

基于QPSO和极限学习的离散过程神经网络及学习算法

刘志刚¹, 许少华^{1,2}, 李盼池¹

(1. 东北石油大学 计算机与信息技术学院, 黑龙江 大庆 163318;

2. 山东科技大学 信息科学与工程学院, 山东 青岛 266590)

摘要: 连续过程神经网络在权函数正交基展开时, 基函数个数无法有效确定, 因此逼近精度不高. 针对该问题, 提出一种离散过程神经网络, 使用三次样条数值积分处理离散样本和权值的时域聚合运算. 模型训练采用双链量子粒子群完成输入权值的全局寻优, 通过量子旋转门和非门完成种群进化. 局部使用极限学习, 通过 Moore-Penrose 广义逆计算输出权值. 以时间序列预测为例进行仿真实验, 结果验证了模型的有效性, 且训练收敛能力和逼近能力都有一定程度的提高.

关键词: 过程神经网络; 极限学习; 量子粒子群; Moore-Penrose 广义逆; 网络训练

中图分类号: TP183

文献标志码: A

Discrete process neural networks and learning algorithms base on QPSO and extreme learning machine

LIU Zhi-gang¹, XU Shao-hua^{1,2}, LI Pan-chi¹

(1. School of Computer and Information Technology, Northeast Petroleum University, Daqing 163318, China;

2. College of Information Science and Engineering, Shandong University of Science and Technology, Qingdao 266590, China. Correspondent: LIU Zhi-gang, E-mail: dqpilzg@163.com)

Abstract: When the weight functions of the continuous process neural network are expanded by orthogonal basis, the number of the basis function can not be determined effectively. The continuous process neural network has lower approach accuracy. Therefore, a discrete process neural network is presented. The three spline numerical integration is applied to deal with the aggregation of discrete samples and weights in time-domain. The double chain quantum particle swarm algorithm is used to the global optimization of model parameters. The evolution of the population is executed by the quantum rotation gate and quantum not gate. The extreme learning algorithm is applied to the local search and the output weights are computed by the Moore-Penrose inverse. The results of the simulation experiment based on the time sequence prediction verify the effectiveness of the proposed model, and show that the capability of training convergence and accurate approximation are improved at a certain degree.

Keywords: process neural network; extreme learning; quantum particle swarm; moore-penrose inverse; network training

0 引言

过程神经网络(PNN)是由He等^[1]提出的一种神经网络模型, 输入和权值是与时间有关的函数, 引入时间聚合算子实现输入信号的时间累积, 减小复杂非线性系统逼近时的时间累积误差. 文献[2]给出了模型连续性、逼近性等性质的理论证明. 由于PNN模型计算复杂, 文献[3]提出利用正交基对输入函数和权函数进行有限项展开, 根据正交特性简化时域聚合运算. 在此基础上, 文献[4-6]相继提出了自组织、反

馈、量子等多种过程神经网络模型. 目前PNN已成功应用于时间序列预测^[7]、故障诊断^[8]、非线性系统建模^[9]等实际问题, 并取得了较好的效果. 以上这些PNN的输入都是连续函数, 在学习过程中, 首先将离散样本采样点按时间拟合, 然后选取正交基函数展开, 最后利用正交特性简化映射关系, 因此存在以下问题:

1) 离散采样数据拟合时, 可能得不到合适的解析函数;

收稿日期: 2015-12-28; **修回日期:** 2016-03-13.

基金项目: 国家自然科学基金项目(61170132); 黑龙江省教育厅科学技术研究项目(12521369).

作者简介: 刘志刚(1979—), 男, 副教授, 博士生, 从事神经网络、进化算法的研究; 许少华(1962—), 男, 教授, 博士生导师, 从事过程神经网络等研究.

- 2) 解析函数拟合阶次过高, 易导致龙格震荡现象;
- 3) 输入函数正交基展开时, 基函数个数无法确定;
- 4) 现有的梯度下降训练算法对初值敏感, 不易收敛.

这些问题使 PNN 在应用方面受到很大限制. 极限学习机 (ELM) 是由 Huang 等^[10]在 2004 年提出的一种前馈神经网络快速学习算法, 通过理论分析和实验已经证明可以极大地提高神经网络的学习速度 (通常 10 倍以上)^[11]. 目前, ELM 理论体系虽不完善, 但是正处于迅速发展中, 已成为国内外的研究热点^[12-15].

综上所述, 本文提出一种离散过程神经网络 (DPNN), 不同于连续 PNN, 输入和权值是离散数据序列, 用向量代表每一个过程输入, 并提出采用三次样条数值积分处理离散输入与权值之间的时域聚合运算, 避免连续 PNN 因输入拟合带来的解析函数不存在、基函数个数无法确定等问题. 对于 DPNN 的训练, 在局部利用 ELM, 根据 Moore-Penrose 广义逆和隐层输出矩阵, 完成 DPNN 隐层输出权值快速计算. 对于 DPNN 隐层输入权值和阈值的训练, 由于参数较多, 利用双链结构的量子粒子群算法 (QPSO)^[16]进行全局寻优, 并根据目标函数的梯度信息, 对 QPSO 中量子旋转门中幅角的更新方式进行改进. 最后将改进后的 QPSO 与 ELM 相结合, 提出 QELM-DPNN 学习算法, 以解决 DPNN 的训练问题, 避免梯度下降法训练时易陷入局部极小的缺点, 提高了模型训练的稳定性. 最后进行仿真实验, 以 Mackey-Glass 混沌时间序列和太阳黑子预测为例验证了所提出方法的有效性, 结果表明, QELM-DPNN 具有较好的预测精度.

1 极限学习机 ELM

在单隐层前馈神经网络中, 设训练样本 $X = (x_1, x_2, \dots, x_N)$, 期望输出 $T = (t_1, t_2, \dots, t_N)$, 模型输入输出关系为

$$\sum_{i=1}^L \beta_i g(a_i x_j + b_i) = o_j, \quad j = 1, 2, \dots, N. \quad (1)$$

其中: $g(x)$ 为激活函数, a_i 为输入权重, β_i 为输出权重, b_i 为阈值. 模型学习目标是使代价函数 $E(w)$ 最小, $E(w)$ 为目标和期望输出的误差平方和, 有

$$E(w) = \sum_{j=1}^N \|o_j - t_j\|^2, \quad (2)$$

即存在 β_i 、 a_i 和 b_i , 式 (1) 可表示为

$$H\beta = T. \quad (3)$$

其中: H 为隐层输出矩阵, β 为隐层输出权重向量. 前

馈神经网络使用梯度下降法学习训练时, 存在初值敏感、易陷入局部最小等问题.

引理 1^[10] 对于任意 N 个不同样本 (x_i, t_i) 、 M 个隐层节点和一个可导激活函数 g , 前馈神经网络在 $a_i \in R^n$ 和 $b_i \in R^n$ 任意赋值的情况下, 存在可逆的隐层矩阵 H , 使方程组有精确解, 即代价函数 $E(w) = 0$.

引理 2^[10] 给定任意 N 个不同样本、任意小误差 $e > 0$ 和可导激活函数 g , 总存在一个包含 M 个隐层节点的前馈神经网络, 使得 $a_i \in R^n$ 和 $b_i \in R^n$ 在任意赋值的情况下, 误差 $E(w) \leq e$.

基于引理 1 和引理 2, Huang 等提出了 ELM 算法, 即前馈神经网络只要在权值和阈值赋值的情况下, 即可确定一个具有可逆特性的隐层输出矩阵, 完成系统逼近, 从而将神经网络训练转化为最小二乘问题, 得到隐层输出权值的最小二乘解, 有

$$\hat{\beta} = H^\dagger T, \quad (4)$$

其中 H^\dagger 为输出矩阵 H 的 Moore-Penrose 广义逆矩阵.

2 离散过程神经网络 DPNN

2.1 模型结构

离散过程神经元包括加权、聚合和激励输出 3 部分, 输入是离散的时间序列, 用向量代表每一个过程输入, 不同于普通神经网络, DPNN 的各维输入是不同时刻的采样值, 即为一个离散时间序列. 此时单个样本的表述形式不同于普通神经网络的向量, 而是一个矩阵. 设 $x_1(t_i), x_2(t_i), \dots, x_n(t_i)$ 为输入, $w_{ij}(t_i)$ 为权值, DPNN 的结构如图 1 所示.

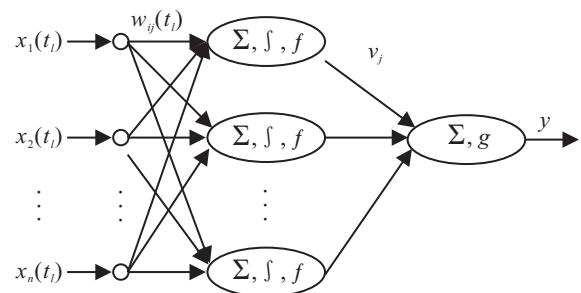


图 1 离散过程神经元网络

DPNN 的输出与输入的映射关系为

$$y = g\left(\sum_{j=1}^m v_j f\left(\sum_{i=1}^n \int_0^T w_{ij}(t) x_i(t) dt - \theta_j\right)\right). \quad (5)$$

对于 K 个样本 $(x_{k1}(t_i), x_{k2}(t_i), \dots, x_{kn}(t_i); d_k)$, $k = 1, 2, \dots, K, l = 1, 2, \dots, T$, 网络误差函数定义为

$$E = \sum_{k=1}^K \left(\sum_{j=1}^m v_j f\left(\sum_{i=1}^n \int_0^T w_{ij}(t) x_i(t) dt - \theta_j\right) - d_k\right)^2. \quad (6)$$

2.2 三次样条数值积分

为解决 DPNN 的时域聚合计算问题, 本文提出使用三次样条数值积分完成离散样本输入与权值之间的时域聚合运算. 设 $[a, b]$ 为积分区间, 记

$$x_k = a + kh, y_k = f(x_k),$$

$$k = 0, 1, \dots, n, h = (b - a)/n,$$

边界条件如下:

$$\begin{cases} y'_0 = \frac{1}{2h}(-3y_0 + 4y_1 - y_2), \\ y'_n = \frac{1}{2h}(3y_n - 4y_{n-1} + 3y_{n-2}), \end{cases} \quad (7)$$

三次样条函数可写为

$$S(x) = \sum_{j=1}^{n+1} c_j \Omega_3\left(\frac{x - x_0}{h} - j\right). \quad (8)$$

$\Omega_3(x)$ 为样条函数, 有

$$\Omega_3(x) = \begin{cases} 0, & |x| \geq 2; \\ 0.5|x|^3 - x^2 + \frac{2}{3}, & |x| \leq 1; \\ \frac{-|x|^3}{6} + x^2 - 2|x| + \frac{4}{3}, & |x| \in (1, 2). \end{cases} \quad (9)$$

参数 c_j 计算如下:

$$\begin{bmatrix} 4 & 2 & & & & \\ 1 & 4 & 1 & & & \\ & & \ddots & \ddots & \ddots & \\ & & & 1 & 4 & 1 \\ & & & & 2 & 4 \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_{n-1} \\ c_n \end{bmatrix} = \begin{bmatrix} 6y_0 + 2hy'_0 \\ 6y_1 \\ \vdots \\ 6y_{n-1} \\ 6y_n - 2hy'_n \end{bmatrix}, \quad (10)$$

$$c_{-1} = c_1 - 2hy'_0, \quad (11)$$

$$c_{n+1} = c_{n-1} - 2hy'_n. \quad (12)$$

被积函数 $f(x)$ 在 $[a, b]$ 上的积分可由三次样条函数 $S(x)$ 的数值积分获得, 有

$$I(f(x)) = \int_a^b S(x)dx =$$

$$\frac{h}{24}(c_{-1} + c_{n+1}) + \frac{h}{2}(c_0 + c_n) +$$

$$\frac{23h}{24}(c_1 + c_{n-1}) + h \sum_{k=2}^{n-2} c_k. \quad (13)$$

根据三次样条数值积分, 式(5)可写为

$$y = g\left(\sum_{j=1}^m v_j f\left(\sum_{i=1}^n I(w_{ij}(t)x_i(t)) - \theta_j\right)\right). \quad (14)$$

3 QELM-DPNN 学习算法

3.1 隐层输出权值计算

对于 K 个样本 $(x_{k1}(t_1), x_{k2}(t_1), \dots, x_{kn}(t_1); d_k)$, $k = 1, 2, \dots, K$, 当 DPNN 的隐层输入权值和阈值确定后, 隐层输出矩阵为

$$H = \begin{bmatrix} f(w_{i1}(t_1), x_{1i}(t_1), \theta_1) & \cdots & f(w_{im}(t_1), x_{1i}(t_1), \theta_m) \\ \vdots & \ddots & \vdots \\ f(w_{i1}(t_1), x_{Ki}(t_1), \theta_1) & \cdots & f(w_{im}(t_1), x_{Ki}(t_1), \theta_m) \end{bmatrix}. \quad (15)$$

在矩阵 H 中, 第 k 行是训练样本中第 k 个样本在每个隐层离散过程神经元的输出向量; 第 j 列是对于输入样本中的每个样本, 在第 j 个隐层离散过程神经元的输出向量. 根据式(3)和(14), DPNN 输入输出映射关系为

$$HV = O. \quad (16)$$

其中: $O = [d_1, d_2, \dots, d_K]_{K \times 1}^T$ 为期望输出, $V = [v_1, v_2, \dots, v_m]_{m \times 1}^T$ 为隐层输出权值.

根据 ELM, 当隐层输入权值和阈值确定后, DPNN 的训练转化为最小二乘问题. 由式(16), DPNN 隐层输出权值 $V = H^\dagger O$.

记 $U = H^T H$, 当 U 为非奇异矩阵时, 可使用正交投影法直接计算 H^\dagger , 有

$$H^\dagger = (H^T H)^{-1} H^T, \quad (17)$$

$$V = (H^T H)^{-1} H^T O. \quad (18)$$

当 U 为奇异矩阵时, 可使用通用性更好的奇异值分解(SVD)法求解 H 的 Moore-Penrose 广义逆 H^\dagger . 利用 SVD 将 H 分解为

$$H = P \begin{bmatrix} \Sigma_r & 0 \\ 0 & 0 \end{bmatrix} Q^T. \quad (19)$$

其中: P 为 $m \times m$ 的正交矩阵, Q 为 $n \times n$ 的正交矩阵, r 为矩阵 H 的秩, Σ_r 为由 U 的特征值平方根对角排列组成的矩阵. H^\dagger 计算如下:

$$H^\dagger = Q \begin{bmatrix} \Sigma_r^{-1} & 0 \\ 0 & 0 \end{bmatrix} P^T. \quad (20)$$

此时, DPNN 隐层输出权值为

$$V = Q \begin{bmatrix} \Sigma_r^{-1} & 0 \\ 0 & 0 \end{bmatrix} P^T H^T O. \quad (21)$$

3.2 QELM-DPNN 学习算法

在 ELM 算法中, 通过隐层输出矩阵和 Moore-Penrose 广义逆, 可以一次求解出隐层输出权值向量, 但这种随机赋值造成模型训练稳定性较差. 同时, DPNN 输入权值的训练参数较多, 为提高模型训练稳定性和计算能力, 对于输入权值的训练, 选用文献[16]提出的双链结构 QPSO 算法. 同时, 为提高搜索效率, 加快进化进程, 本文提出根据目标函数的梯度信息对量子旋转门幅角的更新方式进行改进. QPSO 的全局迭代寻优与 ELM 局部快速计算相结合的 QELM-DPNN 学习算法步骤如下.

Step 1: 种群参数初始化. 设定种群规模 Popsizel, 最大进化代数 Max_I, 误差精度 ε , 转角初值 θ .

Step 2: 粒子编码. 设 DPNN 的输入输出结构为 $n - m - o$, 单个样本输入序列长度为 L , 利用 QPSO 进行输入权值离散序列和隐层阈值优化时, 每个粒子的维度是 $d = nmL + m$, 具体的双链编码结构如下:

$$p_i = \begin{bmatrix} \cos \theta_{i1} \\ \sin \theta_{i1} \end{bmatrix} \begin{bmatrix} \cos \theta_{i2} \\ \sin \theta_{i2} \end{bmatrix} \cdots \begin{bmatrix} \cos \theta_{i\omega} \\ \sin \theta_{i\omega} \end{bmatrix}. \quad (22)$$

其中: $\theta_{ij} = 2\pi \times r$, $i = 1, 2, \dots, n$, $j = 1, 2, \dots, \omega$, $r \in (0, 1)$ 为随机数, ω 为问题空间维数 ($\omega = 1, 2, \dots, nmL + m$). 每个量子比特代表一个优化求解的 DPNN 参数, 每个粒子分别对应量子态 $|0\rangle$ 和 $|1\rangle$ 的概率幅 p_{ic} 和 p_{is} , 从而占据问题搜索空间中的两个位置, 有

$$p_{ic} = (\cos \theta_{i1}, \cos \theta_{i2}, \dots, \cos \theta_{in}), \quad (23)$$

$$p_{is} = (\sin \theta_{i1}, \sin \theta_{i2}, \dots, \sin \theta_{in}). \quad (24)$$

Step 3: 解空间变换, 完成单位空间到实际问题解空间转换. 记粒子 p_j 上第 i 个量子位为 $[\alpha_i^j, \beta_i^j]^T$, 问题空间为 $[a_i, b_i]^n$, 则转换后的解空间变量为

$$X_{ic}^j = \frac{b_i(1 + \alpha_i^j) + a_i(1 - \alpha_i^j)}{2}, \quad (25)$$

$$X_{is}^j = \frac{b_i(1 + \beta_i^j) + a_i(1 - \beta_i^j)}{2}. \quad (26)$$

Step 4: 计算 DPNN 的隐层输出矩阵 H . 对 $H^T H$ 进行 SVD 分解, 计算广义逆矩阵 $(H^T H)^\dagger$, 根据 ELM 理论和式 (21) 计算 DPNN 输出权值向量.

Step 5: 由式 (6) 定义的 DPNN 误差函数, 取其倒数作为适应度函数 $f(X)$, 计算当代种群每个量子粒子的适应度.

Step 6: 完成每个粒子的最优位置和种群全局最优位置的更新.

Step 7: 采用量子旋转门改变每个粒子上的量子比特相位角, 完成种群更新. 量子旋转门中幅角变化为

$$\Delta \theta_{ij}(t+1) = \Delta \theta_0 \times \exp\left(-\frac{\nabla f(X_i^j) - \nabla f_j \min}{\nabla f_j \max - \nabla f_j \min}\right). \quad (27)$$

其中: $\nabla f(X_i^j)$ 为适应度函数 $f(X)$ 在 X_i^j 处的梯度值, $\Delta \theta_0$ 为转角初值, $\nabla f_j \max$ 和 $\nabla f_j \min$ 为

$$\nabla f_j \max = \max \left\{ \left| \frac{\partial f(X_1)}{\partial X_1^j} \right|, \dots, \left| \frac{\partial f(X_1)}{\partial X_1^j} \right| \right\}, \quad (28)$$

$$\nabla f_j \min = \min \left\{ \left| \frac{\partial f(X_1)}{\partial X_1^j} \right|, \dots, \left| \frac{\partial f(X_1)}{\partial X_1^j} \right| \right\}. \quad (29)$$

该幅角的变化方式与文献 [16] 的单位圆几何变换方式不同, 在幅角的转角步长中引入适应度函数的梯度变化信息, 若粒子在搜索点处的适应度函数变化较大, 则幅角变化较小, 否则加大幅角变化. 该方式可以实现“平坦之处迈大步, 陡峭之处迈小步”的自适应搜索方式, 使得每个粒子根据当前搜索点的变化趋势决定自身的幅角变化, 更有利于得到问题最优解. 更新后的量子幅角为

$$\theta_{i1}(t+1) = \theta_{i1}(t) + \Delta \theta_{i1}(t+1). \quad (30)$$

Step 8: 采用量子非门完成种群的变异. 设定变异概率 p_m , 为每个粒子设定一个 $(0, 1)$ 之间的随机数 r_i , 循环判断 $r_i < p_m$, 条件成立则粒子被选中. 对每个选中的粒子再随机选择 $\lceil n/2 \rceil$ 个量子位, 使用量子非门进行概率幅的变换, 有

$$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \cos \theta_{ij} \\ \sin \theta_{ij} \end{bmatrix} = \begin{bmatrix} \cos(\theta_{ij} + \pi/2) \\ \sin(\theta_{ij} + \pi/2) \end{bmatrix}. \quad (31)$$

Step 9: 返回 Step 3 循环计算, 直到达到最大进化代数 max_I 或者满足小于误差精度 ε , 训练结束.

4 仿真实验

为检验所提出的 QELM-DPNN 的有效性、算法训练的稳定性 and 模型的逼近能力, 将其与 OBPNN^[3]、SSDPNN^[17]、未经优化的 ELM-DPNN 进行对比分析.

4.1 Mackey-Glass 混沌时间序列预测

Mackey-Glass 混沌时间序列可由如下方程产生:

$$\frac{dx}{dt} = \frac{\alpha x(t-\tau)}{1 + x(t-\tau)^{10}} - \beta x(t). \quad (32)$$

设 $\alpha = 0.2$, $\beta = 0.1$, $\tau = 17$, x 的初始值为 $x(0) = 0.55$, 实验采用四阶 Runge-Kutta 方法, 以 0.1 为步长, 得到一组 206 个时间序列数据, 将其作为模型的学习测试数据集, 记 $\{x_k\}_{k=1}^{206}$.

1) 学习样本构造. 为体现 Mackey-Glass 数据的时间累积效应, 减小离散样本带来的训练误差, 在训练之前首先将离散数据进行拟合, 按照 $\{x_k\}_{k=1}^{206}$ 中的数据先后关系, 将连续的 6 个拟合成一个时变函数并作为网络输入, 将第 7 个作为该函数的输出, 以此类推, 共计可以得到 200 组样本数据. 随机选择其中的 100 组作为模型的训练样本, 选择剩余的 100 组作为测试样本检验模型的预测能力.

2) 参数设置. 由于 PNN 结构较为复杂, 目前关于隐层节点数量的确定还较少见到相关文献研究, 经过多次实验, 最终确定模型的输入输出结构为 1-10-1. 最大迭代数 max_I = 5000, 误差精度 $\varepsilon = 0.001$. 其中: QELM-DPNN 的粒子群规模 Popsizel = 100, 每个染色体的量子位数 70 位, 转角初值 $\theta = 0.01\pi$. OBPNN 的正交基函数选用勒让德基函数, 初始项

数 $L = 16$. OBPNN 和 SSDPNN 的学习速度 $\alpha = \beta = \gamma = 0.5$, 惯性系数 $\eta = 0.1$.

3) 实验对比. 对 QELM-DPNN 的学习能力(迭代步数、学习时间、收敛次数)和预测能力(相关系数、确定性系数、均方误差)进行检验, 其中相关系数、确定性系数越接近 1, 均方误差越接近 0, 预测效果越好, 模型逼近能力越强. 针对 4 种模型和 100 组训练样本, 各进行 10 次仿真实验, 训练和预测情况的指标对比如表 1 所示, 100 组测试样本的预测值和实际值的对比如图 2 所示. 从实验结果中可以看出, ELM-DPNN 的学习速度比 OBPNN 和 SSPNN 有了很大提高, 达到 7~9 倍左右, 但稳定性很差, 仅是 30%.

QELM-DPNN 的训练时间最长, 但稳定性相对较高, 达到 90%, 并且预测时对系统的逼近能力相对于其他 3 种模型, 有一定程度的提升, 确定性系数达到 0.942 87.

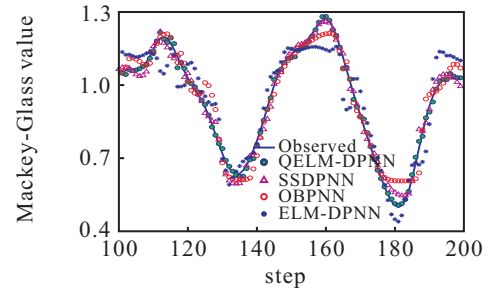


图 2 Mackey-Glass 序列预测结果与实际结果对比

表 1 4 种模型的训练和预测结果对比

数据集	模型	训练结果			预测结果		
		平均步数	平均时间 /s	收敛次数	相关系数	确定性系数	均方误差
Mackey-Glass	OBPNN	4 361	78.495 3	7	0.971 14	0.785 36	0.048 13
	SSDPNN	3 973	61.371 4	6	0.989 25	0.895 91	0.030 91
	ELM-DPNN	—	9.391 0	3	0.848 21	0.640 32	0.264 76
	QELM-DPNN	814	145.315 7	9	0.996 95	0.942 87	0.020 11
太阳黑子	OBPNN	4 594	90.164 1	6	0.973 09	0.747 16	11.920 42
	SSDPNN	4 281	79.622 9	7	0.985 26	0.834 74	18.661 8
	ELM-DPNN	—	10.985 3	2	0.900 60	0.607 96	23.683 4
	QELM-DPNN	736	178.549 4	9	0.996 31	0.919 46	4.384 52

4.2 太阳黑子预测

太阳黑子是太阳活动水平的基本参数, 与地磁扰动、电离层电子浓度等有密切关系, 因此太阳黑子的预报是空间预报的一项重要内容, 从而为通信、导航、定位提供重要的参考信息. 太阳黑子的预测问题在统计学中常被用于检测和评价各种模型的预测能力. 本文选用 SIDC 公布的 1700~2007 年 308 年太阳黑子年均值时间序列数据, 图 3 是这 308 年的数据分布特征. 由图 3 可见, 太阳黑子年均值时间序列数据存在很高的非线性, 为提高预测精度, 本文使用过程神经网络进行预测.

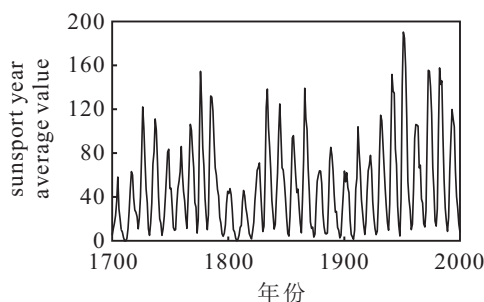


图 3 1700~2007 年太阳黑子原始数据

1) 学习样本构造.

从 1700 年开始, 选用连续 10 年数据进行拟合, 期望输出是第 11 年的数据, 以此类推, 共计可得到样本 298 组, 其中前 248 组为训练样本, 后 50 组为测试样本, 检验模型的预测能力. 这种样本构造过程的目的是利用 1700~1957 年的数据预测 1958~2007 年太阳黑子的年均值走势.

2) 参数设置.

经过多次实验确定 4 种模型的拓扑结构为 1-12-1 的结构. 其中: QELM-DPNN 最大迭代数 $Max_I = 5000$, 误差精度 $\varepsilon = 0.005$, 每个染色体的量子位数 132 位, 转角初值 $\theta = 0.01 \pi$.

3) 实验对比.

针对 OELM-DPNN、OBPNN、SSDPNN 和 ELM-DPNN 进行 10 次仿真实验, 训练和预测情况的指标对比如表 1 所示, 50 组测试样本的预测对比如图 4 所示. 从预测情况看, QELM-DPNN 除训练时间较长外, 模型训练的收敛能力和逼近能力都有一定程度的提高.

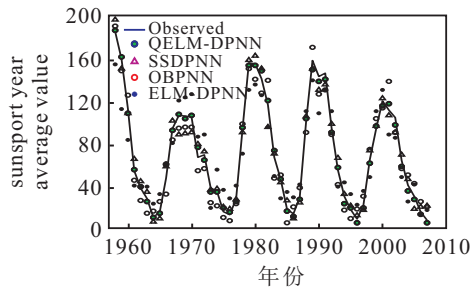


图4 太阳黑子预测结果与实际结果对比

4.3 仿真实验分析

模型训练情况分析. ELM-DPNN: 训练速度有明显的优势, 一次迭代即可完成模型输出权值的快速求解. 训练速度很快, 但是劣势也非常明显, 训练稳定性差, 在10次仿真实验过程中, 针对两个时间序列, 仅有2或3次的收敛, 很难在实际中应用. OBPNN: 训练时间较长, 主要由于训练中需要对离散序列进行基函数拟合, 拟合后的函数和权函数再利用正交基展开, 计算量增大. SSDPNN: 用离散输入代表过程输入, 无须函数拟合, 通过二次样条数值积分完成时域聚合, 计算量较小, 因此学习速度优于OBPNN. 此外, OBPNN和SSDPNN的训练稳定性并不理想, 仅达到70%左右, 主要由于训练时使用了梯度下降法, 训练参数较多, 对初值敏感, 易陷入局部极小值. QELM-DPNN: 全局优化利用QPSO进行迭代, 局部通过ELM快速求解输出权值. 双链结构保证了种群进化一次可以得到多组可行解, 扩大了搜索空间, 因此迭代步数小于SSDPNN和OBPNN. 由于量子旋转门和非门进化机制, 避免了梯度下降易陷入局部最小的缺点, 训练收敛能力较强, 达到90%. 但在训练时间上, 种群进化较为耗时, 一次迭代需要对多组可行解进行操作, 而OBPNN和SSDPNN的一次迭代只计算一组可行解. 设种群规模为 A , 粒子维数为 B , 最大迭代次数为 C , QELM-DPNN的时间复杂度为 $O(ABC)$, 空间复杂度为 $O(AB)$, OBPNN和SSDPNN的时间复杂度为 $O(BC)$, 空间复杂度为 $O(B)$. 所以QELM-DPNN迭代过程的计算量要大于OBPNN和SSDPNN, 学习时间最长.

模型预测能力分析. OBPNN: 内部利用正交基函数展开权函数和输入函数, 根据正交特性简化模型映射关系, 但根据基函数展开理论, 理论上需要基函数项数为无穷多个, 采用有限数目的基函数必然会影响模型精度, 因此模型的逼近精度较低. SSDPNN: 虽然可以直接处理离散过程输入, 但内部使用二次样条函数完成时域聚合, 模型较为简单, 可调参数较少, 虽然计算效率较高, 但是逼近精度略低. QELM-DPNN: 通过三次样条数值积分完成离散输入和权值序列的时域聚合运算, 并且在离散输入数据序列首尾各补充了

一个插值点, 模型逼近能力相对于OBPNN和SSDPNN有一定程度的提高. 同时, 通过量子旋转门和量子非门的协同作用, 利用适应度函数的梯度信息完成幅角更新, 能够精确捕捉输入与输出之间复杂的非线性映射关系, 也提高了QELM-DPNN逼近能力.

最后, QELM-DPNN的逼近能力和训练收敛能力要高于其他3种模型, 但模型的训练和计算效率最低, 较为耗时. 根据无免费午餐定理, QELM-DPNN是通过牺牲计算效率来换取逼近精度的提高.

5 结论

本文在连续过程神经网络的基础上, 提出了一种离散过程神经网络, 将离散序列直接作为过程输入, 模型内部使用三次样条数值积分处理样本的时域聚合运算, 避免了连续过程神经网络在正交基展开时的基函数个数、函数拟合中的不足. 模型训练时, 为扩大搜索空间, 采用双链结构的量子粒子群进行全局寻优, 同时对量子旋转门幅角的更新方式进行了改进, 由单位圆中的几何变换转变为根据目标梯度自适应地计算幅角变化. 在局部的输出权值计算时, 采用极限学习, 根据Moore-Penrose广义逆计算输出权值. 仿真实验表明, 极限学习机理论可以用于过程神经网络的训练, 学习速度快, 但是必须使用适当的方法进行辅助, 以保证模型的稳定性. 同时, QELM-DPNN的训练时间虽然较长, 但算法有很好的训练稳定性, 模型逼近能力也较强.

参考文献(References)

- [1] He X G, Liang J Z. Procedure neural networks[C]. Proc of Conf on Intelligent Information. Beijing: Publishing House of Electronic Industry, 2000: 143-146.
- [2] 许少华, 何新贵. 关于连续过程神经网络的一些理论问题[J]. 电子学报, 2006, 34(10): 1838-1842.
(Xu S H, He X G. Some theoretical issues on continuous process neural networks[J]. Acta Electronica Sinica, 2006, 34(10): 1838-1842.)
- [3] 许少华, 何新贵. 基于函数正交基展开的过程神经网络学习算法[J]. 计算机学报, 2004, 27(5): 645-650.
(Xu S H, He X G. Learning algorithms of process neural networks based on orthogonal function basis expansion[J]. Chinese J of Computers, 2004, 27(5): 645-650.)
- [4] 许少华, 何新贵, 李盼池. 自组织过程神经网络及其应用研究[J]. 计算机研究与发展, 2003, 40(11): 1612-1615.
(Xu S H, He X G, Li P C. Research and applications of self-organization process neural networks[J]. J of Computer Research and Development, 2003, 40(11): 1612-1615.)
- [5] 何新贵, 许少华. 一类反馈过程神经网络模型及其学习算法[J]. 自动化学报, 2004, 30(6): 801-806.

- (He X G, Xu S H. A feedback process neural network model and its learning algorithm[J]. *Acta Automatica Sinica*, 2004, 30(6): 801-806.)
- [6] 李盼池, 王海英, 戴庆. 量子过程神经网络模型算法及应用[J]. *物理学报*, 2012, 61(16): 24-32.
(Li P C, Wang H Y, Dai Q. Quantum process neural networks model algorithm and applications[J]. *Acta Physica Sinica*, 2012, 61(16): 24-32.)
- [7] 丁刚, 钟诗胜. 基于过程神经网络的时间序列预测及其应用研究[J]. *控制与决策*, 2006, 21(9): 1037-1041.
(Ding G, Zhong S S. Time series prediction based on process neural networks and its applications[J]. *Control and Decision*, 2006, 21(9): 1037-1041.)
- [8] Ding G, Lin L. Elman-style process neural network with application to aircraft engine health condition monitoring[C]. *Proc of the 8th Int Symposium on Neural Networks*. New York: IEEE, 2011: 484-494.
- [9] 肖红, 许少华. 基于数值积分的多聚合过程神经网络算法[J]. *信息与控制*, 2013, 42(5): 608-612.
(Xiao H, Xu S H. Multi-aggregation process neural networks algorithm based on numerical integration[J]. *Information and Control*, 2013, 42(5): 608-612.)
- [10] Huang G B, Zhu Q Y, Siew C K. Extreme learning machine: A new learning scheme of feedforward neural networks[C]. *Proc Int Joint Conf on Neural Networks*. Budapest: IEEE, 2004: 985-990.
- [11] Huang G B. Extreme learning machine for regression and multiclass classification[J]. *IEEE Trans on Systems*, 2012, 42(2): 513-516.
- [12] Liyanaarachchi L C K, Zhou H M, Huang G B. Representational learning with extreme learning machine for big data[J]. *IEEE Intelligent System*, 2013, 28(6): 31-34.
- [13] 邓万宇, 郑庆花. 神经网络极速学习方法研究[J]. *计算机学报*, 2010, 33(2): 279-286.
(Deng W Y, Zheng Q H. Research on extreme learning of neural networks[J]. *Chinese J of Computers*, 2010, 33(2): 279-286.)
- [14] 贺彦林, 王晓, 朱群雄. 基于主成分分析——改进的极限学习机方法的精对苯二甲酸醋酸含量软测量[J]. *控制理论与应用*, 2015, 32(1): 80-85.
(He Y L, Wang X, Zhu Q X. Modeling of acetic acid content in purified terephthalic acid solvent column using principal component analysis based improved extreme learning machine[J]. *Control Theory & Application*, 2015, 32(1): 80-85.)
- [15] 王超, 王建辉. 改进式混合增量极限学习机算法[J]. *控制与决策*, 2015, 30(11): 1982-1985.
(Wang C, Wang J H. Improved hybrid incremental extreme learning machine algorithm[J]. *Control and Decision*, 2015, 30(11): 1982-1985.)
- [16] 刘志刚, 杜娟. 一种求解非线性方程组的量子粒子群算法[J]. *微计算机应用*, 2011, 32(2): 31-37.
(Liu Z G, Du J. A quantum particle swarm optimization algorithm for solving nonlinear equations[J]. *Microcomputer Applications*, 2011, 32(2): 31-37.)
- [17] 李盼池. 基于样条插值函数的离散过程神经网络训练[J]. *计算机应用研究*, 2011, 28(1): 75-77.
(Li P C. Discrete process neural networks training based on spline function[J]. *Application Research of Computers*, 2011, 28(1): 75-77.)

(责任编辑: 郑晓蕾)