



`p=[p1;p2;p3];` % 一个 $3 \times N$ 的矩阵, $p1$ 、 $p2$ 、 $p3$ 、分别为一个行向量, 代表三个特征数据序列, 称之为训练样本

`t=[];` % 一个 $1 \times N$ 的矩阵, 称之为目标样本

`pr=[min(p1),max(p1);`

`min(p2),max(p2);`

`min(p2),max(p2);]` % 限制每一个特征序列的范围, 此处代码在徐小军论文中有错, 他把 `max` 放在前面了

`net=newff(pr, [10, 1], { 'logsig' 'logsig' }, 'traincgf');` % 创建 BP 网络

`net.trainParam.epochs=4000;` % 设置训练步数为 4000 步

`net.trainParam.goal=0.001;` % 设置误差性能目标值为 0.001

`net.trainParam.show=10;` % 设置显示的间隔次数

`net.trainParam.searchFcn='srchgol';` % 设置线性搜索方法为黄金分割搜索方法

`net=train(net, p, t);` % 训练 BP 网络, 训练完的数据存放在 `net` 结构体中, 主要包括输入层和隐层之间的权值矩阵, 隐层和输出层之间的权值矩阵, 隐层的阈值矩阵, 输出层的阈值矩阵

`a=sim(net, p);` % 网络仿真

其中 \mathbf{pr} 为输入向量（此处为颜色特征向量）中每行的最大值和最小值组成的矩阵； \mathbf{t} 为目标输出向量； \mathbf{p} 为输入向量；`traincgf` 为 Fletcher-Reeves 修正算法。

上述程序对应 `test.m`, 可以在 Matlab6p5 环境下运行

经训练过的神经网络的权值矩阵和阈值矩阵如下：

① 输入层和隐层之间的权值矩阵

$$W^1 = \begin{bmatrix} -0.3643 & -0.1065 & -0.3312 \\ -0.1473 & -0.5396 & -0.2883 \\ 0.3516 & 0.1070 & 0.3210 \\ -0.1336 & 0.0414 & 0.0132 \\ 0.1236 & 0.3612 & 0.1949 \\ 0.0302 & 0.0551 & 0.0605 \\ 0.5641 & -0.9457 & 0.5882 \\ 0.5530 & -0.9395 & 0.5764 \\ 0.0244 & 0.0502 & 0.0554 \\ 0.4968 & -1.2490 & -0.1940 \end{bmatrix}$$

② 隐层和输出层之间的权值矩阵

$$W^2 = \begin{bmatrix} -113.7783 & 3.7027 & -113.4687 & -2.8966 & 4.1178 \\ 190.5734 & -716.7848 & 716.7389 & -187.8721 & -2.6480 \end{bmatrix}$$

③ 隐层的阈值矩阵

$$b^1 = \begin{bmatrix} 4.6663 & 5.5916 & -4.5353 & 0.4894 & -3.8923 \\ -0.8279 & -1.2904 & -1.1926 & -0.7363 & 5.3518 \end{bmatrix}^T$$

④ 输出层的阈值矩阵

$$b^2 = 111.2488$$

导入徐晓军的神经网络数据文件 `net12.mat`

显示网络系数矩阵的方法：

输入 “`net.b{1,1}`” 指令后回车即可以得到隐层的阈值矩阵

`ans =`

```
7.1577
2.1508
2.5953
0.4067
1.7915
```

输入 “`net.b{2,1}`” 指令后回车即可以得到输出层的阈值矩阵

```
ans =
```

```
4.9868
```

输入 “net.IW{1,1}” 指令后回车即可以得到输入层与隐含层之间的权值矩阵

```
ans =
```

```
-3.2793   -2.7845   -0.7264   -2.3839   -3.7191    0.5869    2.2155    2.3349  
 9.8483   -4.6122   -8.0675  -15.3815   -3.2041    3.1093   -15.9278   12.4728  
 5.5347   -2.7028    6.8865   -3.2420   -2.5283   -1.3592   -5.3062   -3.0484  
-3.9233    2.3402   -1.2426    2.0186    7.9014   -3.8327   -5.1550    0.7612  
-5.9098   -1.7415   -3.2503  -13.5507   15.0947  -11.7699    0.2845    9.0854
```

输入 “net.LW{2,1}” 指令后回车即可以得到隐含层与输出层之间的权值矩阵

```
ans =
```

```
-1.4327   -3.3081   -4.9301   -7.7902    5.5754
```

下面为我在 matlab 中的实作

```
>> net.IW
```

```
ans =
```

```
[5x8 double]
```

```
[]
```

注：上述结果表明结构体中{1, 1}位置对应一个[5x8 double]即权值矩阵，

结构体中{2, 1}位置对应一个空矩阵，

```
>> net.IW{1,1}
```

```
ans =
```

```
-3.2793   -2.7845   -0.7264   -2.3839   -3.7191    0.5869    2.2155    2.3349  
 9.8483   -4.6122   -8.0675  -15.3815   -3.2041    3.1093   -15.9278   12.4728  
 5.5347   -2.7028    6.8865   -3.2420   -2.5283   -1.3592   -5.3062   -3.0484  
-3.9233    2.3402   -1.2426    2.0186    7.9014   -3.8327   -5.1550    0.7612  
-5.9098   -1.7415   -3.2503  -13.5507   15.0947  -11.7699    0.2845    9.0854
```

```
>> net.IW{2,1}
```

```
ans =
```

```
[]
```

```
>> net.LW
```

ans =

```
      []      []  
[1x5 double]  []
```

>> net.LW{2,1}

ans =

```
-1.4327  -3.3081  -4.9301  -7.7902   5.5754
```