Unit 5 - Week 2 - Rosenblatt's Perceptron

Assignment 02

The data file for this assignment is named assign02.txt. Read and analyze the data file to help you understand the data.

Due on 2019-08-27, 23:59 HKT

Instructions

1. Download the assignment data file assign02.txt.
2. Read and analyze the data file assign02.txt.

Requirements:

- Submit a single document containing your solutions.
- Include a clear explanation of your approach and reasoning.
- Submit the document before the deadline.

Question 1

(a) Implement a simple perceptron algorithm using the pseudo-random weights in assign02.txt.

(b) Add a bias term to the perceptron algorithm.

(c) Test the perceptron algorithm with different training data sets.

(d) Analyze the convergence of the algorithm.

Question 2

(a) Discuss the limitations of the perceptron algorithm.

(b) Describe how the perceptron algorithm can be extended to handle multi-class classification.

(c) Implement a multi-class perceptron algorithm.

Question 3

(a) Explain the concepts of support vectors and support vectors in the context of the perceptron algorithm.

(b) Discuss the role of support vectors in the decision boundary.

Question 4

(a) Discuss the implications of using a non-linear activation function in the perceptron algorithm.

(b) Implement a non-linear activation function in your perceptron algorithm.

Question 5

(a) Discuss the importance of feature scaling in the perceptron algorithm.

(b) Implement feature scaling in your perceptron algorithm.

(c) Compare the performance of the scaled and unscaled perceptron algorithms.

Question 6

(a) Discuss the role of regularization in the perceptron algorithm.

(b) Implement regularization in your perceptron algorithm.

(c) Compare the performance of the regularized and non-regularized perceptron algorithms.

Question 7

(a) Discuss the challenges of training a perceptron algorithm with large datasets.

(b) Implement a mini-batch stochastic gradient descent algorithm in your perceptron algorithm.

(c) Compare the performance of the mini-batch and full batch algorithms.

Question 8

(a) Discuss the limitations of the perceptron algorithm in terms of computational complexity.

(b) Implement a more efficient perceptron algorithm.

(c) Compare the performance of the efficient and non-efficient algorithms.

Questions 9 and 10

(a) Discuss the importance of choosing an appropriate learning rate in the perceptron algorithm.

(b) Implement an adaptive learning rate in your perceptron algorithm.

(c) Compare the performance of the adaptive and fixed learning rate algorithms.

(d) Discuss the implications of choosing a sub-optimal learning rate.

Questions 11 and 12

(a) Discuss the importance of choosing an appropriate number of hidden layers in the perceptron algorithm.

(b) Implement a deep perceptron algorithm.

(c) Compare the performance of the deep and shallow perceptron algorithms.

Questions 13 and 14

(a) Discuss the importance of choosing an appropriate activation function in the perceptron algorithm.

(b) Implement a custom activation function in your perceptron algorithm.

(c) Compare the performance of the custom and standard activation functions.

Questions 15 and 16

(a) Discuss the importance of choosing an appropriate training set in the perceptron algorithm.

(b) Implement a data augmentation technique in your perceptron algorithm.

(c) Compare the performance of the augmented and original datasets.

Questions 17 and 18

(a) Discuss the importance of choosing an appropriate validation set in the perceptron algorithm.

(b) Implement a cross-validation technique in your perceptron algorithm.

(c) Compare the performance of the cross-validated and non-cross-validated algorithms.

Questions 19 and 20

(a) Discuss the importance of choosing an appropriate testing set in the perceptron algorithm.

(b) Implement a holdout validation technique in your perceptron algorithm.

(c) Compare the performance of the holdout and cross-validated algorithms.

Questions 21 and 22

(a) Discuss the importance of choosing an appropriate regularization technique in the perceptron algorithm.

(b) Implement a custom regularization technique in your perceptron algorithm.

(c) Compare the performance of the custom and standard regularization techniques.