SRIP Project 1 Documentation
Linear Perceptron Learning

Task allotted

1. In virtual-labs repository, pattern-recognition-iiith lab, the task was to resolve Issue No: 240
2. Issue No: 240 was to Convert following Linear Perceptron learning experiment to JavaScript. Link to the experiment:- http://cse20-iiith.vlabs.ac.in/exp3/Experiment.html?domain=Computer%20Science&lab=Pattern%20Recognition%20Lab

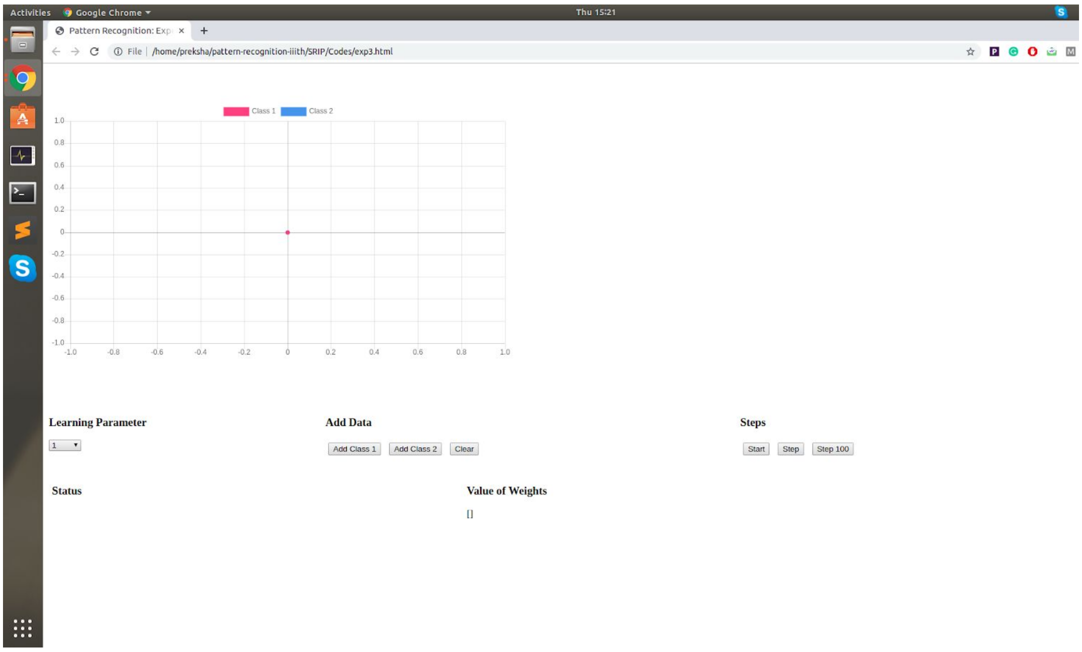
Experiment Explanation

- Perceptron: In machine learning, the perceptron is an algorithm for supervised learning of binary classifiers. A binary classifier is a function which can decide whether or not an input, represented by a vector of numbers, belongs to some specific class. It is a type of linear classifier, i.e. a classification algorithm that makes its predictions based on a linear predictor function combining a set of weights with the feature vector.
- Given multiple points belonging to 2 classes(Class 1 and Class 2), classify those points into the 2 classes with the help of the linear perceptron algorithm. Draw the linear perceptron line which divides the 2 classes.

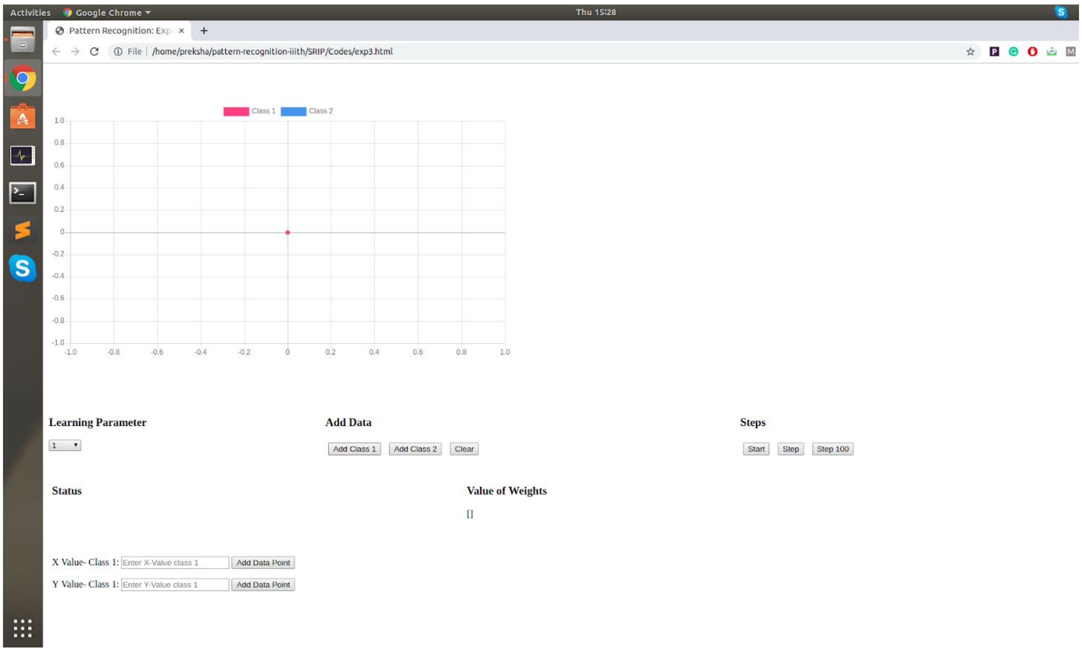
How to Run the Experiment

1. My forked repository(https://github.com/prekshap24/pattern-recognition-iiith) contains a folder named “SRIP”.
2. SRIP folder contains a folder named as Project-1 Issue Number 240 which contains a folder named as Codes and Libraries. Codes contains all the files containing code for the experiment written in JavaScript, HTML, CSS. Libraries contain JavaScript libraries used in the codes.

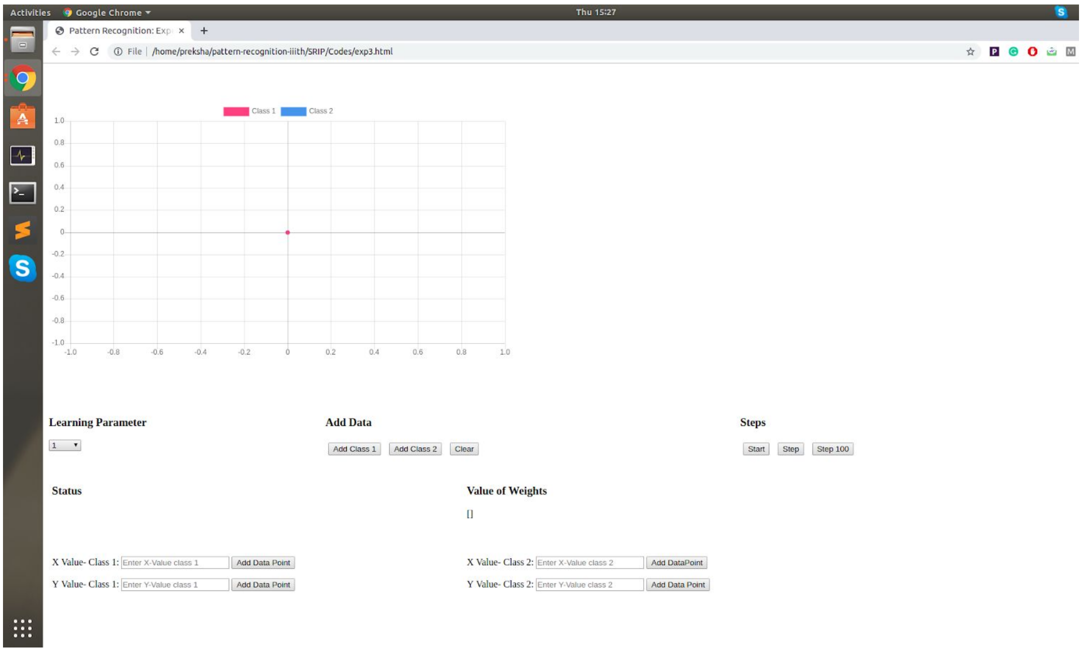
3. The Codes folder contains 3 files. To run the experiment, simply run the exp3.html file by clicking on it.
4. The experiment will open in the browser.



5. To run the experiment of perceptron, add the 2 classes. After clicking on the ‘Add Class 1’ button, this will appear. Add the x-axis and y-axis of the points you need to add for the class 1.



6. Similarly add data points for class 2 after clicking the ‘Add Class 2’ button and then adding the x-axis point and y-axis point.



Learning Parameter: 1

Value of Weights: []

X Value- Class 1: Enter X-Value class 1

Y Value- Class 1: Enter Y-Value class 1

X Value- Class 2: Enter X-Value class 2

Y Value- Class 2: Enter Y-Value class 2

7. Select a learning parameter value from the drop down list under ‘Learning Parameter’.
8. After clicking on the ‘Start’ button, the perceptron algorithm will run for the given data points of the 2 classes and the perceptron line will be plotted on a new graph created.
9. The value of the final weights will be displayed under ‘Value of Weights’.
10. If you click on the ‘Step’ and ‘Step 100’ button, it will display the number on iterations taken to complete the algorithm.

Formulas used in the Experiment

- Final training dataset made would be of this type

```
dataArray = [ [x1   y1   0],
              [x1   y1   0],
              [x2   y2   1],
              [x2   y2   1],
              [x2   y2   1] ]
```

- Initially the weights vector is assigned [0, 0].

- There are two inputs values (x1 and y1) and 2 weight values (w1 and w2). The activation equation used is

  activation = (w1 * x1 + w2 * y1) + bias

- If activation greater than or equal to 0, then function returns 1, else returns 0
- There are 2 loops running,
  - Loop over each row in the training data for each iteration
  - Loop over each weight and update it for a row for each iteration
- Then the error(error = 0/(or)1 - prediction) is calculated. Until the errors becomes 0, weights are modified using the following formula

  Weights [j] = weights [j] + (learning Parameter * error * dataArray [c][j])

- Now the final weights are displayed under 'Value of Weights'. The final perceptron line is drawn using the following 2 points:

  x1 = (- bias / weights [0]) x2 = 0

  y1 = 0 y2 = (- bias / weights [1])

  and the perceptron equation is:

  y = (- (b / w2) / (b / w0)) * x + (- b / w1)