HW2

The main aim of the assignment is to make you familiar with the linear classifiers, specifically with the perceptron algorithm. Please solve the problems individually and cheating will be harshly punished.

To generate the **inputs** you should read images from train folder and convert the images into vector format. Resize all image to 256x256x3 form. The label of each image should be saved in a target vector, called **t**. For **cannon** class, the class label is **0**. For **cellphone** class, the label is **1**. You should read test image and predict the class label for each one.
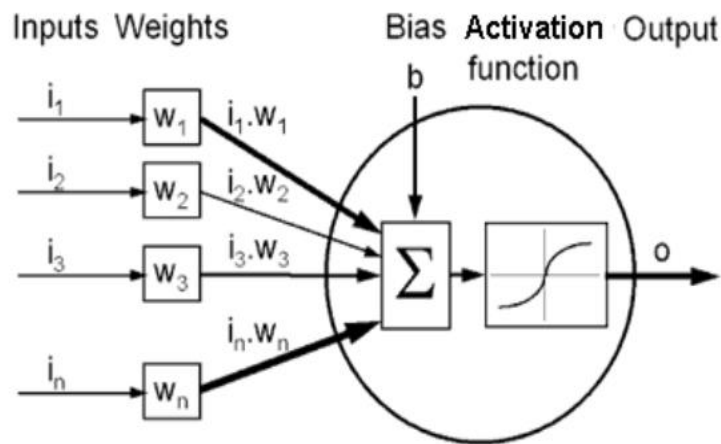

Fig. 1. General demonstration of perceptron algorithm [1].

Fig. 1 shows the general demonstration of perceptron algorithm. In the last head of activation, we will use the sigmoid function.

We will use the gradient descent algorithm in case of training our perceptron algorithm.

perceptron(*inputs*, w_{init}, η)

Each training sample is a pair of the form $\langle \vec{x}, t \rangle$ where $\vec{x}$ is the vector of input values, and t is target output value. η is *0.001*.

➢ Initialize each weight, w_i to some small value.

➢ Until the all samples classified correctly or iteration condition met. Do

   ➢ Initialize each $\triangle w_i$ (*called gradient*) to zero

   ➢ For each training samples, Do

      ➢ Input the instance $\vec{x}$ to the unit and compute the output o

   $$g(\mathrm{x}) = \frac{1}{1+e^{-x}}$$

   ➢ For each linear unit weight w_i Do

   $$\triangle w_i = \eta (t - y) g'(y) x_i,$$
   $$(t - y) \ \textit{referes to error}$$
   $$g'(y) \ \textit{refers to derivative of sigmoid function}$$
   $$g'(y) = g(y)(1 - g(y))$$

   ➢ For each linear unit weight w_i, Do

   $$w_i = w_i + \triangle w_i$$

[1] http://aass.oru.se/~lilien/ml/seminars/2007_02_01b-Janecek-Perceptron.pdf

Step 1:

Implement a `trainPerceptron(inputs, t, weights, rho, iterNo):` function in PYTHON, in order to train the linear classifer, so called perceptron algorithm.

inputs **:** Feature vectors belonging to classes (0 and 1)

Note that, for bias, you should add 1 to input vector.

If the input vector size is n, then after adding bias, the input vector size will be n+1.

Don't forget to shuffle your inputs before the training procedure. Recall from the class, we have touched about shuffle of training data. You can use sklearn library with following snippet code.

```
from sklearn.utils import shuffle
X_train, Y_train = shuffle(X_train, Y_train)
```

t : labels of classes (0 and 1)

weights: Initial weights for the linear discriminant function. You should initial weights random. The vector dimension of weights and input vector is same. Both of them are n+1.

rho : learning rate is 0.001

iterNo: refers to number of iterations in perceptron is 1000

After the training procedure completed, you should save weights for testing purpose. In case of testing stage load the weights vector.

Once the

Step 2:

To implement your *testPerceptron(sample_test, weights)* function, you should return predicted value.

In this stage, you will test your perceptron algorithm

First of all, load weights which had been trained during the training stage.

Secondly, read a test image from test folder and send the feed forward process of perceptron algorithm.

Finally, the perceptron will give an output about label of test sample.

- You will graded over 100% if you have implemented the given train and test functions. Otherwise, you will be graded over 75%.
- Also, you have to add comment to your codes.
- In training part, you have to show the feed-forward process and feed-backward process by inserting comments.

Submit the Assignment

Please write all answers on this document and forward this document as pdf format.
Ex: No_Name_Surname_HW#.pdf

Hint

You can look the implementation perceptron on notes of the course.