Neural Network Assignment

To get our feet a little wet with neural networks, we are going to walk through portions of the process.

You are given a data set with 5000 handwritten digits and their corresponding labels. Each training example is a 20 pixel by 20 pixel grayscale image of the digit. Each pixel is represented by a number indicating the grayscale intensity at that location. Thus, your neural network will have 400 inputs.

Your network will have 3 layers: an input layer with 400 inputs, output layer with 10 outputs (corresponding to the ten digits), and a hidden layer with 25 units. You will add bias units at the first and second layers. Thus between the first and second layers there are 401*25 = 10,025 weights. Between the second and third layers there are 260 weights. The total number of weights is therefore 10, 285. You are also provided with the set of weights to use for this assignment.

1) Implement a neuron unit: Write a function that takes as input the activations from the previous layer and the input weights for that that layer, and returns the activation value for that neuron.

2) Write a function that takes as input a data point $\mathbf{x}^{(i)}$ of dimension 400 and the weights and outputs the activation values $\mathbf{a}^{(2)}$ at layer 2.

3) Write a function that takes as input the activation values $\mathbf{a}^{(2)}$ at layer 2 and the weights and produces the activation values $\mathbf{a}^{(3)}$ at layer 3.

4) Write a function that uses the above two functions to do the following: It takes input $\mathbf{x}^{(i)}$ of dimension 400 and the weights and classifies the corresponding image as a number between 0 and 9. Thus the output of the function is a number between 0 and 9.

5) Use your above function to classify all 5000 digits. What is the error rate?

6) Write down the cost function for this neural network. Write a program to evaluate the cost with the given weights and with the 5000 test examples.

7) In order to find the optimal weights, we need to take the partial derivatives. For this we use back propagation. Provide pseudo-code showing how we calculate the partial derivatives using back propagation. (No actual coding is needed here.) Be fully detailed with the equations.