

# BP 算法改进及在软件成本估算中的应用

周启超

(渤海大学 信息科学与技术学院, 辽宁 锦州 121013)

**摘要:** 软件成本估算在软件开发过程中扮演着重要角色,它是控制软件进度、降低软件风险和保证软件质量的有效措施。软件成本估算是复杂的工作,受众多因素影响且具有不确定性。针对软件成本难以估算的现状,文中基于 BP 算法展开研究。首先,研究 BP 算法原理和数学表示;然后,在分析 BP 算法存在问题的基础上,将自适应学习率、附加动量项法与退火模拟算法、遗传算法等方法相结合对其进行改进;最后,构建软件成本估算指标体系,并将改进的 BP 算法应用于软件成本估算。结果表明,算法具有估算结果精确、节省人力物力、减少资金浪费等优点,对加快软件开发进度、提高软件质量具有重要作用。

**关键词:** BP 算法;改进;成本估算;应用

**中图分类号:** TP39

**文献标识码:** A

**文章编号:** 1673-629X(2016)02-0195-04

**doi:** 10.3969/j.issn.1673-629X.2016.02.043

## Improvement of BP Algorithms and Its Application in Software Cost Estimation

ZHOU Qi-chao

(College of Information Science and Technology, Bohai University,  
Jinzhou 121013, China)

**Abstract:** Software cost estimation plays an important role in the software development process, which is an effective measure to control software progress and reduce software risk and ensure software quality. Software cost estimation is a complicated task, affected by many factors and uncertainties. According to the current situation of software costs are difficult to estimate, a study based on BP algorithm is carried out. Firstly, BP algorithm principle and mathematical representation is researched. Then, based on the analysis of existing problems on BP algorithm, the adaptive learning rate, the additional momentum method and simulated annealing, and genetic algorithm are combined to improve it. Finally, the software cost estimation index system is established and the improved BP algorithm is used in software cost estimation. The results show that the improved BP algorithm has the advantages of accurate estimated result, saving manpower and material resources, reducing waste of money, to accelerate software development progress and improve software quality with an important role.

**Key words:** BP algorithm; improvement; cost estimation; application

## 0 引言

软件系统规模不断增大且越来越复杂化,随之带来了软件项目延期和预算超支等问题,准确估算软件成本是控制项目进度、减少项目成本的重要手段。软件成本估算主要指软件开发过程中所花费的工作量及相应的代价<sup>[1]</sup>,影响软件成本的因素不仅多且异常复杂,而且随着项目的不断开发,影响因素也在不断变化,存在复杂的非线性关系,要准确高效地进行软件成本估算是非常困难的。因为 BP 神经网络具有较强的非线性映射能力,能以任意精度逼近非线性函数,其高

度的自我学习能力和自适应能力,能够快速地学习和提取知识,将其应用于软件成本估算中,不必费力去寻找和发现成本驱动因素与软件成本之间的关系<sup>[2-3]</sup>。

BP 神经网络在前馈神经网络中应用最广泛,是一种 3 层或者 3 层以上的神经网络,包括一个输入层,一个或多个隐含层,一个输出层,相邻两层之间的所有神经元互相连接,而处于同一层的神经元不能相互连接<sup>[4]</sup>。采用的 BP 算法具有计算量小、并行性强、可操作性强等优点。随着 BP 算法的不断应用,一些不足逐渐被发现。因此,文中研究 BP 算法的原理、存在的

收稿日期:2015-04-02

修回日期:2015-07-08

网络出版时间:2016-01-04

基金项目:2014 年辽宁省教育科学研究项目(L2014248)

作者简介:周启超(1990-),女,硕士,研究方向为人工智能。

网络出版地址: <http://www.cnki.net/kcms/detail/61.1450.TP.20160104.1453.006.html>

问题及改进方法,并应用到复杂的软件成本估算中。

## 1 BP 算法原理

### 1.1 原理

BP 算法也称误差反向传播算法,实质是一边传播一边不断调整权值的过程,使实际输出与期望输出的误差达到最小。主要思想是输入信号从输入层经过隐含层处理,到达输出层,输出端产生输出信号,在信号向前传递过程中网络的权值是固定不变的,每一层神经元的状态只影响下一层神经元的状态<sup>[5]</sup>,这是工作信号的正向传播。当实际输出没有达到期望输出,将误差信号由输出端开始逐层向前传播,按照原来神经元连接的路线返回,这是信号的反向传播。在反向传播过程中,网络的实际输出与期望输出之间的差值即为误差信号,将误差分摊给各层所有的单元,根据误差反馈修改各层权值,经过多次的迭代,使实际输出越来越逼近期望值或达到预先设定的学习次数,停止训练。

### 1.2 BP 算法数学表达式

三层 BP 神经网络是 BP 神经网络中最典型的,所以介绍如下<sup>[6]</sup>:如输入层、隐含层和输出层的节点个数分别是  $m, n, s$ , 输入样本总数  $P$ 。把阈值写入连接权中,即把阈值看成节点的一个输入的权值,则各节点的输出可以表示为向量的形式  $v_{ji}x_{pi}$ , 隐含层、输出层节点的输出分别为:

$$d_{pj} = f\left(\sum_{i=0}^m v_{ji}x_{pi}\right), j = 0, 1, \dots, n \quad (1)$$

$$o_{pk} = f\left(\sum_{j=0}^n w_{kj}d_{pj}\right), k = 0, 1, \dots, s \quad (2)$$

其中,  $x_{pi}$  表示第  $p$  个样本的第  $i$  个输入值;  $v_{ji}$  表示输入层第  $i$  个节点到隐含层第  $j$  个节点的权值;  $w_{kj}$  表示隐含层第  $j$  个节点到输出层第  $k$  个节点的权值。

在式(1)、(2)中激励函数选取 Sigmoid 函数:

$$f(x) = \frac{1}{1 + e^{-x}} \quad (3)$$

$f(x)$  具有连续可导的特点,且有:

$$f'(x) = f(x)(1 - f(x)) \quad (4)$$

全局误差函数为:

$$E = \sum_{p=1}^P E_p \quad (5)$$

其中,  $E_p$  是第  $p$  个样本的误差;  $t_{pk}$  是第  $p$  个样本第  $k$  个节点的期望输出。

$$E_p = \frac{1}{2} \sum_{k=1}^s (t_{pk} - o_{pk})^2 \quad (6)$$

在反向传播中,采用梯度下降方法,其各层调整公式如下所示。其中  $\eta$  是学习率,一般取值(0,1)。

输出层各神经元权值调整公式为:

$$\Delta w_{kj} = -\eta \frac{\partial E}{\partial w_{kj}} = \eta \sum_{p=1}^P \left(-\frac{\partial E_p}{\partial w_{kj}}\right) = \eta \sum_{p=1}^P \delta_{pk} d_{pj} \quad (7)$$

其中:

$$\delta_{pk} = (t_{pk} - o_{pk}) o_{pk} (1 - o_{pk}) \quad (8)$$

隐层各神经元权值调整公式为:

$$\Delta v_{ji} = -\eta \frac{\partial E}{\partial v_{ji}} = \eta \sum_{p=1}^P \left(-\frac{\partial E_p}{\partial v_{ji}}\right) = \eta \sum_{p=1}^P \delta_{pj} x_{pi} \quad (9)$$

其中:

$$\delta_{pj} = \left(\sum_{k=1}^s \delta_{pk} w_{kj}\right) d_{pj} (1 - d_{pj}) \quad (10)$$

根据以上公式进行网络训练,BP 算法的实现步骤<sup>[7-8]</sup>如图 1 所示。

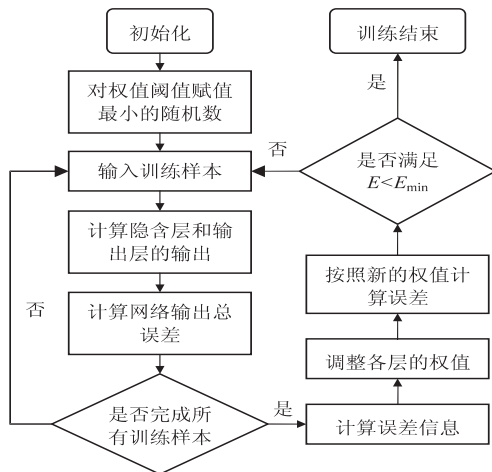


图 1 BP 算法流程图

## 2 BP 算法存在的问题及改进

### 2.1 BP 算法存在的问题

BP 算法因为固定的学习率,需要长时间的调整和多次的迭代,因此训练次数多,收敛速度慢,训练时间较长,对于一些复杂问题,所需的训练时间会更长,这是由于学习率太小所致,如果加快收敛速度,则容易产生振荡;由于网络误差曲面是非常繁杂的高维的凹凸不平的曲面,BP 算法收敛到某个值时不能保证误差是全局最小值,也可能是局部极小值,不同的起始点可能导致不同的极小值,而无法得到全局最优解<sup>[9]</sup>。由于在一些平坦地区内误差的改变很小几乎为零,造成网络不能得到训练,收敛速度就会变慢,难以达到给定的误差。还有网络结构的选择,包括输入层和输出层节点数、网络隐层数、隐层神经元数,只能根据经验或者通过反复实验去确定。网络的学习和记忆具有不稳定性,容易造成过拟合现象导致泛化能力差等缺点。

### 2.2 改进 BP 算法的方法

目前随着对 BP 算法的大量研究,已经得出了很多改进的方法,在前人基础上进行再改进,将几种改进后与其他智能算法相结合,如下所示。

自适应调节学习率的方法是为了减少训练时间,实现快速有效的学习收敛的过程。基本思想:根据误差增量的大小来调节学习率的大小,当误差减小,增大学习率;当误差增大,减小学习率<sup>[10]</sup>。对此进行改进,修正公式为:

$$\eta(t+1) = (1 - \alpha_1 \cdot \Delta E + \alpha_2 \cdot \frac{E_t}{E_{t+1}}) \eta(t) \quad (11)$$

其中,  $\eta(t+1)$ 、 $\eta(t)$  分别表示学习率迭代到第  $t$ 、 $t+1$  次时的值;  $\Delta E = E_{t+1} - E_t$ ;  $0 < \alpha_1 < \alpha_2 < 1$  为常量。

此公式不仅考虑了误差增量,还考虑了相对误差  $E_{t+1}/E_t$ , 根据它适当减小学习率的调整幅度,进而有更合理的学习率进行训练,避免出现振荡。优点是算法稳定性得到保证,又使算法收敛速度加快。

附加动量项法,BP算法中  $\eta$  的选取很重要,  $\eta$  值大网络收敛速度快,但过大会引起不稳定;若  $\eta$  值小虽避免了不稳定,收敛速度变慢,加入动量项来改变  $\eta$  从而提高网络性能。动量项能记忆上一时刻的权值修改方向<sup>[11]</sup>,使权值调节向着误差曲面底部的平均方向变化,不仅可以微调权值的修正量,也避免陷入局部极小值,加速了网络的收敛,避免了网络的来回振荡,起到了缓冲平滑的作用。其中动量项  $\beta \Delta w(t) = \beta[w(t) - w(t-1)]$ ,  $\beta$  为动量因子,  $0 < \beta < 1$ ,一般取  $0.3 \sim 0.6$ 。

$$\Delta w(t+1) = \eta \frac{\partial E}{\partial w} + \beta \Delta w(t) \quad (12)$$

模拟退火算法是通过模拟金属热加工中温度与能量的关系,温度高原子的能量高,随着温度降低,能量越来越小而达到最低点确定关系。概率为  $P(E) \propto \exp(-E/kT)$ , 定义人工温度  $T$ , 需要加入一个干扰  $\Delta w$ , 则  $\Delta E = E(w + \Delta w) - E(w)$ ,  $\Delta E < 0$ , 干扰接受,  $\Delta E > 0$ , 则依据概率公式判断是否接受, 干扰接受  $w = w + \Delta w$ , 否则不变。

文中将自适应学习率法和附加动量项法与模拟退火算法、遗传算法相结合。基本思想是:在BP算法进行样本数据训练时,首先利用遗传算法产生初始种群,确定网络的权值,根据式(1)、(2)计算节点的输出,式(5)计算全局误差,得到测试误差,然后判断训练是否达到平衡状态,是则调用自适应学习率调整算法和附加动量项法,根据权值修正公式(7)、(9),式(11)和式(12)的调整学习率和附加动量项,来进行权值调整。若  $E_{t+1} < E_t$ , 说明第  $t$  时刻的迭代是比先前有所提高,可以接受的,应该增大学习率;若  $E_{t+1} > E_t$ , 则说明  $t$  时刻迭代是比先前的误差更大了,没有效果,应该减小学习率。否则调用模拟退火算法对权值进行加干扰等进行调节,致使网络训练进入到新的状态时<sup>[12]</sup>,再利用遗传算法的选择、交叉、变异等遗传操作得到新群体,计算群体适应度、计算精度,如果满足条件则训练

结束,否则返回前面的步骤依次执行,对新状态进行优化改进,反复进行计算,直到满足算法的要求为止。

### 3 改进 BP 算法的应用

将以上改进 BP 算法和智能算法相结合的方法,应用到软件项目的成本估算中,以提高软件成本估算的效率和准确性。

#### 3.1 构建指标体系

软件成本估算通常是通过估算工作量来进行,综合得到,一级指标是产品元素、计算机元素、人员因素、项目因素;相对应的16个二级指标<sup>[13-15]</sup>如图2所示。

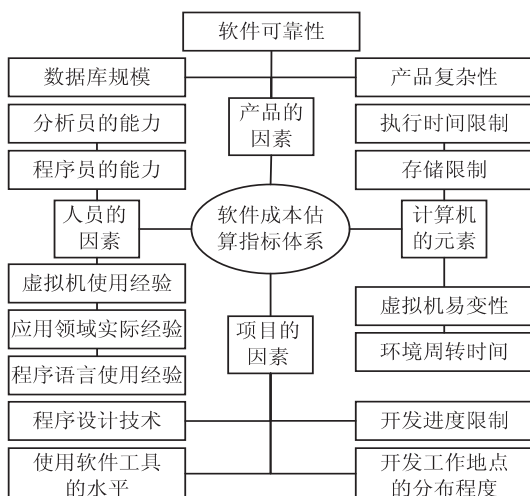


图2 软件成本估算指标体系

实现步骤如下:

(1) 对输入训练样本进行归一化,公式如下:

$$\bar{X}_p = (X_p - X_{\min}) \div (X_{\max} - X_{\min}) \quad (13)$$

(2) 定义网络输入和期望输出。

(3) 设BP神经网络中输入层节点数为17个,前16个为成本驱动因子,第17个节点取KDSI代码千行数,只需要一个输出节点,根据前人的研究理论,隐层节点为  $2 \times 17 + 1 = 35$  个。建立相应BP网络,根据改进算法进行。

(4) 设  $x_1, x_2, \dots, x_{16}$  为成本驱动因子的评分;输入向量为  $X = [x_1, x_2, \dots, x_{17}]^T$ , 根据上述改进算法进行训练。

#### 3.2 仿真结果比较

用MATLAB仿真改进前后的算法比较输出结果。基于COCOMO数据库<sup>[16]</sup>的80个项目的训练样本得到的结果如图3和图4所示。

结果表明在使用同一组数据时,改进后的训练步数减少很多,大约是改进前的一半,且随着训练步数增加,误差更趋于稳定收敛,没有很多跳跃的点,减少陷入局部极小值的问题,大大减少了估算时间。同样在比较实际输出与期望输出结果,改进前虽然基本上和

实际输出大体一致,但存在很多相差很大的部分,而改进后的基本与实际输出一致。准确率也相应提高,误差率也达到了实际应用的要求。

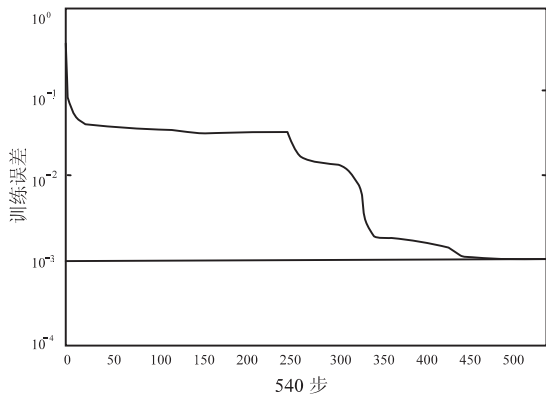


图 3 改进前训练误差

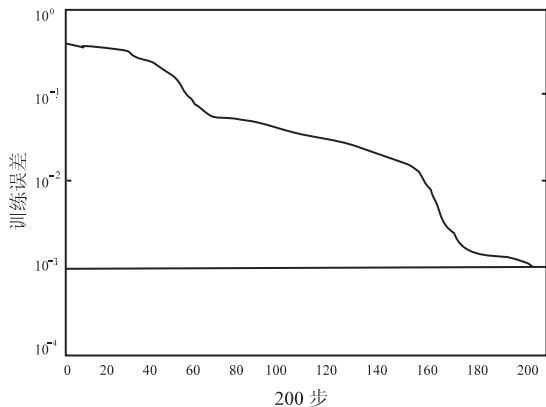


图 4 改进后训练误差

因此,结果更加表明有效的改进更适用于实际估算,它不仅兼具了标准 BP 算法的优点,还带有自适应学习率、附加动量项,模拟退火算法和遗传算法的优点。使软件成本估算时间短且相对简单,并且带有准确性<sup>[17-18]</sup>。

## 4 结束语

随着对 BP 神经网络理论不断深入的研究,以及网络计算能力不断的提高,BP 算法更适应于很多复杂难以解决的问题,对其不断改进将会使应用范围不断扩大。为了改善 BP 神经网络的各种不足,前人已经在各个方面采用不同的方法,有了大量的研究及改进。

文中已有研究成果的基础上,将其改进并和遗传算法、模拟退火算法等智能算法相结合,算法各自发挥自己的优势,具有全局优化搜索性质、容易得到全局最优解等特点,又根据影响软件成本估算的因素众多且不具有确定性等特点,将算法应用到软件成本估

算中。结果表明不仅减少了估算的时间,估算的精确度也得到了很大提高。

## 参考文献:

- [1] 任永昌. 软件项目管理[M]. 北京:清华大学出版社,2012.
- [2] 张红. 基于神经网络的软件项目工作量估算系统实现[J]. 宁波职业技术学院学报,2014,18(4):87-90.
- [3] Mittas N, Andreas S. Integrating non-parametric models with linear components for producing software cost estimations[J]. Journal of Systems and Software,2015,99(1):120-134.
- [4] 王盛源,王颖,刘连光,等. 基于误差反向传播 ANN 的电网磁暴灾害风险评估方法[J]. 计算机与数字工程,2014,42(12):2223-2226.
- [5] 孟栋,樊重俊,王家桢. 混沌遗传算法对 BP 神经网络的改进研究[J]. 数学理论与应用,2014,34(1):102-110.
- [6] 蒋亮. BP 神经网络的优化研究[D]. 南昌:南昌大学,2014.
- [7] 王晶晶,王剑. 一种 BP 神经网络改进算法研究[J]. 软件导刊,2015,14(3):52-53.
- [8] 李晓华,周加波,周韦. 基于误差反向传播神经网络的江苏省用电量预测研究[J]. 科技与企业,2014,23(20):197-198.
- [9] 白牧可,唐巍,张璐,等. 基于 BP 神经网络群的中压配电网电压降落估算[J]. 电力系统保护与控制,2014,42(2):132-138.
- [10] 苏丽冰. BP 神经网络的改进综述[J]. 信息与电脑:理论版,2013,7(6):110-111.
- [11] 江丽. 基于粒子群与模拟退火算法的 BP 网络学习方法研究[D]. 合肥:安徽大学,2013.
- [12] 孙海涛,李仲秋. 模拟退火算法改进 BP 算法的区域物流中心选址[J]. 计算机技术与发展,2014,24(9):222-225.
- [13] Boehm B W. Software cost estimation with COCOMO II[M]. [s.l.]:Prentice Hall PTR,2009.
- [14] Seo Yeong-Seok, Jeffery R. Software effort estimation based on multiple regressions with adaptive recursive data partitioning[J]. Information and Software Technology, 2013, 55(10):1710-1725.
- [15] Nassif A B, Ho D. Towards an early software estimation using log-linear regression and a multilayer perceptron model[J]. Journal of Systems and Software,2013,86(1):144-160.
- [16] 焦玉婷. 软件成本估算模型的研究[D]. 北京:北京工业大学,2013.
- [17] 段美美,于本海,朱萌. 基于 CBR 的软件项目成本估算方法[J]. 计算机工程与设计,2014,35(11):3837-3844.
- [18] 郝胜兰. 基于模糊神经网络的房产软件项目成本估算研究[D]. 大连:大连海事大学,2012.

BP算法改进及在软件成本估算中的应用

作者：[周启超](#)，[ZHOU Qi-chao](#)  
作者单位：[渤海大学 信息科学与技术学院, 辽宁 锦州, 121013](#)  
刊名：[计算机技术与发展](#)[ISTIC](#)  
英文刊名：  
年，卷(期)：2016 (2)

引用本文格式：[周启超](#), [ZHOU Qi-chao](#) [BP算法改进及在软件成本估算中的应用](#) [期刊论文]-[计算机技术与发展](#)  
2016 (2)