

Name - Shrikant Adhikarla
email - sadhika4@ur.rochester.edu
URID – 31407229
Coursework - CSC446-Machine Learning
Assignment – 2

Objective:

1. Implement perceptron for the adult income dataset using Python.
2. Experiment with performance as a function of number of iterations.
3. Discuss the interpretation of the results.
4. Create a README file for the same.

Files (included in submission):

1. adhikarla_perceptron.py: This is the perceptron implementation python3 file.
2. README.txt: Same file as this in txt format.
3. README.pdf: Currently, Open file.
4. adhikarla_perceptron.png: Shows the plot of Accuracy vs. #Iterations. Implementation is commented out in the python script.

Algorithm implemented

Perceptron algorithm implemented on a generic basis and adult income dataset is used as a case to predict whether a person's income exceeds \$50K/yr based on census data.

Repeat

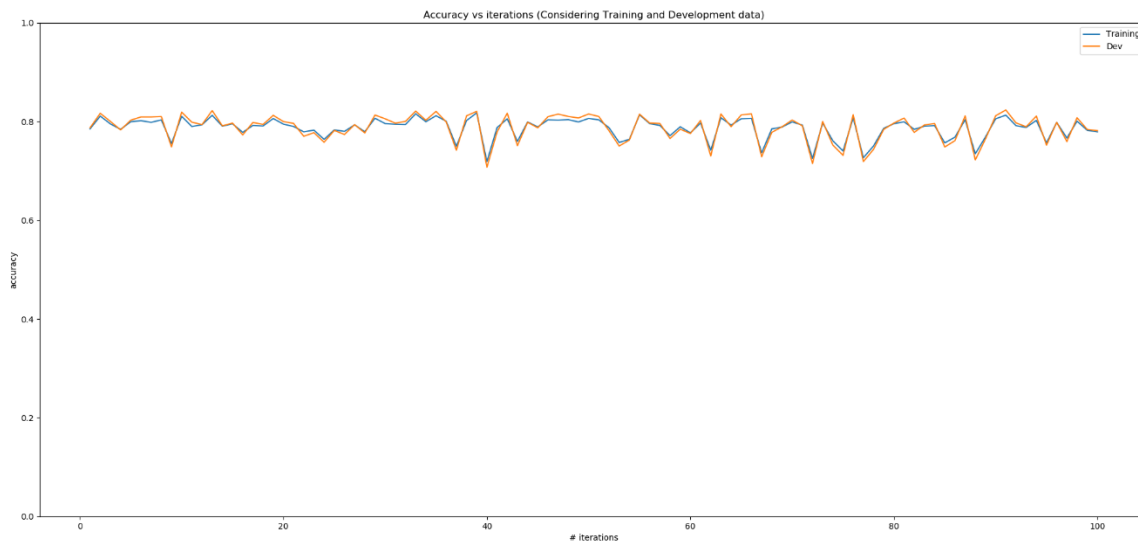
```
    for n = 1 to N
        if  $t_n * y_n \leq 0$  then
             $w = w + \text{learning\_rate} * y_n * x_n$ 
        end if
    end for
until for all n  $t_n = y_n$  or maxiters
```

Discuss the interpretation of results:

Usage of Development Dataset: Development data or validation data is used to alongside training data to test the accuracy after every iteration of training (or adjusting weight vector in this case). Dev data helps in avoiding the problem of high variance or overfitting. It is often used to fine-tune the hyper parameters of the classifier such as number of iterations, learning rate, etc. Multiple classifiers are

usually build using different set of hyperparameters and whichever model achieves higher accuracy in development data is chosen and finally an unbiased test data accuracy is presented.

Looking at the results of the accuracy vs #Iterations plot with both training and development data. Accuracy doesn't seem to improve with the number of Iterations at learning rate = 1. Results remain almost similar at lower values of learning rates as well. This most likely mean that the data points are non-linearly separable. Hence, there is wiggly behavior in the plot provided as the perceptron will never converge in this case and keep updating the weight vector indefinitely. Below attached is the plot for the experiment performed.



However, we're still getting around ~80% accuracy (+- 4% variability) in the development and test datasets, which is a good classifier accuracy for starters.