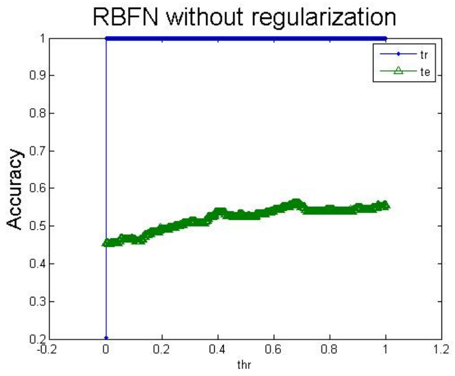
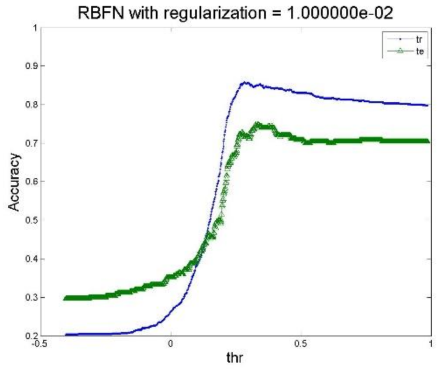
Q2.

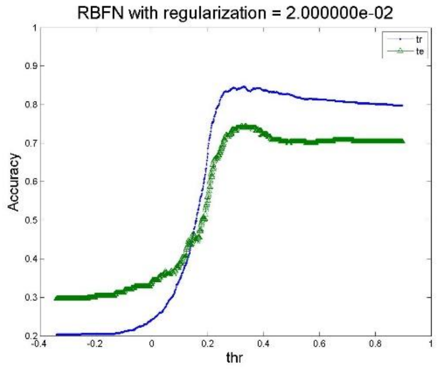
Matric number: A0191818W. Choose the class 1 and 8.

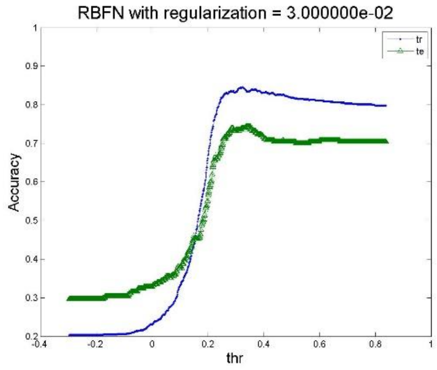
a) RBF without regularization
The performance is quite bad without regularization.

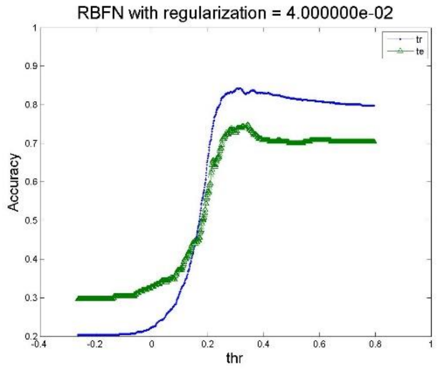


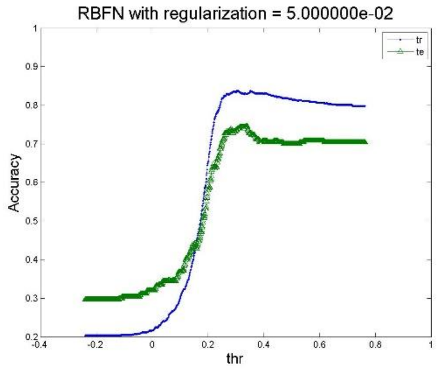
RBF with regularization
The performance is quite well.

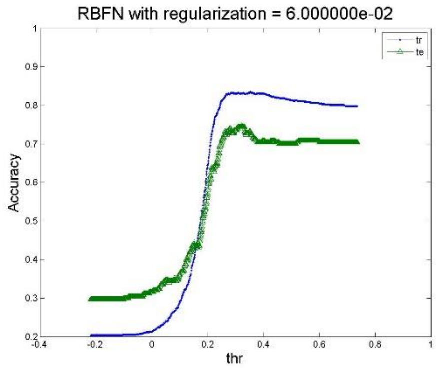


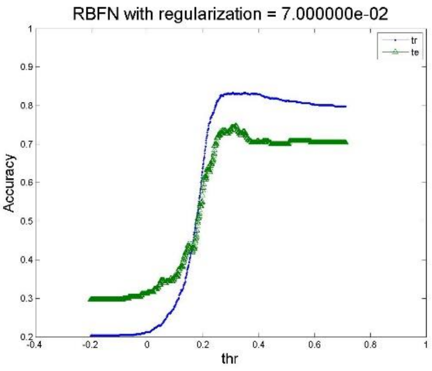




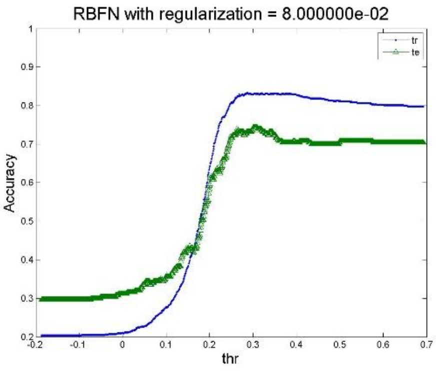




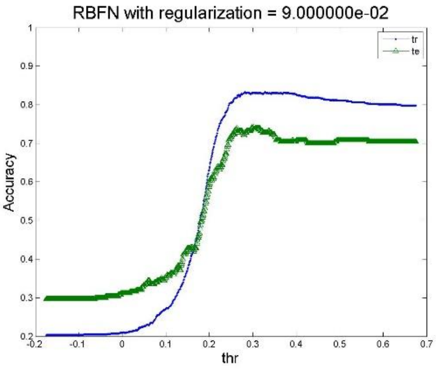




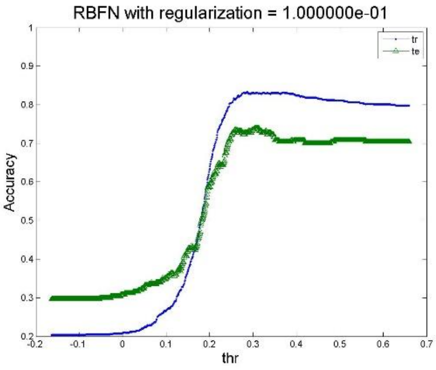
RBFN with regularization = 7.000000e-02



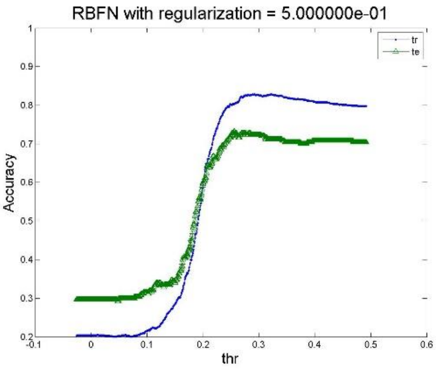
RBFN with regularization = 8.000000e-02



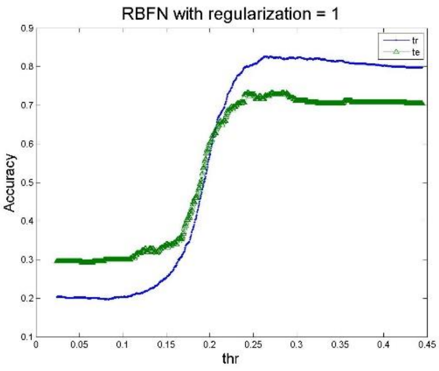
RBFN with regularization = 9.000000e-02



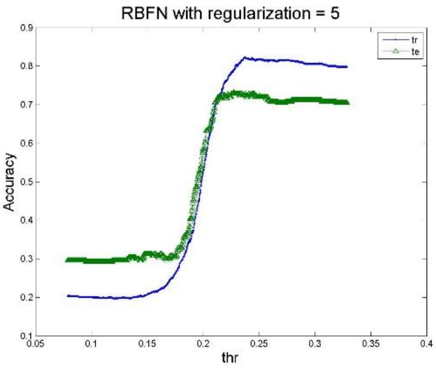
RBFN with regularization = 1.000000e-01



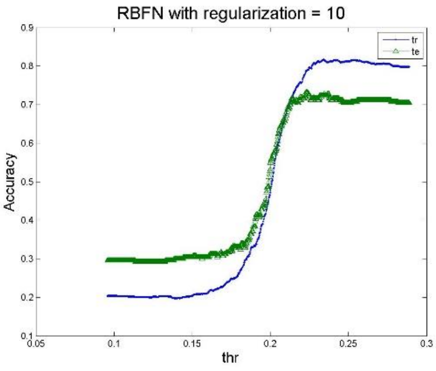
RBFN with regularization = 5.000000e-01



RBFN with regularization = 1



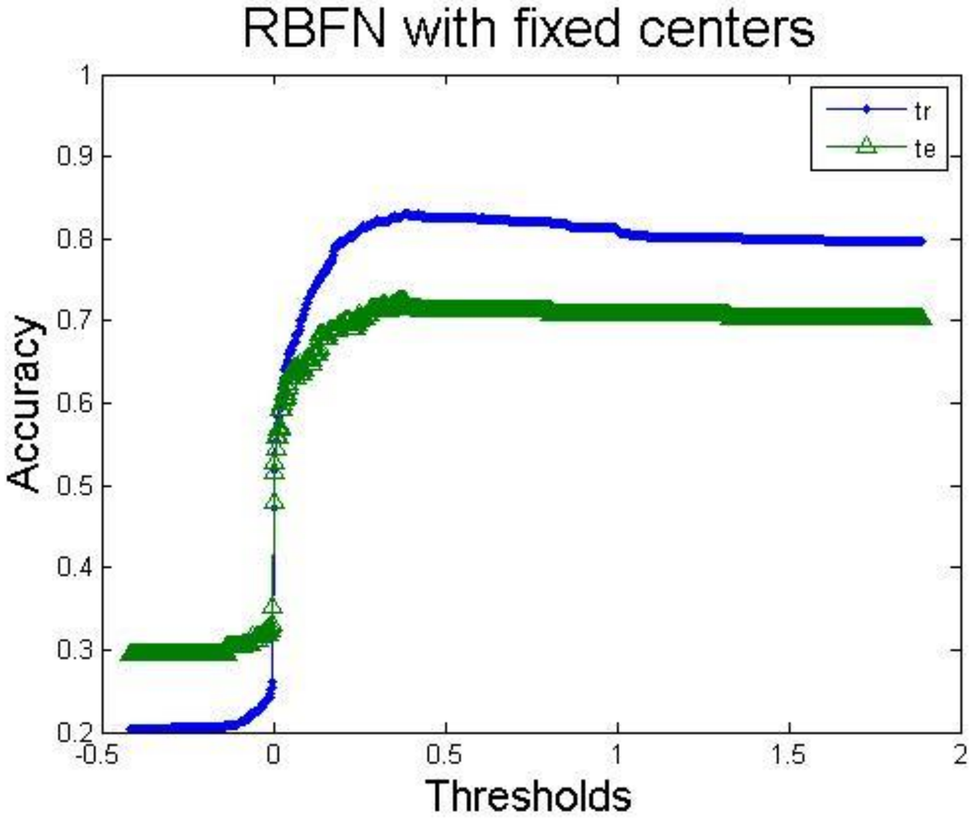
RBFN with regularization = 5



RBFN with regularization = 10

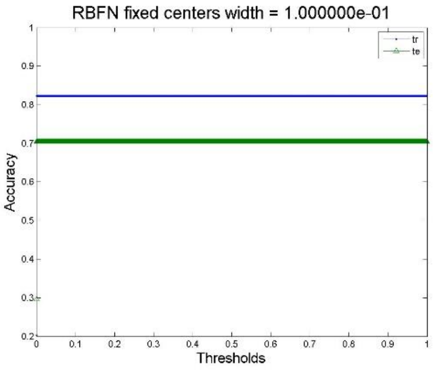
b) Use the strategy of “Fixed Centers Selected at Random” with widths fixed at an appropriate size. The result is shown below.

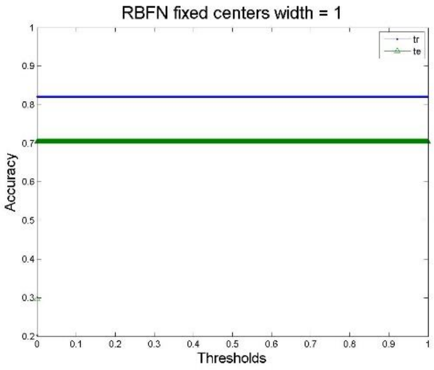
Compare to the result of a), the this performance is better obviously.

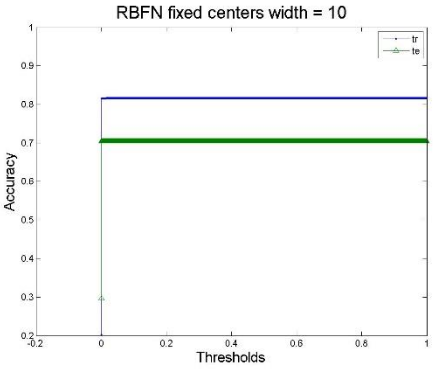


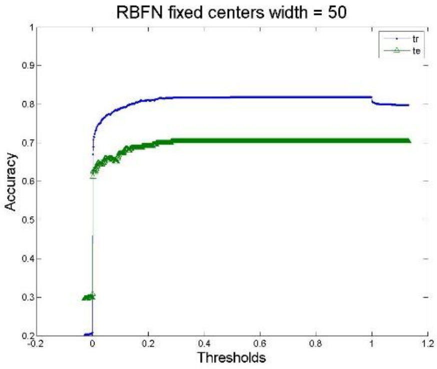
Vary the value of width from 0.1 to 10000, the result is shown below.

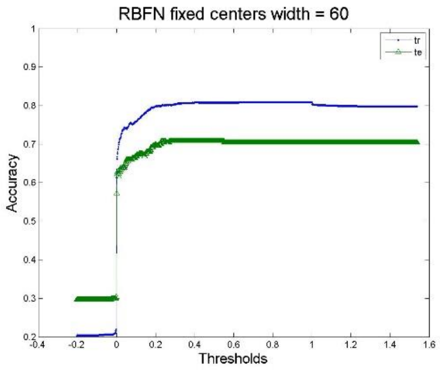
When the value of width is around 100, the performance of RBFN is the best.

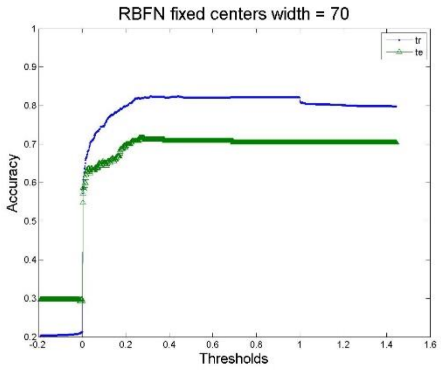


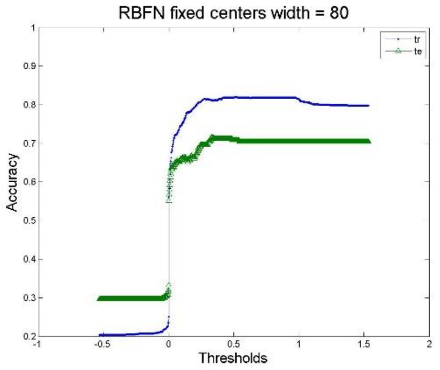


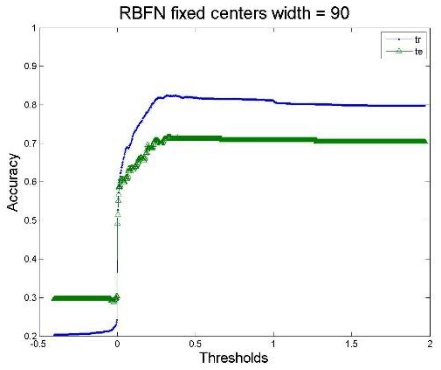


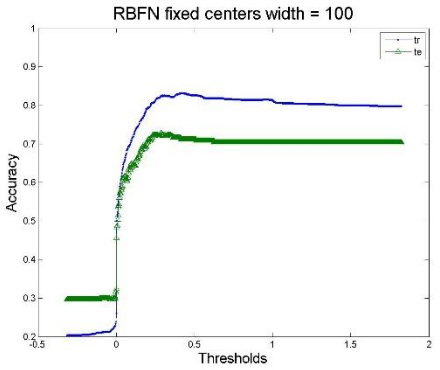


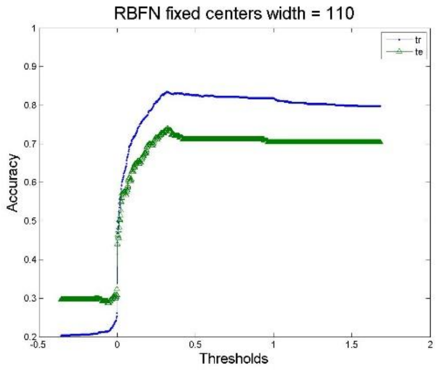

RBFN fixed centers width = 60

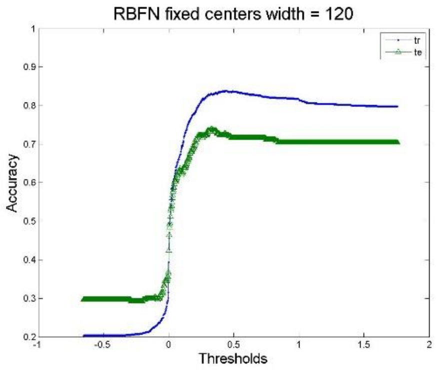

RBFN fixed centers width = 70

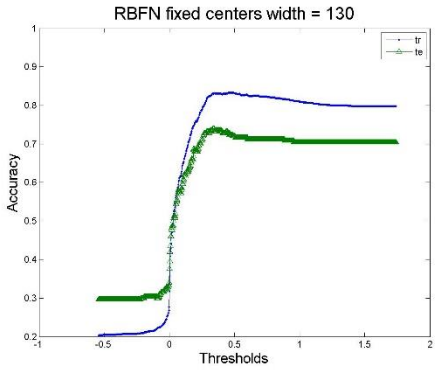

RBFN fixed centers width = 80

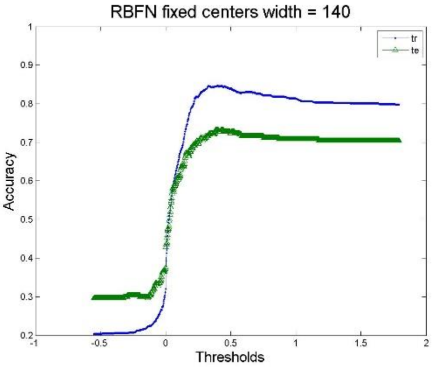

RBFN fixed centers width = 90

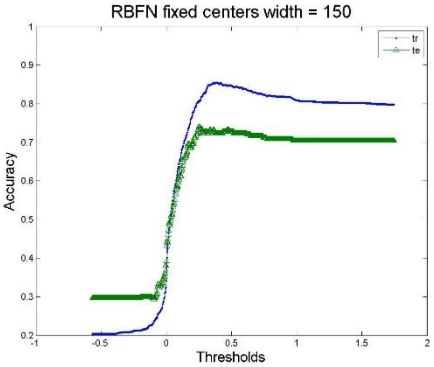

RBFN fixed centers width = 100

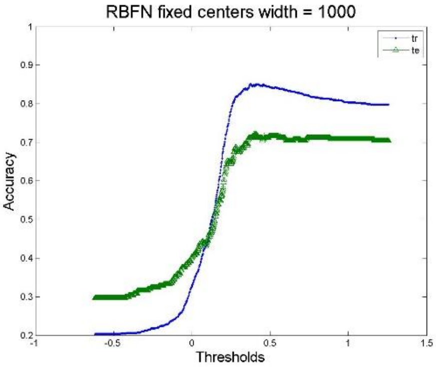

RBFN fixed centers width = 110

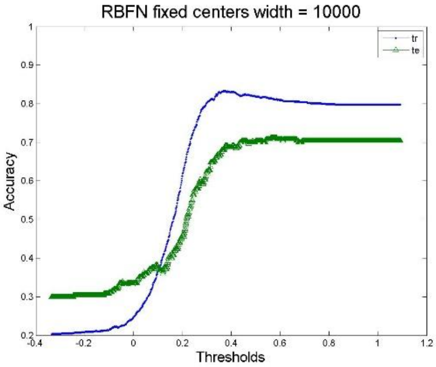

RBFN fixed centers width = 120


RBFN fixed centers width = 130

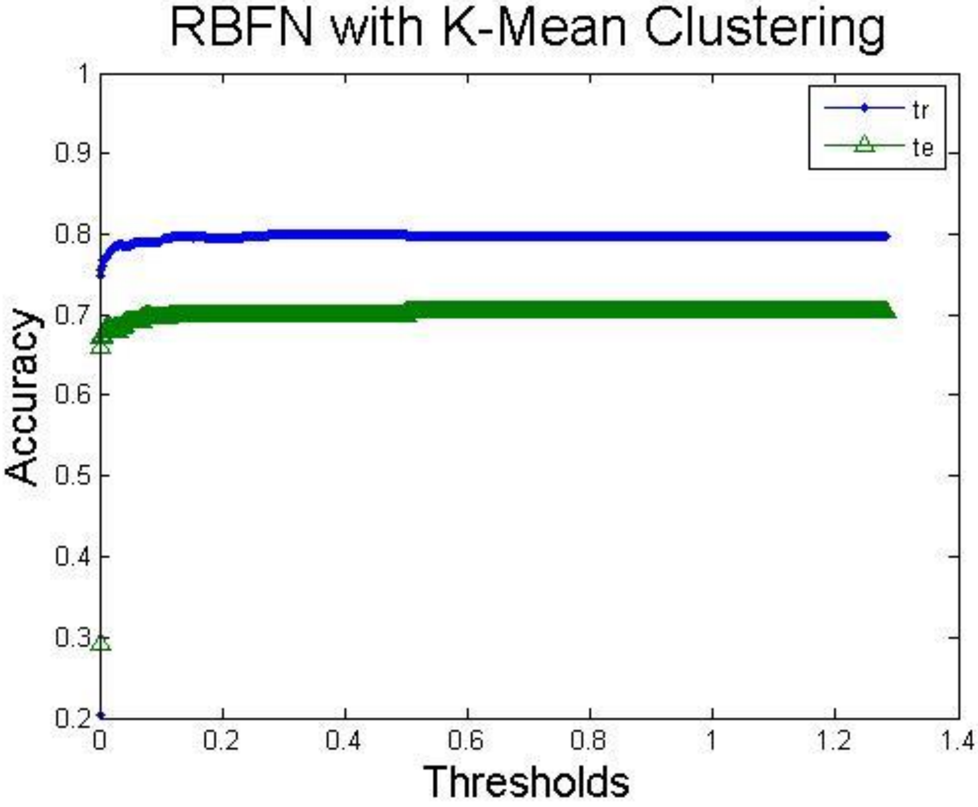

RBFN fixed centers width = 140
Accuracy
Thresholds
tr
te
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
-1
-0.5
0
0.5
1
1.5
2


RBFN fixed centers width = 150
Accuracy
Thresholds
tr
te
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
-1
-0.5
0
0.5
1
1.5
2

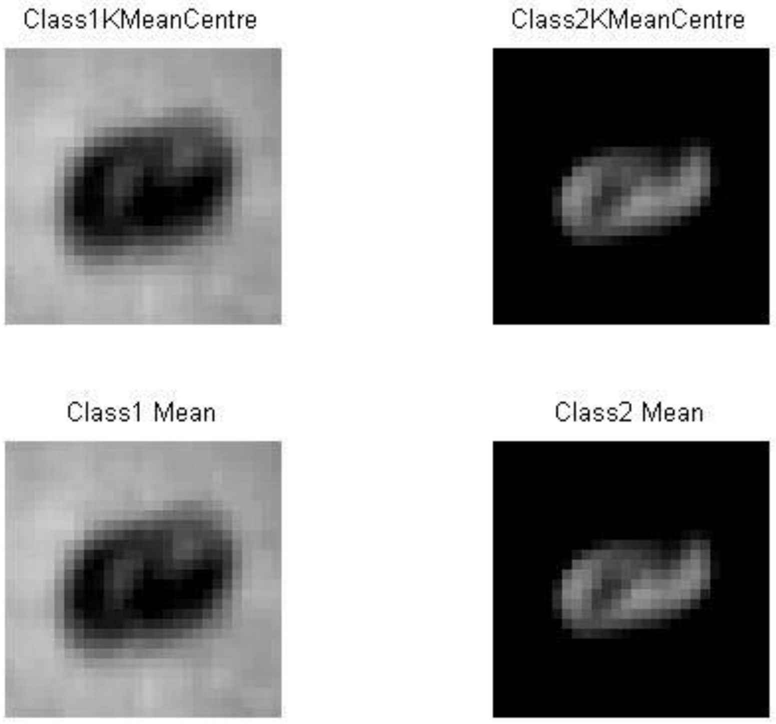

RBFN fixed centers width = 1000
Accuracy
Thresholds
tr
te
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
-1
-0.5
0
0.5
1
1.5


RBFN fixed centers width = 10000
Accuracy
Thresholds
tr
te
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1
1.2

c) Try classical “K-Mean Clustering”, the result is shown below.
The performance is quite well when use “K-Mean Clustering”



Visualize the obtained centres and compare them to the mean of training images of each class. It could find that they are almost the same.


Class1KMeanCentre

Class2KMeanCentre

Class1 Mean

Class2 Mean

The following is the MATLAB code of Q1_1:

```
%%
%INPUT
clc;
clear all;
close all;
load('mnist_m.mat');
regularization = 5;
Index_train = find(train_classlabel==1|train_classlabel==8);
Index_test = find(test_classlabel==1|test_classlabel==8);
for i=1:203
   train_classlabel(Index_train(i)) = 1;
end
for i=1:1000
   if train_classlabel(i)~=1
      train_classlabel(i) = 0;
   end
end
for i=1:48
   test_classlabel(Index_train(i)) = 1;
end
for i=1:250
   if test_classlabel(i)~=1
      test_classlabel(i) = 0;
   end
end
train_data = double(train_data);
test_data = double(test_data);
train_classlabel = double(train_classlabel);
test_classlabel = double(test_classlabel);

train_data_mean=mean(mean(train_data,2));
sigma=std2(train_data);
train_data=(train_data-train_data_mean)./sigma;
test_data=(test_data-train_data_mean)./sigma;
%%
%CACULATION
function_RBF = exp(-
(dist(train_data',train_data)).^2/20000);%Gaussian Functions
w =
pinv(function_RBF'*function_RBF+regularization*eye(length(train_class
label)))*function_RBF'*train_classlabel';%Weight matrix
function_RBF_test = exp(-(dist(test_data',train_data)).^2/20000);
```

```
%%
%Evaluate
TrPred = function_RBF*w;
TePred = function_RBF_test*w;

TrAcc = zeros(1,1000);
TeAcc = zeros(1,1000);
thr = zeros(1,1000);
TrN = length(train_classlabel);
TeN = length(test_classlabel);
for i = 1:1000
t = (max(TrPred)-min(TrPred)) * (i-1)/1000 + min(TrPred);
thr(i) = t;
TrAcc(i) = (sum(train_classlabel(TrPred<t)==0) +
sum(train_classlabel(TrPred>=t)==1)) / TrN;
TeAcc(i) = (sum(test_classlabel(TePred<t)==0) +
sum(test_classlabel(TePred>=t)==1)) / TeN;
end
plot(thr,TrAcc,'.-',thr,TeAcc,'^-');
legend('tr','te');
hold on;
discription1 = sprintf('RBFN with regularization
= %d',regularization);
title(discription1,'FontSize',20);
xlabel('thr','FontSize',16);
ylabel('Accuracy','FontSize',16);
discription2 = sprintf('Q2_1_%d.jpg',regularization);
saveas(gcf,discription2);
close;
```

The following is the MATLAB code of Q1_2:

```
%%
%INPUT
clc;
clear all;
close all;
load('mnist_m.mat');
Index_train = find(train_classlabel==1|train_classlabel==8);
Index_test = find(test_classlabel==1|test_classlabel==8);
for i=1:203
    train_classlabel(Index_train(i)) = 1;
end
for i=1:1000
    if train_classlabel(i)~=1
        train_classlabel(i) = 0;
    end
end
for i=1:48
    test_classlabel(Index_train(i)) = 1;
end
for i=1:250
    if test_classlabel(i)~=1
        test_classlabel(i) = 0;
    end
end
train_data = double(train_data);
test_data = double(test_data);
train_classlabel = double(train_classlabel);
test_classlabel = double(test_classlabel);

train_data_mean=mean(mean(train_data,2));
sigma=std2(train_data);
train_data=(train_data-train_data_mean)./sigma;
test_data=(test_data-train_data_mean)./sigma;

temp = randperm(size(train_data,2));
mu = train_data(:,temp(1:100));%fixed centres selected at random
dm = max(max(dist(mu',mu)));
%%
%CACULATION
function_RBF = exp(-size(mu,2)/(dm^2)*(dist(train_data',mu)).^2);%RBF
Functions
w =
pinv(function_RBF'*function_RBF)*function_RBF'*train_classlabel';%Wei
ght matrix
function_RBF_test = exp(-size(mu,2)/(dm^2)*(dist(test_data',mu)).^2);
%%
%Evaluate
TrPred = function_RBF*w;
TePred = function_RBF_test*w;

TrAcc = zeros(1,1000);
TeAcc = zeros(1,1000);
thr = zeros(1,1000);
TrN = length(train_classlabel);
TeN = length(test_classlabel);
for i = 1:1000
t = (max(TrPred)-min(TrPred)) * (i-1)/1000 + min(TrPred);
thr(i) = t;
TrAcc(i) = (sum(train_classlabel(TrPred<t)==0) +
sum(train_classlabel(TrPred>=t)==1)) / TrN;
TeAcc(i) = (sum(test_classlabel(TePred<t)==0) +
sum(test_classlabel(TePred>=t)==1)) / TeN;
end
plot(thr,TrAcc,'.-',thr,TeAcc,'^-');
legend('tr','te');
hold on;
title('RBFN with fixed centers','FontSize',20);
xlabel('Thresholds','FontSize',1);
ylabel('Accuracy','FontSize',16);
```

The following is the MATLAB code of Q1_3:

```
%%
%INPUT
clc;
clear all;
close all;
load('mnist_m.mat');
Index_train = find(train_classlabel==1|train_classlabel==8);
Index_test = find(test_classlabel==1|test_classlabel==8);
for i=1:203
    train_classlabel(Index_train(i)) = 1;
end
for i=1:1000
    if train_classlabel(i)~=1
        train_classlabel(i) = 0;
    end
end
for i=1:48
    test_classlabel(Index_train(i)) = 1;
end
for i=1:250
    if test_classlabel(i)~=1
        test_classlabel(i) = 0;
    end
end
train_data = double(train_data);
test_data = double(test_data);
train_classlabel = double(train_classlabel);
test_classlabel = double(test_classlabel);

train_data_mean=mean(mean(train_data,2));
sigma=std2(train_data);
train_data=(train_data-train_data_mean)./sigma;
test_data=(test_data-train_data_mean)./sigma;

no_clusters = 2;
uk=rand(size(train_data,1),no_clusters);
%%
%CACULATION
for i=1:100
    uk1=uk;
    function_RBF = dist(train_data',uk);
    [m,n]=min(function_RBF,[],2);
    i1=find(n==1);
    i2=find(n==2);
    uk(:,1)=mean(train_data(:,i1),2);
    uk(:,2)=mean(train_data(:,i2),2);
    err=norm(uk-uk1);
    if err<0.001
      break
    end
end
m1 = 100;
dm = dist(uk(:,1)',uk(:,2));
function_RBF = exp(-m1/ dm^2*dist(train_data',uk).^2);
w =
pinv(function_RBF'*function_RBF)*function_RBF'*train_classlabel'; %We
ight matrix
function_RBF_test = exp(-m1/dm^2*dist(test_data',uk).^2);
%%
%Evaluate
TrPred = function_RBF*w;
TePred = function_RBF_test*w;

TrAcc = zeros(1,1000);
TeAcc = zeros(1,1000);
thr = zeros(1,1000);
TrN = length(train_classlabel);
TeN = length(test_classlabel);
for i = 1:1000
t = (max(TrPred)-min(TrPred)) * (i-1)/1000 + min(TrPred);
thr(i) = t;
TrAcc(i) = (sum(train_classlabel(TrPred<t)==0) +
sum(train_classlabel(TrPred>=t)==1)) / TrN;
TeAcc(i) = (sum(test_classlabel(TePred<t)==0) +
sum(test_classlabel(TePred>=t)==1)) / TeN;
end
plot(thr,TrAcc,'.-',thr,TeAcc,'^-');
legend('tr','te');
hold on;
title('RBFN with K-Mean Clustering','FontSize',20);
xlabel('Thresholds','FontSize',16);
ylabel('Accuracy','FontSize',16);

sumclass1 = zeros(784,1);
for i = 1:size(i1,1)
    sumclass1 = sumclass1+train_data(:,i1(i));
```

```
end
meanclass1 = sumclass1/size(i1,1);

sumclass2 = zeros(784,1);
for i = 1:size(i2,1)
   sumclass2 = sumclass2+train_data(:,i2(i));
end
meanclass2 = sumclass2/size(i2,1);

tmp1=reshape(uk(:,1),28,28);
subplot(2,2,1);
imshow(tmp1);
title('Class1KMeanCentre');

tmp2=reshape(uk(:,2),28,28);
subplot(2,2,2);
imshow(tmp2);
title('Class2KMeanCentre');

tmp3=reshape(meanclass1,28,28);
subplot(2,2,3);
imshow(tmp3);
title('Class1 Mean');

tmp4=reshape(meanclass2,28,28);
subplot(2,2,4);
imshow(tmp4);
title('Class2 Mean');
```