

改进的 OLS 算法选择 RBFNN 中心的方法

郑明文

ZHENG Ming-wen

中国石油大学(华东) 计算机与通信工程学院, 山东 东营 257061

Department of Computer and Communication Engineering, China University of Petroleum, Dongying, Shandong 257061, China

ZHENG Ming-wen. RBFNN center choice method based on Kohonen network and OLS algorithm. Computer Engineering and Applications, 2009, 45(25): 52-54.

Abstract: This article proposes an optimized choice radial basis function neural network data central algorithm. This algorithm unifies the Kohonen network's pattern classification ability, classifies firstly the result to make RBFNN the initial data center, and then uses the OLS algorithm to carry on optimized choice. The contrast simulation experiments indicate that this algorithm produces better RBFNN performance than using OLS algorithm independently.

Key words: Radial Basis Function Neural Network(RBFNN); data center; Kohonen network; Orthogonal Least Squares(OLS) method

摘要:提出了一种优化选择径向基神经网络数据中心的算法,该算法结合了 Kohonen 网络的模式分类能力,将初步分类结果用做 RBFNN 的初始数据中心,然后采用 OLS 算法进行优化选择,对比仿真实验表明该算法效果比单独使用 OLS 算法生成的 RBFNN 性能更好。

关键词:RBF 神经网络(RBFNN); 数据中心; Kohonen 网络; 正交最小二乘法

DOI:10.3778/j.issn.1002-8331.2009.25.016 **文章编号:**1002-8331(2009)25-0052-03 **文献标识码:**A **中图分类号:**TP183

1 引言

RBFNN(Radius Basis Function Neural Network)是一种三层的前馈网络,其网络结构简单,而且只有一个隐层,所以网络的训练速度较快,经常用于非线性函数逼近、时间序列分析、模式识别、信息处理、图像处理等较多场合。和其他神经网络一样,对于 RBF 网络,同样需要它具有很好的逼近能力和泛化能力。尤其是泛化能力,它是衡量神经网络性能的主要指标,因为它是对未知模式的逼近或者分类的能力,未知的信息更加重要。影响网络泛化能力的因素很多,如训练样本的质量和数量、网络的结构复杂性、对所学习的目标规则的先验知识、网络的初始权值、训练时间等^[1]。现在大多数对 RBFNN 研究的重点放到了径向基中心的选取和宽度的确定,以及权值的优化方面^[2-4]。在这些算法中,有的是包含了全部的输入样本,然后通过一些算法在构造网络的时候进行修剪来减少网络的规模;有的是使用全部的输入样本通过无监督聚类技术如 K-means 方法来构造径向基函数的中心,几乎都没有考虑网络输入的样本的选择问题。然而,文献[5]指出,拒绝一些非信息的训练样本的输入能够提高网络的泛化能力和解释能力。前面的方法虽然能够有效地自动选择数据中心,但是如果输入样本集很大,则会使计算量非常大,算法的效率明显下降。

提出了一种简单的对输入样本进行选择的方法,利用 Kohonen 网络模式分类能力对所有的训练样本进行初次分类

(分类模式根据实际情况预先粗略确定),获得一组新的具有分类特征的训练样本,然后对该分类结果进行归一化处理,获得训练 RBFNN 的新的样本集,该样本集数量会明显少于原来的训练样本数量,然后再用 OLS 方法对该样本进行径向基中心的优化选择,由此获得的 RBFNN 本身就是一种稀疏的网络(相对于原始训练样本获得的 RBFNN 而言),在进行的对比仿真测试中,该网络除了能够减少 OLS 算法的计算时间外,根据网络的最简原则,该网络还具有比由原始训练样本获得的网络更好的泛化能力。

2 RBFNN 结构描述

RBFNN 是一个只有一个隐藏层的三层前馈神经网络,因隐藏层的转换函数是径向基函数而得名,这是一种局部响应的函数,因此可以任意精度地逼近任意连续函数,而转换函数一般选择高斯基函数。RBFNN 的网络结构模型如图 1 所示^[6]。

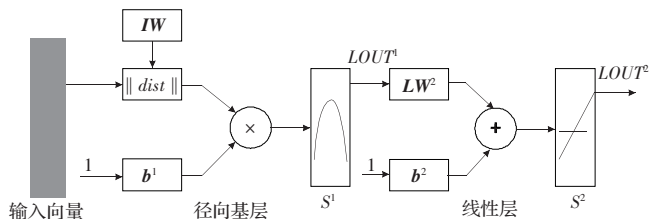


图1 径向基函数网络模型

在图 1 中, IW 和 $LW_i^2=(w_{i1},w_{i2},\dots,w_{ip})^T,i=1,2,\dots,m$ 分别表示输入层和隐藏层 S^1 的权向量,为了便于计算,将 IW 取为输入向量的转置, $\|dist\|$ 是输入向量与径向基中心之间的欧式距离, b^1, b^2 分别是隐藏层 S^1 与输出层 S^2 阈值向量。设隐藏层输出和网络层输出分别用 $LOUT^1, LOUT^2$ 表示。

假设网络的输入样本是 $X=(x_1,x_2,\dots,x_n)^T, IW=X^T$, 实际输出为 $y_k, k=1,2,\dots,p$, 网络输出为 $\hat{y}_k, k=1,2,\dots,p, c_i$ 是第 i 个隐层节点的径向基函数的中心,它与 X 具有相同的维数, σ_i 是第 i 个隐节点的宽度,也称为标准化常数,设隐层节点数为 m , 输出层节点数为 p , 由此可知该 RBFNN 的结构为 $n-m-p$ 。径向基函数一般采用高斯基函数,如式(1)所示:

$$LOUT_i^1(x_j)=\exp\left(-\frac{\|x_j-c_i\|^2}{2\sigma_i^2}\right), i=1,2,\dots,m j=1,2,\dots,n \quad (1)$$

对应所有的隐层输出而得到的矩阵称为设计矩阵(design matrix), 记为 $LOUT$ 。

线性层的输出如式(2)所示:

$$LOUT_k^2(x)=\sum_{i=1}^m w_{ik} \cdot LOUT_i^1(x), k=1,2,\dots,p \quad (2)$$

RBFNN 的学习主要考虑各个径向基函数的数据中心、标准化常数以及输出节点的权值三个参数。文中 RBFNN 的数据中心和标准化常数是通过对 Kohonen 网络获得的数据中心的预处理获得的,这也是该文区别于其他选择该参数的方法的最大区别。Kohonen 网络处理过程相对简单,而且不需要经过复杂的计算,能够节省大量的时间。之后采用 OLS 方法(正交最小二乘法)对数据中心进行二次优化选择。OLS 算法^[7]的过程如下:

(1)计算 $LOUT$,并令 $LOUT$ 的 N 个列向量 $LOUT_1^1, LOUT_2^1, \dots, LOUT_N^1$ 。

(2)把输出数据矢量 y 投影到 $LOUT_1^1, LOUT_2^1, \dots, LOUT_N^1$, 如果 y 与某个 $LOUT_1^k$ 具有最大的夹角,即 $\frac{y^T \cdot LOUT_1^k}{\|y\|^T \cdot \|LOUT_1^k\|}$ 的绝对值最大,则把 $LOUT_1^k$ 对应的样本输入选为第 1 个数据中心。

(3)计算网络的输出权值,得到网络对样本的训练误差。如果误差小于目标值则终止算法,否则对前一步中剩下的 $N-1$ 个向量做 Gram-Schmidt 正交化,使之与 $LOUT_1^k$ 正交,得到 $LOUT_2^1, LOUT_2^2, \dots, LOUT_2^{N-1}$ 。

(4)找出与 y 有最大投影的 $LOUT_2^j$,选择与之对应的第 2 个数据中心,计算输出权值和训练误差,并判断是否终止算法。

(5)重复(2)~(4),直到找到 M 个数据中心,使网络误差小于给定值。

该算法在样本空间很大的时候容易造成“失衡”问题—全局网络样本数量的提高容易导致网络过学习的风险加大。上述算法将作为方法的一部分,用来从新的训练样本(相对原来的样本数有明显减少)中选择 RBFNN 的数据中心,减少了“失衡”问题出现的风险,也节省了正交化过程计算的开支。

3 Kohonen 神经网络结构描述

Kohonen 网络是一种自组织竞争型神经网络。它是基于人

的视网膜及大脑皮层对刺激的反应而引出的。自组织竞争型神经网络的基本思想是网络竞争层的各神经元竞争对输入模式的响应机会,最后仅有一个神经元成为竞争的胜者,并且只将与获胜神经元有关的各连接权值进行修正,使之朝着更有利于它竞争的方向调整。本质上讲,利用 Kohonen 网络对样本进行选择其实是一种聚类方法,依据是根据样本之间的相似性。该神经网络工作时,对于某一输入模式,网络中与该模式最相近的学习输入模式相对应的竞争层神经元将有最大的输出值,即以竞争层获胜神经元来表示分类结果。Kohonen 网络结构如图 2^[8]所示。

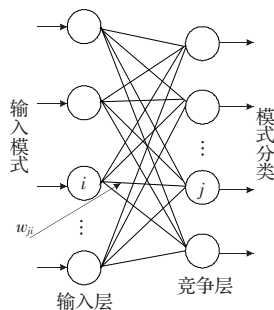


图 2 Kohonen 网络结构

网络学习时, Kohonen 网的权值用满足 $\sum_i w_{ji} \approx 1$ 的小随机数初始化,然后计算每个单元输入的加权和,对第 j 个单元输出总和如式(3)所示:

$$S_j = \sum_i w_{ji} x_i \quad (3)$$

当所有单元的输入总和计算完毕,便开始竞争,竞争规则如式(4)所示:

$$x_j^c = \begin{cases} 1, & S_j > \max(S_k, k \neq j, k=1, 2, \dots, n) \\ 0, & \text{others} \end{cases} \quad (4)$$

x_j^c 表示竞争层的第 j 个单元输出状态。对于一个输入模式,当竞争获胜的单元被确定后,更新权值,只有获胜单元的权值被增加,使得再次遇到该模式输入时,该单元有更大的输入总和。权值更新规则如式(5)所示:

$$\Delta w_{ji} = \eta \left(\frac{x_i}{m} - w_{ji} \right) \quad (5)$$

其中 η 是学习因子, $0 < \eta < 1$, 一般取值为 0.01~0.3, 反映权值的更新速率, m 表示输入层状态为 1 的单元个数。

4 基于 Kohonen 网络和 OLS 算法的 RBFNN 中心选择方法

首先根据实际应用粗略确定样本的类别数,然后利用 Kohonen 网络的模式分类能力对输入向量进行分类,记录分类结果,然后对分类结果进行预处理,得到一组用来径向基中心的候选集,然后利用 OLS 算法对该中心候选集进行选择,最终确定出 RBFNN 的径向基中心,得到最后的 RBFNN。径向基中心算法的流程框图如图 3 所示。算法具体过程如下:

(1)对训练集中的所有样本进行归一化处理,并根据实际应用假设样本首先分为 K 类,设置 Kohonen 网的学习速率,创建一个的 Kohonen 网络。设定 RBFNN 的网络误差 ε , 最大训练次数 T 。

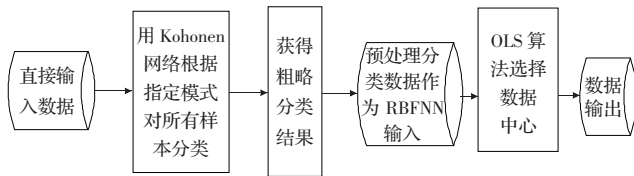


图3 算法处理过程框图

(2)从样本集中选择 K 个样本作为隐层节点中心向量的初始值,按下面方式训练 Kohonen 网络,将每个训练样本和每个中心的内积作为评价两个向量距离的尺度,与当前训练模式最近的中心可以得到确定,即内积最大的中心,这一中心要向这个训练样本作微小改动,直到中心向量体现出样本数据的统计特性。此时的中心向量记为 $\{x'_i, i=1, 2, \dots, k\}$ 。

(3)将 $\{x'_i, i=1, 2, \dots, k\}$ 作为 RBFNN 的隐层节点,采用第 2 章的 OLS 算法进行选择,直到满足给定的网络误差 ε 或者达到训练次数 T 。

(4)由上面确定的 RBFNN 的数据中心,采用固定的宽度,建立最终的 RBFNN。

5 实验及结果分析

5.1 逼近问题研究

为了验证方法的有效性,分别用该文提出的方法和只用 OLS 方法对下面一种变形的 Hermite 函数(6)进行逼近:

$$y=1.1(1-x+2x^2)\exp(-\frac{x^2}{2})+0.4\sin(\pi ix) \quad (6)$$

取区间 $(-2, 2)$ 之间,随机产生 400 个训练样本,并对应样本的实际输出添加均值为 0,均方差为 0.1 的高斯噪声。根据样本分布情况,将所有样本初步划分成 50 类,划分方法可参考文献[9]由此创建 Kohonen 网络,经过训练后,得到一个含有 50 个元素的数据中心,将这 50 个数据点再作为 RBFNN 的初始隐节点用 OLS 算法进行优化选择,得到最终的 RBFNN。而 OLS 算法初始时是将所有的输入样本作为隐节点,然后利用 Gram-Schmidt 正交化方法进行优化选择,得到最后的 RBFNN。为了说明算法的有效性,在进行对比时径向基宽度都用 $\sigma=0.6$ 。两种方法最终获得的结果如表 1 和图 4~图 5 所示。

表1 两种方法最终结果比较

	Kohonen+OLS	OLS
运行时间/s	50.652 314+1.065 597	240.198 974
最终产生的隐节点数/个	50	400
逼近误差	0.001	0.001

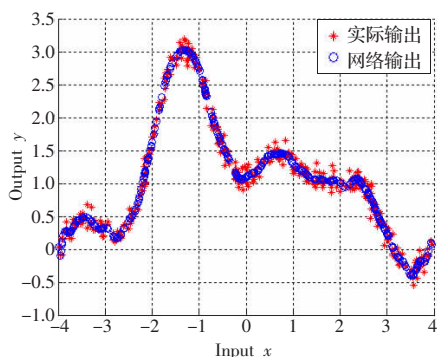


图5 OLS 方法训练样本逼近结果

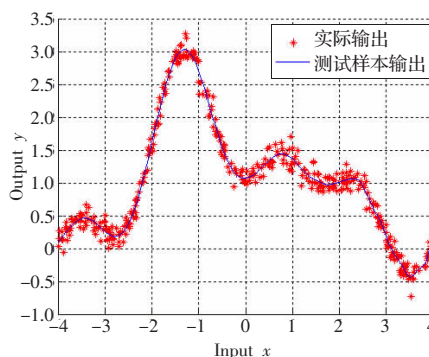


图6 Kohonen-OLS 方法测试样本逼近结果

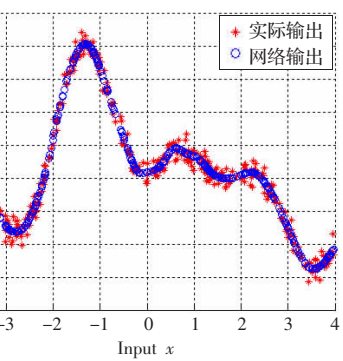


图4 Kohonen+OLS 方法训练样本逼近结果

注:Kohonen+OLS 算法的运行时间中前面是 Kohonen 方法进行样本分类的时间,后一个是 OLS 方法运行的时间(前后两个时间对比可以看出,隐层节点数目对 OLS 方法运行的时间起到重要作用)。

在 CPU 为 Intel T2400,内存为 1 GB 的机器上实验仿真表明当训练样本数较多时,如例中的 400 个样本,在设定的逼近误差阈值较小时(如小于 0.001),OLS 运行速度很慢,并没有按照预期的那样出现优化的隐节点,最终隐节点数目竟然是 400 个,而且出现“震荡”现象。而采用该文提供的方法能够较快地得到较为满意的结果,虽然采用 OLS 算法得到的隐节点数仍为 50,但是运行时间却明显减少,而且效果和只用 OLS 方法几乎相同。

5.2 泛化能力研究

由式(6)产生 100 组测试样本,添加均值为 0,均方差为 0.1 的高斯噪声,对生成的网络进行泛化能力的比较。最终得到方法的泛化误差为 $Error=0.044\ 923$,而采用只用 OLS 方法获得的误差 $Error=0.052\ 296$,说明该文方法的泛化效果比较好。网络的逼近曲线如图 6、图 7 所示。

6 结论

提出了一种新的选择 RBFNN 径向基中心的方法,首先利用 Kohonen 网络较强的模式分类能力对所有训练样本进行粗略分类,并且产生一组类别中心,将该类别中心作为 RBFNN 的初始隐层,然后利用 OLS 算法进行再次优化选择,最终获得 RBFNN 的隐节点。实验仿真表明该方法比单独使用 OLS 算法进行数据中心的选花费时间少,而且生成的网络泛化能力较强。但是,根据反复试验,算法在对训练样本进行分类时如果有一定的先验知识,那么获得的 RBFNN 效果会更好。

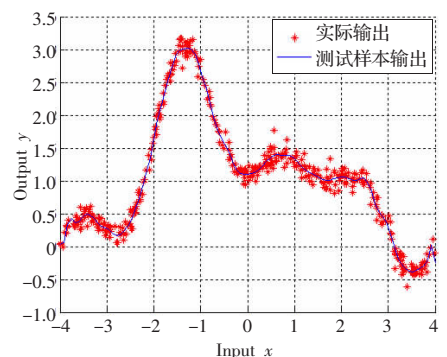


图7 OLS 方法测试样本逼近结果