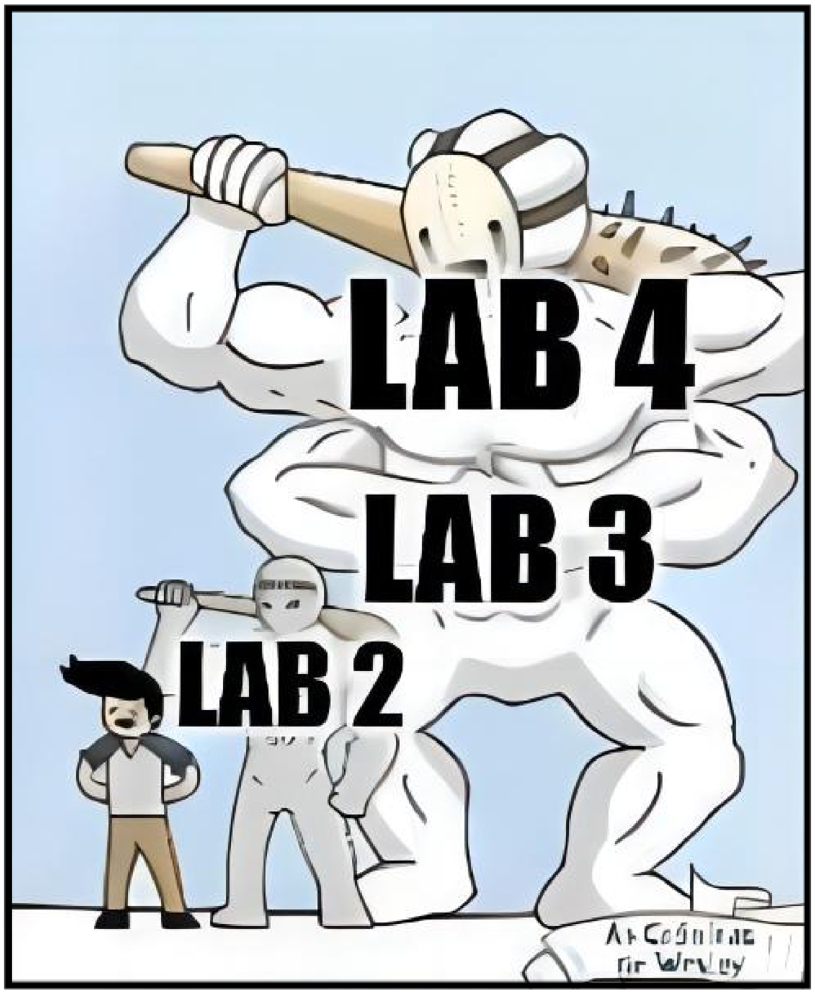
ECE4179 - Lab 3
Multi-Layer Perceptrons (MLP)



Contents

Academic integrity and the Use of Generative AI

Academic integrity

Every lab submission will be screened for any collusion and/or plagiarism. Breaches of academic integrity will be investigated thoroughly and may result in a zero for the assessment and interviews with the plagiarism officers at Monash University.

Lab Instructions and the Use of Generative AI

- You should use Matplotlib to display images and any intermediate results.
- You may use a generative AI tool or coding assistance. We recommend understanding the concepts and coding as much as possible, as you may be performing hand calculations during the final exam. If you use generative AI tools, please indicate the prompts you used and explain in detail any changes you made to modify the code to complete the assignment.
- Please refrain from using Generative AI in preparing your report, as the purpose of this lab is to assess your understanding of deep learning concepts.

Late submission policy and grading guidelines

Late submissions

Late lab submission will incur a penalty of 10% for each day late. That is, with one day delay, the maximum mark you can get from the lab is 90% of the total lab mark, so if you score 95%, we will (sadly) give you 90%. Labs submitted with more than a week delay will not be assessed. Please apply for special consideration for late submission as soon as possible (*e.g.*, documented serious illness). You can do this by applying through Monash University's special consideration application website.

Lab Grading

This lab is worth 5/100 (5 out of 100) marks of the entire unit, and it is made up of two components:

- Code 60% and discussion submission 30% (.ipynb file)
- Lab quiz - 10%

Lab attendance is not compulsory. The lab sessions are there for you to get help.

For the .ipynb submission, there are a number of tasks. Each task will have coding components and a discussion component. A task is only considered complete if you can demonstrate a working program and show an understanding of the underlying concepts. Note that later tasks should reuse code from earlier tasks.

There are 3 tasks. Each task is worth one third of the lab mark (1.5% of the unit total each).

The lab quiz will require an understanding of the lab and will only be accessible from Saturday morning the second lab week to the following Monday night. The lab quiz is only accessible after you have submitted your lab notebook. More information can be found in the lab tile on Moodle. Please read the instructions carefully before attempting the lab quiz.

This lab is about understanding and applying PyTorch-Lightning to simple multi-layer perceptron tasks. By the end of the lab, you will have developed simple MLPs for binary and multi class classification tasks. The following are the three tasks that you should complete.

- Task 1: Perform binary classification using a shallow MLP. This section will explore PyTorch data-loaders and other functionalities. Features for a COVID-19 chest x-ray dataset will be extracted via a pre-trained CNN and you will implement MLP to perform classification
- Task 2: Analyse the understand the implications of the MLP and how it performs on medical data. This will involve analysing the results of a binary confusion matrix and its performance metrics.
- Task 3: Train a two-layer MLP for the FashionMNIST dataset to perform multiclass classification.

The learning outcomes for this lab are:

- Familiarising yourself with PyTorch and PyTorch-Lightning
- Understanding implementation of MLPs and network training.
- Applying CNNs to extract features from image data
- Analysing and describing accuracy and loss plots.
- Understanding the implications of different MLP results.

Note: Most of the lab does not require you to use the Numpy library. You will be using Pytorch/Pytorch-Lightning in-built methods to create your tensors instead of Numpy. Follow the Pytorch/Pytorch-Lightning videos for more information.

Introduction.

The fundamentals of deep learning models start off with understanding multilayer perceptrons and the training processes required. There are additional hyper-parameters that we can choose and tune, such as the type of optimizers, learning rate, activation functions, and model architecture to name a few. The ability of activation functions to model non-linearities is what allows MLPs to be able to generalise better than traditional machine learning models.

In this lab, you will learn how to code MLPs with PyTorch and PyTorch-lightning frameworks. Tasks 1 and 2 takes you the step by step process of training a MLP based on extracted features from a pre-trained CNN. Task 3 contains a different dataset where you will apply MLP to a multi-class classification problem.

The submission deadline is 11st of September (Monday) 9:30 AM AEST.

It is recommended that you go through the following videos/documents prior to attending your lab 3:

1. Begin by reading through this document. This document contains all the relevant information for lab 3
2. Follow the lecture series by Luke Ditria on PyTorch/PyTorch-Lightning
3. Watch the introductory video for this lab
4. You may choose to use GoogleColab for this lab so that you can utilise the free GPU provided. [1]

In the lab and in your own time, you will be completing the lab 3 notebook by going through this document and the provided notebooks.

[1] You can choose not to use GoogleColab if you have your own GPU or if you prefer to use your own CPU. It is not as necessary for this lab but it will be useful for the next lab and assignment.

Task 1: MLP for classifying COVID-19 x-ray images

This section involves developing and evaluating a shallow neural network on extracted convolutional features from the COVID-19 X-ray image dataset, You will use a pre-trained CNN to extract feature maps from raw images and train an MLP in order to perform binary classification using the feature maps. The binary classes for this dataset are lungs that are either functioning normally or have pneumonia.

- 1.1 Data: Create custom dataset and dataloaders
- 1.2 Custom Class Creation based on ImageFolder
- 1.3 Model: Design Shallow MLP model
- 1.4 Train & Evaluate: Train and evaluate model's performance

For this dataset, you start off with 64x64 images. We will provide you with a pre-trained convolutional neural network (CNN)[2] that will extract the features for you. The features are basic combination of shapes from the original input image and these are higher level features (compared to just pixel by pixel analysis). Your main task is to use the extracted features to train a MLP to classify the COVID-19 images.

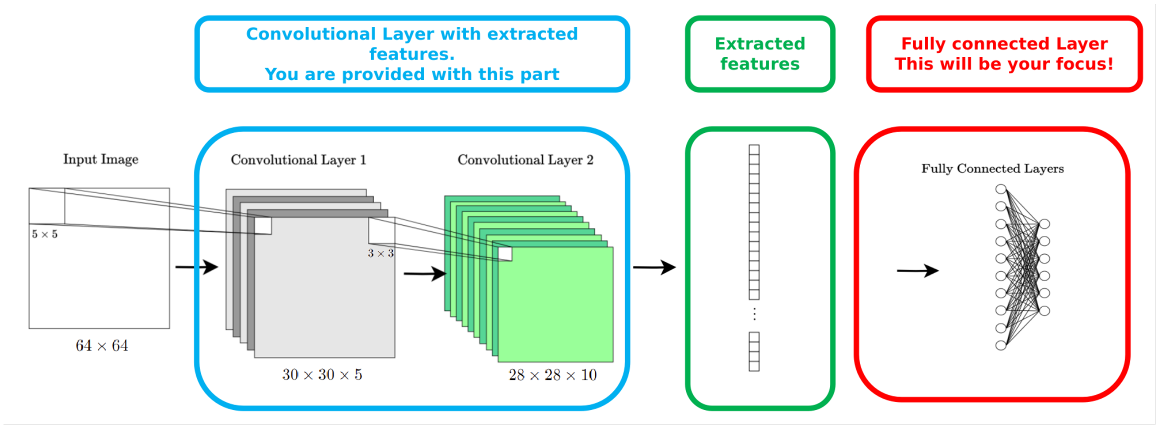


Figure 1: End to end process to classify the COVID-19 images

In terms of the MLP you will be training, the specific details of the model (*i.e.* hyperparameters) will be provided within the notebook. The general structure will look like:

$$\mathbb{R}^{7840} \ni \boldsymbol{x} \rightarrow \text{fc1} : \text{Linear}(7840 \times \text{n}) \rightarrow \text{ReLU} \rightarrow \text{fc2} : \text{Linear}(\text{n} \times 2) = \hat{\boldsymbol{y}},$$

where n represents the number of neurons in the hidden layer.

[2] We will be covering CNNs in more detail for lab 4, but for now you can think of CNN as looking at adjacent pixels within an image to extract information. This is done across the whole image

Remember, the reason for the high dimensional space (7840) is because MLP only takes in one-dimensional data, hence we need to reshape the features from Convolutional Layer 2 in Figure 1 to one dimensional data. After constructing the MLP model, you can train the model using the *Trainer* constructor as it has useful built-in methods such as tensorboard logging, model checkpointing, training and validation loop, early-stopping. [3]

Task 2: Anaylse and Visualisation

In this task, you will write code to plot out and analyse:

- 2.1 Loss, accuracy and a few predictions with their original images and feature maps
- 2.2 Confusion Matrix

For the confusion matrix, you will write your own binary confusion matrix and analyse the implications stemming from your model predictions. Detailed explanation and instruction of how to construct a confusion matrix can be found in the lab3 notebook. With the confusion matrix, you are able to analyse and discuss the classification performance of the MLP.

[3] Make sure to check out the PyTorch videos to understand these concepts

Task 3: MLP for multiclass classification of the FasionMNIST dataset

In this task, you will be training a model to perform multi class classification. The first few steps in this task aims to setup a dataloader class for the FashionMNIST data. The Fashion-MNIST is a dataset of Zalando's article images—consisting of a training set of 60,000 examples and a test set of 10,000 examples. The images are 28 × 28 grayscale images (so only one colour channel) with 10 classes.

Task 3 is broken down into the following sub tasks:

- 3.1 Define transforms and create a custom dataset
- 3.2 Design a two hidden layer MLP
- 3.3 Train & Evaluate: Train and evaluate model's performance
- 3.4 Visualization: Analysis and Visualization

Instead of directly using a CNN to extract features, you will be using MLP straight away at the input layer. You will reshape the input into one-dimension, and pass the inputs through the MLP which contain two hidden layers. At the output layer, you will have one neuron for each class.

$$\mathbb{R}^{784} \quad \ni \quad \boldsymbol{x} \quad \rightarrow \text{fc1} : \text{Linear}(784 \times n_1) \rightarrow \text{ReLU} \rightarrow \\ \text{fc2} : \text{Linear}(n_1 \times n_2) \rightarrow \text{ReLU} \rightarrow \text{fc3} : \text{Linear}(n_2 \times 10) = \hat{\boldsymbol{y}}$$

Hopefully this lab has given you insight in training a simple model via the PyTorch-Lightning framework, and provided realistic scenarios in which optimizing hyper-parameters and considering data augmentation is required for any deep learning problem!

Credits

Lab 3 could not have been completed without the following people:

- Jing, Mingyi, Haoyang, Himashi, Aaron and Grace for writing up Lab 3
- Mehrtash for reviewing
- Yasir for the meme