

# 带启发信息的蚁群神经网络训练算法

赵章明 冯 径 施 恩 舒晓村

(国防科技大学气象海洋学院 南京 211101)

**摘要** ACO<sub>R</sub> 算法是一种应用于连续域实值优化的蚁群算法(Ant Colony Optimization, ACO)。ACO<sub>R</sub> 算法可用于训练神经网络。与常规的蚁群算法不同,ACO<sub>R</sub> 在训练神经网络时未考虑启发式信息(Heuristic Information)。在ACO<sub>R</sub> 的基础上,提出了一种将启发式信息与ACO<sub>R</sub> 相结合的神经网络训练算法——h-ACO<sub>R</sub>。其中,启发式信息是通过计算神经网络的误差关于网络的权值向量的偏导数而得到的梯度向量。通过十折交叉验证方法,将h-ACO<sub>R</sub> 应用于UCI数据集中的zoo,iris和tic-tac-toe 3组数据的模式分类问题中来训练神经网络。与ACO<sub>R</sub> 相比,h-ACO<sub>R</sub> 算法在减小分类误差的同时能够提高收敛速度,其收敛的代数约为ACO<sub>R</sub> 算法的1/2,且经过完全训练,对zoo,iris和tic-tac-toe 3组数据的分类准确率分别为91.1%,93.3%和95.6%,高于ACO<sub>R</sub> 算法的83.1%,88.7%和91.9%。

**关键词** 蚁群算法,启发式信息,人工神经网络,神经网络训练

**中图分类号** TP183 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2017.11.043

## h-ACO<sub>R</sub>: An ACO<sub>R</sub> Algorithm with Heuristic Information for Neural Network Training

ZHAO Zhang-ming FENG Jing SHI En SHU Xiao-cun

(Institute of Meteorology and Oceanography, National University of Defense Technology, Nanjing 211101, China)

**Abstract** The ACO<sub>R</sub> algorithm is an ant colony optimization(ACO) algorithm for real-valued optimization. The ACO<sub>R</sub> can be used for training neural network. Unlike most of the conventional ACO algorithms, ACO<sub>R</sub> does not consider heuristic information when training neural networks. So in this work, a new algorithm named h-ACO<sub>R</sub> that incorporates the heuristic information into the framework of ACO<sub>R</sub> was proposed for neural network training. The heuristic information in h-ACO<sub>R</sub> is a gradient vector, which is obtained by computing the partial derivative of error term of the neural network with respect to weight vector. Using 10-fold cross-validation method, h-ACO<sub>R</sub> is applied to train neural networks for pattern classification problems of zoo, iris and tic-tac-toe in UCI datasets. Compared with ACO<sub>R</sub>, h-ACO<sub>R</sub> can reduce classification errors while speeding up the convergence process, with the average training generations of h-ACO<sub>R</sub> being nearly 1/2 of that of ACO<sub>R</sub>. After completely training by h-ACO<sub>R</sub>, the classification accuracy about zoo, iris and tic-tac-toe are respectively 91.1%, 93.3% and 95.6%, which have better performance than that of ACO<sub>R</sub>'s 83.1%, 88.7% and 91.9%.

**Keywords** ACO, Heuristic information, Artificial neural network, Neural network training

## 1 引言

神经网络(Artificial Neural Networks, ANN)被广泛应用于数据挖掘、模式识别和分类等诸多领域<sup>[1]</sup>,其中最常用的是前馈神经网络(Feed-Forward Neural Network, FFNN)。Hornik等<sup>[2]</sup>证明,多层前馈神经网络能够在任意精度模拟复杂的非线性关系,而实现该性能是基于对网络的充分训练。因此,神经网络训练算法对神经网络的性能有着决定性的影响。

根据是否需要梯度信息,可将常用的神经网络训练算法分为两类:1)基于随机梯度下降的算法,如BP算法(Back Propagation)、共轭梯度法(Conjugate Gradient)等,这类算法通过梯度信息引导权值的移动,收敛速度较快,但是Rmulhart等<sup>[3]</sup>证明这类算法依赖于初始权值的选择,且容易陷入

局部最优;2)基于遗传算法(Genetic Algorithm, GA)、粒子群算法(Particle Swarm Optimization, PSO)等智能算法的训练算法,这类训练算法继承了智能算法的并行搜索策略和全局优化特性,能够提高网络的训练准确率,但随着问题规模和复杂程度的增加,训练时间呈指数级增长<sup>[4]</sup>。

蚁群算法(ACO)是Dorigo受蚂蚁在寻找食物的过程中寻找最短路径的行为的启发而提出的一种智能算法<sup>[5]</sup>。Socha K等<sup>[6]</sup>基于蚁群算法,提出了解决连续域实值优化问题的ACO<sub>R</sub>算法,并将其成功应用于神经网络训练<sup>[7]</sup>。类似于GA和PSO,ACO<sub>R</sub>算法在神经网络训练过程中未使用梯度信息,其在训练过程中所依靠的问题的相关信息只有差错函数。由于梯度信息能够快速、直接地引导权值的移动,因此基于梯度信息的训练算法是目前训练神经网络最常用的方法。

到稿日期:2016-10-13 返修日期:2016-12-21 本文受国家自然科学基金(61371119)资助。

赵章明(1993-),男,硕士生,主要研究方向为人工智能、神经网络,E-mail:zzm\_ust@qq.com;冯 径(1962-),女,博士,教授,主要研究方向为计算机网络、信息系统集成;施 恩(1992-),男,硕士生,主要研究方向为神经网络;舒晓村(1983-),男,硕士,主要研究方向为信息系统集成。

因此,本文在  $ACO_R$  算法的基础上,提出一种将网络的梯度信息加入  $ACO_R$  算法框架中并使其作为蚁群的启发式信息的改进算法,本文称之为  $h-ACO_R$ ,同时用这种改进的  $ACO_R$  算法训练固定拓扑网络。

本文选取加州大学欧文分校(University of California Irvine,UCI)机器学习库中的 zoo,iris 和 tic-tac-toe 3 组数据,分别利用  $h-ACO_R$  算法和  $ACO_R$  算法训练 3 层固定拓扑结构的前馈神经网络(FFNN)对上述 3 组数据进行分类,以比较两种算法的优劣。实验采用十折交叉验证法<sup>[8]</sup>,结果表明  $h-ACO_R$  算法在减小测试集的均方误差(Mean Squared Error, MSE)的同时能够提高收敛速度,其收敛的代数约为传统  $ACO_R$  算法的 1/2。

## 2 相关研究

### 2.1 蚁群算法

蚁群算法(ACO)是 Dorigo 等最早提出的一种基于种群的模拟进化算法,常用来求解诸如旅行商问题(Traveling Salesman Problem)、调度问题(Scheduling Problem)等组合优化问题(Combinatorial Optimization Problems, COPs)<sup>[9]</sup>。定义组合优化问题的模型如下<sup>[10]</sup>:

**定义 1** 组合优化问题模型  $P=(S, \Omega, f)$ , 其中:

(1)  $S$  为由有限个值为  $v_i^j \in D_i = \{v_i^1, \dots, v_i^{D_i}\}$  的离散变量  $X_i (i=1, \dots, n)$  组成的搜索空间;

(2)  $\Omega$  关于变量  $X_i (i=1, \dots, n)$  的约束集;

(3) 目标函数  $f: S \rightarrow R^+$ 。

用蚁群算法求解组合优化问题可以被视为:寻找至少一个全局最优解  $s^* \in S$ , 使  $f(s^*) \leq f(s), \forall s \in S$ , 其中  $s \in S$  为满足  $\Omega$  中的所有约束条件的可行解。所有全局最优的集合记作  $S^* \subseteq S$ 。

蚁群算法求解过程中首先将所求的问题转换为有向图  $G=(N, E, T)$ , 其中  $N=\{c_1, \dots, c_n\}$  为图的结点集;  $E=\{(c_i, c_j) | c_i, c_j \in N\}$  表示图中结点间的连通情况;  $T=\{\tau_{ij} | i, j \in N\}$  为信息素向量, 其中  $\tau_{ij}$  表示结点  $c_i$  到  $c_j$  的边上的信息素, 算法最开始会初始信息素向量  $T$ ; 人工蚁群在构建的图  $G$  上游走以构建路径。在路径构建过程中, 蚂蚁将依据一定的概率在未访问过的结点中随机选择下一个访问结点。对于处于结点  $c_i$  的蚂蚁  $k$ , 其选择下一个访问结点  $c_j$  的概率为:

$$p_{ij}^k = \begin{cases} \frac{\tau_{ij}^\alpha \cdot \eta_{ij}^\beta}{\sum_{c_j \in N(s^p)} \tau_{ij}^\alpha \cdot \eta_{ij}^\beta}, & c_j \in N(s^p) \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

其中,  $s^p$  为蚂蚁  $k$  当前已构建的部分路径,  $N(s^p)$  表示当前可访问的结点, 即目前未访问过的结点。  $\eta_{ij}$  为启发式信息, 参数  $\alpha$  和  $\beta$  分别控制信息素和启发信息的权重。

在每次迭代结束之后, 蚁群中的所有蚂蚁都将根据其搜索到的路径更新信息素  $\tau_{ij}$ , 其更新方法如下:

$$\tau_{ij} = (1-\rho) \cdot \tau_{ij} + \sum_{k=1}^m \Delta\tau_{ij}^k \quad (2)$$

其中,  $\rho$  是信息素的挥发率,  $m$  为蚁群规模,  $\Delta\tau_{ij}^k$  是蚂蚁  $k$  在边  $(c_i, c_j)$  上释放的信息素:

$$\Delta\tau_{ij}^k = \begin{cases} 1/L_k, & \text{if ant } k \text{ visited } (c_i, c_j) \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

### 2.2 神经网络

人工神经网络广泛应用于模式识别、数据处理和分类领域, 其中最常见的是前馈神经网络<sup>[11]</sup>。典型的前馈神经网络由若干层组成: 一个输入层、一个输出层、中间一个或多个隐藏层, 各层之间依次连接(见图 1)。

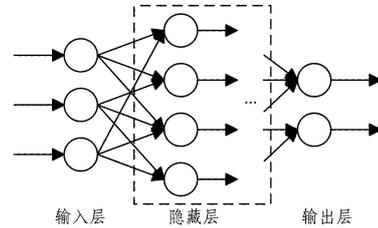


图 1 前馈神经网络的分层结构

尽管神经网络能够实现十分复杂的功能, 但是构成网络的单个神经元的结构却很简单。对于任一神经元  $i$ , 都可以将其视为一个接受  $r$  个输入  $o_1, \dots, o_r$  (整个网络的输入或者前一层神经元的输出) 产生一个输出  $o_i$  的计算单元:

$$net_i = \sum_{j=1}^r w_{ij} o_j + \theta_i \quad (4)$$

$$o_i = f(net_i) \quad (5)$$

其中,  $w_{ij}$  表示神经元  $i$  和神经元  $j$  之间的权重;  $\theta_i$  表示与神经元  $i$  自身相关的权重, 称之为自偏(self-bias);  $f$  是非线性激活函数(Nonlinear Activation Function), 其中最常用的激活函数为逻辑回归函数(Logistic Regression, logsig), 其公式如下<sup>[12]</sup>:

$$f(x) = \text{logsig}(x) = \frac{1}{1 + e^{-x}} \quad (6)$$

在对神经网络进行训练的过程中, 对于训练样本集  $P$ , 将其对网络的输入矢量记为  $x$ , 期望输出矢量记为  $y$ , 而实际输出矢量记为  $y'$ 。可用一个差错函数  $E$  来比较其期望输出  $y$  和实际输出  $y'$  之间的差值, 常用的是均方差方程(MSE):

$$E = \frac{1}{m|P|} \sum_{p \in P} E_p \quad (7)$$

$$E_p = \text{MSE} = \frac{1}{2} \sum_{i=1}^m (y_i - y'_i)^2 \quad (8)$$

其中,  $m$  是训练样本的类数。在 FFNN 中, 由各个神经元的权重和自偏组成的向量被记为网络的权值向量  $w$ 。由式(4)到式(8)可知, 若权值向量  $w$  固定, 则网络的输出是关于输入的函数, 总体误差  $E$  是关于训练集  $P$  的函数; 若训练集固定, 则误差  $E$  是关于权值向量  $w$  的函数。而神经网络的学习过程, 实质上就是通过不断更新权值向量  $w$  使得误差  $E$  最小。

在梯度下降法中, 根据式(9)来更新网络的权值矢量:

$$w_j^{(new)} = w_j^{(old)} + \alpha \left( -\frac{\partial E}{\partial w_i} \right) \quad (9)$$

其中, 梯度信息  $\frac{\partial E}{\partial w_i}$  为总体误差  $E$  关于每个权值矢量的偏导数,  $\alpha$  为网络的学习率。

## 3 $ACO_R$ 算法

Socha K 等将蚁群算法应用于连续域优化, 提出了  $ACO_R$  算法(ACO for real-valued), 并将之成功运用于神经网络训练<sup>[6-7]</sup>。在常规 ACO 算法中, 蚁群在每次迭代中寻找的路径(问题的候选解)在蚁群完成对信息素的更新之后将被丢弃;

而在ACO<sub>R</sub>算法中会记录其中R个解,并将之存放在表T中,其结构如图2所示。

$s_1$	$s_1^1$	...	$s_1^n$	$Q(s_1)$	$w_1$
$s_2$	$s_2^1$	...	$s_2^n$	$Q(s_2)$	$w_2$
...	...	...	...	...	...
$s_R$	$s_R^1$	...	$s_R^n$	$Q(s_R)$	$w_R$

图2 ACO<sub>R</sub>法中用于记录解的表T的结构

表T中的每一个元素 $s_a$ 代表一个已知解,且其都是一个n维的向量 $s_a = (s_a^1, s_a^2, \dots, s_a^n)$ ,其中n为所求解问题的变量个数。例如, $s_{a,j}$ 表示第a个解向量中的第j个值。同时,表T中的解都是根据其质量排序的,即 $Q(s_1) \geq Q(s_2) \geq \dots \geq Q(s_R)$ ,且每一个解 $s_a$ 都会根据其质量 $Q(s_a)$ 对应于一个权值 $w_a$ ,也即 $w_1 \geq w_2 \geq \dots \geq w_R$ 。

类似于ACO算法,ACO<sub>R</sub>算法的每一次迭代可以分为路径构建和信息素更新两个阶段。在路径构建阶段,每只蚂蚁都根据表T依概率选择路径。在算法的初始阶段,会随机生成R个解存放于表T中,其中R是该算法的一个参数;然后在信息素更新阶段,将每一次迭代产生的m(m为蚁群的规模)个新的解附加到表T中,此时表T的规模变成了R+m;之后继续对这R+m个解依据其质量进行排序;最后将排在最后的m个解丢弃,则表T的规模又恢复为R。

ACO<sub>R</sub>算法的核心在路径构建阶段,任一蚂蚁i在构建其路径 $s_i$ 时,都会依据概率从表T中的R条已知路径中选择一条:

$$p(s_a) = \frac{w_a}{\sum_{r=1}^R w_r} \tag{10}$$

由上式可知,路径 $s_a$ 被选择的概率与其权值相关。由图2可知,表T中的R个已知解是按照质量排序的,因此每个解在表中都有一个序号 $a \in (1, R)$ 。在每次迭代过程中,式(10)中使用的权值 $w_a$ 可根据下式计算:

$$w_a = \frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{(a-\mu)^2}{2\sigma^2}} \tag{11}$$

显然,式(11)以是a为变量的正态分布函数,其中 $\mu=1$ 为均值, $\sigma$ 为方差,且有:

$$\sigma = qR \tag{12}$$

其中,q是算法的另一个控制参数,由正态分布函数的性质可知,当q越小时排序靠前的解会获得较大的权值,而q越大时各个解的权值分布越均匀。

假设在某次迭代的路径构建中根据式(10)选出的解是 $s_a$ ,则蚂蚁i将对正态概率密度函数进行(Gaussian PDF)采样(采用Box-Muller法<sup>[13]</sup>),以确定本次迭代的解 $s_i$ 中的第j个变量的值 $s_i^j$ :

$$s_i^j \sim N(\mu_a^j, \sigma_a^j) \tag{13}$$

其中, $N(\mu, \sigma)$ 是正态概率密度函数,均值 $\mu_a^j = s_a^j$ ,方差 $\sigma_a^j$ 由下式计算:

$$\sigma_a^j = \xi \sum_{r=1}^R \frac{|s_a^j - s_r^j|}{R-1} \tag{14}$$

方差 $\sigma_a^j$ 表示的是解 $s_a$ 中第j个变量到表T中其他解的第j个变量之间的平均距离乘以参数 $\xi$ ,其中 $\xi$ 的作用类似于ACO算法中的挥发率, $\xi$ 越大,算法的收敛速度越慢。每次蚂

蚁在完成上述路径构建之后,都会根据前文的方法更新表T。整个算法将一直重复上述过程,直至达到终止条件。

当ACO<sub>R</sub>算法应用于神经网络训练时,其目的是在神经网络的n个神经元中寻找一个权值向量 $w$ ,使得网络的误差E最小。蚁群每次搜寻得到的路径向量 $s_a$ 即为神经网络的权值向量 $w_a$ ,因此后文中对解向量 $s_a$ 和权值向量 $w_a$ 不做区分。当ACO<sub>R</sub>算法用于神经网络训练时,用每次求得的解 $s_a$ 作为权值向量来构建一个神经网络,然后将训练集输入网络中,再根据式(7)和式(8)计算整个网络的均方误差(MSE)。上文所述的表T(见图1)中存放的解的质量可用下式计算:

$$Q(s_a) = 2 \cdot (1 - \text{MSE}) \tag{15}$$

由上式可知, $0 \leq Q(s_a) \leq 11$ 代表解的质量最好,0代表解的质量最差。

### 4 h-ACO<sub>R</sub>算法

ACO<sub>R</sub>算法是适用于连续域的蚁群优化算法,目前对ACO<sub>R</sub>的改进和优化方法有很多。Sampaio等<sup>[14]</sup>将ACO<sub>R</sub>算法应用到通信网络中的资源分配与异常检测中;Karimi A等<sup>[15]</sup>将ACO<sub>R</sub>与禁忌搜索算法相结合后作用于连续多极值函数全局优化;Huang C L等<sup>[16]</sup>将ACO<sub>R</sub>与粒子群算法(PSO)结合后用于数据聚类等。这些改进和应用都是基于不同领域的应用,本文针对BP神经网络训练,也对ACO<sub>R</sub>算法进行了改进。

在传统蚁群算法中,蚂蚁依据两种信息寻找路径:信息素 $\tau_{ij}$ 和启发式信息 $\eta_{ij}$ ,如式(1)所示。在ACO<sub>R</sub>算法中,存放已知解的表T相当于信息素,但是并无相应的启发式信息指导蚁群的活动。在BP神经网络中,最有用的启发式信息是网络的梯度信息。梯度信息能够快速、直接地引导权值的移动,因此基于梯度信息的训练算法是目前训练神经网络最常用的方法。本文提出将神经网络的梯度信息加入ACO<sub>R</sub>算法框架中作为蚁群的启发式信息,以指导蚁群的活动,本文将这种改进的算法称为启发式ACO<sub>R</sub>(heuristic ACO<sub>R</sub>),记作h-ACO<sub>R</sub>。在h-ACO<sub>R</sub>算法中通过计算网络偏差(Error Term)关于神经元权值向量的偏导数来获得网络的梯度向量,记为 $G^{(p)}$ 。本文通过以下几个步骤在ACO<sub>R</sub>算法的基础上实现h-ACO<sub>R</sub>算法。

Step1 ACO<sub>R</sub>算法会在每次迭代中对当前解进行评价,即计算信息素表T中各个解的质量 $Q(s_a)$ , $s_a \in T$ ,并根据质量对T中的各个解向量进行排序。本文提出在计算 $Q(s_a)$ 的同时,增加一个步骤来计算残差 $E_p$ 关于权值向量 $w_a$ 的偏导数,得到梯度向量 $G_a^{(p)}$ :

$$G_a^{(p)} = - \frac{\partial E_p}{\partial w_a} \tag{16}$$

其中, $E_p$ 由式(8)计算得到,且有 $|G^{(p)}| = |w|$ 。由上式可知,从严格意义上讲, $G^{(p)}$ 不是残差梯度向量,而是其相反数,但为了方便,称其为梯度向量。之后可以求得整体的梯度向量 $G_a$ ,并将其也保存在表T中:

$$G_a = \sum_{p \in P} G_a^{(p)} \tag{17}$$

Step2 在ACO<sub>R</sub>算法中,蚁群在某次迭代中构建路径

时,由式(10)根据权值向量从信息素表  $T$  中选择一个解  $s_a \in T$ ,然后蚂蚁  $i$  根据式(13)和式(14)计算本次解  $s_i$  中的第  $j$  个变量的值  $s_i^j$ 。其中,  $s_a$  的作用类似于常规蚁群优化中的信息素  $\tau$ 。本文将由式(16)、式(17)计算出的梯度向量  $G_i^j$  作为蚁群的启发式信息。若  $G_i^j$  为正,则表示网络的误差  $E$  会随着权值向量  $w_i$  中第  $j$  个神经元的权值的增加而减小;若  $G_i^j$  为负,则表示网络的误差  $E$  会随着  $w_i^j$  的值的减小而减小。

由上述分析可知,  $G_i$  的符号能够指导权值向量  $w_i$  的移动方向。因此,本文定义一个集合  $S_a^j$  来存放信息素表  $T$  中满足  $(s_r^j - s_a^j)$  与  $G_i^j$  同符号的解向量  $s_r, s_r \in T$ :

$$S_a^j = \{s_r \mid \text{sgn}(s_r^j - s_a^j) = \text{sgn}(G_i^j)\} \quad (18)$$

其中,  $\text{sgn}$  为符号函数。通过上式,若  $G_i^j$  为正,则  $S_a^j$  中将存放表  $T$  中所有满足  $s_r^j > s_a^j$  的解向量  $s_r$ ;若  $G_i^j$  为负,则  $S_a^j$  中将存放表  $T$  中所有满足  $s_r^j < s_a^j$  的解向量  $s_r$ 。则方差  $\sigma_a^j$  的计算(式(14))改为:

$$\sigma_a^j = \xi \sum_{s_r \in S_a^j} \frac{|s_a^j - s_r^j|}{|S_a^j|} \quad (19)$$

其中,  $|S_a^j|$  表示  $S_a^j$  中元素的个数。上式可以保证蚂蚁在路径构建过程中,权值向量的第  $j$  维只会沿着梯度向量  $G_i$  指示的减小网络误差的方向变化。

Step3 根据梯度信息和式(19)计算方差  $\sigma_a^j$  后,将本次解向量中第  $j$  个元素  $s_i^j$  的求解法(式(13))改为:

$$s \sim N(s_a^j, \sigma_a^j) \quad (20)$$

$$s_i^j = s_a^j + \text{sgn}(G_i^j) \cdot |s - s_a^j| \quad (21)$$

即本次的解向量中  $s_i^j$  的值由  $s_a^j$  向着向量指示的方向移动  $|s - s_a^j|$  得到。本算法的伪代码如下算法 1 所示。

**算法 1** h-ACO<sub>R</sub> 算法

1. Begin
2. 设置参数,随机生成信息素表;
3. 计算表  $T$  中每个解的质量  $Q(s_i)$  和梯度向量  $G_i$ ;
4. 将表  $T$  中的解按其质量  $Q(s_i)$  排序;
5. while 不满足终止条件 do
6. for 蚁群中每个蚂蚁 do
7. 依概率从表  $T$  中选择解  $s_a$
8. for 每个解构造步
9. 根据梯度向量  $G_i$  计算  $S_a^j$
10. 计算方差  $\sigma_a^j$
11. 计算路径的每一个解  $s_i^j$
12. end for
13. 将本只蚂蚁求解的路径放入表  $T$  中
14. end for
15. 计算表  $T$  中每个解的质量  $Q(s_i)$  和梯度向量  $G_i$ ;
16. 将表  $T$  中的解按其质量  $Q(s_i)$  排序;
17. 移除表  $T$  中质量最差的  $m$  个解
18. end while
19. End

**5 实验方法及结果分析**

**5.1 实验方法**

由于 h-ACO<sub>R</sub> 算法是基于 ACO<sub>R</sub> 算法的改进,因此本文

将通过 h-ACO<sub>R</sub> 与 ACO<sub>R</sub> 的对比实验来对其进行评价。本文实验所选用的数据集来自加州大学欧文分校(University of California Irvine, UCI)提出的用于机器学习的数据库——UCI 机器学习库,其中包含 350 多个数据集,是机器学习及模式识别领域常用的标准测试数据集<sup>[17]</sup>。

本文选择了 UCI 数据集中 3 组常用的数据集 zoo, iris 及 tic-tac-toe,分别利用 h-ACO<sub>R</sub> 和 ACO<sub>R</sub> 两种算法对前馈多层感知神经网络进行训练。神经网络的输入层和输出层神经元个数根据数据而定,其中输出层神经元个数与数据集的类数相同,即某个数据集有  $m$  个类,则神经网络的输出层设置为  $m$  个神经元,其输出为一个  $m$  维的向量。为了将网络的实际输出与样本的目标输出进行对比验证,将样本的目标输出  $y$  也映射为  $m$  维的向量:将  $m$  个类编号,对于编号为  $c$  的类,将目标输出向量中第  $k=c$  个值设为 1,其余设为 0,即:

$$y_k = \begin{cases} 1, & \text{if } k=c \\ 0, & \text{otherwise} \end{cases} \quad (22)$$

网络的输入层由数据集的特征属性决定。每个数据集的特征一般分为两类:数字特征和类别特征。数字特征是范围为  $[0, 1]$  的连续量;类别特征类似于目标输出,通过式(22)将每个有  $m$  个类别的特征变量映射成网络的  $m$  个输入。因此,网络的输入层神经元个数为其数字特征数加上类别特征数乘以每个特征类数之和。本文采用 3 层网络结构,并且将网络的隐藏层神经元个数设置为输入层和输出层之和。表 1 列出了本文实验所用的数据集及其对应的神经网络结构。

表 1 数据集信息及其相关的网络拓扑结构

数据集	样本数	类数	特征			网络结构		
			总数	数字	类别	输入层	隐藏层	输出层
zoo	101	7	16	0	16	36	43	7
iris	150	3	4	4	0	4	7	3
tic-tac-toe	958	2	9	0	9	27	29	2

本文在实验过程中对算法的参数设定请参考文献[18]中对 ACO<sub>R</sub> 算法进行评价时的参数设置,其设置的参数信息如表 2 所列。

表 2 算法的参数设置

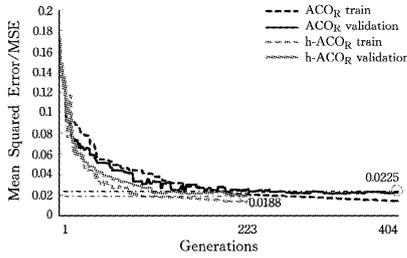
参数	符号	取值
蚂蚁数量	$m$	2
解集 $T$ 的大小	$R$	50
挥发率	$\xi$	0.85
控制权值分布	$q$	0.05

本文采用十折交叉验证法<sup>[8]</sup>来进行实验。十折交叉验证法是标准的数据处理方法,其将数据集分为 10 份,轮流将其其中 9 份作为训练数据,另外 1 份作为测试数据来进行实验。每次实验都会得出相应的正确率,然后将 10 次正确率的平均值作为对算法性能的估计。在训练过程中,两种算法的迭代次数达到设定最大值(本文设置为 5000 次)或者 50 次迭代未提高解的质量时训练结束。

**5.2 结果分析**

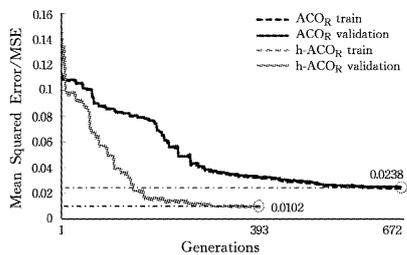
分别用 h-ACO<sub>R</sub> 算法和 ACO<sub>R</sub> 算法对表 1 所列 3 组数据及其对应的网络进行训练。图 3—图 5 分别记录两种算法训练 zoo, iris 和 tic-tac-toe 时,其训练和验证误差随训练代数的

变化曲线。其中,训练误差(Train Error)用虚线表示,验证误差(Validation Error)用实线表示;黑色线代表 ACO<sub>R</sub> 算法,灰色线代表 h-ACO<sub>R</sub> 算法。



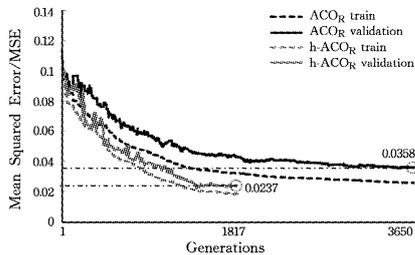
注:其中,ACO<sub>R</sub> 算法收敛于第 404 代,最小验证误差为 0.0225;  
h-ACO<sub>R</sub> 算法收敛于第 223 代,最小验证误差为 0.0188

图 3 2 种算法针对 zoo 的训练及验证过程误差(MSE)的下降曲线



注:其中,ACO<sub>R</sub> 算法收敛于第 672 代,最小验证误差为 0.0238;  
h-ACO<sub>R</sub> 算法收敛于第 393 代,最小验证误差为 0.0102

图 4 2 种算法针对 iris 的训练及验证过程误差(MSE)的下降曲线



注:其中,ACO<sub>R</sub> 算法收敛于第 3650 代,最小验证误差为 0.0358;  
h-ACO<sub>R</sub> 算法收敛于第 1817 代,最小验证误差为 0.0237

图 5 2 种算法针对 tic-tac-toe 的训练及验证过程误差(MSE)的下降曲线

由图 3—图 5 可以看出,h-ACO<sub>R</sub> 算法和 ACO<sub>R</sub> 算法的误差的下降趋势都随着训练代数的增加而逐渐变缓,但是整体而言,h-ACO<sub>R</sub> 算法的误差的下降趋势较 ACO<sub>R</sub> 更快。实验的整体结果如表 3 所列。对于每一组数据集,表 3 列出了其在两种训练算法下的训练代数(Generations)、测试集的最小均方误差(MSE)和准确率(Accuracy),其中采用数据加粗的方式来表示 MSE、Generations 较小数值和 Accuracy 较高的数值。

表 3 3 组数据实验的训练代数、最小均方误差(MSE)和准确率

数据集	训练代数		最小均方误差		准确率/%	
	ACO <sub>R</sub>	h-ACO <sub>R</sub>	ACO <sub>R</sub>	h-ACO <sub>R</sub>	ACO <sub>R</sub>	h-ACO <sub>R</sub>
zoo	404	<b>223</b>	0.0225	<b>0.0188</b>	83.1	<b>91.1</b>
iris	672	<b>393</b>	0.0238	<b>0.0102</b>	88.7	<b>93.3</b>
tic-tac-toe	3650	<b>1817</b>	0.0358	<b>0.0237</b>	91.6	<b>95.9</b>
Total/ means	4726	<b>2433</b>	0.0273	<b>0.0176</b>	87.8	<b>93.4</b>

由表 3 中的实验结果可知,h-ACO<sub>R</sub> 算法的收敛速度明显快于 ACO<sub>R</sub> 算法,其收敛的代数约为 ACO<sub>R</sub> 的 1/2;同时,h-ACO<sub>R</sub> 算法能够减小网络的误差,其平均值为 0.0176,优于 ACO<sub>R</sub> 算法的 0.0273;经 h-ACO<sub>R</sub> 算法训练的网络,对 3 组数据的分类准确率平均为 93.4%,高于 ACO<sub>R</sub> 算法的 87.8%。上述实验结果表明,本文提出的 h-ACO<sub>R</sub> 算法较 ACO<sub>R</sub> 算法有更好的性能表现,能够有效地应用于神经网络训练中。

**结束语** 本文在连续域蚁群优化算法(ACO<sub>R</sub>)的基础上,提出了一种带有启发式信息的神经网络学习算法(h-ACO<sub>R</sub>),其启发式信息是通过计算神经网络的误差(error term)关于网络的权值向量的偏导数而得到的梯度向量。蚁群根据启发式信息寻找最优路径,能够有效减少训练代数,提高训练准确率。本文提出的 h-ACO<sub>R</sub> 算法会在 ACO<sub>R</sub> 算法的基础上增加一个计算和存储梯度向量的过程,虽然能有效地减少训练的代数,但是训练过程中的空间需求会相应变大,且训练所花费的 CPU 时间的减少还有待进一步的实验验证。

### 参考文献

- [1] HAYKIN S. Neural Networks and Learning Machines(3rd ed)[M]. New York, NY, USA: Pearson Education, 2009.
- [2] HORNIK K, STINCHCOMBE M, WHITE H. Multilayer feed-forward networks are universal approximators[J]. Neural Networks, 1989, 2(5): 359-366.
- [3] RUMELHART D, DHINTON G, WILLIAMS J. Learning representations by back propagating errors[J]. Nature, 1986, 323(6088): 533-536.
- [4] GAO H B, GAO L, ZHOU C, et al. Particle Swarm Optimization Based Algorithm for Neural Network Learning[J]. Acta Electronica Sinica, 2004, 32(9): 1572-1574. (in Chinese)  
高海兵, 高亮, 周驰, 等. 基于粒子群优化的神经网络训练算法研究[J]. 电子学报, 2004, 32(9): 1572-1574.
- [5] DORIGO M, BIRATTARI M, STITZLE T. Ant Colony Optimization: Artificial Ants as a Computational Intelligence Technique [J]. IEEE Computational Intelligence Magazine, 2006, 1(4): 28-39.
- [6] SOCHA K, DORIGO M. Ant Optimization for Continuous Domains[J]. European Journal of Operational Research, 2008, 185: 1155-1173.
- [7] SOCHA K, BLUM C. An ant Colony Optimization Algorithm for Continuous Optimization: Application to Feed-Forward Neural Network Training[J]. Neural Computing & Applications, 2007, 16(3): 235-247.
- [8] WRITTEN I H, FRANK E. Data Mining-Practical Machine Learning Tools and Techniques(3rd ed)[M]. Morgan Kaufmann Publishers, 2012.
- [9] LI Y M, WANG W J, XU Z B. About ACO Algorithm For Hard Combinatorial Optimization Problem [J]. Computer Science, 2002, 29(3): 115-118. (in Chinese)  
李有梅, 王文剑, 徐宗本. 关于求解难组合优化问题的蚁群优化算法[J]. 计算机科学, 2002, 29(3): 115-118.

- [3] XIA R, XU F, ZONG C Q, et al. Dual Sentiment Analysis: Considering Two Sides of One Review [J]. *IEEE Transactions on Knowledge and Data Engineering*, 2015, 27(8): 2120-2133.
- [4] YU Z, WANG H X, LIN X M, et al. Understanding Short Texts through Semantic Enrichment and Hashing [J]. *IEEE Transactions on Knowledge and Data Engineering*, 2016, 28(2): 566-579.
- [5] AZAM N, YAO J. Comparison of term frequency and document frequency based feature selection metrics in text categorization [J]. *Expert Syst. Appl.*, 2012, 39(5): 4760-4768.
- [6] DUAN J, HU Q H, ZHANG L J, et al. Feature Selection for Multi-Label Classification Based on Neighborhood Rough Set [J]. *Journal of Computer Research and Development*, 2015, 52(1): 56-65. (in Chinese)  
段洁, 胡清华, 张灵均, 等. 基于邻域粗糙集的多标记分类特征选择算法 [J]. *计算机研究与发展*, 2015, 52(1): 56-65.
- [7] TANG B, KAY S, HE H B. Toward Optimal Feature Selection in Naïve Bayes for Text Categorization [J]. *IEEE Transactions on Knowledge and Data Engineering*, 2016, 28(9): 2508-2521.
- [8] TANG B, HE H B, BAGGENSTOSS P M, et al. A Bayesian Classification Approach Using Class-Specific Features for Text Categorization [J]. *IEEE Transactions Knowledge and Data Engineering*, 2016, 28(6): 1602-1606.
- [9] CHENG V C, LEUNG C H C, LIU J M, et al. Probabilistic Aspect Mining Model for Drug Reviews [J]. *IEEE Transactions on Knowledge and Data Engineering*, 2014, 26(8): 2002-2013.
- [10] TANG B, HE H, DING D, et al. A parametric classification rule based on the exponentially embedded family [J]. *IEEE Transactions on Neural Networks and Learning Systems*, 2015, 26(2): 367-377.
- [11] LIU S H, CHENG X Q, LI F X, et al. TASC: Topic-Adaptive Sentiment Classification on Dynamic Tweets [J]. *IEEE Transactions on Knowledge and Data Engineering*, 2015, 27(6), 1696-1709.
- [12] SUN K W, LEE C H, WANG J. Multilabel Classification via Co-Evolutionary Multilabel Hypernetwork [J]. *IEEE Transactions on Knowledge and Data Engineering*, 2016, 28(9): 2438-2451.
- [13] ZHANG T, TANG Y Y, FANG B, et al. Document clustering in correlation similarity measure space [J]. *IEEE Transactions on Knowledge and Data Engineering*, 2012, 24(6): 1002-1013.
- [14] MORI U, MENDIBURI A, LOZANO J A. Similarity Measure Selection for Clustering Time Series Databases [J]. *IEEE Transactions on Knowledge and Data Engineering*, 2016, 28(1): 181-195.
- [15] KANG Y B, HAGHIGH P D, BURSTEIN F. TaxoFinder: A Graph-Based Approach for Taxonomy Learning [J]. *IEEE Transactions on Knowledge and Data Engineering*, 2016, 28(2): 524-536.
- [16] WANG Q, CUI M Y, LIANG H Z. Semantic-Aware Blocking for Entity Resolution [J]. *IEEE Transactions on Knowledge and Data Engineering*, 2016, 28(1): 166-180.
- [17] MCCALLUM A, NIGAM K, et al. A comparison of event models for naive bayes text classification [C]//AAAI-98 Workshop on Learning for Text Categorization. 1998: 41-48.
- [18] SCHOENHARL T W, MADEY G. Evaluation of measurement techniques for the validation of agent-based simulations against streaming data [C]//Proc. ICCS. Kraków, Poland, 2008: 6-15.
- [19] STREHL A, GHOSH J. Value-based customer grouping from large retail data-sets [C]//Proc. SPIE. Orlando, FL, USA, 2000: 33-42.
- [20] BISWAS S K, MILANFAR P. One Shot Detection with Laplacian Object and Fast Matrix Cosine Similarity [J]. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2016, 38(3): 546-562.

(上接第 288 页)

- [10] XIA Y M, CHENG B, CHEN J L, et al. Optimizing Services Composition Based on Improved Ant Colony Algorithm [J]. *Chinese Journal of Computers*, 2012, 35(2): 270-281. (in Chinese)  
夏亚梅, 程渤, 陈俊亮, 等. 基于改进蚁群算法的服务组合优化 [J]. *计算机学报*, 2012, 35(2): 270-281.
- [11] RAZAVI S, TOLSON B A. A New Formulation for Feedforward Neural Networks [J]. *IEEE Transactions on Neural Networks*, 2011, 22(10): 1588-1598.
- [12] WU X, WILAMOWSKI B M. Advantage Analysis of Sigmoid Based RBF Networks [C]//IEEE 17th International Conference on Intelligent Engineering Systems. Costa Rica, 2013.
- [13] MULLER B. A note on the generation of random normal deviates [J]. *Annals of Mathematical Statistics*, 1958, 29: 610-611.
- [14] SAMPAIO L H D, ADANIYA M H A C, DE PAULA MARQUES M, et al. Ant colony optimization for resource allocation and anomaly detection in communication networks [M]. IN-TECH Open Access Publisher, 2013.
- [15] KARIMI A, NOBAHARI H, SIARRY P. Continuous ant colony system and tabu search algorithms hybridized for global minimization of continuous multi-minima functions [J]. *Computational Optimization and Applications*, 2010, 45(3): 639-661.
- [16] HUANG C L, HUANG W C, CHANG H Y, et al. Hybridization strategies for continuous ant colony optimization and particle swarm optimization applied to data clustering [J]. *Applied Soft Computing*, 2013, 13(9): 3864-3872.
- [17] BACHE K, LICHMAN M. UCI machine learning repository [OL]. URL: <http://archive.ics.uci.edu/ml>, 2013.
- [18] LIAO T, DORIGO M. Ant colony optimization for mixed-variable optimization problems [J]. *IEEE Transactions on Evolutionary Computation*, 2014, 18(4): 503-518.