EE243: Advanced Computer Vision
Assignment #4

Ali Jahanshahi (862029266)

May 2019

1 Problem 1: Feature Extraction

In this problem, we are asked to extract the Deep Convolutional Neural Network (CNN) features for a dataset. The provided dataset named $tiny-UCF101$ that is a sub-sampled version of $UCF101$ 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 $UCF101$ dataset. We extracted features from these images using the $ResNet50$ architecture which is available in PyTorch, shown in Figure 1. We used the starter code for this problem, $features.py$. The code loads 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 going to be used in the next problem. This file contains:

- '$feature$' of dimension 13320$\times$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.

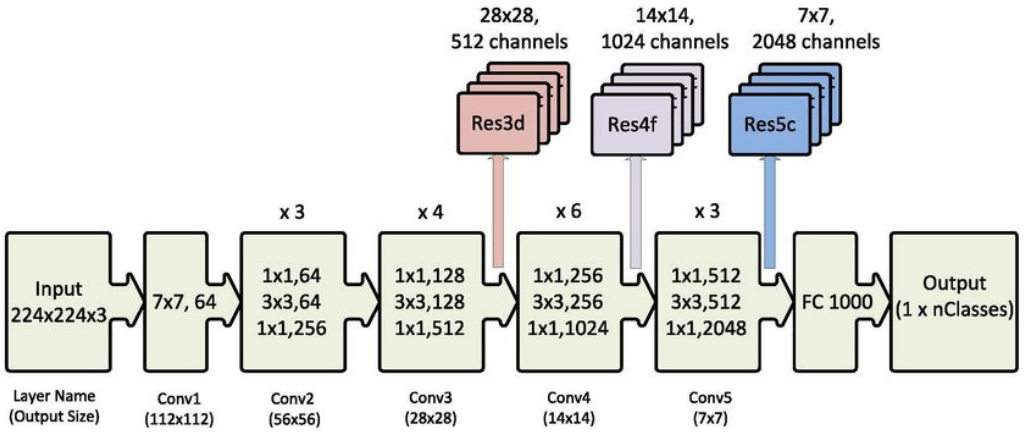


Figure 1: $ResNet50$ architecture.

The codes and generated features are attached to this report.

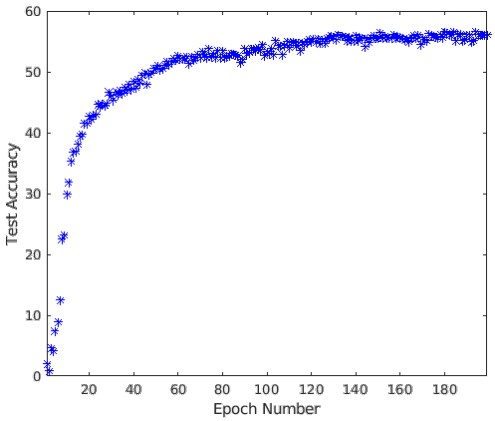
2 Problem 2: Logistic Regression

2.1 Part a

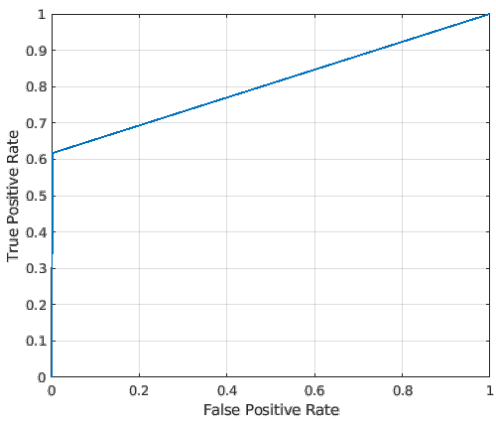
In this part, we are asked to implement the multinomial logistic regression using the dataset extracted in Problem 1. The train and test split are mentioned in $subset.mat$. We used the provided codem $logistic\_regression.m$. The code first separates the train and test sets. We used variables '$trfeature'$ and '$trlabel'$ for training and '$tefeature'$ and '$telabel'$ for testing. Function named $apply\_gradient.m$ returns the updated parameter θ after a single pass of gradient descent using the given data points and labels. We obtained 55.828707% accuracy, which varies a little bit at each run.

2.2 Part b

In this part we are asked to to fill in the getROC.m function to implement the Receiver Operating Characteristics curve. The outputs of this function are TPR and FPR, representing 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. Figure 2 shows obtained ROC and TPR-FPR curve plotted by the code.



(a) ROC Curve.



(b) TPR vs FPR.

Figure 2: ROC & TPR-FPR.