Assignment 4: Introduction to Neural Networks

To get our feet wet with neural networks, we are going to walk through portions of the process. You are to do to this without using numpy.

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 consist of an input layer with 400 inputs, a hidden layer with 25 units, and a softmax layer producing a probability distribution over the ten digits.

You will add bias to the inputs to the hidden layer and to the inputs of the output layer. Thus between the input layer and hidden layer there are 401*25 = 10,025 weights. Between the hidden layer and output layer there are 260 weights. The total number of weights is therefore 10,285. Each hidden unit uses a logistic activation function. You are also provided with the set of weights to use for this assignment.

1) Implement a unit in the hidden layer: Write a function that takes as input the inputs **x** and the weights **w** and bias b for that unit, and outputs the activation value a for that unit.

2) Implement the softmax function, which takes an input vector **z** of dimension 10 and outputs and output vector **a** of dimension 10.

3) Implement a function that takes as input the outputs of the hidden layer $\mathbf{a}^{h}$ (dimension 25), the weight matrix **w** between the hidden and output layers (dimension 10x25), and a bias vector **b** (dimension 10), and outputs the output **a** of the softmax output layer (dimension 10). Note this function should call the function developed in part (2).

4) Using your functions in (1) and (3), write a function that takes input $\mathbf{x}^{(i)}$ of dimension 400 and the 10,025 weights and outputs the 10 output values (after the softmax) of the neural network as a list.

5) Using the function in (4), write a function that classifies an image as a number between 0 and 9. Thus the output of the function is an integer between 0 and 9.

6) Use the function in (5) to classify all 5000 digits in the data set. What is the error rate? Time how long it takes to classify all 5000 digits.

7) Write down the cross-entropy loss function for this neural network. Write a program to evaluate the loss function with the given weights and with the 5000 test examples.

8) Repeat (1)-(6) using numpy. Compare the amount of time required to do part (6) with and without numpy.