第45卷第1期 郑州大学学报(理学版) Vol. 45 No. 1
2013年3月 J. Zhengzhou Univ.(Nat. Sci. Ed.) Mar. 2013

# 一种基于粒子群优化的极限学习机

王 杰, 毕浩洋

(郑州大学 电气工程学院 河南 郑州 450001)

摘要: 极限学习机(ELM)是一种新型的前馈神经网络,相比于传统的单隐含层前馈神经网络(SLFN),ELM具有速度快、误差小的优点.由于随机给定输入权值和偏差,ELM通常需要较多隐含层节点才能达到理想精度.粒子群极限学习机算法为使用粒子群算法(particle swarm optimization,PSO)选择最优的输入权值矩阵和隐含层偏差,从而计算出输出权值矩阵.一维Sinc函数拟合实验表明,相比于ELM算法和传统神经网络算法,粒子群极限学习机算法依靠较少的隐含层节点能够获得较高精度.

关键词: 粒子群; 极限学习机; 隐含层节点

中图分类号: TP 183 文献标志码: A 文章编号: 1671-6841(2013)01-0100-05

DOI: 10.3969/j.issn/1671-6841.2013.01.024

## 0 引言

传统的基于梯度下降算法神经网络(如BP神经网络)已经被广泛应用于多层前馈神经网络的训练中[1],但是该网络学习速度慢、容易陷入局部最优解以及在不同的应用场合参数调整复杂.

Huang等人在2004年提出了一种新型的前馈神经网络[2],即极限学习机(ELM).在ELM中,输入权值和隐含层偏差被随机初始化给定,输出权值矩阵利用广义逆(MP)计算得到.相比传统前馈神经网络,ELM学习速度更快、精度更高、参数调整简单,已经得到了不少学者的关注研究[3-4],但是在实际应用过程中,为了达到理想的误差精度,ELM通常需要大量的隐含层节点.

粒子群算法由Kennedy和Eberhart[5-6]在1995年提出,该算法模拟鸟集群飞行觅食的行为,鸟之间通过集体的协作使群体达到最优目的,是一种基于群体智能的优化方法.鸟类捕食最优策略就是搜索目前离食物最近的鸟的周围区域.在粒子群优化算法中,每个待优化的问题的解都是搜索空间中的一个粒子.每个粒子对应一个适应度(fitness value),每个粒子同时对应一个速度,速度决定粒子的移动方向和距离,粒子群就在解空间中搜索最优粒子.粒子群优化算法因为其算法规则简单、收敛速度快、可调参数少、寻优能力强,已经在神经网络优化方面得到了广泛的应用[7-9].

本文提出了粒子群极限学习机算法,利用粒子群优化算法优化选择极限学习机的输入层权值和隐含层偏差,从而计算出输出权值矩阵.通过与其他算法比较,验证了粒子群极限学习机在隐含层节点数目以及网络泛化性上的优势.

## 1 极限学习机(ELM)

对于$N$个任意的各不相同的样本$(\boldsymbol{x}_i,\boldsymbol{y}_i)$,其中$\boldsymbol{x}_i=[x_{i1},x_{i2},\cdots,x_{in}]^{\mathrm{T}}\in\mathbf{R}^n$,$\boldsymbol{y}_i=[y_{i1},y_{i2},\cdots,y_{im}]^{\mathrm{T}}\in\mathbf{R}^m$,则一个具有$L$个隐层节点,激励函数为$g(x)$的前馈神经网络的输出可以表示为

$$f_L(x)=\sum_{i=1}^{L}\boldsymbol{\beta}_i G(\boldsymbol{a}_i\cdot\boldsymbol{x}_i+\boldsymbol{b}_i),\boldsymbol{x}_i\in\mathbf{R}^n,\boldsymbol{a}_i\in\mathbf{R}^n,\boldsymbol{\beta}_i\in\mathbf{R}^m,\tag{1}$$

收稿日期:2012-09-27

基金项目:国家自然科学基金资助项目,编号60905039/F030507.

作者简介:王杰(1959-),男,博士生导师,教授,主要从事模式识别与智能控制研究,E-mail:wj@zzu.edu.cn; 通讯作者:毕浩洋(1988-),男,硕士研究生,主要从事模式识别与智能控制研究,E-mail:bi_haoyang@163.com.

其中,$a_i = [a_{i1}, a_{i2}, \cdots, a_{in}]^T$,是输入层到第 $i$ 个隐含层节点的输入权值,$b_i$ 是第 $i$ 个隐藏层节点的偏差(bias);$\beta_i = [\beta_{i1}, \beta_{i2}, \cdots, \beta_{im}]^T$ 是连接第 $i$ 个隐藏层结点的输出权值; $a_i \cdot x_i$ 表示向量 $a_i$ 和 $x_i$ 的内积. 其中激励函数 $g(x)$ 可以选择为"Sigmoid","Sine"或"RBF"等.

如果这个具有 $L$ 个隐含层结点的前馈神经网络能以零误差逼近这 $N$ 个样本,则存在 $a_i, b_i, \beta_i$ 使

$$f_L(x) = \sum_{i=1}^{L} \beta_i G(a_i \cdot x_i + b_i) = y_i, \quad i = 1, 2, \cdots, L, \tag{2}$$

(2) 式可以简化为

$$H\beta = Y, \tag{3}$$

$H$ 被称作网络的隐含层输出矩阵,在极限学习机算法中,输出权值和偏差可以随机给定,隐层矩阵 $H$ 就变成一个确定的矩阵,这样前馈神经网络的训练就可以转化成一个求解输出权值矩阵的最小二乘解的问题,只需要求出输入权值的最小二乘解就能完成网络的训练,输出权值矩阵 $\beta$ 可由(4) 式得到

$$\tilde{\beta} = H^{+} Y, \tag{4}$$

其中 $H^{+}$ 表示隐含层输出矩阵 $H$ 的 Moore-penrose 广义逆.

## 2　粒子群优化算法(PSO)

PSO 中,每个优化问题的解都是搜索空间中的一只鸟,称之为"粒子". 所有的粒子都有一个由适应度函数决定的适应值,每个粒子被赋予记忆功能,能记住所搜寻到的最佳位置,并且每个粒子还有一个速度决定它们飞翔的方向和距离,然后粒子们就追随当前的最优粒子在解空间中搜索. 搜索的过程为在每一次迭代中,粒子通过跟踪两个"极值"来更新自己. 一个就是粒子本身所找到的最优解,这个解叫做个体极值 $p_{\mathrm{Best}}$,另一个极值是整个种群目前找到的最优解,这个极值是全局极值 $g_{\mathrm{Best}}$.

在 $D$ 维空间中,由 $n$ 个粒子组成的种群 $X = (X_1, X_2, \cdots, X_n)$,其中第 $i$ 个粒子的位置为 $X_i = (x_{i1}, x_{i2}, \cdots, x_{iD})^T$. 将 $X_i$ 带入适应度函数 $f(X_i)$ 中计算出适应度值粒子位置 $\mathrm{Fitness}_i$. 第 $i$ 个粒子的速度为 $V_i = (v_{i1}, v_{i2}, \cdots, v_{iD})^T$,其中粒子 $i$ 的个体极值为 $P_i = (P_{i1}, P_{i2}, \cdots, P_{iD})^T$,种群的全局极值为 $P_g = (P_{g1}, P_{g2}, \cdots, P_{gD})^T$.

在迭代寻优过程中,粒子通过个体极值和全局极值更新自身的速度和位置,即

$$V_{id}^{k+1} = \omega V_{id}^{k} + c_1 r_1 (P_{id}^{k} - X_{id}^{k}) + c_2 r_2 (P_{gd}^{k} - X_{id}^{k}), \tag{5}$$

$$X_{id} = X_{id} + V_{id}, \tag{6}$$

式中,$\omega$ 为惯性权重,调节对解空间的搜索范围,$d = 1, 2, \cdots, D$,$i = 1, 2, \cdots, n$. 其中 $D$ 为待优化问题的维数,$n$ 为种群大小,$k$ 为当前迭代次数,$V_{id}$为粒子速度,$c_1$,$c_2$(非负常数) 为加速度因子,通常取 $c_1 = c_2 = 2$,$r_1$,$r_2$ 为两个随机数,取值范围为(0,1),以增加搜索随机性. 为了防止粒子盲目搜索,一般将位置和速度限制在 $[-X_{\max}, X_{\max}]$、$[-V_{\max}, V_{\max}]$.

## 3　粒子群极限学习机(PSOELM)

由于 ELM 随机给定输入权值矩阵和隐含层偏差,由式(1) ~ (4) 可知输出权值矩阵是由输入权值矩阵和隐含层偏差计算得到的,可能会存在一些输入权值矩阵和隐含层偏差为 0,即部分隐含层节点是无效的. 因此在某些实际应用中,ELM 需要大量的隐含层节点才能达到理想的精度. 并且 ELM 对未在训练集中出现的样本反应能力较差,即泛化能力不足.

针对以上问题本文提出了一种粒子群极限学习机算法,利用粒子群优化和极限学习机网络相结合的学习算法,即利用粒子群优化算法优化选择极限学习机的输入层权值和隐含层偏差,从而得到一个最优的网络. 训练步骤如下所示:

1) 产生种群. 粒子数 popsize 一般设置为 20 ~ 40 个,对于某些复杂问题可以设置 100 ~ 200 个,种群中的个体即粒子是由输入权值矩阵和隐含层偏差构成,粒子长度为 $D = k \cdot (n+1)$,其中,$k$ 为隐含层节点数目,$n$

为输入层神经元数目,即输入向量的维数. $\theta^m = [\omega_{11}^m, \omega_{12}^m, \cdots, \omega_{12}^m, \omega_{21}^m, \omega_{22}^m, \cdots, \omega_{2k}^m, \cdots, \omega_{n1}^m, \omega_{n2}^m, \cdots, \omega_{nK}^m, \cdots, b_1^m, b_2^m, \cdots, b_k^m]$,其中,$\theta^m$ 为种群中的第 $m(1 \leqslant m \leqslant \text{popsize})$ 个粒子;$\omega_{ij}^m$、$b_j^m$ 为 $[-X_{\max}, X_{\max}]$ 中的随机数. 一般 $X_{\max}=1$.

2) 对于种群中的每个个体(输入层权值矩阵和隐含层偏差),利用 ELM 算法(其中网络的隐含层激活函数选取应用最普遍的 sigmoid 函数)即式(1)~(4)可以计算出输出权值矩阵,利用训练样本计算出初始化种群的每个个体的均方根误差(RMSE),将 RMSE 作为粒子群优化算法的适应度. 计算种群中的第 1 个粒子 $\theta^1$ 的适应度值 Fitness_1,令最优适应度值 Fitness_best = Fitness_1; 并将 $\theta^1$ 作为 $p_{\text{Best}}$; 然后执行以下伪代码.

```
Step1  For(1 < i ≤ popsize)
    {
    计算 θ^i 的适应度值 Fitness_i;
    If   Fitness_i > Fitness_best
         则 p_Best(i) = θ^i
         Fitness_best = Fitness_i
      Else   p_Best(i) = θ^(i-1)
    }
    令 Fitness_best 对应的粒子 θ 为全局极值 g_Best.
Step2  For (1 ≤ i ≤ itera)   itera 为最大迭代次数
    {
         由(5)~(6)更新粒子速度和位置;
         If   满足约束条件或者 i > itera
         Break
         Else   跳转到 step1
    }
```

3) 执行以上伪代码可以得到最优适应度所对应的粒子即输入权值矩阵和隐含层偏差,利用 ELM 算法可以求得输出权值矩阵.

## 4 仿真实验

为了测试 PSOELM 的效果,本文做了一维 Sinc 函数拟合实验,通过与基于 LM 算法的 BP 神经网络(LMBP)、ELM、支持向量机(SVM) 3 种网络进行比较来测试 PSOELM 的效果.

Sinc 函数表达式为

$$f(x) = \begin{cases} \dfrac{\sin x}{x}, x \neq 0 \\ 0, x = 0 \end{cases}.$$

选取 5 000 个数据集 $\{x_i, f(x_i)\}$,其中 $x_i$ 服从 $[-10, 10]$ 的均匀分布; 产生一组 $[-0.2, 0.2]$ 的均匀分布的噪音 $\zeta_i$,则训练样本集为 $\{x_i, f(x_i)+\zeta_i\}$. 同时选取 5 000 个测试样本集为 $\{x_i, f(x_i)\}$,其中 $x_i$ 为 $[-10, 10]$ 的等间隔采样.

将 PSOELM 的隐含层节点设置为 10 个,进化代数分别选择 $i=1, 2, \cdots, 7$,考虑到测试结果的波动,每次试验做了 20 组求取平均值(下同),测试 PSOELM 的 RMSE,测试结果如表 1 所示; 设置 PSOELM 的最大迭代次数为 5,将 ELM 与 PSOELM 的隐含层节点数目逐渐增加,测试它们的 RMSE. 测试效果如表 2 所示.

从表 1 中可知,当隐含层节点固定时,随着迭代次数的增加,训练时间也逐渐增加,训练误差和测试误差则减小. 当迭代次数达到一定值时,测试误差减小变缓. 由表 2 可以看出,随着隐含层节点的增加,PSOELM 与 ELM 的 RMSE 都逐渐减小,当隐含层节点为 20 个时 PSOELM 效果达到最佳,而 ELM 当隐含层节点数目增加到 100 个时才逐渐稳定,但是仍然没有 PSOELM 效果好. 当隐含层节点数目增加到 1 000 个时,ELM 算法得到的 RMSE 为 0.008 1,误差变化不大,考虑到 PSOELM 的学习时间较长,没有对其做测试. 综合表 1、表 2,设隐含层设置 PSOELM 的最大迭代次数为 5,隐含层节点数目为 20,Sinc 函数的拟合曲线如图 1 所示.

**表 1**　迭代次数对 PSOELM 的影响

**Tab. 1**　The impact of the iteration number to PSOELM

| 最大迭代次数 | 训练时间/s | 训练误差 | 测试误差 |
| --- | --- | --- | --- |
| 1 | 2.515 6 | 0.119 2 | 0.033 1 |
| 2 | 3.312 5 | 0.116 5 | 0.022 4 |
| 3 | 4.500 0 | 0.116 4 | 0.018 2 |
| 4 | 5.593 8 | 0.115 1 | 0.017 3 |
| 5 | 6.828 1 | 0.114 9 | 0.014 7 |
| 6 | 8.046 9 | 0.114 2 | 0.014 2 |
| 7 | 9.187 5 | 0.113 0 | 0.014 0 |

**表 2**　隐含层节点对 ELM 与 PSOELM 影响

**Tab. 2**　The impact of hidden layer nodes to ELM and PSOELM

| 隐含层节点数目 | 训练时间/s | | 训练误差 | | 测试误差 | |
| --- | --- | --- | --- | --- | --- | --- |
| | ELM | PSOELM | ELM | PSOELM | ELM | PSOELM |
| 5 | 0 | 6.828 1 | 0.279 1 | 0.114 9 | 0.253 2 | 0.014 7 |
| 10 | 0.031 3 | 6.656 3 | 0.119 9 | 0.115 8 | 0.039 1 | 0.019 6 |
| 15 | 0.042 5 | 10.265 6 | 0.119 8 | 0.113 7 | 0.038 8 | 0.006 4 |
| 20 | 0.066 9 | 14.703 1 | 0.113 9 | 0.113 7 | 0.008 8 | 0.006 2 |
| 50 | 0.2188 | 45.046 9 | 0.113 8 | 0.113 6 | 0.008 4 | 0.006 2 |
| 100 | 0.562 5 | 140.906 3 | 0.113 8 | 0.113 6 | 0.008 2 | 0.006 2 |
| 150 | 0.968 8 | 256.171 9 | 0.113 8 | 0.113 6 | 0.008 2 | 0.006 2 |
| 1 000 | 39.609 4 | ~ | 0.113 8 | ~ | 0.008 1 | ~ |

在表 3 中,本文给出了 PSOELM、ELM、SVM 和 LMBP 在本实验中的性能对比,其中 PSOELM 隐含层节点设置为 20 个,最大迭代次数设置为 5,ELM 隐含层节点选择为 100 个,隐含层激活函数为'sigmoid'. SVM 采用 libsvm 工具包[10],SVM 的参数采用交叉验证来完成,LMBP 采用 matlab 自带的神经网络工具箱函数来进行训练,4 种算法的性能对比如表 3 所示.

从表 3 中可以看出,针对一维 Sinc 函数拟合实验,PSOELM 与 SVM 的误差最低,泛化性能最好. PSOELM 只需要 20 个隐含层节点就能达到比需要 100 个隐含层节点的 ELM 更好的效果.

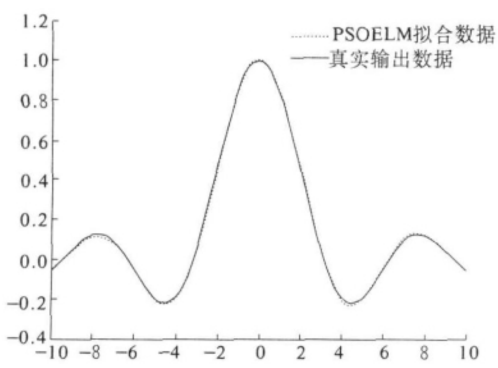


**图 1**　PSOELM 拟合 Sinc 函数

**Fig. 1**　The approximation curves of the Sinc function

**表 3**　4 种算法性能比较

**Tab. 3**　Four algorithms performance comparison

| 算法 | 训练时间/s | 训练误差 | 测试误差 |
| --- | --- | --- | --- |
| PSOELM | 14.703 1 | 0.113 7 | 0.006 2 |
| ELM | 0.562 5 | 0.113 8 | 0.008 2 |
| SVM | 9.796 9 | 0.113 0 | 0.006 4 |
| LMBP | 36 | 0.000 1 | 0.009 1 |

## 5 结论

本文提出了一种基于粒子群优化的极限学习机算法,采用ELM的学习算法,利用最小二乘法快速求解输出权值矩阵而不是利用迭代调整的算法;同时采用PSO算法优化输入权值矩阵和隐含层偏差;该算法综合了ELM和PSO的优点:参数调整简单、全局最优性、泛化能力强.并给出了该算法的详细步骤.仿真实验结果表明,该算法只需要较少的隐含层节点就能取得较好的效果.

## 参考文献:

[1] 刘天舒.BP神经网络的改进研究及应用[D].东北农业大学,2011.
[2] Huang Guangbin, Zhu Qinyu, Siew Cheekheong. Extreme learning machine: theory and applications[J]. Neurocomputing, 2006,70(1/2/3):489-501.
[3] Deng W, Chen L. Color imagewatermarking using regularized extreme learning machine[J]. Neural Network World, 2010,20(3):317-330.
[4] Zong Weiwei, Huang Guangbin. Face recognition based on extreme learning machine[J]. Neurocomputing, 2011,74(16):2541-2551.
[5] Kennedy J, Eberhart R. Particle swarm optimization[C]//Proceedings of IEEE International Conference on Neural Networks. Piscataway, 1995: 1942-1948.
[6] Eberhart R, Kennedy J. A new optimizer using particle swarm theory[C]//Proc of the Sixth International Symposium on Micro Machine and Human Science. Nagoya, 1995: 39-43.
[7] 冯冬青,杨书显.氧乐果合成过程的PSO回归BP网络建模方法[J].郑州大学学报:理学版,2011,43(3):113-117.
[8] 严晓明,郑之.基于混合仿生算法的SVM参数优化[J].广西师范大学学报:自然科学版,2011,29(2): 115-118.
[9] 徐海龙,王晓丹,廖勇,等.一种基于PSO的RBF-SVM模型优化新方法[J].控制与决策,2010,25(3):368-377.
[10] Chih Chung Chang, Chih Jen Lin. LIBSVM: a library for support vector machines[EB/OL]. [2012-11-16] http://www.csie.ntu.edu.tw/~cjlin/libsvm.

# A New Extreme Learning Machine Optimized by PSO

WANG Jie, BI Hao-yang

(*School of Electrical Engineering, Zhengzhou University, Zhengzhou* 450001, *China*)

**Abstract:** Extreme learning machine (ELM) was a new type of feedforward neural network. Compared with traditional single hidden layer feedforward neural networks, ELM possessed higher training speed and smaller error. Due to random input weights and hidden biases, ELM might need numerous hidden neurons to achieve a reasonable accuracy. A new ELM learning algorithm, which was optimized by the particle swarm optimization (PSO), was proposed. PSO algorithm was used to select the input weights and bias of hidden layer, then the output weights could be calculated. To test the validity of proposed method, two simulation experiments were drawn on the approximation curves of the Sinc function. Experimental results showed that the proposed algorithm achieved better performance with less hidden neurons than other similar methods.

**Key words:** particle swarm optimization; extreme learning machine; hidden neurons