Intelligent Systems

20-EECE-5136

Homework 1

Due Thursday, November 16th, 2017 A. D.

By Tomas Seymour

M05873855

**Problem 1**

System Description:

The neural network system in development is a system that takes input as activation for the first layer of 748 neurons (perceptron). The first layer will receive the greyscale value input for each 748 pixels in a digitalized hand drawn image of an integer 0 to 9. The greyscale values are put through a sigmoid function as soon as they are known, then the resulting values are used as the activation for the input layer. Each greyscale number/activation is given to one of these 748 perceptrons.   
The next layer is a hidden layer of 200 perceptrons of the same kind which are necessary for machine learning behavior. The sum of the products of weights and activations for the previous layer are fed into these perceptrons and again ran through a sigmoid function, note that there is no biasing signal for any perceptron in the network.

At last, the final layer of neurons will be the output or classifier. The vector of 200 activation signals for the hidden layer are multiplied with another set (matrix) and input into a sigmoid function to produce the activations for this final layer. There are 10 total neurons for this layer, one for each classification of a handwritten digit, 0 to 9. The ideal behavior for this layer is a reading of nearly zero activity for all neurons except the one which corresponds to the number given at the input layer.

Error and delta, δ, values are calculated at the very end of this process (the forward pass), then they are used for resetting the weights as part of the backwards pass of the backpropagation algorithm.

The data given to the system is presented in a text file called MNISTnumImages5000.txt which contains as it suggests 5000 different digitalized handwritten numbers. Another text file MNISTnumLabels5000.txt has the ASCII label for each of the handwritten numbers, this is useful for error determination and correction.

The learning rate chosen is approximately 100 input patterns of 748, i.e. 100 handwritten digits, for each Epoch.

The choice of initial weights is made random between negative and positive integer 2.

**Problem 2**

System Description:

Using the same data, create a network that does a reverse process as in the first problem. It will be a feed forward network of 784 inputs and 784 outputs, with one hidden layer of 200 neurons again. The task will be to present as the output the same activations as was given in the input.   
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
This is the incomplete code for problem 1.

filename = 'MNISTnumImages5000.txt';

A = importdata(filename); % loading pixels. Each row of the matrix represents one number

A = 1./(1 + exp(-A)); % put values in sigmoid function

filename = 'MNISTnumLabels5000.txt';

L = importdata(filename); % loading labels. Each row of the matrix is the number label for each row of matrix A.

alpha = 1;

eta = 0.09;

weights1 = randi([-2.0 2.0],200,784); % randomized weights between first and second layers

weights2 = randi([-2.0 2.0],10,200); % randomized weights between second and final layers

Y1 = [];

Y2 = [];

y1 = [];

y2 = [];

error = [];

delta\_forweights1 = [];

delta\_forweights2 = [];

C = [];

% for determining the random sample of 4000, we shuffle a vector of

% incrementing integrers 1 through 5000, then suffle them through random

% permutation

vector = 1:5000;

vector = vector(randperm(length(vector))).';

%testset = A(vector(1:4000),:);

dW2 = 0;

dW1 = 0;

%FORWARD PASS FOR ONE EPOCH

for i = 1:100

activation1 = A(i,:).'; %transposed for easy multiplication with weights

y1(i,:) = (weights1\*activation1).';

Y1(i,:) = 1./(1 + exp(-(weights1\*activation1).')); % holds activations for the next layer of 200 neurons (784 -> 200), for 100 different inputs of 784

activation2 = Y1(i,:).';

y2(i,:) = (weights2\*activation2).';

Y2(i,:) = 1./(1 + exp(-(weights2\*activation2).')); % should hold the final layer of neurons which bright up which neuron most represents the given input data of 784 pixels

% the ideal output of the network would be something like all low

% activations except for the one neuron which classifies the

% handwritten number

%next, determine costs, error, and delta

desiredoutput = zeros(10,1); %begin with vector of zero

desiredoutput(L(i)+1) = 1; %must add 1 for MATLAB vector syntax.

sum = 0;

for k = 1:10

cost = 0;

e = 0;

e = desiredoutput(k) - Y2(i,k); %subtract from the desired output the actual output

cost = e^2;

error(i,k) = e;

delta\_forweights2(i,k) = Y2(i,k) \* (1 - Y2(i,k)) \* (desiredoutput(k) - Y2(i,k)); %derivative of sigmoid function

sum = sum + cost;

sum2 = 0;

%change weights between the second and last layers according to

%delta\_forweights2. Also, compute values for later computation of

%other delta.

for h = 1:200

dW2 = eta\*delta\_forweights2(i,k)\*y1(i,h);

weights2(k,h) = weights2(k,h) + dW2;

end

% incomplete insofar as calculation for values necessary for learning to

% happen in the layer between the first and second layer

end

C(i,:) = sum;

end