Assignment 4

SUBMIT ASSIGNMENT

Due Feb 14 by 11:59pm **Points** 16 **Submitting** a file upload

Recognition of Handwritten Digits

In this assignment, we will use neural network to recognize handwritten digit. We are given a set of handwritten digit images which ranges from 1 to 5. Each image has 20 x 20 pixels and we have 2500 such images. If an image has 3 x 3 pixels and if the pixel intensity matrix of the pixels is given by $A = \begin{bmatrix} 123 & 50 & 240 \\ 10 & 99 & 129 \\ 167 & 157 & 67 \end{bmatrix}$, then this image will be stored as a single row as $A = [123 \;\; 50 \;\; 240 \;\; 10 \;\; 99 \;\; 129 \;\; 167 \;\; 157 \;\; 67]$.

The images has already been converted into a matrix such that, each row is a sample image and the columns are the pixel intensity values of each pixel. The pixel intensities have been normalized to This matrix is given by 'image_data.txt'. Each image also has a label from 1 to 5 corresponding to the actual digit. This label matrix is given by 'labels.txt'.

We will use a three layer Neural Network with 400 nodes in input layer, 20 nodes in the hidden layer and 5 nodes in the output layer to predict handwritten digits. We will take an image as input to the Neural Network and predict the most probable digit. The 400 input nodes corresponds to 20 x 20 pixels of an image and 5 output nodes corresponds to 5 different labels. Note that we have excluded the extra bias term +1 in the input and hidden layer. You should take this into consideration while writing your code.

Let s_i represents the activation of layer i, W^i represents the weighing matrix from layer i to layer $i+1$ and $\sigma(z)$ be the sigmoid activation function. You should implement the feedforward propagation steps to predict the digit. The weighing matrix W^1 and W^2 has already been trained by us, and is given by 'W1.txt' and 'W2.txt'. Note that Backpropagation algorithm can be used to find the optimal values of W^1 and W^2 (not a part of this exercise).

The output layer gives the probabilistic prediction of the digit from 1 to 5. The prediction from the neural network is the label that has the largest probability (note that this way of recognizing the image is slightly different from what we did in the lectures, since it works well in practice). For example, if the output is [0.1, 0.33, 0.6, 0.45, 0.47], then we label the image as 3. On the other hand, if the output is [0.6, 0.1 0.35, 0.23, 0.14], the we label the image as 1, and so on.

Answer the following questions:

a) Write a code to import image_data.txt with separator ";", labels, W^1 and W^2 into the workspace and print the dimensions of each of the matrices. (2 points)

b) A sigmoid function is given by $\sigma(z) = \frac{1}{1+e^{-z}}$. Write a function that takes a matrix as input and gives out a matrix that consists of sigmoid of each of the element in the input matrix. Print the sigmoid of the matrix $A = \begin{bmatrix} 0.8558 & 0.5236 \\ 0.6708 & 0.2988 \end{bmatrix}$. (3 points)

c) Implement feedforward propagation steps with sigmoid activation function to predict the labels of 2500 images. The steps for forward propagation are:

Step 1: Pick a row from the matrix image_data.txt and transpose it. Let it be s_1. Add an additional element '1' at the beginning of the array s_1. Let the new array be s_{1b}. The dimension of s_{1b} should be 401 x 1.

Step 2 : Let the activation of second layer be s_2. Then, $s_2 = \sigma\left(W^1 * s_{1b}\right)$.

Step 3: Add an additional element '1' at the beginning of the array s_2. Let the new array be s_{2b}. The dimension of s_{2b} should be 21 x 1.

Step 4: Let the activation of second layer be s_3. Then, $s_3 = \sigma\left(W^2 * s_{2b}\right)$.

Step 5: Label the image corresponding to the largest value in s_3.

Note that, the above steps has to be repeated for every single image which corresponds to a row in image_data. If you are using a vectorized method, then it can be done in a single iteration.

Write a code to count and print the number of estimated images under each label. (5 points)

d) Compare your result with the actual labels given in 'labels.txt' and compute the prediction accuracy in percentage. The prediction accuracy is given by

$$Percentage\ correct = \frac{Number\ of\ labels\ correctly\ predicted}{Total\ number\ of\ images} \times 100$$

Print the prediction accuracy. You should see a prediction accuracy of more than 93%. (3 points)

e) Generate a random 20x20 image using runif (see http://www.cookbook-r.com/Numbers/Generating_random_numbers/ (http://www.cookbook-r.com/Numbers/Generating_random_numbers/)). Recall that the pixel intensity has to be between 0 to 1.

Plot the image. Give this image as input to the neural network (follow steps in Question c above) and print the output of the last layer of the neural network. Find the label corresponding to the largest value in the output of the last layer. Now, answer the following questions (2 marks):

i) Why does the neural network classify a completely random image, which does not look like any digit?

ii) Now imagine that an autonomous car is using its camera to classify cars, pedestrians etc. using neural network. There was a snowstorm and the riders of the car did not clean up the snow properly. The camera's input is disturbed by the snow flakes on the camera's lens (one could also have bug splatter on the camera's lens that disturbs its input). Would you trust the classification of the objects in the images taken by the camera? Your justification should use your intuition along with some geometric understanding of how neural network classification works.

f) How would you get around the issue stated in i) above? [You are allowed to augment the output, training set, etc. if you want] (1 marks)