ASSIGNMENT-1

Neural Network and Fuzzy Logic (BITS F312)

Submission date: March 3, 2022 (Hard deadline)

Instructions:

1. The python codes of this assignment questions must be implemented from the scratch.
2. Only the numpy, pandas, matplotlib packages are allowed. Scikit Learn is not allowed for this assignment.
3. If anyone will use the Scikit Learn package then his/her assignment will not be evaluated. For selecting training, validation and test instances you can use the train_test_split function.

Q1. The training feature matrix and training output for linear regression and polynomial regression models are given in the folder Q1_Q2_data with file names as xtr.mat, ytr.mat respectively. Evaluate the weight vectors from the training data using linear regression (LR), ridge regression (RR), and least angle regression (LAR) models. Use the estimated weight vector from the training of each model to evaluate the predicted output vector for the test feature matrix. Evaluate the mean square error (MSE), mean absolute error (MAE), and correlation coefficient (CC) between the actual test output and predicted test output. The test feature matrix and test output are given in the folder Q1_Q2_data with file names as xte.mat and yte.mat, respectively. You must implement LR, RR, and LAR algorithms using batch gradient descent (BGD), stochastic gradient descent (SGD) and mini-batch gradient descent (MBGD), respectively. You must perform feature normalization for implementing the model in the Jupyter notebook. [6 marks]

Q2. Implement polynomial regression (PR), PR with L_2-norm regularization, PR with L_1-norm regularization for the dataset given in Question 1. Evaluate MSE, MAE and CC measures for the test data. You must have to implement PR, PR with L_2-norm regularization and PR with L_1-norm regularization using BGD, SGD and MBGD respectively. You can do the feature mapping for PR as $X = [x_1\ x_2] \rightarrow Z = [x_1\ x_2\ x_1^2\ x_2^2\ x_1x_2]$. You must perform feature normalization after feature mapping for implementing the LR models in Jupyter notebook. [6 marks]

Q3. Implement logistic regression (LOR), LOR with L2-norm regularization, and LOR with L1-norm regularization models for binary classification tasks. The training feature matrix, training output vector, test feature matrix, test output vector are given in the Q3_data folder with file names as xtr.xlsx, ytr.xlsx, xte.xlsx, and yte.xlsx, respectively. For the test data, evaluate the accuracy, sensitivity and specificity measures for each type of LOR model. You must have to perform feature normalization for implementing the LOR models in Jupyter notebook. [6 marks]

Q4. Implement multiclass LOR models using 'one vs. one' and 'one vs. all' multiclass coding-based approaches. The dataset (data.xlsx) contains the feature matrix (first 60 columns) and class label vector (61th column is the class label vector). You must use holdout validation (70% as training, 10% as validation and 20% as testing) and 10-fold cross-validation (CV) to select training and test instances for each type of multiclass LOR model. Evaluate individual class accuracy and overall accuracy along each fold for Multiclass LOR model with 10-fold CV. Similarly, for holdout validation, evaluate the individual class accuracy

and overall accuracy of multiclass LOR models. You must consider multiclass LOR, multiclass LOR with L2-norm regularization, and multiclass LOR with L1-norm regularization models and evaluate the classification performance using holdout and 10-fold CV methods. You must perform feature normalization for implementing the multiclass LOR models in the Jupyter notebook. [6 marks]

Q5. Implement Maximum likelihood (ML) and maximum a posteriori (MAP) classifiers for the multi-class classification task. You must use the dataset as mentioned in Q4 to implement MAP and ML classifiers. You can select 70% of the instances as training and the remaining 30% as testing for both ML and MAP classification models. Evaluate the individual class accuracy and overall accuracy measures for MAP and ML classifiers. You must have to perform feature normalization for implementing the MAP and ML models in Jupyter notebook. [6 marks]