遗传算法优化BP神经网络权值和阈值的通用MATLAB源代码

遗传算法优化神经网络有两种情况，一种是把训练好的神经网络作为黑箱函数，用遗传算法搜索该黑箱函数的最优解；另外一种，则是把遗传算法用于神经网络的训练，充分利用遗传算法全局搜索的特性，得到一个初始的权值矩阵和初始的阈值向量，再用其它训练算法（如BP算法），得到最终的神经网络结构。GreenSim团队大量实践证明，这种GA和BP网络相结合的方法，能显著地提高BP神经网络的性能，基本上和支持向量机的性能相当，有时甚至优于支持向量机。由于BP网络的权值优化是一个无约束优化问题，而且权值要采用实数编码，所以直接利用Matlab遗传算法工具箱。以下贴出的代码是为一个19输入变量，1个输出变量情况下的非线性回归而设计的，如果要应用于其它情况，只需改动编解码函数即可。

程序一：GA训练BP权值的主函数

function net=GABPNET(XX,YY)

%--------------------------------------------------------------------------

% GABPNET.m

% 使用遗传算法对BP网络权值阈值进行优化，再用BP算法训练网络

% GreenSim团队——专业级算法设计&代写程序

% 欢迎访问GreenSim团队主页→http://blog.sina.com.cn/greensim

%--------------------------------------------------------------------------

%数据归一化预处理

nntwarn off

XX=premnmx(XX);

YY=premnmx(YY);

%创建网络

net=newff(minmax(XX),[19,25,1],{'tansig','tansig','purelin'},'trainlm');

%下面使用遗传算法对网络进行优化

P=XX;

T=YY;

R=size(P,1);

S2=size(T,1);

S1=25;%隐含层节点数

aa=ones(S,1)\*[-1,1];

popu=50;%种群规模

initPpp=initializega(popu,aa,'gabpEval');%初始化种群

gen=100;%遗传代数

%下面调用gaot工具箱，其中目标函数定义为gabpEval

[x,endPop,bPop,trace]=ga(aa,'gabpEval',[],initPpp,[1e-6 1 1],'maxGenTerm',gen,...

'normGeomSelect',[0.09],['arithXover'],[2],'nonUnifMutation',[2 gen 3]);

%绘收敛曲线图

figure(1)

plot(trace(:,1),1./trace(:,3),'r-');

hold on

plot(trace(:,1),1./trace(:,2),'b-');

xlabel('Generation');

ylabel('Sum-Squared Error');

figure(2)

plot(trace(:,1),trace(:,3),'r-');

hold on

plot(trace(:,1),trace(:,2),'b-');

xlabel('Generation');

ylabel('Fittness');

%下面将初步得到的权值矩阵赋给尚未开始训练的BP网络

[W1,B1,W2,B2,P,T,A1,A2,SE,val]=gadecod(x);

net.LW{2,1}=W1;

net.b{2,1}=B1;

net.b{3,1}=B2;

XX=P;

YY=T;

%设置训练参数

net.trainParam.show=1;

net.trainParam.lr=1;

net.trainParam.epochs=50;

net.trainParam.goal=0.001;

%训练网络

net=train(net,XX,YY);

程序二：适应值函数

function [sol, val] = gabpEval(sol,options)

% val - the fittness of this individual

% sol - the individual, returned to allow for Lamarckian evolution

% options - [current\_generation]

load data2

nntwarn off

XX=premnmx(XX);

YY=premnmx(YY);

P=XX;

T=YY;

R=size(P,1);

S2=size(T,1);

S1=25;%隐含层节点数

S=R\*S1+S1\*S2+S1+S2;%遗传算法编码长度

for i=1:S,

x(i)=sol(i);

end;

[W1, B1, W2, B2, P, T, A1, A2, SE, val]=gadecod(x);

程序三：编解码函数

function [W1, B1, W2, B2, P, T, A1, A2, SE, val]=gadecod(x)

load data2

nntwarn off

YY=premnmx(YY);

P=XX;

T=YY;

R=size(P,1);

S2=size(T,1);

S1=25;%隐含层节点数

S=R\*S1+S1\*S2+S1+S2;%遗传算法编码长度

% 前R\*S1个编码为W1

for i=1:S1,

for k=1:R,

W1(i,k)=x(R\*(i-1)+k);

end

end

% 接着的S1\*S2个编码（即第R\*S1个后的编码）为W2

for i=1:S2,

for k=1:S1,

W2(i,k)=x(S1\*(i-1)+k+R\*S1);

end

end

% 接着的S1个编码（即第R\*S1+S1\*S2个后的编码）为B1

for i=1:S1,

B1(i,1)=x((R\*S1+S1\*S2)+i);

end

% 接着的S2个编码（即第R\*S1+S1\*S2+S1个后的编码）为B2

for i=1:S2,

B2(i,1)=x((R\*S1+S1\*S2+S1)+i);

end

% 计算S1与S2层的输出

A1=tansig(W1\*P,B1);

A2=purelin(W2\*A1,B2);

% 计算误差平方和

SE=sumsqr(T-A2);

val=1/SE; % 遗传算法的适应值

注意：上面的函数需要调用gaot工具箱，请从网上搜索下载

源代码运行结果展示



