

DOI: 10.13718/j.cnki.xdzk.2016.03.026

# 一种改进的 BP 神经网络剪枝算法研究<sup>①</sup>

张 虹, 王 丹

西南大学 计算机与信息科学学院, 重庆 400715

**摘要:** 将相关性剪枝算法(CPA)和变学习率、附加动量方法结合提出了一种基于 CPA 的改进的 BP 神经网络剪枝算法. 实验结果表明, 改进的算法可以降低训练步数, 加快神经网络的收敛速度, 在测试数据集上的均方误差也得到了进一步的优化.

**关键词:** BP 神经网络; 相关性剪枝算法; 变学习率; 附加动量

**中图分类号:** TP391

**文献标志码:** A

**文章编号:** 1673-9868(2016)03-0165-06

BP(Back Propagation)网络是 1986 年以 Rumelhart 和 McClelland 为首的科学家小组提出的一种人工神经网络<sup>[1-2]</sup>, 它是一种按误差逆传播算法训练的多层前馈网络, 是目前应用最广泛的神经网络模型之一<sup>[3-6]</sup>. BP 神经网络的性能主要取决于网络的性能, 尤其是隐含层节点数. 长期以来, 隐含层节点数的确定尚缺乏严格的理论指导<sup>[7]</sup>, 选择合适的隐含层节点数是值得研究的问题<sup>[8]</sup>. 此外, BP 神经网络还存在收敛速度慢、收敛过程存在震荡、容易陷入局部最优等问题. 许多学者根据实际需要提出了改进的方法, 在算法的理论研究和实际应用方面都取得了丰硕的成果<sup>[9-14]</sup>.

本文提出了一种基于相关性剪枝算法(CPA)的改进的 BP 神经网络剪枝算法. 其基本思想是: 当隐含层两个节点的输出对所有样本具有较大的相关性时, 将它们合并成一个节点, 从而去掉隐含层中冗余节点. 在调整隐含层节点的同时, 使用变学习率的方法提高神经网络的收敛速度, 采用附加动量来减少神经网络收敛过程中的震荡. 实验结果表明, 本文提出的方法能够进一步提高神经网络的性能.

## 1 变学习率和附加动量方法

通常, 我们可以采用梯度下降的方法求解 BP 神经网络, 但是, 这种方法容易导致陷入局部最优、训练过程震荡、收敛速度慢等问题. BP 神经网络可以通过附加动量和变学习率进行改进.

### 1.1 附加动量方法

附加动量方法的基本思想是: 为原始的 BP 神经网络的权值修正量增加动量因素, 使得权值变化量的幅度变小, 从而减小神经网络的震荡.

**定义 1** 设  $\Delta w_{ij}(t)$  为当前周期的权值变化量,  $\Delta w_{ij}(t-1)$  为上一周期的权值变化量,  $\eta$  为动量因子,  $l$  为学习率,  $E_j$  为下一层节点的误差,  $O_i$  为上一层节点的输出. 则附加动量方法的基本思想是按照如下公式修改神经网络的权值:

$$\Delta w_{ij}(t) = \eta \times \Delta w_{ij}(t-1) + l \times E_j \times O_i \quad (1)$$

① 收稿日期: 2015-04-01

基金项目: 国家自然科学基金项目(71201129).

作者简介: 张 虹(1975-), 女, 四川绵阳人, 副教授, 主要从事智能算法与数据挖掘研究.

一般而言,  $\eta$  可以设置为 0~1 之间的值, 通过增加动量因子, 可以使得权值变化的幅度减小, 减少 BP 神经网络的震荡.

### 1.2 变学习率方法

变学习率方法的基本思想是, 如果当前神经网络的误差减小, 则考虑增大学习率, 加快收敛的过程, 如果神经网络的误差变大, 则减小学习率, 减小神经网络误差的震荡.

可以采用如下的公式对神经网络的学习率进行调整:

$$l(t) = \begin{cases} 1.05l(t), & RMSE(t) < RMSE(t-1) \\ 0.7l(t), & RMSE(t) > RMSE(t-1) \\ l(t), & RMSE(t) = RMSE(t-1) \end{cases} \quad (2)$$

### 1.3 相关性剪枝算法

相关性剪枝方法(correlation pruning algorithm, CPA)是一种较为简单有效的剪枝方法, 其基本思想是: 计算隐含层节点之间的相关性, 然后基于相关性对隐含层节点进行删除(剪枝). CPA 方法相比其他方法而言(如灵敏度计算方法、权衰减方法等), 计算复杂度更低, 原理更为简单. 但是, CPA 方法可能由于剪枝而导致神经网络陷入局部最优.

确定隐含层神经元个数可以采用逐渐增加的方法. 即刚开始只选择一个神经元, 依次增加神经元个数, 并不断训练, 获取每次的训练误差. 随着神经元个数的增加, 误差会不断减少. 一般而言, 三层神经网络基本可以满足要求了. 如果三层神经网络无法满足要求的时候, 可以进一步尝试增加隐含层神经元层数, 从而达到期望的效果.

**定义 2** 设  $x$  为隐含层神经元个数,  $m, n$  分别为输入层、输出层的神经元个数,  $a$  为常数, 其范围为  $(0, 1)$ , 则

$$x = \sqrt{m+n} + a \quad (3)$$

$$x = \sqrt{0.43mn + 0.12n^2 + 2.54m + 0.77n + 0.35 + 0.51} \quad (4)$$

以上两个公式仅仅只能确定初始的神经元个数, 要获得较好的神经网络, 需要在(3),(4)式确定的神经元个数的基础之上, 做修剪或者增长, 即减少节点或者增加节点.

## 2 基于变学习率和附加动量的神经网络相关性剪枝算法

本文基于变学习率、附加动量和 CPA 算法, 提出了一种新的神经网络剪枝算法——LMCPA 算法. 基本思想是: 在训练的过程中, 使用 CPA 算法对神经网络隐含层节点进行动态调整, 然后通过变学习率和附加动量的方法增加神经网络的收敛速度. LMCPA 算法的基本步骤如下:

Step 1: 初始化. 确定输入层、隐含层和输出层节点数; 初始化各个节点的阈值以及网络的权值; 确定初始学习率以及动量因子; 给定目标误差  $e$ ; 给定相关系数阈值  $h_1$  和  $h_2$ ; 给定最大学习周期数 MaxEcho; 给定隐含层节点剪枝周期 CutEcho.

Step 2: 使用 BP 神经网络的梯度下降方法学习网络权值和阈值. 定义第  $t$  个周期的误差为  $RMSE(t)$ , 第  $(t-1)$  周期的误差为  $RMSE(t-1)$ . 按照如下公式计算学习率变化:

$$l(t) = \begin{cases} 1.05l(t-1), & RMSE(t) < RMSE(t-1) \\ 0.95l(t-1), & RMSE(t) > RMSE(t-1) \\ l(t-1), & RMSE(t) = RMSE(t-1) \end{cases}$$

Step 3: 使用如下公式计算神经网络权值变化:

$$\Delta w_{ij}(t) = \eta \times \Delta w_{ij}(t-1) + l \times E_j \times O_i$$

其中:  $\Delta w_{ij}(t-1)$  表示上一周期的权值变化量,  $\Delta w_{ij}(t)$  表示当前周期的权值变化量,  $\eta$  为动量因子,  $l$  为学习率,  $E_j$  为下一层节点的误差,  $O_i$  为上一层节点的输出.

Step 4: 判断当前周期  $t \bmod \text{CutEcho}$  是否等于 0, 若等于 0, 则进入 Step 5 进行剪枝, 否则进入 Step 2.

Step 5: 计算所有隐含层节点之间的相关系数. 令训练数据样本总数为  $n$ , 令隐含层节点  $i$  的第  $q$  个样本的输出为  $O_{iq}$ , 则隐含层节点  $i$  的输出序列为  $O_i = (O_{i1}, O_{i2}, \dots, O_{in})$ . 这里按如下公式定义隐含层节点输出序列的标准差:

$$S_i^2 = \sum_{q=1}^n (O_{iq} - \overline{O_i})^2$$

其中  $\overline{O_i}$  表示节点  $i$  输出序列的平均值. 隐含层节点  $i$  和  $j$  之间的相关系数  $C_{ij}$  可以定义如下:

$$C_{ij} = \frac{\sum_{q=1}^n O_{iq} O_{jq} - n \overline{O_i} \overline{O_j}}{S_i S_j}$$

Step 6: 隐含层节点修剪. 如果隐含层节点  $i$  和  $j$  之间的相关系数  $C_{ij} > h_1$ , 同时, 隐含层节点  $i$  和  $j$  各自的标准差满足  $S_i^2 > h_2$  和  $S_j^2 > h_2$ , 则删除隐含层节点  $j$ . 对于隐含层各个节点到输出层节点的边的权值按如下公式调整:

$$w_{iu}^{(12)} = w_{iu}^{(12)} + a w_{iu}^{(12)}$$

其中  $w_{iu}^{(12)}$  表示隐含层节点  $i$  到输出层节点  $u$  之间的权值(上标(12)表示隐含层和输出层之间的权值), 隐含层为第1层, 输出层为第2层.  $a$  为调整系数, 计算公式如下:

$$a = \frac{\frac{1}{n} \sum_{q=1}^n O_{iq} O_{jq} - \overline{O_i} \overline{O_j}}{\frac{1}{n} \sum_{q=1}^n O_{iq}^2 - \overline{O_i}^2}$$

此外, 还需要按如下公式对输出层各节点的阈值进行调整:

$$\theta_j^{(2)} = \theta_j^{(2)} + b \theta_j^{(2)}$$

其中  $b$  为阈值调整系数, 其定义如下:

$$b = \overline{O_j} - a \overline{O_i}$$

其中:  $\overline{O_j}$  为要被删除的节点  $j$  的输出序列的平均值,  $\overline{O_i}$  为节点  $i$  的输出序列的平均值.

Step 7: 计算均方误差  $RMSE(t)$ , 如果均方误差小于给定的误差要求  $e$ , 则终止训练. 判断当前的迭代周期  $t$  是否达到最大迭代次数  $MaxEcho$ , 如果达到, 则停止训练. 若均方误差没有达到, 且迭代次数小于  $MaxEcho$ , 则转 Step 2.

## 3 实验及算法性能分析

### 3.1 数据集及参数设置

选取如下函数对本文提出的 LMCPA 神经网络进行验证:

$$z = 0.5 \times \sin(6x) \times e^{(-y)}$$

其中:  $-1 < x < 1$ ,  $-1 < y < 1$ . 已有研究证明, 该函数能够较好地检验神经网络的性能. 本文用该函数随机生成了 400 组样本, 其中, 300 组用于训练神经网络, 100 组用于检验.

此外, 本文使用均方根误差  $RMSE$  作为算法有效性的评价标准, 均方误差的定义为:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (O_i - T_i)^2}$$

其中:  $O_i$  为神经网络的实际输出值,  $T_i$  则是神经网络的期望输出值.

首先, 设置神经网络的权值和阈值随机初始化为  $[-1, 1]$  之间的随机值. 均方根误差要求为 0.03, 最大训练步数为 5 000. 通过多次实验, 对各种参数配置(如隐含层节点数、学习率、动量参数等)进行测试, 求取使得误差最小、迭代周期较为合理的参数. 最后, 对于各个比较算法最优的参数配置如表 1 所示.

表 1 参数配置

算法	隐含层节点数/个	学习率	动量参数	$h_1$	$h_2$	修剪周期/次
传统 BP 神经网络	15	0.50	—	—	—	—
CPA 相关性分析剪枝方法	16	0.71	—	0.70	0.70	50
复合 BP 神经网络	16	0.80	0.15	—	—	—
LMCPA 神经网络	20	0.80	0.14	0.70	0.71	50

### 3.2 不同 BP 神经网络的性能对比

在完成了参数配置之后,对 4 种算法在最优参数配置下的性能进行对比.实验结果如表 2 所示:

表 2 4 种 BP 神经网络算法的性能对比

算法	训练数据集 RMSE	训练步数/步	测试数据集 RMSE
传统 BP 神经网络	0.03	2 041	0.041 41
CPA 相关性分析剪枝方法	0.03	1 524	0.043 89
复合 BP 神经网络	0.03	1 153	0.038 83
LMCPA 神经网络	0.03	797	0.031 37

相比于前 3 种方法,本文提出的方法进一步提高了神经网络的性能.一方面,本文提出的方法降低了训练步数,加快了神经网络的收敛速度,另一方面在测试数据集上的均方误差也得到了进一步优化.

### 3.3 LMCPA 神经网络在不同参数下性能分析

由于 LMCPA 引入了更多的参数,因此,这里需要对各个参数对于 LMCPA 的影响进行讨论.本小节分别对隐含层节点数、剪枝周期、相关系数阈值  $h_1$  和  $h_2$ 、神经网络学习率以及附加动量因子对于 LMCPA 的影响进行讨论.

#### 3.3.1 隐含层节点数对于 LMCPA 的性能影响

我们设置隐含层节点为 2 到 30,对 LMCPA 的 RMSE 进行比较.表 3 给出了 LMCPA 神经网络在不同隐含层节点情况下的 RMSE.从表中可以看出,在隐含层节点较少时,神经网络处理信息能力不够强,因此神经网络的误差相对比较大.当隐含层节点设置较大时,神经网络中存在一些冗余信息,因此效果也不理想.当将隐含层节点设置为 20 时,LMCPA 的迭代次数最少,且能够达到最小的 RMSE.

表 3 LMCPA 神经网络在不同隐含层节点下的 RMSE

初始隐含层节点	最终隐含层节点	训练周期/次	RMSE	初始隐含层节点	最终隐含层节点	训练周期/次	RMSE
2	2	1 375	0.042 75	16	9	897	0.032 67
3	3	1 242	0.041 75	17	7	995	0.032 27
4	3	1 255	0.039 56	18	8	1 021	0.032 59
5	4	1 342	0.042 48	19	7	835	0.032 51
6	5	1 277	0.036 50	20	9	797	0.031 37
7	6	1 242	0.033 19	21	10	866	0.031 74
8	6	1 320	0.034 05	22	10	935	0.032 35
9	7	1 369	0.033 20	23	14	996	0.032 93
10	7	1 315	0.034 33	24	12	1 021	0.033 72
10	7	1 023	0.034 33	25	16	1 022	0.032 09
11	6	1 077	0.032 86	26	10	1 154	0.033 98
12	6	1 132	0.032 14	27	13	1 204	0.033 42
13	8	1 067	0.032 59	28	13	1 275	0.034 67
14	9	1 022	0.032 94	29	20	1 237	0.033 95
15	14	1 124	0.032 75	30	19	1 234	0.033 64

#### 3.3.2 不同剪枝周期对于 LMCPA 的性能影响

图 1 以图形的形式给出了 LMCPA 误差曲线图.从图 1 中可以看出,当 RMSE 较小时,迭代周期也较短,剪枝周期为 50 时,其误差最小,且迭代周期最短.

### 3.3.3 相关系数阈值 $h_1$ 和 $h_2$ 对 RSME 的影响

由于本文引入了 CPA 的相关阈值系数  $h_1$  和  $h_2$ , 因此这里需要对  $h_1$  和  $h_2$  在不同取值下的误差情况进行分析. 由于 LMCPA 的参数较多, 这里假定剪枝周期为 50, 学习率为 0.8, 动量因子为 0.14, 讨论  $h_1$  和  $h_2$  分别在  $[0.1, 1.0]$  下的 RMSE 情况, 并将 RMSE 的分布情况绘制成三维图(图 2).

从图 2 中可以看出: 当  $h_1$  和  $h_2$  取值过小时, 神经网络的误差较大; 当  $h_1$  的取值较小时,  $h_2$  在任何取值下 RMSE 均较大; 当  $h_1$  取值较大(如  $h_1 > 0.7$ )时, RMSE 随着  $h_2$  的增加而减小, 当  $h_2$  达到 0.7 时, RMSE 达到最小, 随着  $h_2$  的继续增大, RMSE 有一定程度的增大, 但仍然处于较低的水平.

### 3.3.4 不同的学习率和不同动量因子对 RMSE 的影响

为了分析不同学习率和不同附加动量对 RMSE 的影响, 我们首先对其他参数进行了设定: 剪枝周期设置为 50,  $h_1$  和  $h_2$  均设置为 0.7, 学习率的变化区间为  $[0.05, 1.0]$ , 附加动量的变化区间为  $[0.05, 0.4]$ .

图 3 展示了不同学习率和不同动量因子下, LMCPA 神经网络的 RMSE. 从图 3 中可以看出: 附加动量较小时, RMSE 非常大, 而随着附加动量的不断增加, 误差随着学习率的变化呈现波浪形波动, RMSE 在学习率取 0.8, 附加动量取 0.14 时误差达到最小值 0.028 58.

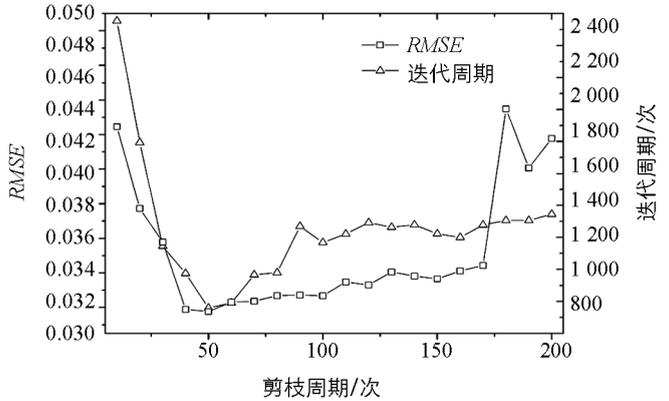


图 1 LMCPA 神经网络在不同剪枝周期下的 RMSE 情况

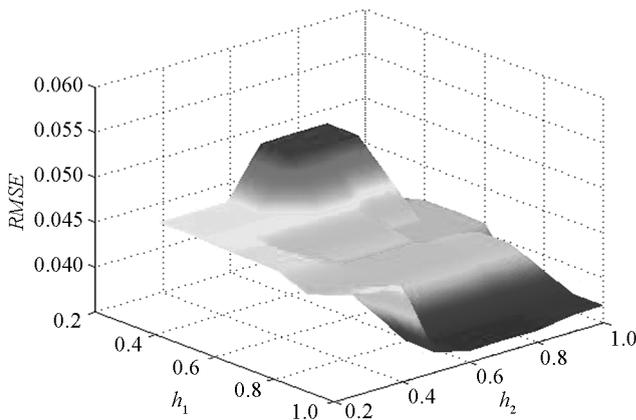


图 2 LMCPA 神经网络在不同  $h_1, h_2$  下的 RMSE 变化情况

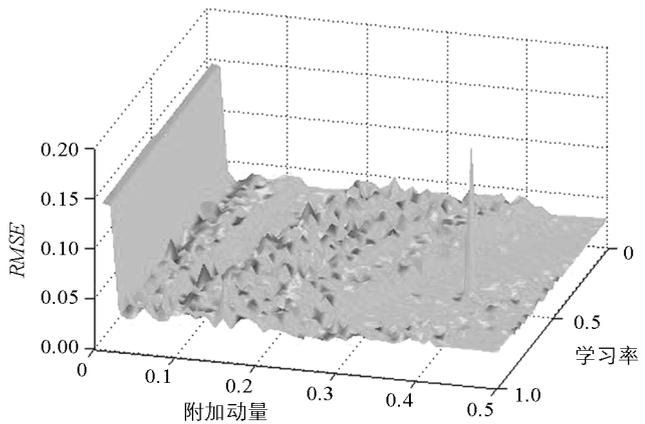


图 3 RMSE 随学习率和动量因子的变化图

## 4 结 论

本文基于变学习率和附加动量的方法, 结合相关性剪枝算法优化隐含层结构, 提高了神经网络的性能. 此外, 基于非线性函数产生的数据集, 将本文算法和其他 3 种不同的算法进行了对比, 实验结果表明: 本文提出的 LMCPA 算法收敛时间更短, 在训练集上的实验精度更高.

### 参考文献:

[1] DURBIN R, RUMELHART D E. Product Units: A Computationally Powerful and Biologically Plausible Extension to Backpropagation Networks [J]. Neural Computation, 1989, 1(1): 133-142.

[2] RUMELHART D E, HINTON G E, WILLIAMS R J. Learning Representations by Back-Propagating Errors [M]. Massachusetts: MIT Press, 1988.

[3] LAN Y, SOH Y C, HUANG G. Constructive Hidden Nodes Selection of Extreme Learning Machine for Regression [J].

- Neurocomputing, 2010, 73(16): 3191–3199.
- [4] LEE C, KO C. Time Series Prediction Using RBF Neural Networks with a Nonlinear Time-Varying Evolution PSO Algorithm [J]. Neurocomputing, 2009, 73(1): 449–460.
- [5] 田增山, 崔永全. 基于 BP 神经网络的 GPS/SINS 姿态测量融合算法[J]. 重庆邮电大学学报: 自然科学版, 2014, 26(4): 478–482.
- [6] 王 林, 罗 毅, 姚 毅. 输电线路覆冰载荷在线检测分析与神经网络预测研究[J]. 重庆邮电大学学报: 自然科学版, 2012, 24(2): 254–258.
- [7] SUN J. Learning Algorithm and Hidden Node Selection Scheme for Local Coupled Feedforward Neural Network Classifier [J]. Neurocomputing, 2012, 79(1): 158–163.
- [8] ZHUO L, ZHANG J, DONG P, et al. An SA-GA-BP Neural Network-Based Color Correction Algorithm for TCM Tongue Images [J]. Neurocomputing, 2014, 134(0): 111–116.
- [9] COELLO C A C, PULIDO G T, LECHUGA M S. Handling Multiple Objectives with Particle Swarm Optimization [J]. IEEE Transactions on Evolutionary Computation, 2004, 8(3): 256–279.
- [10] HAN H, QIAO J. A Structure Optimisation Algorithm for Feedforward Neural Network Construction [J]. Neurocomputing, 2013, 99(1): 347–357.
- [11] PU X, SUN P. A New Hybrid Pruning Neural Network Algorithm Based on Sensitivity Analysis for Stock Market Forecast [J]. Journal of Information and Computational Science, 2013, 10(3): 883–892.
- [12] HAN H, CHEN Q, QIAO J. An Efficient Self-Organizing RBF Neural Network for Water Quality Prediction [J]. Neural Networks, 2011, 24(7): 717–725.
- [13] DING S, SU C, YU J. An Optimizing BP Neural Network Algorithm Based on Genetic Algorithm [J]. Artificial Intelligence Review, 2011, 36(2): 153–162.
- [14] 龚 恒, 张建军. 基于数字图像处理和神经网络技术的 45 钢热处理温度软测量研究[J]. 西南大学学报: 自然科学版, 2013, 23(06): 123–126.

## An Improved BP Neural Network Based on CPA

ZHANG Hong, WANG Dan

*School of Computer and Information Science, Southwest University, Chongqing 400715, China*

**Abstract:** BP (back propagation) neural network is one of the most widely used artificial neural networks. It is known that the performance of the BP neural network depends mainly on its structure. What's more, the BP neural network takes a long time to achieve convergence and the results may fall in local optimum. By combining CPA, variable learning rate and additional momentum, we propose an improved BP neural network pruning algorithm named as LMCPA neural network in this paper. Experimental results have shown that the performance of the neural network has been improved by LMCPA algorithm.

**Key words:** bp neural network; correlation pruning algorithm (CPA); variable learning rate; additional momentum

责任编辑 张 桢

