Q1 Download the wine data set from the UCI database. Train a network with 2 hidden units on the data, holding out 10 items from each category for testing.

```
% set up
clear;
rs = rand()*100000
```

```
rs =
       21886
```

```
% obtain data
wine = str2num(urlread("https://archive.ics.uci.edu/ml/machine-learning-databases/wine/wine.dat
% turn 1,2,3 output to [1 0 0], [0 1 0], [0 0 1]
data = [dummyvar(categorical(wine(:,1))) wine(:,2:end)];

% catalog data to test_set(10), and training_set(whatever left)
test_set = [];
valid_set = [];
for i = 1:3 % for each category
    cat_wine = data(data(:, i) > 0, :)
    % Randomly select unrepeatable 10 item for testing
    indexes = randperm(size(cat_wine, 1), 20)
    test_set = [test_set; cat_wine(indexes(:,1:10), :)];
    valid_set = [valid_set; cat_wine(indexes(:,11:20), :)];
end
```

```
cat_wine = 59×16
          1          0          0      14.23       1.71       2.43 ···
          1          0          0       13.2       1.78       2.14
          1          0          0      13.16       2.36       2.67
          1          0          0      14.37       1.95        2.5
          1          0          0      13.24       2.59       2.87
          1          0          0       14.2       1.76       2.45
          1          0          0      14.39       1.87       2.45
          1          0          0      14.06       2.15       2.61
          1          0          0      14.83       1.64       2.17
          1          0          0      13.86       1.35       2.27
     ⋮

cat_wine = 71×16
          0          1          0      12.37       0.94       1.36 ···
          0          1          0      12.33        1.1       2.28
          0          1          0      12.64       1.36       2.02
          0          1          0      13.67       1.25       1.92
          0          1          0      12.37       1.13       2.16
          0          1          0      12.17       1.45       2.53
          0          1          0      12.37       1.21       2.56
          0          1          0      13.11       1.01        1.7
          0          1          0      12.37       1.17       1.92
          0          1          0      13.34       0.94       2.36
     ⋮

cat_wine = 48×16
          0          0          1      12.86       1.35       2.32 ···
          0          0          1      12.88       2.99        2.4
```

```
        0           0           1       12.81       2.31       2.4
        0           0           1        12.7       3.55      2.36
        0           0           1       12.51       1.24      2.25
        0           0           1        12.6       2.46       2.2
        0           0           1       12.25       4.72      2.54
        0           0           1       12.53       5.51      2.64
        0           0           1       13.49       3.59      2.19
        0           0           1       12.84       2.96      2.61
    ⋮
```

```
% training = original - test
train_set = setdiff(data, [test_set; valid_set], "rows");
isequal(sortrows(data),sortrows([train_set;valid_set;test_set])) % sanity check
```

```
ans = logical

   1
```

```
% reshape data
training.smat = train_set(:,4:end);
training.tmat = train_set(:,1:3);

testing.smat = test_set(:,4:end);
testing.tmat = test_set(:,1:3);

validation.smat = valid_set(:,4:end);
validation.tmat = valid_set(:,1:3);
```

a. Show the error curves over time

```
% init random 1 layer network with stimulus(13) input, 2 hidden, and 3 output
q1net0 = initnet3(13,2,3,1,1,rs)
```

```
q1net0 = struct with fields:
      wih: [2×13 double]
    hbias: [0.095941 -0.0036671]
    whout: [3×2 double]
    obias: [-0.10738 0.27788 -0.30611]
```

```
q1net0.wih
```

```
ans = 2×13
     -0.43501    -0.017961      0.27792      0.14222     -0.30811     0.065113 ⋯
     0.089276      0.39934      0.36728      0.29855     -0.45169     -0.37207
```

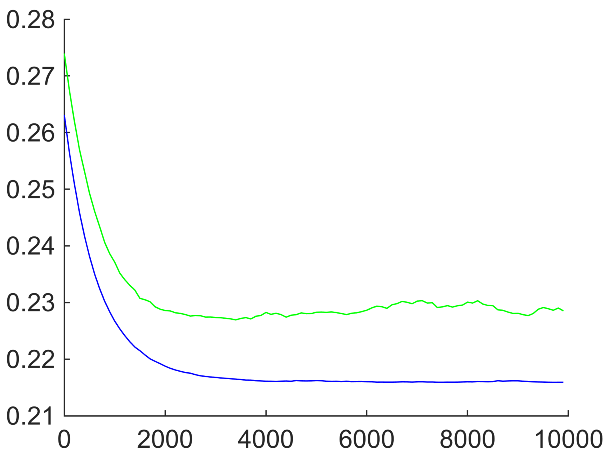
```
q1net0.whout
```

```
ans = 3×2
      0.47371     0.020177
    -0.081111      0.24464
       0.3105     -0.25053
```

```
niter = 10000; % n-th iteration
eta = 0.001; % Eta, learning rate
```

```
nlev = 0; % noise level
eps = 0.02; % stopping criterion
[bestnet,finalnet,errmat] = bp3trvats(q1net0,training,validation,testing,niter,eta,nlev,eps);

figure
err = errmat(1:niter/100:niter,:);
hold on
plot(err(:,1),err(:,2),'b')
plot(err(:,1),err(:,3),'r')
plot(err(:,1),err(:,4),'g')
hold off
```



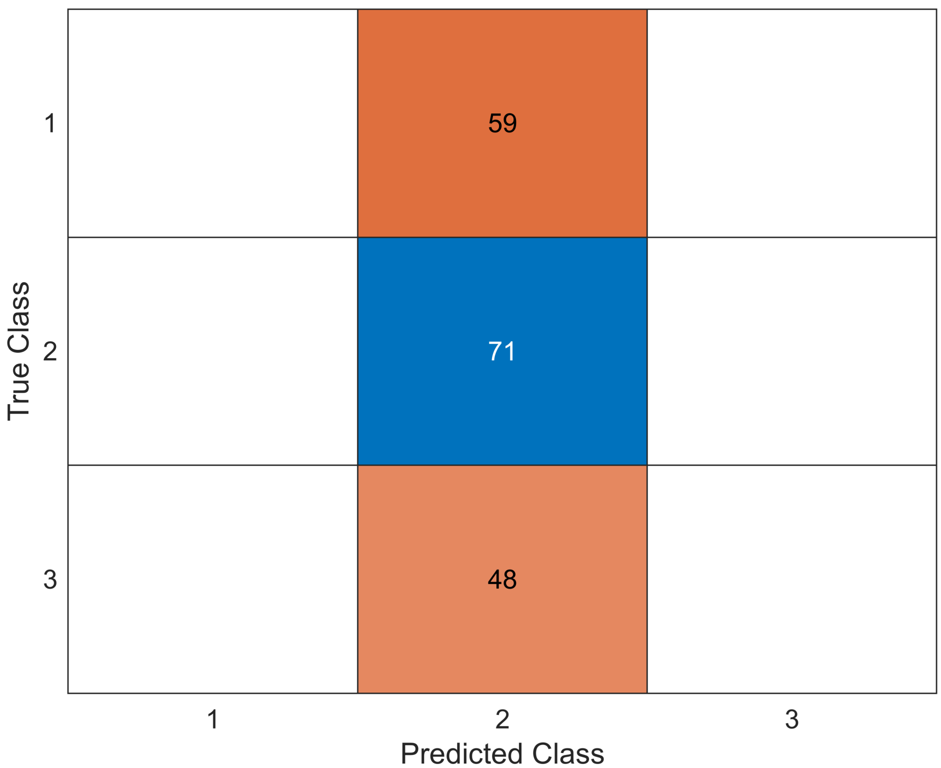
b. Show the final confusion matrix (3 categories)

```
target = data(:,1:3);
stim = data(:,4:end);

pats.smat = stim;
pats.tmat = target;
bestact=forw3(bestnet, pats)
```

```
bestact = struct with fields:
    stim: [178×13 double]
     hid: [178×2 double]
     out: [178×3 double]
```

```
% build confusion matrix
[~,t] = max(target, [], 2);
[~,o] = max(bestact.out,[],2);
figure
confusionchart(t,o)
```

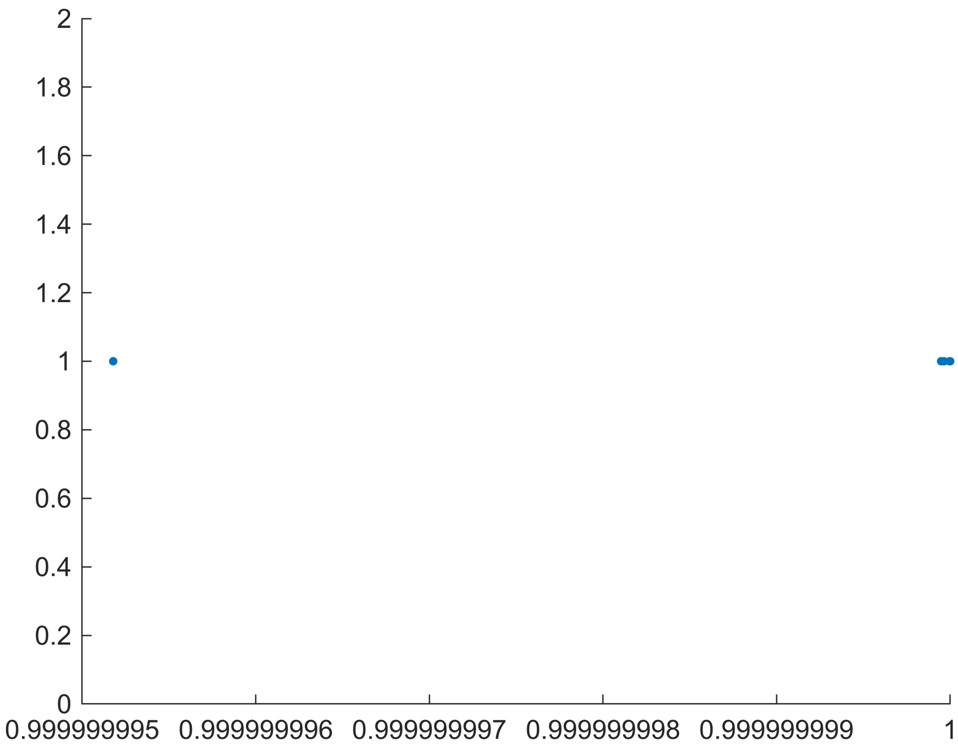


c. Plot the hidden unit representations using different colors for the wine type.

```
act = forw3(bestnet,training)
```

```
act = struct with fields:
    stim: [118×13 double]
     hid: [118×2 double]
     out: [118×3 double]
```

```
act.hid;
color = [training.tmat,zeros(size(training.tmat)),1-training.tmat];
scatter(act.hid(:,1),act.hid(:,2),10,"filled")
```



Q2 The following table summarizes BMI data from 300 patients, of which 100 have been diagnosed with breast cancer.

```
%{
BMI (kg/m3)   <18 18-20 20-22 22-24 24-26 26-28 28-30 30-32 32-34 34-36 36-38 38-40
Cancer (n=100) 0    1     4     10     19     19     13     13     4     10     6     1
No Cancer       2    21     60     37     27     26     10     3     8     2     3     1
%}

bmi = [    "<18"     "18-20"     "20-22"     "22-24"     "24-26"     "26-28"     "28-30"     "30-32"
cancer = [0    1     4     10     19     19     13     13     4     10     6     1];
no_cancer = [2     21     60     37     27     26     10     3     8     2     3     1];
```

a. Compute the number of true positives, true negatives, false positives and false negatives at each BMI level.

```
cdata = []
```

```
cdata =
     []
```

```
b = 18;
for i=1:length(bmi)
    cdata = [cdata; [b sum(cancer(:,i+1:end)) sum(cancer(:,1:i)) sum(no_cancer(:,i+1:end)) sum(
    b = b+2;
end
```

```
["bmi_cutoff" "tp" "tn" "fp" "fn"; cdata]
```

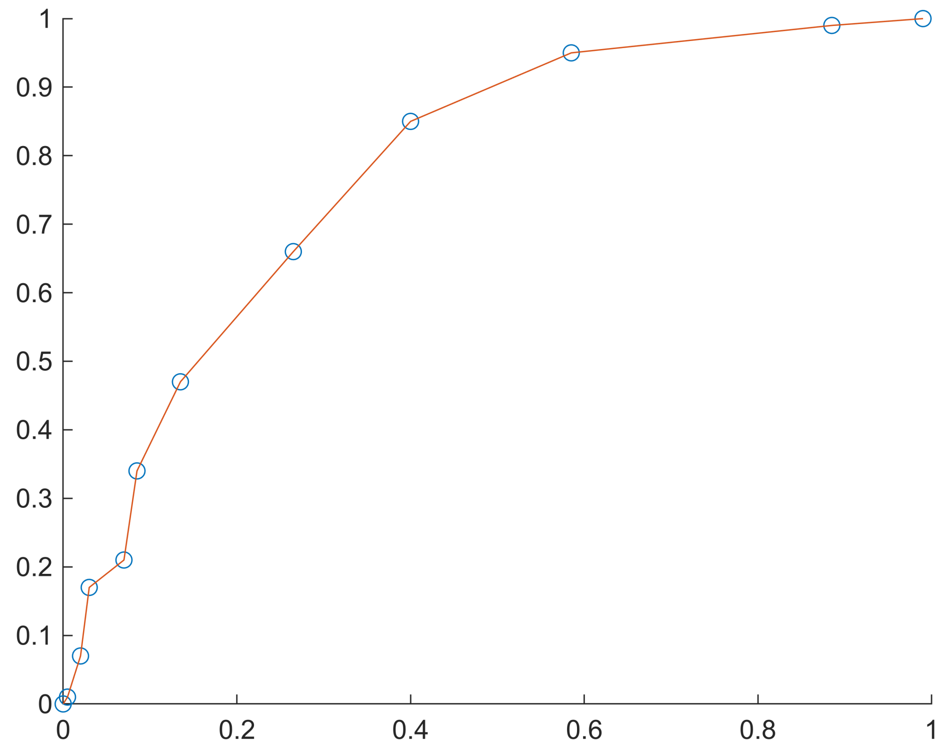
```
ans = 13×5 string
"bmi_cutoff" "tp"         "tn"         "fp"         "fn"
"18"         "100"        "0"          "198"        "2"
"20"         "99"         "1"          "177"        "23"
"22"         "95"         "5"          "117"        "83"
"24"         "85"         "15"         "80"         "120"
"26"         "66"         "34"         "53"         "147"
"28"         "47"         "53"         "27"         "173"
"30"         "34"         "66"         "17"         "183"
"32"         "21"         "79"         "14"         "186"
"34"         "17"         "83"         "6"          "194"
     ⋮
```

b. Construct an ROC curve.

```
tpr = [];
fpr = [];
for i = 1:size(cdata,1)
    tpr = [tpr; cdata(i,2)/100];
    fpr = [fpr; cdata(i,4)/200];
end
["tpr" "fpr"; tpr fpr]
```

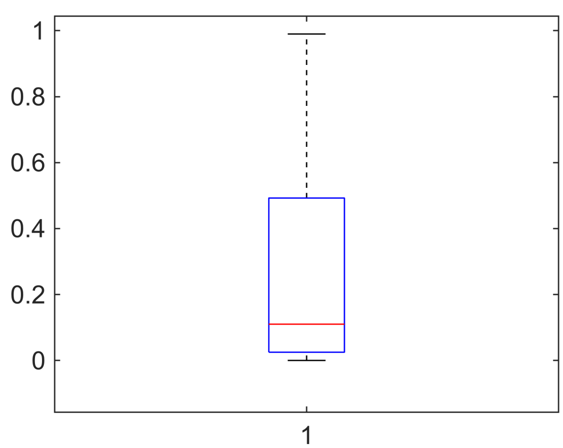
```
ans = 13×2 string
"tpr"        "fpr"
"1"          "0.99"
"0.99"       "0.885"
"0.95"       "0.585"
"0.85"       "0.4"
"0.66"       "0.265"
"0.47"       "0.135"
"0.34"       "0.085"
"0.21"       "0.07"
"0.17"       "0.03"
     ⋮
```

```
figure
hold on
scatter(fpr,tpr)
plot(fpr,tpr)
hold off
```

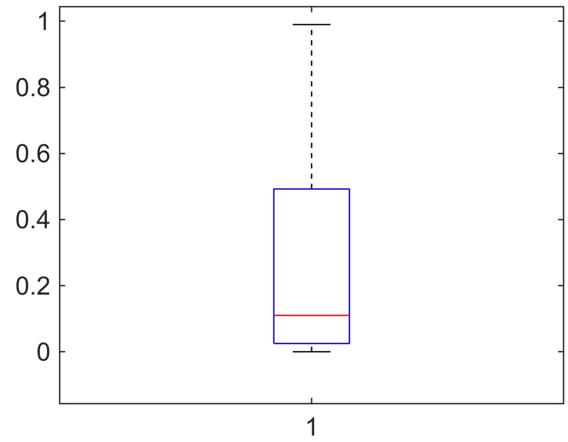


c. How would you generate an approximate boxplot for the two categories?

```
boxplot(fpr)
```




```
boxplot(fpr)
```



3. The code below constructs and displays a training set and test set for a simple 2-D classification task.

a. Paste the code into Matlab to generate pattern sets train and test.

```
train.smat=2*rand(30,2)-1
```

```
train = struct with fields:

    smat: [30×2 double]
```

```
dtrain = sqrt(diag(train.smat*train.smat'));
train.tmat=(dtrain<.85).*(dtrain>.45)
```

```
train = struct with fields:
    smat: [30×2 double]
    tmat: [30×1 double]
```

```
ktrain=[train.tmat,zeros(30,1),1-train.tmat];
figure
scatter(train.smat(:,1),train.smat(:,2),300,ktrain,'filled')
test.smat=2*rand(20,2)-1
```

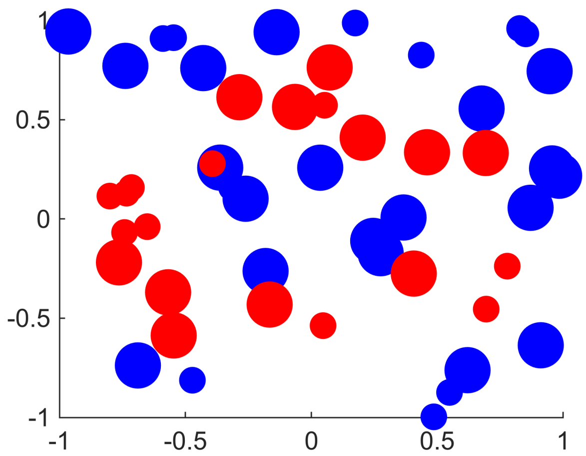
```
test = struct with fields:

    smat: [20×2 double]
```

```
dtest = sqrt(diag(test.smat*test.smat'));
test.tmat=(dtest<.85).*(dtest>.45)
```

```
test = struct with fields:
    smat: [20×2 double]
    tmat: [20×1 double]
```

```
ktest=[test.tmat,zeros(20,1),1-test.tmat];
hold on
scatter(test.smat(:,1),test.smat(:,2),100,ktest,'filled')
```



b. Initialize a network with 4 hidden units, and run bp3 for n iterations (where 10000<n<20000) at a learning rate dt (where .02 < dt < .2) using the pattern set train.

```
% init network
q3net0 = initnet3(2,4,1,2,2,rs);
% train
q3net10k = bp3(q3net0,train,10000,.1,0,rs)
```

```
q3net10k = struct with fields:
      wih: [4×2 double]
    hbias: [3.2135 -2.9748 0.17731 -0.77433]
    whout: [2.7299 -2.0062 -0.2237 0.90881]
    obias: -4.0992
```

c. Use forw3 to test the final network on the test pattern set.

```
% applying the testing pattern to the trained network
testresult = forw3(q3net10k,test)
```

```
testresult = struct with fields:
    stim: [20×2 double]
     hid: [20×4 double]
     out: [20×1 double]
```

```
testresult.hid
```

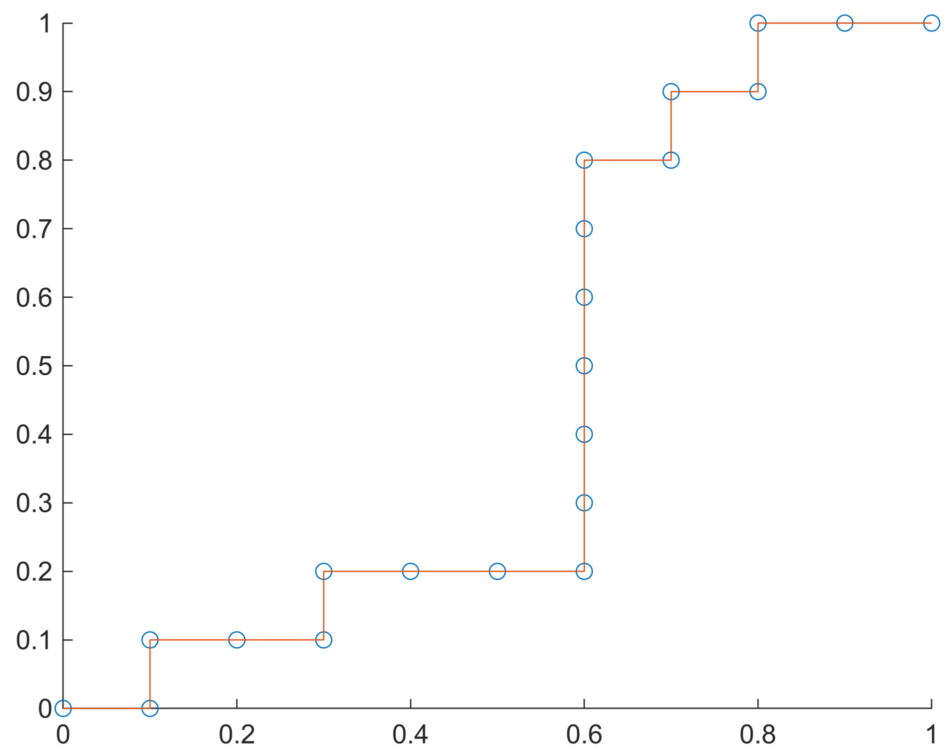
```
ans = 20×4
      0.99644     -0.44546      0.18129     -0.73883
      0.98209     -0.73429      0.15695     -0.52889
      0.96734     -0.95568     -0.13719     -0.84834
    -0.022376     -0.99677     -0.25063     -0.44836
       0.9533     -0.70515      0.26004    -0.032049
      0.43662     -0.88037      0.31821       0.7082
      0.48989     -0.86611      0.32721      0.70419
      0.99409     -0.65381      0.11667     -0.76034
      0.84718     -0.77156      0.31119      0.42243
     -0.13027     -0.99029    -0.034867      0.18643
        ⋮
```

```
testresult.out
```

```
ans = 20×1
      0.23195
      0.38679
       0.4301
     0.075024
      0.45775
      0.36156
      0.38759
      0.31198
      0.51892
     0.091881
        ⋮
```

d. Generate an ROC curve from the responses to the test set.

```
[tpr,fpr,thresholds]=roc(test.tmat', testresult.out');
figure
hold on
scatter(fpr,tpr)
plot(fpr,tpr)
```



Attachements

```
function [bestnet,finalnet,errmat]=bp3trvats(net0,trlist,valist,tslist,niter,eta,nlev,eps)
netk=net0;
maxiter=niter;
```

```matlab
ase=2*eps ; % not use in this function
valbest=10;
i=0 ;
errmat=[] ;
while ((ase>eps)&&(i<maxiter))
    netk=cyc3(netk,trlist,eta,nlev) ; % Training procedure

    activ=forw3(netk,trlist) ;
    diff = trlist.tmat-activ.out;
    sse=sum(sum(diff.*diff));
    trerr=sse/prod(size(diff)) ; % Training error

    activ=forw3(netk,valist) ;
    diff = valist.tmat-activ.out;
    sse=sum(sum(diff.*diff));
    valerr=sse/prod(size(diff)) ; % Validation error
    if (valerr<valbest)
        % Save the network with the lowest validation error
        ibest=i ;
        valbest=valerr;
        bestnet=netk;
    end

    activ=forw3(netk,tslist) ;
    diff = tslist.tmat-activ.out;
    sse=sum(sum(diff.*diff));
    tserr=sse/prod(size(diff)) ; % Test error

    errmat=[errmat;[i,trerr,valerr,tserr]];
    i=i+1 ;
end
finalnet=netk;
finalnet.iter=i;
finalnet.err=ase;
bestnet.iter=ibest; % fixed
bestnet.err=valbest; % fixed
end
```