CS584: Machine Learning - Assignment 3

Theory Questions(full marks: 1)

1. Bishop's Book "Pattern Recognition and Machine learning" - Exercise 5.1

2. Bishop's Book "Pattern Recognition and Machine learning" - Exercise 5.7

Programming Questions(full marks: 5)

1. Load the CIFAR10 dataset and select a subset of three classes. Split the training set into a training and validation subsets. Vectorize the images and encode the known class labels using categorical encoding. Design a fully connected neural network to perform multi-class classification of this data. Justify your network design decisions (number of hidden layers, number of units per layer, loss function, and evaluation metric). Build and compile the network you designed. Plot training and validation loss as a function of epochs, then plot training and validation accuracy. Tune model hyper parameters to improve performance. Retrain the final model and test performance on the test collection. Report the performance you obtain. Make sure that your program can save and load the weights of the trained network.

2. Load the spam email data from the UCI repository "spambase"(https://archive.ics.uci.edu/ml/datasets/Spambase), Prepare the data you loaded as a tensor suitable for a neural network. Normalize features as needed. Explain the steps you perform in preparing the data and justify them. Write a function "load_spam_data" to load the data and split it into a training and testing subsets. Repeat the steps you performed in the CIFAR10 question.

3. Load the crime data from the UCI repository "Communities and crime''(https://archive.ics.uci.edu/ml/datasets/Communities+and+Crime). Prepare the data you load as a tensor suitable for a neural network. Normalize features as needed. Explain the steps you perform in preparing the data and justify them. Write a function "load _ crime _ data" to load the data and split it into a training and testing subsets. Repeat the steps you performed in the CIFAR 10 question except that in this case you are required to perform k-fold cross validation. To report the cross validation results, average the validation error of all the folds and plot it as a function of epochs. Retrain the final model on all training data (i.e. all folds) and test final performance on test set.

Submission instruction:

1. For programming part, use Python Keras or Tensorflow or Pytorch. We must be able to view your report and execute your program in order to grade it.

2. You must submit your program and report. The report should contain the summary of program design issues, description of specific problems you faced and the way in which you solved them and sample input/output results (text/graphic). Report prepared as a PDF file.

3. Due by Oct. 24, 11:59 pm.