

### 3.5.1 基于粒子群算法的神经网络学习算法

PSO 作为一种新兴的进化算法,其收敛速度快、鲁棒性能好、全局搜索能力强,且不需要借助问题本身的特征信息。本文将 PSO 与神经网络结合在一起,用 PSO 算法来优化神经网络的连接权值<sup>[55-56]</sup>,可以较好地克服神经网络容易陷入局部极小的问题,不仅能发挥神经网络的泛化能力,而且能够提高神经网络的收敛速度和学习能力。

假设对一个有  $M$  个输入节点,  $Q$  个隐含节点和  $L$  个输出节点的神经网络如下图所示,  $w_{ij}$  为输入层神经元  $j$  与隐含层神经元  $i$  之间的连接权值;  $\theta_i$  为隐含层神经元  $i$  的阈值;  $w_{ik}$  为输出层神经元  $k$  与隐含层神经元  $j$  之间的连接权值;  $\theta_k$  为输出层神经元  $k$  的阈值。

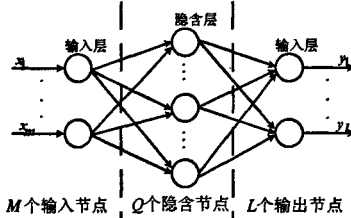


图 3.5.1 神经网络结构

假定其中的某一个样本  $p$  的输入/输出模式对  $\{X_p\}$  和  $\{T_p\}$  对网络进行训练,隐含层的第  $i$  个神经元在样本  $p$  作用下的输入为:

$$net_i^p = \sum_{j=1}^M O_j^p - \theta_i = \sum_{j=1}^M w_{ij} x_j^p - \theta_i \quad (i=1,2,\dots,q) \quad (3.5.1)$$

$x_j^p$  和  $O_j^p$  分别为输入节点  $j$  在样本  $p$  作用时的输入和输出,对输入节点而言两者相当;  $w_{ij}$  为输入层神经元  $j$  与隐含层神经元  $i$  之间的连接权值;  $\theta_i$  为隐含层神经元  $i$  的阈值;  $M$  为输入层的节点数。

则隐含层第  $i$  个神经元的输出为:  $O_i^p = g(net_i^p)$  ( $i=1,2,\dots,q$ ), 式中  $g(x)$  为激活函数。

对于一个仅有三层的神经网络来说,隐含层的输出将作为输出层的输入,故隐含层第  $i$  个神经元的输出  $O_i^p$  将通过权系数向前传播到输出层第  $k$  个神经元并作为它的输入之一,而输出层第  $k$  个神经元的总输入为:

$$net_k^p = \sum_{i=1}^q w_{ik} O_i^p - \theta_k \quad (k=1,2,\dots,L) \quad (3.5.2)$$

式中  $w_{ik}$  为隐含层神经元  $i$  与输出层神经元  $k$  之间的连接权值;  $\theta_k$  为输出层神经元  $k$  的阈值;  $q$  为输出层的节点数。

则输出层第  $k$  个神经元的输出为:  $O_k^p = g(\text{net}_k^p)$  ( $k=1,2,\dots,L$ )

对每一个样本  $p$  的输入模式对的二次型误差函数为:  $J_p = \frac{1}{2} \sum_{k=1}^L (t_k^p - O_k^p)^2$  则系统

对所有  $N$  个训练样本的总误差函数为:

$$J = \sum_{p=1}^N J_p = \frac{1}{2} \sum_{p=1}^N \sum_{k=1}^L (t_k^p - O_k^p)^2 \quad (3.5.3)$$

式中,  $N$  为模式样本对数目;  $L$  为网络输出节点数。

基于粒子群神经网络的学习算法实际上就是一个优化问题。优化问题<sup>[57]</sup>就是通常所说的最优化问题,但在实际中最优解通常是难以求得,或者在某些情况下根本不存在。为了更加符合实际情况,本文中把把这类问题统一称作优化问题。

优化问题是指在给定指标和元件、参数的允许取值范围条件下,确定一组独立的设计参数,使系统达到最佳性能。系统性能的优劣通常用一个关于设计参数的函数来描述,该函数称为“目标函数”;待定的设计参数称为“优化变量”;参数范围和未包含在目标函数中的一些设计指标称之为构成优化变量的“约束条件”。

不失一般性,假设目标函数为  $f(X)$ , 其中  $X = [x_1, x_2, x_3, \dots, x_n]^T$  为优化变量,寻求系统的最佳性能,在数学上通常是指求取最小化目标函数  $\min f(X)$  或者是最大化目标函数  $\max f(X)$ , 而最大化目标函数可以通过求取  $\min[-f(X)]$  来实现。可见,所有优化问题都可以归结为求取  $\min f(X)$ 。因此,优化问题的数学模型可以描述为:

$$\begin{cases} f(X^*) = \min_{X \in R^*} f(X) \\ S_i(X) < 0, i=1,2,\dots,k \end{cases} \quad (3.5.4)$$

其中,  $X^*$  为优化问题的最优解,  $f(X^*)$  为优化问题的最优目标函数,  $S_i(X)$  为优化问题的约束条件。

本文取神经网络的输出均方误差  $J$  作为目标函数为  $f(X)$ , 神经网络的权值与阈值作为优化变量, 权值与阈值的取值范围作为优化问题的约束条件  $S_i(X)$ , 最优解  $X^*$  就是使得神经网络的输出均方误差  $J$  的输出值最小。

利用 PSO 学习算法来对神经网络进行训练步骤如下:

**Step 1:** 确定神经网络的结构—输入层有  $M$  个节点, 隐含层有  $Q$  个节点和输出层有  $L$  个节点;

**Step 2:** 根据网络的结构, 把网络的权值和阈值确定粒子的维度, 即:

$X = [w_{ij}, \theta_j, w_{jk}, \theta_k]$ , 其中  $j=1,2,\dots,n$ ,  $i=1,2,\dots,q$ ,  $k=1,2,\dots,l$ ;

**Step 3:** 初始化粒子群算法的参数, 包括: 种群大小  $Pop\_Size$ 、粒子活动范围  $Region\_Size$ 、惯性权重  $\omega$ 、加速常数  $C_1$  和  $C_2$ 、最大速度  $V_{\max}$ 、最大迭代次数  $Max\_Gen$

和精度要求  $\varepsilon$ 、粒子的初始位置  $Init\_Pos$  与初始速度  $Init\_Vec$ 。

Step 4: 计算种群中每一个粒子的适应值  $f(X)$ , 确定每个当前粒子的最好位置  $pbest$  和种群当前全局最优位置  $gbest$ ;

Step 5: 计算神经网络的输出均方误差  $J$ ;

Step 6: 根据公式(3.4.1)和(3.4.2)更新种群中每一个粒子的速度  $V_i$  与位置  $X_i$ ;

Step 7: 判断  $V_i$  是否大于  $V_{max}$ ,  $X_i$  是否超出  $Region\_Size$ ;

Step 8: 更新粒子当前最好位置  $pbest$  和种群当前全局最优位置  $gbest$ ;

Step 9: 计算均方误差  $J$  是否满足精度要求, 若  $J < \varepsilon$ , 则转入 Step 12;

Step 10: 调整惯性权重  $\omega$ ;

Step 11: 判断是否超过最大迭代次数  $Max\_Gen$ , 若  $n < Max\_Gen$ , 重复 Step 6;

Step 12: 结束训练学习。

算法流程图如下:

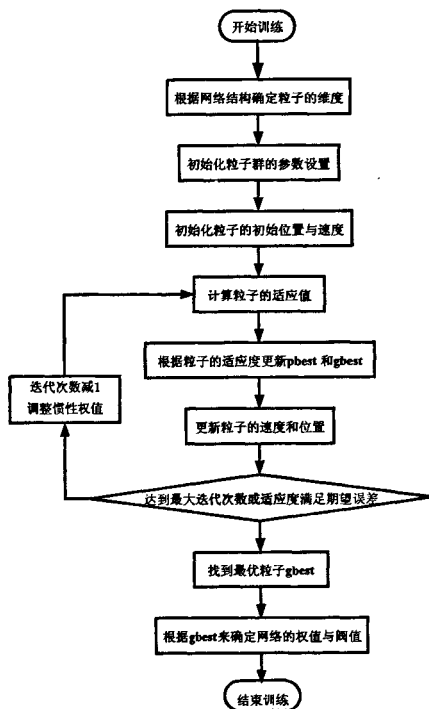


图 3.5.2 PSO 算法训练神经网络流程