

支持向量机改进的神经网络的函数逼近

张朝元¹, 胡光华¹, 徐天泽², 殷英¹

(1. 云南大学 数学系, 云南 昆明 650091; 2. 云南警官学院, 云南 昆明 650223)

摘要: 为了避免神经网络的收敛速度慢和局部极小点, 采用统计学习理论中的支持向量机代替梯度下降法对三层神经网络中隐层到输出层的过程进行改进. 分别采用由支持向量机改进的神经网络和传统的神经网络对昆明市“一二一”大街交通交流的实时预测, 实验结果验证了改进后的神经网络的优越性和先进性.

关键词: 神经网络; 支持向量机; 函数逼近; 结构风险最小化; 交通流量

中图分类号: TP183 **文献标识码:** A **文章编号:** 1007-855X(2004)06-0148-05

Function Approximation of Neural Network Modified by Supporting Vector Machines

ZHANG Chao-yuan¹, HU Guang-hua¹, XU Tian-ze², YIN Ying¹

(1. Department of Mathematics, Yunnan University, Kunming 650091, China;

2. Yunnan Police Officer's Academy, Kunming 650223, China)

Abstract: In order to avoid slow convergence and local minimum of three-layered neural network, Supporting Vector Machines of statistical learning theory is used to modify this process from hidden-layer to output-layer instead of gradient descent method. The traffic flow at a crossing of Kunming City was respectively forecast by modified neural network and by classical neural network. And the advantages of modified neural network have been shown and proved by the results of experiment.

Key words: neural network; Supporting Vector Machines; function approximation; structural risk minimization; traffic flow

0 引言

BP 神经网络^[1]是一种多层感知机网络, 它的学习算法采用误差反向传播. 它的模型是将一组样本的 I/O 问题转变为一个非线性优化问题, 使用了优化中最普通的梯度下降法, 用迭代运算求解相应于学习记忆的问题. 通过加入隐层节点, 使得优化问题的可调参数增加, 从而可得到更精确的解. 如果将 BP 神经网络看成从输入到输出的映射, 则这个映射是高度非线性的, 它通过对简单的非线性函数进行有序而复杂的组合来实现最终的复杂函数输出, 且具有这样的性质: 任一连续函数或映射均可用一个三层网络加以实现. BP 网络虽然具有重要的函数映射功能, 且系统结构简单易于编程实现. 但是它也存在着以下问题:

(1) 从数学上看, BP 网络是一个非线性优化问题, 因此不可避免地存在局部极小值问题;

(2) BP 网络的学习收敛速度很慢, 通常需要几千步迭代或更多.

对于 BP 神经网络这些缺点, 国内外进行了不少的改进, 主要有 MFBP 算法、MBP 算法和前向网络的自构形学习算法等. 针对 BP 网络这两个缺点, 本文采用最新的支持向量机算法^[2]来对 BP 网络进行改进.

统计学习理论是由 Vapnik 等人提出的一种小样本统计理论, 它有着完备的理论基础和严格的理论体

收稿日期: 2004-04-28. 基金项目: 云南省教育厅自然科学基金项目资助(项目编号: 02ZY011); 云南大学理(工)科校级科研项目资助(项目编号: 2002Q019SL); 云南省自然科学基金项目资助(项目编号: 2003E0086M).

第一作者简介: 张朝元(1978~), 男, 硕士生. 主要研究方向: 神经网络和统计学习理论的应用.

系, 其中支持向量机是统计学习理论的核心和重点. 支持向量机是结构化风险最小化原理的近似实现, 它能够提高学习机的泛化能力, 既能够由有限的训练集样本得到小的误差, 又能够保证对独立的测试集仍保持小的误差. 而且支持向量机算法是一个凸优化问题, 因此局部最优解一定是全局最优解. 这样, 由 Vapnik 等人提出的新的神经网络——支持向量机就克服了传统神经网络收敛速度慢和局部极小点等缺陷.

本文正是利用支持向量机这些 BP 网络不具有的优点来改进 BP 网络, 使得 BP 网络得到更好地提高和使用. 基于支持向量机这些优点, 目前支持向量机方法被广泛地应用于文本识别^[3]、函数估计^[4]、模式识别^[5]、回归分析^[6]和金融时间序列预测^[7]等各类问题.

1 三层神经网络

设有 l 个样本 $(x_1, y_1), (x_2, y_2), \dots, (x_l, y_l)$ 其中, $x_i \in R^n, y_i \in R, i = 1, 2, \dots, l$. 我们的目的是通过算法来寻找 x 与 y 之间的关系. 现在采用三层的 BP 神经网络来对其进行估计和预测. 这样输入层单元数为 n , 输出层单元数为一个. 设隐层单元数为 m , 从输入层到隐层的权值和阈值分别为 $W' \in R^{n \times m}, \theta \in R^m$, 隐层的输入和输出分别为 $u \in R^m (u = (u_1, u_2, \dots, u_m)), v \in R^m (v_1, v_2, \dots, v_m)$. 同时, 从隐层到输出层的权值和阈值分别为 $w \in R^m, b \in R$. 输出层的输入和输出均为 $t \in R$ (本文采用线性函数). 隐层间和输出层的传递函数可以选用具有可微分的 S 型函数和非线性函数或线性函数. 为了方便, 我们将隐层的阈值添加到权值里去, 同时增加一个输入量 $x_0 = -1$, 则 $W = [\theta; W']$.

BP 网络主要包括输入层到隐层和隐层到输出层这两个过程的前向计算和反向计算. 通过前向计算和反向计算来调整权值和阈值. 我们主要是采用通常的梯度下降法来对输入层到隐层之间的权值和阈值进行调整. 下面给出输入层到隐层之间的权值 W 的推导.

设在第 k 步迭代中输出端的输出为 $t(k)$, 该输出的误差为 $e(k) = y(k) - t(k)$, 则输出端的平方误差瞬时值为: $E(k) = \frac{1}{2} e^2(k)$. 由于训练样本总数为 l , 则平方误差的均值为: $E_{AV} = \frac{1}{l} \sum_{k=1}^l E(k)$. E_{AV} 为学习的目标函数, 学习的目的使 E_{AV} 达到最小, E_{AV} 是网络所有权值和阈值以及输入的函数. 下面采用批处理方式的情况推导 BP 算法.

由于 $w_j(k) = \sum_{i=0}^n W_{ij}(k) x_i(k), v_j(k) = \Phi(u_j(k))$, 求 E_{AV} 对 W_{ij} 的梯度:

$$\frac{\partial E_{AV}}{\partial W_{ij}(k)} = \frac{\partial E_{AV}}{\partial E(k)} \frac{\partial E(k)}{\partial e(k)} \frac{\partial e(k)}{\partial v_j(k)} \frac{\partial v_j(k)}{\partial u_j(k)} \frac{\partial u_j(k)}{\partial W_{ij}(k)}$$

其中 $\frac{\partial E_{AV}}{\partial E(k)} = \frac{1}{l}, \frac{\partial E(k)}{\partial e(k)} = e(k), \frac{\partial e(k)}{\partial v_j(k)} = \frac{\partial e(k)}{\partial t(k)} \frac{\partial t(k)}{\partial v_j(k)}, \frac{\partial v_j(k)}{\partial u_j(k)} = \phi'(u_j(k)), \frac{\partial u_j(k)}{\partial W_{ij}(k)} = x_i(k)$.

由于 $t(k) = \sum_{j=1}^m w_j(k) v_j(k) + b(k)$, 所以有: $\frac{\partial e(k)}{\partial t(k)} = -1, \frac{\partial t(k)}{\partial v_j(k)} = w_j(k)$, 即: $\frac{\partial e(k)}{\partial v_j(k)} = -w_j(k)$.

故有: $\frac{\partial E_{AV}}{\partial W_{ij}(k)} = -\frac{1}{l} (y(k) - t(k)) w_j(k) \phi'(u_j(k)) x_i(k)$.

权值 W_{ij} 的修正量为: $\Delta W_{ij}(k) = -\eta \frac{\partial E_{AV}}{\partial W_{ij}(k)} = \frac{\eta}{l} (y(k) - t(k)) w_j(k) \phi'(u_j(k)) x_i(k)$.

当我们通过下一节的支持向量机方法对从隐层到输出层的权值进行修正后, 就能通过反向计算来修正输入层到隐层的权值. 即:

$$W_{ij}(k+1) = W_{ij}(k) + \frac{\eta}{l} (y(k) - t(k)) w_j(k) \phi'(u_j(k)) x_i(k)$$

如果我们选择 $v_j(k) = \Phi(u_j(k)) = \frac{1}{1 + \exp(-u_j(k))}$, 则 $\phi'(u_j(k)) = \frac{\partial v_j(k)}{\partial u_j(k)} = v_j(k)(1 - v_j(k))$.

所以得到批处理下的权值为: $W_{ij} = W_{ij} + \frac{\eta}{l} \sum_{k=1}^l (y(k) - t(k)) w_j(k) \phi'(u_j(k)) x_i(k)$.

从下一节可知, 事实上 w 是不随 k 变化的 (在批处理方式), 所以就有更简单的权值表达式:

$$W_{ij} = W_j + \frac{1}{l} \sum_{k=1}^l (y(k) - t(k)) w_j \phi'(u_j(k)) x_i(k). \quad (1)$$

2 支持向量机

采用支持向量机来求出从隐层到输出层之间的权值和阈值. 由于本文考虑的是输出层的输入与输出相等, 也就是说是一个量 t , 而实际输出为 y , 那么隐层得到的输出 $v = [v(1), v(2), \dots, v(m)]$ 就作为新的输入, 这样就有 l 个样本 $(v_1, y_1), (v_2, y_2), \dots, (v_l, y_l)$, 由于隐层到输出层的权值和阈值分别为 w, b , 则我们要寻找的逼近函数为:

$$t = f(v) = w \cdot v + b \quad (2)$$

为了找到(2)式中最佳的 w, b , 必须求解下面的最优问题:

$$\min E(w, b) = \frac{1}{2} \|W\|^2 + C \sum_{i=1}^l |y_i - f(v_i)| \varepsilon$$

其中 $|y_i - f(v_i)| \varepsilon = \max\{0, |y_i - f(v_i)| - \varepsilon\}$ 为 ε 不敏感损失函数, 当然还有其他的损失函数, 如平方损失函数和 Huber 损失函数^[2] 等. 这样上式就等价于下列式子:

$$\min E(w, b, \xi^*, \xi) = \frac{1}{2} \|w\|^2 + C \left(\sum_{i=1}^l \xi_i^* + \sum_{i=1}^l \xi_i \right) \quad (3)$$

s. t $y_i - f(v_i) \leq \varepsilon + \xi_i^*, f(v_i) - y_i \leq \varepsilon + \xi_i, \xi_i^* \geq 0, \xi_i \geq 0$, 其中 $i = 1, 2, \dots, l$, 要求解带有不等式类型约束的优化问题, 我们必须找到下面拉格朗日泛函的鞍点:

$$L(w, b, \xi^*, \xi, C, \alpha^*, \alpha, \gamma^*, \gamma) = E(w, b, \xi^*, \xi) - \sum_{i=1}^l \alpha [y_i - f(v_i) + \varepsilon + \xi_i] - \sum_{i=1}^l \alpha_i^* [f(v_i) - y_i + \varepsilon + \xi_i^*] - \sum_{i=1}^l (\gamma_i^* \xi_i^* + \gamma_i \xi_i) \quad (4)$$

$$\text{s. t } \alpha_i^*, \alpha \geq 0 \quad i = 1, 2, \dots, l$$

将(4)式分别对 w, b, ξ^* 和 ξ 求导并令其等于零: $\frac{\partial L}{\partial w} = 0, \frac{\partial L}{\partial b} = 0, \frac{\partial L}{\partial \xi^*} = 0, \frac{\partial L}{\partial \xi} = 0$.

$$\text{则有: } w = \sum_{i=1}^l (\alpha_i^* - \alpha) v_i, \sum_{i=1}^l (\alpha_i^* - \alpha) = 0, C = \alpha_i^* + \gamma_i^* = \alpha_i + \gamma_i \quad (5)$$

把上面的 w, C 代入(4)式得到(3)式的对偶形式:

$$\max W(\alpha, \alpha^*) = -\varepsilon \sum_{i=1}^l (\alpha_i^* + \alpha) - \frac{1}{2} \sum_{i,j} (\alpha_i^* - \alpha_i)(\alpha_j^* - \alpha_j)(v_i \cdot v_j) + \sum_{i=1}^l y_i (\alpha_i^* - \alpha) \quad (6)$$

$$\text{s. t } \sum_{i=1}^l (\alpha_i^* - \alpha_i) = 0, 0 \leq \alpha_i^*, \alpha_i \leq C \quad (i = 1, 2, \dots, l)$$

这样, 函数逼近问题就归结为一个二次规划问题(6)式. 求解(6)式这个简单的二次规划问题就能得到 α^*, α , 从而得到: $w = \sum_{i=1}^l (\alpha_i^* - \alpha) v_i$.

根据 Karush-Kuhn-Tucker(KKT) 条件, 在最优解处, 有

$$\begin{cases} \alpha_i^* (\varepsilon + \xi_i^* - y_i + f(v_i)) = 0 \\ \alpha_i (\varepsilon + \xi_i + y_i - f(v_i)) = 0 \end{cases} \quad \text{和} \quad \begin{cases} (C - \alpha_i^*) \xi_i^* = 0 \\ (C - \alpha_i) \xi_i = 0 \end{cases}$$

由上式可以得出, 位于不敏感区内的样本点, 其对应的 α_i 和 α_i^* 都等于零, 外部的点对应有 $\alpha = C$ 或 $\alpha_i^* = C$, 而在边界上, ξ_i 和 ξ_i^* 均为零, 因而 $\alpha_i, \alpha_i^* \in (0, C)$, 从而有:

$$\begin{cases} b = y_i - w \cdot v_i - \varepsilon, \alpha_i \in (0, C) \\ b = y_i - w \cdot v_i + \varepsilon, \alpha_i^* \in (0, C) \end{cases} \quad (7)$$

通过求解(7)就可得到 b . 于是逼近函数为: $t = f(v) = \sum_{i=1}^l (\alpha_i^* - \alpha_i)(v_i \cdot v) + b$

3 函数逼近算法

到现在为止, 我们已经分别通过梯度下降法和支持向量机找到了从输入层到隐层与隐层到输出层的权值和阈值的更新规律, 即(1)式、(5)式和(7)式. 下面把由支持向量机改进的 BP 神经网络学习过程的具体步骤归纳如下:

(1) 初始化: 选定隐层单元个数合适的网络; 置权值和阈值 W_{ij} 为均匀分布的较小数值; 确定学习次数及误差的预定值.

(2) 对每一个输入样本, 计算 $u_j(k) = \sum_{i=0}^n W_{ij}x_i(k)$ 和 $v_j(k) = \varphi(u_j(k))$, 得到 $v = [v(1), v(2), \dots, v(m)]$, 将其作为支持向量机的输入向量.

(3) 求解(6)式这个二次规划问题, 可以得到 $w = \sum_{i=1}^l (\alpha_i^* - \alpha_i) v_i$, 同时由(7)式得到 b , 即隐层到输出层之间的权值和阈值.

(4) 计算 $\phi(u_j(k)) = \frac{\partial v_j(k)}{\partial u_j(k)}$, 将得到的 $\phi(u_j(k))$ 和 w 代入(1)式得到更新的 W .

(5) 根据更新的 W, w, b 计算 E_{AV} , 判断误差是否小于预定值或学习次数是否等于预定值, 若是则学习完毕; 否则学习次数加 1, 返回到(2)步.

4 实验和结果

为了验证和评价本文提出的由支持向量机进行改进的 BP 神经网络的有效性和可行性, 下面我们将利用该算法对云南省昆明市一二一大街的交通流量进行模拟和预测.

为了能较好地分析和评价预测的结果, 首先引进以下五个误差指标:

$$\text{平均相对误差: } mrerr = \frac{1}{N} \sum_t \frac{T_{pred}(t) - T_{real}(t)}{T_{real}(t)},$$

$$\text{平均绝对相对误差: } marerr = \frac{1}{N} \sum_t \frac{|T_{pred}(t) - T_{real}(t)|}{|T_{real}(t)|},$$

$$\text{最大绝对相对误差: } mxarerr = \max \left| \frac{T_{pred}(t) - T_{real}(t)}{T_{real}(t)} \right|,$$

$$\text{相对误差平方和均值平方根: } rmrerr = \sqrt{\frac{1}{N} \sum_t \left(\frac{T_{pred}(t) - T_{real}(t)}{T_{real}(t)} \right)^2},$$

$$\text{均等系数: } EC = 1 - \frac{\sqrt{\sum_t (T_{pred}(t) - T_{real}(t))^2}}{\sqrt{\sum_t (T_{pred}(t))^2 + \sum_t (T_{real}(t))^2}}$$

其中 EC 表示预测值与真实值之间的拟合, 一般在 0.90 以上表示拟合程度较好.

本文根据某交通路段在过去一段时间的交通流量记录, 预测后一时段的交通流量. 昆明市一二一大街是昆明市的主要交通要道, 对它的研究和预测具有重要的现实意义. 经过对一二一大街(建设路交叉路

口至圆通北路交叉路口)由西向东方向机动车(除摩托车外)交通流量的人工统计, 我们得到了一系列实际观测数据. 现在, 分别利用由支持向量机改进的神经网络和 BP 神经网络来对其进行预测, 预测结果和误差指标值见表 1 和图 1(a) 及图 1(b). 从图 1(a) 和图 1(b) 的比较可见, 由改进的方法得到的预测交通流量和实际的交通流量基本上是一致的. 表 1 中的误差指标值 ($marerr$) 也显示出了较低的平均误差为 6.09%,

表 1 改进的神经网络和未改进的神经网络各种误差值的比较
Tab. 1 The Error comparison of the modified neural network(NN) and classcal neural network

误差指标	$mrerr$ /%	$marerr$ /%	$mxarerr$ /%	$rmrerr$ /%	EC
修正的神经网络	1.26	6.09	22.58	8.60	0.9562
普通神经网络	0.88	7.98	21.33	9.45	0.9531

而且表1中的误差指标值 EC 达到了 0.9562 的高拟合度, 而采用 BP 神经网络的平均绝对相对误差达到了 7.98%, 明显不如改进的好. 这样就说明由支持向量机改进的 BP 神经网络比传统的神经网络在函数逼近和模拟预测方面更进一步, 更具有实际现实意义.

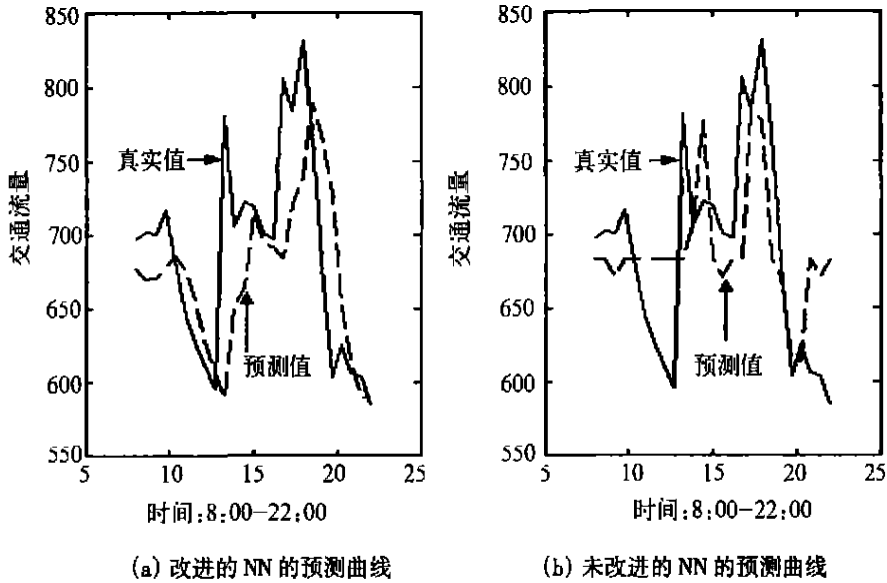


图1 改进的神经网络和未改进的神经网络预测结果图

Fig.1 The predicted results of the modified neural network and classical neural network

5 结论

以 Vapnik 创建的统计学习理论为基础的支持向量机算法采用结构风险最小化准则, 在最小化样本点误差的同时, 缩小模型泛化误差的上界, 即最小化模型的结构风险, 从而提高了模型的泛化能力; 同时, 支持向量机是将函数逼近和估计问题转化成求解一个凸二次规划优化问题, 从而局部最优解就一定是全局最优解; 而且支持向量机方法具有收敛速度快等特点. 基于这些优点, 本文将支持向量机和神经网络有机地结合起来, 用于解决神经网络中隐层到输出层的权值优化问题, 使神经网络得到进一步完善, 从而有效地提高神经网络的泛化能力和推广能力. 采用由支持向量机改进的神经网络对昆明市“一二一”大街交通流量进行实时地模拟预测, 从预测的结果可以看出, 经过改进的神经网络在进行预测时能取得理想的效果, 比普通的神经网络更进了一步; 同时, 也说明经过支持向量机改进的神经网络更具有推广性和实用价值.

参考文献:

- [1] 阎平凡, 张长水. 人工神经网络与模拟进化计算[M]. 北京: 清华大学出版社, 2000. 17~ 20.
- [2] Vladimir N. Vapnik. 统计学习理论的本质[M]. 张学工译. 北京: 清华大学出版社, 2000, 126~ 130.
- [3] Joachims T. Text categorization with support vector machine: Learning with many relevant feature[A]. Proceedings of the 10th European Conference on Machine Learning[C]. Berlin: Springer, 1998, 137~ 142.
- [4] Girori F. An equivalence between sparse approximation and support vector machines[J]. Neurocomputing, 1998, 10(6): 1455~ 1480.
- [5] Amari S, Wu S. Improving support vector machine classifiers by modifying kernel function[J]. Neurocomputing., 1999, (12): 783~ 789.
- [6] Gao J B, Gunn S R, Harris R J. Mean field method for the support vector machine regression[J]. Neurocomputing., 2003, (50): 391~ 405.
- [7] Francis E H Tay, Cao G J. Modified support vector machines in financial time series forecasting[J]. Neurocomputing, 2002, (48): 847~ 861.