Assignment 4
Due May 17, 2020 at 11.59 PM via ilearn Assignment-4 Submission folder

Independent Reading. 1. Chapter 4 of Pattern Recognition and Machine Learning by C. M. Bishop.

Note: Sample codes for this assignment are either in PyTorch or Matlab. You are free to convert this code in the framework of your choice in python.

Problem 1. [4 pts] **Feature Extraction.** In this problem, you are required to extract the Deep Convolutional Neural Network (CNN) features for a dataset. The dataset can be downloaded from iLearn. This dataset named tiny-UCF101 is a sub-sampled version of the UCF101 dataset. It is an activity recognition dataset with 101 categories. Under the root directory of the dataset are the category directories, each containing images sampled from original videos of the UCF101 dataset.

You need to extract features from these images using the ResNet50 architecture available in PyTorch. A sample code for this problem in Pytorch is given in `extract_features.py`. Some portions of the code is already filled in for convenience. The final code should load the images, extract the features and append them to the 'feature' list along with the corresponding labels. The output of this code is the file 'ucf101dataset.mat', which is to be used in the next problem. This file should have

- 'feature' of dimension 13320×2048, where 13320 is the number of images and 2048 is the feature dimension obtained using ResNet50
- 'label' is a vector of length 13320 containing labels from 0 to 100 for the 101 categories.

Problem 2. [6 pts]

(a) [4 pts] **Logistic Regression.** In this problem, you should implement the multinomial logistic regression using the dataset extracted in Problem 1. The train and test split are mentioned in `subset.mat`. The starter code for this problem is `logistic_regression.m`. This code first separates the train and test sets. You should use variables 'trfeature' and 'trlabel' for training and 'tefeature' and 'telabel' for testing. Please remember to map the labels properly for testing. You need to fill in the function named `apply_gradient.m`, which returns the updated parameter θ after a single pass of gradient descent using the given data points and labels. You also need to fill up certain the portions as mentioned in `logistic_regression.m`. Report the test accuracy you obtain.

(b) [2 pts.] **ROC.** In this problem you need to fill in the getROC.m function to implement the Receiver Operating Characteristics curve. The output of this function should be TPR, FPR represening True Positive Rate and False Positive Rate respectively. `logistic_regression.m` calls `getROC.m` at the end for the 50^{th} category of tiny-UCF101 and plots it. Display it in your report.

Submission Protocol. You should submit codes as well as explanations to each problem (if required). You should add comments to your codes to make them reader friendly. If you may require to call functions for a problem, you may do so, but include them in your submission. You MUST also include a report (in pdf) written electronically (using the likes of LaTeXor MS Word). It should contain explanations, images, etc (as required).

Each student must do the assignment independently, although you may discuss prior to that. While discussion is allowed, we will be particularly careful about any plagiarism, whether from each other or from other sources.