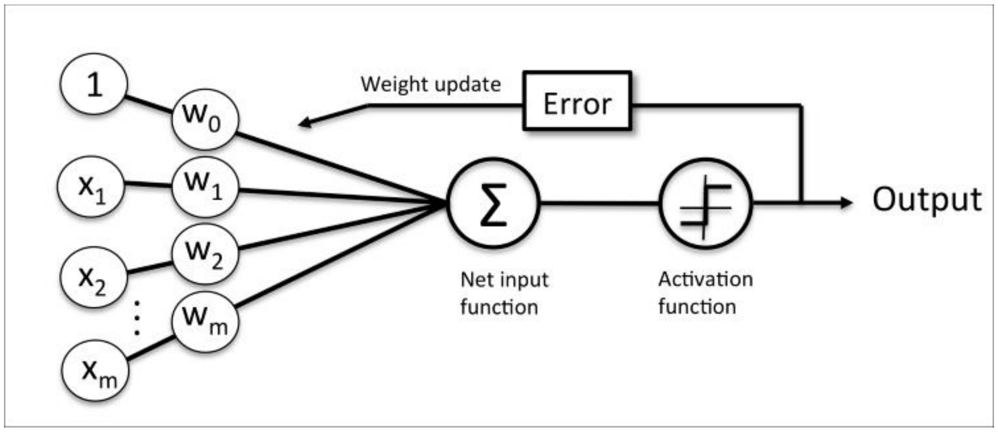
EAST TENNESSEE STATE UNIVERSITY

Project 5: Supervised Learning

Creating and Training a Perceptron

Recall the basic structure of a Perceptron.



Part 1: Perceptron Class

A Perceptron Class Framework is provided in **project5_perceptron.py**. You must fill in areas marked TODO. A basic outline of this class is:

Class Members:	`learning_rate` : float (between 0.0 and 1.0) `iterations` : int (The number of passes over the data) `w_` : 1d-array of floats (The weights for each input) `errors_` : list (Contains the number of miscalculations in every pass).
`train(self, X, y)`	TODO: Trains the perceptron. • X is a numpy array of shape [n_samples, n_features] (where n_samples is the number of samples and n_features is the number of features in the data set) • y is the list of target values for each `X[i]` Algorithm: (1) Initialize weights to a small random number between 0 and 1. (2) For the number of iterations, (a) For each item in data set X: (i) Predict the answer (ii) Calculate the error based on the answer and the known `y[i]` (iii) Update the bias weight and input weights if the predicted answer is erroneous (use the Perceptron Learning Rule: $W(i)=W(i)+\alpha[T-A]*P(i)$) (b) Append the error to self.errors_ so it can be plotted later
`net_input(self, X)`	TODO: Returns the net input for the input vector X
`activate(self, X)`	TODO: Returns 1 if net_input(X) >= 0.0 or -1 otherwise. Implements the step function and returns the predicted class.

Running the Program

The **project5.py** program is intended to run on the command line as follows.

```
python project5.py [datafile] [resultfile]
```

Test Data Sets

Two test data sets are provided for the Perceptron learning algorithm. The first, in `iris2_X.npy` and `iris2_y.npy`, is the Iris Data Set which ignores the third flower. The second, in 3d_X.npy and 3d_y.npy, which is a set of clustered 3-dimensional points where k=2.

Analysis

1. Iris Data
   a. Train your perceptron on the Iris Data set. Following the training of your Perceptron, paste a print of the Perceptron's w_ (weights) list. Then, record the learning rate and iterations. Paste a screenshot of your learning errors over time.
   Then, conduct the same experiments you conducted in previous labs. Note that, this time, the Perceptron will predict either the setosa or versicolor flower. Returning -1 indicates the setosa flower, while 1 indicates versicolor. Place an X in the column for the flower predicted.

Weights:	
Learning Rate:	
Iterations:	
<<PASTE PLOT OF ERRORS HERE>>	

	Setosa	Versicolor
`[[7.2, 3.1, 4.8, 1.5]]`		
`[[3.6, 2.8, 1.8, 0.5]]`		
`[[5.5, 3.8, 2.8, 1.2]]`		
`[[7.8, 1.9, 5.9, 2.1]]`		
`[[18.2, 9.1, 15.4, 5.5]]`		
`[[0.5, 0.25, 0.9, 0.3]]`		

   b. Explain why the Perceptron should be able to predict the flowers for this particular data set.

2. 3-dimensional Data.
   a. I've also included a three-dimensional data set. You can find the data set in 3d_X.npy (the 3-dimensional points) and 3d_y.npy (the cluster for each point). I have also included commented-out code that will plot the 3d array in **project5.py**.

   SET THE LEARNING RATE to 0.01 AND THE ITERATIONS to 10. Paste a print of the Perceptron's w_ list, and the plot of errors.

Weights:	
<<PASTE PLOT OF ERRORS HERE>>	

   b. Next, perform the following predictions and place an X in the appropriate column. Every point would NOT be in the same cluster. Point [60,60,60] is only slightly skewed toward the same cluster as point [100,100,100].

	Cluster -1	Cluster 1
`[0, 0, 0]`		
`[60, 60, 60]`		
`[100, 100, 100]`		

   c. Now, **adjust** the learning rate and iterations for your Perceptron. Recheck the predictions. When the predictions make more sense, report the weights found and the clusters below. Also, note the learning rate and iterations that produced the result.

Weights:	
Learning Rate:	
Iterations:	

	Cluster -1	Cluster 1
`[0, 0, 0]`		
`[70, 70, 70]`		
`[100, 100, 100]`		

d. Finally, explain why the learning rate and number of iterations affected the results.

Submission and Due Date

Place all files used for this project into a folder named lastname_project5 (of course, you should use your own last name). ZIP the folder and submit it to the dropbox **at or before Sunday, April 21, 2019 at 11:59pm**. The rubric follows.

Rubric

	Points Possible	**Points Gained**	**Comments**
Part 1: Perceptron Class			
train	20		
net_input	10		
activate	10		
Testing	10		
Part 2: Analysis Writeup			
1.a. Iris Data Set	10		
1.b. Explanation of Iris predictions	5		
2.a. 3d Data Set	10		
2.b. 3d predictions	10		
2.c. Updated 3d predictions	10		
2.d. Explanation of rate changes	5		