SRIP Project Documentation
Linear Perceptron Learning

Task allotted

1. In virtual-labs repository, pattern-recognition-iiith lab
2. 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

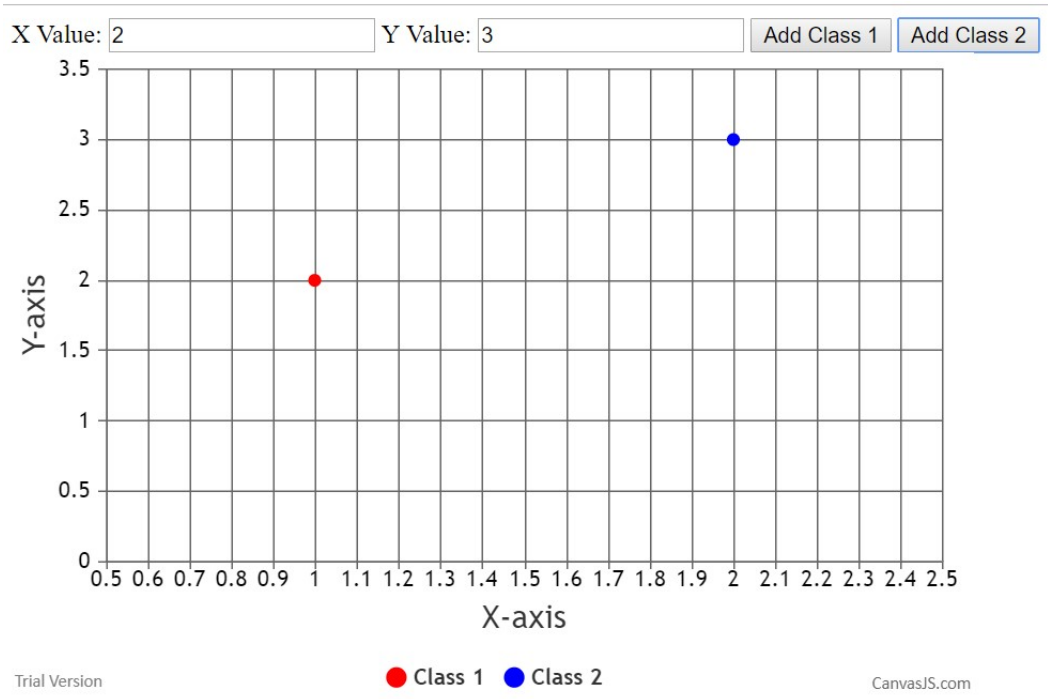
- Perceptron: In machine learning, the perceptron is an algorithm for supervised learning of binary classifiers. A binary classifier is a function which can decide whether or not an input, represented by a vector of numbers, belongs to some specific class. It is a type of linear classifier, i.e. a classification 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/pavansaladi/pattern-recognition-iiith)
   contains a folder named “SRIP”.

2. SRIP folder contains folder named as Linear Perceptron and it contains all the files containing code for the experiment written in JavaScript, HTML, CSS.
3. To run the experiment, simply run the text.html file by clicking on it.
4. The experiment will open in the browser.

5. To run the experiment of perceptron, add x,y values for class1 and clickon ‘Add Class1’ button and same for class2. After clicking on the ‘Add Class 1’ and ‘Add Class2’ button, this will appear



Steps

Start Step Step100

Status **Value of Weights**

6. 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.
7. The value of the final weights will be displayed under 'Value of Weights'.
8. 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 data = [x y 0]

- Initially the weights vector(data1) is assigned [0, 0,0].
- There are two inputs values (x and y) and 2 weight values (w1 and w2).

  The activation equation used is

  activation = (w1 * x + w2 * y) + 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

  data1[i]=data1[i]+data[iterations-1][i]* learning_rate*res;

- Now the final weights are displayed under 'Value of Weights'.

  The final perceptron line is drawn using the following 2 points:

  xVal = data[j][0];

  yVal = (-data1[0]/data1[1])*xVal -(data1[2]/data1[1]);