

Perceptron project

Abdullah Alhabib
(371111359)
And
Abdullah Alhasan
(382124116)

Machine learning course

—

CS 435

—

Supervisor:
Dr.Mohammed Alsuhaibani

Tasks and Questions

- A. Using your own words, describe the binary classifier Perceptron and provide its pseudocode.
- B. Using Python, implement a binary perceptron.
- C. Using the binary perceptron, train the classifier to discriminate between:
 - class 1 and class 2
 - class 2 and class 3
 - class 1 and class 3
- D. Plot the train error rate and test error rate against the number of iterations.
 - According to your plot, what would be the ideal number of iterations to terminate the training?
- E. What is the train and test classification accuracies for each one of the classifiers after 30 epochs (iterations)?

THE PROCESS

Task A:

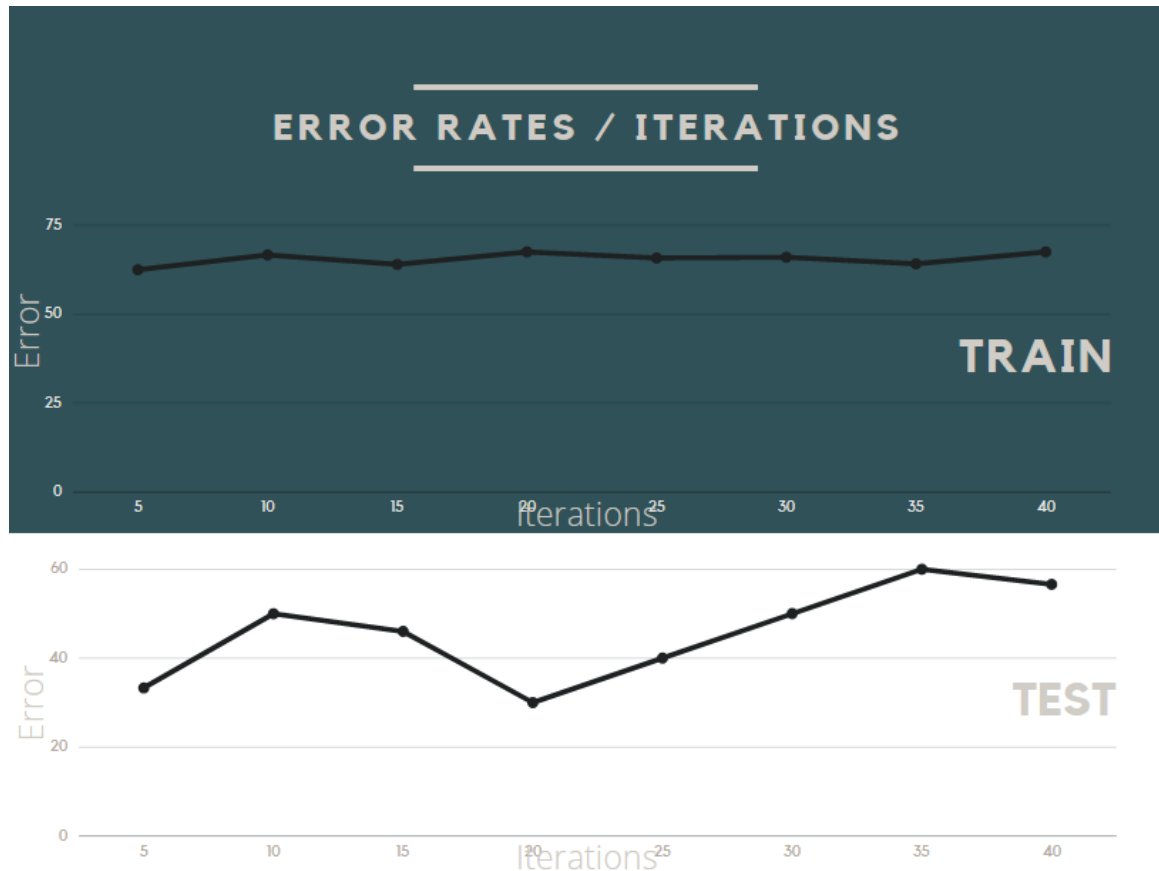
A binary perceptron is a machine learning model that inputs features, weights, and a predicted class; Then it determines whether those features and weights activate the perceptron for that class or not.

Task B:

```
1 def perceptron(x, y, weight, bias):
2     """
3     :param x: list of inputs.
4     :param y: expected output.
5     :param weight: list of weights for inputs.
6     :return: True if activation is positive, False if activation is negative.
7     """
8     activation = 0
9     for x_weight in zip(x, weight):
10         activation += weight*x + bias
11     if activation*y <= 0:
12         return False
13     return True
```

Task C is implemented in the program

Task D:



According to this plot, 35 iterations gives the best overall result

Task E:

Since the program runs on random input arrangements, we have train accuracies ranging from 50% to 60%, As well as test accuracies.

Resources

<https://scikit-learn.org/stable/modules/generated/sklearn.multiclass.OneVsRestClassifier.html>

https://en.wikipedia.org/wiki/Multiclass_classification#cite_note-bishop-2

https://jermwatt.github.io/machine_learning_refined/notes/7_Linear_multiclass_classification/7_3_Perceptron.html

<https://www.simplilearn.com/tutorials/deep-learning-tutorial/perceptron>