COMP3314/CSIS0314
Machine Learning
Assignment 4

Due date: May 5, 2015 (11:30 pm)

Tiny Image Classification (100 Marks)

You will be working in teams of 2 to design a multi-class classifier for object recognition. We will use a subset of the CIFAR-10 dataset collected by Krizhevsky, Nair, and Hinton, which consist of 32 × 32 RGB images. We will only use 6 object categories instead of the original 10. The specific target labels are: **0-airplane, 1-automobile, 2-bird, 3-cat, 4-deer, 5-dog**. You should take a look at example visualzations on this website: `http://www.cs.toronto.edu/~kriz/cifar.html` You are provided with 3000 labeled images (500 from each category) as well as 31800 additional unlabeled images.

The goal is to maximize the accuracy of your classifier on a test set. You're allowed to use any supervised or unsupervised learning models you wish and can use any publicly available code. You will be marked not only on the resulting accuracy, but also on the creativity of your solutions and how well you can explain them.

We will use the Kaggle platform for you to compare your solutions to other teams in class (`https://inclass.kaggle.com/c/tiny-images-classification`.) Note that you must participate in the competition to receive marks for the assigment. We will provide the necessary data and further information on the Kaggle website. Please create an account using an email address ending with **hku.hk** and join the competition hosted on the link above. Specifically, the website allows you to upload lists of classification results on a particular test set and automatically performs evaluation and ranking. You will be provided with the test set without labels.

Baseline classifier

This is a difficult classification task. A baseline KNN classifier which only uses the labeled data will be provided. The KNN classifier achieves 36.17% accuracy. You will be marked based in part on how much you can improve upon the baseline.

Report

You should hand in a report (no more than 6 pages including all graphics, single space, 12-pt Times New Roman). The report should include

Introduction. At a high-level, what are the approaches you considered and why?

Method. Describe the approach you chose. For example, how did you preprocess the data and what algorithms did you implement/use?

Results. Experiments demonstrating the process you took to arrive at the solution. For example, compare your approach against one other approach you tried, as well as the KNN baseline classifer. Compare against different setting of model parameters (e.g., regularization constants, number of hidden units or structure of a neural network, types of kernel in a SVM, etc.), or different usage of unlabeled data if you choose to use them. Describe the approach you took to decide on the model parameters.

Conclusion. Summarize what you've learned from the experiments and speculate on how you might improve the performance.

References. References to any publically available code you used or writings you drew ideas from.

Code

Submit all the code you wrote for the assignment, specifically, include a function following the format of run_prediction.m, and your final trained model as a .mat file. The function loads your trained model and generates a vector of predictions on data_test.mat. Save this vector in mypredictions.mat. No training is involved in the script, only testing. It should not run for more than 10 minutes.

Hints

This assignment is an open-ended exercise designed to give you a chance to attack a real machine learning problem. Feel free to try any of the approaches you learned in class (e.g., KNN, neural networks, SVM) as well as other approaches as long as you can explain and justify them. You do not have to make use of the unlabeled data. They are provided to simulate the typical real-world scenario of having much more unlabeled data avaialble than labeled data.

You can of course make use of code provided in previous assignments, but you might find more efficient/flexible implementations online. The ability to understand and make use of existing APIs is an important skill to have.

A couple of examples to consider:

Netlab `http://www.aston.ac.uk/eas/research/groups/ncrg/resources/netlab/`

libSVM `http://www.csie.ntu.edu.tw/~cjlin/libsvm/#matlab`

Submission

Submit your code (including the trained model to be loaded by run_prediction.m) to Moodle as a zip/tar.gz file and the report as a pdf before 11:30 pm on May 5th. No paper write-up is required.