

文章编号:1000-582x(2001)03-0076-04

多项式前向神经网络

谢开贵¹, 柏森², 周家启¹

(1. 重庆大学电气工程学院电力系统研究所, 重庆 400044; 2. 重庆通信学院 一系, 重庆, 400035)

摘要: 提出一种新型的前向神经网络模型——多项式神经网络。该网络具有三层结构, 隐层、输出层神经元激活函数分别为: $f(x) = x^p$ 和线性函数; 网络隐层-输出层的权值采取最速下降法学习, 输入层-隐层的权值采用遗传算法进行学习; 网络学习时, 其误差函数单调递减, 学习算法具有较好的收敛性; 该网络能逼近任意的连续函数, 且具有较好的稳定性。应用实例表明该网络的性能是优良的。

关键词: 多项式前向神经网络; 最速下降法; 遗传算法; 算法收敛性; 逼近能力

中图分类号: TP 18

文献标识码: A

在人工神经网络(ANN, Artificial Neural Network)的研究和应用中, 人们已提出许多神经网络模型^[1,2], 有前向网络, 如: 单层感知器、MP模型、Adline网络; 反馈网络, 如: Hopfield网络、BAM网络、细胞神经网络等; 随机神经网络: 基于遗传算法、进化规划、模拟退火算法的网络; 自组织神经网络, 如: ART网络、自组织特征映射、CPN网络等。这些网络模型中, 多层前向网络(MFNN, Multi-layer Feedforward Neural Network)是其中最常用的形式, 它为输入和输出提供一种非线性关系, 其应用大多是以这一映射关系为基础。

当ANN结构确定后, ANN的训练便是一个非常重要的阶段, 一方面要考虑提高学习的速度, 另一方面又要兼顾ANN的逼近能力等因素。由已知的输入输出数据对神经网络进行训练, 实际上是一种解析的无约束非线性寻优过程, 但是, 由于神经元函数的非线性特性, 导致神经网络输入与输出之间有高度的非线性关系, 所以, 在一定的时间范围内寻出最优解往往是困难的。

目前, 对多层前向网络训练算法以Rumelhart的BP算法为主。BP算法本质上是一种梯度下降算法, 其有以下几点不足: 如果学习率选择不当, 收敛速度相当慢; 误差函数并非总是权值 w_{ij} 和 v_{ki} 的凸函数, 从而在某些初始条件下, 易陷入局部最小; 误差函数的非凸性

还导致其结果出现震荡现象, 达不到训练的效果。虽然许多学者对此进行了深入研究, 采取变学习率、加入动量因子等措施^[2], 但具有普遍实用意义的可行的算法还是没有的。

笔者针对前向网络难于学习的特点, 给出了一种新型的神经网络模型——多项式前向神经网络(Polynomial Feedforward Artificial Neural Network, PFANN)。在以下的内容中分别讨论网络的结构、学习算法、网络的非线性映射能力、网络的稳定性等内容, 最后还给出PFANN的几个应用实例。

1 PFANN的学习算法

PFANN是一由输入层、隐层、输出层组成的三层前向网络, 与通常三层前向神经网络的结构相同。

设 $X^\mu = (x_1^\mu, x_2^\mu, \dots, x_n^\mu)$, $Y^\mu = (y_1^\mu, y_2^\mu, \dots, y_m^\mu)$, $D^\mu = (d_1^\mu, d_2^\mu, \dots, d_n^\mu)$ 分别为第 μ 输入模式、实际输出模式和理想输出模式, 其中 n 为输入模式分量数(即输入层神经元个数); m 为输出模式分量数(即输出层神经元个数)。该网络输入层映射函数为恒等函数 $f_{in}(x) = x$, 隐层神经元的非线性映射函数不是通常的Sigmoid函数 $f_{hide}(x) = 1/(1 + \exp(-(x - \theta)))$, 而是多项式函数 $f_{hide}(x) = (x + \theta)^P$ (其中 P 为某一选定的整数), 其输出层映射函数为一线性函数, 特别地, $f_{out}(x) = x$ 。

• 收稿日期: 2000-08-18

基金项目: 重庆市科委科技计划资助项目(99-5918)

作者简介: 谢开贵(1972-), 男, 四川眉山人, 重庆大学博士生。从事电力系统规划与可靠性、遗传算法及神经网络等研究。

设 v_{ki} 为上述 PFANN 的输入层第 i 个神经元与隐层第 k 个神经元的连接权值 ($|v_{ki}| \leq 1$), w_{jk} 为隐层第 k 个神经元与输出层第 j 个神经元的连接权值。

记 $E_\mu = \frac{1}{2} \sum_{j=1}^m (y_j^\mu - d_j^\mu)^2$, E 是定义在训练空间上的全局误差函数, 即:

$$E = \sum_{\mu=1}^N E_\mu = \frac{1}{2} \sum_{\mu=1}^N \sum_{j=1}^m \left(\sum_{k=1}^l w_{jk} \left(\sum_{i=1}^n v_{ki} x_i^\mu - \theta_k \right)^P - \theta_j - d_j^\mu \right)^2$$

$$\frac{\partial E}{\partial w_{jk}} = \frac{\partial \left(\frac{1}{2} \sum_{\mu=1}^N \sum_{j=1}^m (y_j^\mu - d_j^\mu)^2 \right)}{\partial w_{jk}} = - \sum_{\mu=1}^N (d_j^\mu - y_j^\mu) h_k^\mu$$

$$\frac{\partial^2 E}{\partial w_{jk}^2} = \sum_{\mu=1}^N \frac{\partial (y_j^\mu - d_j^\mu)}{\partial w_{jk}} h_k^\mu = \sum_{\mu=1}^N (h_k^\mu)^2 \geq 0 \quad (1)$$

式中 N 为模式数,

$$h_k^\mu = \left(\sum_{i=1}^n v_{ki} x_i^\mu - \theta_k \right)^P$$

显然 $E = E(w_{jk}, v_{ki})$ 是关于 w_{jk} 的凸函数, 故可采用最速下降法训练 w_{jk} , 取:

$$w_{jk}(t+1) = w_{jk}(t) + \beta \sum_{\mu=1}^N (d_j^\mu - y_j^\mu) h_k^\mu$$

其中 β 为学习率。

由(1)式可知: 在权空间对 w_{jk} 沿梯度方向进行调整, 误差函数总是下降的。

当权重 $w_{jk}, \theta_j, \theta_k$ 已知时, v_{ki} 的学习可以用如下的规划问题进行求解:

$$\text{Min } E(v) = \frac{1}{2} \sum_{\mu=1}^N \sum_{j=1}^m \left(\sum_{k=1}^l w_{jk} \left(\sum_{i=1}^n v_{ki} x_i^\mu - \theta_k \right)^P - \theta_j - d_j^\mu \right)^2$$

s. t. $|v_{ki}| \leq 1 \quad (2)$

其中 $|v_{ki}| \leq 1$ 是对 v_{ki} 的一种限制。由

$$y_j^\mu = \sum_{k=1}^l \left(\sum_{i=1}^n w_{jk} \left(\sum_{i=1}^n v_{ki} x_i^\mu - \theta_k \right)^P - \theta_j - d_j^\mu \right)^2$$

可知, 当 v_{ki} 同时缩小 k 倍时, 可由 w_{jk} 同时扩大 k^P 倍实现, 所以限制 $|v_{ki}| \leq 1$ 是合理的。问题(2)可用遗传算法进行求解^[3-6]。

对 w_{jk}, v_{ki} 的学习可用如下算法进行:

Step1: 用随机数初始化 $w_{jk}, v_{ki}, \theta_j, \theta_k$ 。其中 $i = 1, 2, \dots, n; k = 1, 2, \dots, L; j = 1, 2, \dots, m; \mu = 1, 2, \dots, N$ 。确定算法终止误差 ϵ_0 (或迭代代数), 确定多项式次数。 L 为隐层神经元数目, N 为训练模式总数。

Step2: 选择第 μ 个模式 x^μ 。

Step3: 计算隐单元输入:

$$\text{net}_k^\mu = \sum_{i=1}^n v_{ki} x_i^\mu - \theta_k$$

Step4: 计算隐层单元输出:

$$h_k^\mu = f_{\text{indec}}(\text{net}_k^\mu) = \left(\sum_{i=1}^n v_{ki} x_i^\mu - \theta_k \right)^P$$

Step5: 计算输出层单元输入:

$$\text{net}_j^\mu = \sum_{k=1}^l w_{jk} h_k^\mu - \theta_j$$

Step6: 计算隐层单元输出:

$$y_j^\mu = f_{\text{out}}(\text{net}_j^\mu) = \sum_{k=1}^l w_{jk} h_k^\mu - \theta_j$$

Step7: 调整权值 w_{jk} :

$$w_{jk}(t+1) = w_{jk}(t) + \beta \sum_{\mu=1}^N (d_j^\mu - y_j^\mu) h_k^\mu$$

Step8: 用保留最好解的遗传算法^[6](GA) 求解权值 v_{ki} , 即求解(2)式带约束的规划问题。

Step9: 误差小于 ϵ_0 , 则停止; 否则, 转 Step2。

2 PFANN 的非线性映射能力

由上一节可知, 网络的输出节点 $m = 1$ 时:

$$y_1^\mu = y^\mu(x_1^\mu, x_2^\mu, \dots, x_n^\mu) = \sum_{k=1}^l \left(w_{1k} \left(\sum_{i=1}^n v_{ki} x_i^\mu - \theta_k \right)^P - \theta_1 \right)$$

该式表明网络的输入和输出之间是一种 $R^n \rightarrow R$ 的函数关系。

PFANN 函数逼近要解决的问题是: 任何一连续函数 $y: R^n \rightarrow R$, 能否找到一个合适的 PFANN 来逼近它? 对三层 PFANN 函数逼近能力, 有如下结论:

定理 1 对三层 PFANN, 其隐层神经元数目可任意多, 隐层神经元特性为多项式函数, 输出层神经元特性为线性函数, 则单输出的 PFANN 可逼近任意的连续函数 $y: R^n \rightarrow R$ 。

证明: 设

$$\Psi_p = \text{span} \left\{ f \left(\sum_{i=1}^n w_i x_i \right) + \theta, \forall w_i \in R, \theta \in R \right\} =$$

$$\text{span} \left\{ \left(\sum_{i=1}^n w_i x_i + \theta \right)^P \right\}$$

$\forall w_i, w'_i, \theta$ 及 $\delta \neq 0$, 下式成立:

$$[(\sum_{i=1}^n w_i x_i - \theta)^P - (\sum_{i=1}^n w'_i x_i - \theta)^P] / \delta \in \Psi_p$$

则

$$\frac{\partial f(\cdot)}{\partial w_i^k} = P(\sum_{i=1}^n w_i x_i - \theta)^{P-1} x_i \in \Psi_p$$

所以:

$$\frac{\partial^k f(\cdot)}{\partial w_i^k} \in \Psi_p, \quad \frac{\partial^{k_1+k_2+\dots+k_s} f(\cdot)}{\partial w_{i_1}^{k_1} \partial w_{i_2}^{k_2} \dots \partial w_{i_s}^{k_s}} \in \Psi_p$$

从而,对 $0 \leq k_1 + k_2 + \dots + k_s < P$, 有:

$$x_{i_1}^{k_1} x_{i_2}^{k_2} \dots x_{i_s}^{k_s} P(P-1)\dots(P-k_1-\dots-k_s) \times (\sum_{i=1}^n w_i x_i - \theta)^{P-k_1-\dots-k_s} \triangleq \Psi \in \Psi_p \quad (3)$$

显然,上式中

$$P(P-1)\dots(P-k_1-\dots-k_s) \cdot (\sum_{i=1}^n w_i x_i - \theta)^{P-k_1-\dots-k_s}$$

是 $f(\cdot)$ 的低于 P 阶的导数,故可以找到 $\theta = \theta_0$, 满足:

$$P(P-1)\dots(P-k_1-\dots-k_s) \cdot (\sum_{i=1}^n w_i x_i - \theta_0)^{P-k_1-\dots-k_s} = C_0 \neq 0 \quad (4)$$

由(3)、(4)式知

$$x_{i_1}^{k_1} x_{i_2}^{k_2} \dots x_{i_s}^{k_s} = \frac{\Psi}{C_0} \in \Psi_p \quad (5)$$

由(5)式知: R^P 上的任意多项式均属于 Ψ_p .

根据 Weistrass 逼近定理^[7],任意的连续函数均可由多项式逼近,因此本定理结论正确. 证毕

对于多输出的情形,即 $y: R^n \rightarrow R^m$ 时,有如下更广泛的结论.

定理 2 对三层 PFANN,其隐层神经元数目可任意多,隐层神经元特性多项式函数,其输出层神经元特性为线性函数,则 PFANN 可逼近任意的连续函数 $g(X): R^n \rightarrow R^m$.

证明:由定理 1 可知,对于函数 $g(X)$ 的 m 个分量 g_1, g_2, \dots, g_m , 可分别找到一个单输出的三层多项式神经网络来逼近它.将这 m 个单输出的 PFANN 并联,可得到如图 1 所示 m 输出的 PFANN,显然这个多输出 PFANN 可逼近函数 $g(X)$.

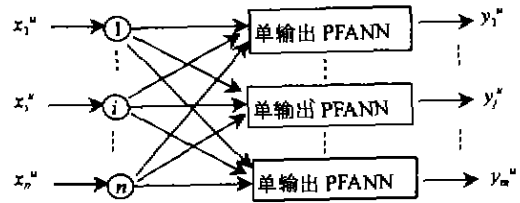


图 1 多输出 PFANN 结构图

3 PFANN 的稳定性

$E(t)$ 为第 t 次学习后的误差函数,设 $E_1(t)$ 为第 t 次学习时调整 w_{jk} 后的误差函数, $E_2(t)$ 为第 t 次学习时调整 v_{ki} 后的误差函数.

定理 3 对 PFANN,满足第一节算法时,在并行运行下,网络是稳定的.

证明:由(1)式知, $E(t)$ 是关于 w_{jk} 的凸函数,所以按梯度下降法调整 w_{jk} 时,有: $E_1(t) \leq E(t-1)$.

由学习算法知,调整 v_{ki} 的过程是保留最好解的遗传算法.由文献[6]知,在选择前保留最好解的遗传算法能以概率 1 收敛到全局最优解.从而有: $E_2(t) \leq E_1(t)$.

事实上, $E(t) = E_2(t)$, 所以: $E(t) \leq E(t-1)$, 即 $E(t)$ 单调递减.

因为 $E(t) \geq 0$, 由数列收敛的单调有界定理可知, $E(t)$ 是收敛的.

又因为 PFANN 可以逼近任意的连续函数,所以有: $\lim_{t \rightarrow +\infty} E(t) = 0$.

从而网络是稳定的.

证毕

4 PFANN 应用实例

为验证 PFANN 的性能,文中给出两类典型问题: 1) 异或(XOR)问题:代表离散分类问题, 2) 对函数 $f(x, y) = (\sin(x+y), \cos(x-y))$ 的逼近问题:代表连续逼近问题.文中给出了 PFANN 与采用具有固定步长和固定动量因子的 BP 算法的比较研究结果.实验中 PFANN 的初值在 $[-1.0, 1.0]$ 之间随机选取; BP 网络中隐层神经元的作用函数选用 S 型函数 $g(x) = 1/(1 + \exp(-x))$, 输出层选取线性作用函数.

例 1 XOR 问题

用 BP 算法和本文中的 PFANN 网络学习算法的实验结果见表 1(网络结构 2-3-1, 收敛精度为

0.000 001)。

表 1 BP 算法和 PANN 网络
学习算法对比实验结果(XOR 问题)

	次数	时间	备注
BP 算法	1 255	1*19	学习率 $\eta=0.9$, 动量因子 $\alpha=0.25$
PANN 网络 学习算法	24	1*25	GA 中每代选取样本 15 个, 每次迭代 50 次

两种算法的仿真迭代结果见图 2、图 3, 其中 ϵ 为迭代误差值, M 为迭代次数。

用 PFANN 算法求解 XOR 问题时, 其最优权值 v_{ki} 、 w_{kj} 分别为:

$$(v_{ki})_{3 \times 2} = \begin{bmatrix} -0.177\ 382\ 3 & 0.007\ 631\ 8 \\ -0.159\ 074\ 9 & -0.150\ 767\ 2 \\ -0.003\ 003\ 1 & 0.771\ 808\ 2 \end{bmatrix}$$

$$(w_{jk})_{1 \times 3} =$$

$$(-0.779\ 962\ 0, -2.929\ 130\ 7, 1.141\ 618\ 5)$$

多项式逼近次数为 2。

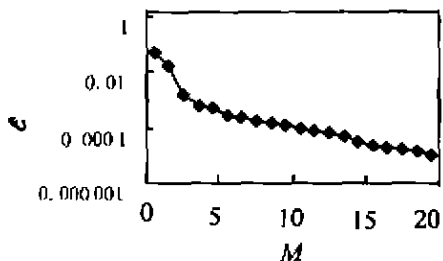


图 2 PFANN 对 XOR 问题的仿真

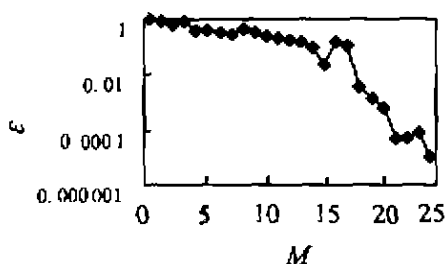


图 3 BP 算法对 XOR 问题的仿真

例 2 $R^2 \rightarrow R^2$ 的映射 $f(x, y) = (\sin(x + y), \cos(x - y))$, $x \in [-1, 1], y \in [-1, 1]$ 。

BP 算法和本文中的 PFANN 网络学习算法的实验结果见表 2(随机产生 10 个样本作为训练集网络结构 2-5-2, 收敛精度为 0.000 01)。

表 2 BP 算法和 PANN 网络
学习算法对比实验结果

	次数	时间	备注
BP 算法	10 396	23*31	误差函数有明显的震荡
PFANN 网络 学习算法	29	19*43	GA 中每代选取样本 25 个, 每次迭代 100 次

上述 BP 算法的结果是在不断调整学习率和动量因子得到的最好的结果, 在实践中应用 BP 算法时, 误差函数通常大都会出现震荡情形, 甚至不收敛。

5 结论

文中提出一种新型的多层前向神经网络——多项式网络。该网络具有三层结构, 其隐层采用 $f(x) = x^P$ 作为激活函数, 输出层采用线性函数, 该网络能充分逼近任意的多维连续函数。该网络输出层权值用最速下降法学习, 隐层权值用改进的遗传算法进行学习。该算法有两个优点: 1) 能收敛到全局最小, 从而克服 BP 算法易陷入局部极小的不足; 2) 误差函数是单调递减的, 从而克服 BP 算法出现震荡的情形。文中实例表明, 该网络相当稳定, 逼近的精度高, 学习速度较快, 是一种很有效的前向神经网络。与神经网络中隐层节点数的选择一样, P 的选择是难度较大的。通常 P 的选择将影响收敛速度, 其值可视问题的非线性程度而定, 实践中取 $P=2, 3$ 等一般能达到较好的效果。

参考文献:

- [1] 焦李成. 神经网络计算[M]. 西安: 西安电子科技大学出版社, 1996. 35-104.
- [2] 王科俊, 王克成. 神经网络建模、预报与控制[M]. 哈尔滨: 哈尔滨工程大学出版社, 1996. 37-89.
- [3] JENKINS W M. Toward structural optimization via the genetic algorithm[J]. Computer & Structure, 1991, 40(5): 1 321-1 327.
- [4] GALLETLY J E. An overview of Genetic algorithms[J]. Kybernetes, 1989, 21(6): 26-30.
- [5] HOLLAND J H. Genetic algorithm[J]. Scientific American, 1992, (4): 44-50.
- [6] 潘正君, 康立山. 演化计算[M]. 北京: 清华大学出版社, 1998. 37-43.
- [7] 蒋尔雄, 赵风光. 数值逼近[M]. 上海: 复旦大学出版社, 1996. 105-148.

(下转 98 页)

虽已解决,但在某些方面处理速度稍逊,例如在 VB 中大量使用 SendKeys 函数向 AUTOCAD 命令行传递键盘指令。目前在 AutoCAD 2000 版本中对该技术的支持得到了极大加强,在此新环境下,对 ACRPTS 系统进行改进可获得更好的运行效率。

参考文献:

- [1] 郭朝勇. AutoCAD R14 二次开发技术[M]. 北京:清华大学出版社. 1999.
- [2] HAROLD DAVIS. Visual Basic 6 Secrets[M]. California, USA: IDG Books Worldwide, Inc. 1998.

Development of House Architectural Area Computing and Plotting System

WANG Cheng-liang¹, HE Yu-lin²

(1. College of Computer Science and Engineering, Chongqing University, Chongqing 400044, China;

2. College of Mechanical Engineering, Chongqing University, Chongqing 400044, China)

Abstract: The exact computing of house architectural area and apportionment of area is strongly related to the actual interests of people. If it is done manually, there are many problem such as longer time spending, more workload, requiring detailed verification, nonstandard calculating report. Using ActiveX automation and ADO techniques, the house architectural area computing and plotting system was developed under Autocad R14 environment with VB 6.0 and Access database in order to solve above problem. The developed system can automatically compute house architectural area and apportionment of area for each floor of each building and can print the related data report and drawing. The system works well in many departments of house measurement. Several key technical problem of the system are discussed.

Key words: house architectural area; house measurement; database; second development

(责任编辑 吕赛英)

※ ※

(上接 79 页)

Polynomial Feedforward Artificial Neural Network

XIE Kai-gui¹, BAI Sen², ZHOU Jia-qi¹

(1. Electric Power System Research Institute of Electrical Engineering College, Chongqing University, Chongqing 400044, China; 2. Chongqing Communication College Dept. 1, Chongqing 400035, China)

Abstract: A new artificial neural network, i. e., polynomial feedforward artificial neural network(PFANN), which has three layers(input layer, hidden layer and output layer) is presented. The neural activation functions of hidden layer and output layer are $f(x) = x^p$ and linear function, respectively. The learning method of hidden-output layer weights is the steepest descent method and the one of input-hidden layer weights is genetic algorithm(GA). During the learning process, the error function is decreased monotonely. So the learning algorithm is convergent and the network, which can approximate to arbitrary continuous function, is stable. Some applying samples of PFANN, which reveals the remarkable quality, are proposed, too.

Key words: polynomial feedforward artificial neural network; the steepest descent method; genetic algorithm; algorithm convergence; approximation ability

(责任编辑 李胜春)