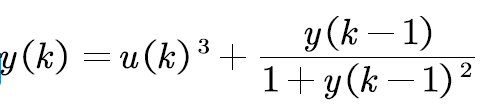
## 一、实验目的

（1）7.2.5 节：使用BP 网络逼近仿真实例，采样时间取1ms。输入信号为u(k)=0.5sin（6Πt）。神经网络为2-6-1结构，权值W1，W2的初始值取[-1,+1]之间的随机值，取η=0.50，α= 0.05。BP网络逼近程序见exp7\_1.m；

（2）取标准样本为三输入两输出样本，如表7-1所示。BP网络为3-6-2结构,权值ωij，ωjo的初始值取[-1,+1]之间的随机值,学习参数取η=0.50，α= 0.05。BP网络模式识别程序包括网络训练程序exp7\_2a.m和网络测试程序exp7\_2b.m。



## 二、程序代码

### 2.1、BP网络逼近仿真实例程序代码

exp7\_1.m，具体代码如下。

%BP识别

xite=0.50; %学习速率

alfa=0.05; %α

%输出层及隐层的连接权值wjo，rands（6，1）返回随机值的6x1矩阵

w2=rands(6,1);

w2\_1=w2;w2\_2=w2\_1; %权值W2取随机6x1矩阵

%输入层及隐层的连接权值wij

w1=rands(2,6); %返回随机值的2x6矩阵

w1\_1=w1;w1\_2=w1;

%初始化△wij矩阵

dw1=0\*w1;

%输入矩阵，x(1)为输入u(k)，x(2)为输出y(k)

x=[0,0]';

u\_1=0; %系统输入初始值

y\_1=0; %系统输出初始值

I=[0,0,0,0,0,0]'; %隐层神经元的输入

Iout=[0,0,0,0,0,0]'; %隐层神经元的输出

FI=[0,0,0,0,0,0]'; %∂xj'/∂xj

ts=0.001;

for k=1:1:1000

time(k)=k\*ts;

u(k)=0.50\*sin(3\*2\*pi\*k\*ts); %系统输入

y(k)=u\_1^3+y\_1/(1+y\_1^2); %系统输出

for j=1:1:6

I(j)=x'\*w1(:,j); %隐层神经元的输入xj

Iout(j)=1/(1+exp(-I(j))); %隐层神经元的输出xj'

end

yn(k)=w2'\*Iout; % 输出层神经元的输出yn(k)

e(k)=y(k)-yn(k); %误差计算

%相当于课本上wjo(k+1)的公式表达,即加入动量因子后学习过程权值

w2=w2\_1+(xite\*e(k))\*Iout+alfa\*(w2\_1-w2\_2);

for j=1:1:6

FI(j)=exp(-I(j))/(1+exp(-I(j)))^2;%∂xj'/∂xj

end

for i=1:1:2

for j=1:1:6

dw1(i,j)=e(k)\*xite\*FI(j)\*w2(j)\*x(i);%△wij

end

end

%相当于课本上wij(k+1)的公式表达,即加入动量因子后学习过程权值

w1=w1\_1+dw1+alfa\*(w1\_1-w1\_2);

%%%%%%%%%%%%%%Jacobian%%%%%%%%%%%%%%%%

yu=0;

for j=1:1:6

yu=yu+w2(j)\*w1(1,j)\*FI(j); %对象输出对输入的敏感度∂y(k)/∂u(k)（Jacobian信息）

end

dyu(k)=yu;

%为系统两个输入赋值

x(1)=u(k);

x(2)=y(k);

%调整当前与上次权值

w1\_2=w1\_1;w1\_1=w1;

w2\_2=w2\_1;w2\_1=w2;

%改变系统输入输出初始值

u\_1=u(k);

y\_1=y(k);

end

figure(1);

plot(time,y,'r',time,yn,'b');

xlabel('times');ylabel('y and yn');

figure(2);

plot(time,y-yn,'r');

xlabel('times');ylabel('error');

figure(3);

plot(time,dyu);

xlabel('times');ylabel('dyu');

### 2.2、BP网络模式识别仿真实例程序代码

1、exp7\_2a.m，具体代码如下。

%3-6-2结构BP网络

xite=0.50; %学习速率

alfa=0.05;%α

%输出层及隐层的连接权值wjo

w2=rands(6,2);

w2\_1=w2;w2\_2=w2\_1;

%输入层及隐层的连接权值wij

w1=rands(3,6);

w1\_1=w1;w1\_2=w1;

dw1=0\*w1; %初始化△wij矩阵

I=[0,0,0,0,0,0]'; %隐层神经元的输入

Iout=[0,0,0,0,0,0]'; %隐层神经元的输出

FI=[0,0,0,0,0,0]'; %∂xj'/∂xj

OUT=2; %输出个数

k=0; %训练次数

E=1.0; %网络训练的最终指标

NS=3; %样本给出的输入输出行数

while E>=1e-020

k=k+1; %k为训练次数

times(k)=k;

for s=1:1:NS %三输入两输出样本

xs=[1,0,0;

0,1,0;

0,0,1]; %理想输入

ys=[1,0;

0,0.5;

0,1]; %理想输出

x=xs(s,:);

for j=1:1:6

I(j)=x\*w1(:,j); %隐层神经元的输入xj

Iout(j)=1/(1+exp(-I(j))); %隐层神经元的输出xj'

end

yl=w2'\*Iout;

yl=yl'; %y1为输出矩阵

el=0;

y=ys(s,:);

for l=1:1:OUT

el=el+0.5\*(y(l)-yl(l))^2; %输出误差

end

es(s)=el; %es(s)为每一次输入的输出误差

E=0;

if s==NS

for s=1:1:NS

E=E+es(s);

end

end

ey=y-yl; %输出误差矩阵

%相当于课本上wjo(k+1)的公式表达，即加入动量因子后学习过程获得的权值

w2=w2\_1+xite\*Iout\*ey+alfa\*(w2\_1-w2\_2);

for j=1:1:6

S=1/(1+exp(-I(j)));

FI(j)=S\*(1-S);

end

for i=1:1:3

for j=1:1:6

dw1(i,j)=xite\*FI(j)\*x(i)\*(ey(1)\*w2(j,1)+ey(2)\*w2(j,2));

end

end

%相当于课本上wij(k+1)的公式表达，即加入动量因子后学习过程获得的权值

w1=w1\_1+dw1+alfa\*(w1\_1-w1\_2);

w1\_2=w1\_1;w1\_1=w1;

w2\_2=w2\_1;w2\_1=w2;

end

Ek(k)=E;

end

figure(1);

plot(times,Ek,'r');

xlabel('k');ylabel('E');

save wfile w1 w2; %最终权值为用于模式识别的知识

2、exp7\_2b.m，具体代码如下。

load wfile w1 w2;

%实际样本输入数据

x=[0.970,0.001,0.001;

0.000,0.980,0.000;

0.002,0.000,1.040;

0.500,0.500,0.500;

1.000,0.000,0.000;

0.000,1.000,0.000;

0.000,0.000,1.000];

for i=1:1:7

for j=1:1:6

I(i,j)=x(i,:)\*w1(:,j);

Iout(i,j)=1/(1+exp(-I(i,j)));

end

end

y=w2'\*Iout';

y=y'

## 三、运行实例及分析

### 3.1、BP网络逼近仿真实例运行结果

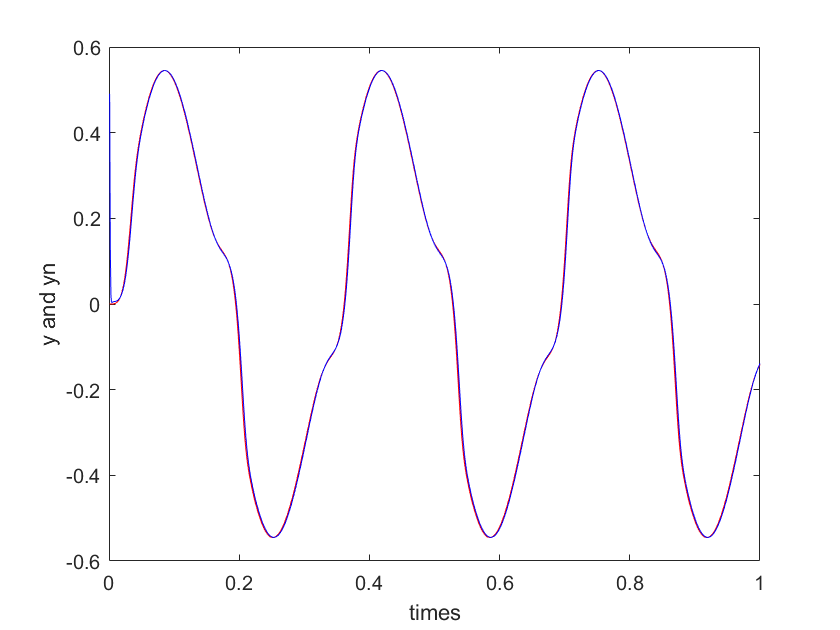


图1 BP网络逼近效果

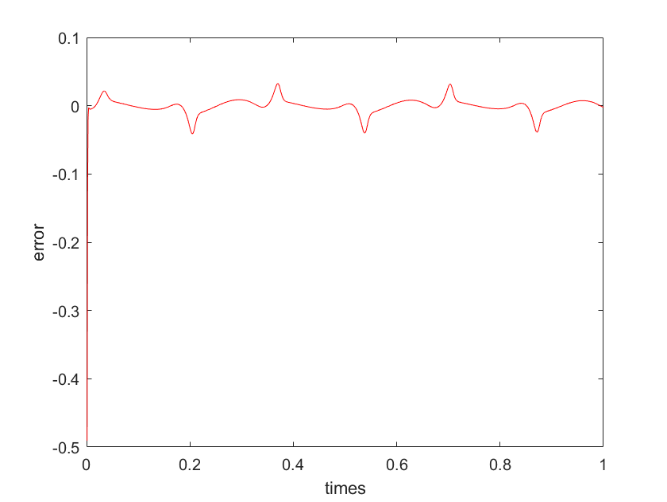


图2 BP网络逼近误差

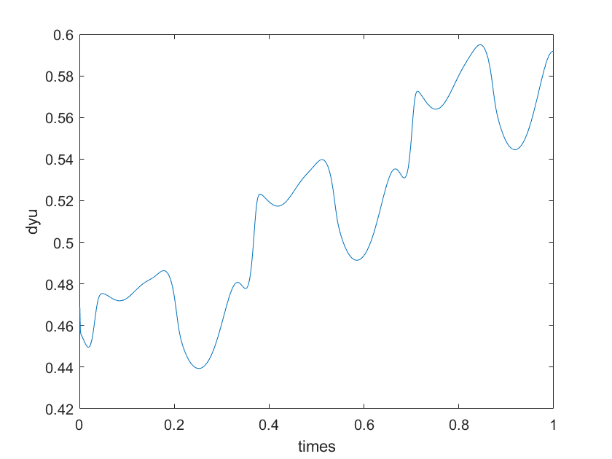


图3 Jacobian信息的辨识

### 3.2、BP网络模式识别仿真实例运行结果

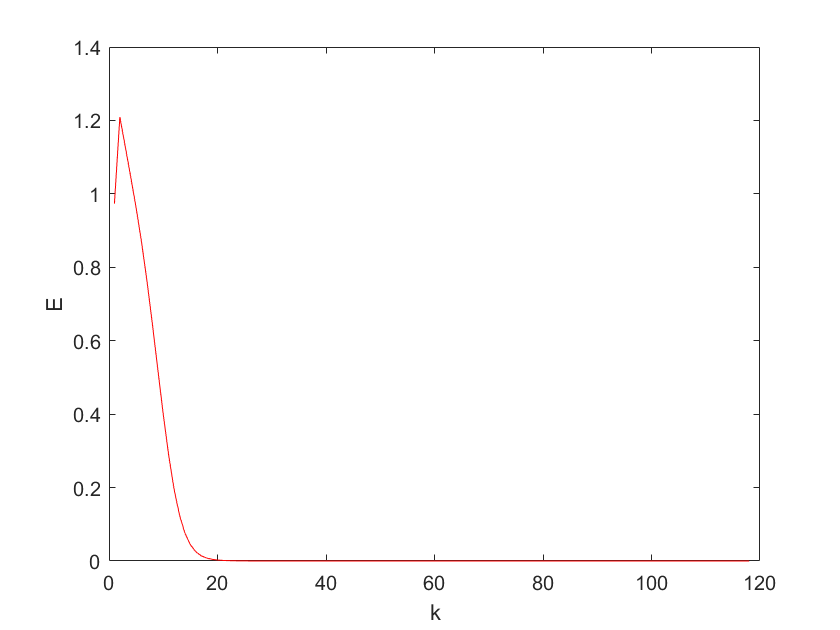


图4 样本训练收敛过程

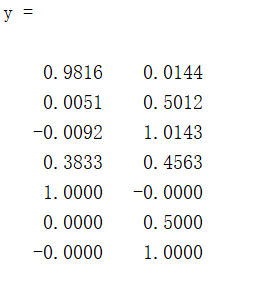


图5 测试样本输出结果

## 四、结果分析及总结

### 4.1、BP网络逼近仿真实例结果分析

由图1-3中可以看出，使用BP网络逼近，其输出结果与理想输出曲线几乎重合，输出结果误差在0周围呈规侓性变化。Jacobian信息整体呈现上升趋势，即对象输出对输入的敏感度随着神经网络的不断辨识而逐渐上升。

由此可以得出，随着BP神经网络的利用系统输出与网络输出的误差作为逼近器的调整信号不断逼近，可以使得系统的实际输出逐渐逼近理想输出。

### 4.2、BP网络模式识别仿真实例结果分析

由样本训练收敛过程可知，随着训练次数的增加，网络训练的最终误差性能指标E不断减小，也即系统输出与理想输出的误差值逐渐减小，最终在E< 1e-020时，得到最终权值，通过使用训练所得权值对实际样本的各输入值进行处理，最终得到各输出值如图5。

从图5最后三行结果可得，训练所得权值已经可以准确满足输入训练样本的输入值可以得到训练样本的理想输出值。故而此次训练所得权值效果应该可以判定为较为理想。