Table of Contents

```
% homework 2 ECE 559 - Neural Networks - Fall 2018
% Montagna Marco

% exercise no. 2
close all
clear all
clc

sympref('HeavisideAtOrigin',1); % set 1 at heaviside origin

% open and read the file

% training set image file
train_images = fopen('train-images.idx3-ubyte');
magic_nr = fread(train_images, 1, 'uint', 'ieee-be');
nr_images = fread(train_images, 1, 'uint', 'ieee-be');
nr_row = fread(train_images, 1, 'uint', 'ieee-be');
nr_col = fread(train_images, 1, 'uint', 'ieee-be');
k = nr_col * nr_row;
xtrain = zeros(k, nr_images);

for i = 1:nr_images
    xtrain(:, i) = fread(train_images, k, 'uint8', 'ieee-be');
end
fclose(train_images);

% training set label file
train_labels = fopen('train-labels.idx1-ubyte');
magic_nr = fread(train_labels, 1, 'uint', 'ieee-be');
nr_items = fread(train_labels, 1, 'uint', 'ieee-be');
dtrain = zeros(10, nr_items);
for i = 1:1:nr_items
    l = fread(train_labels, 1, 'uint8', 'ieee-be');
    dtrain(l + 1, i) = 1;
end
fclose(train_labels);

% test set image file
test_images = fopen('t10k-images.idx3-ubyte');
magic_nr = fread(test_images, 1, 'uint', 'ieee-be');
nr_images = fread(test_images, 1, 'uint', 'ieee-be');
nr_row = fread(test_images, 1, 'uint', 'ieee-be');
nr_col = fread(test_images, 1, 'uint', 'ieee-be');
```

```
k = nr_col * nr_row;
xtest = zeros(k, nr_images);

for i = 1:nr_images
    xtest(:, i) = fread(test_images, k, 'uint8', 'ieee-be');
end

fclose(test_images);

% test set label file
test_labels = fopen('t10k-labels.idx1-ubyte');
magic_nr = fread(test_labels, 1, 'uint', 'ieee-be');
nr_items = fread(test_labels, 1, 'uint', 'ieee-be');
for i = 1:1:nr_items
    l = fread(train_labels, 1, 'uint8', 'ieee-be');
    dtest(l + 1, i) = 1;
end
fclose(test_labels);
```

section e

let the user choose the values of this parameters promt1 = 'Enter the value of eta:'; eta = input(promt1); promt2 = 'Enter the value of epsilon:'; epsilon = input(promt2); promt3 = 'Enter the value of n:'; n = input(promt3);

```
eta = 1;
epsilon = 0;
n = 1000;
% inizialize randomly W
W = rand(10, 784); % matrix 10x784 random
% inizialize epoch and misclassification
epoch = 1;
error = zeros(20000, 1); % i decided to put a cap to the epochs vector
 because it's possible that the alorithm does not converge
flag = 0;
% loop where we count the misclassification errors
while flag == 0 && epoch < 20000
    error(epoch) = 0;
    for i = 1:1:n
        v = W * xtrain(:, i); % Calculate the induced local fields
 with the current training sample and weights
        [M,j] = max(v); % find largest number in the vector and its
 position
        [M,I] = max(dtrain(:, i));
        if j ~= I  % if j is different than the input label
            error(epoch) = error(epoch) + 1; % increase
 misclassifictions
        end
    end

    epoch = epoch + 1;

    for i = 1:n
```

```
        W = W + eta*(dtrain(:, i) - heaviside(W * xtrain(:, i)))*
 xtrain(:, i)'; % update weights
    end

    if error(epoch - 1)/n > epsilon
        flag = 0;
    else
        flag = 1;
    end
end

% end of the training part
```

start of test

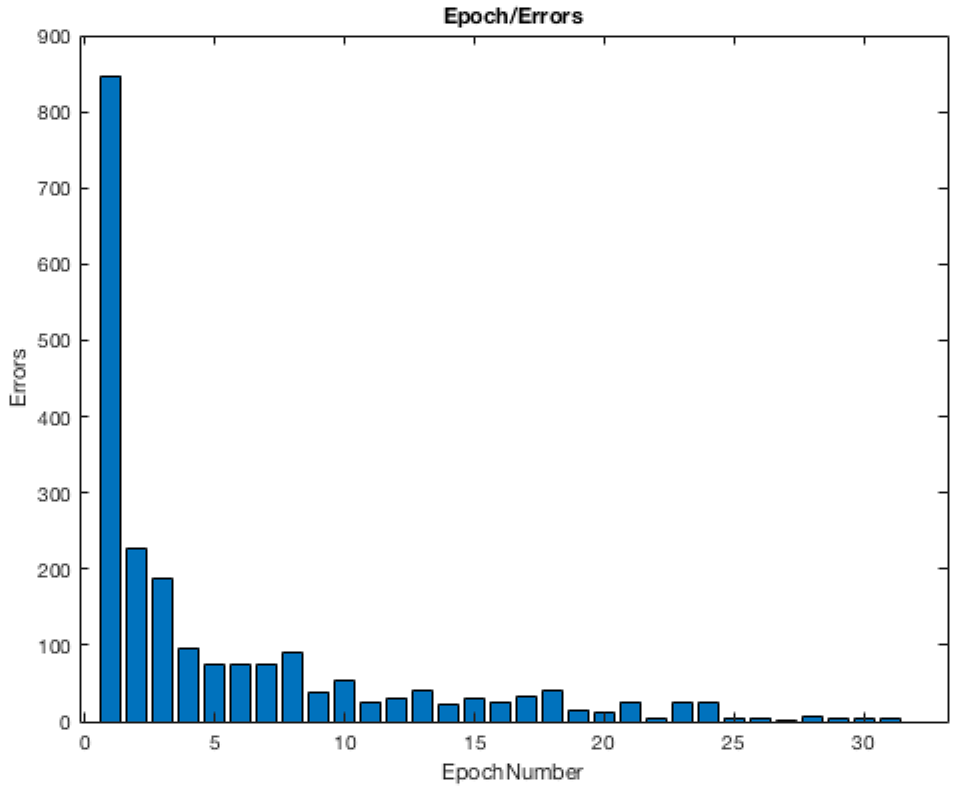
```
misclass = 0; % inizialize errors
for i = 1:1:10000
    v1 = W*xtest(:, i); % Calculate the induced local fields with the
 current test sample and weights
    [M,j] = max(v1); % find largest number in the vector and its
 position
    [M,I] = max(dtest(:, i));
    if I ~= j  % if j is different than the input label
        misclass = misclass + 1; % increase misclassifictions
    end
end
```

results

```
space = (1:1:epoch-1);
figure (1)
bar(space, error(1:epoch-1, 1));
hold on
title('Epoch/Errors')
xlabel('EpochNumber')
ylabel('Errors')

misclassified_test_samples = (misclass/10000)*100;
% text = ['%d %', misclassified_test_samples];
disp('The following numeber is the percentage of misclassified test
 samples (over all 10000 test samples):')
disp(misclassified_test_samples)
```

```
The following numeber is the percentage of misclassified test samples
 (over all 10000 test samples):
   17.7000
```



Published with MATLAB® R2017b