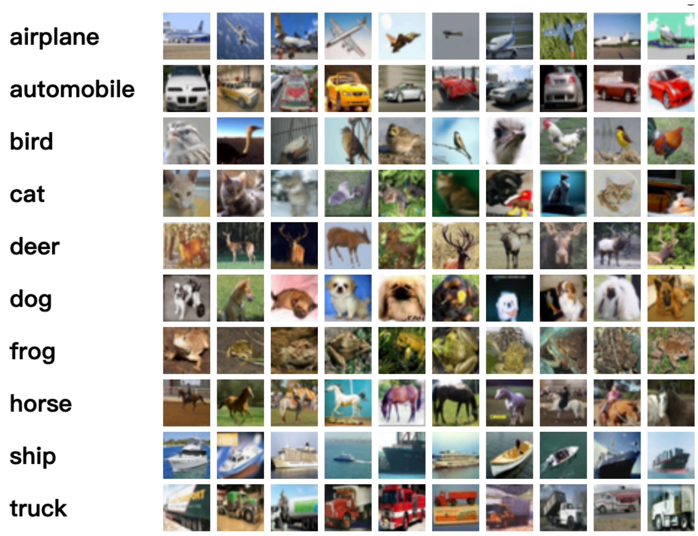
COMP3055 Machine Learning Coursework

Deadline: 4pm Tuesday Dec 31, 2019
Submit an electronic copy via Moodle

The coursework aims to make use of the machine learning techniques learned in this course to classify objects in images using CIFAR-10 dataset. The CIFAR-10 dataset consists of 60000 32x32 color images in 10 classes, with 6000 images per class. There are 50000 training images and 10000 test images. The dataset is divided into five training batches and one test batch, each with 10000 images. The test batch contains exactly 1000 randomly-selected images from each class. The training batches contain the remaining images in random order, but some training batches may contain more images from one class than another. Between them, the training batches contain exactly 5000 images from each class. Here are the classes in the dataset, as well as 10 random images from each:



See http://www.cs.toronto.edu/~kriz/cifar.html for more details.

You will perform the following tasks using Python with necessary libraries such as Scikit-learn and pytorch:

Task 1: You can find CIFAR-10 dataset from above link. You can download the dataset and load the training and testing data according to the description from above link. Or you can use PyTorch to download the dataset (see task 3). Apply PCA to reduce the original input features into new feature vectors with different amount of information kept, e.g. 10%, 30%, 50%, 70%, 100%.

Task 2: Design and implement object recognition system using SVM. Do the following:

1. Apply linear SVM with training data to do 10-fold cross validation to train and validate your models with different input feature vectors from Task 1 (original input and reduced input calculated from Task 1).
2. Using test data to compute f1 values (for each class) and accuracy for your models and plot figures showing result vs feature dimension.
3. Use polynomial and RBF kernels to train different SVM models with original input features (non-PCA) and do 10-fold cross validation to train and validate your models.

Note that each kernel has different parameters to set, for example, orders for polynomial model and sigma for RBF kernels. You should try different parameters as well.

4. Use test data to compute the f1 values for each class and accuracy for your models with different kernels and parameters.

Task 3: Design and implement object recognition system using CNN. You should use PyTorch as deep learning framework. Note that there is no specific requirement on the actual architecture of your CNN. However it should not be the same as the one used in lab 10. You should try to play around with convolutional and pooling layers (for example, more layers or more kernel windows) for best result you can get. Use test data to compute the f1 values for each class and accuracy for your CNN.

Note that in fact PyTorch does include classes and functions for downloading and making use of CIFAR-10 dataset. See https://pytorch.org/tutorials/beginner/blitz/cifar10_tutorial.html#sphx-glr-beginner-blitz-cifar10-tutorial-py for more details.

Task 4: Based on your experiences of performing task 2 and task 3 and findings therein, in your own words, compare and contrast the performances (accuracy, precision and recall, f1), computational complexity (time), level of overfitting of the two approaches. To look at the level of overfitting, you can compare the performance of a given model on the training data with test data and see how different they are. State which one you think would be a better approach to this problem under certain situation and explain why.

Important Notes: CIFAR-10 contains 60000 images which may cost a lot of time for training. Depending on your computer, using the whole dataset may take too much time for for both Task 2 and 3. You may use only a subset of CIFAR-10, (e.g. 5000 or 10000 images for the training). Or a better way, you can make use of HPC server from our school. While HPC offers (probably) better computational power than your own computer, more importantly HPC can run your program in background while you and your computer are resting. Using HPC is especially useful if you want to try your CNN for Task 3 with GPU. Upon request, I will offer a tutorial on how to use our HPC for your coursework and give your access to HPC all the way to coursework deadline. However if you are not interested in using HPC, you will not lose any marks for this coursework.

What to submit: A report of no more than 6 pages including all the figures and tables summarizing how above tasks are done, justification on your decisions involved, and the results of your analysis. A zipped file with all your source code. Note that you should properly organize your code with appropriate comments for easy of marking and running.

Marking scheme: this coursework takes 30% of your total marks in this module. The marking distribution is given in 100 scaling as follows:

1) Completeness of task 1 (10 marks)
2) Completeness of task 2 (30 marks)
3) Completeness of task 3 (30 marks)
4) Completeness of task 4 (15 marks)
5) Report writing (10 marks)
6) Coding with proper comments and organization (5 marks)