

改进粒子群算法对 BP 神经网络的优化^①

沈学利 张红岩 张纪锁 (辽宁工程技术大学 电子与信息工程学院 辽宁 葫芦岛 125105)

摘要: 介绍一种基于改进粒子群算法优化 BP 网络的权值调整综合方法。该算法在传统 BP 算法的误差反传调整权值的基础上, 引入粒子群算法的权值修正, 并且在训练神经网络权值的同时优化其连接结构, 删除冗余连接, 从而建立了基于粒子群算法优化的 BP 网络新模型。结果表明, 改进算法不仅可以克服传统 BP 算法收敛速度慢和易陷入局部权值的局限, 而且很大程度地提高了结果精度和 BP 网络学习能力。

关键词: 粒子群算法; 惯性权值; 神经网络; BP 算法; 优化

Improved Particle Swarm Optimization Algorithm

SHEN Xue-Li, ZHANG Hong-Yan, ZHANG Ji-Suo

(School of Electronic and Information Engineering, Liaoning Technical University, Huludao 125105, China)

Abstract: A new method to adjust weights of BP network is proposed. The new model is based on the weight adjustments of traditional BP algorithm by tuning the structure and connection weights of BP network and improved particle swarm optimization simultaneously. The result shows that the improved algorithm can not only overcome the limitations in both the slow convergence and the local extreme values of traditional BP algorithm, but also improve the precision of the result and the learning ability greatly.

Keywords: particle swarm optimization; inertia weight; neural networks; back propagation arithmetic; optimization

粒子群优化算法(Particle Swarm Optimization, PSO)最初是由 Eberhart 博士和 Kennedy 博士于 1995 年提出并成功地用于函数优化^[1], 后来又进行了有效的拓展, 是计算智能领域除蚁群优化算法(Ant Colony Optimization, ACO)^[2]外的另外一种群体智能算法(Swarm Intelligence, SI)^[3]。PSO 算法属于进化算法的一种, 和遗传算法相似, 它也是从随机解出发, 通过迭代寻找最优解, 通过适应度来评价解的品质。但是它比遗传算法规则更为简单, 没有遗传算法的“交叉”(Crossover)和“变异”(Mutation)操作。它通过追随当前搜索到的最优值来寻找全局最优。由于原理上十分简单, 所需参数也较少, 收敛速度快和算法本身的易实现性, 目前 PSO 算法已经被广泛应用于神经网络训练、目标函数优化、模糊控制系统等许多领域。

BP 网络是一种有效的自学习神经网络, 具有一些独特的性质: 信息的分布式存储和并行处理, 具有自组织、自学习能力等, 已经被广泛应用于模式识别、知识工程、智能控制等方面。尽管在诸多应用领域取得了巨大的成功, 然而仍存在着一些问题: (1)学习算法的收敛速度慢; (2)局部极小问题; (3)BP 神经网络隐层神经元的个数选取目前尚无理论上的依据, 只能根据经验选取。

因此, BP 网络的优化改进成为了研究的热点之一。近年来, 有人将 PSO 的思想引入到 BP 网络优化中, 用粒子群的迭代来代替 BP 算法中的梯度修正。此策略可以缩短网络训练的时间, 提高算法的收敛速度。根据不同的系统目标, 改进原有的粒子群算法, 就能进一步提高 BP 网络的性能, 使其实用性更强。为了在全局搜索和局部搜索之间取得更好的平衡。本文根据

① 收稿时间:2009-06-08

惯性权重和学习因子对粒子群优化算法性能影响的研究, 分别对他们做出了改进, 同时本文还对神经网络存在的冗余连接会降低神经元的处理效率, 大量的冗余连接甚至会影响到 BP 网络分类的正确率。改进的 PSO 在训练神经网络权重的同时训练其连接结构, 删除冗余连接, 使神经网络获得与给定问题匹配的信息处理能力。通过实验分析显示: 改进粒子群优化算法用于神经网络优化不仅能更快地收敛于最优解, 而且很大程度地提高了结果的精度。

1 BP神经网络

BP 网络是一种多层结构的前向网络, 在此重要简单介绍三层的 BP 网络, 即输入层、隐含层 (也称中间层) 和输出层, 各层之间实行全连接, 同一层内的神经元无连接, 图 1 表示具有一个隐含层的 BP 神经网络结构, 输入层和输出层节点数根据实际情况确定, 而隐含层节点数一般按如下公式计算: 设 N_H 表示隐含层节点数, N_i 表示输入层节点数, N_o 表示输出层节点数, 则:

$$N_H = \sqrt{N_i + N_o} + L \quad L=0,1,2,\dots,10 \quad (1)$$

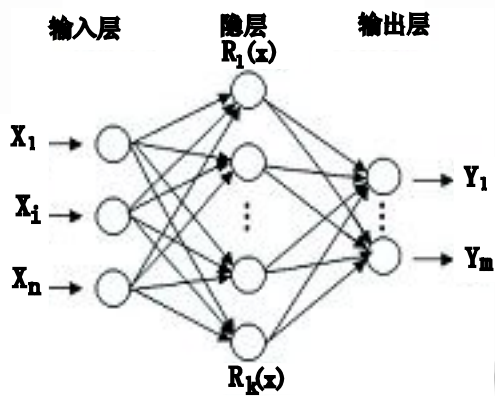


图 1 三层网络结构

其隐层神经元转移函数有: 阶跃函数、准线性函数、Sigmoid 函数和双正切函数等; 典型的转移函数 Sigmoid 函数如下:

$$F(x) = \frac{1}{1 + e^{-2x/u_0}} \quad (2)$$

$$\text{导函数: } f'(x) = \frac{2}{u_0} f(x)(1 - f(x)) \quad (3)$$

$$\text{误差: } E = \frac{1}{2} \sum_{k=1}^N (T_k - Y_k)^2 \quad (4)$$

2 粒子群优化算法原理

PSO 算法中每个粒子即解空间中的一个解, 它根据自己的移动经验和同伴的移动经验来调整自己的移动。每个粒子在移动过程所经历过的最好位置, 就是粒子本身找到的最优解。整个群体所经历过的的最好位置, 就是整个群体目前找到的最优解。前者叫做个体极值(p_{best}), 后者叫做全局极值(g_{best})。每个粒子都通过上述两个极值不断更新自己, 从而产生新一代群体。实际操作中通过由优化问题所决定的适应度函数值(fitness value)来评价粒子的优劣程度。显然, 每个粒子的行为就是种追随着当前的最优粒子在解空间中搜索。

在 PSO 算法中每个粒子可以看作是解空间中的一个点。如果粒子的群体规模为 N , 则第 i ($i = 1, 2, \dots, N$) 个粒子的位置可表示为 x_i , 它所经历过的“最好”位置记为 $p_{best}[i]$, 它的速度用 v_i 表示, 群体中“最好”粒子的位置的索引号用 $best$ 表示。所以粒子 i 将根据下面的公式来更新自己的速度和位置:

$$v_i = w * v_i + c_1 * rand() * (p_{best}[i] - x_i) + c_2 * rand() * (g_{best}[i] - x_i) \quad (5)$$

$$x_i = x_i + v_i \quad (6)$$

其中 c_1 、 c_2 为常数, 称为学习因子, 通常 $c_1 = c_2 = 2$; $rand()$ 是 $[0, 1]$ 上的随机数, w 为惯性权重(inertia weight)。公式由三部分组成, 第一部分是粒子先前的速度, 说明了粒子目前的状态, 起到平衡全局搜索和局部搜索的作用; 第二部分是认知部分(Cognition Modal), 表示粒子本身的思考, 使粒子有了足够强的全局搜索能力, 避免局部极小; 第三部分为社会部分(Social Modal), 体现粒子间的信息共享。三个部分共同决定了粒子的空间搜索能力。

另外, 粒子在不断根据速度调整自己的位置时, 还要受到最大速度(V_{max})的限制。当 v_i 超过 V_{max} 时将被限定为 V_{max} 。

3 粒子群算法改进

粒子群优化算法结构简单, 运行速度较快, 当在寻优过程中, 如果某个粒子找到一个局部最优解, 其它粒子将迅速向其移动, 这样就会陷入局部最优解, 从而失去找到全局最优解的机会。

为了降低陷入局部机会, 始终保持粒子具有一定

的多样性,从而引入改进粒子群优化(Improved Particle Swarm Optimization 简记为:IPSO)算法,本文给出学习因子 c_1 、 c_2 和惯性权重 w 的改进。

3.1 学习因子改进

在粒子群优化算法中,加速系数 c_1 和 c_2 分别控制“认知”部分和“社会”部分对粒子速度的影响。一般来讲,在基于种群的优化方法中,总是希望个体在初始阶段能够在整个寻优空间进行搜索,不至于过早陷入局部值;而在结束阶段能够提高算法收敛速度和精度,有效地寻找全局最优解。因此,我们可以在进化过程中动态的调整加速系数 c_1 和 c_2 的值,使得在算法的初始阶段,具有大的“认知”部分(大的 c_1)和小的“社会”部分(小的 c_2),以利于算法在整个寻优空间进行搜索;而在算法后期,应有小的“认知”部分(小的 c_1)和大的“社会”部分(大的 c_2),更利于算法收敛于全局最优解,提高算法收敛速度和精度^[4]。为此,取 c_1 和 c_2 分别如下:

$$c_1 = (c_{start} - c_{end}) \frac{Maxiter - iter}{Maxiter} + c_{end} \quad (7)$$

$$c_2 = 4 - c_1 \quad (8)$$

其中 $MAXiter$ 和 $iter$ 分别为算法最大叠代次数和当前代数。 c_{start} 和 c_{end} 为 c_1 的初始值和最终值, $0 < c_{end} < c_{start} \leq 4$ 。

实验结果表明,这种方法加速算法的收敛,在单峰值函数的测试表现优异。为此付出的代价是算法容易陷入局部最小值,在多峰值函数的测试中容易过早收敛。

3.2 惯性权值改进

固定权值的选择就是选择某一常数为惯性权值,在优化过程中不变。然而在实验中的效果并不十分理想,所以开始研究惯性权值 w 对优化性能的影响,发现较大的 w 值有利于跳出局部极小点,而较小的 w 值有利于算法收敛,提出了一种是根据算法叠代次数使惯性权值线性递减的方法。算法在初期使用较大惯性权值,具有较强全局搜索能力,后期则使用较小惯性权值,提高局部搜索能力。为了找到一种能在全局搜索和局部搜索之间取得最佳平衡的惯性权值选取方法,研究人员进行了大量的研究工作,先后提出了线形递减权值(LDIW)策略^[5]、模糊惯性权值(FIW)策略^[6]和随机惯性权值(RIW)策略^[7]。惯性权值的计算公式

如下:

$$w = (w_{start} - w_{end}) \frac{Maxiter - iter}{Maxiter} + w_{end} \quad (9)$$

其中, w_{start} 和 w_{end} 分别是惯性权值的初始值和最终值,且 $0.1w_{end} < w_{start} \leq 0.9$, $Maxiter$ 和 $iter$ 分别为算法最大叠代次数和当前代数。

3.3 测试

利用典型函数 Schaffer 函数、Rosenbrock 函数和 Rastrigrin 函数测试算法^[8]。参数设置:种群个数 50,最大叠代次数 5000,每个函数优化 30 次,学习因子 $c_1=c_2=2$,典型函数都取 2 维测试;测试结果如表 1、2、3,其中 $iter$ 代表叠代次数, fit_value 代表适应值。

表 1 Schaffer 函数测试结果

w 选择	平均 iter	平均 fit_value (10^{-6})
0.6	212	8.45
0.8	287	6.67
0.4-0.9	220	9.43
0.2-0.6	165	5.66

表 2 Rosenbrock 函数测试结果

w 选择	平均 iter	平均 fit_value (10^{-6})
0.6	2194	0.0652
0.8	2635	1.0453
0.4-0.9	1926	1.0835
0.2-0.6	1186	0.0497

表 3 Rastrigrin 函数函数测试结果

w 选择	平均 iter	平均 fit_value ^o (10^{-6})
0.6	113	2.88
0.8	402	9.04
0.4-0.9	1142	9.15
0.2-0.6	79	3.20

从结果中可以看出:①在标准粒子群优化算法,使用固定惯性权值与时变惯性权值对比,可以看出收敛速度有很大的进步;②采用典型的线性递减收敛性质,收敛速度以及平均最好适应值方面都有了很大的改善。从对测试函数的测试结果来看,LDWPSO 取得了较好的效果。但它也有缺点,迭代初期局部搜索能力较弱,即使初始粒子已接近于全局最优点,也往往错过,而

在迭代后期,则因全局搜索能力变弱,而易陷入局部极值。然而本文把学习因子 c_1 、 c_2 和惯性权重 w 的改进综合改进 PSO 算法,这样明显弥补了单独改进一方面所带来的缺陷。

4 优化BP神经网络

将IPSO用于训练BP网络的方法是:粒子群中每个粒子的位置表示BP网络中当前迭代中的权值集合,每个粒子的维数由网络中起连接作用的权值的数量和阈值个数决定。以给定训练样本集的神经网络输出误差作为神经网络训练问题的适应函数,适应度值表示神经网络的误差,误差越小则表明粒子在搜索中具有更好的性能。粒子在权值空间内移动搜索使得网络输出层的误差最小,改变粒子的速度即更新网络的权值,以减少均方误差(MSE)。通过这种方式,IPSO优化搜索训练神经网络的权值和阈值来获得更小的MSE。每次迭代过程中产生MSE最小的粒子为目前全局最优的粒子,即算法公式5中的 g_{best} 。本网络采用隐层数目可变的方法,即一开始放入较多的隐层节点,随着训练过程的进行,逐步修减掉在网络训练中不起作用的节点。对于每一神经元,若在训练过程中,其连接权的数值的绝对值小于事先确定的数值 ϵ ,则删除此神经节点。

次数,则结束;

步骤 3: 对每个粒子 i 在网络中正向传播,计算其在输出层的输出,得到误差 E_i , 如果 $E_i < p_{best}$, 则 $p_{best} = E_i$, $p_i = x_i$, 如果 $E_i < g_{best}$, 则 $g_{best} = E_i$, $pg = x_i$;

步骤 4: 优化网络连接结构,删除冗余连接,然后根据改进后的公式分别计算并更新每个粒子的速度和位置;

步骤 5: 迭代次数加 1, 并转步骤 2;

5 试验结果分析

为了测试该算法的效果,我们利用经典的异或问题(XOR)分别对传统BP、遗传算法改进BP(GA-BP)、PSO-BP和IPSO-BP进行实验测试比较。采用3层BP网络,共有4个样本,输入向量为 $p=[0,0;1,1;0,1;1,0]$,目标向量为 $t=[0,0,1,1]$ 。其中3层BP网络初始隐层节点数为60,最大迭代次数 $Maxiter=500$ 。试验结果如表5所示:

表 5 异或实验结果对比

算法	隐层节点数	平均迭代次数	期望误差 (10^{-2})	平均CPU时间(ms)
BP	12	209	0.99	562
GA-BP	9	156	0.05	453
PSO-BP	10	43	0.01	281
IPSO-BP	7	29	0.001	143

从表5中还可以看出,利用IPSO优化后的BP神经网络与其它3种算法相比,在连接数、平均迭代次数和平均CPU时间等方面都有很大的改善,尤其是远远好于传统BP网络,这说明本文提出的利用IPSO优化后的BP神经网络取得了良好的训练效果。

为了进一步验证IPSO优化后的BP神经网络有效性和收敛性,与其他算法在训练时的误差(适应值)比较曲线图3所示:

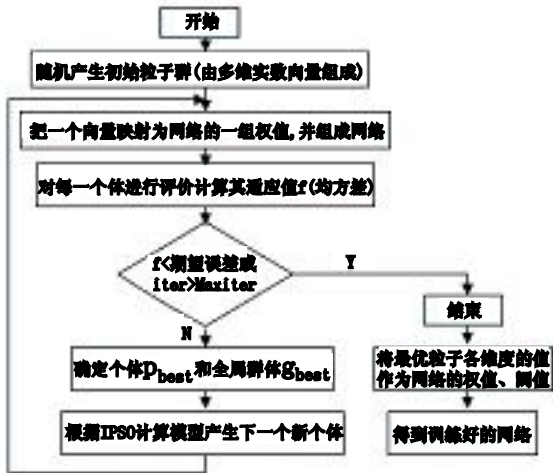


图 2 IPSO-BP 算法流程图

IPSO-BP 算法如下:

步骤 1: 初始化群体属性,包括神经网络拓扑结构、种群大小、迭代最大次数、误差限等

步骤 2: 如果算法收敛准则满足或达到最大迭代

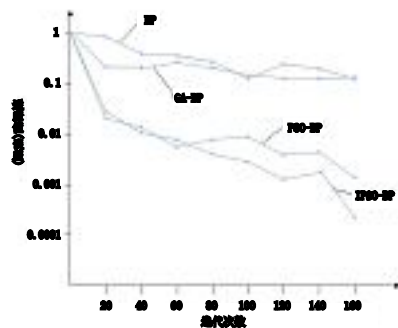


图 3 各个算法收敛情况

通过图 3 还可以看出在训练误差上, IPSO-BP 算法比其它 3 种算法都要小很多, 尤其对传统 BP 网络。表明 IPSO-BP 算法有很强的学习能力和泛化能力。

6 结语

IPSO-BP 算法作了两个方面的改进, 即改进了经典 PSO 算法的学习因子和惯性权值; 其次, 删除了 BP 神经网络的冗余连接, 精简网络结构。实验表明, 该综合算法可以有效提高 BP 神经网络的收敛速度和精度。比单独的改进学习因子或惯性权值在收敛速度、计算精度、最优解的搜索能力、算法稳定性等方面有明显的优势, 并且 IPSO-BP 算法还不受网络结构的影响。它与传统 BP 算法、GA-BP 算法和标准 PSO-BP 算法的训练结果相比, 具有更小的训练误差和检验误差, 不仅能更快地收敛于最优解, 而且很大程度地提高了结果的精度。在一定程度上提高了学习能力与泛化能力。

参考文献

- 1 Kennedy J, Eberhart R. Particle Swarm Optimization. IEEE on Networks, 1995. 1942 - 1948.
- 2 刘峡壁. 人工智能导论. 北京: 国防工业出版社, 2008, 257 - 260.
- 3 Angeline P J. Evolutionary Optimization Versus Particle Swarm Optimization: Philosophy and Performance Differences. Proc. Seventh Annual Conference on Evolutionary Programming, 1998, 256 - 260.
- 4 何佳, 陈智慧, 杨迎新. 综合改进的粒子群神经网络算法. 计算机工程与设计, 2008, 29(11): 2890 - 2896.
- 5 Shi Y, Eberhart R. Empirical study of particle swarm optimization. International Conference on Evolutionary Computation. Washington, USA: IEEE, 1999. 1945 - 1950.
- 6 Shi Y, Eberhart R. Fuzzy adaptive swarm optimization. The IEEE Congress on Evolutionary Computation. San Francisco, USA: IEEE, 2001. 101 - 106.
- 7 Eberhart R, Shi Y. Tracking and optimizing dynamic systems with particle swarms. The IEEE Congress on Evolutionary Computation. San Francisco, USA: IEEE, 2001. 94 - 100.
- 8 Yasuda K, Ide A, Iwasaki N. Adaptive particle swarm optimization. 2003 IEEE International Conference on Systems, Man and Cybernetics. Piscataway: IEEE Service Center, 2003. 1554 - 1559.