中，随机的选中两个交换点，在这两点之

间的染色体串部分进行交换生成两个子体。类似的，可以定义n点交换。

3)变异

变异操作用来模拟生物在遗传过程中由于各种偶然因素引起的基因突变。实现变异最常用的方式是以一定的变异率p_m（给定的很小概率值）对染色体的某一基因位进行突变。当遗传算法通过交叉算子已接近最优解邻域时，利用变异算子的局部随机搜索能力，可以加速向最优解收敛。遗传算法中，交叉算子因其全局搜索能力而作为主要算子，变异算子因其局部搜索能力而作为辅助算子。基本的变异算子主要有：基本变异、均匀变异和边界变异等。变异操作可确保群体中遗传基因类型的多样性，以使搜索能在尽可能大的空间中进行，防止种群收敛并滞留在任何局部最优点，从而获得质量较高的优化解。

5、终止条件

遗传算法的终止条件，一般根据所求解问题的不同有不同的终止方式。理想的终止条件是连续若干代不再产生使目标函数更优的变量集合，即不再能繁殖出更优的个体。比较一般的终止条件是：种群中最优的适应度超过了预先的设定值；也可以设定最大遗传代数。

2.3.3 遗传算法的基本理论

遗传算法作为一种模拟生物进化过程这一自然界运动规律的计算方法，其算法本身是朴素的、自然的、直接的，它很难用数学方法来预见其行为特征。到目前为止，还没有能够完全阐明遗传算法运算特征的基础理论。但是，为了解释遗传算法的运算操作和在某种程度上预测遗传算法的进化特性，一些描述遗传算法进化特性的方法，比如“模式定理”、“基因块假设”、“Walsh 模式变换”等，作为遗传算法的基本理论。本文主要介绍前两种。

1、模式定理

模式理论（Schema Theorem）最初由 Holland 提出，被认为是遗传算法的重要里程碑，后来经过诸多学者的不断完善，是研究较广、影响较大的遗传算法数学理论，它将 GA 的运算过程理解为模式操作过程，并从模式运算的角度解释 GA 的性能特点。模式定理：在遗传算子选择、交叉与变异的作用下，具有平均水平之上的、低阶、短定义距以及平均适应度高于群体平均适应度的模

式在遗传算法的进化中将按几何级数增长。根据这一定理，随着遗传算法的逐代进行，那些低阶、短定义距和高适应度的模式(GA 算法的较优解)将越来越多，最后得到的串就是这些模式的组合，因而可以期望性能会逐代得到改善，并最终趋向全局最优点。

2、基因块假设

低阶、短距、高平均适应度的模式(基因块)在遗传算子的作用下相互结合，朝着高阶、长距、高平均适应度的模式的方向演化，最终获得全局最优解。该假设指出，GA 算法具备寻找全局最优解的能力。

2.4 本章小结

本章系统学习和介绍了自回归（AR）模型、人工神经网络和遗传算法的基础理论，从而为时间序列预测的模型建立及算法优化提供了理论依据。

3 NARBP 预测模型研究

3.1 引言

目前，解决非线性时间序列的预测问题的模型和方法有两种：第一，用线性模型和方法近似，这种方法简单，但预测精度不高；第二，采用非线性模型和方法，但对于实际问题，很难建立确定的非线性方程来概化实际模型，因此传统的非线性方法仅对某些问题有用，不具有实用性，而人工神经网络为解决非线性预测问题通用性问题提供了好的途径，但 BP 网络是一种静态的网络，没有时间处理能力，所以不能进行时间序列模式的识别，对于时间序列问题的识别，如何由时变的模式样本通过特征提取，形成具有代表性的训练样本？这往往被认为是困难的。本章就综合了线性自回归模型的阶次易辨识能力与神经网络模型的非线性处理能力，构建了非线性自回归神经网络（NARBP）预测模型。

3.2 AR 模型辨识[21]~[24]

3.2.1 AR 模型阶次辨识

$AR(p)$ 模型 $Y_t=\Phi_1Y_{t-1}+\Phi_2Y_{t-2}+\cdots+\Phi_pY_{t-p}+e_t$ 的阶次 p 的辨识方法有：零极点相消检验法、残差统计特性检验法、F 检验法、赤池检验法、MDL 法等，由于前几种检验方法具有不确定性和人为性的缺点，本文介绍赤池检验法中的 AIC（最小信息准则）和 MDL（最短描述长度准则），这两种方法对于辨识自回归模型的阶次较为有效。

1、确定阶的 AIC 准则

AIC 准则是 Average Information Criterior 的字头，又称平均信息判据，1972 年由 Akaika 提出，是一个基于信息量判据提出的准则。AIC 准则的定义是

$$AIC(\theta_n)=-2\ln L(\hat{\theta},n)+\theta_n \tag{3.1}$$

其中，θ_n 为对应于 n 阶的参数个数

$\hat{\theta}$为参数的极大似然估计

L为似然函数

确定阶的办法是取$AIC(\theta_n)$的极小值，即可求出θ_n，而对应的阶即可得到。即$AIC(\theta_n)$＝极小。这时参数个数为θ_n，阶为n。

*AIC*准则用来确定时间序列模型得到很好的效果，在使用时所需的计算工作量比较大。1977年Soderstrom证明*AIC*准则和*F*检验是渐进等价的。有些学者对*AIC*准则做了全面的研究，认为由它确定的模型的阶往往偏高，其应用范围范围也还需要进一步研究。

2、确定阶的*MDL*准则

*MDL*准则是由Rissanen在1983年提出的另一种信息量准则，又称最小描述长度准则。*MDL*准则的定义为：

$$MDL(\theta_n) = N \ln L(\hat{\theta}, n) + \theta_n \ln N \tag{3.2}$$

确定阶的办法是取$MDL(\theta_n)$的极小值，即可求出θ_n，而对应的阶即可得到。即$MDL(\theta_n)$＝极小。这时参数个数为θ_n，阶为n。

其中N为已观测的时间序列长度，其他符号的代表同*AIC*中的相同符号的代表。*MDL*信息量准则是统计一致的。实验结果显示，对一个短的数据长度，*AR*阶数应选择在$N/3 \sim N/2$范围内才会有更好的结果。

3.2.2 AR模型参数辨识

系统模型参数辨识的方法有最小二乘估计、梯度校正法估计、极大似然估计等方法。最小二乘估计和梯度校正法估计计算简单，而且参数估计量具有许多优良的统计性质，对噪声特性的先验知识要求也不高。而极大似然估计需要构造一个以观测数据和未知参数有关的似然函数，并通过极大化这个函数获得模型的参数辨识，据此，极大似然法通常要求具有能够写出输出量的条件概率密度函数的先验知识，因而，计算工作量较大。但是，极大似然参数估计方法可以对具有有色噪声的系统模型进行辨识，在动态辨识中有着广泛的应用。下面介绍最基本的参数辨识方法－最小二乘估计方法。

最小二乘法是一种古老的方法，最早由高斯（Karl Gauss）提出来的，时

间追溯到 1795 年。当时，高斯为了解决从观测得到的行星轨道的数据推算行星轨道的参数，提出了最小二乘原理。到 20 世纪 60 年代，随着电子计算机的飞速发展，最小二乘法成了系统辨识中参数估计的最基本方法。

AR 模型的最小二乘法参数估计描述如下：

AR(*p*) 模型 $Y_t = \Phi_1 Y_{t-1} + \Phi_2 Y_{t-2} + \cdots + \Phi_p Y_{t-p} + e_t$ 的模型的阶次为 p。现在的问题归结为怎样由已经测得的序列 $\{y(t)\}$ 估计 AR 模型的参数 $\Phi_1, \Phi_2, \ldots, \Phi_p$。

将 *AR*(*p*) 模型写成

$$y_t = \varphi^T_{t-1}\theta + e_t \tag{3.3}$$

式中 y_t 表示 t 时刻的输出，e_t 为噪声，φ_{t-1} 为已知的观测值，θ 为参数向量，分别为

$$\varphi^T_{t-1} = [-y_{t-1} \ -y_{t-2} \ - \ldots - y_{t-p}]$$

$$\theta = [\Phi_1 \ \Phi_2 \ \ldots \Phi_p]^T$$

记 θ 的估计值为 $\hat{\theta}$。估计模型输出计算值为 $\hat{Y}_t$，则有

$$\hat{Y}_t = \varphi^T_{t-1}\hat{\theta} \tag{3.4}$$

上式表示 t 时刻以前的观测对 t 时刻输出的预测，预测误差为

$$\varepsilon_t = y_t - \hat{y}_t = y_t - \varphi^T_{t-1}\hat{\theta} = \varphi^T_{t-1}(\theta - \hat{\theta}) + e_t$$

根据最小二乘原理基本思想：未知参数的最大可能值在各实际观测值与计算值之差的平方乘以度量其精度的数值后所得和达到极小值处，参数估计准则（目标函数）应做如下选择

$$J = \sum_{t=1}^{N} \varepsilon_t^2 = \sum_{t=1}^{N} [\varphi^T_{t-1}(\theta - \hat{\theta}) + e_t]^2 \tag{3.5}$$

由上所述，*AR* 模型参数估计的最小二乘法可以归结为：给定 *AR* 模型结构和估计准则，求估计值 $\hat{\theta}$，使得 $J(\hat{\theta})$ 达到最小。

当观测 N 次，则可列出 N 个 $p+1$ 元 *AR* 方程及目标方程，写成向量形式

$$Y = \Phi\theta + e \tag{3.6}$$

$$J=\varepsilon^{T}\varepsilon=(Y-\Phi\hat{\theta})^{T}(Y-\Phi\hat{\theta}) \tag{3.7}$$

式中

$Y=[y(1)\ y(2)\ \ldots\ y(N)]^{T}$

$$\Phi=\begin{bmatrix}\varphi^{T}(0)\\ \varphi^{T}(1)\\ \ldots\\ \varphi^{T}(N-1)\end{bmatrix}=\begin{bmatrix}-y(0) & -y(-1)\ \ldots & -y(1-p)\\ -y(1) & -y(0)\ \ldots & -y(2-p)\\ \ldots & \ldots & \ldots\\ -y(N-1) & -y(N-2) & \ldots -y(N-p)\end{bmatrix}$$

$e=[e(1)\ e(2)\ \ldots\ e(N)]^{T}$，$\varepsilon=[\varepsilon(1)\ \varepsilon(2)\ \ldots\ \varepsilon(N)]^{T}$

将 J 对 $\hat{\theta}$ 求偏导，当偏导为 0 时即可求得 $\hat{\theta}$ 的最小二乘估计为：

$$\hat{\theta}_{LS}=(\Phi^{T}\Phi)^{-1}\Phi^{T}Y，\ \Phi\text{ 要求满秩} \tag{3.8}$$

3.3 预测模型及结构确定

3.3.1 预测模型选择

本文选择发展比较成熟、应用较广泛的一种前馈神经网络模型——BP 网络模型来构建非线性时间序列预测模型。

3.3.2 BP 预测模型结构确定

确定了网络层数、每层节点数、传递函数、初始权系数、学习算法等也就确定了 BP 网络。确定这些选项时有一定的指导原则，但更多的是靠经验和试凑。

1) 隐层数的确定：

1988 年 Robert Hecht-Nielson 证明了对任何在闭区间内的连续函数，都可以用一个隐层的 BP 网络来逼近，因而一个三层的 BP 网络可以完成任意的 n 维

到m维的映射。但实际中常常无法估计问题真实的复杂程度，所以实际应用中首先考虑神经元节点采用 S 型传递函数的双隐层网络设计的方案，然后选择单隐层网络设计方案进行实验对比，最后根据映射精度来确定隐层数。

2) BP 网络常用转移函数：

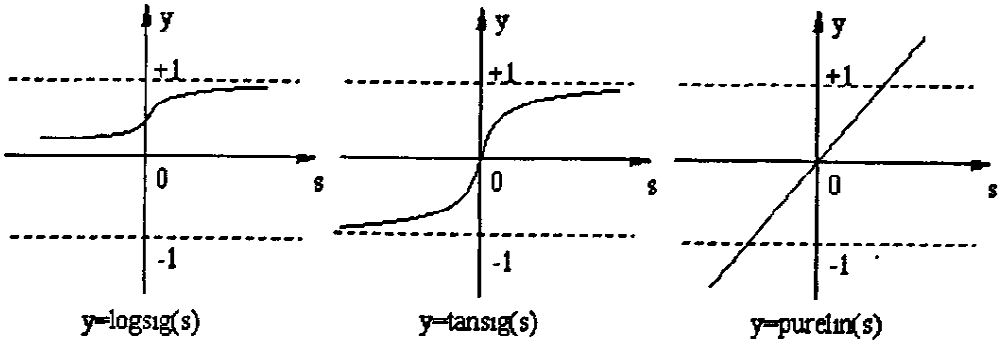

图 3-1 bp 网络常用的传递函数

BP 网络的传递函数有多种。Log-sigmoid 型函数的输入值可取任意值，输出值在 0 和 1 之间；tan-sigmod 型传递函数 tansig 的输入值可取任意值，输出值在-1 到+1 之间；线性传递函数 purelin 的输入与输出值可取任意值。BP 网络通常有一个或两个隐层，该层中的神经元均采用 sigmoid 型传递函数，输出层的神经元则采用线性传递函数，整个网络的输出可以取任意值。各种传递函数如图 3-1 所示。

3) 每层节点数的确定：

对于时间序列的 BP 神经网络来说，时间序列的变量只有一个，所以输出层节点为一个，输入层的神经元节点数由 $AR(p)$ 辨识的 p 决定，而隐层节点数的确定是成败的关键。若隐含层神经元节点数量太少，则网络所能获取的用以解决问题的信息太少；若数量太多，不仅增加训练时间，更重要的是隐层节点过多还可能出现所谓“过渡吻合”(Overfitting) 问题，即测试误差增大导致泛化能力下降，因此隐层神经元节点数的选择直接影响网络对复杂问题的映射能力。目前尚无令人信服的理论来确定隐层神经元数，实际中常选择试算法来确定。一般原则是：在能正确反映输入输出关系的基础上，应选用较少的隐层节点数，以使网络结构尽量简单。可采用网络结构增长型方法进行试算，即先设置较少的节点数，对网络进行训练，并测试学习误差，然后逐渐增加节点数，

直到学习误差不再有明显减少为止。

4）误差的选取

在神经网络训练过程中选择均方误差 MSE 较为合理，原因如下：

① 标准 BP 算法中，误差定义为：

$$E_p = \frac{1}{2}\sum_{j=1}^{m}(t_j^p - y_j^p)^2 \tag{3.9}$$

每个样本作用时，都对权矩阵进行了一次修改。由于每次权矩阵的修改都没有考虑权值修改后其它样本作用的输出误差是否也减小，因此将导致迭代次数的增加。

② 累计误差 BP 算法的全局误差定义为：

$$E = \frac{1}{2}\sum_{p=1}^{P}\sum_{j=1}^{m}(t_j^p - y_j^p)^2 = \sum_{p=1}^{P}E_p \tag{3.10}$$

其中：m 为输出节点的个数，P 为训练样本个数，t 为期望输出，y 为网络实际输出。

这种算法是为了减小整个训练集的全局误差，而不针对某一特定样本，因此如果作某种修改能使全局误差减小，并不等于说每一个特定样本的误差也都能同时减小。它不能用来比较 P 和 m 不同的网络性能，因为对于同一网络来说，P 越大，E 也越大；p 值相同，m 越大 E 也越大。

③ 均方误差 MSE：

$$MSE = \frac{1}{mp}\sum_{p=1}^{P}\sum_{j=1}^{m}(\hat{y}_{pj} - y_{pj})^2 \tag{3.11}$$

其中：m —输出节点的个数，p —训练样本数目，$\hat{y}_{kj}$ —网络实际输出值，y_{kj} —网络期望输出值。均方误差克服了上述两种算法的缺点，所以选用均方误差算法较合理。

3.4 NARBP 预测模型

3.4.1 NARBP 预测模型的建立

目前，已提出了许多非线性随机模型，例如 MARS[25]、投影追逐[26]、阈自

回归模型[27]、双线性模型[28]和 Velterra 级数展开[29]等。而 Lapadus 和 Farber[30]最早把多层前向神经网络用作时间序列预测的 NAR 模型。本文则采用 BP 神经网络逼近 *AR* 模型来构建 NARBP 预测模型，由 *AR* 成熟的辨识理论对时间序列模式进行识别，从时变的模式样本通过特征提取，形成具有代表性的训练样本，然后用代表性的训练样本对 BP 神经网络进行训练，从而综合实现了 *AR* 系统辨识模型的和 BP 网络实现模型的非线性映射能力。

线性 $AR(p)$ 模型对非线性情况的自然推广是非线性自回归 (NAR) 模型

$$Y_t = h(\Phi_1 Y_{t-1} + \Phi_2 Y_{t-2} + \cdots + \Phi_p Y_{t-p}) + e_t \tag{3.12}$$

其中 h 是未知的光滑函数。设 $E(e_t \mid Y_{t-1}, Y_{t-2}, \ldots, Y_1) = 0$，$e_t$ 为 0 均值，且有有限方差 σ^2，即为白噪声。在这些条件下，给定 $Y_{t-1}, Y_{t-2}, \ldots, Y_1$ 的最小均方误差最优预测器为条件均值

$$\begin{aligned} \hat{Y}_t &= E(Y_t \mid Y_{t-1}, Y_{t-2}, \ldots, Y_{t-p}) \\ &= h(Y_{t-1}, Y_{t-2}, \ldots, Y_{t-p}) \\ &= h(\Phi_1 Y_{t-1} + \Phi_2 Y_{t-2} + \ldots + \Phi_p Y_{t-p}) \end{aligned} \tag{3.13}$$

$(t \geq p+1)$

这个预测器的均方误差为 σ^2。

根据上述原理，分别构建一个隐层和两个隐层的 NARBP 预测模型：

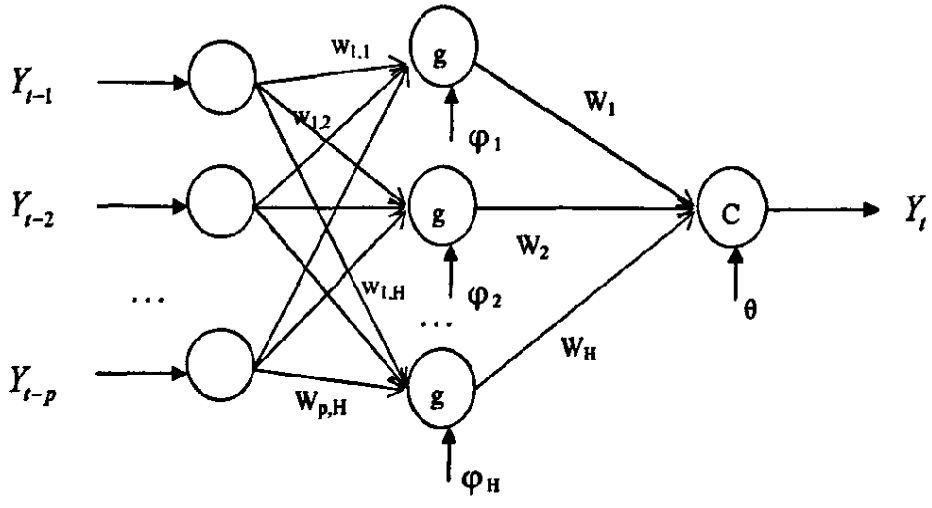

图 2-6 具有一个隐层的 NARBP 预测模型

由图可见：

$$\hat{Y}_t = \hat{h}(Y_{t-1}, Y_{t-2}, ..., Y_{t-p}) = c\{\sum_{i=1}^{H}[\omega_i g(\sum_{j=1}^{p} Y_{t-j}\omega_{i(t-j)} + \varphi_i)] + \theta\} \quad (3.14)$$

其中 p 为输入神经元数，H 为隐层神经元数，$g(.)$ 是隐含层的转移函数，$c(.)$ 为输出层的转移函数，$w_{i(t-j)}$ 为神经元节点 $t-j$ 到神经元节点 i 的连接权值，φ_i、θ 为阈值。训练样本对和检验样本对组成方式为：$[Y_{t-1}, Y_{t-2}, ..., Y_{t-p}]'$，$Y_t$。

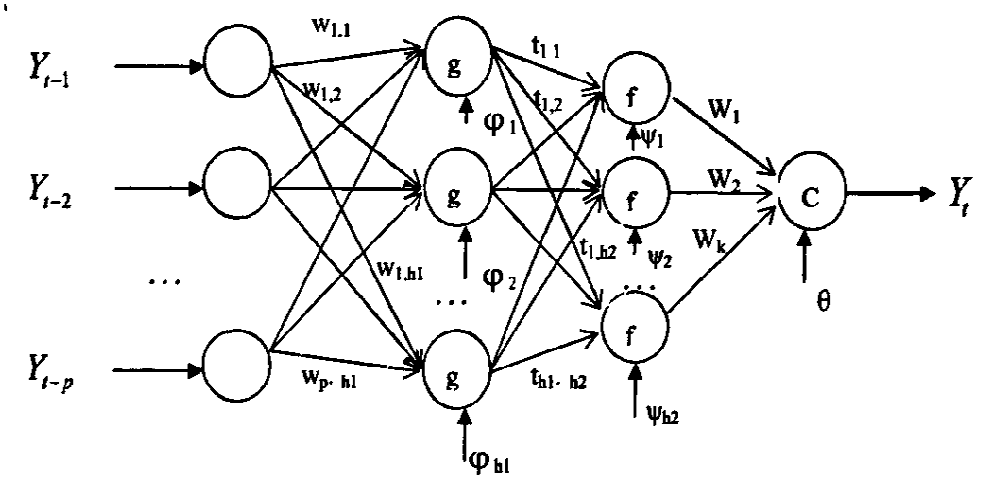

图 2-7 具有两个隐层的 NARBP 预测模型

第一个隐层输出为：$Z_j = g(\sum_{k=1}^{p} Y_{t-k} w_{j(t-k)} + \varphi_j)$ (3.15)

第二个隐层的输出为：$S_i = f(\sum_{j=1}^{h1} Z_j t_{ij} + \psi_i)$ (3.16)

输出层的输出为：$\hat{Y}_t = c(\sum_{i=1}^{h2} S_i w_i + \theta)$ (3.17)

3.4.2 NARBP 预测模型的实现

由上述建立模型的步骤可知，该模型的实现步骤如下：

1、选择时间序列最优 $AR(p)$ 预测器模型 $\hat{Y}_t = \Phi_1 Y_{t-1} + \Phi_2 Y_{t-2} + \cdots + \Phi_p Y_{t-p}$；

2、对 $AR(p)$ 预测器模型 $\hat{Y}_t=\Phi_1 Y_{t-1}+\Phi_2 Y_{t-2}+\cdots+\Phi_p Y_{t-p}$ 的阶次进行辨识；

3、设计 BP 神经网络模型结构；

4、设计 BP 神经网络模型算法；

5、由 $AR(p)$ 的阶次辨识来组织 BP 网络模型的训练样本和检验样本；

6、用训练样本训练 BP 网络，用检验样本检验训练好的 BP 网络，合格后用该网络模型进行预测。

3.5 本章小结

本章详细介绍了时间序列的 AR（p）预测模型的辨识方法和 BP 神经网络预测模型的结构确定方式选取，并基于 AR 模型的辨识能力和 BP 神经网络模型的非线性映射能力，建立了时间序列的非线性自回归 BP 神经网络预测模型(NARBP 预测模型)。最后给出了该模型的实现过程。

4 NARBP预测模型算法优化设计

4.1 引言

NARBP预测模型的主体是BP神经网络，BP神经网络通过多个具有简单处理功能的神经元的复合作用，使网络具有非线性映射能力。尽管BP算法理论具有依据可靠、推导过程严谨、精度较高、通用性较好等优点，但标准BP算法存在以缺点：收敛速度缓慢；非全局搜索能力；容易陷入局部极小值。

1、BP算法是通过训练误差反传修改网络权重来实现对客观对象的识别，对一个非线性方程的识别一般需要几千次训练，但要实现对复杂非线性关系或模糊不确定关系的识别，则需要训练几万次，甚至几十万次，花费几个小时或十几个小时，BP算法收敛速度慢是目前难以克服的困难。

2、非全局搜索能力。BP算法基于误差函数减少的方向，从输出层经隐含层，然后从隐含层到输入层逐步修正网络的各个连接权值和阈值。该算法实质上是单点搜索，不具有全局搜索能力。

3、局部极小点问题。BP神经网络的全局误差E是全体连接权系数$w_{q,j}^{l}$的函数，这就意味着由E构成的连接权空间不是只有一个极小点的曲面，而是存在多个局部极小点的超曲面。BP神经网络学习训练开始时网络的结构参数是随机给定的，因此结构存在一定的随机性。由于初始权值的随机性，利用BP算法训练网络易陷入局部极小点。

由于以上BP算法的缺点是算法本身所固有的，所以在实际应用中，BP算法很难胜任，因此有必要研究如何改进BP网络算法。

4.2 标准BP算法改进[18][32]

目前，许多研究者已经提出了许多改进的BP算法，大致有：

1、利用动量法改进BP算法

标准BP算法实质上是一种简单的梯度下降静态寻优方法，在修正$w(k)$时，

只按照第k步的负梯度方向进行修正，而没有考虑到以前积累的经验，即以前时刻的梯度方向，从而常常使学习过程发生振荡，收敛缓慢。动量法权值调整算法的具体做法是：将上一次权值调整量的一部分迭加到按本次误差计算所得的权值调整量上，作为本次的实际权值调整量，即：

$$\Delta W(n)=-\eta\nabla E(n)+\alpha\Delta W(n-1) \tag{4.1}$$

其中：α为动量系数，通常$0<\alpha<0.9$；η—学习率，范围在0.001～10之间。这种方法所加的动量因子实际上相当于阻尼项，它减小了学习过程中的振荡趋势，从而改善了收敛性。动量法降低了网络对于误差曲面局部细节的敏感性，有效的抑制了网络陷入局部极小。

2、自适应调整学习速率

标准BP算法收敛速度缓慢的一个重要原因是学习率选择不当，学习率选得太小，收敛太慢；学习率选得太大，则有可能修正过头，导致振荡甚至发散。可采用下图所示的自适应方法调整学习率。

调整的基本指导思想是：在学习收敛的情况下，增大η，以缩短学习时间；当η偏大致使不能收敛时，要及时减小η，直到收敛为止。

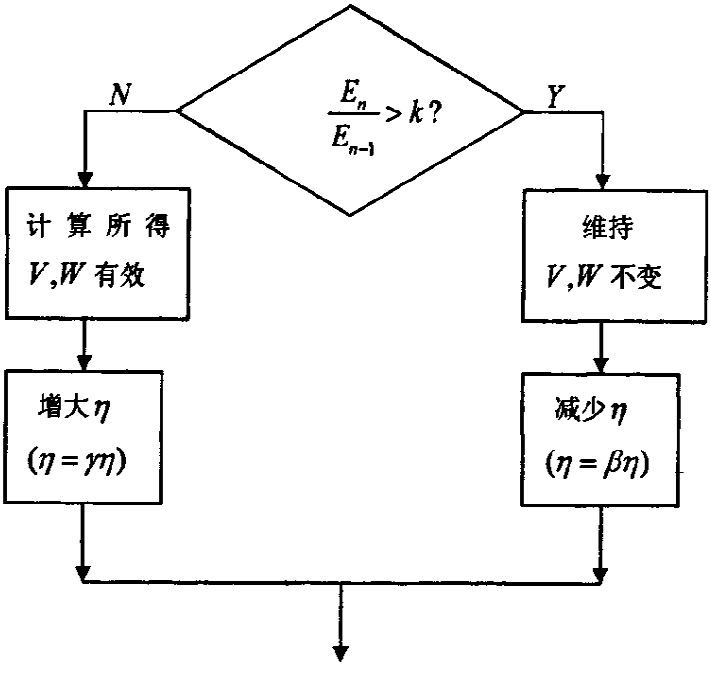

图2-4 自适应学习

3、动量-自适应学习速率调整算法

采用动量法时，BP 算法可以找到更优的解；采用自适应学习速率法时，BP 算法可以缩短训练时间。将以上两种方法结合起来，就得到动量-自适应学习速率调整算法。

4、L-M 学习规则

L-M（Levenberg-Marquardt）算法比前述几种使用梯度下降法的 BP 算法要快得多，但对于复杂问题，这种方法需要相当大的存储空间。L-M(Levenberg-Marquardt)优化方法的权值调整率选为：

$$\Delta w=(J^TJ+\mu I)^{-1}J^Te \tag{4.2}$$

其中：e—误差向量；J—网络误差对权值导数的雅可比（Jacobian）矩阵；μ—标量，当μ很大时上式接近于梯度法，当μ很小时上式变成了 Gauss-Newton 法，Levenberg-Marquardt 算法是建立在优化算法基础上的一种学习算法。在学习过程中μ按照一定的方式进行自适应调整。在迭代过程中，当误差e出现增加的趋势时，μ也会增加，随着μ的增加，J^TJ就可以忽略，此时剩下$\mu^{-1}J^Te$项，但μ增加，则μ^{-1}就减小，这样就使$\mu^{-1}J^Te$项不会因为误差的增大而过分增大，从而使Δw在较短的时间内得到正确的调整，有效缩短学习的时间。当达到希望的误差或达到了最大的训练次数时，网络就会停止。

实践表明 Levenberg-Marquardt 算法比其他算法快，因此本文选择 LMBP 算法作为 BP 网络模型的训练算法。

4.3 GA 与 BP 相结合算法

上述改进措施在很大程度上加快了神经网络的学习速度和逼近能力，但针对易于落入局部最小点问题，单纯的改进神经网络的算法未必会有好的效果，BP 算法及前面提及的改进算法是先在某一区间内随机生成初始权阈值，然后网络进行训练，经过反复的调整直至获得稳定的权阈值。既无从知道初始网络权阈值位于误差曲面的什么位置，也无从知道训练过程中，误差在误差曲面上沿什么样的途径下降，更无法判断最后的训练结果是否真的落入了局部极小点。所以初始值对最终结果有很大影响，它是影响网络最终能达到某一可接受误差的重要因素。因此，网络初始权阈值的优化便成为解决 BP 神经网络易陷局部极

小点的突破口了。近些年来，随着人工智能的另一个分支遗传算法的发展，遗传算法的全局搜索能力逐渐引起研究者的关注，作为全局优化的一种通用方法，将其引入与应用到神经网络中进行有着必然的趋势。

遗传算法是一种全局优化算法，它的目标函数既不要求连续也不要求可微。这种算法实质上不同于单点搜索方法，它是采用同时处理搜索空间中多个个体的方法，有较好的全局搜索性能，有效减少了陷入局部最优解的可能。因而遗传算法的出现使得神经网络的训练有了一个崭新的面貌，将遗传算法应用于神经网络，可使神经网络具有自进化、自适应能力。从而构造出进化的神经网络。

目前，对遗传算法与神经网络的结合方式的研究有几种[5]，大致归纳如下：

1、进化神经网络的权阈值；

2、进化神经网络的拓扑结构；

3、同时进化神经网络的权阈值与拓扑结构；

4、分析神经网络的训练数据与训练结果。

这些结合方式目前还处于探索与研究中，其中研究最为广泛的是利用遗传算法进化神经网络的权阈值系数，但遗传算法也存在着自己的弱点:对于局部的精确搜索能力不强，收敛速度缓慢等。与基于梯度下降法与高斯牛顿法结合的Levenberg-Marquadt(LM) BP 算法相比较，无论从收敛速度还是学习精度都有所逊色[46]。

鉴于 GA 和 BP 算法的特点，本文提出了基于改进遗传算法（IGA）的 LMBP 训练算法:对标准遗传算法加以改进，改善遗传算法的收敛速度和性能。由改进遗传算法优化确定 BP 神经网络的初始权阈值，其基本思想是：首先，在一定的范围内（如[0,1]、[-1,1]）初始化多组（种群规模）网络权阈值；其次，根据遗传算法“适者生存、优胜劣汰”的进化原理，以“网络误差最小”作为进化原则；最后，经过多次迭代，获得一组优化的初始权阈值，该组值是遗传算法所能遍历的权阈值中使网络误差最小的，是给定结构下趋于全局能量最小的网络权阈值。用它代替一般初始权阈值的随机生成，然后应用 LM 算法对 BP 网络进行训练、学习，直至网络收敛。这样充分发挥了遗传算法的全局搜索能力和神经网络的非线性映射能力，从而使神经网络在非线性领域的应用更为广泛。

4.4 SGA 的改进方式[6][38]

4.4.1 SGA 的主要缺点

标准遗传算法（SGA）主要是由基因编码、产生初始群体、评价个体优劣、选择、杂交、变异等一系列演变过程组成，其核心技术包括两方面内容：一是选择方法，选出的解应具有良好的特征或适应值，以便产生优良的后代，同时在解空间应相当分散，以保证求得全局最优解；二是遗传算子应具备良好的计算特征，即一方面要保留原有解的优良特性，另一方面要有恢复丢失的重要信息或优良特征的功能。目前，遗传算法本身在理论上和方法上都尚待完善。因而在应用上受到较大的限制，主要表现在如下几个方面：

1、在 SGA 拓扑结构方面还没有指导性的理论，解群规模、选择方式、收敛判据、杂交变异方式等控制参数均需经验确定，也有可能出现早熟收敛，使得它并不一定总是获得全局最优解；

2、SGA 的全局优化能力部分来自于初始群体的随机生成和杂交算子，但由于群体的有限性与解空间的高维性之间的矛盾，使得搜索的全局性受到很大限制。另外，变异操作也赋予 SGA 一定的全局搜索能力，但由于变异概率常常取得较小，从而使得这些少量变异新个体会被大量老个体“同化”，而且当早熟个体的数量大大超过变异个体的数量时，这种“同化”作用就非常迅速，特别是有限的群体规模和选择操作不可避免地使 SGA 存在“近亲繁殖”、早熟收敛的缺陷；

3、从它的计算机理来看，GA 最适于解决缺乏解析知识的、复杂的、有噪声的和随时间变化的动态系统。目前，SGA 更适于求解组合优化问题，对实变量的优化问题不太适合；

4、SGA 通常需要比较长的计算时间。

所有这些均需进一步深入研究和改进，以完善其理论基础和拓广其应用领域。

4.4.2 SGA的改进方式

在SGA的实际应用中，为了满足具体的应用要求，可从以下各方面对SGA进行改进：

1、改进算法控制参数的设置

SGA中需要选择的算法参数主要有编码串长度，群体规模L，杂交概率p_c，变异概率p_m等，这些参数的设置对SGA的运行性能影响很大。目前许多学者认识到这些算法参数需要随SGA的运行进程而作自适应变化，以便SGA具有更好的鲁棒性、全局最优性和寻优效率。例如，根据操作串的适应度值来调整参数p_c, p_m的大小；p_c, p_m随进化迭代次数而变化。

2、改进编码方式

编码是GA应用中的首要问题，也是GA理论中的基础。从数值优化的观点来看，将优化问题的解表示为数字串的形式称为编码，编码的逆过程称为译码或解码。不合适的编码不仅影响了GA的收敛速度，而且也会极大地影响GA的搜索效率，因此在应用GA时须认真考虑编码方案。选择或设计一种便于GA求解问题的编码方法经常需要对问题有深入的了解。SGA采用二进制数编码，但它的缺点是编码过长，尤其是在处理多变量优化问题时该缺点更为突出。为此，近年来有学者提出使用网格编码(grid coding)、浮点编码(floating coding)以及嵌入编码(embeded coding)等方法。

3、改进初始群体的生成方式

虽然在理论上，初始群体的个体串分布与SGA全局收敛性无关，但实际应用中初始分布却会影响SGA的搜索效率。由于SGA的随机性较大(初始群体、杂交位置、变异位置都是随机选择的)，这使得SGA的收敛速度对初始群体的选样的依赖性较强，收敛不稳定。用随机方法产生初始群体时，要求它们均匀分布于解空间，使SGA从全局范围内搜索最优解。为避免出现初始个体差异不大而导致的GA算法“早熟收敛”现象，若能根据应用问题的具体情况，突破SGA所采用的随机方式产生初始群体，避免产生不可行的初始个体，采用其它优化方法如启发式方法或人工设定方式来生成初始群体，则会有助于提高SGA求解的效率和解的质量。近年来有学者建议将人工智能、专家系统技术引入SGA，

提出了基于知识的初始群体的生成方法。

4、改进适应度函数的定义方式

适应度函数是指导 GA 工作的主要信息来源，它的定义是 SGA 应用中的一个重要问题，对于有约束优化问题就更加重要。在解有约束优化问题时，SGA 必须将其转化为无约束优化问题，一般采用罚函数方法实现。设计的适应度函数，既要使 SGA 的群体尽可能在解空间内，又要能对 SGA 的搜索方向提供正确的指导，此外，该函数的计算应尽可能简便。总之，应综合考虑这些因素来设计该函数，这通常是 SGA 应用中比较困难的一环。另外，在 SGA 运行早期个体差异较大，当采用经典的赌轮选择方式时，后代产生的个数与父代适应度值大小成正比，因此在早期容易使个别好的个体的后代充斥整个群体，造成早熟(premature)，而在 SGA 运行后期个体适应度值趋于一致，优秀的个体在产生后代时优势不明显，使整个群体进化停滞不前(stalling)。因此，为调节 SGA 执行过程中串的复制数目以提高算法的性能，对适应度函数 F 进行适当的拉伸或比例变换(scaling or stretching)是很有必要的。常用的变换有：线性比例变($u(F)=aF+b$)、幂比例变换($u(F)=F^a$)和指数比例变换($u(F)=e^{-bF}$)。

5、改进选择算子操作方式

选择算子操作的主要思想是模拟生物进化过程中的适者生存、优胜劣汰规则。在选择中，低适应度值个体趋向于被淘汰，而高适应度值个体趋向于被复制，所以选择算子的作用效果是提高了群体的平均适应度值，但与此同时也损失了群体的多样性。选择在总体上决定个体向着目标函数值改善的方向前进。选择算子并没有产生新个体，当然群体中最佳个体的适应度值不会提高。改进选择算子操作的目的是为了避免有效基因的缺失，提高 GA 的全局收敛性和搜索效率。选择操作与编码方式无关，而与适应度函数有关。SGA 采用比例选择方式，又称为轮盘赌选择法或蒙特卡罗法，是利用比例于各个个体适应度的概率决定其子孙的遗留可能性。若某个个体 k，其适应度为 f_k，则其被选择的概率为：$p=f_k/\sum_{k=1}^{L} f_k$，当选择的概率给定后，产生[0,1]区间的均匀随机数来决定哪个个体参加分配。显然选择概率大的个体，能多次被选中，它的遗传因子就会在种群中扩大。由于适应度函数的分布特性与具体问题有关，因此一律用 SGA 的比例选择方式是不合适的，可采用适当的适应度函数变换或适应度尺度变换

方法等来予以改进，其中排序选择方式与适应度函数的分布和取值范围无关，所以它是常被采用的改进方式。另外，也可考虑随着GA的运行，选择概率随适应度值的改变作动态调整。显然，最优染色体保存策略也可视为选择算子操作的一种改进方式。

6、改进交叉算子操作方式

交叉算子操作的主要思想是模拟群体内部染色体的信息交换(基因材料交换)机制，也即，通过两个父代串的杂交产生新的子代串，杂交产生的子代串一般与其父代串不同，并且彼此也不相同，每个子代串部分包含两个父代串的遗传物质。杂交算子的作用是可以产生新的个体，从而检测搜索空间中的新点，它有可能使群体中最佳个体的适应度函数值有所提高。同时须降低对有效因式的破坏概率，以免子代不如父代。交叉也是GA区别于其它进化算法的重要特征，采用不同的编码方式，则相应的交叉算子也将不同。杂交算子可取一点、两点、均匀、多点等杂交方式，也可随操作串的适应度函数值作自适应变化。SGA只采用一点交叉算子，而采用多点交叉有利于一个个体承载多个变量问题的信息以提高遗传搜索效率。现在常用的多点杂交主要为两点杂交与均匀杂交(uniform crossover)。目前普遍认为这两种杂交方式都优于单点杂交，而这两者相比谁优谁劣则尚无定论。在交叉算子操作之前须先配对，目前常用的配对策略是随机配对，当然也可考虑根据个体的适应度函数值来配对。

7、改进变异算子操作方式

变异算子操作的主要思想是模拟生物个体的随机突变现象，也即对串的某些基因位值进行随机改变，其作用是增加GA运行中群体的多样性，是对有效基因缺失的一种补救措施，同时也对因选择操作失去的群体多样性的恢复具有潜在的作用。没有变异，GA的解很可能陷入局部极值状态，但是，它在SGA中的作用是第二位的，p_m一般很小，当p_m很大时SGA就退化为蒙特卡罗法，变异算子操作与编码策略有关，变异算子操作可取单点、两点、多点、有效基因、均匀等变异方式，也可随操作串的适应度函数值作自适应变化。

8、算法终止条件的改进

常用的改进方法是基于某种判定标推，判定群体已收敛并不再有进化趋势后作为终止条件。例如，根据连续几代个体平均适应度值(或最佳适应度值)之差小于某个小值，或根据群体中个体适应度的方差小于某个小值，或根据群体

中最佳适应度值与群体平均适应度值十分接近，作为终止条件。由于实际优化问题的复杂性和 GA 本身的运行机理尚不完全被认识清楚，实用中也常常用经验固定进化迭代次数作为算法的终止条件。

9、改进父代替换方式

如按某比例从父代群体 $P(t)$ 中选择性能最佳的个体作为下一代个体 $P(t+1)$。

经过三十多年的发展，GA 的研究开始逐步走向成熟，尤其在数值优化领域得到了广泛应用。目前人们对 SGA 已进行了大量改进，并应用于更广泛的领域。这些改进的 GA 之间及其与 SGA 之间均有很大差别，甚至与其它进化算法的界限也难以区分。实际上，GA 只是提供了一类算法的基本框架，它是一种算法体系，根据不同观点，针对个同类型的问题可以研制不同的遗传算法。

4.5 IGA-NARLMBP 预测模型算法优化设计

4.4.1 IGA 算法设计

1、 IGA 算法控制参数的设计

遗传算法设计所涉及的参数主要有：种群规模、交叉率、变异率、进化代数等等，另外，选择具体算子时，有时还会涉及选择与算子相关的参数。遗传算法参数的选择合理与否直接关系到算法的收敛速度和精度。但是，由于影响参数选择的因素很多，有的与问题本身内涵的客观规律有关，有的与所选择的算子有关，所以在实际应用中，首先要对具体问题进行深入、细致的分析，找出问题的内在规律，将问题合理地模型化，即将实际问题概化为遗传算法可以解决的问题，才能发挥遗传算法的寻优作用。

1）种群规模

一般来说，种群规模选择小遗传算法易陷入局部解，种群规模只有达到一定的数值时，遗传算法才能保证搜索到全局最优解，因此，初始种群规模根据具体问题选择偏大一些。选为 L=80～500。

2）交叉率 p_c 和变异率 p_m

交叉率 p_c 和变异率 p_m 采用 Srinvivas 等学者提出的自适应遗传算法

（Adaptive GA），p_c 和 p_m 能够随适应度自动改变。自适应遗传算法中，p_c 和 p_m 按如下公式进行自适应调整：

$$p_c = \begin{cases} p_{c1} - \dfrac{(p_{c1} - p_{c2})(f' - f_{avg})}{f_{\max} - f_{avg}}, f' \geq f_{avg} \\ p_{c1}, f' < f_{avg} \end{cases} \tag{4.3}$$

$$p_m = \begin{cases} p_{m1} - \dfrac{(p_{m1} - p_{m2})(f - f_{avg})}{f_{\max} - f_{avg}}, f \geq f_{avg} \\ p_{m1}, f < f_{avg} \end{cases} \tag{4.4}$$

其中，$f_{\max}$ 为最大的个体适应度，f_{avg} 为群体的平均适应度，f' 为要交换的两个个体中较大的适应度，f 为要变异个体的适应度。p_{c1}，p_{c2}，p_{m1}，p_{m2} 在(0,1)区间中取值，且 $p_{c1} > p_{c2}$，$p_{m1} > p_{m2}$。

3）进化终止条件

进化计算的终止条件可以从以下几个方面进行控制：

a) Genetations(代数)——当产生的代的数目达到规定的代数的值时，算法停止；

b) Time limit(时限)——在运行时间的秒数等于时限时，算法停止；

c) Fitness limit(适应度限)——当适应度的值对于当前种群的最佳点小于等于适应度限时，算法停止；

d) Stall generations(停滞代数)——在连续繁殖的时间序列中，若长时间不繁殖新代，亦即目标函数无改进，到达停滞代数规定的代数时，则算法停止；

e) Stall time limit(停滞时限)——在秒数停滞等于停滞时限的时间间隔期间，若目标函数无改进，则算法停止。

若这 5 个条件中任何一个条件满足，则该算法停止，这样可以使算法执行速度加快。

2、IGA 算法编码方式的设计

当自变量较多（如 BP 神经网络中的权阈值参数），即染色体的基因较多时，如果用二进制编码，染色体的长度会很长，搜索空间也会很大，这样的遗传搜索相当困难。因此，设计浮点编码的染色体表达方式，即个体的每个基因值用某一范围内的一个浮点数来表示，个体的编码长度等于其决策变量的个数。每

个染色体向量被编码成一个与解向量相同长度的浮点向量。每个元素的初始选择都是在要求的区域里进行，而且遗传与进化运算亦被精心设计，以确保这种约束条件。这样一来，在执行上遗传空间即为问题空间，染色体即为问题变量，直接反映了问题的规律与特性。与基于二进制编码的 SGA 相比，浮点编码存在许多优势：它可以提高运算的精度和速度，特别是在搜索空间较大时更为明显，而且避免了编码带来的附加问题，便于和其他搜索技术相结合。

3、IGA 算法适应度函数的设计

将目标函数转换成适应度函数一般要遵循两个原则：优化过程中目标函数的优化方向（如寻求目标函数的最大值或最小值）与种群进化过程中适应度函数值增加的方向一致；适应度函数大于等于 0。

基于这两个原则，在实际问题中，当目标函数 $f(x)$ 本身就是求其最大值问题时，设计适应度函数为：

$$fit(x)=f(x)+C, C+f_{min}\geq 0 \tag{4.5}$$

当目标函数 $f(x)$ 为求其最小值问题时，设计适应度函数为：

$$fit(x)=C-f(x), C-f_{max}\geq 0 \tag{4.6}$$

如果 $f(x)$ 为非负，则设计适应度函数为：

$$fit(x)=\frac{1}{f(x)+0.001}, f(x)\geq 0 \tag{4.7}$$

若 $fit(x)$ 值较小，则可将其乘以一比例系数进行拉伸。

4、IGA 算法选择算子的设计

选择运算是推动进化的直接推动力。如果选择压力（最佳个体选中的概率与平均选中概率的比值）过大，遗传算法会过早收敛，使搜索落入局部极小点；选择压力过小，搜索过程会非常缓慢。一般来讲，在进化的初始阶段，宜采用较小的选择压力，尽量保持种群个体的多样性；在进化的后期，宜采用较大的选择压力，以提高优良个体的竞争力，使搜索向着最优解的方向发展。

选择算子的特性决定种群的进化趋势，设计选择策略为最佳个体保存法与排序选择法相结合的混合选择策略：保留当前代中最佳的染色体，将其直接进入下一代，其他染色体的生存概率采用排序法。将种群规模为 L 的染色体按适应值由大到小降序排序，排序为 1 的直接复制到下一代中，在其他排序号中，设计排在第 k 位染色体的生存概率为：

$$p_k = \frac{p_{\max}}{1-(1-p_{\max})^L}(1-p_{\max})^{k-1} = t(1-p_{\max})^{k-1} \tag{4.8}$$

其中，$p_{\max}$ 为排序为 $2\sim L$ 中适应值最大的染色体的生存概率。

该混合选择策略既实现了进化过程中某一代的最优解可不被交叉和变异操作所破坏，从而也加快了搜索速度，又体现了排序法以种群中染色体适应值的大小顺序来确定各个染色体的生存概率，而不是以适应值的数值大小来确定，从而克服了直接基于适应值进行选择易陷入局部最优的缺点，而保证了大适应值染色体获得高选择概率，同时阻止某些超级染色体过快地把持遗传过程。

5、IGA 算法交叉算子的设计

浮点编码的交叉算子与经典的交叉算子有很大的不同，它直接在问题空间进行运算。在浮点编码的执行中，每个染色体向量被编码成一个浮点数向量，且各向量在规定的区域内，交叉算子的设计也要保证后代在规定的区域内，这是浮点编码交叉算子设计的首要条件。据此，可以设计算术交叉算子。算术交叉算子对于凸搜索空间有这样的性质：对解空间 D 里的任意两点 x_1, x_2 的线性组合 $ax_1+(1-a)x_2, a\in[0,1]$ 同样是 D 中的一个点，此性质是实现算术交叉的重要保证。其算术交叉的操作过程为：假定种群中的 x_1, x_2 被选为交叉运算的双亲，则由双亲产生后代为：

$$\begin{cases} x_1' = ax_1+(1-a)x_2, a\in[0,1] \\ x_2' = ax_2+(1-a)x_1, a\in[0,1] \end{cases} \tag{4.9}$$

随机数 $a\in[0,1]$，从而保证了 $x_1', x_2' \in D$。该交叉也称为线性交叉。

6、 IGA 算法变异算子的设计

变异运算是指对染色体的某些基因位作改变的操作。浮点编码的变异操作方法是根据浮点数的染色体个体设计的。设计变异的方式为均匀变异。均匀变异由单个的亲体 x 变异产生单个的子代 x'，其操作为：设算子选择了父代 $x=(x_1,x_2,...,x_k,...,x_d)$ 中的一个随机元素 x_k，$k\in(1,2,...,d)$。以问题空间 $[l(k),r(k)]$ 里的一个均匀分布的随机数 x_k' 代替 x_k。该算子在进化的早期，推动可能解在搜索空间内自由移动；在进化计算的后期，允许在搜索过程中离开局部最优解，进行可能的移动，以便搜索到更好的可能解。

4.4.2 BP 神经网络的编码描述方法

考虑到一个输入结点为 i、隐含结点为 h、输出结点为 o 个隐层的 BP 神经网络，BP 网络的权阈值将包含以下几个矩阵：

1) 输入层到隐含层的权值矩阵

$$i\omega=\begin{bmatrix} i\omega_{11} & i\omega_{12} & \cdots & i\omega_{1h} \\ i\omega_{21} & i\omega_{22} & \cdots & i\omega_{2h} \\ \vdots & \vdots & \vdots & \vdots \\ i\omega_{i1} & i\omega_{i2} & \cdots & i\omega_{ih} \end{bmatrix} \tag{4.10}$$

2)隐含层到输出层的权值矩阵

$$l\omega=\begin{bmatrix} l\omega_{11} & l\omega_{12} & \cdots & l\omega_{1o} \\ l\omega_{21} & l\omega_{22} & \cdots & l\omega_{2o} \\ \vdots & \vdots & \vdots & \vdots \\ l\omega_{h1} & l\omega_{h2} & \cdots & l\omega_{ho} \end{bmatrix} \tag{4.11}$$

2) 隐含层的阈值矩阵

$$\varphi=\begin{bmatrix} \varphi_1 \\ \varphi_2 \\ \vdots \\ \varphi_h \end{bmatrix} \tag{4.12}$$

4)输出层的阈值矩阵

$$\theta=\begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_o \end{bmatrix} \tag{4.13}$$

为了利用 GA 进行 BP 网络的权阈值优化，即上述 4 个矩阵 $i\omega, l\omega, \varphi$ 和 θ 的优化，必须考虑将以上四个矩阵转换成方便 GA 操作的染色体串。设种群规模为 L，则任一组完整的 BP 神经网络权阈值的浮点编码映射关系如下所示：

$$W_i=\{iw_i,lw_i,\varphi_i,\theta_i\},i=1,2,\cdots,L \tag{4.14}$$

W_i 相当于一个染色体，这样的染色体共有 L 个。

4.4.3 IGA—NARLMBP 模型优化算法设计步骤

步骤一 参数设定及种群初始化

确定种群规模(pop_ size) L、网络层数和每层神经元数。在算法的仿真过程中，可变换它们的取值，研究其对进化过程的影响。

在某一取值范围内随机产生初始种群 $P\{W_1,W_2,\cdots,W_L\}$，对任一神经网络 w_i (i=1, 2, …, L)即为前面所映射的神经网络的所有权阈组成的浮点编码染色体。在 MATLAB 仿真中使用[a+(b-a)*rand()]函数进行初始化。(b-a)为取值范围长度，以便确定搜索空间；a+(b-a)*rand()可产生一个(a, b) 之间均匀分布的随机数。

步骤二 根据神经网络权阈值的特点设计适应度函数

在此算法中，遗传算法的搜索目标是所有进化代中使网络的误差平方和最小的网络权阈值，而遗传算法只能朝着使适应度函数值增大的方向进化。所以，可以根据产生的权阈值所对应的神经网络，计算出 BP 网络的误差平方和，适应度函数采用误差平方和的倒数，为保证适应值不至于太小，引入一较大系数 10^4，适应度函数的形式如下：

$$E(W_i)=\frac{1}{2n}\sum_{j=1}^{n}(y_j-\hat{y}_j)^2,\ i=1,2,\cdots,L \tag{4.15}$$

$$\mathrm{fit}(W_i)=\frac{10^4}{E(W_i)+0.001},\ i=1,2,\cdots,L \tag{4.16}$$

步骤三 进化运算

采用保留最佳染色体和排序法进行染色体的选择。选择算子的实现过程如下：

1、确定选择概率 p_s；

2、计算标准分布值：

$$t=\frac{p_s}{1-(1-p_s)^L} \tag{4.17}$$

3、计算染色体的选择概率：

$$p = t(1-p_s)^{N(k)-1} \qquad k=1,2,\cdots,L \tag{4.18}$$

式中 N(k) —种群中 k 染色体的适应值在种群中由大到小排列的序号。

4、适应值最大的染色体直接进入下一代，并在下一代种群中的序号为 1：

$$NewR_1 = R_{N(k)=1} \tag{4.19}$$

5、计算染色体的累计选择概率值：

$$q_k = \sum_{i=1}^{k} fit(i), k = 1,2,\cdots,L \tag{4.20}$$

6、在[0,1]区间产生按升序排列的随机数序列 r。

7、对染色体进行选择(下一代种群染色体的开始序号为 2)，若 $q_{k-1} < r_j < q_k$，则选择第 k 个染色体进入下一代中。

步骤四 遗传运算

1、交叉算子

1）由自适应交叉算子确定交叉概率 p_c；

2）确定交叉操作的父代：父代染色体从序号为 2～L 的种群范围内均匀随机产生；

3）进行交叉操作，本文采用线性交叉，即假定种群中的 x_1, x_2 被选为交叉运算的双亲，则按 4.9 式由双亲产生后代 x_1', x_2'。

2、变异算子

1）由自适应变异算子确定变异概率 p_m；

2）变异运算：采用均匀变异，设算子选择了父代 $x = (x_1, x_2, ..., x_k, ..., x_d)$ 中的一个随机元素 x_k，$k \in (1,2,...,d)$。以问题空间 $[l(k), r(k)]$ 里的一个均匀分布的随机数 x_k' 代替 x_k。

步骤五 进化终止判断

经过反复执行适应度评价和选择、交叉、变异等过程直至满足 IGA 算法中的五个收敛条件之一，则终止进化，否则转到步骤二。

步骤六 样本数据的预处理

在采用遗传算法得到的初始权阈值进行 BP 网络训练前，需要对样本数据进

行预处理，这是由于BP神经网络隐层多采用S型函数，这种函数的特点是中间部分对输入的变化比较敏感，而两端对输入的变化反应迟钝，这就意味着转移函数对取值范围在两端的数据不如中间部分映射得精确。为了解决这个问题，常常将网络的输入样本进行归一化处理。本文采用如下的归一化处理公式，将样本处理到[-1,1]区间内。

$$T = 2 * \frac{(X - X_{min})}{(X_{max} - X_{min})} - 1 \tag{4.21}$$

式中： X——原始数据

X_{max}、X_{min} ——原始数据的最大值和最小值

T——变换后的数据，也称之为目标数据

网络运行后，数据的还原公式为：

$$X = 0.5 * (T + 1) * (X_{max} - X_{min}) + X_{min} \tag{4.22}$$

步骤七 BP网络训练

经过遗传操作后，选出适应度最大的个体，可反映出最优的网络权阈值的分布，将染色体中的编码值赋给BP网络，作为BP网络的初始权阈值，再使用本文采用前面已经介绍过的Levenberg-Marquardt改进BP算法对此进行进一步的训练，即在遗传算法全局寻优的基础上进行BP算法局部寻优，进而计算出网络的最优解。

4.4.4 IGA-NARLMBP模型算法流程图

根据上述的算法设计步骤，总结IGA-NARLMBP预测模型优化算法流程图如下：

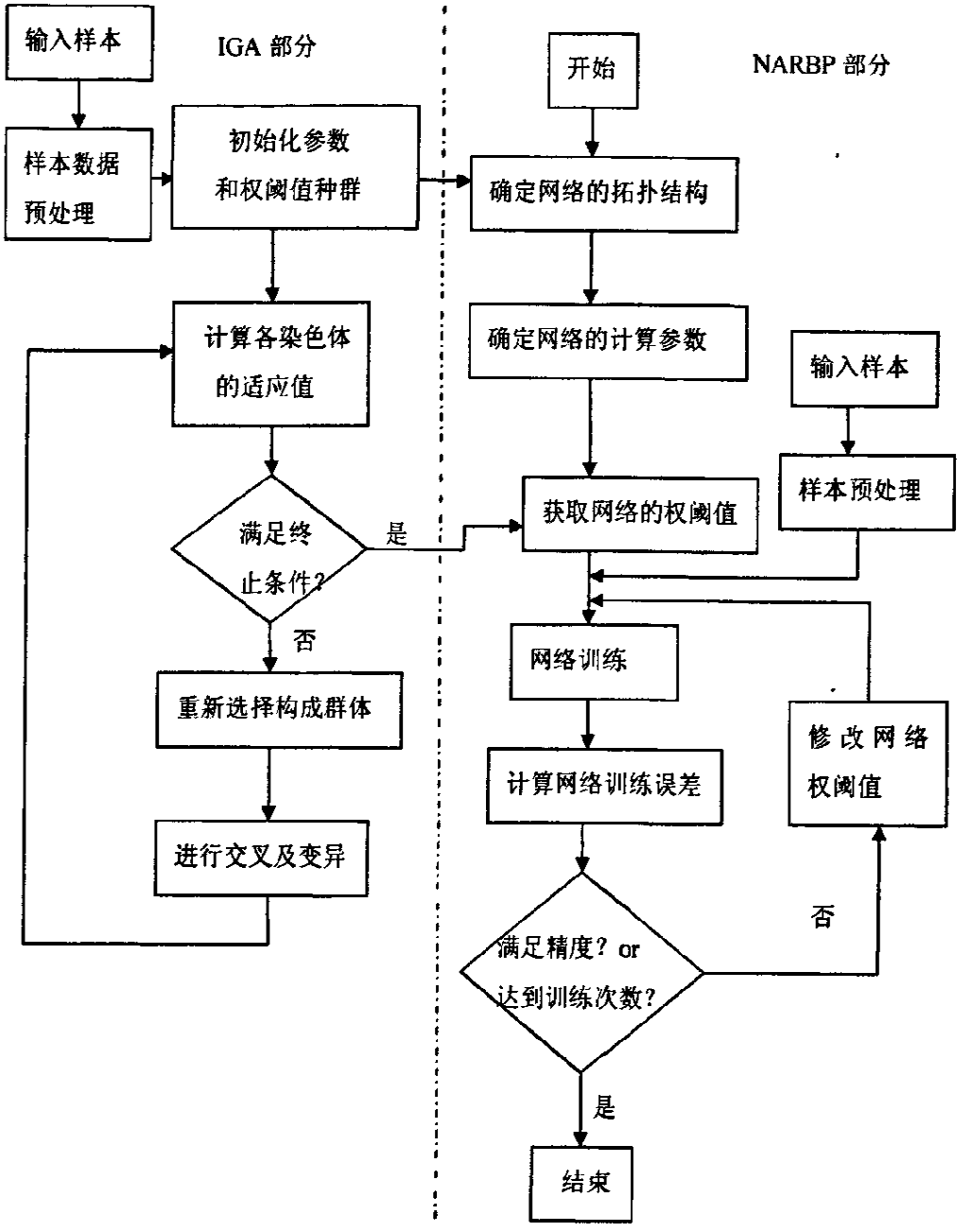

图 4.1　IGA-LMNARBP 预测模型算法框图

4.5 本章小结

本章分析了上章所构建的 NARBP 神经网络预测模型算法的缺陷，在提高网络收敛速度方面，提出了采用改进的 BP 算法—LMBP 算法；从搜寻到全局最优点的角度，提出了采用遗传算法和 BP 算法的结合算法，并对标准遗传算法(SGA 算法)进行改进，设计了快速遗传算法(IGA 算法)，最后给出了 IGA 算法和 LMBP 算法相结合的优化算法对上章所构建的预测模型进行算法优化的步骤。

5 应用与仿真研究

5.1 引言

本文根据研究生期间所作课题——都江堰灌区渠首岷江上游来水预测为应用实例，进行线性 AR 预测模型、本文构建的 NARBP 模型、本文提出的基于改进遗传算法和 LMBP 训练算法的 NARBP 神经网络预测模型的预测仿真实验，进而对各模型进行比较。

5.2 课题背景简介

都江堰灌区渠首(如图 5.1)所在位置位于呈扇形伸展的成都平原的顶部，是整个都江堰灌区的至高点，上游是岷江所经过的高山峡谷，下游是广阔的成都平原。良好的地理位置使其不仅可以扼制住刚出峡谷的岷江水势又可因地势高而控灌整个都江堰灌区。都江堰灌区渠首共有 6 个进口，分别是蒲阳河进口、柏条河进口、走马河进口、江安河进口、沙沟河进口和黑石河进口。这六条河流供应着整个川西平原的生活、生产、生态用水。

渠首岷江上游来水预测的目的为如何向六条河流进口分配水资源提供可分配水量这一依据。在紫坪埔水库没有完全发挥计划分水之前，渠首来水完全是靠岷江上游的自然来水，因此具有较强的随机性，但是随机性又是相对的，在对于多年资料进行研究后可以发现来水量是有一定规律可循的，这使得渠首来水量具有了可预测性。近年来，都江堰管理局一直采用经验法和灰色理论混合预测法来进行来水预测，其需要经验丰富的专家，每年都要进行一次手动计算，通用型不强，效率不高，因此有必要将渠首岷江上游来水预测方法科学化、信息化、智能化。

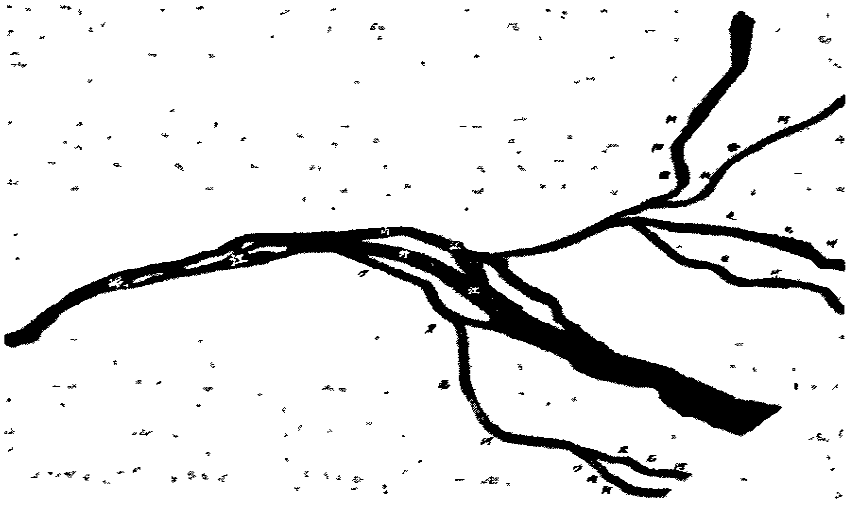

图 5.1 都江堰灌区渠首图

5.3 数据分析

都江堰渠首岷江上游来水的历史观测资料有 68 年（1937 年到 2004 年）每旬的旬平均来水流量，总共有 68×36＝2448 个数据。在 MATLAB 环境下从不同角度绘制图形，以利于观察和把握来水规律。

逐年的来水流量曲面如图 5.2:

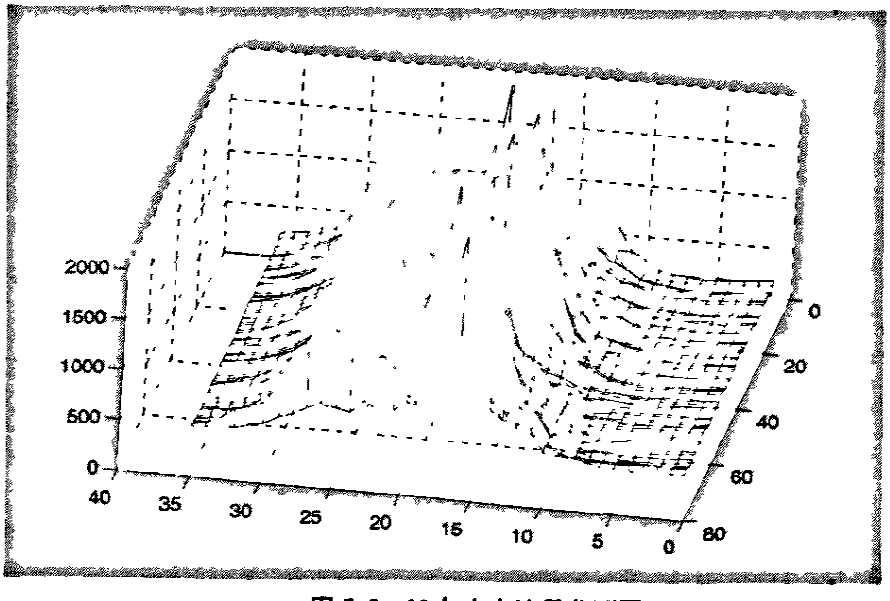

图 5.2 68 年来水流量曲面图

2000－2004 年 5 年的来水流量时序图如图 5.3：

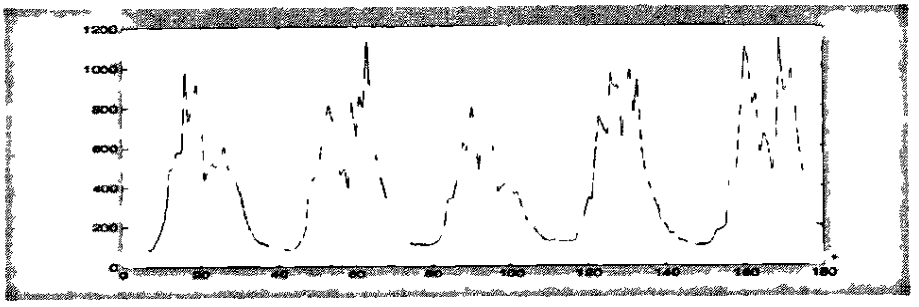

图 5.3 2000－2004 年来水流量时序图

历年第 1、2 月各旬流量图如图 5.4：

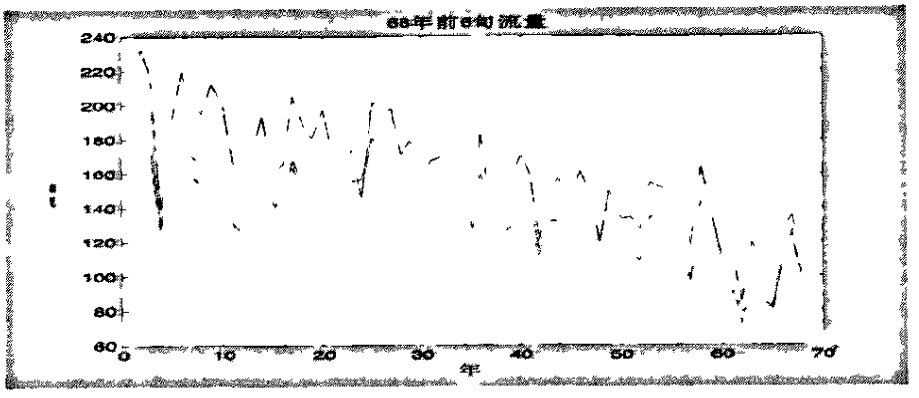

图 5.4 历年第 1、2 月各旬流量图

历年第 6、7 月各旬流量图如图 5.5：

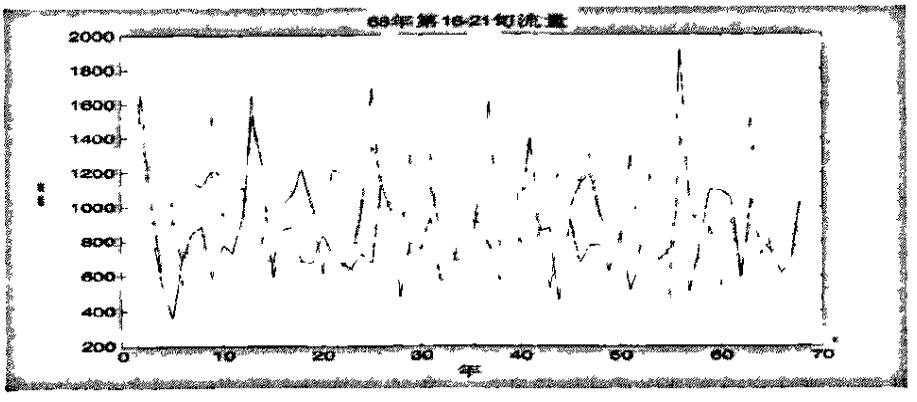

图 5.5 历年第 6、7 月各旬流量图

历年第11、12月流量图如图5.6：

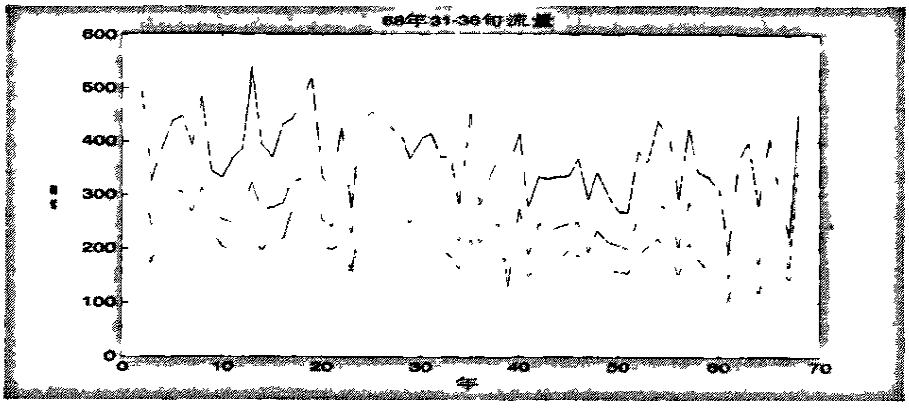

图5.6 历年第11、12月各旬流量图

从这几组图中可以分析出历年来水流量的一些特点：

1、历年的来水量规律相似：枯水期（每年的1－4月、10－12月间）各旬来水量少且较平稳，而丰水期（每年的5－9月）各旬来水量大且波动较大。

2、历年同一旬的来水量在枯水期有下降的趋势，而同一旬的来水量在丰水期波动大，规律性不强。

3、历年来水量的时间序列是非线性的。

从数据分析中可以看出，历年的流量时间序列是有规律的，也就是说，是可预测的，另外，该时间序列是非线性的，对于非线性的时间序列的预测可采用线性模型近似预测与构建合适的非线性模型进行更好的预测。本文分别构建自回归(AR)预测模型、本文构建的NARBP神经网络预测模型和基于改进遗传算法和LMBP算法的NARBP神经网络预测模型进行预测。

5.4 模型构建及仿真

5.4.1 水文预测精度评定标准[40]

SL250－2000《水文情报预测规范》对预测项目的精度评定规定如下：

1、过程预测许可误差取预见期内实测变幅的20%，一次预测的误差小于许可误差时，为合格预测。合格次数与预测总次数之比的百分数为合格率，表

示多次预测总体的精度水平。合格率按下列公式计算：

$$QR=\frac{n}{m}\times 100\% \tag{5.1}$$

式中 QR——合格率

n——合格预测次数

m——预测总次数

2、预测项目的精度按合格率或确定性系数的大小分为三个等级。精度等级按表 5.1 规定确定。

表 5.1 预测项目精度登记表

精度等级	甲	乙	丙
合格率（%）	$QR\geq 85.0$	$85.0\geq QR\geq 70.0$	$70.0\geq QR\geq 60.0$

5.4.2 AR 模型

1、数据预处理

将数据处理为广义平稳随机序列，即去除数据趋势项，将原始数据序列零均值化。设原始数据序列为 flow，在 MATLAB 中，可采用 nflow=detrend(flow) 命令来实现。

2、模型阶次和参数辨识

采用 MATLAB 系统辨识工具箱对时间序列 nflow 进行辨识，辨识结果为：MDL 准则辨识的模型阶次为 36 阶，AIC 准则辨识的模型阶次为 76 阶。辨识的参数见下述模型。

3、AR 预测模型

$$A(q)y(t) = e(t) \tag{5.2}$$

36 阶预测模型：

$$\begin{aligned}A(q) = 1 &- 0.5764\,q^{-1} - 0.001114\,q^{-2} - 0.02694\,q^{-3} + 0.002817\,q^{-4}\\ &+ 0.01858\,q^{-5} + 0.01798\,q^{-6} + 0.01606\,q^{-7} - 0.01835\,q^{-8}\\ &- 0.00403\,q^{-9} + 0.01104\,q^{-10} - 0.006824\,q^{-11} + 0.03025\,q^{-12}\\ &- 0.00141\,q^{-13} + 0.01829\,q^{-14} + 0.02126\,q^{-15} + 0.01305\,q^{-16}\\ &+ 0.01972\,q^{-17} + 0.01861\,q^{-18} + 0.01488\,q^{-19} + 0.01106\,q^{-20}\end{aligned}$$

+ 0.01212 q^-21 + 0.006928 q^-22 + 0.004545 q^-23 + 0.0007283 q^-24
- 0.003493 q^-25 - 0.01178 q^-26 + 0.02618 q^-27 - 0.02232 q^-28
- 0.008863 q^-29 + 0.03136 q^-30 + 0.009914 q^-31 + 0.003038 q^-32
- 0.04684 q^-33 - 0.09139 q^-34 - 0.04505 q^-35 - 0.124 q^-36 (5.3)

76 阶预测模型：

A(q) = 1 - 0.545 q^-1 + 0.002157 q^-2 - 0.04353 q^-3 - 0.01261 q^-4
+ 0.006219 q^-5 + 0.02005 q^-6 + 0.02114 q^-7 - 0.02198 q^-8
+ 0.0004572 q^-9 + 0.001622 q^-10 - 0.0108 q^-11 + 0.03161 q^-12
- 0.005422 q^-13 + 0.01089 q^-14 + 0.008393 q^-15 + 0.001343 q^-16
+ 0.009969 q^-17 + 0.009536 q^-18 + 0.008112 q^-19 + 0.004553 q^-20
+ 0.007391 q^-21 + 0.006147 q^-22 + 0.006578 q^-23 + 0.006031 q^-24
- 0.003845 q^-25 - 0.01263 q^-26 + 0.02617 q^-27 - 0.02884 q^-28
- 0.008257 q^-29 + 0.03344 q^-30 + 0.009924 q^-31 + 0.01608 q^-32
- 0.03029 q^-33 - 0.06307 q^-34 - 0.02395 q^-35 - 0.1004 q^-36
+ 0.01396 q^-37 - 0.001053 q^-38 + 0.05756 q^-39 - 0.04698 q^-40
+ 0.02973 q^-41 - 0.02317 q^-42 - 0.0009482 q^-43 + 0.03721 q^-44
- 0.03171 q^-45 + 0.02611 q^-46 - 0.007514 q^-47 + 0.005925 q^-48
+ 0.006537 q^-49 + 0.007512 q^-50 + 0.01138 q^-51 + 0.008287 q^-52
+ 0.009034 q^-53 + 0.01403 q^-54 + 0.006922 q^-55 + 0.01451 q^-56
+ 0.006292 q^-57 + 0.01453 q^-58 - 0.01008 q^-59 - 0.02073 q^-60
+ 0.006282 q^-61 + 0.03917 q^-62 + 0.00826 q^-63 + 0.01197 q^-64
- 0.009761 q^-65 - 0.01098 q^-66 + 0.007487 q^-67 + 0.02322 q^-68
- 0.06391 q^-69 - 0.001233 q^-70 - 0.04368 q^-71 - 0.06837 q^-72
- 0.07259 q^-73 + 0.06968 q^-74 - 0.008395 q^-75 + 0.05325 q^-76 (5.4)

4、仿真

以 00－04 年(5 年)来水流量数据作为检验数据，预测情况如下：

1)36 阶自回归预测模型 5 次预测，每次结果相同，预测合格率为 68.3333%，如图 5.7：

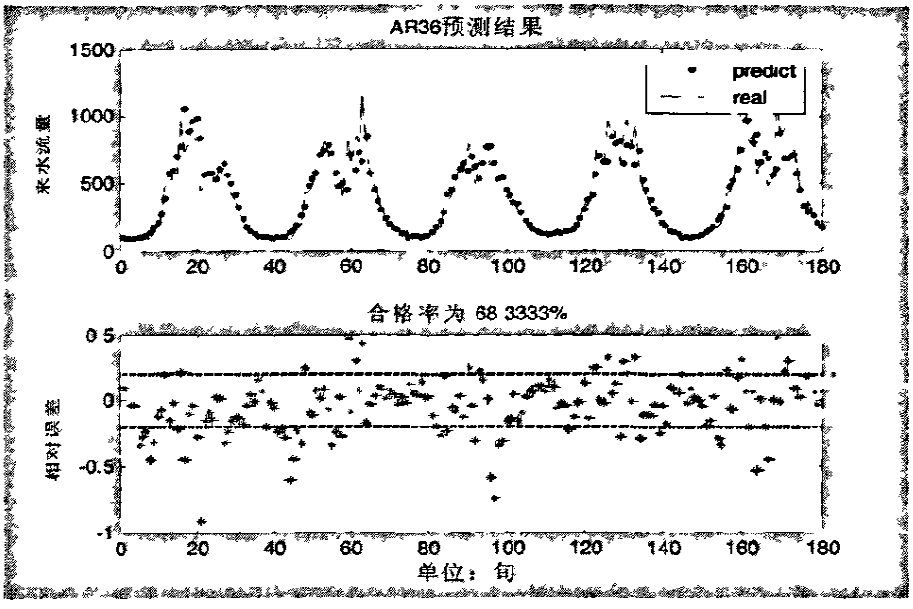

图 5.7 AR36 预测模型预测情况图

2)76 阶自回归预测模型 5 次预测，每次结果相同，预测合格率为 71.6667%，如图 5.8：

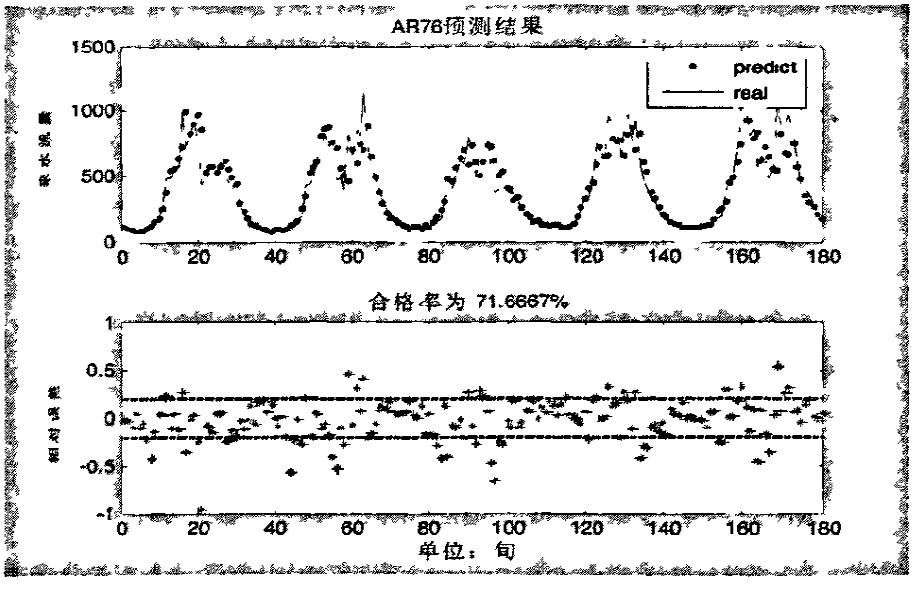

图 5.8 AR76 预测模型预测情况图

5.4.3 NARBP 模型

1、数据预处理

采用公式$T=2*\frac{(X-X_{min})}{(X_{max}-X_{min})}-1$对原数据序列进行归一化处理。

2、模型辨识与建立

采用 AR 模型 MDL 准则辨识模型阶次为 36 阶，构建输入层 36 个神经元，1 个输出神经元的 BP 神经网络，通过试算，确定采用单隐层结构 36－18－1。隐层节点的传递函数选为 tansig 双曲正切 S 型函数，输出层节点的传递函数选为 purelin 函数，训练算法选为 LMBP 算法。参数选择为 MSE＝0.016，训练最大次数取为 600。

3、模型仿真

1）取 1980－1999 年（20 年）数据序列组成训练样本进行 NARBP 网络模型训练，然后取 2000 年－2004 年（5 年数据）序列作检验样本进行检验。仿真结果如下：

第一次训练与预测

训练性能图如图 5.9：

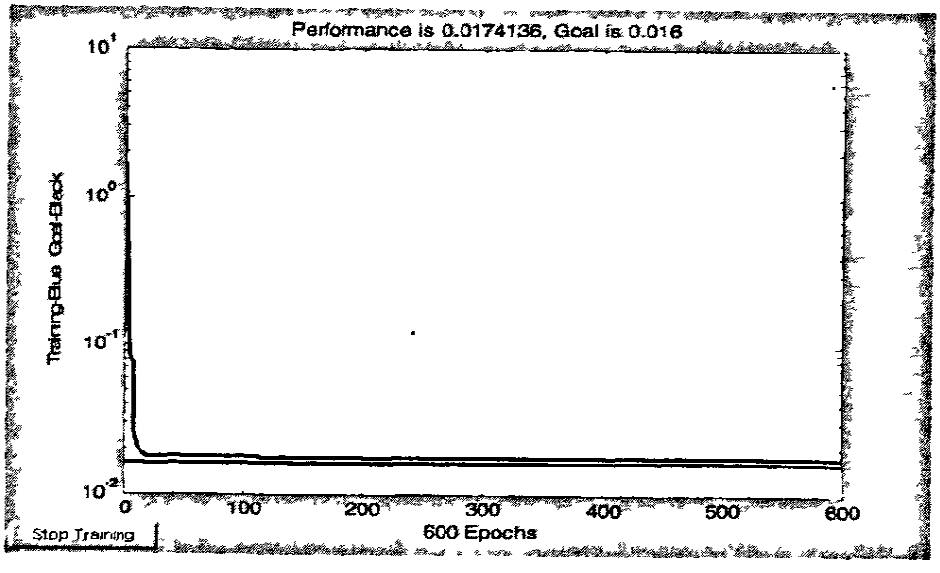

图 5.9 NARBP 预测模型训练性能图

预测情况图如图 5.10：

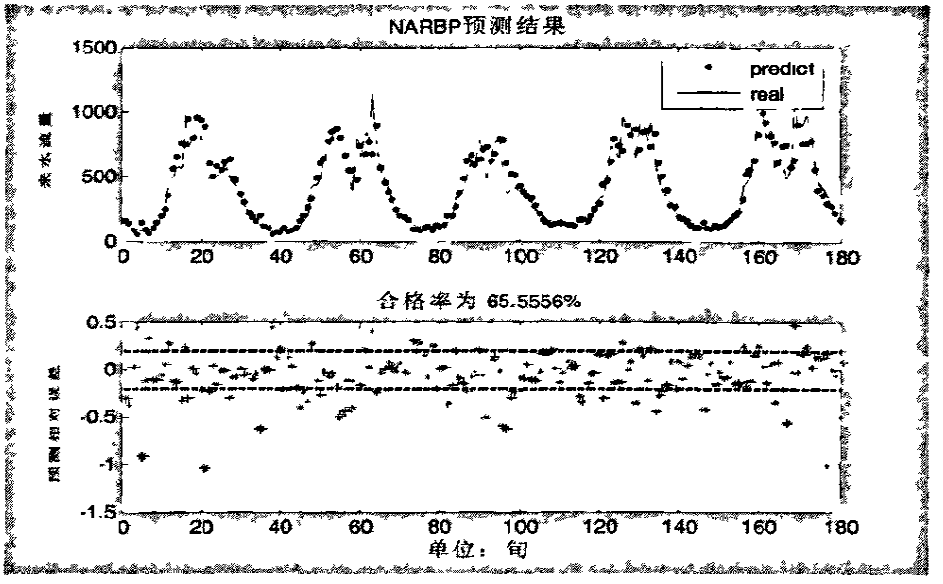

图 5.10 NARBP 预测模型预测情况图

第二次训练与预测

训练性能如图 5.11

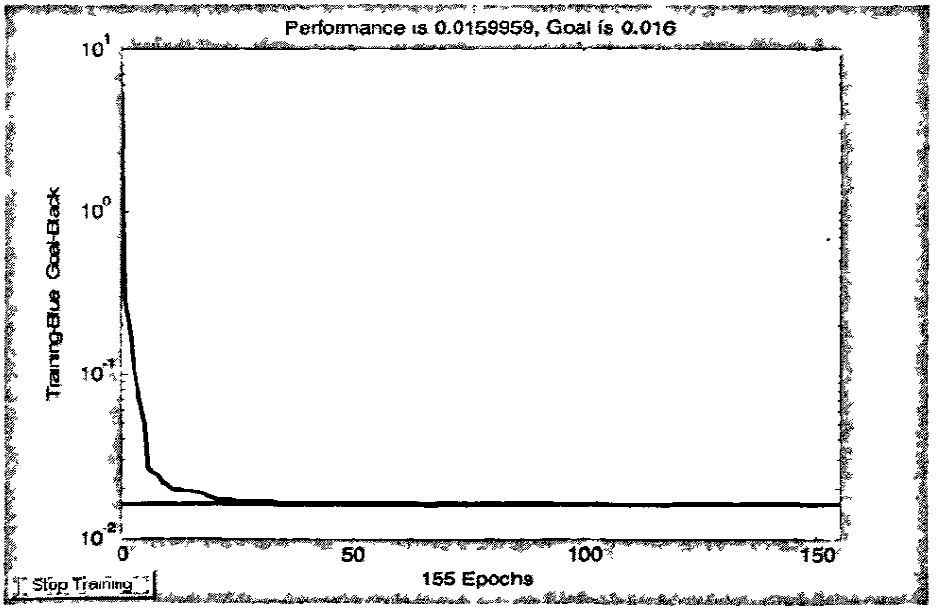

图 5.11 NARBP 预测模型训练性能图

预测情况如图 5.12：

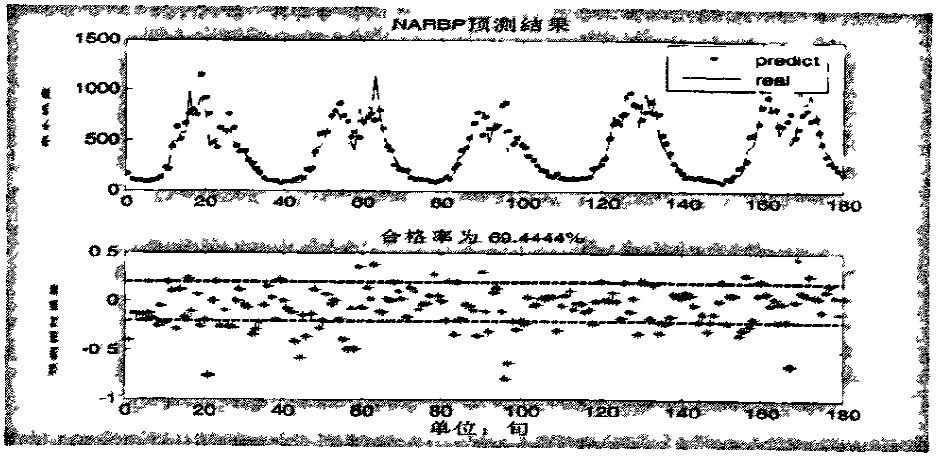

图 5.12 NARBP 预测模型预测情况图

可见，第一次训练 600 次还未达到性能指标，预测合格率为 65.5556%;第二次训练 155 次达到性能指标，预测合格率为 69.4444%。分别训练和预测 5 次，发现每次结果不同，合格率从 55～72%不等。

3）取 1960－1999 年（40 年）数据序列组成训练样本进行 NARBP 网络模型训练，然后取 2000 年－2004 年（5 年数据）序列作检验样本进行检验。仿真结果如下：

第一次训练和预测

训练性能如图 5.13：

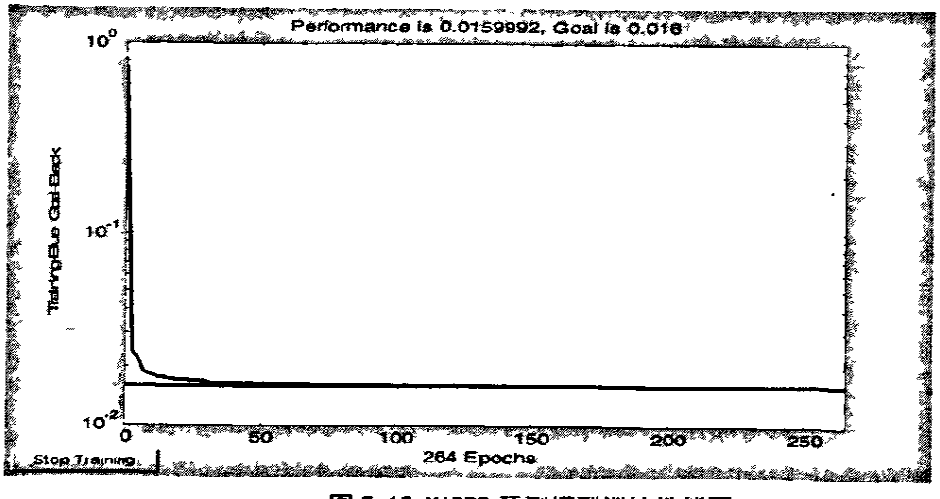

图 5.13 NARBP 预测模型训练性能图

预测情况如图 5.14：

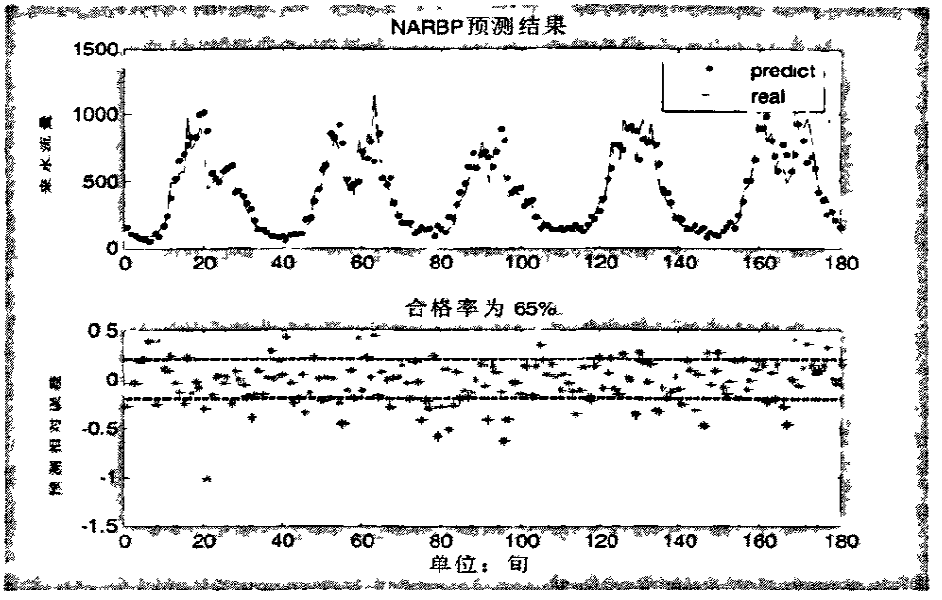

图 5.14 NARBP 预测模型预测情况图

第二次训练和预测

训练性能如图 5.15：

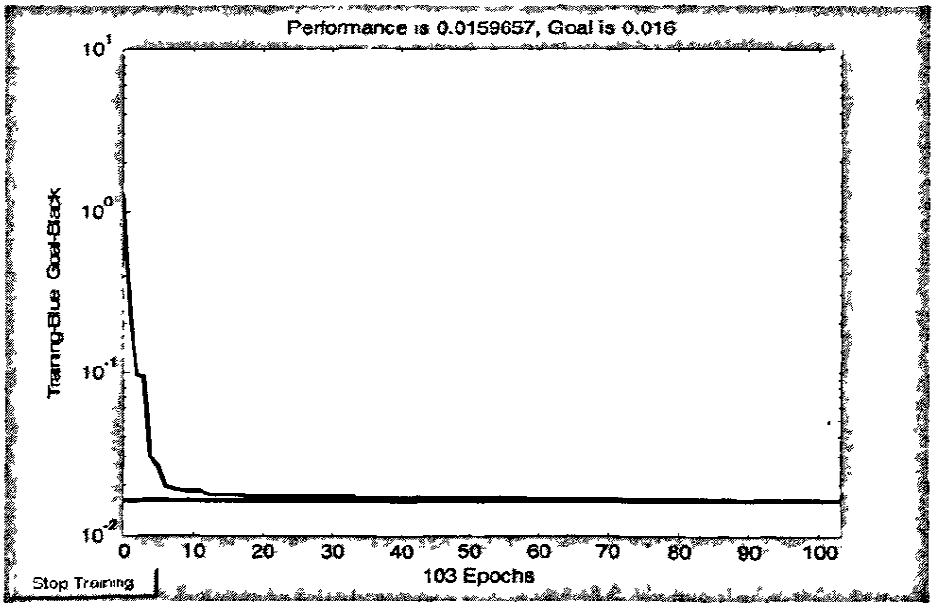

图 5.15 NARBP 预测模型训练性能图

预测情况如图 5.16:

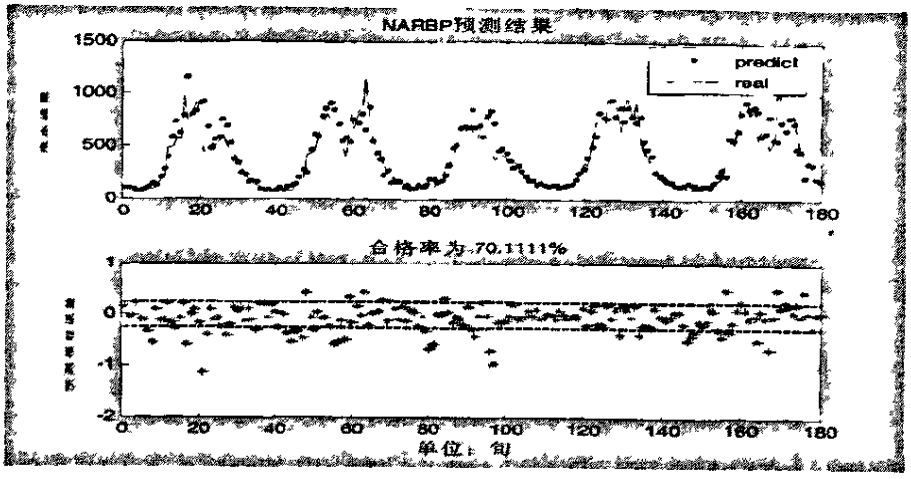

图 5.16 NARBP 预测模型预测情况图

可见，第一次训练 264 次达到性能指标，预测合格率为 65%；第二次训练 103 次达到性能指标，预测合格率为 70.1111%。分别训练和预测 5 次，发现每次结果不同，合格率从 55～72%不等。

3) 取 1937－1999 年（63 年）数据序列组成训练样本进行 NARBP 网络模型训练，然后取 2000 年－2004 年（5 年数据）序列作检验样本进行检验。仿真结果如下：

第一次训练和预测

训练性能如图 5.17:

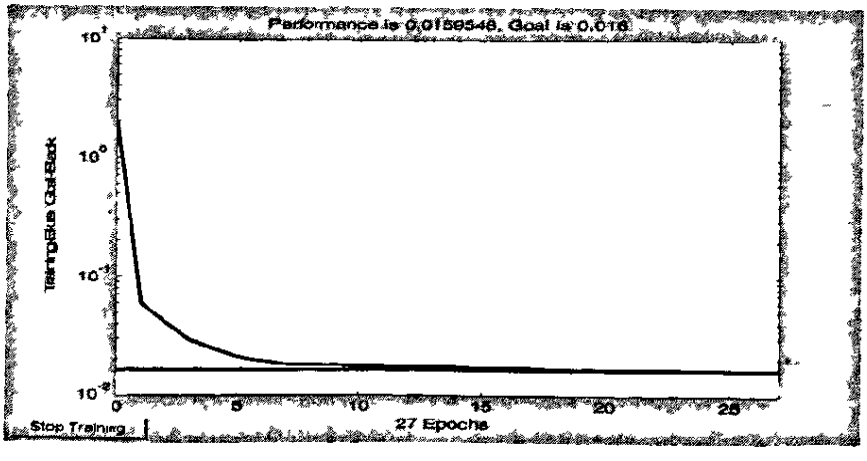

图 5.17 NARBP 预测模型训练性能图

预测情况如图 5.18:

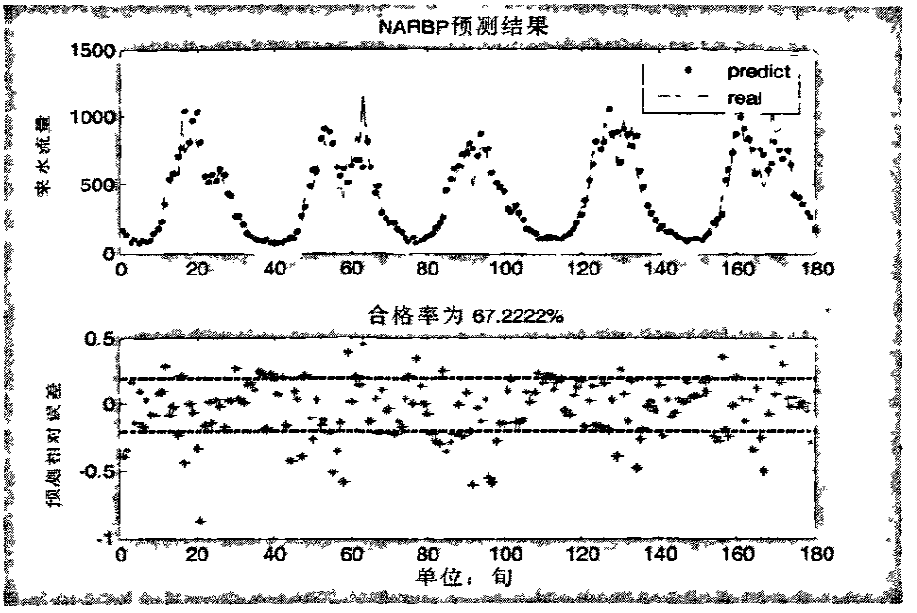

图 5.18 NARBP 预测模型预测情况图

第二次训练和预测

训练性能如图 5.19:

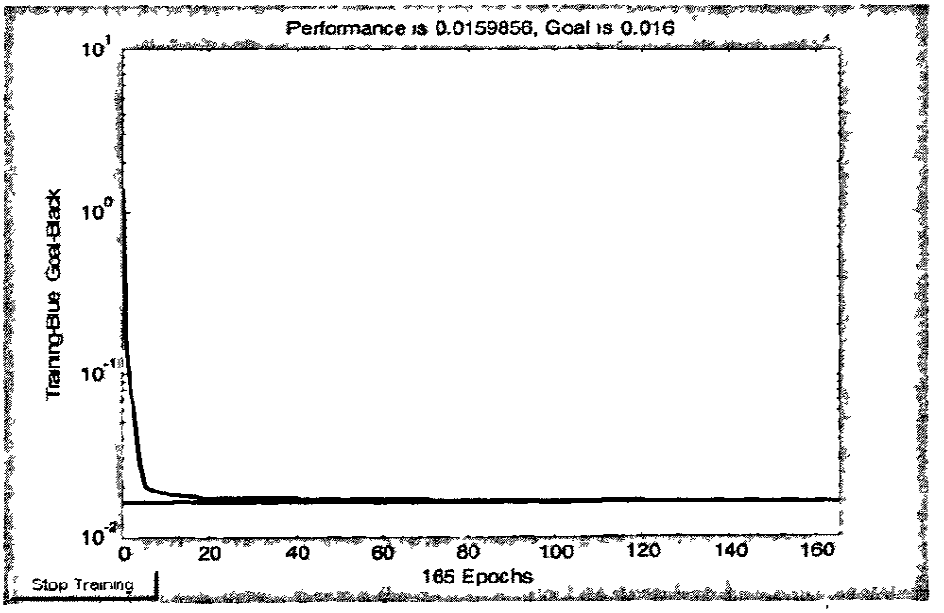

图 5.19 NARBP 预测模型训练性能图

预测情况如图 5.20:

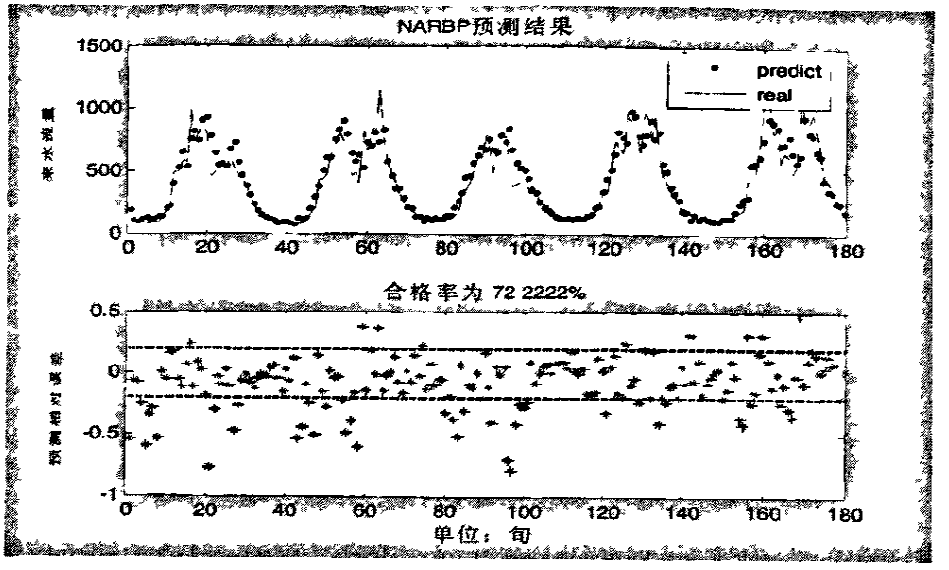

图 5.20 NARBP 预测模型预测情况图

可见，第一次训练 27 次达到性能指标，预测合格率为 67.2222%;第二次训练 165 次达到性能指标，预测合格率为 72.2222%。分别训练和预测 5 次，发现每次结果不同，合格率从 60～75%不等。

4、分析

从以上几组仿真可以看出：

1） 本文构建的 NARBP 预测模型的预测均能够正确跟踪实际的观测值，可见，该模型在进行时间序列的预测是可行的。

2） NARBP 模型每次预测结果都不一样，可能是后传算法修正的权阈值最终陷入了初始权阈值附近的局部极小点。初始权阈值不同，落入的极小点也不同，因此每次的预测结果也就不同。

5.4.4 IGA—NARLMBP 模型

1、模型结构同 NARBP 模型

2、由 IGA 算法优选 NARBP 神经网络模型的初始优选权阈值，然后采用 LMBP 算法再次进行寻优。IGA 算法的参数选择如下：种群规模 L=100;权阈值初始化

空间为[-10, 10]；选择概率 $p_s = 0.05$；最大进化代数为 100。取 1937－1999 年(63 年）数据序列组成训练样本，将 IGA 算法优选出来的初始权阈值输入 NARBP 网络进行模型训练，然后取 2000 年－2004 年（5 年数据）序列作检验样本进行检验。仿真情况如下：

训练性能如图 5.21：

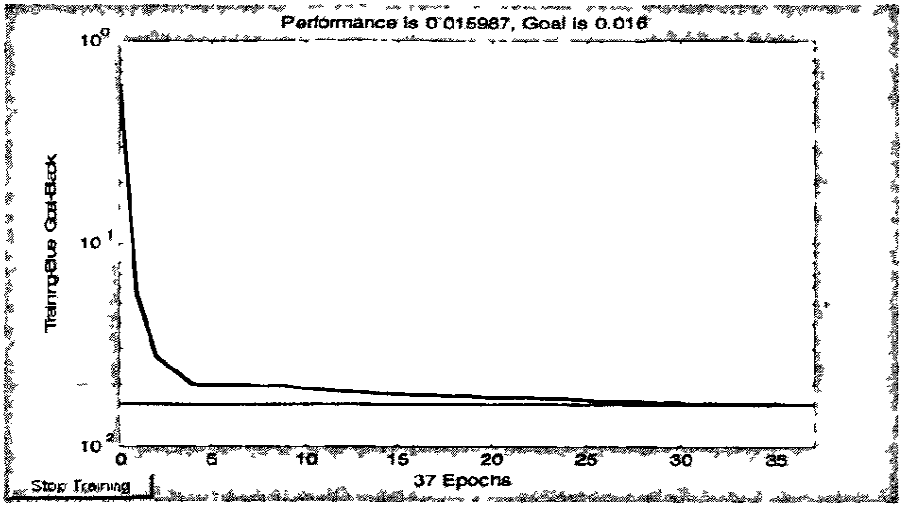

图 5.21 IGA－NARLMBP 预测模型训练性能图

预测情况图如图 5.22：

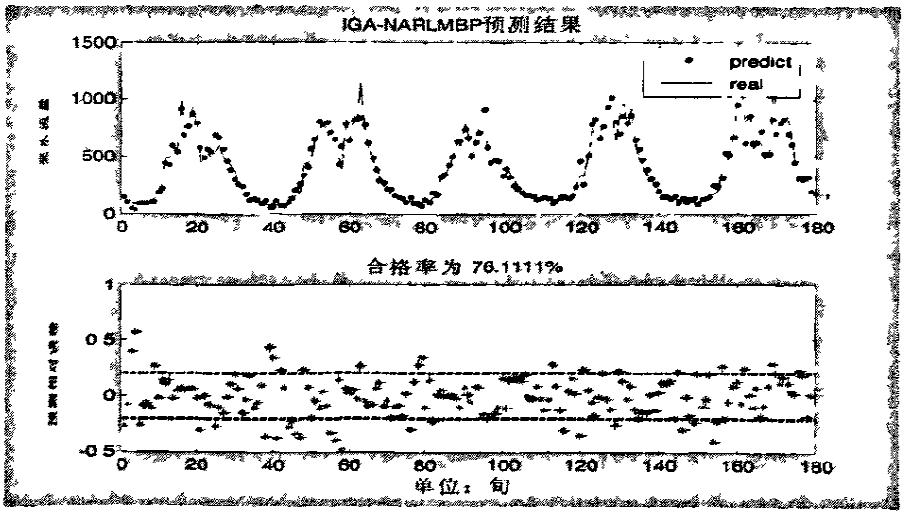

图 5.22 IGA－NARLMBP 预测模型预测情况图

分别进行5次训练，每次训练和预测情况相同，训练37次收敛，预测合格率达到76.1111%。可见，采用IGA算法进行了网络初始权阈值优选后，NARBP预测模型运行速度快，结果稳定，基本上落入全局最优，预测精度较高。

5.4 各模型仿真结果比较

以上三种预测模型的仿真结果总结比较如表5.2:

表5.2 各预测模型仿真结果比较表

预测模型	模型稳定性	收敛速度	预测精度
AR模型	稳定	快	一般
NARBP模型	不太稳定	速度不一	较好
IGA-NARLMBP模型	稳定	快	好

可见，NARBP预测模型比AR预测模型的预测精度高，但稳定性和模型收敛速度不如AR预测模型。IGA—NARLMBP预测模型的的模型稳定性、收敛速度、预测精度都好，综合比较，此改进算法后的模型用于预测时间序列比其它两个模型的性能要好，从而验证了此模型与算法用于时间序列预测的可行性、有效性和优越性。

5.5 本章小结

本章结合本人在研究生期间所作课题：都江堰灌区闸口岷江上游来水预测项目，介绍所构建的IGA-NARLMBP预测模型在河水来水预测中的应用。并分别与传统的AR预测模型和BP神经网络预测模型的预测结果进行比较，结果表明本文构建的基于IGA算法与LMBP算法相结合的NARBP预测模型的预测精度、预测平均合格率、预测模型的稳定性等指标比前两者好，从而验证了本文所建立的基于IGA算法与LMBP算法相结合的NARBP预测模型的有效性和优越性。

6 总结与展望

6.1 总结

目前，对于非线性时间序列的预测方法有两种：建立线性模型近似预测与构建合适的非线性模型进行预测。在非线性模型研究方面，近几年最为热门的是建立神经网络进行预测，其中，BP 神经网络模型是最广为应用的模型。但 BP 神经网络对时间序列预测的固有缺陷也成为共识：不具有时间序列的识别能力，标准 BP 算法收敛速度慢，算法易陷入局部最小点。这些均成为研究的热点与要点。本文也基于此进行了研究，在第三章采用自回归（AR）模型对时间序列的良好的辨识能力进行序列辨识，然后结合 BP 神经网络的良好的非线性映射能力，构建了 NARBP 神经网络时间序列预测模型；算法研究方面，对标准 BP 算法和现有的改进 BP 算法进行了研究，针对 BP 算法收敛速度慢的问题，选用现在较为成熟、特性较好的 LMBP 算法来进行神经网络模型的训练，针对算法易陷入局部极小点问题，根据遗传算法在求解优化问题、全局寻优的良好特性，提出将遗传算法与 BP 算法相结合的思路，通过对遗传算法的研究，知道了标准遗传算法的一些主要缺点—易出现“近亲繁殖”、早熟收敛、计算时间较长等缺陷，因此本文从控制参数、编码、适应度函数、选择算子、交叉算子、变异算子各方面对其作了一些改进，以提高其运算速度和全局寻优的性能，然后用改进的改进遗传算法与 LMBP 算法结合用于神经网络的全局寻优。最后将所构建的基于改进遗传算法和 LMBP 算法的非线性自回归神经网络应用于都江堰灌区渠首岷江上游来水预测的实验与仿真表明，本文所构建的基于 IGA 算法与 LMBP 算法相结合的 NARBP 预测模型的预测精度、预测平均合格率、预测模型的稳定性等指标比线性的 AR 模型和没经过算法改进的 NARBP 的效果都好，进而验证了本文所建立模型及算法的有效性和优越性。

6.2 展望

对于非线性时间序列的预测仍将是未来几年的研究热点，本文所构建的模型在这个课题方面有一定的通用性，但是这个模型本身还有许多需要研究的方面：对 BP 神经网络模型神经元节点的传递函数没有进行深入的研究；隐含层神经元的确定是由试算确定的，没有理论依据，缺乏科学性；在算法方面，BP 算法的改进仍在研究中，遗传算法也还在不断的发展与研究中，遗传算法与 BP 神经网络模型结合的方式也需要再研究。此外，引入一些新的技术与方法在本模型中以提高本模型的通用性与优越性也是进一步研究的需要。

1、在神经元节点的传递函数方面，可以寻求更好的函数（比如小波基函数）来替代 S 型函数，以进一步提高神经网络的非线性映射能力。

2、引入或设计更好的改进 BP 算法，使模型的收敛速度和训练精度提高，这需要进一步的研究与实验。

3、遗传算法与 BP 神经网络结合方面，可以尝试通过遗传算法优化神经网络结构，以弥补现在采用试算的方式来确定网络结构的不足。

4、在实际预测中，如果能够在模型中，加入外部影响因子项，那么实际预测效果应该会更好。

由于时间仓促和本人水平有限，本论文中尚存在许多不足与疏漏之处，希望得到各位专家、老师和同学的不吝教导和指正，本人不胜感激。

参考文献

[1]霍俊编著.实用预测学,第三册,实际序列预测分析.中国发明创造者基金会,中国预测研究会.1984,9

[2]George E. Box , Gwilym M. Jenkins, Gregory C. Reinsel，时间序列分析预测与控制，中国统计出版社. 1999.

[3]杜金观，项静怡，戴俭华编著．时间序列分析一建模与预测．安徽教育出版社．1997年5月

[4]王科俊，王克成著．神经网络建模、预测与控制．哈尔滨工业大学出版社．1996.12

[5]苑希民，李鸿雁，流树坤，崔广涛．神经网络和遗传算法在水科学领域的应用．中国水利水电出版社．2002，8

[6]夏安邦，王硕．定量预测引论．东南大学出版社，2001：1-229

[6]王小平，曹立明著．遗传算法理论、应用及软件实现．西安交通大学出版社.2002，1

[7]焦李成著．神经网络系统理论．西安电子科技大学出版社．1996:1-58

[8]Mclulloch W, Pitts W. A Logical Calculus of the Ideas Immanent in Nervous Activity. Bulletin of Mathematical Biophysics. 1943, 7

[9]B.Widrow and M.Hoff,1960 IRE WESCON Conr.Record: 96-104,Aug.1960

[10]M.Minsky and S.Papert,Perciptrons,MIT Press. 1969

[11]D.E.Rumelhart and McClelland.Parallel Distributed Processing: Exploration in the Microstructure of cognition.Cambridge, MANIT Press. 1986

[12]D. E. Goldberg. Genetic algorithms in search, optimization, and machine learning. Addison-Wesley, Publishing Company,INC. 1989: 1-180

[13]J. H. Holland. Adaptation in natural and artificial systems. University of Michigan Press, Michigan. 1975

[14]Kohonen T. An Introduction to Neural Computing. Neural Networks. 1988.Vol. 1(1):3 一 16P

[15] Lippmannn R P.An Introduction to Computing with Neural Nets[,T].IEEE ASSP

MIGAzines. 1987.4(2):4-22P

[16]Holland J,H.基因算法.科学（中译本）.1992,（11）：22－27

[17]Holland J.H.Adaptation in Natural and Artificial Systems.2nd ed.Cambridge,MA:NIT Press.1992：1-186

[18]高隽编著.人工神经网络原理及仿真实例.2003年08月

[19] Kolmogorov A.N. On the Representation of Continuous Functions of Several Variables by Superposition of Continuous Functions of One Variable and Addition. DokladyAkademiia Nauk SSSR. 1957,114(5)：953－956P

[20]靳蕃.神经计算智能基础.成都:西南交通大学出版社.2000.1

[21]张成乾，张国立编著.系统辨识与参数估计.北京：机械工也出版社.1986.11:86-111

[22]候媛彬，汪梅，王立琦编著.系统辨识及其MATLAB仿真.北京：科学出版社.2004.2：53－143

[23]王祯学，古中璧编著.系统辨识与自适应控制.四川科学出饭社.1998.3:1－84

[24]张贤达著.现代信号处理.北京：清华大学出版社.1995:87-92

[25]Kano M,Kawato M et al.Learning trajectory control of a redundant arm by feedback-error-learning.IEICE Technical Report.1989(463):1-6

[26]Newton R T & Xu Y S.Neural network control of a space manipulator.IEEE Control Systerms.Dec 1993,14-22

[27]Gomi H & Kawato M.Neural network control for a closed-loop sustem using feedback-error-learning .Neural Networks.1993,6:933-946

[28]Psaltis D et al.A multilayered neural network controller.IEEE Control Systems MIGAzine.Aprel,1988,17-24

[29]Saerens K J & Soquet A.A neural controller.Proceedings of 1st IEE conf. On NN.1989,211-215

[30]Nguyen D & Widrow B. The track backer-upper an example of self-learning in neural networks.Proceedings of international joint conference on Neural Networks,Washington DC.1989,2:11357-11363

[31]安鸿志，陈敏.非线性时间序列分析[M],上海科学技术出版社.1998 年 8 月.第一版 60-72

[32]王洪元，史国栋主编.人工神经网络技术及其应用.北京：中国石化出版社.2002年12

月

[33]魏海坤编著.神经网络结构设计的理论与方法.北京：国防工业出版社.2005 年 2 月

[34]闻新，周露，李翔，张宝伟编著.MATlAB 神经网络仿真与应用.北京：科学出版社.2003 年 7 月

[35]郑丽敏主编.人工智能与专家系统原理及其应用.中国农业出版社.2004 年 10 月.290-328

[36]玄光男[日]，程润伟，汪定伟，唐加福，黄敏译．遗传算法与工程设计．科学出版社，2000

[37]李敏强，寇纪淞，林丹，李书全．遗传算法的基本理论与应用．科学出版社，2002

[38]金菊良，丁晶著.遗传算法及其在水科学中的应用.四川大学出版社.2000 年 8 月

[39]雷英杰，张善文，李续武，周创明编著．MATLAB 遗传算法工具箱及应用．西安电子科技大学出版社．2005 年 4 月

[40]中华人民共和国水利部 SL250——2000 水文情报预测规范．北京：中国水利水电出版社，2000

[41]袁振洲．应用自回归积分移动平均法预测铁路货源货流发展趋势．铁道学报．1996 年 4 月

[42] 徐峰，王志芳，王宝圣．AR 模型应用于振动信号趋势预测的研究[J]．清华大学学报(自然科学版)．1999 年 04 期

[43]杨红卫，何晨．基于人工神经网络的毫米波雨衰减预测[J]．上海交通大学学报．2001，35(1)：13-16

[44]吴微，陈维强.用 BP 神经网络预测股票市场涨跌[J]．大连理工大学学报．2001,41(1)：9-15

[45]施式亮，刘宝深．基于神经网络的煤矿安全性预测模型及应用[J]．中国安全科学学报，1999，9(3)：23-26

[46] 张株，蔡开元.基于遗传算法的神经网络两阶段学习方案仁.TJ.系统仿真学报．2003，15；8)： 1088-1090 页

作者攻读硕士学位期间科研成果简介

1. 硕士研究生期间发表的文章

罗凤曼，周新志，占钟璧. 水资源规划智能决策支持系统的结构研究. 人民黄河，2005.8: 30-31

2. 硕士研究生期间参与的课题

(1) 水资源规划智能决策支持系统

(2) 都江堰灌区渠首岷江上游来水预测

(3) 都江堰灌区渠首配水系统

声 明

本人声明所呈交的学位论文是本人在导师指导下进行的研究工作及取得的研究成果。据我所知，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，也不包含为获得四川大学或其他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谢意。

本学位论文成果是本人在四川大学读书期间在导师指导下取得的，论文成果归四川大学所有，特此声明。

学位论文作者签名：罗凤霞　　签字日期：2006年 5 月 25 日

导　师　签　名：　　签字日期：2006年 5 月 25 日

学位论文版权使用授权书

本学位论文作者完全了解四川大学有关保留、使用学位论文的规定，有权保留并向国家有关部门或机构送交论文的复印件和磁盘，允许论文被查阅和借阅。本人授权四川大学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、汇编学位论文。

（保密的学位论文在解密后适用本授权书）

学位论文作者签名：罗凤雯　　导师签字：

签字日期：2006年 5月 25日　　签字日期：2006年 5月 25日

学位论文作者毕业后去向：

工作单位：成都电子机械高级专科学校　电话：13880649774

通讯地址：　　邮编：

致　谢

时光飞逝，三年的硕士研究生学习生涯即将结束。回首这三年所承载的学习、科研、生活，其中的努力、探索、乐趣等等，不免感慨。

这段求知与科研的历程，以及学位论文完成的过程中，离不开两位导师古钟璧教授和周新志副教授的谆谆教诲和悉心指导。古老师的厚德教学，所推崇和践行的海纳百川、有容乃大的气概与胸襟，以及教书解惑外，注重学生的内外兼修特别是人文素养的培养，这些对我均受益匪浅。周老师的严谨治学，对待学术和科研的一丝不苟与不断探索，个人丰富的专业知识，精湛的学术造诣，以及正直坦诚、为人师表的优秀品质，也将让我终生受益。非常感谢两位老师在我论文的选题、定位、成稿过程中给予的精心指导。值此论文完成之际，衷心向两位导师表示崇高的敬意与深深地感谢！

在作课题期间，我们水利模型组的组长贺新博士、赵成萍老师在课题中所提供的资料、所开展的与水利学院的老师和都江堰管理局的领导、高级工程师以及组间的各种研讨会，为课题的完成提供了好的研究条件与氛围，非常感谢他们，特别感谢同组的丁宁师弟在自回归预测部分所作的工作。

在智能控制研究所作课题和论文期间，得到了所里各位老师和同学、师弟师妹们提供的各种帮助与方便之处，在此一并呈上感激之情。

在四川大学电子信息学院2003级智能控制的这个集体中，杨真、陈瑜、卫朝霞、李敬、费克玲、孙天健、孙锐、陈鹏、胡敬鹏、艾龙，各位同级同学各具特色，塌实、认真、搞笑、幽默…，在这个奋进、团结、融洽的集体里，我的研究生生涯过得很多彩，我的论文也在互相加油与“比较”中愉悦地完成，一直想对你们说：很幸庆遇到你们，很幸庆和你们是同窗！

在这段大学毕业几年后再深造、完成学业的日子里，时时有父母、公公婆婆、丈夫和女儿给予的物资、精神和生活上的各种支持相伴，在此谨表达我对他们深深的谢意与无限的爱！

衷心感谢所有关心、支持、帮助过我的所有的人们！