

BP 神经网络

模型六的建立

神经网络结构图为：

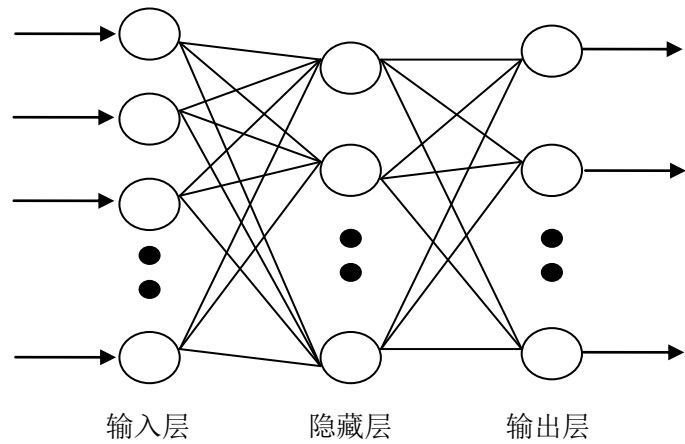


图 2 神经网络结构图

BP 神经网络的工作过程主要分为两个阶段：第一个阶段是学习期，此时各计算单元状态不变，各连线上的权值可通过学习来修改；第二阶段是工作期，此时各连接权固定，计算单元状态变化，以达到某种稳定状态^[6]。具体流程如下：

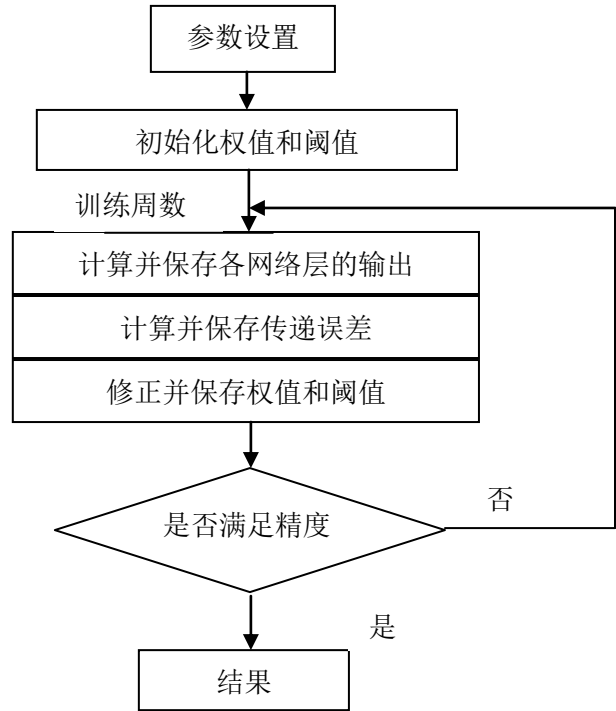


图 3 BP 神经网络流程图

设置各权重和阈值的初始值 $w_{ji}^{(l)}[0], \theta_j^{(l)}[0] (l = 0, 1, \dots, L)$ 为小随机数，输入训练样本 (I_q, d_q) ，对每个样本进行一下几个步骤：

Step1：计算各网络层的实际输出

$$x^{(l)} = f(s^{(l)}) = f(w^{(l)} x^{(l-1)} + \theta^{(l)}) \quad (30)$$

Step2: 计算训练误差

$$\tau_j^{(l)} = (d_{qj} - x_j^{(l)})f'(s_j^{(l)}), \text{ 输出层}$$

$$\tau_j^{(l)} = f'(s_j^{(l)}) \sum_{k=1}^{n_{l+1}} \tau_k^{(l+1)} w_{kj}^{(l+1)}, \text{ 隐含层和输入层}$$

Step3: 修正权值和阈值

$$w_{ji}^{(l+1)}[k+1] = w_{ji}^{(l)}[k] + \mu \tau_j^{(l)} x_i^{(l-1)} + \eta w_{ji}^{(l)}[k] - w_{ji}^{(l)}[k-1] \quad (31)$$

$$\theta_j^{(l+1)}[k+1] = \theta_j^{(l)}[k] + \mu \tau_j^{(l)} + \eta \theta_j^{(l)}[k] - \theta_j^{(l)}[k-1] \quad (32)$$

当样本集中的所有样本都经历了 Step1,2,3 后,即完成一个训练周期,计算性能指标 $E = \sum_{q=1}^Q E_q$, 其中 $E_q = \frac{1}{2} \sum_{j=1}^m (d_{qj} - x_{qj})^2$ 。如果性能指标满足精度要求,即 $E < \varepsilon$, 那么训练结束, 否则, 继续下一个训练周期^[6]。 ε 为最小正数, 根据实际情况而定。

程序 1: BP 神经网络	运行环境: Matlab2011a
<pre> P= ; % 取点只能在 0 到 1 之间 t=[]; % 对应点的值 net = newff(minmax(p),[2,1],{'tansig' 'purelin' 'sigmoid'},'trainlm'); net.trainParam.epochs=1000; % 网络训练步骤设置为 1000 net.trainParam.goal=0.0000001; % 网络训练精度设置为 0.0000001 net=train(net,p,t); % 开始训练 n 个点 p1= ; % 训练的点, 接着 P, 与 P 矩阵连续或者部分重复 y2=sim(net,p1) % 预测 </pre>	

程序 2: BP 神经网络	运行环境: Matlab2011a
<pre> %-----BP 神经网络主要程序----- % 产生训练样本与测试样本 input_train=[2012 2013 2014]; output_train=[3517.72 4020.27 3997.2]; [inputn,inputps]=mapminmax(input_train); [outputn,outputps]=mapminmax(output_train); % 函数接口赋值 NodeNum = 49; % 隐层节点数 TypeNum = 1; % 输出维数 Epochs = 15000; % 训练次数 input_text=[2013 2014 2015]; % 设置网络参数 TF1 = 'radbas'; TF2 = 'purelin'; % TF1 = 'radbas';TF2 = 'purelin'; net = newff(minmax(inputn),[NodeNum TypeNum],{TF1 TF2},'trainlm'); % 指定训练参数 net.trainParam.epochs = Epochs; % 最大训练次数 net.trainParam.goal = 1e-50; % 最小均方误差 net.trainParam.min_grad = 1e-20; % 最小梯度 net.trainParam.show = 200; % 训练显示间隔 net.trainParam.time = inf; % 大训练时间 </pre>	

```
% 训练与测试
inputn_test=mapminmax('apply',input_text,inputps);
net = train(net,inputn,outputn);      % 训练
X = sim(net,inputn_test)              % 测试 - 输出为预测值
xiaobooutput=mapminmax('reverse',X,outputps)
```

程序 3: BP 神经网络

运行环境: Matlab2011a

```
p = [2601 2703 2858;
      2703 2858 2822;
      2858 2822 3192;
      2822 3192 3065;
      3192 3065 3110;
      3065 3110 2874;
      3110 2874 2696];%注意数据录入的形式
t = [2822 3192 3065 3110 2874 2696];%每次预测矩阵需人工更改（与 P 矩阵中标
红的数据相对应，预测下一个数）
pmax = max(p);%p 矩阵每列的最大值
pmax1 = max(pmax);%p 矩阵的最大值
pmin = min(p);%p 矩阵每列的最小值
pmin1 = min(pmin);%p 矩阵的最小值
for i = 1:6%t 矩阵有六个数
    pl(i,:) = (p(i,:) - pmin1)/(pmax1 - pmin1);%矩阵中的元素减去最小值除以
    %最大最小的差值（归一
    化处理）
end
t1 = (t - pmin1)/(pmax1 - pmin1);%（归一化处理）
t1 = t1';
net = newff([0 1;0 1;0 1],[7 1],{'tansig','logsig'},'traingd');%以下为固定程序
for i = 1:6%t 矩阵有六个数
    net.trainParam.epochs = 15000;
    net.trainParam.goal = 0.01;
    LP.lr = 0.1;
    net = train(net,pl(i,:)',t1(i));
end
y = sim(net,[3110 2874 2696]');%每次预测矩阵需人工更改（其中的数值与 t 矩阵
黄底纹的相对应）
y1 = y*(pmax1 - pmin1) + pmin1;
y1
```