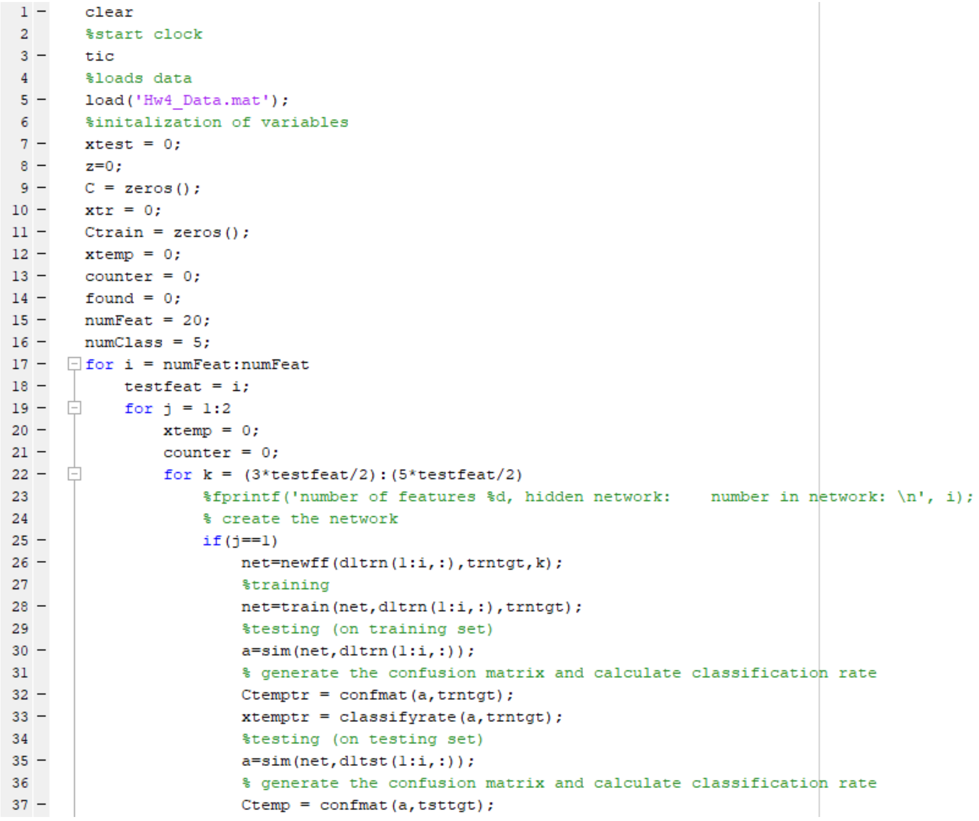
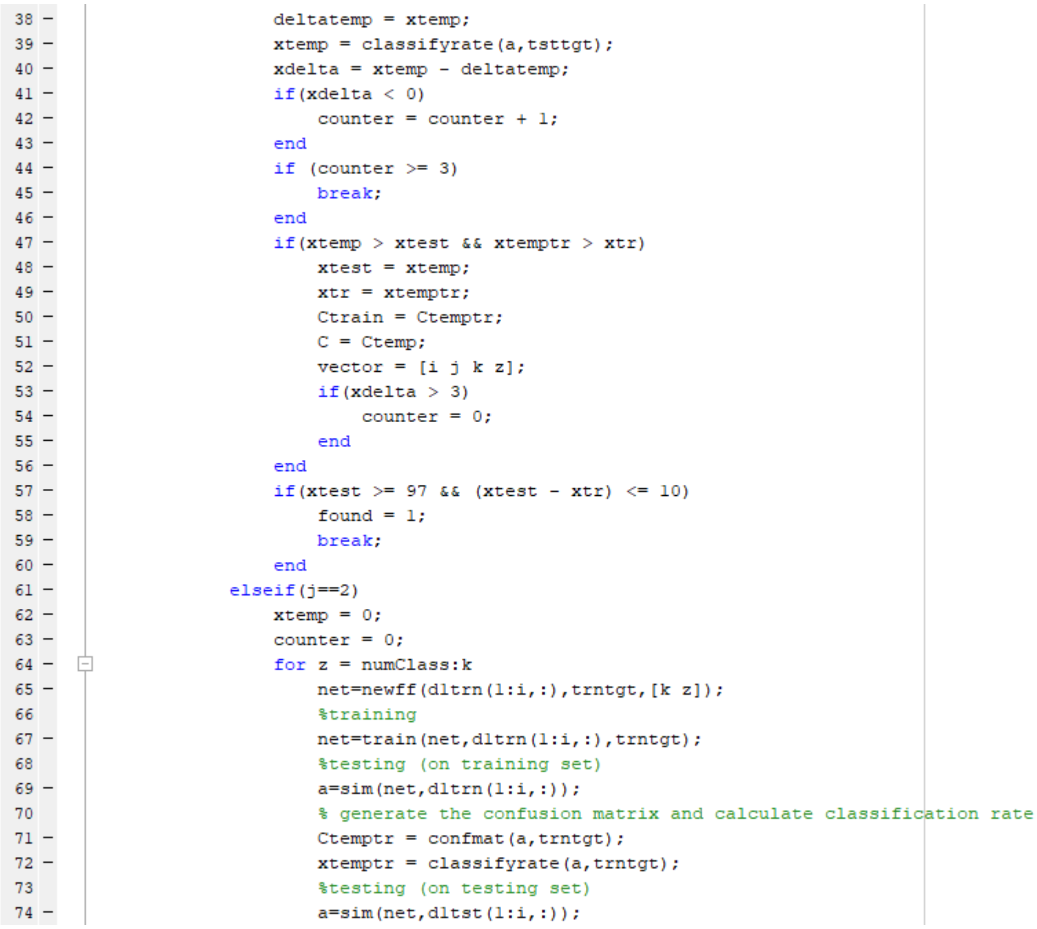
Shane Bolding 6/18/2019

Homework 4

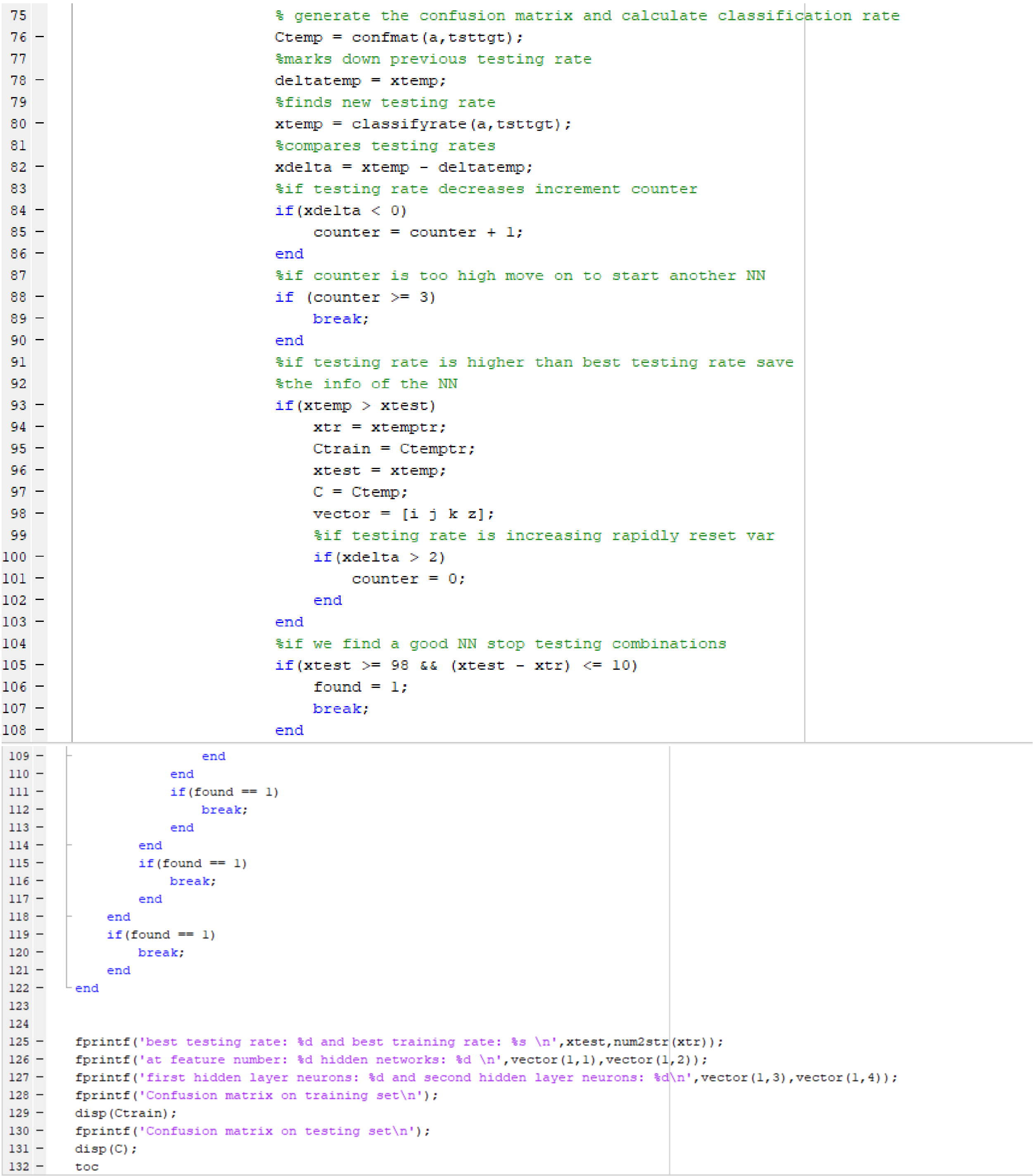
Code



```
clear
%start clock
tic
%loads data
load('Hw4_Data.mat');
%initalization of variables
xtest = 0;
z=0;
C = zeros();
xtr = 0;
Ctrain = zeros();
xtemp = 0;
counter = 0;
found = 0;
numFeat = 20;
numClass = 5;
for i = numFeat:numFeat
    testfeat = i;
    for j = 1:2
        xtemp = 0;
        counter = 0;
        for k = (3*testfeat/2):(5*testfeat/2)
            %fprintf('number of features %d, hidden network:    number in network: \n', i);
            % create the network
            if(j==1)
                net=newff(dltrn(1:i,:),trntgt,k);
                %training
                net=train(net,dltrn(1:i,:),trntgt);
                %testing (on training set)
                a=sim(net,dltrn(1:i,:));
                % generate the confusion matrix and calculate classification rate
                Ctemptr = confmat(a,trntgt);
                xtemptr = classifyrate(a,trntgt);
                %testing (on testing set)
                a=sim(net,dltst(1:i,:));
                % generate the confusion matrix and calculate classification rate
                Ctemp = confmat(a,tsttgt);
```



```
            deltatemp = xtemp;
            xtemp = classifyrate(a,tsttgt);
            xdelta = xtemp - deltatemp;
            if(xdelta < 0)
                counter = counter + 1;
            end
            if (counter >= 3)
                break;
            end
            if(xtemp > xtest && xtemptr > xtr)
                xtest = xtemp;
                xtr = xtemptr;
                Ctrain = Ctemptr;
                C = Ctemp;
                vector = [i j k z];
                if(xdelta > 3)
                    counter = 0;
                end
            end
            if(xtest >= 97 && (xtest - xtr) <= 10)
                found = 1;
                break;
            end
        elseif(j==2)
            xtemp = 0;
            counter = 0;
            for z = numClass:k
                net=newff(dltrn(1:i,:),trntgt,[k z]);
                %training
                net=train(net,dltrn(1:i,:),trntgt);
                %testing (on training set)
                a=sim(net,dltrn(1:i,:));
                % generate the confusion matrix and calculate classification rate
                Ctemptr = confmat(a,trntgt);
                xtemptr = classifyrate(a,trntgt);
                %testing (on testing set)
                a=sim(net,dltst(1:i,:));
```



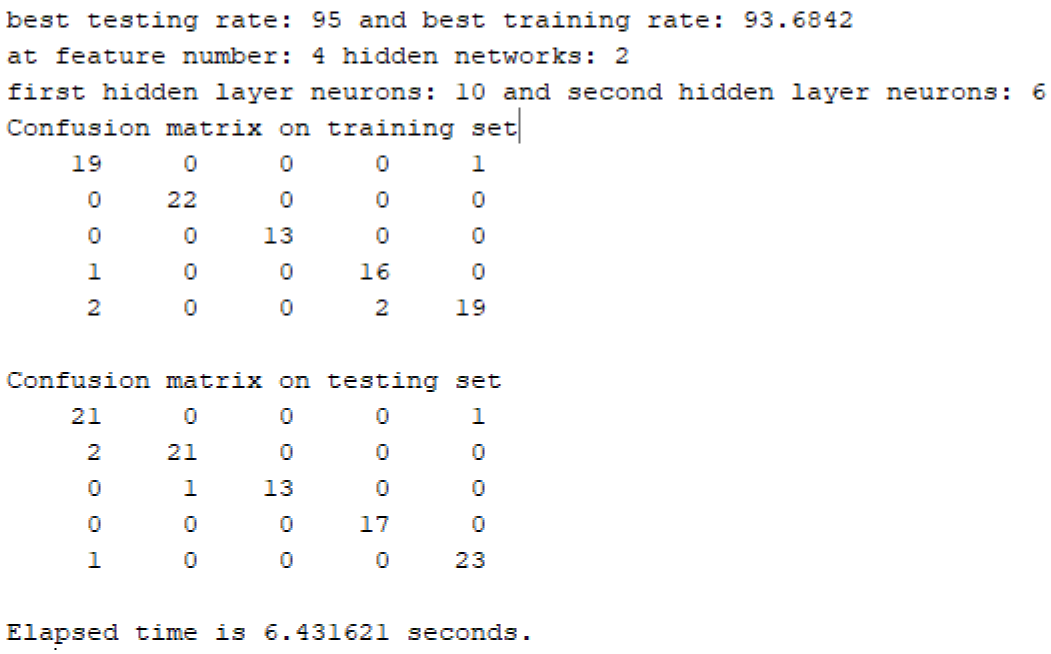
```
                    % generate the confusion matrix and calculate classification rate
                    Ctemp = confmat(a,tsttgt);
                    %marks down previous testing rate
                    deltatemp = xtemp;
                    %finds new testing rate
                    xtemp = classifyrate(a,tsttgt);
                    %compares testing rates
                    xdelta = xtemp - deltatemp;
                    %if testing rate decreases increment counter
                    if(xdelta < 0)
                        counter = counter + 1;
                    end
                    %if counter is too high move on to start another NN
                    if (counter >= 3)
                        break;
                    end
                    %if testing rate is higher than best testing rate save
                    %the info of the NN
                    if(xtemp > xtest)
                        xtr = xtemptr;
                        Ctrain = Ctemptr;
                        xtest = xtemp;
                        C = Ctemp;
                        vector = [i j k z];
                        %if testing rate is increasing rapidly reset var
                        if(xdelta > 2)
                            counter = 0;
                        end
                    end
                    %if we find a good NN stop testing combinations
                    if(xtest >= 98 && (xtest - xtr) <= 10)
                        found = 1;
                        break;
                    end
                end
            end
            if(found == 1)
                break;
            end
        end
        if(found == 1)
            break;
        end
    end
    if(found == 1)
        break;
    end
end


fprintf('best testing rate: %d and best training rate: %s \n',xtest,num2str(xtr));
fprintf('at feature number: %d hidden networks: %d \n',vector(1,1),vector(1,2));
fprintf('first hidden layer neurons: %d and second hidden layer neurons: %d\n',vector(1,3),vector(1,4));
fprintf('Confusion matrix on training set\n');
disp(Ctrain);
fprintf('Confusion matrix on testing set\n');
disp(C);
toc
```

Figure 1 Main code

The code above will run and test multiple neural networks with multiple levels until it finds a superb NN. If it runs through all the neural networks, then it will show the best combination like shown below. I used this in order to find general combinations that might be good for neural networks

Example of output



```
best testing rate: 95 and best training rate: 93.6842
at feature number: 4 hidden networks: 2
first hidden layer neurons: 10 and second hidden layer neurons: 6
Confusion matrix on training set
    19     0     0     0     1
     0    22     0     0     0
     0     0    13     0     0
     1     0     0    16     0
     2     0     0     2    19

Confusion matrix on testing set
    21     0     0     0     1
     2    21     0     0     0
     0     1    13     0     0
     0     0     0    17     0
     1     0     0     0    23

Elapsed time is 6.431621 seconds.
```

Number of Features	Network Structure	Number of Training Epochs	Correct Classification Rate on Training Set	Correct Classification Rate on Testing Set
4	[10 6]	23	93.68%	95%

Confusion matrix on training set

```
  19   0   0   0   1
   0  22   0   0   0
   0   0  13   0   0
   1   0   0  16   0
   2   0   0   2  19
```

Confusion matrix on Testing set

```
  21   0   0   0   1
   2  21   0   0   0
   0   1  13   0   0
   0   0   0  17   0
   1   0   0   0  23
```

Number of Features	Network Structure	Number of Training Epochs	Correct Classification Rate on Training Set	Correct Classification Rate on Testing Set
4	[11 16]	15	90.53%	96%

Confusion matrix on training set

```
  16   0   0   0   4
   0  21   1   0   0
   0   0  13   0   0
   0   0   0  17   0
   2   0   0   2  19
```

Confusion matrix on Testing set

```
  21   0   0   0   1
   2  21   0   0   0
   0   0  14   0   0
   0   0   0  16   1
   0   0   0   0  24
```

Number of Features	Network Structure	Number of Training Epochs	Correct Classification Rate on Training Set	Correct Classification Rate on Testing Set
4	[9 2]	11	92.63%	94%

Confusion matrix on training set

```
 17   0   0   0   3
  0  22   0   0   0
  0   0  13   0   0
  0   0   0  17   0
  2   0   0   2  19
```

Confusion matrix on Testing set

```
 20   0   0   0   2
  2  21   0   0   0
  0   0  14   0   0
  0   0   0  16   1
  0   0   0   1  23
```

Number of Features	Network Structure	Number of Training Epochs	Correct Classification Rate on Training Set	Correct Classification Rate on Testing Set
8	[23 29]	13	96.84%	96%

Confusion matrix on training set

```
 20   0   0   0   0
  0  22   0   0   0
  0   0  13   0   0
  0   0   0  17   0
  2   0   0   1  20
```

Confusion matrix on Testing set

```
 22   0   0   0   0
  1  22   0   0   0
  0   0  14   0   0
  0   0   0  17   0
  2   0   0   1  21
```

Number of Features	Network Structure	Number of Training Epochs	Correct Classification Rate on Training Set	Correct Classification Rate on Testing Set
8	[18 6]	38	93.68%	96%

Confusion matrix on training set

```
  18    0    0    0    2

   0   22    0    0    0

   0    0   13    0    0

   0    0    0   17    0

   2    0    0    2   19
```

Confusion matrix on Testing set

```
  21    0    0    0    1

   1   22    0    0    0

   0    1   13    0    0

   0    0    0   16    1

   0    0    0    0   24
```

Number of Features	Network Structure	Number of Training Epochs	Correct Classification Rate on Training Set	Correct Classification Rate on Testing Set
8	[18 15]	21	94.74%	96%

Confusion matrix on training set

```
  18    0    0    1    1

   0   22    0    0    0

   0    0   13    0    0

   0    0    0   17    0

   2    0    0    1   20
```

Confusion matrix on Testing set

```
  22    0    0    0    0

   1   22    0    0    0

   0    1   13    0    0

   0    0    0   16    1

   1    0    0    0   23
```

Number of Features	Network Structure	Number of Training Epochs	Correct Classification Rate on Training Set	Correct Classification Rate on Testing Set
12	[21 20]	21	93.68%	97%

Confusion matrix on training set

```
   19    0    0    0    1

    0   21    1    0    0

    0    0   13    0    0

    0    0    0   16    1

    2    0    0    1   20
```

Confusion matrix on Testing set

```
   21    0    0    0    1

    1   22    0    0    0

    0    0   14    0    0

    0    0    0   17    0

    1    0    0    0   23
```

Number of Features	Network Structure	Number of Training Epochs	Correct Classification Rate on Training Set	Correct Classification Rate on Testing Set
12	[23 6]	37	94.74%	97%

Confusion matrix on training set

```
   18    0    0    0    2

    0   22    0    0    0

    0    0   13    0    0

    0    0    0   17    0

    3    0    0    0   20
```

Confusion matrix on Testing set

```
   21    0    0    0    1

    1   22    0    0    0

    0    1   13    0    0

    0    0    0   17    0

    0    0    0    0   24
```

Number of Features	Network Structure	Number of Training Epochs	Correct Classification Rate on Training Set	Correct Classification Rate on Testing Set
12	[23 15]	15	96.84%	%97

Confusion matrix on training set

```
19   0   0   0   1
 0  22   0   0   0
 0   0  13   0   0
 0   0   0  17   0
 1   0   0   1  21
```

Confusion matrix on Testing set

```
22   0   0   0   0
 1  21   0   0   1
 0   1  13   0   0
 0   0   0  17   0
 0   0   0   0  24
```

Number of Features	Network Structure	Number of Training Epochs	Correct Classification Rate on Training Set	Correct Classification Rate on Testing Set
16	[37 5]	47	96.84%	98%

Confusion matrix on training set

```
19   0   0   0   1
 0  22   0   0   0
 0   1  12   0   0
 0   0   0  17   0
 0   0   0   1  22
```

Confusion matrix on Testing set

```
22   0   0   0   0
 0  22   0   0   1
 0   1  13   0   0
 0   0   0  17   0
 0   0   0   0  24
```

Number of Features	Network Structure	Number of Training Epochs	Correct Classification Rate on Training Set	Correct Classification Rate on Testing Set
16	[30 8]	20	94.74%	97%

Confusion matrix on training set

```
 19   0   0   0   1

  0  22   0   0   0

  0   0  13   0   0

  0   0   0  17   0

  2   0   0   2  19
```

Confusion matrix on Testing set

```
 22   0   0   0   0

  2  21   0   0   0

  0   1  13   0   0

  0   0   0  17   0

  0   0   0   0  24
```

Number of Features	Network Structure	Number of Training Epochs	Correct Classification Rate on Training Set	Correct Classification Rate on Testing Set
16	30	52	98.95%	98%

Confusion matrix on training set

```
 20   0   0   0   0

  0  22   0   0   0

  0   0  13   0   0

  0   0   0  17   0

  0   0   0   1  22
```

Confusion matrix on Testing set

```
 22   0   0   0   0

  0  22   0   0   1

  0   1  13   0   0

  0   0   0  17   0

  0   0   0   0  24
```

Number of Features	Network Structure	Number of Training Epochs	Correct Classification Rate on Training Set	Correct Classification Rate on Testing Set
20	[34 5]	14	95.79%	97%

Confusion matrix on training set

```
18   1   0   0   1

 0  22   0   0   0

 0   0  13   0   0

 0   0   0  17   0

 2   0   0   0  21
```

Confusion matrix on Testing set

```
21   0   0   0   1

 1  22   0   0   0

 0   1  13   0   0

 0   0   0  17   0

 0   0   0   0  24
```

Number of Features	Network Structure	Number of Training Epochs	Correct Classification Rate on Training Set	Correct Classification Rate on Testing Set
20	[34 10]	27	94.74%	96%

Confusion matrix on training set

```
19   0   0   0   1

 0  22   0   0   0

 0   0  13   0   0

 0   0   0  17   0

 3   0   0   1  19
```

Confusion matrix on Testing set

```
21   0   0   0   1

 2  21   0   0   0

 0   1  13   0   0

 0   0   0  17   0

 0   0   0   0  24
```

Number of Features	Network Structure	Number of Training Epochs	Correct Classification Rate on Training Set	Correct Classification Rate on Testing Set
20	[35 5]	59	98.95%	94%

Confusion matrix on training set

20	0	0	0	0
0	22	0	0	0
0	1	12	0	0
0	0	0	17	0
0	0	0	0	23

Confusion matrix on Testing set

22	0	0	0	0
2	21	0	0	0
0	1	13	0	0
1	0	0	14	2
0	0	0	0	24

Adding in an extra layer can be super helpful or it could make the NN slower than needed. For example, w/ the tables for 16 features. There is one NN that has one hidden layer w/ 30 neurons in it. This one seemed to preform just fine without using a second layer; however, this may not be the case the next time this NN is made. As or generalizations. The only ones I suspect of generalization are any NN who's training rate was equal to or above the testing rate. This makes me believe the AI was just remembering instead of looking at features and deciding. Finally for the most confusing class, I would say that class five is the most confusing for the NNs and class two is the seconds most confusing.