Perceptron Assignment – Detect 8

This is a group assignment.

You will be provided with 217(87 for 8 and 130 for other digits) text files with black and white images represented as text. The images contain digits and their corresponding labels. The objective of the assignment is to program a single perceptron to train on this dataset to detect the digit **8**.

Pseudocode for a simple perceptron learning model is given below:

```
#TRAINING
weights[ HEIGHT * WIDTH + 1] = 0
#The learning rate is a parameter.
LEARNING_RATE = 0.1
#The bias is a parameter.
#MAX_ITERATIONS is a parameter.
epoch = 0
while epoch++ < MAX_ITERATIONS
    for i=0; i < TRAINING_SIZE; ++i
        img_array = read_image_file(i)
        label = read_label(i)

        #multiply weights with pixel values.
        #for all p, sum = sum + weight[p]*img_array[p]
        #If sum > 0, output = +1, ie. +ve detection
        #else output = -1, ie. -ve detection.
        output = sum_of_weights(img_array)
        #We are trying to train for 8
        if label == 8
            actual_output = 1
        else
            actual_output = -1

        ERROR = actual_output - output
        #For every weight index, p,
        #weight[p] = weight[p] + LEARNING_RATE * img_array[p] * ERROR
        update_weights(img_array, ERROR)



#EVALUATION
FILENAME = user_input()
test_img = read_image_file(FILENAME)
output = sum_of_weights(test_img)
```

The parameters can be tuned to improve training. You can also additionally keep track of the number of misclassifications during iterations of the epoch to have an idea about when to stop training. A misclassification is when your weights classify an image that is not 8 as 8 or vice versa. The error function provided in the pseudocode is a step function between -1 and +1, centered at 0.

You need to try to maximize correct classifications and minimize misclassifications. Note that perfect performance might not be obtained, ie. 100% perfect classification.

Save the weights that you decide at the end of training. You should have a way to enter the filename of a test image and evaluate your weights.

Submission and evaluation:

Submit your code and write a short 1-2 page report about your choice of parameters and your observations on how they affected your training. How did the accuracy vary over iterations?

Have your code running on the laptop of one of your team members and demonstrate your trained model to the TA during the last class of the session. He will present you with new images that need to be classified as **8 or not 8.**