

ECE421 Programming

Assignment 1: Questions

1 Pocket Algorithm – Questions

1. Refer to the documentation, what is the functionality of the tol parameter in the Perceptron class? (2 marks)
 - a. The tol parameter refers to the tolerance parameter when training the model to trigger early stopping of the training process and prevent over-fitting. If the model's performance does not improve by more than this amount after an iteration, the training process will stop early.
2. If we set max iter=5000 and tol=1e-3 (the rest as default), does this guarantee that the algorithm will pass over the training data 5000 times? If not, which parameters (and values) should we set to ensure that the algorithm will pass over the training data 5000 times? (2 marks)
 - a. No. It is possible that the training process will stop early due to seeing no significant improvement in classification. If the model does not improve more than the tol value, it will stop training before 5000 iterations. To ensure that the model goes over the training data 5000 times, we can set the tol value to 'None'.
3. How can we set the weights of the model to a certain value? (2 marks)
 - a. To set the weights of the model to a certain value we create an array filled with this value that is the same length as the number of weights we hope to train and set the Perceptron's .coef_ attribute to equal this array.
4. How close is the performance (through confusion matrix) of your NumPy implementation in comparison to the existing modules in the scikit-learn library? (2 marks)
 - a. My NumPy implementation was fairly close to the existing scikit-learn modules. According to the confusion matrix, it only differed by at most one misclassified data point.

2 Linear Regression - Questions

1. When we input a singular matrix, the function linalg.inv often returns an error message. In your fit LinRegr(X train, y train) implementation, is your input to the function linalg.inv a singular matrix? Explain why. (2 marks)
 - a. My input to the linalg.inv function is not a singular matrix because it is actually the multiplication of X and its transpose
2. As you are using linalg.inv for matrix inversion, report the output message when running the function subtestFn(). We note that inputting a singular matrix to linalg.inv sometimes does not yield an error due to numerical issue. (1 marks)
 - a. The function subtestFn() returns 'ERROR' and no weights
3. Replace the function linalg.inv with linalg.pinv, you should get the model's weight and the "NO ERROR" message after running the function subtestFn(). Explain the difference between linalg.inv and linalg.pinv, and report the model's weight. (2 marks)

- a.** `linalg.pinv` calculates the pseudo-inverse of X , rather than the pure inverse like `linalg.inv`.
The models weights are:
 - i. `[1.04360964e-14, 2.00000000e-01, 4.00000000e-01]`