Introduction

In this report, I will explain the motivation behind each version of the algorithm and compare their performance based on various metrics.

Perceptron Algorithm

First, let's discuss the perceptron algorithm itself. It is a simple linear classifier that can be used for binary classification problems. It works by iteratively updating its weights based on the misclassified points until all the points are classified correctly. The decision boundary of the perceptron is a hyperplane that separates the two classes in the feature space.

Versions of Perceptron Algorithm

Now, let's discuss the three different versions of the perceptron algorithm that you implemented and compare their performance:

Version 1: Normalizing the Data using Mean Centering and using $w \pm X$

In the first version of the algorithm, you have implemented mean centering to normalize the data and introduced a bias term to improve the interpretability and stability of the algorithm. Mean centering involves subtracting the mean of each feature from all the data points. The motivation behind this is that it can make it easier to interpret the weights of the model as the impact of each feature on the output, and the weights will not be confounded by the mean of each feature. Additionally, mean centering can improve the numerical stability of the algorithm. You have reported that the algorithm converges during the 9th iteration and the equation of the decision boundary is: $(1.7)x_1 + (-0.2)x_2 + (0.0)x_2 = 0$. This version of the algorithm has the simplest decision boundary equation among the three versions.

Version 2: Without Normalizing the Data

In the second version of the algorithm, you did not normalize the data. The algorithm converged during the 19th iteration, and the equation of the decision boundary is: $(3.3000000000000003)x_1 + (-0.8999999999999999)x_2 + (0.0)x_2 = 0$.

Version 3: Implemented Perceptron using updated Weight and Learning Rate

In the third version of the algorithm, you have incorporated updated weights and a learning rate. The learning rate determines the step size of the weight updates, while the updated weights take into account the misclassified points and their corresponding feature vectors. The algorithm converged during the 19th iteration, and the equation of the decision boundary is: $(4.596317605509257)x_1 + (-2.1482385816641894)x_2 + (0.6014514711794805)x_2 = 0$.

Comparison of Performance

Based on the results you have reported, all three algorithms achieved the same accuracy and produced decision boundaries that correctly classify all the data points with maximum learning rate. However, considering the number of iterations and the simplicity of the decision boundary equation, the first version of the algorithm, which uses mean centering and $w \pm X$ normalization, appears to be the best choice among the three.

Conclusion

In conclusion, this report has provided a comparison of three different versions of the perceptron algorithm that you have implemented for a given dataset. The report has discussed the motivation behind each version of the algorithm, compared their performance based on various metrics, and concluded that the first version of the algorithm is the best choice among the three.