**BP算法解决水质预测问题**

**臧泽林**

# 代码块1

% bp算法

%该算法使用BP神经网络对水质问题进行预测

%输入为是个连续周期的水质数据

%输出为第十一个周期的水质数据

%训练使用的数据集是404个水质数据

% 作者：邻泽居士

% 日期：2016/5/1

% 版本：v1

%% 状态清除

clc

clear all

%% 导入数据

xlsfile = 'datazzl.txt'; %打开文件，输入数据

[data] = xlsread(xlsfile,'a1:e404');

%% 数据归一化

for i = 1:4

ml(i) = [max(data(:,i))-min(data(:,i))]/2;

data(:,i) = data(:,i)/ml(i);

mm(i) = mean(data(:,i));

traind\_s(:,i) = data(:,i) - mm(i);

end

% traind\_s(:,5) = traind\_s(:,5) +0.8;

%% 网络参数设置

eb = 0.0001 ; %误差容限

eta = 0.15; %学习率

mc =0.00000; %动量

mcb = 0;

maxiter = 404 ; %数据量

iteration = 0; %第一次迭代

ntrainnum =404;

inbox = 10;

%% 网络构造

net.nin = 40;

net.nhidden = 39;

net.nout = 4; %神经网络层数

dWEXold = 0;

dwexold = 0;

w = 2\*(rand(net.nin,net.nhidden)-1/2);

b = 2\*(rand(net.nin,1)-1/2);

net.w1 = [w,b]; %隐层的权值

uk = 1;

W= 2\*(rand(net.nhidden+2,net.nout)-1/2);

net.w2 =[W] ; %输出层的权值

%% 误差矩阵建立

errrec = zeros(1,4); %误差矩阵

outrec = zeros(4,maxiter); %误差记录

errsum(1,:) = zeros(1,4);

NET = [ ];

index = 1;

%% 误差下降

for bianshu =1:150

for i = 1 : maxiter-10

%% 误差的计算

%从数据中截取本次迭代所用数据

sampinex1 = traind\_s(i : i + inbox - 1,:);

for x = 1:10

sampinex2(1,4\*(x-1)+1:4\*(x)) = sampinex1(x,1:4);

end

%产生期望输出

expectout = traind\_s( i + inbox,:);

%迭代运算以及误差的计算

%隐层迭代

hid\_input = sampinex2\*net.w1;

hid\_out = tansig(hid\_input);

%输出层迭代

ou\_input1 = [hid\_out,1];

ou\_input2 =ou\_input1\* net.w2 ;

out\_out = tansig(ou\_input2);

%迭代结果

outrec(:,i) = out\_out';

%计算每项误差

err = expectout - out\_out;

%误差求和

sse = sumsqr(err);

%记录误差

errrec(i,:) = sse;

wuchajilu(i,:) = err;

%% 误差的反向传播

if i < 2000

for ii=1:length(hid\_out )+1

for j=1:4

ipu1(j)=err(j); % 局部梯度

% 输出层与隐含层之间的调整量

delta1(ii,j) = eta.\*ipu1(j).\*ou\_input1(ii);

end

end

for m=1:40

for ii=1:length(hid\_out)

% 局部梯度

ipu2(ii)=hid\_out(ii).\*(1-hid\_out(ii)).\*sum(ipu1.\*net.w2(m,:));

% 输入层和隐含层之间的调整量

delta2(m,ii)= eta.\*ou\_input1(m).\*ipu2(ii);

end

end

else

a = wuchajilu(i,:)-wuchajilu(i-1,:);

b = (net.w2(:,1)-net.w2old(:,1))';

for zzli = 1:4

for zzlj = 1:41

J(zzli,zzlj) = a(zzli)/b(zzlj);

end

end

if wuchajilu(i,1)-wuchajilu(i-1,1)<0

uk = uk/1.3

else

uk = 1.3\*uk

end

delta1 = ( 0.05\*J'\*J + 1\*eye(41) ) \ J' \* wuchajilu(i,1) ;

delta2 = zeros(40);

end

net.w2old = net.w2;

net.w1old = net.w1;

if i==1

net.w2 = net.w2+eta \* delta1;

net.w1= net.w1 + eta \* delta2;

else

net.w2 = net.w2+eta \* delta1 + mc \* dWEXold;

net.w1 = net.w1 + eta \* delta2 + mc \* dwexold;

end

dWEXold = delta1;

dwexold = delta2;

end

bianshu

sse = 0;

wuchajilu(bianshu,:) = err;

for count = 1:394

sse = sse+abs(wuchajilu(count,1))+abs(wuchajilu(count,2))+abs(wuchajilu(count,3))+abs(wuchajilu(count,4));

end

sse

if(bianshu == 100)

i = i;

zzl = figure(1);

for zzl = 1:4

subplot(2,2,zzl);

plot([11:10+404],outrec(zzl,:),'r');

hold on

plot(traind\_s(:,zzl),'b');

hold off

end

end

%% 测试

%%figure(1)

% axis on

% hold on

% grid

% [o,q] = size(errrec);

%%plot(1:q,errrec,'b-','linewidth',1.5);

%%axis([0 12952 0 30]);

%% 滤波

% jisuanwucha;

% if (sume<20)

% break;

% end

end

i= i;

# 代码块2

function [ J ] = gotJ( e,w )

%求取雅各比矩阵

%w，e输入，输出雅各比

if(length(e) == 2)

c = 1;

else

c = length(e);

end

for i = 1:c

if i == 1

de = e(2)-e(1);

elseif i == length(e)

de = e(length(e))-e(length(e)-1);

else

de = (e(i)-e(i-1)+e(i+1)-e(i))/2;

end

for j = 1:length(w)

if j == 1

dw = w(2)-w(1);

elseif j == length(w)

dw = w(length(w))-w(length(w)-1);

else

dw = (w(j)-w(j-1)+w(j+1)-w(j));

end

J(i,j) = de/dw;

end

end

end

# 代码块3

clc

clear all

load out2.mat

load train.mat

outrec = outrec';

ml = [2.35500000000000,0.110000000000000,6.50000000000000,323.500000000000];

mm = [10.9945239746905,69.0063006300631,-6.66869763899466,-0.319387271030040];

for j = 1:4

for i = 1:404

outrec(i,j) = outrec(i,j)\*ml(j)+mm(j);

end

end

for j = 1:4

for i = 1:404

traind\_s(i,j) = traind\_s(i,j)\*ml(j)+mm(j);

end

end

xunlian = 250;

yuce = 152;

a = 1:xunlian;

b = xunlian+1:xunlian+yuce;

figure(1)

%% 拟合曲线

title('fitted');

zzl = 1;

subplot(2,2,zzl);

plot(a,traind\_s(1:xunlian,zzl),'b');

hold on

plot(a+10,outrec(1:xunlian,zzl),'r');

hold off

xlabel('time');

ylabel('temperature');

legend('measured value','fitted value');

zzl =2;

subplot(2,2,zzl);

plot(a,traind\_s(1:xunlian,zzl),'b');

hold on

plot(a+10,outrec(1:xunlian,zzl),'r');

hold off

xlabel('time');

ylabel('pH');

legend('measured value','fitted value');

zzl = 3;

subplot(2,2,zzl);

plot(a,traind\_s(1:xunlian,zzl),'b');

hold on

plot(a+10,outrec(1:xunlian,zzl),'r');

hold off

xlabel('time');

ylabel('Do');

legend('measured value','fitted value');

zzl = 4;

subplot(2,2,zzl);

plot(a,traind\_s(1:xunlian,zzl),'b');

hold on

plot(a+10,outrec(1:xunlian,zzl),'r');

hold off

xlabel('time');

ylabel('ORP');

legend('measured value','fitted value');

%% 预测曲线

figure(2)

zzl = 1;

subplot(2,2,zzl);

plot(b,traind\_s(xunlian+1:xunlian+yuce,zzl),'b');

hold on

plot(b+10,outrec(xunlian+1:xunlian+yuce,zzl),'r');

hold off

xlabel('time');

ylabel('temperature');

legend('measured value','predicted value');

zzl = 2;

subplot(2,2,zzl);

plot(b,traind\_s(xunlian+1:xunlian+yuce,zzl),'b');

hold on

plot(b+10,outrec(xunlian+1:xunlian+yuce,zzl),'r');

hold off

xlabel('time');

ylabel('pH');

legend('measured value','predicted value');

zzl = 3;

subplot(2,2,zzl);

plot(b,traind\_s(xunlian+1:xunlian+yuce,zzl),'b');

hold on

plot(b+10,outrec(xunlian+1:xunlian+yuce,zzl),'r');

hold off

xlabel('time');

ylabel('DO');

legend('measured value','predicted value');

zzl = 4;

subplot(2,2,zzl);

plot(b,traind\_s(xunlian+1:xunlian+yuce,zzl),'b');

hold on

plot(b+10,outrec(xunlian+1:xunlian+yuce,zzl),'r');

hold off

xlabel('time');

ylabel('ORP');

legend('measured value','predicted value');