COMPUTER VISION

III. Decision functions

- Creating a sample data using two classes that are separable linearly and implementing the perceptron to determine an appropriate decision function.
- Repeating this process and including data samples that will make the two classes not linearly separable and see if the perceptron still works albeit with a lesser classification accuracy.
- Implementing the generalized decision function formulation (GDF) for the r^{th} order and with n dimensions. Illustrating that the program works for the cases:
  (a) Second degree with 2 dimensions
  (b) Third degree with 2 dimensions
  (c) Second degree with 3 dimensions
  (d) Third degree with 3 dimensions

INTRODUCTION:

The basic idea behind perceptron is to create a decision boundary that separates the input data into two classes, based on the features of the input. The decision boundary is created by assigning weights to the input features, such that the weighted sum of the inputs and bias term produces an output. The output is then passed through an activation function, such as a step function, to produce a binary output.

Perceptron learning rule is based on the concept of error correction, where the weights are updated iteratively to minimize the error between the predicted output and the actual output. The algorithm updates the weights based on the difference between the predicted output and the actual output, multiplied by the input features. The weights are updated in the direction that reduces the error.

Linearly separable data refers to data that can be separated by a straight line or hyperplane. In other words, it is possible to draw a straight line or hyperplane that completely separates the data points into different classes. The perceptron algorithm is

well-suited for linearly separable data because it can find a linear decision boundary to separate the data.

On the other hand, non-linearly separable data refers to data that cannot be separated by a straight line or hyperplane. In such cases, the perceptron algorithm may not be able to find a decision boundary that completely separates the data points into different classes. In such cases, more complex neural network algorithms, such as multi-layer perceptrons, convolutional neural networks, or support vector machines may be used to find a non-linear decision boundary to separate the data. These algorithms use a combination of linear and non-linear transformations to find a decision boundary that can separate the data.

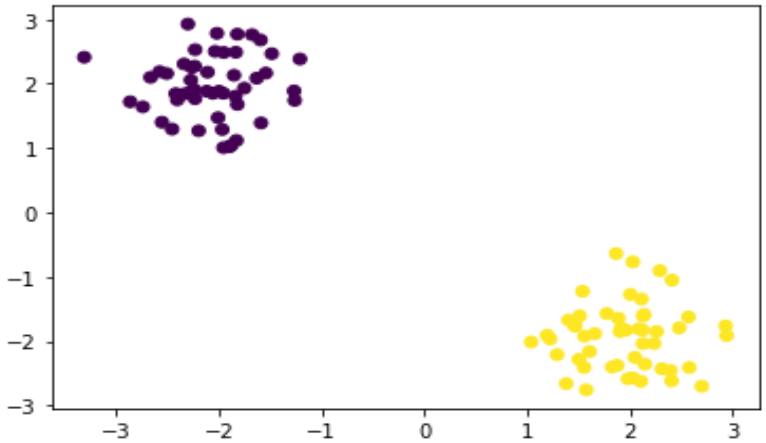
ALGORITHM & PROCEDURE:

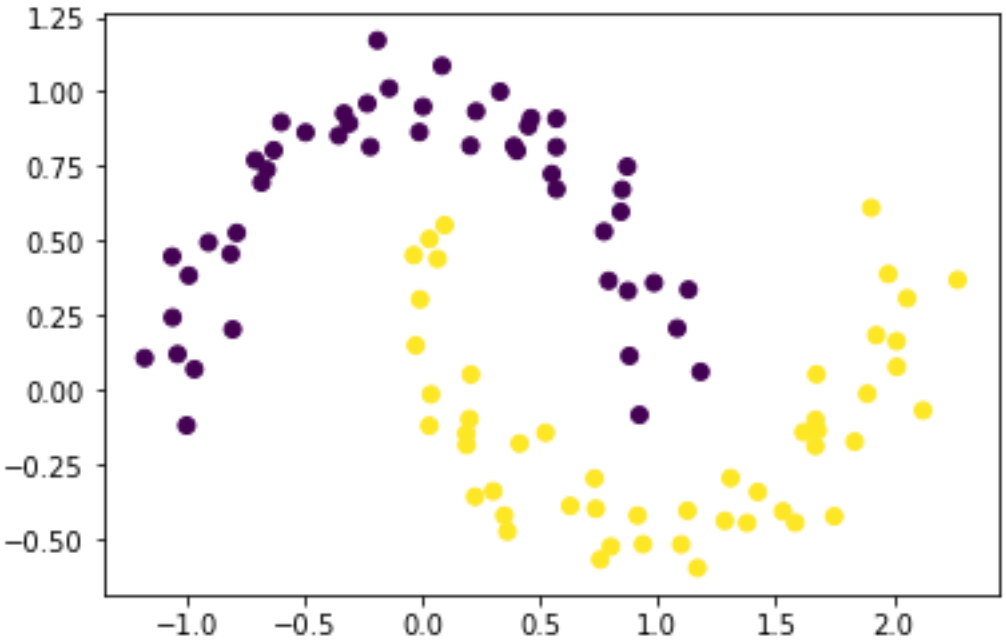
The algorithm works as follows:

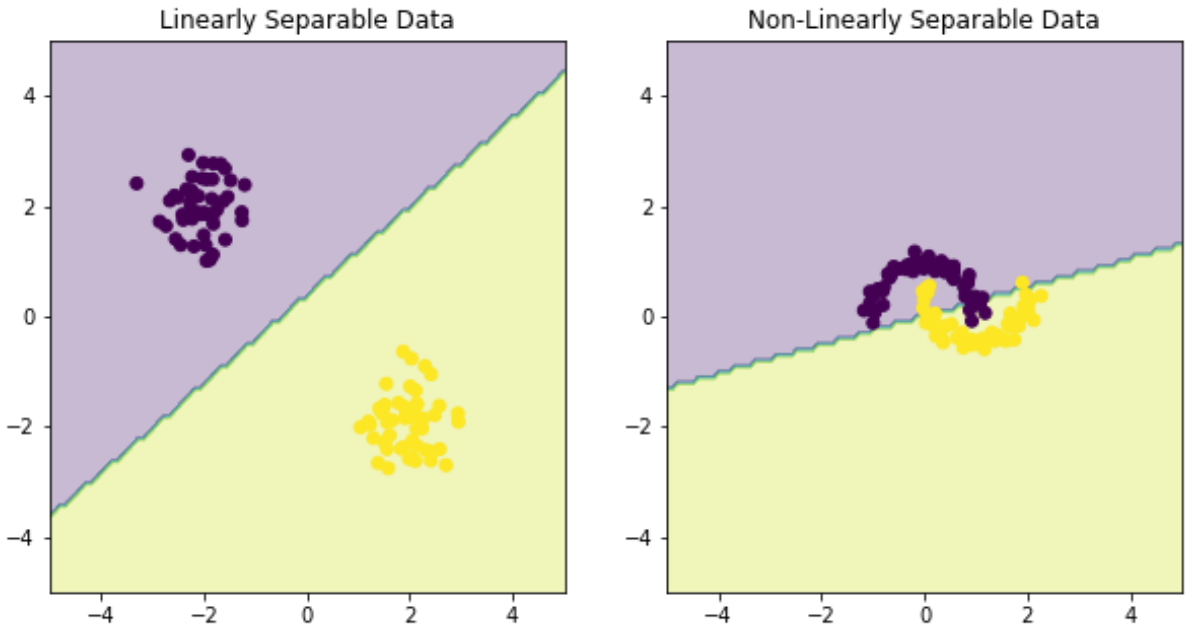
1. Initialize the weights w to small random values.
2. Repeat the following steps until convergence: a. For each input vector x, compute the predicted label y_hat using the current weights w: y_hat = sign(w · x) where sign is the sign function and · is the dot product. b. Update the weights as follows: w = w + α (y - y_hat) x where α is the learning rate and y is the true label of the input vector x.
3. When the algorithm has converged (i.e., the weights no longer change significantly), return the final weight vector w.

The perceptron learning algorithm is guaranteed to converge if the data is linearly separable. If the data is not linearly separable, the algorithm may not converge, or it may converge to a suboptimal solution.

RESULTS:


3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3


1.25
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0




```
Linearly separable data accuracy: 1.0
Non-linearly separable data accuracy: 0.89
```

CONCLUSIONS:

The accuracy achieved for linearly separable data is perfect (1.0), which means that the perceptron algorithm was able to find a linear decision boundary that perfectly separates the two classes. However, for non-linearly separable data, the accuracy is lower (0.89), indicating that the perceptron was unable to find a linear decision boundary that perfectly separates the two classes.

This highlights the limitation of the perceptron algorithm and the importance of choosing appropriate models and algorithms based on the problem at hand. For non-linearly separable data, more advanced techniques such as kernel methods, neural networks, or support vector machines (SVMs) may be necessary to achieve higher accuracy.