

遗传算法的优化计算——输入自变量降维

该案例作者申明：

- 1：本人长期驻扎在此[板块](#)里，对该案例提问，做到有问必答。本套书籍官方网站为：[video.ourmatlab.com](#)
- 2：点此[从当当预定本书](#)：《[Matlab神经网络30个案例分析](#)》。
- 3：此案例有配套的教学视频，视频下载方式[video.ourmatlab.com/vbuy.html](#)。
- 4：此案例为原创案例，转载请注明出处（《[Matlab神经网络30个案例分析](#)》）。
- 5：若此案例碰巧与您的研究有关联，我们欢迎您提意见，要求等，我们考虑后可以加在案例里。

Contents

- [清空环境变量](#)
- [声明全局变量](#)
- [导入数据](#)
- [数据归一化](#)
- [创建单BP网络](#)
- [训练单BP网络](#)
- [仿真测试单BP网络](#)
- [结果显示（单BP网络）](#)
- [遗传算法优化](#)
- [新训练集/测试集数据提取](#)
- [创建优化BP网络](#)
- [训练优化BP网络](#)
- [仿真测试优化BP网络](#)
- [结果显示（优化BP网络）](#)

清空环境变量

```
clear all
clc
warning off
```

声明全局变量

```
global P_train T_train P_test T_test  mint maxt S s1
S = 30;
s1 = 50;
```

导入数据

```
load data.mat
a = randperm(569);
Train = data(a(1:500),:);
Test = data(a(501:end),:);
% 训练数据
P_train = Train(:,3:end)';
T_train = Train(:,2)';
% 测试数据
P_test = Test(:,3:end)';
T_test = Test(:,2)';
% 显示实验条件
total_B = length(find(data(:,2) == 1));
total_M = length(find(data(:,2) == 2));
count_B = length(find(T_train == 1));
count_M = length(find(T_train == 2));
number_B = length(find(T_test == 1));
number_M = length(find(T_test == 2));
disp('实验条件为: ');
```

```
disp(['病例总数: ' num2str(569)...
      ' 良性: ' num2str(total_B)...
      ' 恶性: ' num2str(total_M)]);
disp(['训练集病例总数: ' num2str(500)...
      ' 良性: ' num2str(count_B)...
      ' 恶性: ' num2str(count_M)]);
disp(['测试集病例总数: ' num2str(69)...
      ' 良性: ' num2str(number_B)...
      ' 恶性: ' num2str(number_M)]);
```

实验条件为:
病例总数: 569 良性: 357 恶性: 212
训练集病例总数: 500 良性: 311 恶性: 189
测试集病例总数: 69 良性: 46 恶性: 23

数据归一化

```
[P_train,minp,maxp,T_train,mint,maxt] = premnmx(P_train,T_train);
P_test = trammnx(P_test,minp,maxp);
```

创建单BP网络

```
t = cputime;
net_bp = newff(minmax(P_train),[s1,1],{'tansig','purelin'},'trainlm');
% 设置训练参数
net_bp.trainParam.epochs = 1000;
net_bp.trainParam.show = 10;
net_bp.trainParam.goal = 0.1;
net_bp.trainParam.lr = 0.1;
net_bp.trainParam.showwindow = 0;
```

训练单BP网络

```
net_bp = train(net_bp,P_train,T_train);
```

仿真测试单BP网络

```
tn_bp_sim = sim(net_bp,P_test);
% 反归一化
T_bp_sim = postmnmx(tn_bp_sim,mint,maxt);
e = cputime - t;
T_bp_sim(T_bp_sim > 1.5) = 2;
T_bp_sim(T_bp_sim < 1.5) = 1;
result_bp = [T_bp_sim' T_test'];
```

结果显示 (单BP网络)

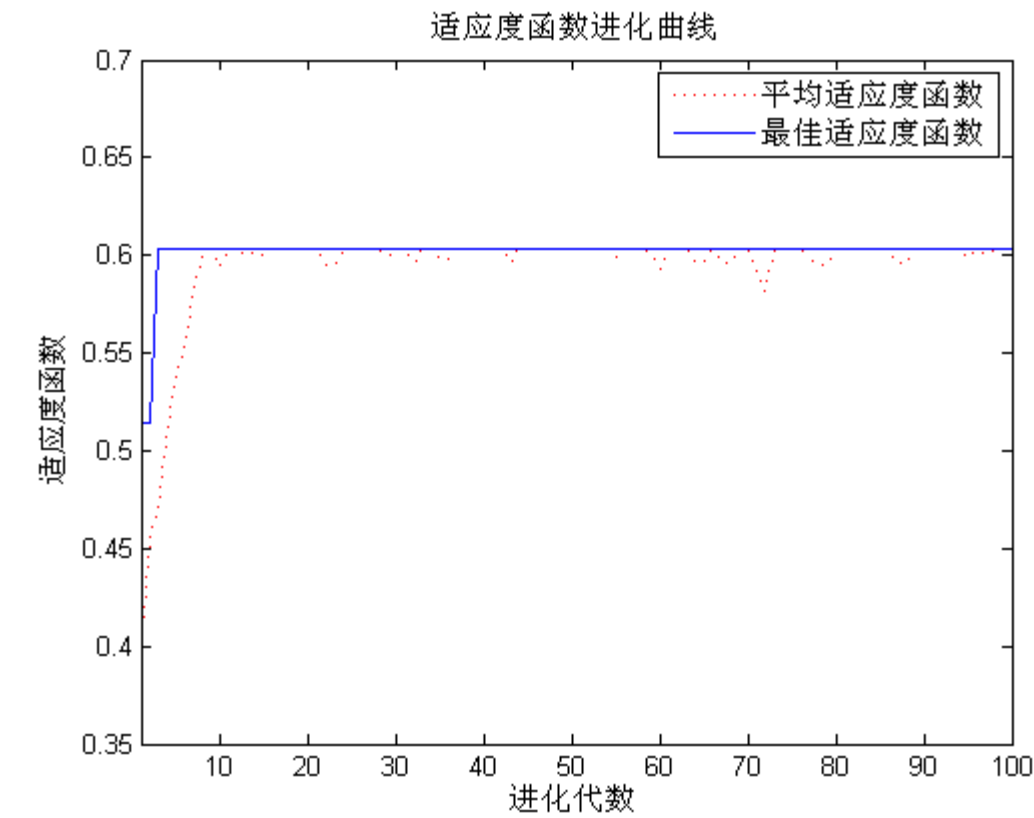
```
number_B_sim = length(find(T_bp_sim == 1 & T_test == 1));
number_M_sim = length(find(T_bp_sim == 2 & T_test == 2));
disp('(1)BP网络的测试结果为: ');
disp(['良性乳腺肿瘤确诊: ' num2str(number_B_sim)...
      ' 误诊: ' num2str(number_B - number_B_sim)...
      ' 确诊率p1 = ' num2str(number_B_sim/number_B*100) '%']);
disp(['恶性乳腺肿瘤确诊: ' num2str(number_M_sim)...
      ' 误诊: ' num2str(number_M - number_M_sim)...
      ' 确诊率p2 = ' num2str(number_M_sim/number_M*100) '%']);
disp(['建模时间为: ' num2str(e) 's'] );
```

(1)BP网络的测试结果为:
良性乳腺肿瘤确诊: 46 误诊: 0 确诊率p1 = 100%
恶性乳腺肿瘤确诊: 19 误诊: 4 确诊率p2 = 82.6087%
建模时间为: 11.9688s

遗传算法优化

```
popu = 20;
bounds = ones(S,1)*[0,1];
% 产生初始种群
initPop = randint(popu,S,[0 1]);
% 计算初始种群适应度
initFit = zeros(popu,1);
for i = 1:size(initPop,1)
    initFit(i) = de_code(initPop(i,:));
end
initPop = [initPop initFit];
gen = 100;
% 优化计算
[X,EndPop,BPop,Trace] = ga(bounds,'fitness',[],initPop,[1e-6 1 0],'maxGenTerm',...
    gen,'normGeomSelect',0.09,'simpleXover',2,'boundaryMutation',[2 gen 3]);
[m,n] = find(X == 1);
disp(['优化筛选后的输入自变量编号为:' num2str(n)]);
% 绘制适应度函数进化曲线
figure
plot(Trace(:,1),Trace(:,3),'r:')
hold on
plot(Trace(:,1),Trace(:,2),'b')
xlabel('进化代数')
ylabel('适应度函数')
title('适应度函数进化曲线')
legend('平均适应度函数','最佳适应度函数')
xlim([1 gen])
```

优化筛选后的输入自变量编号为:3 5 6 7 8 9 10 13 16 17 20 21 26 28 29 30



新训练集/测试集数据提取

```
p_train = zeros(size(n,2),size(T_train,2));
p_test = zeros(size(n,2),size(T_test,2));
for i = 1:length(n)
    p_train(i,:) = P_train(n(i),:);
    p_test(i,:) = P_test(n(i),:);
end
t_train = T_train;
```

创建优化BP网络

```
t = cputime;
net_ga = newff(minmax(p_train),[s1,1],{'tansig','purelin'},'trainlm');
% 训练参数设置
net_ga.trainParam.epochs = 1000;
net_ga.trainParam.show = 10;
net_ga.trainParam.goal = 0.1;
net_ga.trainParam.lr = 0.1;
net_ga.trainParam.showwindow = 0;
```

训练优化BP网络

```
net_ga = train(net_ga,p_train,t_train);
```

仿真测试优化BP网络

```
tn_ga_sim = sim(net_ga,p_test);
% 反归一化
T_ga_sim = postmnmx(tn_ga_sim,mint,maxt);
e = cputime - t;
T_ga_sim(T_ga_sim > 1.5) = 2;
T_ga_sim(T_ga_sim < 1.5) = 1;
result_ga = [T_ga_sim' T_test'];
```

结果显示（优化BP网络）

```
number_b_sim = length(find(T_ga_sim == 1 & T_test == 1));
number_m_sim = length(find(T_ga_sim == 2 & T_test == 2));
disp('(2)优化BP网络的测试结果为: ');
disp(['良性乳腺癌确诊: ' num2str(number_b_sim)...
      ' 误诊: ' num2str(number_B - number_b_sim)...
      ' 确诊率p1=' num2str(number_b_sim/number_B*100) '%']);
disp(['恶性乳腺癌确诊: ' num2str(number_m_sim)...
      ' 误诊: ' num2str(number_M - number_m_sim)...
      ' 确诊率p2=' num2str(number_m_sim/number_M*100) '%']);
disp(['建模时间为: ' num2str(e) 's']);

web browser http://www.matlabsky.com/thread-11198-1-1.html
```

(2)优化BP网络的测试结果为:
良性乳腺癌确诊: 46 误诊: 0 确诊率p1=100%
恶性乳腺癌确诊: 22 误诊: 1 确诊率p2=95.6522%
建模时间为: 3.5781s

[Matlab神经网络30个案例分析](#)

- 相关论坛:
- 《Matlab神经网络30个案例分析》官方网站: video.ourmatlab.com
- Matlab技术论坛: www.matlabsky.com
- Matlab函数百科: www.mfun.la
- Matlab中文论坛: www.ilovematlab.com

Published with MATLAB® 7.8