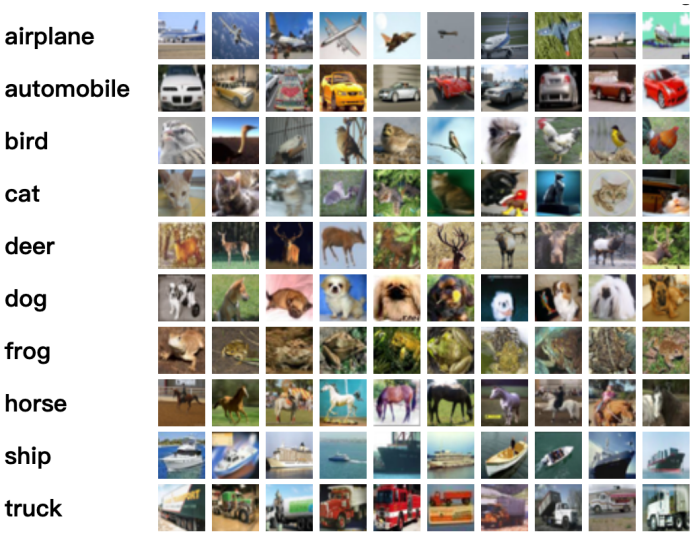
COMP3055 Machine Learning Coursework

Deadline: 4pm Friday Dec 17, 2021
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 10000 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 an object recognition system using Multilayer Perception (MLP). Do the following:

1. Apply MLP with multi-class classification to the training data. 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. Train your MLP with different hyper-parameters (hidden layer size, learning rate, etc.) on the original input features (non-PCA) and do 10-fold cross validation to train and validate your models.
4. Use test data to compute the f1 values for each class and accuracy for your models with different hyper-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. Your CNN does not need to be as complicated as those existing CNN from research papers. However it should not be the same as the one used in the lab. You should show your efforts to design your CNN with different architecture (for example, more layers or more kernel windows, different activation functions, etc.) for the 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 both Task 2 and 3. You may use only a subset of CIFAR-10, (e.g. 5000 images for the training).

What to submit: A report of no more than 2000 words 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 (20 marks)
3) Completeness of task 3 (30 marks)
4) Completeness of task 4 (20 marks)
5) Report writing (15 marks)
6) Coding with proper comments and organization (5 marks)