

一种基于互信息变量选择的极端学习机算法

韩敏, 刘晓欣

(大连理工大学 电子信息与电气工程学部, 辽宁 大连 116023)

摘要: 针对回归问题中存在的变量选择和网络结构设计问题, 提出一种基于互信息的极端学习机(ELM)训练算法, 同时实现输入变量的选择和隐含层的结构优化. 该算法将互信息输入变量选择嵌入到ELM网络的学习过程中, 以网络的学习性能作为衡量输入变量与输出变量相关与否的指标, 并以增量式的方法确定隐含层节点的规模. 在Lorenz、Gas Furnace和10组标杆数据上的仿真结果表明了所提出算法的有效性. 该算法不仅可以简化网络结构, 还可以提高网络的泛化性能.

关键词: 极端学习机; 变量选择; 互信息; 回归分析

中图分类号: TP183

文献标志码: A

An extreme learning machine algorithm based on mutual information variable selection

HAN Min, LIU Xiao-xin

(Faculty of Electronic Information and Electrical Engineering, Dalian University of Technology, Dalian 116023, China.

Correspondent: HAN Min, E-mail: minhan@dlut.edu.cn)

Abstract: To solve the problems of variable selection and architecture design in regression, an extreme learning machine (ELM) based on mutual information is proposed, which can optimize the input layer and the hidden layer simultaneously. The mutual information variable selection is combined with ELM. The performance of the network is used as the criterion of variable selection, and the size of the hidden layer is determined by using the incremental method. Simulation results on two data sets of multivariate time series and 10 benchmark datasets show the effectiveness of the proposed algorithm. The proposed algorithm can not only compact the architecture of the network, but also improve the generalization performance.

Key words: extreme learning machine; variable selection; mutual information; regression analysis

0 引言

极端学习机(ELM)是近年发展起来的一种新型单隐层前馈神经网络^[1-2]. 因其具有结构简单、学习速度快和泛化能力好等优点, 目前在模式识别和回归估计等领域得到了广泛的应用^[3]. ELM的网络结构为3层, 包括1个输入层, 1个隐含层和1个输出层, 其中隐含层包含大量的非线性节点. 传统前馈神经网络采用梯度训练算法, 收敛速度慢, 容易陷入局部最优. ELM克服了传统前馈神经网络的缺陷, 在训练的初始阶段随机产生输入权值和偏置, 并在训练过程中保持不变, 网络的输出权值是唯一需要确定的参数.

已有研究表明, ELM的学习精度和泛化能力在很大程度上取决于隐层神经元的数量^[1], 因此在设计

ELM网络时, 如何确定最优的网络结构一直都是研究的重点. 进化极端学习机(E-ELM)通过引入差分进化(DE)算法来优化输入权值和隐层节点的偏置, 将ELM与DE算法结合起来, 从而得到更加紧凑的网络结构和更快的网络响应^[4]. 最优剪枝极端学习机(OP-ELM)在原始ELM的基础上, 基于剪枝思想自动确定网络结构, 该算法首先对隐层节点进行排序, 然后通过留一法选择具有最优隐层节点的模型^[5].

除网络结构之外, 输入变量能否准确涵盖输出变量的有效信息, 也是决定ELM网络性能的一个重要因素, 特别是对于数据多样和相互作用关系复杂的多变量数据. 因此, 对多元数据进行变量选择是十分必要的. 常用的变量选择方法有主成分分析(PCA)^[6]、

收稿日期: 2013-01-21; 修回日期: 2013-03-22.

基金项目: 国家自然科学基金项目(61074096).

作者简介: 韩敏(1959—), 女, 教授, 博士生导师, 从事神经网络、3S系统及混沌序列分析等研究; 刘晓欣(1987—), 女, 硕士生, 从事多元时间序列相关性分析与变量选择的研究.

独立成分分析(ICA)^[7]、典型相关分析(CCA)^[8]、Granger因果关系分析^[9]、灰色关联分析(GRA)^[10]和互信息(MI)^[11]等.其中,互信息既能描述线性相关关系,又能描述非线性相关关系,并且对数据的分布类型没有要求,所以本文采用互信息作为输入变量选择的方法.

针对回归问题中存在的变量选择和网络结构设计问题,本文提出一种基于互信息的ELM训练算法(MI-ELM),将互信息变量选择嵌入到ELM模型的训练过程之中,同时完成输入变量选择和ELM模型隐层节点个数的确定.所提算法的有效性将在Lorenz、Gas Furnace和10组标杆数据上通过仿真实验验证.

1 极端学习机的实现机理

ELM^[1]网络的实现机理为:考虑有 N 个训练样本 $\{\mathbf{u}_i, \mathbf{t}_i\}, i=1, 2, \dots, N$.其中:输入向量 $\mathbf{u}_i = [u_{i1}, u_{i2}, \dots, u_{ip}]^T \in R^p$,输出向量 $\mathbf{t}_i = [t_{i1}, t_{i2}, \dots, t_{iq}]^T \in R^q$.设ELM共有 m 个隐含层节点,即网络规模为 m ,激活函数为 $g(\cdot)$,网络输出为 \mathbf{y} ,则输入-输出关系为

$$\mathbf{y}_i = \sum_{j=1}^m \beta_j g(\mathbf{w}_j^T \mathbf{u}_i + b_j), i=1, 2, \dots, N. \quad (1)$$

其中: $\mathbf{w}_j = [w_{j1}, w_{j2}, \dots, w_{jp}]^T$ 为连接第 j 个隐含层节点和输入节点的权值; $\beta_j = [\beta_{j1}, \beta_{j2}, \dots, \beta_{jq}]^T$ 为连接第 j 个隐层节点和输出节点的权值; b_j 为第 j 个隐含层节点的偏置.将式(1)表示为矩阵相乘的形式为

$$\mathbf{Y} = \mathbf{H}\beta, \quad (2)$$

$$\mathbf{H}(\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_m, b_1, b_2, \dots, b_m) = \begin{bmatrix} g(\mathbf{w}_1^T \mathbf{u}_1 + b_1) & \cdots & g(\mathbf{w}_m^T \mathbf{u}_1 + b_m) \\ \vdots & \ddots & \vdots \\ g(\mathbf{w}_1^T \mathbf{u}_N + b_1) & \cdots & g(\mathbf{w}_m^T \mathbf{u}_N + b_m) \end{bmatrix}_{N \times m}, \quad (3)$$

$$\beta = \begin{bmatrix} \beta_1^T \\ \beta_2^T \\ \vdots \\ \beta_m^T \end{bmatrix}_{m \times q}, \mathbf{Y} = \begin{bmatrix} \mathbf{y}_1^T \\ \mathbf{y}_2^T \\ \vdots \\ \mathbf{y}_N^T \end{bmatrix}_{N \times q}, \quad (4)$$

其中 \mathbf{H} 为ELM的状态矩阵.

与传统的前馈神经网络不同,在ELM中,输入权值向量 \mathbf{w}_j 和隐含层偏置值 b_j 在训练过程中是不需要调整的,即总存在一组 (\mathbf{w}_j, b_j) ,使得ELM网络的状态矩阵 \mathbf{H} 是可逆的,而且是 $\|\mathbf{H}\beta - \mathbf{T}\| = 0$ 的解.

ELM的训练过程一般如下.

Step 1: 初始化网络参数,随机生成权值向量 \mathbf{w}_j 和隐含层偏置值 b_j .

Step 2: 根据输入变量,通过隐含层网络映射得到

相应的状态矩阵 \mathbf{H} .

Step 3: 采用伪逆方法求得输出权值向量 $\beta = \mathbf{H}^\dagger \mathbf{T}$,伪逆解的具体求法如下:

对状态矩阵 \mathbf{H} 进行奇异值分解(假设 $\text{rank}(\mathbf{H}) = r$),可得

$$\mathbf{H} = \mathbf{U}\Sigma\mathbf{V}^T. \quad (5)$$

其中:矩阵 $\mathbf{U} \in R^{N \times r}$ 和矩阵 $\mathbf{V} \in R^{m \times r}$ 均为酉阵, $\Sigma = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_r)$,且有

$$\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r \geq 0. \quad (6)$$

基于矩阵 \mathbf{H} 的奇异值分解,其伪逆可以表示为

$$\mathbf{H}^\dagger = \mathbf{V}\Sigma^{-1}\mathbf{U}^T \in R^{m \times N}. \quad (7)$$

当 $\text{rank}(\mathbf{H}) = r = N$ 时, $\mathbf{H}^\dagger = (\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T$;而当 $\text{rank}(\mathbf{H}) = r = m$ 时, $\mathbf{H}^\dagger = \mathbf{H}^T (\mathbf{H} \mathbf{H}^T)^{-1}$.此外,当 \mathbf{H} 为方阵且非奇异时, $\mathbf{H}^\dagger = \mathbf{H}^{-1}$.此时,ELM的输出权值可由状态矩阵 \mathbf{H} 的伪逆表示为

$$\beta = \mathbf{H}^\dagger \mathbf{T}. \quad (8)$$

2 基于互信息的ELM模型

变量 X 与 Y 之间的互信息定义^[11]如下所示:

$$I(X; Y) = \iint \mu(x, y) \log \frac{\mu(x, y)}{\mu_X(x)\mu_Y(y)} dx dy. \quad (9)$$

其中: $\mu(x, y)$ 表示变量 X 和 Y 的联合概率密度, $\mu_X(x)$ 和 $\mu_Y(y)$ 分别表示 X 和 Y 的边缘概率密度.互信息越大,变量 X 包含关于 Y 的信息越多,即两个变量的相关性越大.因此,可将互信息输入变量选择与ELM模型结合,在网络的训练过程中完成输入节点与隐层节点的选择.设 M 维输入变量为 $\mathbf{X} = [X_1, X_2, \dots, X_M]$,输出变量为 Y ,样本数为 N ,则基于互信息的ELM训练算法的具体步骤可描述如下.

Step 1: 初始化ELM.设置ELM的类型(分类或回归)、激活函数类型和网络初始规模.

Step 2: 选择相关输入变量.计算每一维输入变量 X_i 与输出变量 Y 之间的互信息 $I(X_i; Y)$,选择满足 $I(X_i; Y) > \alpha$ 的输入变量 X_i ,将其按互信息从大到小排序,并组成相关变量集合 F ,其中 α 通常可取 $1/M$ ^[12].

Step 3: 选择输入节点.选择 F 中的第1个变量作为输入节点,并在 F 中将该变量删除.

Step 4: 训练ELM模型.

Step 5: 判定是否满足停止条件.以校验集上的测试精度作为停止条件,若测试误差小于预先设定的阈值,或者分类精度大于预先设定的阈值,则算法停止,此时的模型即为最终模型,否则,算法继续.

Step 6: 变量贡献度.计算当前迭代过程中新增加变量的贡献度,贡献度由变量增加前后模型的测试精

度的变化量来表示. 对于回归问题, 该变化量为前后 2 个模型的误差之差; 对于分类问题, 该变化量为前后 2 个模型的分精度之差. 若变化量大于阈值 β (本文仿真实验中 $\beta = 0.001$), 则认为新增的变量在模型中是有意义的, 可以继续增加变量, 转至 Step 3, 否则, 停止增加变量, 转至 Step 7.

Step 7: 添加隐含层节点. 由于本文算法采用增量式的方法逐步增加隐含层节点个数, 网络初始规模通常设置得较小. 算法每次迭代添加隐含层节点时, 可以设置步长, 即每次可以只添加一个隐含层节点, 也可以添加一组隐含层节点. 隐含层节点添加之后转至 Step 4.

3 仿真实验

为了验证所提出算法的有效性, 将 MI-ELM 算法分别应用于 Lorenz 和 Gas Furnace^[13] 两组多元时间序列和 10 组多变量标杆数据之中. 实验环境为 Windows XP 系统, Pentium 双核 CPU, 主频 2.93 GHz, 1.96 GB 内存. 仿真实验中采用的模型性能评价指标主要包括程序运行 50 次的平均均方根误差 (RMSE) 和训练时间.

3.1 时间序列数据仿真结果

Lorenz 序列的方程如下所示:

$$\begin{cases} dx/dt = a(-x + y), \\ dy/dt = bx - y - xz, \\ dz/dt = xy - cz. \end{cases} \quad (10)$$

当 $a = 10$, $b = 28$, $c = 8/3$, 初始值 $x(0) = 12$, $y(0) = 2$, $z(0) = 9$ 时, 方程表现出混沌特性. 利用四阶 Runge-Kutta 法求解 Lorenz 时间序列 $x(t)$, $y(t)$, $z(t)$, 步长为 0.02, 取 2 000 组输入输出样本对, 其中 1 500

组数据用于训练, 其余 500 组用于测试. 首先对 $x(t)$, $y(t)$, $z(t)$ 三个序列进行相空间重构, 嵌入维数设为 6, 延迟时间分别设置为 8、7 和 8, 由此得到 18 维输入变量为

$$\begin{aligned} \mathbf{X}(t) = & [x(t), x(t-8), \dots, x(t-40), y(t), y(t-7), \\ & \dots, y(t-35), z(t), z(t-8), \dots, z(t-40)]. \end{aligned} \quad (11)$$

单步预测的输出为

$$\mathbf{Y}(t) = [x(t+1), y(t+1), z(t+1)]. \quad (12)$$

当输出变量分别为 $x(t+1)$, $y(t+1)$ 和 $z(t+1)$ 时, 将本文算法与几种常用 ELM 模型和 RBF 网络进行比较, 可得到如表 1 所示的仿真结果. 其中, ELM 和 E-ELM 的隐层节点个数均设为 100, RBF 网络隐层节点的最大规模设为 100.

由表 1 可以看出, 4 种 ELM 模型的预测精度均高于 RBF 网络. MI-ELM 和 OP-ELM 两种算法可以同时选择输入变量和隐层节点, 虽然选择的输入变量与隐层节点的规模不同, 但都得到了小于原始 18 维输入变量所对应的 ELM 和 E-ELM 模型的预测误差. MI-ELM 在 $x(t+1)$, $y(t+1)$ 和 $z(t+1)$ 三组输出变量上均得到了最小的预测误差. 当输出变量为 $x(t+1)$ 时, MI-ELM 不仅表现出了最好的泛化能力, 而且大大简化了模型的规模.

由于结构优化的引入势必会增加算法的计算复杂度, 在训练时间和测试时间上, E-ELM、OP-ELM 和 MI-ELM 三种算法均大于 ELM 算法. 在 3 种采用结构优化策略的算法中, MI-ELM 算法在 3 组输出变量上的训练时间都是最小的, 同时也明显快于 RBF 网络, 说明了本文方法的快速性.

表 1 Lorenz 仿真结果比较

输出变量	建模方法	输入变量个数	隐层节点数	测试 RMSE		训练时间/s
				Mean	Dev.	
$x(t+1)$	ELM ^[1]	18.000 0	100.000 0	0.005 1	0.002 3	0.112 8
	E-ELM ^[4]	18.000 0	100.000 0	0.004 9	0.001 3	429.475 0
	RBF	18.000 0	100.000 0	0.012 6	0.000 0	23.609 4
	OP-ELM ^[5]	2.666 7	44.444 4	0.005 0	0.001 7	14.177 1
	MI-ELM	2.000 0	6.350 0	0.003 9	0.001 9	0.962 5
$y(t+1)$	ELM ^[1]	18.000 0	100.000 0	0.005 9	0.002 2	0.113 4
	E-ELM ^[4]	18.000 0	100.000 0	0.007 2	0.002 9	427.500 0
	RBF	18.000 0	100.000 0	0.019 1	0.000 0	23.671 9
	OP-ELM ^[5]	3.428 6	42.142 9	0.007 1	0.004 4	14.966 5
	MI-ELM	4.000 0	32.600 0	0.001 4	0.000 4	12.117 2
$z(t+1)$	ELM ^[1]	18.000 0	100.000 0	0.011 3	0.005 3	0.110 9
	E-ELM ^[4]	18.000 0	100.000 0	0.012 1	0.003 6	429.037 5
	RBF	18.000 0	100.000 0	0.018 5	0.000 0	24.898 5
	OP-ELM ^[5]	2.000 0	44.375 0	0.004 4	0.000 8	15.293 0
	MI-ELM	2.000 0	50.000 0	0.002 6	0.000 1	14.831 3

Gas Furnace^[13]是常用的一组多元时间序列分析数据, 输入为气体速率 $u(t)$, 输出为 CO_2 百分比浓度 $y(t)$. 对 $u(t)$ 和 $y(t)$ 序列进行相空间重构, 延迟时间设为 1, 嵌入维数分别设为 6 和 4, 得到 10 维输入变量为

$$\mathbf{X} = [u(t-6), \dots, u(t-1), y(t-4), \dots, y(t-1)]. \quad (13)$$

单步预测的输出为

$$Y = y(t). \quad (14)$$

相空间重构之后共得到输入输出样本对 291 组, 其中 219 组用于训练, 其余 72 组用于测试. Gas Furnace 多元时间序列的仿真结果如表 2 所示.

表 2 Gas Furnace 仿真结果比较

建模方法	输入变量 个数	隐层 节点数	测试 RMSE		训练 时间/s
			Mean	Dev.	
ELM ^[1]	10.000 0	30.000 0	0.070 9	0.008 7	0.006 3
E-ELM ^[4]	10.000 0	30.000 0	0.082 4	0.007 8	17.829 7
RBF	10.000 0	30.000 0	0.124 5	0.000 0	3.375 0
OP-ELM ^[5]	5.875 0	27.750 0	0.077 1	0.015 5	1.326 2
MI-ELM	8.000 0	22.050 0	0.070 8	0.005 7	2.760 9

由表 2 可知, MI-ELM 模型在 Gas Furnace 数据上得到了与 ELM 接近的预测精度, RBF 网络的预测精度最低, OP-ELM 和 E-ELM 的预测误差比 ELM 略大些. 这可能是由于 OP-ELM 选择的输入变量维度过低, 从而损失了部分数据信息. 在训练时间上, 显然 ELM 是最快速的, MI-ELM 算法和 OP-ELM 与 RBF 网络比较接近, 在计算复杂度上均优于 E-ELM.

3.2 标杆数据仿真结果

采用 10 组标杆数据进行仿真实验, 如表 3 所示. 每次仿真实验中随机抽取 2/3 的样本用于训练, 其余 1/3 的样本用于测试. 根据文献[1], 将输入数据归一化到区间 $[-1, 1]$, 相应的输出数据归一化至 $[0, 1]$ 之间. 4 种方法中, ELM 网络均选择 Sigmoid 函数作为激活函数. 表 4 比较了 5 种模型的测试误差, MI-ELM 在 4 组数据上得到了最小的测试误差, 在 3 组数据上得到了次优的结果, 说明 MI-ELM 同样适用于多变量数据.

表 5 比较了 MI-ELM 与 OP-ELM 两种算法所得到的模型规模, 可以看出, 2 种算法在输入节点和隐层节点的个数上结果比较接近, 均有效简化了网络的规模. 在 Bank 数据集上, MI-ELM 模型的优势更为明显, 网络的输入节点个数和隐层节点个数的平均值仅为 3.6 和 23.5, 远远小于 OP-ELM.

表 3 各组标杆数据的具体描述

数据集名	训练样本个数	测试样本个数	输入变量个数
Abalone	2 784	1 393	8
Delta ailerons	4 752	2 377	5
Delta elevators	6 344	3 173	6
Computer activity	5 461	2 731	12
Auto price	106	53	15
CPU	139	70	6
Servo	111	56	4
Breast cancer	129	65	32
Bank	5 461	2 731	8
Stocks	633	317	9

表 4 各组标杆数据的测试 RMSE 仿真结果比较

数据集名	ELM ^[1]	E-ELM ^[4]	RBF	OP-ELM ^[5]	MI-ELM
Abalone	0.077 5 ± 0.002 3	0.077 0 ± 0.003 5	0.076 2 ± 0.002 7	0.079 6 ± 0.003 4	0.074 8 ± 0.003 2
Delta ailerons	0.038 8 ± 0.000 8	0.039 1 ± 0.001 7	0.038 8 ± 0.000 8	0.040 8 ± 0.004 3	0.038 8 ± 0.001 2
Delta elevators	0.053 2 ± 0.000 7	0.054 6 ± 0.001 2	0.053 0 ± 0.000 7	0.053 2 ± 0.000 7	0.053 6 ± 0.000 9
Computer activity	0.036 8 ± 0.005 9	0.046 9 ± 0.031 3	0.036 0 ± 0.001 7	0.055 0 ± 0.028 1	0.050 8 ± 0.008 3
Auto price	0.098 2 ± 0.017 1	0.087 5 ± 0.027 6	0.121 5 ± 0.031 7	0.091 6 ± 0.014 9	0.089 8 ± 0.024 0
CPU	0.055 6 ± 0.016 9	0.073 6 ± 0.041 6	0.060 1 ± 0.034 8	0.053 1 ± 0.015 0	0.054 1 ± 0.018 7
Servo	0.114 0 ± 0.022 7	0.101 6 ± 0.029 8	0.102 3 ± 0.012 9	0.118 3 ± 0.032 7	0.087 7 ± 0.017 6
Breast cancer	0.198 5 ± 0.020 3	0.197 9 ± 0.035 8	0.205 9 ± 0.030 1	0.178 9 ± 0.021 9	0.188 0 ± 0.037 4
Bank	0.043 3 ± 0.000 8	0.040 9 ± 0.000 7	0.072 6 ± 0.001 8	0.042 3 ± 0.000 8	0.044 5 ± 0.001 4
Stocks	0.033 5 ± 0.001 8	0.033 5 ± 0.003 6	0.033 4 ± 0.001 7	0.032 3 ± 0.001 9	0.030 4 ± 0.003 3

表 5 各组标杆数据上 OP-ELM 与 MI-ELM 的模型规模比较

数据集名	OP-ELM ^[5]		MI-ELM	
	输入节点数	隐层节点数	输入节点数	隐层节点数
Abalone	7.615 4	39.923 1	7.000 0	50.000 0
Delta ailerons	4.000 0	56.500 0	4.000 0	42.700 0
Delta elevators	6.000 0	42.363 6	4.966 7	50.000 0
Computer activity	6.666 7	36.250 0	7.800 0	33.300 0
Auto price	11.785 7	11.571 4	12.800 0	18.200 0
CPU	5.818 2	7.090 9	5.600 0	9.300 0
Servo	3.181 8	30.000 0	3.700 0	25.000 0
Breast cancer	20.000 0	5.090 9	12.400 0	11.200 0
Bank	7.272 7	188.636 4	3.600 0	23.500 0
Stocks	9.000 0	152.636 4	9.000 0	192.100 0

表 6 各组标杆数据的训练时间仿真结果比较

数据集名	ELM ^[1]	E-ELM ^[4]	RBF	OP-ELM ^[5]	MI-ELM
Abalone	0.024 4	97.352 5	26.474 4	40.757 2	36.703 1
Delta ailerons	0.103 1	374.338 3	62.634 9	83.470 0	13.477 1
Delta elevators	0.818 1	3 104.000 0	107.804 7	113.964 5	74.613 5
Computer activity	0.723 1	2 785.500 0	86.512 5	163.155 0	39.825 0
Auto price	0.003 4	5.415 6	1.993 7	1.548 0	2.281 3
CPU	0.003 1	4.026 6	0.998 4	0.688 9	1.092 2
Servo	0.005 6	10.992 2	1.987 5	0.568 2	3.078 1
Breast cancer	0.003 7	4.644 5	2.042 2	4.389 2	1.575 0
Bank	1.097 5	4 124.300 0	85.423 4	411.852 3	16.559 4
Stocks	0.068 1	270.421 9	6.484 4	22.565 4	274.425 0

表 6 列出了 5 种算法在 10 组标杆数据上的训练时间, ELM 在时间效率上有绝对的优势. 在 RBF 网络和其余 3 种结合了优化方法的模型中, MI-ELM 在 5 组数据上的训练时间最短, 仅在 Stocks 数据上的时间消耗较多, 这是由于 MI-ELM 在该数据集上训练得到的网络规模较大所致.

4 结 论

针对回归问题中存在的变量选择和网络结构设计问题, 本文基于互信息变量选择和 ELM 的训练机制提出了一种能够自动确定输入变量和隐层节点规模的模型学习算法. 通过 Lorenz 和 Gas Furnace 两组多元时间序列和 10 组多变量标杆数据的仿真实验验证了所提 MI-ELM 算法的有效性. 仿真结果表明, MI-ELM 算法不仅能够同时优化输入层和隐含层的网络结构, 而且能够获得较好的网络性能, 从而能够有效解决多元时间序列与多变量数据中存在的变量选择和网络结构优化问题. MI-ELM 的以上优点是以时间为代价的, 与 ELM 相比, 其计算复杂度仍然较高, 所以当对算法的时间效率要求严格时, MI-ELM 不再适用, 而 ELM 将是不错的选择.

参考文献(References)

- [1] Huang G B, Zhu Q Y, Siew C K. Extreme learning machine: Theory and applications[J]. Neurocomputing, 2006,70(1/2/3): 489-501.
- [2] 韩敏, 王新迎. 基于信赖域 Newton 算法的 ELM 网络[J]. 控制与决策, 2011, 26(5): 757-760.
(Han M, Wang X Y. ELM based on trust region Newton method[J]. Control and Decision, 2011, 26(5): 757-760.)
- [3] 张弦, 王宏力. 限定记忆极端学习机及其应用[J]. 控制与决策, 2012, 27(8): 1206-1210.
(Zhang X, Wang H L. Fixed-memory extreme learning machine and its applications[J]. Control and Decision, 2012, 27(8): 1206-1210.)
- [4] Zhu Q Y, Qin A K, Suganthan P N, et al. Evolutionary extreme learning machine[J]. Pattern Recognition, 2005, 38(10): 1759-1763.
- [5] Miche Y, Sorjamaa A, Bas P, et al. OP-ELM: Optimally pruned extreme learning machine[J]. IEEE Trans on Neural Networks, 2010, 21(1): 158-162.
- [6] Malhi A, Gao R X. PCA-based feature selection scheme for machine defect classification[J]. IEEE Trans on Instrumentation and Measurement, 2004, 53(6): 1517-1525.
- [7] Hyvarinen A, Oja E. Independent component analysis: Algorithms and applications[J]. Neural Networks, 2000, 13(4/5): 411-430.
- [8] Yamamoto H, Yamaji H, Fukusaki E, et al. Canonical correlation analysis for multivariate regression and its application to metabolic fingerprinting[J]. Biochemical Engineering J, 2008, 40(2): 199-204.
- [9] Barrett A B, Barnett L, Seth A K. Multivariate granger causality and generalized variance[J]. Physical Review E, 2010, 81(4): 041907.
- [10] Sallehuddin R, Shamsuddin S M H, Hashim S Z M. Application of grey relational analysis for multivariate time series[C]. Proc of 8th Int Conf on Intelligent Systems Design and Applications. Piscataway, 2008: 432-437.
- [11] Shannon C E A. Mathematical theory of communication[J]. ACM SIGMOBILE Mobile Computing and Communications Review, 2001, 5(1): 3-55.
- [12] 韩敏, 刘晓欣. 基于互信息的分步式输入变量选择多元序列预测研究[J]. 自动化学报, 2012, 38(6): 999-1006.
(Han M, Liu X X. Stepwise input variable selection based on mutual information for multivariate forecasting[J]. Acta Automatica Sinica, 2012, 38(6): 999-1006.)
- [13] Box G E P, Jenkins G M, Reinsel G C. Time series analysis: Forecasting and control[M]. New Jersey: John Wiley & Sons, 2008: 677-678.

(责任编辑: 闫 妍)