Problem Set 2: Linear Classifiers and Gradient Descent

Abhishek Velusamy

October 14, 2024

1 Introduction

The goal of this assignment is to implement a simple linear classifier using both SVM (hinge loss) and Softmax (cross-entropy loss) and optimize it using gradient descent. We apply these methods to a synthetic dataset and document the approach and results using Python and LaTeX.

2 Methodology

2.1 Score Function

The score function is used to compute the raw scores for each class, given input data X, weight matrix W, and bias b. The formula for computing the score is:

$$f(xi, W, b) = Wxi + b$$

This function is fundamental in both the SVM and Softmax loss functions, as it provides the class scores necessary to calculate loss.

2.2 Hinge Loss (SVM)

The SVM classifier uses hinge loss, which encourages a large margin between the correct class score and other class scores.

2.3 Softmax Loss

The Softmax classifier calculates the probabilities for each class and uses cross-entropy loss to measure the difference between the predicted probabilities and the true labels.

2.4 Regularization

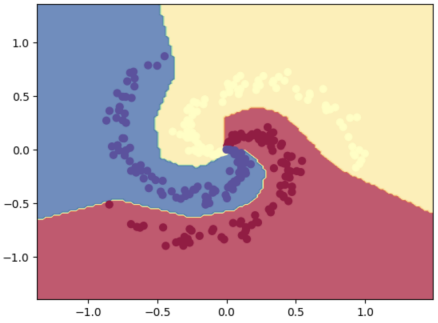
To prevent overfitting, L2 regularization is added to both loss functions. This term penalizes large weight values, encouraging the model to generalize better to new data.

2.5 Gradient Descent

Gradient descent is used to minimize the loss functions by iteratively updating the weights and biases. These updates are repeated until the loss function converges.

3 Results

The classifier was trained on a synthetic spiral dataset. Below are the decision boundaries learned by the model after training, visualized using a contour plot. The training loss decreased steadily over several epochs, as shown in the loss plot.



The figures show that the model successfully learned the decision boundaries for the three classes, and the training loss steadily decreased, indicating that the model was optimizing effectively.

4 Conclusion

In this assignment, we implemented linear classifiers using both SVM (hinge loss) and Softmax (cross-entropy loss), optimized using gradient descent. Regularization was applied to prevent overfitting. The model successfully classified the synthetic dataset, as evidenced by the decision boundary plot and the steady decrease in loss. A challenge faced during the assignment was tuning the learning rate to ensure the model converged properly without diverging.