苏州大学

高校教师申请硕士学位论文

（2006 届）

多层感知器学习算法研究

Research on Multilayer Perceptron Learning Algorithm

研究生姓名 王之仓

指导教师姓名 邓伟（副教授）

专 业 名 称 计算机应用技术

研 究 方 向 神经网络

论文提交日期 2006 年 10 月

中国知网 https://www.cnki.net

苏州大学学位论文独创性声明及使用授权声明

学位论文独创性声明

本人郑重声明：所提交的学位论文是本人在导师的指导下，独立进行研究工作所取得的成果。除文中已经注明引用的内容外，本论文不含其他个人或集体已经发表或撰写过的研究成果，也不含为获得苏州大学或其它教育机构的学位证书而使用过的材料。对本文的研究作出重要贡献的个人和集体，均已在文中以明确方式标明。本人承担本声明的法律责任。

研究生签名：王之仓　日　期：2006.11.25

学位论文使用授权声明

苏州大学、中国科学技术信息研究所、国家图书馆、清华大学论文合作部、中国社科院文献信息情报中心有权保留本人所送交学位论文的复印件和电子文档，可以采用影印、缩印或其他复制手段保存论文。本人电子文档的内容和纸质论文的内容相一致。除在保密期内的保密论文外，允许论文被查阅和借阅，可以公布（包括刊登）论文的全部或部分内容。论文的公布（包括刊登）授权苏州大学学位办办理。

研究生签名：王之仓　日　期：2006.11.25

导师签名：[illegible]　日　期：2006.11.25

中国知网 https://www.cnki.net

多层感知器学习算法研究

中文摘要

多层感知器是一种单向传播的多层前馈网络模型，由于具有高度的非线性映射能力，是目前神经网络研究与应用中最基本的网络模型之一，广泛应用于模式识别、图像处理、函数逼近、优化计算、最优预测和自适应控制等领域。而多层感知器采用的是 BP 算法。BP 算法的收敛速度慢是个固有的缺点，因为它是建立在基于只具有局部搜索能力的梯度法之上的，是只具有局部搜索能力的方法，若用于多个极小点的目标函数时，是无法避免陷入局部极小和速度慢的缺点的。因此，对 BP 算法的研究一直以来都是非常重要的课题。

毕业设计课题旨在对多层感知器的学习算法进行研究，并提出一种新的学习算法。由于 BPWE（权值外推 BP）算法和 TBP（三项 BP）算法都是基于权值调整的改进算法，而考虑将 TBP 算法中的均衡因子融入到 BPWE 算法中，从而使后者对权值的调整由原来的两项增加为三项，从而提出一种新的学习算法---TWEBP 算法。为了验证本算法的优点，采用了三个例子，分别对异或问题、三分类问题和函数逼近问题进行了实验，发现其收敛速度和逃离局部极小点的能力都优于传统算法。

关键词：多层感知器　学习算法　趋势外推　均衡因子　TWEBP

作　　者：王之仓

指导教师：邓　伟

中国知网 https://www.cnki.net

Research on Multilayer Perceptron Learning Algorithm

ABSTRACT

Multilayer Perceptron is a sort of multilayer feed-forward single direct propagation network model. Because of its good nonlinear mapping ability, it is one of the basic models in the research and application of neural network at present, which has been widely applied to pattern recognition, image processing, function approximation, optimization computation, optional prediction, adaptation control and so on. Multilayer Perception trained with BP algorithm often has a low convergence speed as a natural drawback, because it is based on gradient descent method which is only local searching. When applied to an object function with many local minimums, it is not possible for BP algorithm to avoid being trapped in local minimum and to have a low converges speed. In a word, the research on BP algorithm has become very important for a long time.

The purpose of this design task is to study the algorithms of Multilayer Perceptron, and a new BP algorithm is presented. Both BPWE algorithm (back-propagation by weight extrapolation) and TBP algorithm (a three-term back propagation algorithm) are based on weight value adjusted. Considered to add the proportional factor of the TBP algorithm into BPWE algorithm, it made the latter can adjust weight value by three terms too. A new BP algorithm, named TWEBP (the three-term weight extrapolation back propagation algorithm), is presented based on the two algorithm proposed just now. This new TWEBP algorithm is tested on three examples and the convergence behavior of the TWEBP and BP algorithm are compared. The results show that the proposed algorithm generally out-performs the conventional algorithm in terms of convergence speed and the ability to escape from local minima.

Keywords: Multilayer Perceptron, learning algorithm, extrapolation, proportional factor, TWEBP

Written by Wang zhi-cang
Supervised by Deng Wei

中国知网 https://www.cnki.net

目　录

中国知网 https://www.cnki.net

中国知网 https://www.cnki.net

第一章 绪 论

一个神经元有两种状态，即兴奋和抑制，平时处于抑制状态的神经元，其树突和胞体接收其他神经元经由突触传来的兴奋电位，多个输入在神经元中以代数和的方式叠加；如果输入兴奋总量超过某个阈值，神经元就会被激发进入兴奋状态，发出输出脉冲，并由轴突的突触传递给其他神经元。神经元被触发之后有一个不应期，在此期间内不能被触发，然后阈值逐渐下降，恢复兴奋性。神经元是按照“全或无”的原则工作的，只有兴奋和抑制两种状态，但也不能认为神经元只能表达或传递二值逻辑信号。因为神经元兴奋时往往不是只发出一个脉冲，而是发出一串脉冲，如果把这一串脉冲看成是一个调频信号，脉冲的密度是可以表达连续量的。

人工神经网络(ARTIFICIAL NEURAL NETWORK，简称 ANN)是在对人脑组织结构和运行机制的认识理解基础之上模拟其结构和智能行为的一种工程系统。早在本世纪 40 年代初期，心理学家 McCulloch、数学家 Pitts 就提出了人工神经网络的第一个数学模型，从此开创了神经科学理论的研究时代。其后，F.Rosenblatt、Widrow 和 Hopfield 等学者又先后提出了感知模型，使得人工神经网络技术得以蓬勃发展。

1.1 基本概念

从数学的角度讲，人工神经网络是一个由互连接突触的节点和激活连接构成的有向图，具有 4 个主要特征：

1 每个神经元可表示为一组线性的突触连接，一个应用它的外部偏置，以及可能的非线性激活连接。偏置由和一个固定为+1 的输入连接的突触连接表示。

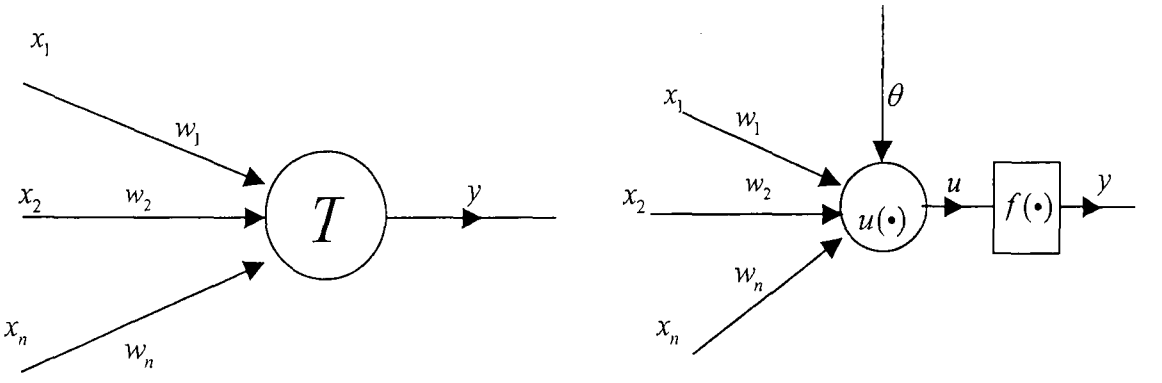
2 神经元的突触连接给它们相应的输入信号加权。

3 输入信号的加权和构成该神经元的诱导局部域。

4 激活连接压制神经元的诱导局部域产生输出。[1]

图 1-1（a）和（b）分别是神经元 MP 模型和通用神经元模型。

中国知网 https://www.cnki.net



(a) 神经元 MP 模型 （b） 通用神经元模型

图 1-1 神经元模型

1.2 神经网络的发展过程

1.2.1 产生背景

19 世纪 90 年代，美国心理学家 William James 发表了《心理学原理》一书，论述了相关学习、联想记忆的基本原理，对人脑功能作了创见性的工作。他指出当前基本的脑细胞曾经一起相继被激活过，其中一个受到刺激重新激活时，会将刺激传播到另一个。同时，James 认为，在大脑皮层上任意点的刺激量，是其他所有发射点进入该点的总和。1913 年，人工神经系统第一个实践是 Russell 描述的水力装置。1943 年，美国心理学家 W.S.Mcculloh 与数学家 W.H.Pirts 合作，用逻辑数学工具研究客观事件在形成神经网络中的数学模型表达，从此开创了对神经网络的理论研究。他们首先提出了二值神经元的 MP 模型（图 1-1（a）），论述了有一定数量输入作用下超过某一阈值，神经元才兴奋，突触联系的神经元之间只有兴奋和抑制两种方式。

1.2.2 发展历史

1 萌芽期（20 世纪 40 年代）

人工神经网络的研究最早可以追溯到人类开始研究自己的智能的时期，到 1949 年止。

中国知网 https://www.cnki.net

1943 年，心理学家 McCulloch 和数学家 Pitts 建立起了著名的阈值加权和模型，简称为 M-P 模型。发表于数学生物物理学会刊《Bulletin of Mathematical Biophysics》。

1949 年，心理学家 D.O.Hebb 提出神经元之间突触联系是可变的假说—Hebb 学习律。

2 第一高潮期（1950~1968 年）

以 Marvin Minsky，Frank Rosenblatt，Bernard Widrow 等为代表人物，代表作是单层感知器（Perceptron），该感知器可用电子线路模拟。

人们乐观地认为几乎已经找到了智能的关键。许多部门都开始大批地投入此项研究，希望尽快占领制高点。

3 潜伏期（1969~1982 年）

在 20 世纪 60 年代感知器的经典时期，好像神经网络可以做任何事情。但是 M.L.Minsky 和 S.Papert 于 1969 年出版《Perceptron》一书，利用数学证明单层感知器指出单层感知器所能计算的根本局限（如不能解决“异或”问题）。在有关多层感知器的简短一节中，他们认为没有任何理由假定单层感知器的任何局限可以在多层的情况下被克服。

另一方面，当时串行计算机正处于全盛发展时期，早期的人工智能研究也取得了很大成就，从而掩盖了发展新的计算型的迫切性，使有关神经网络的研究热潮低落下来。在此期间仍有不少科学家坚持这一领域的研究，对此后的神经网络研究提供了很好的理论基础。

4 第二高潮期

1982 年，J. Hopfield 提出了神经网络的一种数学模型，引入了能量函数的概念（用 Lyapunov 函数作为网络性能判定的能量函数，建立 ANN 稳定性的判别依据），研究了网络的动力学性质；紧接着（1984 年）又设计出用电子线路实现这一网络的方案，同时开拓了神经网络用于联想记忆和优化计算的新途径，大大促进了神经网络的研究。

1986 年，并行分布处理小组的 Rumelhart 等研究者重新独立地提出多层感知器的学习算法——反向传播算法，克服了当初阻碍感知器模型继续发展的重要障碍。

我国脑功能和神经网络课题的研究，早在 40 年前就已经进行，对于人工神经网络能力的研究，是在 20 世纪 80 年代才开始。1980 年，涂序彦先生发表《生物控制

中国知网 https://www.cnki.net

论》一书，书中系统地介绍了神经元和神经网络的结构、功能和模型。1988 年，北京大学组织召开了第一次关于神经网络的讨论会，一些知名学者在会上作了专题报告。1989 年，北京和广州等地召开了神经网络及其应用讨论会和第一届全国信号处理——神经网络学术会议。1990 年 2 月，由中国电子学会及计算机学会等八个学会联合发起并组织了中国第一次神经网络会议，参加人数 400 余人，搜集到会议记录中的论文 358 篇，内容涉及生物、人工神经网络模型、理论、分析应用及实现等各方面。1991 年由 13 个单位发起和组织召开了第二次全国神经网络会议，录用论文 280 篇。1991 年成立中国神经网络学会，大大推动了中国学术界及工程界在人工神经网络理论及应用方面的研究。经过十年的发展，我国人工神经网络的研究和应用正迈向新的高科技时代。

1.2.3 现状

20 世纪 80 年代以来，传统的基于符号处理的人工智能在解决工程问题时遇到了许多困难。现代的串行机尽管有很好的性能，但在解决像模式识别、学习等对人来说是轻而易举的问题上显得困难。这就促使人们怀疑当前的 Von Neumann 机是否能解决智能问题，也促使人们探索更接近人脑的计算模型，于是又形成了对神经网络研究的热潮。

近十年来，神经网络理论与实践有了引人注目的进展，它再一次拓展了计算概念的内涵，使神经计算、进化计算成为新的学科，神经网络的软件模拟得到了广泛的应用。近几年来科技发达国家的主要公司对神经网络芯片、生物芯片独有情钟。例如 Intel 公司、IBM 公司、AT&T 公司和 HNC 公司等已取得了多项专利，已有产品进入市场，被国防、企业和科研部门选用，公众手中也拥有神经网络实用化的工具，其商业化令人鼓舞。尽管神经计算机、光学神经计算机和生物计算机等研制工作的艰巨性和长期性，但有一点可以使人欣慰：它现在还只是初露锋芒，有巨大的潜力与机会，前景是美好的。

Mexico University 的 Forest 领导的小组开发出来的计算机免疫系统更是展现出惊人的潜力。此套免疫系统用于网络防毒，改变了传统的被动杀毒的方法，采用类似于人体免疫系统的主动抗毒，现在美国已经全面介入这套系统的开发。

中国知网 https://www.cnki.net

近年来，我国“863”计划、攻关计划、“攀登”计划和国家自然科学基金等，都对神经网络的研究给予了资助，吸引了大量的优秀青年人才从事神经网络领域的研究工作，促进我国在这个领域取得世界上的领先地位。

总之，在 21 世纪科学技术发展征程中，神经网络理论的发展将与日俱增。

1.3 多层感知器

1.3.1 基本概念

由 Rumelhart 提出的多层前馈神经网络，由于采用误差反传的 BP 学习算法，又被称为误差反向传播神经网络，简称 EBP 网络(Error Back Propagation)。

多层前馈神经网络其神经节点分层排列，一般由输入层，输出层和若干隐层组成。同层神经元节点之间没有连接，相邻两层之间的节点两两连接，而前一层神经元的输出即为后一层神经元的输入，每层神经节点只接收前一层神经元节点的输出信号。[3] 如图 1-2 所示。

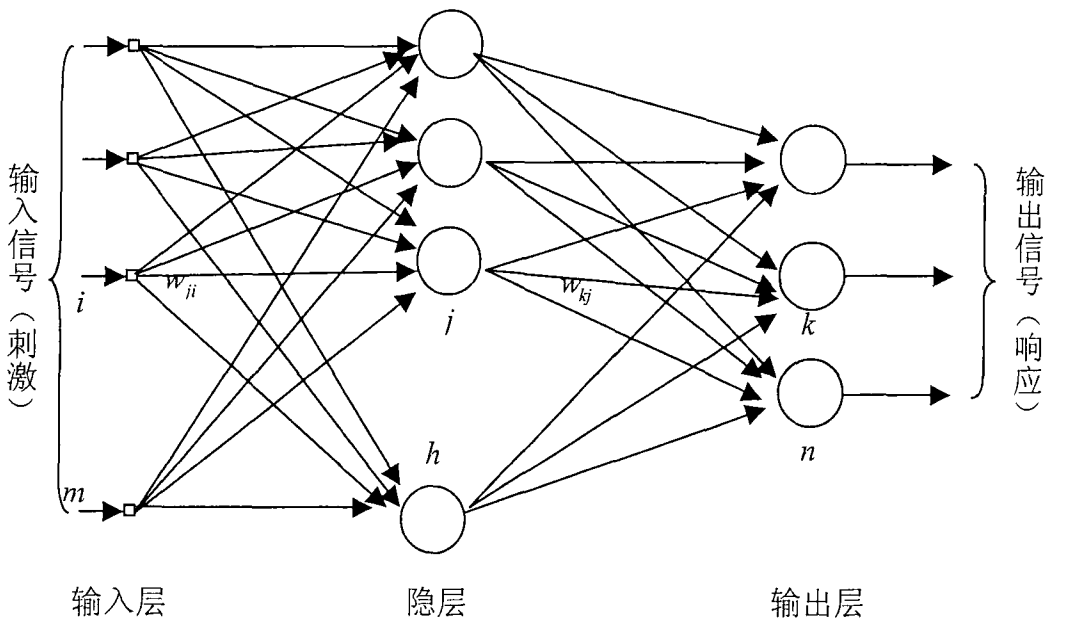


图 1-2 多层前馈网络的示意图

利用人工神经元的非线性特性，可以实现各种逻辑门。由于任何逻辑函数都可以由与非门组成，所以，第一，任何逻辑函数都可以用前馈网络实现。第二，单个阈值神经元可以实现任意多输入的与、或及与非、或非门。由于任何逻辑函数都可以化为

中国知网 https://www.cnki.net

析取（或合取）形式，所以任何逻辑函数都可用一个三层（只有两层计算单元）的前馈网络实现。[2][3]

1.3.2 多层感知器学习算法存在的问题

反向传播算法（Back-Propagation Algorithm，简称 BP 算法）已经成为多层感知器训练的标准算法。它通常作为其他学习算法的基准。但是，它本身存在大量的问题，突出表现在：

1 BP 算法的学习速度很慢，其原因主要有：

1） 由于 BP 算法本质上为梯度下降法，而它所要优化的目标函数又非常复杂，因此，必然会出现“锯齿形现象”，这使得 BP 算法低效。

2）存在麻痹现象，由于优化的目标函数很复杂，它必然会在神经元输出接近 0 或 1 的情况下，出现一些平坦区，在这些区域内，权值误差改变很小，使训练过程几乎停顿。

3）为了使网络执行 BP 算法，不能用传统的一维搜索法求每次迭代的步长，而必须把步长的更新规则预先赋予网络，这种方法将引起算法低效。

2 网络训练失败的可能性较大，其原因有：

1）从数学角度看，BP 算法为一种局部搜索的优化方法，但它要解决的问题为求解复杂非线性函数的全局极值，因此，算法很有可能陷入局部极值，使训练失败；

2） 网络的逼近、推广能力同学习样本的典型性密切相关，而从问题中选取典型样本实例组成训练集是一个很困难的问题。

3）难以解决应用问题的实例规模和网络规模间的矛盾。这涉及到网络容量的可能性与可行性的关系问题，即学习复杂性问题。

4）网络结构的选择尚无一种统一而完整的理论指导，一般只能由经验选定。为此，有人称神经网络的结构选择为一种艺术。而网络的结构直接影响网络的逼近能力及推广性质。因此，应用中如何选择合适的网络结构是一个重要的问题。

5）新加入的样本要影响已学习成功的网络，而且刻画每个输入样本的特征的数目也必须相同。

6）网络的泛化能力与训练能力的矛盾。一般情况下，训练能力差时，泛化能力

中国知网 https://www.cnki.net

也差，并且一定程度上，随训练能力地提高，泛化能力也提高。但这种趋势有一个极限，当达到此极限时，随训练能力的提高，泛化能力反而下降，即出现所谓“过拟合”现象。此时，网络学习了过多的样本细节，而不能反映样本内含的规律。

1.3.3 多层感知器学习算法的研究成果

由于多层感知器突出的优点和广泛的应用领域，对多层感知器学习算法的研究一直以来从未停止过，因而也取得了大量的研究成果。[4]

1 增加“惯性量（动量）”

$$W(n+1)=W(n)-\alpha\Delta W(n)+\eta\Delta W(n-1) \tag{1.1}$$

2 采用动态步长

1） 开始时，α 取大一些，然后逐渐减小 α 值。

2） 动态选取的步长，比如先给一个的初始值，在迭代的过程中按增减的情况不断调整值，如

$$\Delta\alpha=\begin{cases}k_1，\text{连续}n_1\text{次}\Delta E<0\\ k_2，\text{连续}n_2\text{次}\Delta E>0\\ 0，\text{其他}\end{cases} \tag{1.2}$$

还可以采取其他的调节方法。

3 与全局搜索算法相结合

为克服 BP 算法全局搜索能力弱的缺点，将 BP 算法与具有很强全局搜索能力的算法相结合，如与遗传算法结合。

4 模拟退火算法（simulative anneal）

为了克服 BP 算法易陷入局部极小的缺点，人们从退火现象中得到启发，引入模拟退火算法。

为了不陷入局部最小，在用梯度法迭代的过程中，可以不完全按梯度下降的方向进行迭代，而是给予一个小概率的机会，按不同的方向进行迭代，这样就有可能跳出局部极小的陷阱。

将熵空间、统计推断方法和启发式搜索技术相结合，可以有效地降低计算量。

中国知网 https://www.cnki.net

1.4 毕业设计工作及论文结构

1.4.1 毕业设计工作

作者通过广泛地阅读关于BP算法改进的论文，比较提出的各种改进算法，认为Kamarthi和Pittner[5]根据趋势外推的思想提出的基于每个独立的互连权值外推的加速标准BP算法的算法——权值外推算法（BPWE）和Zweiri、Whidborne和Seneviratne[6][7]提出的通过加大收敛率来减少训练时间，并降低学习延迟，同时保持了标准二项BP算法的简单有效的进行权值调整的方法—三项BP算法（TBP）是比较优秀的算法。因为BPWE算法和TBP算法都是基于权值调整的改进算法，而考虑将TBP算法中的三项因子融入到BPWE算法中，从而使后者对权值的调整由原来的两项增加为三项，加快收敛速度，避免陷入局部极小点，从而提出一种新的学习算法---TWEBP算法，从而完成题为“多层感知器学习算法研究”的毕业设计。

1.4.2 论文结构

全文共分为七章，具体介绍如下：

第一章介绍了神经网络的产生背景、发展历史，多层感知器的学习算法存在的问题、研究成果、改进的策略和最终的论文结构。

第二章描述了反向传播算法的原理和对其进行改进的策略。

第三章描述了性能优化的理论基础，比较并分析了几种成熟的基于BP算法的优化算法。

第四章阐述了趋势外推的思想和基于该思想的外推算法，并阐述了三项因子的思想，并在此基础上提出了改进算法---TWEBP算法，并以XOR问题、三分类问题和函数逼近为例进行了仿真试验。

第五章对全文进行总结，提出了对BP算法进行改进的后继工作。

中国知网 https://www.cnki.net

第二章　反向传播算法

最初由 Werbos 开发的反向传播训练算法是一种迭代梯度算法，用于求解前馈网络的实际输出与期望输出间的最小均方差值。BP 网是一种反向传递并能修正误差的多层映射网络。它采用梯度搜索算法，以期使网络实际输出与期望输出的误差均方值为最小。当参数适当时，此网络能够收敛到较小的均方差，是目前应用最广的网络之一。

2.1　反向传播算法

2.1.1　学习规则

误差反向传播算法以一种有教师示教的方式进行学习，学习过程由正向传播过程和反向传播过程组成。首先由教师对每一种输入模式设定一个期望输出值。然后对网络输入实际的学习记忆模式（或称训练样本），并由输入层经中间层向输出层传播(称为“正向传播过程”)。实际输出与期望输出的差即是误差。按照误差平方最小这一规则，如果在输出层不能得到期望的输出，则转入反向传播，根据实际输出与期望输出之间的误差，由输出层往中间层逐层修正连接权值，此过程称为“误差反向传播”。所以误差反向传播神经网络也简称 BP(Back Propagation)网。随着“正向传播”过程和“误差反向传播”过程的交替反复进行。网络的实际输出逐渐向各自所对应的期望输出逼近，网络对输入模式的响应的正确率也不断上升。通过此学习过程，确定下来各层间的连接权值之后就可以工作了。

2.1.2　学习过程

BP 算法是一个有导师的学习算法，它含有隐节点。对于一个输入样本，经过网络的正向推理得出一个输出，然后让它与期望的输出样本进行比较。如果有偏差，就从输出开始向回传播，调整权系数 W_{ji}。

中国知网　https://www.cnki.net

设 X 为输入样本，Y 为输出样本，T 为期望输出样本，η 为学习率(是一个小于 1 的正数)，$f(x)$ 是网络的激活函数，选用 S 形曲线，而是 W_{ji} 第 i 个单元到第 j 个单元联接的权系数，$f'(x)$ 为 $f(x)$ 的导数，正向传播时是从输入一层一层地到输出，上一层的输出作为下一层的输入。

于是有：

正向传播：

$$Y_i = f(\sum_{i=1}^{n} W_{ji} \cdot X_i) \tag{2.1}$$

其中 $f(x) = \dfrac{1}{1-\exp(-x)}$。

学习过程：

$$w_{ji}(n+1) = w_{ji}(n) + \eta \cdot \delta_j \cdot x_i \tag{2.2}$$

对于输出节点：

$$\delta_i = (T_i - Y_i) f'(\sum_{i=1}^{n} w_{ji} \cdot x_i) \tag{2.3}$$

对于非输出节点：

$$\delta'_j = f'(\sum_{i=1}^{n} w_{ji} \cdot x_i) \cdot \sum_{i=1}^{n} \delta_k w_{ji} \tag{2.4}$$

BP 算法收敛慢是由于误差是时间的复杂非线性函数，而 BP 算法本质上是简单的最速下降法，其权值调整依据误差对权值的偏导数。即按误差变化率最小的方向进行，当接近收敛时 $f(x)=0$，导致收敛缓慢。初始值是很小的随机数，而权增量:

$$\Delta w'_{ji} = \eta \cdot \delta_j \cdot x'_i \tag{2.5}$$

该式中各系数对权的修正程度不同，但 η 值保持不变，则会导致对某些系数的过修正，所以只有当 η 很小时才会收敛。

中国知网　https://www.cnki.net

2.1.3　反向传播算法的步骤

Step1: 对权系数 w_{ji} 置初值。对各层的权系数 w_{ji} 置一个较小的非零随机数;

Step2: 输入一个样本 $X=(x_1,x_2,...,x_n)$，以及对应期望输出 $Y=(y_1,y_2,...,y_n)$。

Step3: 对于神经元 j 的输出，有:

$$y_j=f(\sum_{i=1}^{n}w_{ji}\cdot x_i) \tag{2.6}$$

其中 $f(x)=\frac{1}{1-\exp(-x)}$。

Step4: 判断神经网络逼近误差是否满足要求或迭代训练次数达到容许值

$$e<\varepsilon \text{ 或 } iterator\geq iterator_{\max} \tag{2.7}$$

（其中，$e=0.5\cdot\sum_{i=1}^{n}(T_i-Y_i)^2$，$Y_i$是输出单元的期望值;它也在这里用作教师信号，Ti 是实际输出）

若满足条件则结束，否则继续。

Step5: 计算局部梯度

当 j 为隐藏层神经元时，局部梯度:

$$\delta_j(n)=-\varphi'_j(v_j(n))\sum_k\delta_k(n)w_{kj}(n) \tag{2.8}$$

当 j 为输出层神经元时，局部梯度:

$$\delta_j(n)=e_j(n)\varphi'_j(v_j(n)) \tag{2.9}$$

Step6: 修正权系数

w_{ji}（n）的修正 Δw_{ji} 为:

$$\Delta w_{ji}(n)=\eta\delta_j(n)y_i(n) \tag{2.10}$$

于是，新的经过修改的权值系数为:

$$w_{ji}(n+1)=w_{ji}(n)+\eta\cdot\delta_j\cdot y_i(n) \tag{2.11}$$

Step7:　转 Step3。

中国知网　https://www.cnki.net

2.2　反向传播算法的贡献和局限性

反向传播算法是多层感知器的标准算法，是其他学习算法的基准。它的诞生就是非常有意义的。它对神经网络的发展做出了卓越的贡献，但是也有其与生俱来的缺陷。

2.2.1　反向传播算法的贡献

1969 年 Minsky 曾指出前向神经网络的一些局限性，如无法用单层神经网络来实现完成“异或”的对应关系，他的观点大大影响了神经网络的发展。Rumelhart 等人针对多层感知前向神经网络提出 BP 算法，并用该算法求解 XOR 问题，为多层前向神经网络提供了切实可行的学习算法，使前向神经网络得到各种应用成为可能，这是 BP 算法的最大贡献。[4]

其次，由于梯度下降法是法国数学家 Chauchy 于 1847 年提出来的，已有 150 多年，因而 BP 算法理论依据坚实，同时，推导过程严谨，所得公式对称，物理概念清晰（误差的反向传播），通用性好。将学习问题化成一个求函数最小值的问题进行求解，然后利用梯度法进行搜索，这是大多数研究人员都能想到的方法。但是为什么对神经网络的学习问题，长期以来没有人想到这个方法，直至 1986 年 Rumelhart 等人才想到呢？这应该归功于他们提出的前向神经网络的概念，这个概念将网络大大简化，在这样简化的基础上，才有可能推导出普适的学习方法。其次，他们巧妙地引入了“误差”的概念，使求解梯度的公式变得十分对称、优美、简单，且具有鲜明的物理意义。上述这些优点使它至今仍然是前向神经网络的主要算法之一。

2.2.2　反向传播算法的局限性

BP 算法的出现对神经网络的发展所起的作用是具有历史意义的，但是，也存在很多的问题。但是 BP 算法由于采用搜索机制，这也造成了它的本质的缺点。它的缺点有：一是收敛速度慢，一般具有四五个元件的网络用 BP 算法求解，通常需要循环几千次，甚至上万次才能收敛，故难以处理海量数据。二是所得网络容错能力（tolerant capacity）差；三是算法不完备（易陷入局部极小）。[4]

中国知网　https://www.cnki.net

2.3 对反向传播算法的进一步讨论

由于 BP 算法采用了基于梯度搜索的机制，所以收敛速度慢是它的致命的无法避免的缺点。鉴于以该算法为基础的多层感知器网络的重要性，广大的研究者们从从激活函数、动量项、学习速率和误差函数等多个方面提出了改进的意见和算法。

2.3.1 激活函数

计算多层感知器每一个神经元的误差需要关于神经元的激活函数 $\varphi(\bullet)$ 的导数知识。要导数存在，则需要函数 $\varphi(\bullet)$ 连续。用基本术语，激活函数必须满足的要求是可微性。

1 基本激活函数：

通常用于多层感知器的连续可微非线性激活函数的一个例子是 Sigmoid 非线性性。有两种基本的形式。

1）Logistic 函数。这种 Sigmoid 非线性的一般形式由

$$\varphi_j(v_j(n)) = \frac{1}{1+\exp(-av_j(n))} \quad a>0, -\infty < v_j(n) < \infty \tag{2.12}$$

定义，这里 $v_j(n)$ 是神经元 j 的诱导局部域。根据这种非线性性，输出的范围位于 $0 \le y_j \le 1$ 之内。

（1）当神经元 j 位于输出层时，其局部梯度为

$$\delta_j(n) = e_j(n)\varphi'_j(v_j(n)) = a[d_j(n)-o_j(n)]o_j(n)[1-o_j(n)] \tag{2.13}$$

其中，$o_j(n)$ 是神经元 j 输出端的函数信号，而 $d_j(n)$ 是它的期望响应。

（2）当神经元 j 位于隐藏层时，其局域梯度为

$$\delta_j(n) = \varphi'(v_j(n))\sum_k \delta_k(n)w_{kj}(n) = ay_j(n)[1-y_j(n)]\sum_k \delta_k(n)w_{kj}(n) \tag{2.14}$$

导数 $\varphi'(v_j(n))$ 当 $y_j(n)=0.5$ 时取得最大值，当 $y_j(n)=0$ 或 $y_j(n)=1$ 时取它最小值（即 0）。既然网络的一个突触权值的变化总量与导数 $\varphi(v_j(n))$ 成比例，因此对于一个

中国知网 https://www.cnki.net

Sigmoid 激活函数来说，突触权值改变最多的神经元是那些函数信号在它们的中间范围之内的网络的神经元。正是反向传播学习这个特点导致它作为学习算法的稳定性。

2) 双曲正切函数

是另外一个经常使用的 Sigmoid 非线性形式的激活函数，它的最通用的形式由

$$\varphi_j(v_j(n)) = a\tanh(bv_j(n)) \quad (a,b) > 0 \tag{2.15}$$

定义，这里 a 和 b 是常数。

（1）当神经元 j 位于输出层时，其局域梯度为

$$\delta_j(n) = e_j(n)\varphi'_j(v_j(n)) = \frac{b}{a}[d_j(n) - o_j(n)][a - o_j(n)][a + o_j(n)] \tag{2.16}$$

（2）当神经元 j 位于隐藏层时，其局域梯度是

$$\delta_j(n) = \varphi'_j(v_j(n))\sum_k \delta_k(n)w_{kj}(n) = \frac{b}{a}[a - y_j(n)][a + y_j(n)]\sum_k \delta_k(n)w_{kj}(n) \tag{2.17}$$

2 对激活函数的改进

标准激活函数为 S 型函数 $f(x) = 1/1 + \exp(-x)$，其导数 $f'(x) = \exp(-x)/(1+\exp(-x))^2$，其优点为用该函数可使同一网络既能处理小信号，也能处理大信号。这是因为该函数的中间高增益区解决了处理小信号的问题，而伸向两边的低增益区正好适合于处理大的激励信号。这种情况正像生物神经元在输入电平范围很大的情况下能正常工作一样。分析一个 BP 算法后不难发现这个激活函数是一个饱和函数，并且该激活函数的导数被多次运用。当它趋于饱和状态时，导数就接近于零，也就是说，BP 算法学习和收敛的速度慢是由误差信号因子式 $x(1-x)$ 引起的，结合 δ 规则的算法推导并注意到 Sigmoid 函数 $f(x) = 1/(1+\exp(-x))$ 的导数和原函数的关系 $f'(x) = f(x)(1-f(x))$，不难看出，$x(1-x)$ 因子的出现实际上是神经元特性函数选取 Sigmoid 函数所致，从而造成收敛速度减慢。然而，Sigmoid 函数的导数范围仅为（0,0.25），这也使得学习速率较慢。通过激活函数的修改，来提高网络学习能力的研究多围绕这一基础而展开。

1）Fahlman 直接将 $f'(x)$ 加上一个常数 0.1 以避免其为 0。[8]

中国知网 https://www.cnki.net

2）Chen 和 Mars[9]提出将 $o_j(1-o_j)$ 直接从 $f'(x)$ 中移走 ,并在不同层的节点更新表达式中使用不同的学习率,在这种情况下,节点的误差项就是反馈的误差信号。

3）Vitela 和 Reifman[10]使用的方法是当节点输出落入饱和区域时将 $f'(x)$ 用一个常数来表示。与标准的 BP 算法相比，这些方法虽然在一定程度上避免了饱和区域问题，但在收敛速度上却没有显著提高。

4）NG Sin-chun, CHEUNG Chi-chung, LEUNG Shu-hung 等人提出一种放大梯度函数的方法(简称 MGFPROP 方法)[11]，将 $f'(x)$ 表达式中的 $o_j(1-o_j)$ 项修改为 $[o_j(1-o_j)]^{1/s}$ $s>1$。MGFPROP 方法虽然可以提高收敛速度,对 $f'(x)$ 起到放大作用,但这种方法并不能完全解决饱和区域问题，因为当节点的实际输出 o_j 等于 0 或 1 时,仍有 $o_j'=0$，同时，由于 s 值取值过大会引起学习的振荡，影响系统的稳定性，因此引入参数 s 以后，原有算法的学习率需要重新设置，即向下调整到一个较小的值上。

5）李军等[12]人将隐层的激活函数进行线性化后，采取使各层的代价函数对网络权值的梯度为 0 的方法对隐层到输出层和输入层到隐层的网络权值分别交替进行优化，同时，在隐层优化的修正代价函数中加入一个惩罚项，以减少其激活函数线性化带来的误差。

6）朱武亭[13]等对激活函数进行了修正，定义为 $f'(x)=f(x)+\theta\cdot x$，也是对饱和区进行修正以放大误差。

7）邓娟，杨家明[14]等将修改激活函数和构建假饱和预防函数相结合，实现加快网络学习速率。

8）武研，王守觉[15]等从神经网络的权值调节公式入手，通过避免过早饱和、加大权值调节的幅度等手段来加快收敛。

9）李东侠,李平,丁淑艳等通过修改激活函数来减小网络误差，并将该方法应用于广义预测控制。[16]

中国知网 https://www.cnki.net

2.3.2　学习模式

在反向传播算法的实际应用中，学习结果是从将指定训练样本多次呈现给多层感知器而得到的。在一个学习过程中整个训练集的完全呈现称之为一个回合（epoch），即完成一次训练。学习过程是在一个回合接一个回合的基础上进行直到网络的突触权值和误差水平稳定下来，并且整个训练集上的均方误差收敛于某个极小值。从一个回合到下一个回合时将训练样本的呈现顺序随机是一个很好的实践。这种随机化易于在学习循环中使得权空间搜索具有随机性，因此可以在突触权值向量演化中避免极限环出现的可能性。

对于一个给定的训练集，反向传播学习可能会以两种基本方式（串行方式和集中方式）的一种进行。

BP 算法对样本进行串行学习时，常会发生“学了新的，忘了旧的”（遗忘）的现象。为此，只好对样本不断循环重复的学习，这样一来其学习时间必然很长。为了克服这个缺点，将串行学习改为集中学习（对所有样本进行学习后将其误差相加，然后用这个误差之和来对网络的权系数进行调整），但是集中学习又带来新的问题，由于集中学习将各样本的误差加在一起，根据其“和”值对权系数进行调整，由于误差能够相互抵消，这就降低了算法的调整能力，也就延长了学习的时间。同时，集中学习的方法还可能产生新的局部极小点。这是因为如果各误差不为零，但总和为零，这种情况产生之后算法就稳定在这个状态上，这个状态就是由于使用集中学习算法而产生的新的局部极小点。换句话说，按集中学习其收敛速度也必然很慢。若改用误差的平方和，则可避免产生的新的局部极小点，但是仍然会出现误差避免落入局部极值的现象。若改用随机算法，而随机算法的复杂性基本上和逐个学习算法一样。[4]所以，对集中方式的学习算法的改进不失为一个好的研究方向。

2.3.3　动量项

在工作中，学习步长η的选择非常重要，η大收敛快，但过大的η可能引起不稳定（η最大不能超过$\frac{2}{\lambda \max}$）；η小可避免不稳定，但收敛速度慢。增加动量项正是解决这一矛盾的最简单的办法[11]。即

中国知网　https://www.cnki.net

$$\Delta w_{ji}(n)=\alpha\Delta w_{ji}(n-1)+\eta\delta_j(n)y_i(n)\quad 0<\alpha<1 \tag{2.18}$$

式中第二项是常规BP法的修正量，第一项称之为动量项，α为某一正数，其作用为：

当顺序加入训练样本时上式可以写成以t为变量的时间序列，t由0到n，因此上式可看作是$\Delta w_{ji}(n)$的一阶差分方程，对$\Delta w_{ji}(n)$求解得：

$$\Delta w_{ji}(n)=\eta\sum_{t=0}^{n}\alpha^{n-1}\delta_j(t)y_i(t)=-\eta\sum_{t=0}^{n}\alpha^{n-1}\frac{\partial\xi}{\partial w_{ji}(t)} \tag{2.19}$$

当本次的$\frac{\partial\xi}{\partial w_{ji}(t)}$与前一次同号时，其加权求和值增大，使$\Delta w_{ji}(n)$较大，结果在稳定调节时加速了$w$的调节速度。当$\frac{\partial\xi}{\partial w_{ji}(t)}$与前次符号相反时说明有一定的振荡，此时指数加权求和结果使$\Delta w_{ji}(n)$减小，起到稳定作用。

附加动量的引入，可以使网络权值的变化不仅反映局部的梯度信息，而且反映误差曲面最近的变化趋势。这一算法虽然在一定程度上解决了局部极小问题（对大多数实际问题应用问题，这一能力也极为有限），但其训练速度仍然很慢。该算法的另一个缺点是动量系数如何取值，有些学者令$\mu=1-\eta$。但是，在实际应用中还是有待于多次试验。[17]

2.3.4　学习速率

反向传播算法提供使用最速下降方法在权空间计算得到的轨迹的一种近似。使用的学习率参数η越小，从一次迭代到下一次的网络突触权值的变化量就越小，轨迹在权值空间就越光滑。然而这种改进是以减慢学习速度为代价的。另一方面，如果η的值太大以加快学习速度的话，结果就有可能使网络的突触权值的变化量不稳定（即振荡）。

学习速率可以是动态变化的，也可以是静态不变的。但是，如果学习速率不是动态变化的，那么，学习的过程往往是一个冗长而等待的过程。因此，对学习速率的研

中国知网　https://www.cnki.net

究，主要集中从如何动态地调整它的角度上进行。技巧是决定何时改变学习速度和怎样改变学习速率。

1 可变学习速度的 VLBP 算法[18]

这是一种非常直观的批处理过程，它的学习速率是根据算法的性能改变的。其规则如下：

1）如果均方误差(在整个训练集上)权值在更新后增加了，且超过了某个设置的百分数ζ(典型值为 1%至 5%)，则权值更新被取消，学习速度被乘以一个因子$\rho(0<\rho<1)$，并且动量系数（如果有的话）被设置为 0。

2）如果平方误差在权值更新后减少，则权值更新被接受，而且学习速度将乘以一个因子$\eta>1$。如果被动量系数设置为 0．则恢复到以前的值。

3）如果平方误差的增长小于ζ，则权值更新被接受，但学习速率保恃不变。如果动量系数过去被设置为 0，则恢复到以前的值。

2 Jacobs 提出了 delta-bar-delta 学习规则，其中每一个网络参数(权值和偏置值)都有自己的学习速度。如果某个参数在几次迭代中都沿同一方向变化，算法则增加网络参数的学习速度。如果参数的改变方向发生变化，则学习速度递减。Tollenaere 的是 SuperSAB 算法与 delta-bar-delta 规则类似，但它在改变学习速度的规则方面更加复杂。[17]

3 另一种对 SDBP 的启发式变型是 Fahlman 的 Quickprop 算法。它假设误差曲面是抛物面且在极小点附近是向上凹的，另外每个参数的影响被认为是相互独立的。

对 SDBP 进行启发式改进对某些问题会提高收敛速度。但这些方法有两个主要缺点。首先这些改进需要设置一些参数，而 SDBP 只需要一个学习速度参数。某些更复杂的启发式改进需要设置五六个参数。算法的性能对这些参数的改变往往十分敏感。参数的选择还是问题相关的。对 SDBP 的改进的第二个缺点是它们对某些 SDBP 最终能找到解的问题却不能收敛。应用越复杂的算法这些问题越容易发生。

杨东侯，年晓红，杨胜跃等借鉴网络拥塞控制中的“慢启动”策略，提出改进学习率的学习算法。[19]

李宏刚,吕辉,李刚[20]等提出了一种修正学习因子的方法来调整网络的性能，并应

中国知网 https://www.cnki.net

用于在导弹指令跟踪中。

学习速率可变的BP算法是针对定步长的缺陷提出来的。这种算法是一进化论中的进退法为理论基础的，即连续两次观测训练数据，如果误差下降则增大学习率，误差的反弹在一定的范围内，则保持步长。误差的反弹超过一定限度则减小学习率。然而，在 $f(x^{(k)})$ 很小的情况下，仍然存在权值修正量很小的问题，致使学习效率很低。[18]

2.3.5 误差函数

1 传统误差函数

训练网络的过程中，当在神经元 j 呈现第 n 个样本进行训练时，它的误差定义为 $e_j(n)=d_j(n)-o_j(n)$，其中，$d_j(n)$ 为期望响应值，$o_j(n)$ 为实际输出值。该误差的平方和为 $(1/2)e_j^2(n)$。则输出层的所有神经元的误差平方和为 $\varepsilon_{av}=\frac{1}{2}\sum_{j\in C}e_j^2(n)$。在以集中方式训练网络时，采用误差的平方和为代价函数 $\varepsilon_{av}=\frac{1}{2N}\sum_{n=1}^{N}\sum_{j\in C}e_j^2(n)$。

2 对误差函数的改进

标准激活函数为 S 型函数 $f(x)=1/1+\exp(-x)$，当它趋于饱和状态时，输出的变化范围很小，这种情况下，误差信号很小，因而对误差函数的改进多在通过提出新的误差函数从而使误差始终保持较大的数值。

1）Baum 提出将误差定义为 $E=\sum_k[\frac{1}{2}(1+t_k)\log\frac{1+t_k}{1+a_k}+\frac{1}{2}(1-t_k)\log\frac{1-t_k}{1-a_k}]$，这种误差函数不会产生不能完全训练的麻痹现象。

2）苏小红等[21]提出一种基于反馈调控参数的 BP 学习算法，以解决经遗传算法优化后的网络因初始权值过大而陷入饱和区域导致学习停滞的问题。该算法的主要思想是在原有算法的基础上增加了一个称为 excitation 的反馈调控函数,它模拟在前

中国知网 https://www.cnki.net

后突触间隙中释放化学性神经递质的调节系统的作用，对神经元的激活值起调控作用，将反馈的误差信号向误差梯度最大的方向进行调整。

3） Franzini 重新定义了训练误差的表示方法，使得当 $f'(x)$ 趋近于 0 时，误差信号始终保持较大的数值。

4） Van Ooyen 和 Nienhuis 则定义了一种对数形式的误差函数,使得其梯度表达式中不再含有 $o_j(1-o_j)$ 项。

5）赵亚丽，西广成，易建强等[22]根据信息理论用度量两个状态之间的差别程度的 Kullback-Leibler(即相对熵)距离得出网络的输出值和理想输出值的接近程度可用相对熵来度量,也就是将相对熵作为误差函数。

6）武研，王守觉[23]将神经网络输入调整和通常的权值调整的反向传播算法结合起来，通过调整权值和输入矢量值得双重作用来最小化神经网络的误差函数。

2.4 小结

BP 算法作为非常重要的学习算法存在，同时也因为它有一些缺陷，所以广大研究人员从各种角度去研究和改进它，并且取得了大量的成就。通过本章的描述和讨论，可以洞悉对 BP 算法进行改进的方向和策略。

中国知网 https://www.cnki.net

第三章 性能优化

性能学习是区别于联想学习（如 Hebb 学习）和竞争学习之外的一类非常重要的学习规则，其目的在于训练网络时为优化网络性能而调整网络参数(权值和偏置值)。

3.1 性能优化的理论基础

性能优化过程分两个步骤进行。

第一步是定义“性能”的含义。换言之，需要找到一个衡量网络性能的定量标准，即性能指数，性能指数在网络性能良好时很小，反之则很大。本文采用误差作为性能指数。

第二步是搜索减小性能指数的参数空间(调整网络权值和偏置值)。

以调整网络参数的方式来优化网络的方法，可以通过构造优化性能指数 $F(x)$ 的方法来实现。优化的目的是求出使 $F(x)$ 最小化的 x 的值。首先，给定一个初始猜测值 x_0，然后按照等式 $x_{k+1}=x_k+\alpha_k p_k$ 或 $\Delta x_k=(x_{k+1}-x_k)=\alpha_k p_k$ 逐步修正 $F(x)$。当进行最优点迭代时，函数应该在每次迭代时都减小，即 $F(x_{k+1})<F(x_k)$。通过限定泰勒级数展开项的数量，可以用泰勒级数近似估计性能指数。神经网络的性能指数是向量，它是所有网络参数（各个权值和偏置值）的函数，参数的数量可能是很大的。因此，需要将泰勒级数展开形式扩展为多变量形式。对于 n 元函数。

$$F(x)=F(x_1,x_2,...,x_n) \tag{3.1}$$

其在点 x*的泰勒级数展开为

$$F(x)=F(x^*)+\frac{\partial}{\partial x_1}F(x)|_{x=x^*}(x_1-x_1^*)+\frac{\partial}{\partial x_2}F(x)|_{x=x^*}(x_2-x_2^*)$$

$$+...+\frac{\partial}{\partial x_n}F(x)|_{x=x^*}(x_n-x_n^*)+\frac{1}{2}\frac{\partial^2}{\partial x_1^{\,2}}F(x)|_{x=x^*}(x_1-x_1^*)^2$$

$$+\frac{1}{2}\frac{\partial^2}{\partial x_1\partial x_2}F(x)|_{x=x^*}(x_1-x_1^*)(x_2-x_2^*)+... \tag{3.2}$$

中国知网 https://www.cnki.net

用矩阵的形式表示，则为：

$$F(X)=F(X^*)+\nabla F(X)^T|_{x=x^*}(X-X^*)+\frac{1}{2}(X-X^*)\nabla^2F(X)|_{x=x^*}(X-X^*)+... \tag{3.3}$$

其中，定义梯度为：

$$\nabla F(x)=[\frac{\partial}{\partial x_1}F(x)\ \frac{\partial}{\partial x_2}F(x)\ \cdots\frac{\partial}{\partial x_n}F(x)\ ] \tag{3.4}$$

$$\nabla^2F(x)=\begin{pmatrix}\frac{\partial^2}{\partial x_1^{\ 2}}F(x) & \frac{\partial^2}{\partial x_1\partial x_2}F(x) & \cdots & \frac{\partial^2}{\partial x_1\partial x_n}F(x)\\ \frac{\partial^2}{\partial x_2\partial x_1}F(x) & \frac{\partial^2}{\partial x_2^{\ 2}}F(x) & \cdots & \frac{\partial^2}{\partial x_2\partial x_n}F(x)\\ \vdots & \vdots & & \vdots\\ \frac{\partial^2}{\partial x_n\partial x_1}F(x) & \frac{\partial^2}{\partial x_n\partial x_2}F(x) & \cdots & \frac{\partial^2}{\partial x_n^{\ 2}}F(x)\end{pmatrix} \tag{3.5}$$

称为赫森矩阵。

性能学习的目的是使性能指数得到优化。设性能指数的极小点即最优点。对于一个强极小点x^*，在x^*较小的邻域之外可能会存在比$F(x^*)$更小的点，故x^*又称为局部极小点，对于一个全局极小点，$F(x)$在参数空间内任何其他点的值都比$F(x^*)$大。

梯度的第 i 个元素$\partial F(x)/\partial x_i$是性能指数$F(x)$在$x$轴的一阶导数。赫森矩阵的第$i$i个对角元素$\partial^2F(x)/\partial x_i^{\ 2}$是性能指数$F(x)$沿$x_i$轴的二阶导数。通过分析性能函数在任意方向上的一阶导数，发现当方向向量与梯度的内积最大时斜率最大，故当方向向量与梯度同向时会出现最大斜率。因此，在梯度方向上，逼近性能函数最优点的速度是最快的。

二次函数是一种通用的性能指数。因为二次函数应用广泛，并且在很小的邻域内，特别是在局部极小的附近，许多函数可由二次函数来近似。

对于二次函数而言：

1）如果赫森矩阵的听有特征值为正，则函数有一个强极小点。

2）如果赫森矩阵的所有特征值为负，则函数有一个强极大点。

3）如果赫森矩阵的特征值有正有负，则函数有一个鞍点。

中国知网 https://www.cnki.net

4）如果赫森矩阵的所有特征值为非负，但某些特征值为零，则函数要么有一个弱极小点，要么没有驻点。

5) 如果赫森矩阵的所有特征值为非正，但某些特征值为零，则函数要么有一个弱极大点，要么没有驻点。

如果 A 不可逆(存在为零的特征值)且 d 不为零，则不存在驻点。

3.2 最速下降法

最速下降算法的导数是以一阶泰勒级数展开为基础的。在 x_k 的一阶泰勒级数展开为

$$F(x_{k+1}) = F(x_k + \Delta x_k) \approx F(x_k) + g_k{}^T \Delta x_k \tag{3.6}$$

这里，g_k 为在旧猜测值 x_k 的梯度

$$g_k = -\nabla F(x)|_{x=x_k} \tag{3.7}$$

要使 $F(x_{k+1}) < F(x_k)$，则 $g_k{}^T p_k = \alpha_k g_k{}^T p_k < 0$。此处，$\alpha_k$ 选择为较小的正数，所以 $g_k{}^T p_k < 0$，称满足上式的任意向量称为一个下降方向。如果沿此方向取足够小的步长，函数一定递减。最速下降的方向在式 $g_k{}^T p_k$ 为最大的负数时，这是梯度和方向向量之间的内积。当方向向量与梯度反向时该内积为负，而绝对值最大。所以，最速下降方向的向量为 $p_k = -g_k$，所以，最速下降法的表达式为

$$x_{k+1} = x_k - \alpha_k g_k \tag{3.8}$$

对最速下降法，有两个用来确定学习速度 α_k 的常见方法。

方法一：选择固定的值(例如取 α_k =0.02)，或使用预先确定的变量值(例如 $\alpha_k = 1/k$)。

学习速度越快，步长则更大，收敛的速度有望更快。但是如果学习速度太快，算法将变得不稳定。那么，在保证尽可能快地收敛前提下，对任意函数确定最大可行的学习速度是不可能的，但是，对于二次函数可以确定一个上界，$\alpha < 2/\lambda \max$，最大

中国知网 https://www.cnki.net

的稳定学习速度与二次函数的最大的曲率成反比，曲率说明梯度变化的快慢：如果梯度变化太快，可能会导致跳过极小点，进而使新的迭代点妁梯度的值大于原迭代点的梯度的值(但方向相反)。这会导致每次迭代的步长增大。

学习速度受限于赫森矩阵的最大特征值。在最大特征值的特征向量方向上算法收敛最快，且这个方向上不能越过极小点太远。然而，在最小特征值的特征向量方向上算法将收敛最慢。最后，最小特征值与学习速度共同决定算法收敛的快慢。特征值的大小相差越大，最速下降算法收敛越慢。[2][24]

方法二：使 α_k 每次迭代的性能指数最小化。即选择 α_k 使之沿 $x_k - \alpha_k g_k$ 方向使 $F(x_k + \alpha_k p_k)$ 最小化。

对任意函数的这种最小化需要线性搜索。对二次函数解析线性最小化是可能的。式 $F(x_k + \alpha_k p_k)$ 对 α_k 的导数为

$$\frac{d}{d\alpha_k} F(x_k + \alpha_k p_k) = \nabla F(x)^T |_{x=x_k} \; p_k + \alpha_k p_k{}^T \nabla^2 F(x) |_{x=x_k} \; p_k \qquad (3.9)$$

设该导数为零并求出 α_k 为:

$$\alpha_k = -\frac{\nabla F(x)^T |_{x=x_k} \; p_k}{p_k{}^T \nabla^2 F(x) |_{x=x_k} \; p_k} \qquad (3.10)$$

这里 A_k 为在 x_k 的赫森矩阵 $A_k = \nabla^2 F(x) |_{x=x_k}$ 。

3.3 牛顿法

牛顿法则基于二阶泰勒级数：

$$F(x_{k+1}) = F(x_k + \Delta x_k) \approx F(x) + g_k{}^T \Delta x_k + \frac{1}{2} x_k{}^T A_k X_k \qquad (3.11)$$

牛顿法的原理是求 $F(x)$ 的二次近似的驻点，用二次函数的一阶导数 $\nabla F(x) = Ax + d$，求这个二次函数对 Δx_k 的梯度并设它为零，则有

$$g_k + A_k \Delta x_k = 0 \qquad (3.12)$$

于是可将牛顿法定义为

中国知网 https://www.cnki.net

$$\Delta x_k = -A_k^{-1} g_k \tag{3.13}$$

如果函数 $F(x)$ 不是二次函数，则牛顿法一般不能在一步内收敛。实际上根本无法确定它是否收敛，因为这取决于具体的函数和初始点。

牛顿法在许多应用中都能快速收敛，这是因为在一个强极小点的较小的邻域内，解析函数能够被二次函数精确近似：离极小点越近，牛顿法越能精确标识该极小点。

牛顿法的特点是尽管它的收敛速度通常比最速下降法更快，但其表现很复杂，表现在可能收敛到局部的极小点（陷入局部极小），或者收敛到鞍点，此外，算法还可能振荡和发散。如果学习速度不太快或每步都实现线性极小化，最速下降法能够确保收敛。

牛顿法的另一个问题是需要对赫森矩阵及其逆阵的计算和存储，将对最速下降法的定义式（3.8）与牛顿法的定义式（3.13）相比，可以发现当 $A_k = A_k^{-1} = I$ 时，它们的搜索方向将相同。

3.4 共轭梯度法

通过用共轭代替正交的方法调整搜索方向可以提高性能。如果使用共轭方向，函数最多能在 n 步的迭代中被最小化。

最速下降法在每次迭代用线性搜索时相继迭代的搜索方向相互正交。对于轮廓线为椭圆的二次函数，这将产生短步长的锯齿形轨迹。因此，二次搜索方向并非最好的选择，而共轭方向恰恰就是一个确保二次终结法的搜索方向的集合。

对于形式为

$$F(x) = \frac{1}{2} x^T A x + d^T x + c \tag{3.14}$$

的二次函数，当且仅当

$$p_k^T A p_j = 0, k \neq j \tag{3.15}$$

时，称向量集合 $\{p_k\}$ 对于一个正定赫森矩阵 A 两两共扼。对于正交向量，存在无穷个能张成一个 n 维空间的两两共轭向量集。由 A 的特征向量组成的共轭向量集也是其中之一。由于特征向量构成函数轮廓线的主轴，沿赫森矩阵的特征向量搜索就能准确

中国知网 https://www.cnki.net

地使二次函数极小化。但是，在实际运用中，这将是十分困难的，这需要计算二阶导数的算法。

已经证明，如果存在沿一个共轭方向集$\{p_1, p_2, ..., p_n\}$的准确线性搜索序列，就能在最多n次搜索内实现具有n个参数的二次函数的准确极小化。问题在于如何构造这些共轭搜索方向?首先来看式（3.4.2）中不用赫森矩阵的共轭条件。注意到对于二次函数，有

$$\nabla F(x) = Ax + d \tag{3.16}$$

$$\nabla^2 F(x) = A \tag{3.17}$$

将这些等式组合起来，能发现在 k+1 次迭代时梯度的变化为：

$$\Delta g_k = g_{k+1} - g_k = (Ax_{k+1} + d) - (Ax_k + d) = A\Delta x_k \tag{3.18}$$

同时，由于

$$\Delta x_k = (x_{k+1} - x_k) = \alpha_k p_k \tag{3.19}$$

选择α_k使函数$F(x)$在p_k方向上极小化。

所以式（3.4.2）的共轭条件可重写成

$$\alpha_k p_k^T A p_j = x_k^T A p_j = g_k^T p_j = 0, k \neq j \tag{3.20}$$

这样就把共轭条件表示成算法相继迭代的梯度变化形式。如果搜索方向与梯度变化方向垂直，则它们共轭。

第一次搜索方向p_0是任意的，而p_1可以是与Δg_0垂直的任意向量。所以共轭向量集的数量是无限的。通常从最速下降法的方向开始搜索：

$$p_0 = -g_0 \tag{3.21}$$

每次迭代都要构选一个与$\{\Delta g_0, \Delta g_1, ... \Delta g_{k-1}\}$正交的向量$p_k$。可将迭代形式简化为：

$$p_k = -g_k + \beta_k p_{k-1} \tag{3.22}$$

确定系数的方法很多，对二次函数产生的结果相同。通常选择：

中国知网 https://www.cnki.net

$$\beta_k = \frac{\Delta g_{k-1}^T g_k}{\Delta g_{k-1}^T p_{k-1}} \tag{3.23}$$

(由 Hestenes 和 Steifel 确定)

$$\beta_k = \frac{g_k^T g_k}{g_{k-1}^T g_{k-1}} \tag{3.24}$$

(由 Fletcher 和 Reeves 确定)

$$\beta_k = \frac{\Delta g_{k-1}^T g_k}{g_{k-1}^T g_{k-1}} \tag{3.25}$$

（由 Polak 和 Ribiere 确定）

共轭梯度算法的步骤：

Step1：选择如式（3.21）所示的与梯度相反的方向作为第一次搜索方向。

Step2：根据式（3.19）进行下一步搜索，确定α_k以使函数沿搜索方向极小化。

Step3：根据式（3.22）确定下一个搜索方向，用式（3.23），（3.24）或式（3.25）计算β_k。

Step4：如果算法不收敛，回到 Step2。

和最速下降法使用正交的搜索方向的方法不同，共轭梯度算法调节第二次搜索方向要以使它通过函数极小点。

3.5 小结

最速下降法的优点在于其存储量小，但该算法收敛速度慢。最速下降法慢的原因，一方面是由于问题本身的形态决定的，另一方面来自于负梯度方向，因为最速下降方向是函数在$x^{(k)}$的局部性态，试验点$x^{(k+1)}$一旦沿r_k离开$x^{(k)}$，r_k就不再是最速下降方向了。[26]更具体地，对于二次连续可微的函数$f: R^n \to R$，采用精确性搜索的最速下降法产生的点列$\{x^{(k)}\}$收敛于x^*，且$\nabla^2 f(x)$正定，则存在关系式：

$$f(x^{(k+1)}) - f(x^*) \le \left(\frac{\kappa - 1}{\kappa + 1}\right)^2 (f(x^{(k)}) - f(x^*)) \tag{3.26}$$

中国知网 https://www.cnki.net

其中，$\kappa = \lambda\max/\lambda\min$，$\lambda\max$ 和 $\lambda\min$ 分别是 $\nabla^2 f(x^*)$ 的最大特征值和最小特征值。所以，$\kappa = \lambda\max/\lambda\min$ 越大，收敛的越慢，而当该值为 1 时，可一步收敛到极小点。

Newton 法具有二次收敛性，但当 $\nabla^2 f(x^{(k)})$ 不正定时，不能保证算法产生的方向是 f 在 $x^{(k)}$ 处的下降方向。特别，当 $\nabla^2 f(x^{(k)})$ 奇异时，Newton 方向可能不存在。修正 Newton 法可克服 Newton 法的上述困难，但在修正 Newton 法中，$v_k > 0$ 的选取尤为重要。若参数 v_k 过小，则相应的修正 Newton 法方向仍然不能保证是 f 在 $x^{(k)}$ 处的下降方向。参数 v_k 过大，则会影响收敛速度。此处，Newton 法及其修正形式都需要计算函数 f 的二阶导数。[24]

在许多情况下，待搜索的参数空间实际上并不是一个欧氏空间，而是一个黎曼空间。例如，对于固定结构的多层前向网络，它的 m 个自由参数（可调权值和阈值）构成了一个 m 维空间，它可看作是另一个更广泛的空间（如神经网络的所有可能映射的集合）的子流形。S.Amari 从微分几何的观点研究了这一流形的重要性质，提出了信息几何（information geometry）的概念。此时，常规梯度不代表目标函数的最陡下降方向，为此提出了自然梯度（natural gradient）的概念。并说明了用自然梯度代替常规梯度时，不仅在多层前向网络学习中有许多优点，而且可对盲信号处理中的一些不同算法给出统一的表达形式。[2]

中国知网 https://www.cnki.net

第四章 TWEBP 算法

多层前向神经网络模型是神经网络理论和应用研究中使用最广泛的模型，已经在函数逼近、模式识别、图像处理、自适应控制等领域得到应用。在训练前向网络的算法中，误差反向传播（BP）算法是最重要也是运用最广的一种算法，但是由于 BP 算法实质是一种梯度下降的搜索算法，它存在着其固有的不足，如收敛速度较慢，易于陷入误差函数的局部极值等。因此有必要对 BP 算法进行改进，提高收敛速度及精度。

本文提出一种改进的 BP 算法---三项权值外推算法（TWEBP, the three-term weight extrapolations BP algorithm），不仅可以加快网络学习的速度，而且可以得到满意的精度。此算法结合权值外推和三项因子的思想来调整输入层到输出层的权值和阈值，通过 BP 算法的循环训练完成网络的学习。

4.1 趋势外推思想

4.1.1 趋势外推

趋势外推法（Trend extrapolation）是根据过去和现在的发展趋势推断未来的一类方法的总称，用于科技、经济和社会发展的预测，是情报研究法体系的重要部分。趋势外推的基本假设是未来系过去和现在连续发展的结果。

趋势外推法首先由 R.赖恩（Rhyne）用于科技预测。他认为，应用趋势外推法进行预测，主要包括以下 6 个步骤：（1）选择预测参数；（2）收集必要的数据；（3）拟合曲线；（4）趋势外推；（5）预测说明；（6）研究预测结果在制订规划和决策中的应用。

趋势外推法是在对研究对象过去和现在的发展作了全面分析之后，利用某种模型描述某一参数的变化规律，然后以此规律进行外推。为了拟合数据点，实际中最常用的是一些比较简单的函数模型，如线性模型、指数曲线、生长曲线、包络曲线等。

中国知网 https://www.cnki.net

4.1.2 BPWE 算法

1 原理

对未来某回合的网络权值的预测是依赖于此前的 M 个回合的。假定从第 K 个回合开始，此前 M 个回合满足下列两个条件

1) w_k 的每个单独的部分的序列（$w_{K-M+1}, w_{K-M+2}, ..., w_K$）是严格单调递增或递减，并且

2) 每个序列中连续的两个元素之间的距离是单调递增的，也就是：

$$|w_k - w_{k+1}| > |w_{k+1} - w_{k+2}| \text{ 其中 } k = K-M+1, K-M+2, ..., K-2 \qquad (4.1)$$

的权值序列，则称之为半指收敛权值序列。

在训练的第 K 个回合，如果过去 M 个回合的权值序列是否满足为半指收敛的条件，通过外推，就可以直接预测将来第 N 个回合之后的权值，从而减少了回合数，降低了计算量，提高了收敛速度。

为了进行外推，在每个回合结束时记录权值。通过分别在 BP 训练每个网络收敛性的过程中分别检查权值 w，发现可以通过一般形式为 $f(x) = a - b\exp(-cx)$ 的函数来进行使用外推。

$$f(x) = a - b\exp(-cx) \quad \text{其中，b}\neq 0 \quad \text{c}>0 \quad \text{a 为任意常数。} \qquad (4.2)$$

自变量 x 表示回合数，也就是，对第 k 回合有

$$f(k) = w(k) \qquad (4.3)$$

采用这个技巧，未来某回合的网络权值通过权值外推法是可以预测到的。

2 步长的确定

在一个训练过程的初期（此时 K 是很小的），权值向量的收敛性仅用一个短序列（$w_{K-M+1}, w_{K-M+2}, ..., w_K$）的粗略的外推就可以充分地得到改进。而另一方面，在更高的阶段的训练中，如果 w_k 处于接近局部最小点的位置，那么，必须要一个先前求值向量的长序列来进行非常准确的外推，因为，有越过目标的危险。因此，可以在执行外推时将 M 的值扩大 2 倍。假如，第一次外推的开始值 $M_1 = 8$，那么，第 l 次外推时

中国知网 https://www.cnki.net

$M = M_l = 2^{l+2}$。在第 l 次外推中，这种方法通过计算出来的过去 M_l 个回合中的权值向量的信息为指定的未来的 N_l 回合训练的结果进行预测。外推权值用于“跳过”标准 BP 算法的 N_l 个回合。因此，明显地可以节省计算量。跳过一个局部最小点的危险的原因之一是它不擅长预测最终权值的向量 w_*。于是用 N 代替 N_l，定义

$$N = N_l = 70M_l = 35 \times 2^{(l+3)} \tag{4.4}$$

这里因子 70 是通过试验经验确定的。为了预测要考虑的未来回合数 N_l 比更大的数 M_l 还大，因为要获得更加精确的预测需要更大数目的前期的权值向量。

3 算法实现

通过半指收敛权值序列的定义，序列（$w_{K-M+1}, w_{K-M+2}, ..., w_K$）按照等式（4.3）中函数 f 的意义外推，即

$$w_k \approx a - b \cdot \exp(-ck) \tag{4.5}$$

其中，$k = K-M+1, K-M+2, ..., K$

函数 f 用来预测 $N = N_l$ 个附加回合之后的未来权值，并且设置权值 w_{k+1} 等于这个预测，即

$$w_{k+1} = f(K+N) \tag{4.6}$$

$$w_{k+1} = a - b \cdot \exp(-c(K+N)) \tag{4.7}$$

其中，

$$c = \frac{1}{M-2} \sum_{k=K-M+1}^{K-2} (\log|w_k - w_{k+1}| - \log|w_{k+1} - w_{k+2}|) \tag{4.8}$$

$$|b| = \frac{1}{(M-1)(e^c - 1)} \sum_{k=K-M+1}^{K-1} (\exp(c(k+1)) \cdot |w_k - w_{k+1}|) \tag{4.9}$$

$$a = \frac{1}{M}\left(\sum_{k=K-M+1}^{K} w_k + b \sum_{k=K-M+1}^{K} \exp(-c \cdot k) \right) \tag{4.10}$$

（当权值序列严格单调递增时取 $b > 0$，当权值序列严格单调递减时取 $b < 0$）。

中国知网 https://www.cnki.net

4.2 TBP 算法

在网络学习中，学习步长η的选择非常重要，η大收敛快，但过大的η可能引起振荡；η小可避免振荡，但收敛速度慢。增加动量项正是解决这一矛盾的最简单的办法，即

$$\Delta w_{ji}(n)=\alpha\Delta w_{ji}(n-1)+\eta\delta_j(n)y_i(n) \quad 0<\alpha<1 \tag{4.11}$$

式中第二项是常规 BP 法的修正量，第一项称之为动量项，α为某一正数。

对上式的改进是通过增加一个附加项来提高 BP 算法的学习速度。该项是对在每个回合中表示输出和目标差值的$e(w(k))$的均衡。在批量学习中，$e(w(k))=[e_s\ e_s\ e_s\cdots e_s\ ]^T$，其中向量$e$由适当的维数组成，并且$e_s=\sum_{s=1}^{n}[T_s-O_s^m]$，$O_{s,i}^m$表示 m 层上的第 i 个输出，$T_s$是第 s 个输入样本的期望值。

因此，对权值的调整可由下式来完成，即

$$\Delta w(k)=\alpha(\nabla E(w(k)))+\beta\Delta w(k-1)+\gamma e(w(k)) \tag{4.12}$$

这里，γ是均衡因子。即上式给出的 BP 算法有三项，一是均衡$E(w(k))$的导数，二是均衡增量权值的前一个值（即$\Delta w(k-1)$），三是均衡$e(w(k))$。

4.3 TWEBP 算法

TBP 算法是在带有动量项的 BP 算法的基础上增加了一项均衡项（proportional factor），它的突出的优点在于可以通过增大均衡因子γ的值来加速收敛。而 BPWE 算法是通过分析权值的收敛轨迹来加速收敛的。因为 BPWE 算法和 TBP 算法都是基于权值调整的改进算法，而考虑将 TBP 算法中的三项因子融入到 BPWE 算法中，从而使后者对权值的调整由原来的两项增加为三项，加快收敛速度，避免陷入局部极小点，从而提出一种新的学习算法---TWEBP 算法。

具体的步骤如下：

Step1：给定输入层至隐层的初始权值w及偏置值θ，学习率η。

中国知网 https://www.cnki.net

Step2：将训练集样本输入前向网络按照 BP 算法进行训练，调整输入层至隐层的初始权值 w，偏置值 θ，调整隐层至输出层的初始权值和偏置值。

Step3：记录各权值，统计满足半指收敛的权值序列。若是半指收敛权值序列，则 Step4，否则，Step5。

Step4：根据式 $w_{k+1}=a-b\cdot\exp(-c(K+N))$ 进行趋势外推，计算出 N 个回合之后的权值。

Step5：根据 $\Delta w(k)=\alpha(\nabla E(w(k)))+\beta\Delta w(k-1)+\gamma e(w(k))$，进一步加速调整权值。

Step6：计算神经网络的误差函数，如网络训练达到精度要求，转入 Step7，否则 Step2。

Step7：结束。

4.4 计算机仿真

本文中提出了一种基于外推思想和三项因子思想的新的 BP 学习算法，为了验证本算法的优点，采用 XOR 问题、三分类问题和函数逼近问题在 Matlab 平台[3][25]上进行了试验，得到的较好的效果。

4.4.1 XOR 问题

采用了神经网络研究的经典问题，这个问题曾经一度困扰了神经网络的研究者，阻碍了神经网络的发展，该问题是用离散感知器神经元解决异或问题，即在二维平面上规定，（-1，1）和（1，-1）属于一类，（-1，-1）和（1，1）属于另一类。

采用了一个 2 输入 2 输出的 3 层 BP 网络对三类样本进行学习，网络的隐节点数为 3。

学习参数设定为网络隐层和输出层的激活函数采用标准 Sigmoid 函数，学习率为 1，三项系数为 0.8，目标误差为 ε 为 0.5，最大训练次数为 10000 次。经过训练后，随机产生样本 5000 个，对网络进行测试，得到结果。

对 BP 算法、BPWE 算法、TBP 算法和 TWEBP 算法等算法分别进行 10 次试验，数据如表 4-1。在表 4-2 中对其进行了统计。

中国知网 https://www.cnki.net

表 4-1 XOR 问题测试结果

	BP 算法		BPWE 算法		TBP 算法		TWEBP 算法	
试验次数	训练次数	正确率	训练次数	正确率	训练次数	正确率	训练次数	正确率
第 1 次	6607	0.9818	3732	0.9834	1326	0.9884	1242	0.9704
第 2 次	4175	0.987	3618	0.9848	1783	0.9648	1277	0.9682
第 3 次	7292	0.986	2738	0.9968	2046	0.9356	1095	0.9914
第 4 次	3357	0.9964	3425	0.9926	1982	0.978	1047	0.9876
第 5 次	5240	0.9974	4025	0.9956	2313	0.9688	1237	0.9706
第 6 次	1961	0.9868	1344	0.9686	1992	0.9908	1630	0.9758
第 7 次	3099	0.9928	4789	0.9934	1431	0.9718	1193	0.9894
第 8 次	2879	0.9862	4115	0.987	1680	0.9884	1186	0.991
第 9 次	3958	0.9824	2215	0.99	1860	0.989	1629	0.9702
第 10 次	3054	0.988	3001	0.9839	1733	0.971	1520	0.9646
平均	4162	0.9885	3300	0.9876	1815	0.9747	1306	0.9779

表 4-2 XOR 问题测试结果的统计分析

	BP 算法	BPWE算法	TBP 算法	TWEBP 算法
试验次数	10	10	10	10
平均训练次数	4162	3300	1733	1520
最大训练次数	7292	4789	2313	1630
最小训练次数	1961	1344	1326	1047
平均正确率	0.9885	0.9876	0.9747	0.9779
最高正确率	0.9974	0.9968	0.9908	0.9914
最低正确率	0.9818	0.9686	0.9356	0.9646

中国知网 https://www.cnki.net

从表 4-1 和表 4-2 中可以看出，在 10 次试验中，BP 算法要收敛到指定的 SSE，最高需要 7292 次，最低需要 1961 次，平均需要 4162 次，最高正确率为 0.9974，最低正确率为 0.9818，平均正确率为 0.9885。BPWE 算法最高需要 4789 次，最低需要 1344 次，平均需要 3300 次，最高正确率为 0.9968，最低正确率为 0.9686，平均正确率为 0.9876。TBP 算法最高需要 2313 次，最低需要 1326 次，平均需要 1733 次，最高正确率为 0.9908，最低正确率为 0.9356，平均正确率为 0.9747。而 TWEBP 算法最高需要 1630 次，最低需要 1047 次，平均只需要 1520 次，最高正确率为 0.9914，最低正确率为 0.9646，平均正确率则为 0.9779。

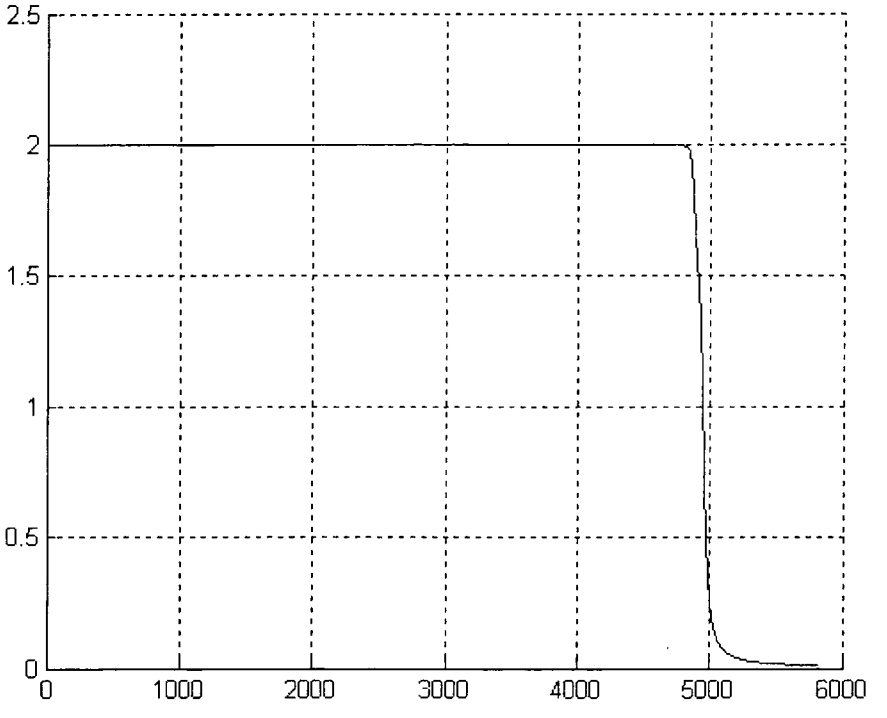


图 4-1 BP 算法的收敛效果

中国知网 https://www.cnki.net

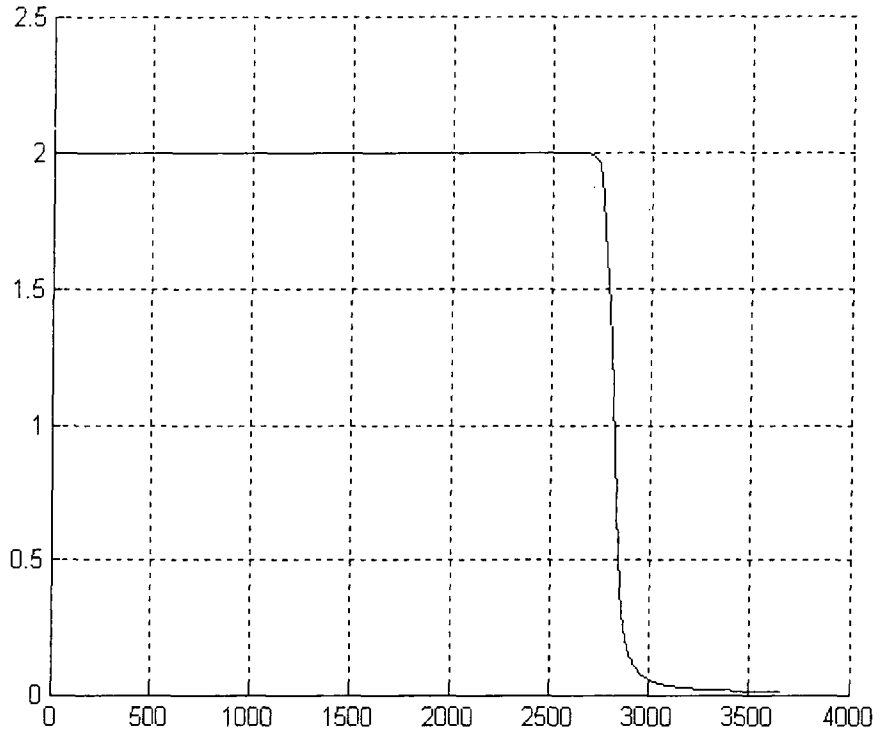


图 4-2 BPWE 算法的收敛效果

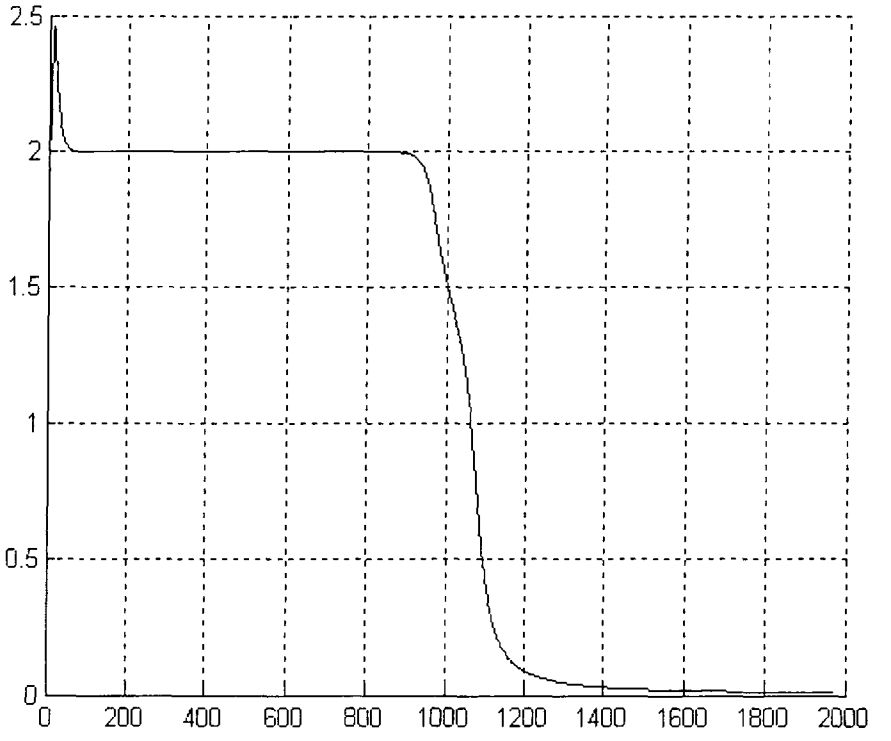


图 4-3 TBP 算法的收敛效果

中国知网 https://www.cnki.net

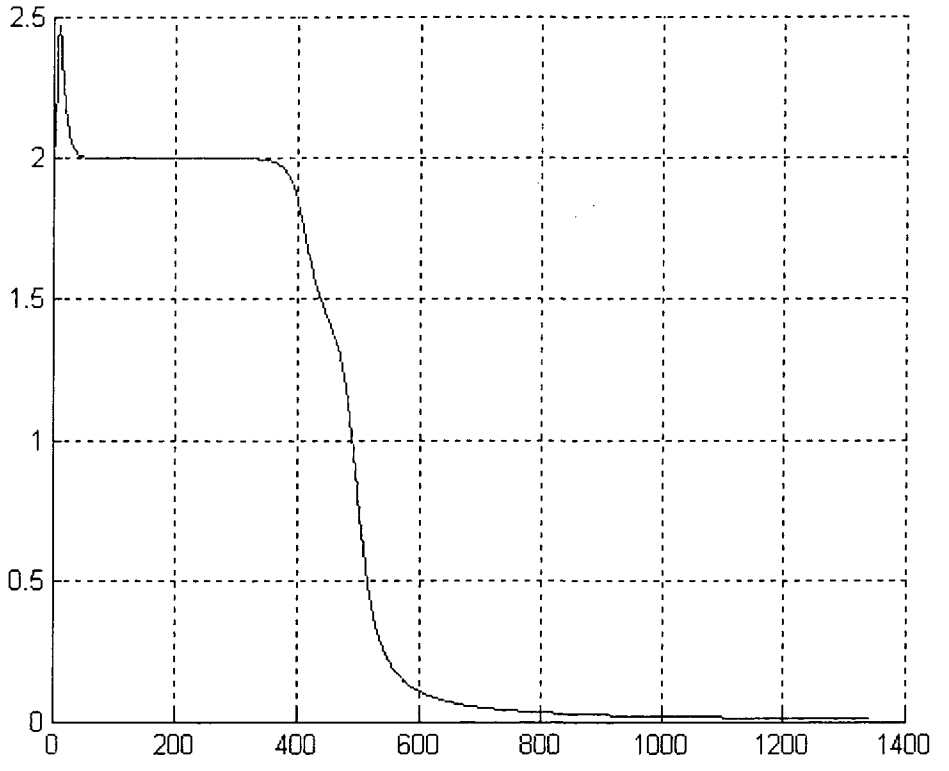


图 4-4 TWEBP 算法的收敛效果

4.4.2 三分类问题

采用了 Cohn 三角形概念学习的例子的推广，这是典型的多分类的例子[26]。在(-2，2）×（-2，2）的范围内随机产生 20 个训练样本。如图 4-5 所示。规定，三角形的样本属于 c2 类，矩形的样本属于 c3 类，其余的样本属于 c1 类。

采用了一个 2 输入 3 输出的 3 层 BP 网络对三类样本进行学习，网络的隐节点数为 10。规定 c1 类样本的目标输出为$[1\ 0\ 0]^T$,c2 类的样本的目标输出为$[0\ 1\ 0]^T$，c3 类的样本的目标输出为$[0\ 0\ 1]^T$。

学习参数设定为：网络隐层和输出层的激活函数采用标准 Sigmoid 函数，学习率为 1，三项系数为 0.8，目标误差为ε为 0.01，最大训练次数为 5000 次。经过训练后，对网络进行测试，用同样的方法产生样本 5000 个，判断分类的结果。

对 BP 算法、TBP 算法、BPWE 算法和 TWEBP 算法等算法分别进行 20 次试验后的数据如表 4-3。在表 4-4 中对其进行了统计。图 4-5 为随机产生的样本的分布图。各算法的收敛效果见图 4-6 至 4-9。

中国知网 https://www.cnki.net

表 4-3 三分类问题测试结果

	BP 算法		TBP 算法		BPWE 算法			TWEBP 算法		
试验次数	训练次数	正确率	训练次数	正确率	训练次数	正确率	外推次数	训练次数	正确率	外推次数
1	4097	0.6438	1126	0.7448	1129	0.7186	8	836	0.7584	5
2	1878	0.75	1455	0.7562	1515	0.8368	8	721	0.6864	5
3	3129	0.8658	3431	0.8214	3995	0.766	9	1048	0.796	6
4	4352	0.7942	1248	0.7572	1946	0.8174	8	2040	0.864	6
5	1906	0.6222	2597	0.8132	2577	0.789	9	3086	0.7674	7
6	2220	0.761	2874	0.7392	4253	0.7762	10	711	0.8114	5
7	2741	0.751	2595	0.7482	2170	0.7744	9	2725	0.797	7
8	1658	0.8236	2271	0.8062	1293	0.8754	8	1344	0.8266	6
9	2813	0.8118	2740	0.7836	1797	0.7756	8	2327	0.671	7
10	1109	0.8166	2687	0.7804	2036	0.6422	8	3059	0.685	7
11	1661	0.8384	2244	0.7718	2495	0.718	9	1402	0.8188	6
12	5114	0.8134	4576	0.7602	2334	0.8586	9	743	0.6836	5
13	1673	0.8528	2188	0.8408	2528	0.8046	9	2015	0.7634	6
14	3384	0.7282	2668	0.841	741	0.6756	7	2702	0.8228	7
15	1850	0.7726	4398	0.811	2021	0.753	8	2205	0.8274	7
16	2840	0.8236	2917	0.7746	1812	0.8026	8	2412	0.7452	7
17	754	0.8046	3311	0.7962	3928	0.6416	9	2090	0.7262	7
18	2764	0.7502	3798	0.6936	1653	0.7876	8	2878	0.6372	7
19	4368	0.8558	2796	0.8298	1265	0.7406	8	2720	0.8258	7
20	2909	0.7016	1038	0.6938	2813	0.8118	9	1770	0.8402	6
平均	2661	0.7791	2648	0.7782	2215	0.7683	8.5	1942	0.7677	6.3

中国知网 https://www.cnki.net

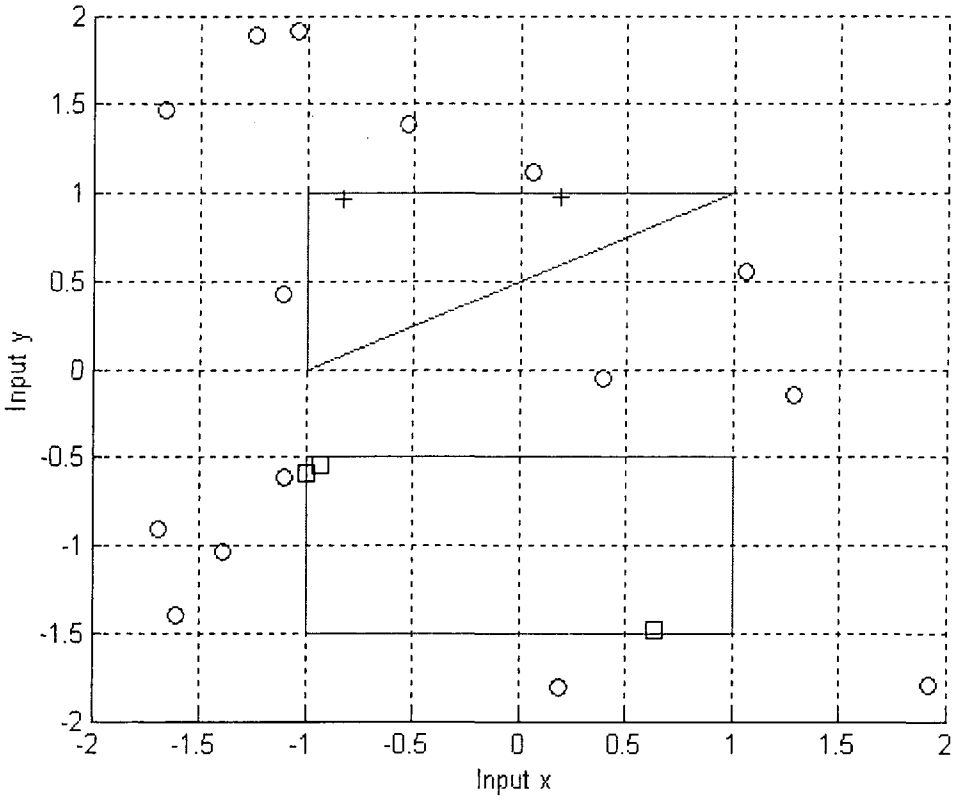


图 4-5 三分类问题的样本分布图

从表 4-4 中可以看出，在 20 次试验中，BP 算法要收敛到指定的 SSE，最高需要 4368 次，最低需要 754 次，平均需要 2661 次，最高正确率为 0.8658，最低正确率为 0.6222，平均正确率为 0.7791。TBP 算法最高需要 4576 次，最低需要 1038 次，平均需要 2648 次，最高正确率为 0.8408，最低正确率为 0.6936，平均正确率为 0.7782。BPWE 算法最高需要 4253 次，最低需要 741 次，平均需要 2215 次，最高正确率为 0.8368，最低正确率为 0.6416，平均正确率为 0.7683。而 TEWBP 算法最高需要 3086 次，最低需要 711 次，平均只需要 1942 次，最高正确率为 0.864，最低正确率为 0.6372，平均正确率则为 0.7677。

TWEBP 算法的平均训练次数的不到 2000 次，低于 BP 算法和两种经过改进的 BP 算法，其中较 BP 算法和 TBP 算法在训练次数减少了近 40%的情况下达到了收敛，并且准确率相当。同时，它较外推算法训练次数减少了 20%，外推次数减少了 25%。显然，训练次数的减少必然提高了算法的收敛速度，而外推次数的减少也降低了计算量，因而，也提高了算法的收敛速度。问题在于各算法都会出现不能收敛到预定精度

中国知网 https://www.cnki.net

的情况，并且不能收敛到预定精度的次数除权值外推法稍多以外，其他算法次数相当。

表 4-4 三分类问题测试结果的统计分析

	BP 算法	TBP 算法	BPWE 算法	TWEBP 算法
试验次数	20	20	20	20
平均训练次数	2661	2648	2215	1942
最高训练次数	4368	4576	4253	3086
最低训练次数	754	1038	741	711
平均正确率	0.7791	0.7782	0.7683	0.7677
最高正确率	0.8658	0.8408	0.8368	0.864
最低正确率	0.6222	0.6936	0.6416	0.6372
平均外推次数	--	--	8.5	6.3

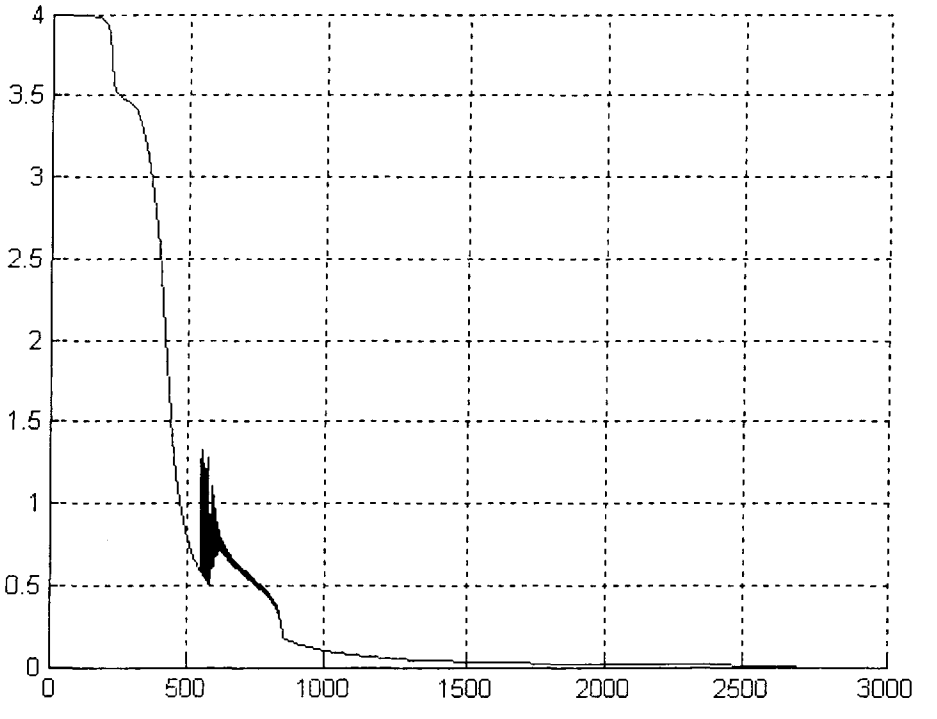


图 4-6 BP 算法的收敛效果图

中国知网 https://www.cnki.net

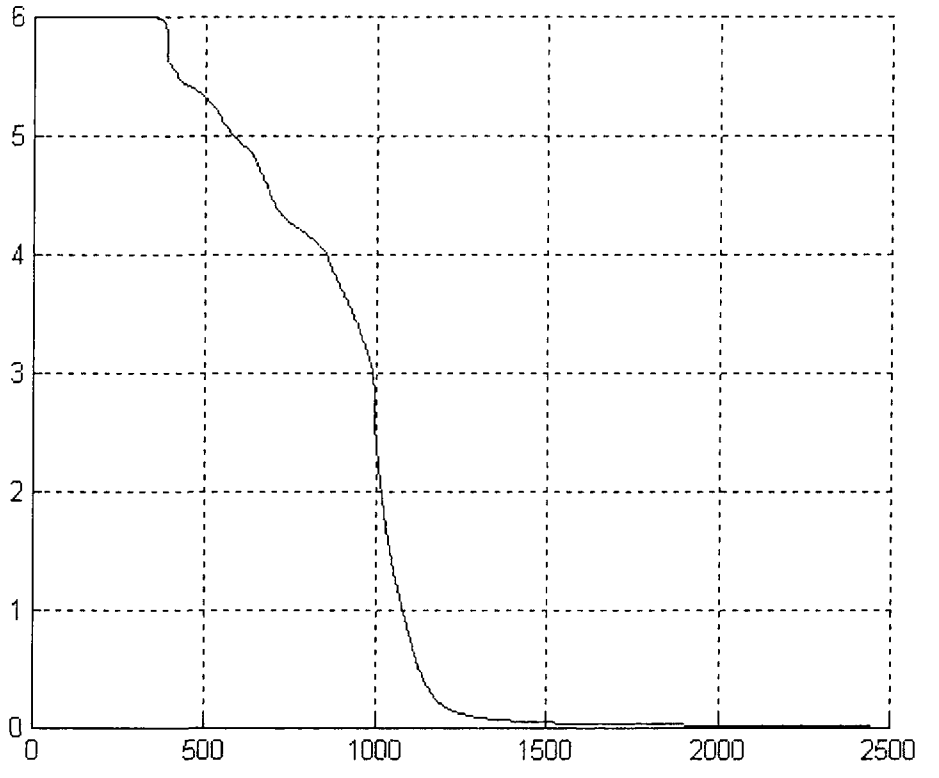


图4-7 TBP算法的收敛效果图

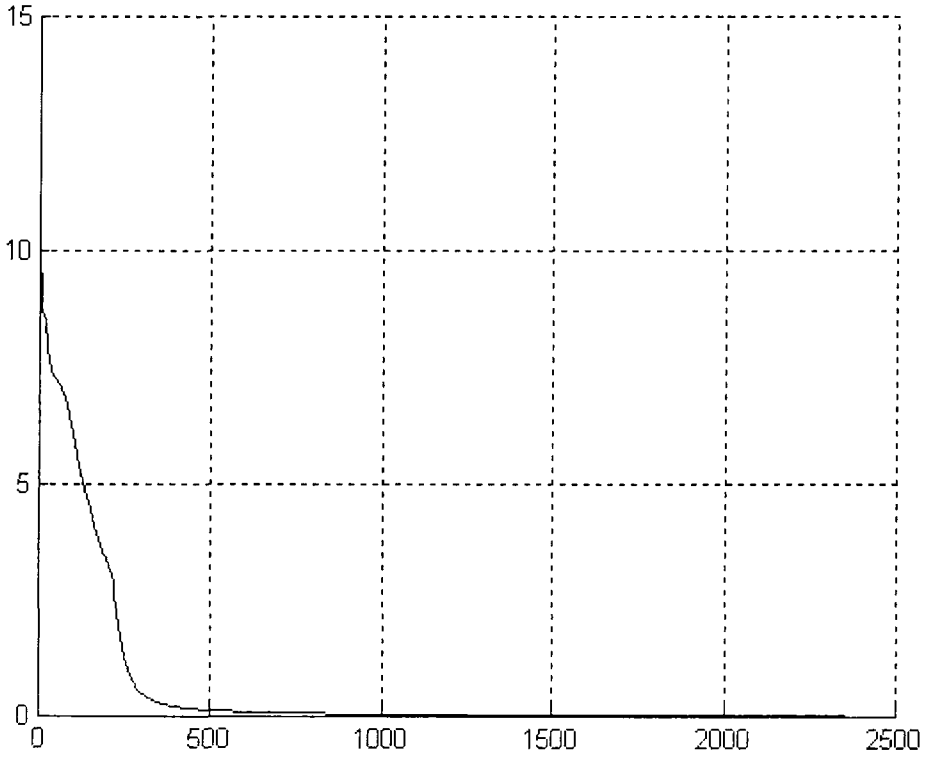


图4-8 BPWE算法的收敛效果图

中国知网 https://www.cnki.net

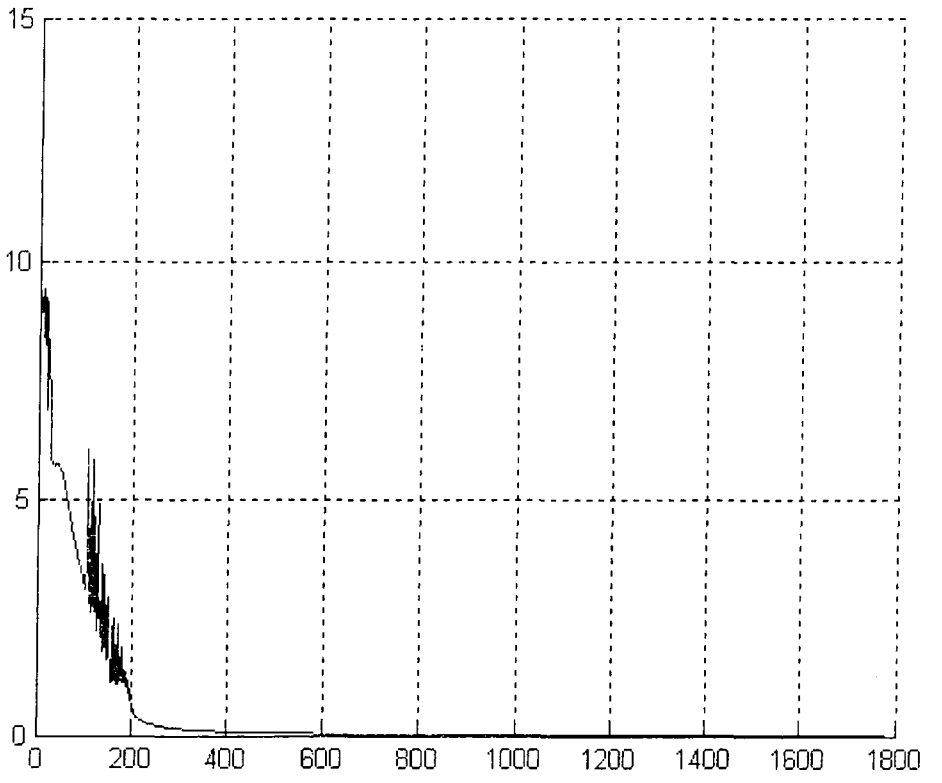


图4-9 TWEBP算法的收敛效果图

4.4.3 函数逼近问题

本例采用了由 Mackay 提出的 Hermit 多项式的逼近问题[16]：

$$F(x)=1.1(1-x+2x^2)\exp(-x^2/2) \quad x\in R$$

训练样本的产生方式为：样本数 N=100，其中样本输入 x_i 服从区间[-4，4]内的均匀分布，样本输出为 $F(x_i)+e_i$，e_i 为添加的噪声，服从均值为 0，标准差为 0.1 的正态分布。对于该函数逼近问题，可以用一个 1 输入 1 输出的 3 层 BP 网络对样本进行拟合，网络的隐节点数为 10。隐层采用标准 Sigmoid 激活函数，输出层采用线性激活函数，学习率 lr=0.003，三项系数 pf=0.5，目标误差为 ε=0.5，训练次数分别为 10000 和 20000，初始权值和偏移值取[-0.1，0.1]内的随机值。

对 BP 算法、BPWE 算法、TBP 算法和 TWEBP 算法分别进行 10 次试验，每次试验训练次数为 10000 次时产生的数据如表 4-5，对该表的数据在表 4-6 中进行了统计。每次试验训练次数为 20000 次时产生的数据如表 4-7，对该表的数据在表 4-8 中

中国知网 https://www.cnki.net

进行了统计。

表 4-5 函数逼近问题的测试结果（训练次数=10000）

	BP 算法	BPWE 算法	TBP 算法	TWEBP 算法
第 1 次	1.137	1.0641	1.0168	1.0168
第 2 次	9.7067	9.5782	0.9433	0.9433
第 3 次	6.1288	6.1021	1.1167	1.1167
第 4 次	0.8551	0.819	1.1323	1.1323
第 5 次	10.7401	10.7002	1.0912	1.0912
第 6 次	1.0487	1.011	0.9163	0.9163
第 7 次	1.2142	1.1326	1.2006	1.2006
第 8 次	1.106	1.0925	1.0974	1.0974
第 9 次	7.7266	7.7191	7.7281	7.7281
第 10 次	1.1438	1.0703	1.0306	1.0306

表 4-6 函数逼近问题的测试结果的统计分析（训练次数=10000）

	BP 算法	BPWE 算法	TBP 算法	TWEBP 算法
训练回合数	10000	10000	10000	10000
训练次数	10	10	10	10
平均 SSE	4.0807	4.0289	1.7273	1.7273
最大 SSE	10.7401	10.7002	7.7281	7.7281
最小 SSE	0.8551	0.819	0.9163	0.9163

由于在训练次数为 10000 次的情况下，各算法都不能收敛到指定的 SSE，所以，对 BP 算法、BPWE 算法、TBP 算法和 TWEBP 算法在各进行 10 次试验进行了试验。通过试验发现，训练次数为 10000 次时，BP 算法收敛的最大 SSE 为 10.7401，最小 SSE 为 0.8551，平均 SSE 为 4.0807。BPWE 算法收敛的最大 SSE 为 10.7002，最小

中国知网 https://www.cnki.net

SSE 为 0.819，平均 SSE 为 4.0289。TBP 和 TWEBP 算法敛的最大 SSE 为 7.7281，最小 SSE 为 0.9163，平均 SSE 为 1.7273。

从表 4-6 可以看出，当最大训练次数定为 10000 时，TWEBP 算法在 10000 次内的平均误差率 SSE 为 BP 算法误差率的 40%，前者的收敛速度明显较后者快。

表 4-7 函数逼近问题的测试结果（训练次数=20000）

	BP 算法	BPWE 算法	TBP 算法	TWEBP 算法
第 1 次	0.773	0.7647	0.7495	0.7495
第 2 次	1.0101	0.9993	0.9429	0.9429
第 3 次	0.9483	0.9351	0.9001	0.9001
第 4 次	0.7015	0.6937	0.6865	0.6865
第 5 次	0.8583	0.8543	0.886	0.886
第 6 次	0.8593	0.8535	0.8713	0.8713
第 7 次	0.9037	0.896	0.921	0.921
第 8 次	0.8586	0.8551	0.8387	0.8387
第 9 次	0.8049	0.7992	0.7964	0.7964
第 10 次	7.4732	7.4693	7.4673	7.4673

训练次数为 20000 次时，BP 算法收敛的最大 SSE 为 7.4732，最小 SSE 为 0.7015，平均 SSE 为 1.5190。BPWE 算法敛的最大 SSE 为 7.4693，最小 SSE 为 0.6937，平均 SSE 为 1.5120。TBP 和 TWEBP 算法敛的最大 SSE 为 7.4673，最小 SSE 为 0.6865，平均 SSE 为 1.5059。

表 4-8 函数逼近问题的测试结果统计分析（训练次数=20000）

	BP 算法	BPWE 算法	TBP 算法	TWEBP 算法
平均 SSE	1.5190	1.5120	1.5059	1.5059
最小 SSE	0.7015	0.6937	0.6865	0.6865
最大 SSE	7.4732	7.4693	7.4673	7.4673
训练次数	20000	20000	20000	20000

中国知网 https://www.cnki.net

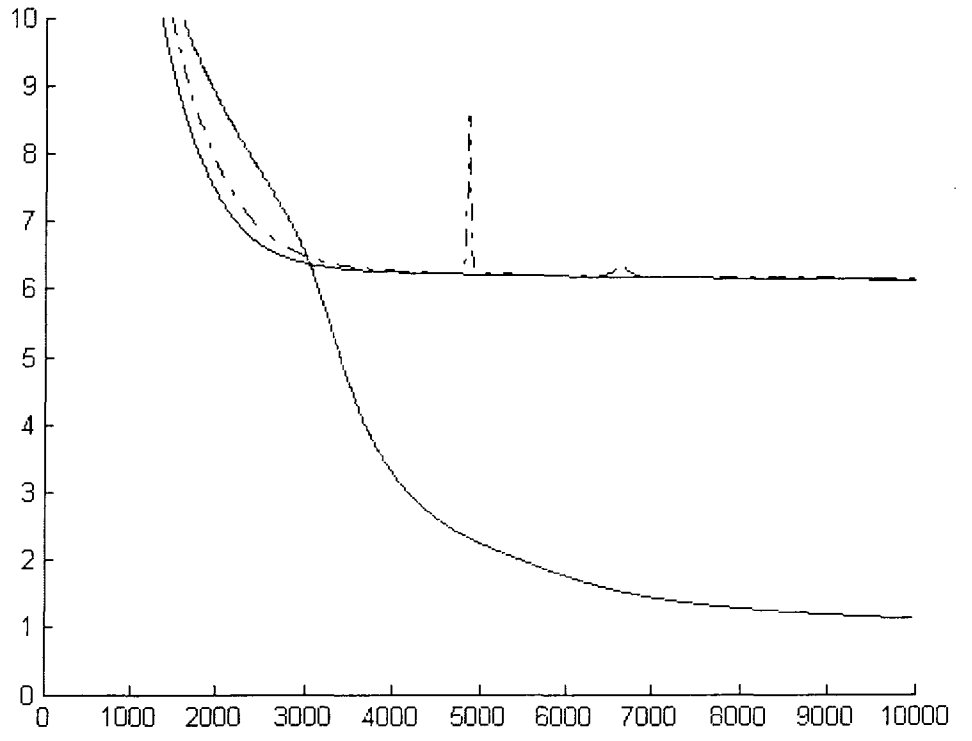


图 4-10 训练次数为 10000 时，各算法的收敛图

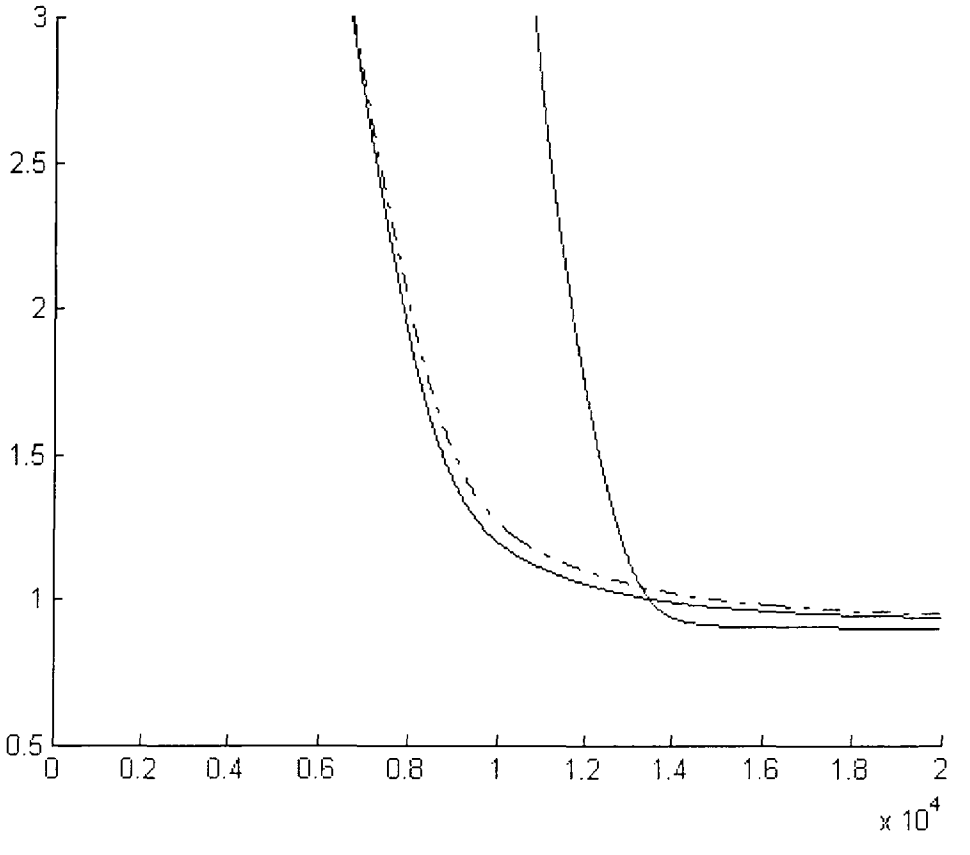


图 4-11 训练次数为 20000 时，各算法的收敛图

中国知网 https://www.cnki.net

4.5 小结

本章提出的改进网络权值算法，将三项均衡因子的思想和外推思想结合在一起，应用于 BP 算法的改进，能够有效的提高前向网络的收敛速率。仿真表明：改进算法能以较小的迭代步数取得了和 BP 算法的精度相当甚至高于 BP 算法的精度，而且在多次运行中的性能也很稳定。

中国知网 https://www.cnki.net

第五章 总结与展望

多层前馈神经网络是一种单向传播的多层前馈网络模型，由于具有高度的非线性映射能力，是目前神经网络研究与应用中最基本的网络模型之一，广泛应用于模式识别、图像处理、函数逼近、优化计算、最优预测和自适应控制等领域。

多层感知器以 BP 算法为标准算法。BP 算法的收敛速度慢是个固有的缺点，因为它是建立在基于只具有局部搜索能力的梯度法之上的。而只具有局部搜索能力的方法，若用于多个极小点的目标函数时，是无法避免陷入局部极小和速度慢的缺点的。因此，对 BP 算法的研究一直以来都是非常重要的课题。它是神经网络三大研究内容（包括神经网络理论研究、神经网络实现技术研究和神经网络应用研究）中的最基础的一项内容。

随着控制学科领域的理论发展和应用的不断深入，神经网络的应用已经越来越广泛，多层感知器作为一种非常重要的网络，广泛地用于非线性不可分模式的分类问题和非线性函数逼近问题。但是作为多层感知器标准算法的 BP 算法，本质上是一种基于梯度下降的搜索算法，存在收敛速度慢、网络容错能力（tolerant capacity）差和算法不完备（易陷入局部极小），而且对于较大搜索空间，多峰值和不可微函数也常不能搜索到全局极小点等缺陷，因此有必要对 BP 算法进行改进。

通过结合权值外推的思想和三项均衡因子的思想提出了一种新的 BP 算法，计算机仿真表明，此改进算法不仅有效的加快了网络收敛的速度，而且提高了逃离局部极小点的能力。

BP 算法是神经网络中非常重要的算法，是多层感知器的标准算法。目前对 BP 算法的改进方法中大多数还只是在激活函数的修改，误差的重新定义，如何动态调整学习率等方面进行。在利用心理学方面的知识（如遗忘系数[26]）进行改进的方法还不是太多，能否心理科学、神经学等学科来进一步提高 TWEBP 算法的性能是值得深入研究和探讨的课题。

中国知网 https://www.cnki.net

参考文献

[1] Simon Haykin 著. 叶世伟 史忠植译. 神经网络原理[M]. 北京: 机械工业出版社.2004

[2] 阎平凡,张长水. 人工神经网络与模拟进化计算[M]. 北京:清华大学出版社. 2000

[3] 飞思科技产品研发中心. MATLAB6.5 辅助神经网络分析与设计[M]. 北京:电子工业出版社.2003

[4] 周志华,曹存根. 神经网络及其应用[M]. 北京:清华大学出版社. 2004

[5] S.V.Kamarthi, S.Pittner, Accelerating neural network training weight extrapolations [J], Neural Networks. 12 (1999):1285-1299

[6] Y.H.Zweiri, J.F.Whidborne, L.D.Seneviratne, A three-term backpropagation algorithm, Neurocomputing. 50 (2003):305-318

[7] Y.H.Zweiri, L.D.Seneviratne, Kaspar Alothefer, Stability analysis of a three-term back propagation algorithm [J], Neural Networks. 18(2005):1341-1347

[8] Fahlman, S E. (1989). Fast learning variations on back-propagation: an empirical study [A].Proceedings of the1988 Connectionist Models Summer School[C]. 38-51 Pittsburgh: ,PA:Morgan Kaufmann

[9] CHEN J R, MARS P. Stepwise variation methods for accelerating the back – propagation algorithm[A]. Proceedings of the International Joint Conference on Neural Networks[C]. 1990. 601 - 604.

[10] VITELA J E, REIFMAN J. Premature saturation in back - propagation networks: Mechanism and necessary conditions [J]. Neural Networks. 1997. 10 (4):721 - 735.

[11] NG Sin-chun, CHEUNG Chi-chung,LEUNG Shu-hung. A new adaptive learning algorithm using magnified gradient function [A]. Proceedings of the International Joint Conference on Neural Networks [C]. 2001. 156 - 159.

[12] 李军,丁萃菁. 一种改进的 BP 算法在实际应用的研究[J].计算机仿真.2004.21(2):119-121

[13] 朱武亭.BP 网络应用中的问题及解决[J].上海海事大学学报.2005.26(2):64-66

中国知网 https://www.cnki.net

[14] 邓娟，杨家明.一种改进的 BP 算法神经网络[J].东华大学学报（自然科学版）.2005.31(3):123-126

[15] 武研，王守觉.一种新的快速收敛的反向传播算法[J].同济大学学报（自然科学版）.2004.32(8):1092-1095

[16] 李东侠,李平,丁淑艳.基于改进的 BP 网络误差修正的广义预测控制[J].计算机仿真.2004,21(12):143-145

[17] 曹均阔,舒远仲,叶永生.一种结构自组织的改进 BP 网络[J].计算机工程.2005.31(17):165-167

[18] Martin T.Hagan 著.戴葵 译.神经网络设计[M].北京:机械工业出版社.2002

[19] 杨东侯,年晓红,杨胜跃.两种改进的 BP 神经网络学习算法[J].长沙大学学报.2004.18(4):54-57

[20] 李宏刚,吕辉,李刚.一种 BP 神经网络的改进方法及其应用［J］.中国工程科学.2005,7(5):63－65

[21] 苏小红,王亚东,马培军,基于反馈调控参数的 BP 学习算法研究[J].哈尔滨工业大学学报,2005.37(10):1311-1314

[22] 赵亚丽,西广成,易建强.基于相对熵函数准则 BP 算法[J].计算机工程.2005.31(19):12-14

[23] 武研,王守觉.一种通过反馈提高神经网络学习性能的新算法[J].计算机研究与发展.2004.41(9):1488-1492

[24] 张可林,赵英良.数值计算的算法与分析[M].北京:科学出版社.2003

[26] 魏海坤.神经网络结构设计的理论与方法[M].北京:国防工业出版社.2005

[25] 张志涌,徐彦琴.MATLAB 教程[M].北京:北京航空航天大学出版社.2001

[26] 叶强,卢涛,李一军. 遗忘神经网络模型及其 BP 算法[J]. 计算机工程. 2003. 29(20): 135-136

中国知网 https://www.cnki.net

攻读学位期间公开发表的论文

1、王之仓，邓伟．多层感知器学习算法研究．青海师范大学学报．已录用．

2、王之仓．从 Linux 源代码看程序设计技巧．青海师范大学学报．2005.83(4):56-59

中国知网 https://www.cnki.net

致 谢

本论文是在导师邓伟博士的亲切关怀与悉心指导下完成的，邓老师知识渊博、治学严谨、教学认真，对我在学习、工作和生活各个方面均关怀备至。邓老师对学问孜孜不倦的态度及为人处事的原则对我今后的学习和工作都将产生重要的影响，在此，特向邓老师致以我崇高的敬意与衷心的感谢。

衷心感谢计算机科学与技术学院各位老师和同学们对我的关心、帮助和鼓励，尤其感谢苏美娟同学和聂永同学对我的帮助。

中国知网 https://www.cnki.net

高校教师申请硕士学位论文摘要

（2006 届）

多层感知器学习算法研究

Research on Multilayer Perceptron Learning Algorithm

研究生姓名 王之仓

指导教师姓名 邓伟（副教授）

专业名称 计算机应用技术

研究方向 神经网络

论文提交日期 **2006 年 10 月**

中国知网 https://www.cnki.net

多层感知器学习算法研究

详细摘要

多层感知器是一种单向传播的多层前馈网络模型，由于具有高度的非线性映射能力，是目前神经网络研究与应用中最基本的网络模型之一，广泛应用于模式识别、图像处理、函数逼近、优化计算、最优预测和自适应控制等领域。而反向传播算法是多层感知器的标准算法，通常作为其他学习算法的基准。反向传播算法的收敛速度慢是个固有的缺点，因为它是建立在基于只具有局部搜索能力的梯度法之上的。而只具有局部搜索能力的方法，若用于多个极小点的目标函数时，是无法避免陷入局部极小和速度慢的缺点的。因此，对反向传播算法算法的研究一直以来都是非常重要的课题。

为了提高反向传播算法的收敛率，本文提出了一种新的改进型的 BP 算法---TWEBP 算法。由于 BPWE（权值外推 BP）算法和 TBP（三项 BP）算法都是基于权值调整的改进算法，而考虑将 TBP 算法中的均衡因子融入到 BPWE 算法中，从而使后者对权值的调整由原来的两项增加为三项，从而提出 TWEBP 算法。为了验证本算法的优点，采用 XOR 问题、三分类问题和函数逼近三个问题进行了实验，发现收敛速度和逃离局部极小点的能力都有了一定程度的提高。

本文首先介绍了神经网络的产生背景、发展历史，多层感知器的学习算法存在的问题、研究成果、改进的策略，并简单描述了本文采用的改进反向传播算法的策略。接着分析和讨论反向传播算法的原理和对其进行改进的方法。然后，对性能优化理论和现有的比较优秀的几种 BP 优

中国知网 https://www.cnki.net

化算法进行了分析和讨论。

接着，对趋势外推思想、趋势外推算法和均衡因子的思想进行了分析和讨论，在此基础上提出了新的 BP 学习算法---TWEBP 算法。

最后，采用 XOR 问题、三分类问题和函数逼近三个问题对提出的算法进行了仿真，和其他算法进行了比较。

本文提出的 TWEBP 算法是一种优秀的算法，可以在方法应用于实际问题的解决。但是还可以深入研究，包括考虑采用可变学习率，将心理科学、生物神经学等的技术融入到该算法中，以进一步提高它的收敛速度。

关键词：多层感知器；学习算法研究；趋势外推；均衡因子；TWEBP

作　　者：王之仓
指导教师：邓　伟

中国知网 https://www.cnki.net