Machine Learning
COMP3010

ASSIGNMENT

Due Date: Week 12 - Friday 20 May 2022 at 11:59pm.
Weight: 2**0%** of the unit mark.

Note: *This document is subject to corrections and updates. Announcements will be made promptly on Blackboard and during lectures. Always check for the latest version of the assignment. Failure to do so may result in you not completing the tasks according to the specifications.*
Your total score for this assignment will need to be at least **20 marks** out of the total 100 marks. You will fail this unit if you cannot meet this basic pass mark, regardless of your scores in the mid-semester test and final exam.

1 Overview

This assignment provides an opportunity for you to demonstrate how you can use what you have learned from lectures and tutorials to train a deep neural network and investigate the properties of the learnt convolutional neural network features on data sets with relatively small size. External codes are allowed but you must refer them properly in your report. Feel free to use the work you have done in your practical exercises.

A substantial attempt for this assignment is required to pass this unit. A mark of 20% or more is considered a substantial attempt. This means that your codes should work for at least method and produce reasonable performance. You will not pass this unit if your total mark of this assignment is lower than 20%, even if you achieve full marks in your mid-semester test and final exam.

2 Background

It is widely believed that the pretrained deep neural network models on large image dataset can be used for other small image datasets through fine tuning or building simple linear models on top of the learnt convolutional neural network features. In this assignment, you will develop machine learning programs to train and test deep neural networks for some small image datasets and investigate the learnt features in the last and intermediate layers through building simple machine learning models (e.g. nearest neighbor methods, linear SVM, linear regression or softmax regression) on the extracted features and testing the performance on testing data.

3 The Tasks

3.1 Task 1: Comparing the performance of fine tuning and training from scratch

[30 marks]

- Choose one from the following pretrained models (or any more advanced neural networks) based on the image classification dataset ImageNet, which you can find in https://pytorch.org/vision/stable/models.html,
  - AlexNet
  - VGG16
  - ResNet18
  - ResNet50

  and choose one image dataset from MNIST, CIFAR-10 or CIFAR-100.
- Develop two programs that train the selected neural network model with two different initializations: 1) from scratch with randomly initialized weights; and 2) using fine tuning (i.e., starting from the pretrained weights in the pretrained model), and compare their performances on the testing data, and report your findings.
- You are expected to report how the models are modified for the new task, the hyper-parameters, the performances, and your explanation.

3.2 Task 2: Comparing the performance of different pretrained models

[30 marks]

- Use the same dataset as in Task 1.
- Use the best training strategy found in Task 1, fine tuning, or training from scratch.
- Choose three pretrained models listed in Task 1 (including the one used in Task 1) and compare the performance of these three neural network architectures.
- Report how the models are modified for the new task, the accuracies, and your explanation.

3.3 Task 3: Investigate the learnt CNN features from the last convolution layer and one intermediate layer.

[40 marks]

- Use the best model for the selected dataset you have trained in Task 2.
- Train a simple model (e.g., SoftMax regression, k-nearest neighbor method or linear SVM)

using each of the following three different features as inputs:

1) The original input, i.e., the pixel values of the original images.
2) The CNN features of the last convolution layer, i.e., the outputs of the layer (usually a pooling layer) just before the output layer of the whole model for all the training and testing examples.
3) The CNN features of one intermediate layer. You can use list(net.children) to find all the conv layers of a model net. Select one of them except for the last one which was used in 2) and add a pooling layer to reduce the dimension to be the same as the number of channels of the selected conv layer.

For SoftMax regression, you can either 1) extract the features first and then train the simple models, or 2) during training, freeze the weights of the layers which are used to extract features and only update the weights of the linear model of SoftMax regression.

- Compare the performances and report your findings.

4 Specifications and Marking Guide

4.1 Report: 50%

A written report must be submitted, in PDF format, to Blackboard by the due date. This submission must contain

- A completed assignment cover sheet
- Printout of your source code
- A document that includes:
  - Statements on how much you have attempted the assignment.
  - The detail of your implementation for each task: this must clearly indicate your approach, and how the features you extract, the methods you use for model selection. It must allow the marker to understand how you approach the machine learning tasks. If a validation dataset is not available, you are expected to split the training dataset into two subsets: one for training and the other for model selection.
  - The performance of your program on the testing dataset.
  - Supporting diagrams, figures, tables that help describe your programs and performance clearly.
  - References that your implementation is based on or inspired from.

Your report will be marked based on: 1) the clarity and presentation (20%); 2) the description of your implementation and the judgements of your design (40%); and 3) experimental results on the testing data and discussions (40%).

4.2 Implementation: 50%

Your implementation will be marked based on the quality of your code (30%) and whether your Jupyter Notebook programs work in Google Colab or the d2l package provided by the textbook and produce reasonable performance (70%). Your codes are expected to be well written with comments and good structures.

4.2.1 Evaluation Environment

- Your implementation will be tested using Google Colab or the d2l package provided by the textbook.

4.2.2 Your electronic submission

- Your electronic submission of the source code (i.e., Jupyter Notebook) to Blackboard must be a compressed file (zip) with the following naming convention

  [surname]_[given names]_[student ID].zip

  For example, if your name is Mike Jordan and your student ID is 123456 then your compressed filename is

  jordan_mike_123456.zip

- Your electronic submission should contain
  - Source codes with Jupyter Notebook, the best model of your assignment and a Jupyter Notebook to verify the performance of this model reported in your written report: submit to "Assignment – JupyterNotebook" in Blackboard
  - Your written report: submit to "Assignment – Written Report" in Blackboard

4.2.3 Your demonstration

A demonstration session will be conducted during a practical to verify your submission. You will be asked questions about your programs. The purpose of this demonstration is to make sure that your submission is your own work, you know exactly what you are doing, and the reported performance are correct in your written report.

5 Submission

You are required to submit your assignment, including your written report and source code, by Friday 20-May-2022, 11:59pm Perth time.

Upload your submission electronically via Blackboard, under the Assessments section.

You are responsible for ensuring that your submission is correct and not corrupted. You may make multiple submissions, but only your newest submission will be marked.

You will need to make yourself available for the demonstration session. Exact date and time will be announced on Blackboard.

The late submission policy (see the Unit Outline) will be strictly enforced. A submission 1 second late, according to Blackboard, will be considered 1 day late. A submission 24 hours and 1 second late will be considered 2 days late, and so on.

You must also submit a completed, signed **“Declaration of Originality”** form.

6 Academic Misconduct – Plagiarism and Collusion

Please note the following, which is standard across all units in the department:

> Copying material (from other students, websites or other sources) and presenting it as your own work is plagiarism. Even with your own (possibly extensive) modifications, it is still plagiarism.
>
> Exchanging assignment solutions, or parts thereof, with other students is collusion. Engaging in such activities may lead to a grade of ANN (Result Annulled Due to Academic Misconduct) being awarded for the unit, or other penalties. Serious or repeated offences may result in termination or expulsion.
>
> You are expected to understand this at all times, across all your university studies, with or without warnings like this.

END OF ASSIGNMENT.