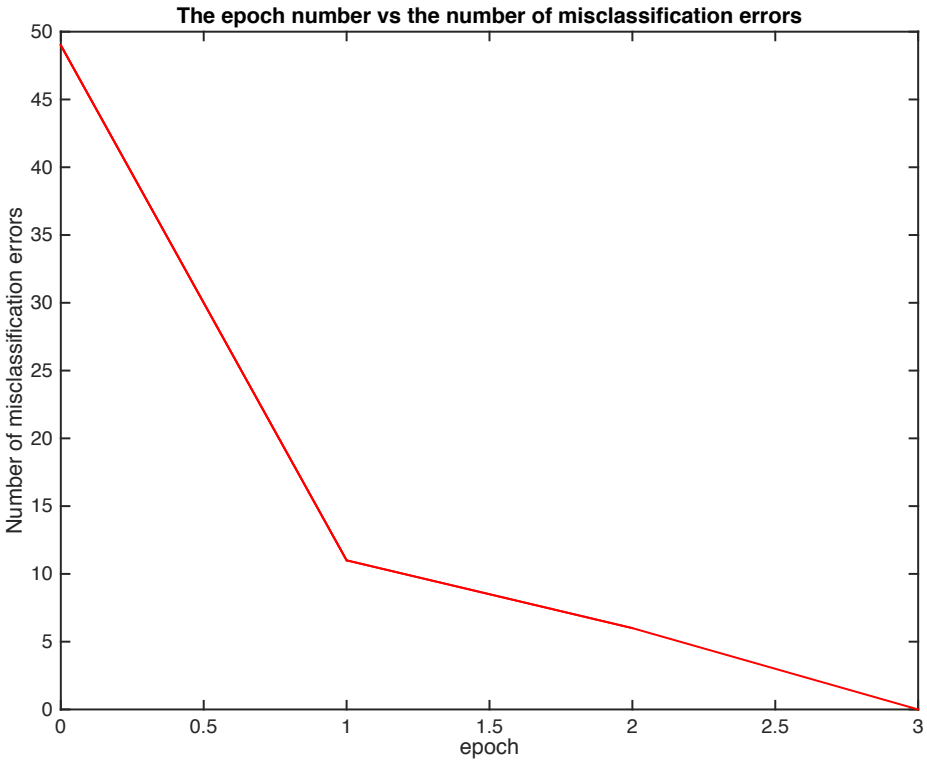
Problem 2

(f)

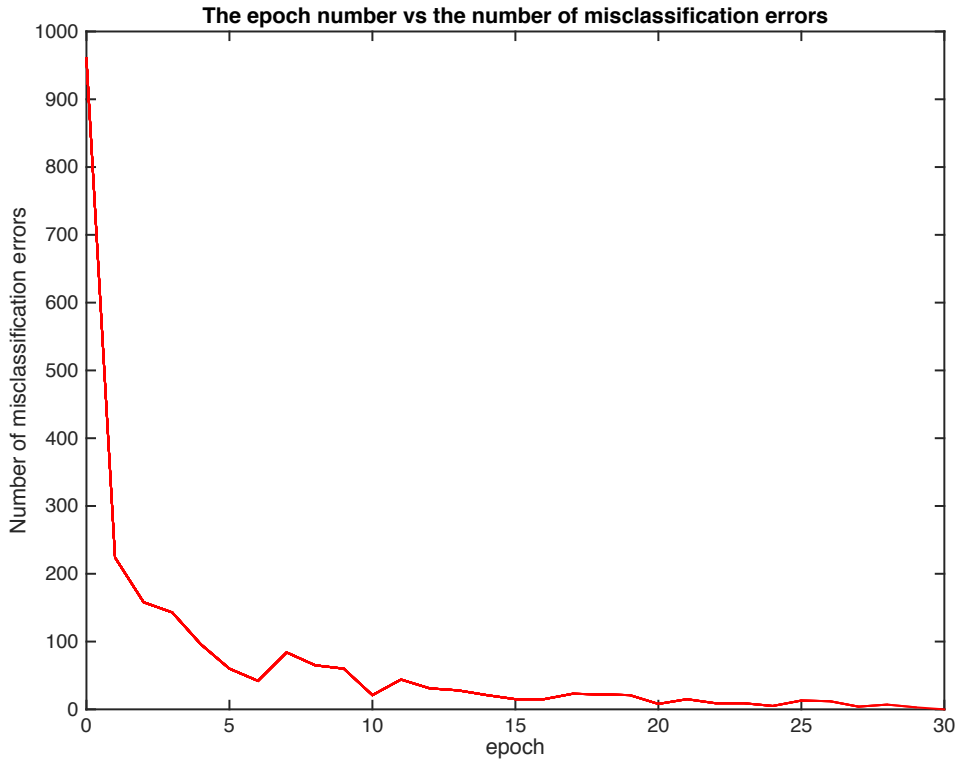


epoch vs errors: (0,49),(1,11),(2,6),(3,0)

The percentage of misclassified test samples is 0.4052.

The percentage of misclassified training samples is 0, but for test samples is 0.4052. They are different. When we use training samples to train the network, we obtain the optimal weights when the system is converged. So the percentage of misclassified training samples is 0. When we use this optimal weights to test the test samples, there may be some misclassification errors. Because the number of training samples is very small, the accuracy is not high, the percentage of misclassified test samples is very big, around 40%.

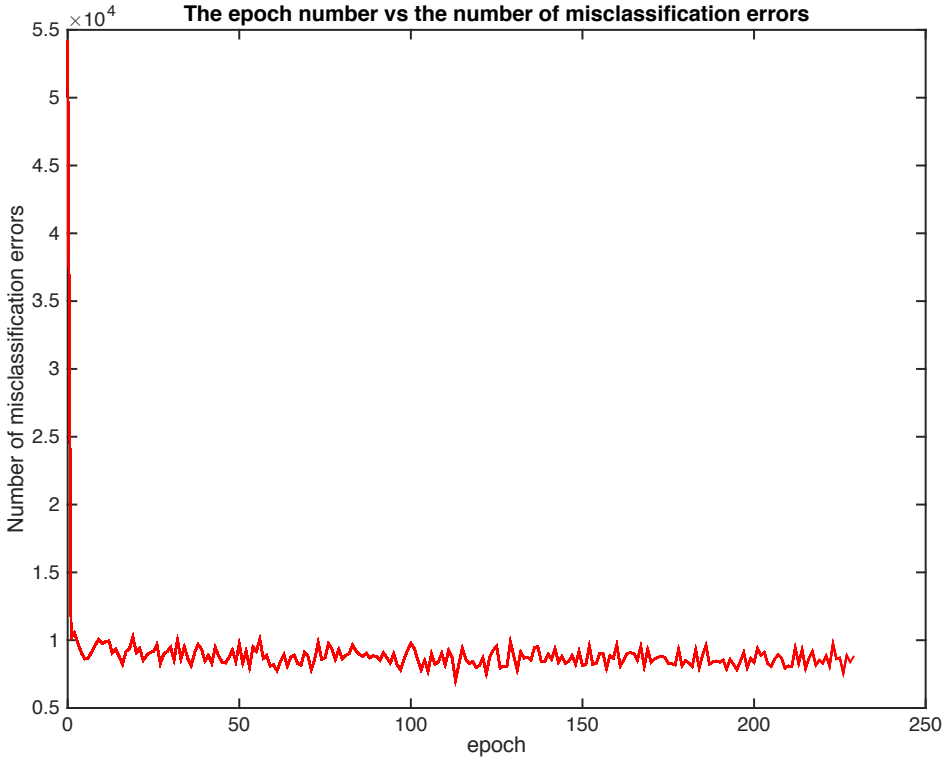
(g)



The percentage of misclassified test samples is 0.1710.

The percentage of misclassified training samples is 0, but for test samples is 0.1710. They are different. When we use training samples to train the network, we obtain the optimal weights when the system is converged. So the percentage of misclassified training samples is 0. When we use this optimal weights to test the test samples, there may be some misclassification errors. Because we increase the number of training samples, the accuracy is improved, but there are still some errors. The percentage of misclassified test samples is decrease obviously, around 17%.

(h)

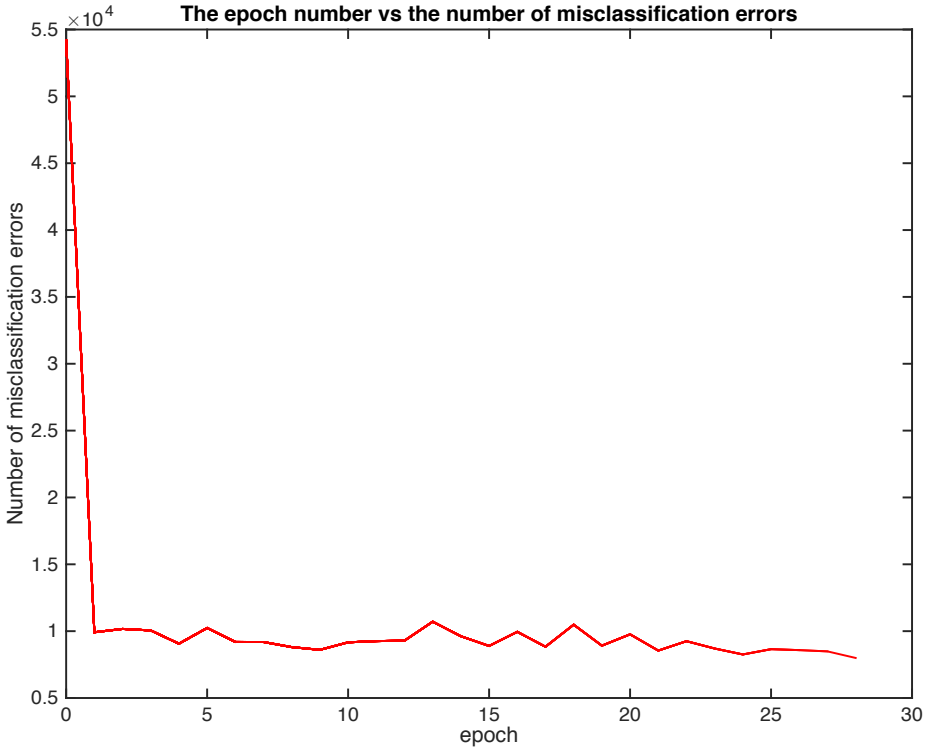


When n=60000, $\varepsilon = 0$, $\eta = 1$, we can know from the the figure above that, as the training samples increase to 60000, the errors are grow large, about 10000. Because the patterns are not linearly separable, the algorithm can not converge, it becomes to oscillate. The misclassification errors can not converge to 0. It oscillates around 10000.

(i)

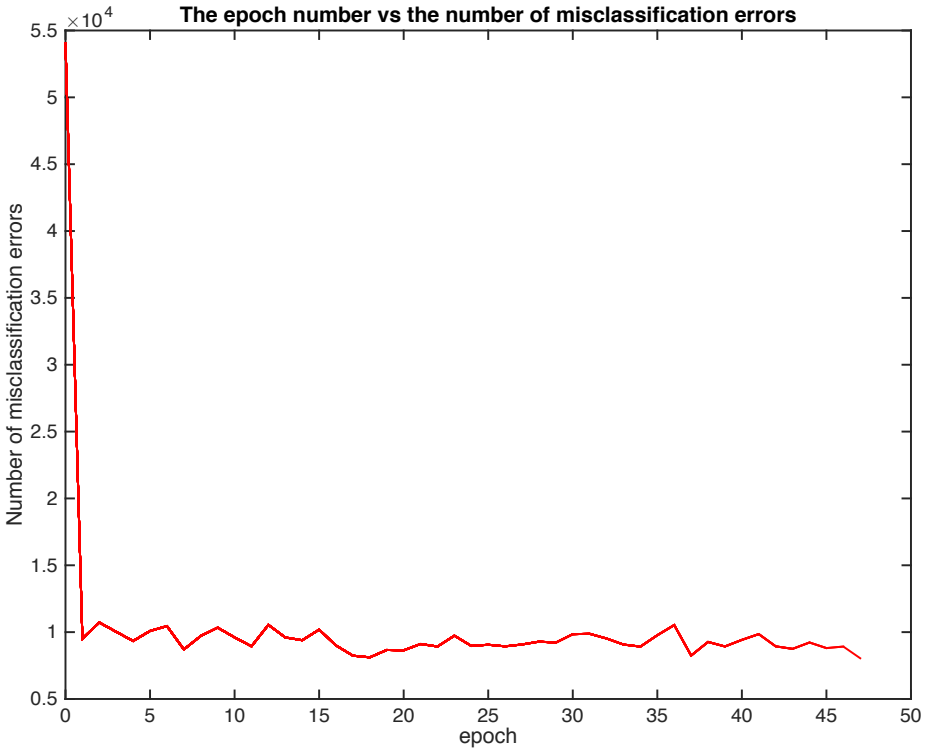
First time:

I choose $\varepsilon = 0.135,\ \eta = 1$. When epoch=28, the algorithm eventually terminate. The percentage of misclassified test samples is 0.1423.



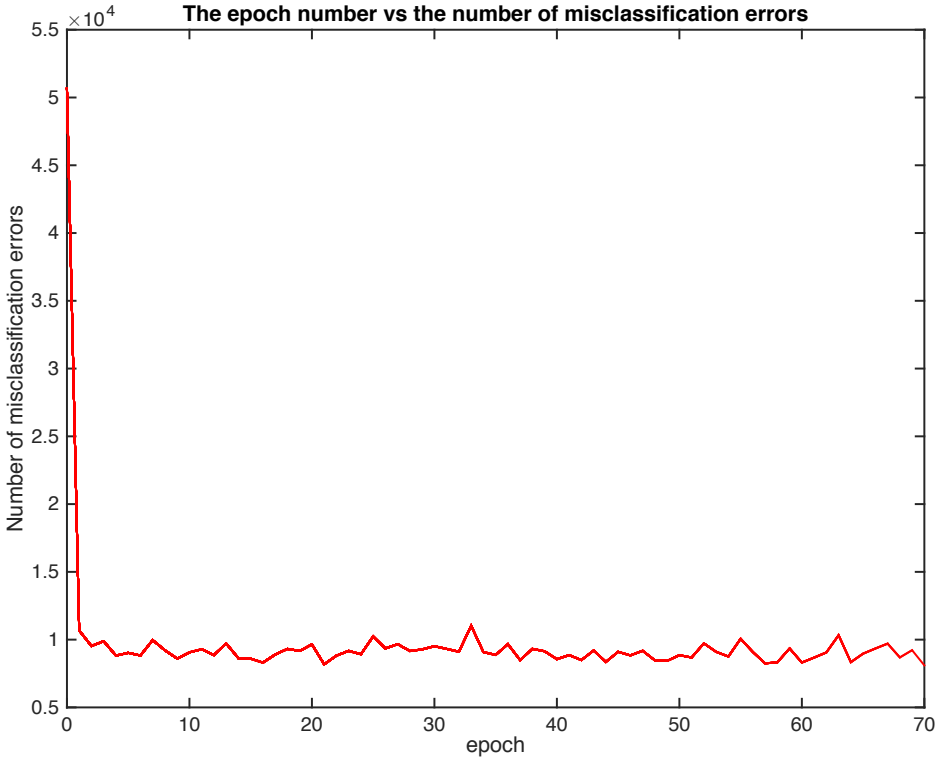
Second time:

I choose $\varepsilon = 0.135,\ \eta = 1$. When epoch=47, the algorithm eventually terminate. The percentage of misclassified test samples is 0.1538.



Third time:

I choose $\varepsilon = 0.135,\ \eta = 1$. When epoch=70, the algorithm eventually terminate. The percentage of misclassified test samples is 0.1630.



Comment:

Because the number of misclassification errors is around 10000, errors/n is around 0.16. I choose ε a little bit smaller than 0.16. When I choose $\varepsilon = 0.135$, the algorithm can eventually terminate in about 1 minute. I use different initial weights, same $\varepsilon = 0.135$ and $\eta = 1$. I do the experiment three times. From figures above, all the percentages of misclassified test samples are little less than the percentage when n=1000. It means more training samples we trained, more accurate the classifier is.

Code

```
function ux=u(v)
for i=1:10
    if v(i,:)>=0
        ux(i,:)=1;
    else
        ux(i,:)=0;
    end
end
end

function labels = loadMNISTLabels(filename)
fp = fopen(filename, 'rb');
assert(fp ~= -1, ['Could not open ', filename, '']);
magic = fread(fp, 1, 'int32', 0, 'ieee-be');
assert(magic == 2049, ['Bad magic number in ', filename, '']);
numLabels = fread(fp, 1, 'int32', 0, 'ieee-be');
labels = fread(fp, inf, 'unsigned char');
assert(size(labels,1) == numLabels, 'Mismatch in label count');
fclose(fp);
end

function images = loadMNISTImages(filename)
fp = fopen(filename, 'rb');
assert(fp ~= -1, ['Could not open ', filename, '']);
magic = fread(fp, 1, 'int32', 0, 'ieee-be');
assert(magic == 2051, ['Bad magic number in ', filename, '']);
numImages = fread(fp, 1, 'int32', 0, 'ieee-be');
numRows = fread(fp, 1, 'int32', 0, 'ieee-be');
numCols = fread(fp, 1, 'int32', 0, 'ieee-be');
images = fread(fp, inf, 'unsigned char');
images = reshape(images, numCols, numRows, numImages);
images = permute(images,[2 1 3]);
fclose(fp);
images = reshape(images, size(images, 1) * size(images, 2),
size(images, 3));
images = double(images)/255;
end

%(d)
trainImages = loadMNISTImages('train-images.idx3-ubyte');
trainLabels = loadMNISTLabels('train-labels.idx1-ubyte');
elta=1;
```

```
n=50;
e=0; %when n=60000,set e=0.135
trainimages=trainImages(:,1:n);
trainlabels=trainLabels(1:n);
W=(1-(-1)).*rand(10,784)+(-1);
epoch=0;
errors=1; %when n=60000,set errors=100000
while ((errors/n)>e)
    errors=0;
    for i=1:n
        x=trainimages(:,i);
        v=W*x;
        vj=max(v);
        for r=1:10
            if v(r,:)==vj
                j=r-1;
            end
        end
        if j==trainlabels(i)
            errors=errors;
        else
            errors=errors+1;
        end
    end
    epoch=epoch+1;
    for i=1:n
        dx=zeros(10,1);
        x=trainimages(:,i);
        label=trainlabels(i);
        dx(label+1,:)=1;
        v=W*x;
        ux=u(v);
        W=W+elta*(dx-ux)*x';
    end
    N(epoch,1)=epoch-1;
    N(epoch,2)=errors;
    N1=N(:,1);
    N2=N(:,2);
    plot(N1,N2,'r')
    hold on
    xlabel('epoch');
    ylabel('Number of misclassification errors');
    title('The epoch number vs the number of misclassification
errors')
```

```
end
%(e)
testImages = loadMNISTImages('t10k-images.idx3-ubyte');
testLabels = loadMNISTLabels('t10k-labels.idx1-ubyte');
errors_test=0;
for i=1:10000
    xtest=testImages(:,i);
    vtest=W*xtest;
    vjtest=max(vtest);
    for r=1:10
        if vtest(r,:)==vjtest
            jtest=r-1;
        end
    end
    if jtest==testLabels(i)
        errors_test=errors_test;
    else
        errors_test=errors_test+1;
    end
end
per=errors_test/10000;
```