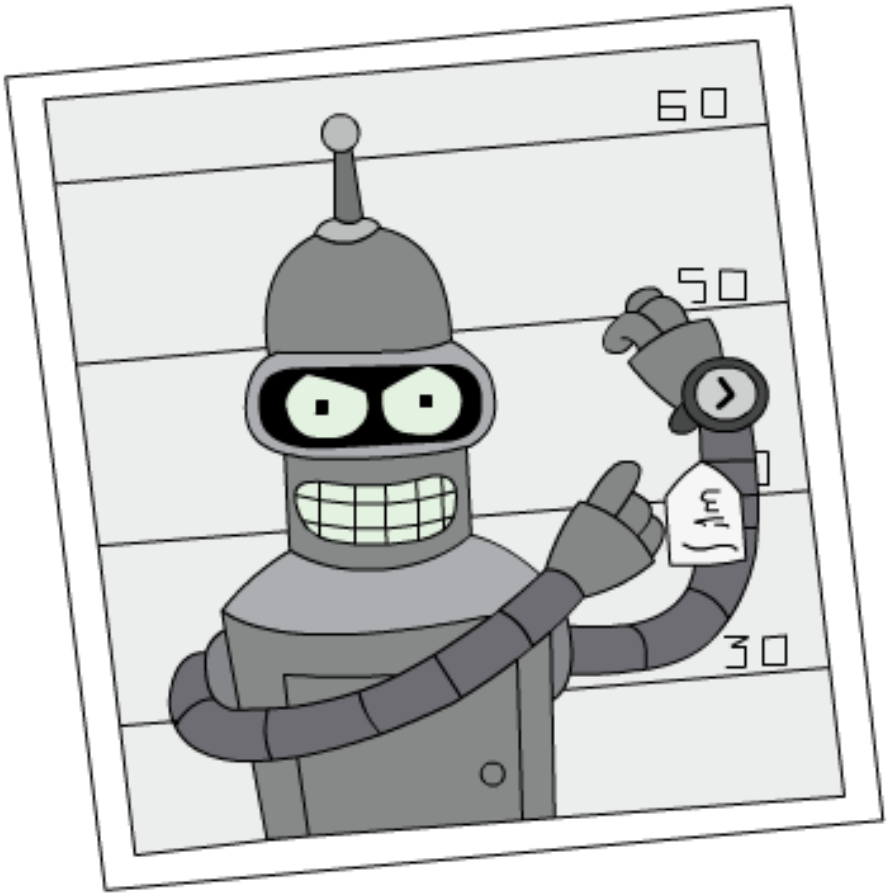
Python Basics

ECE4179/5179/6179: Neural networks and deep learning
Lab1 (Weeks 1,2)



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 along with interviews with the plagiarism officers at Monash University.

Late submission. Late submission of the lab will incur a penalty of 10% for each day late. That is with one day delay, the maximum mark you can get from the assignment is 9 out of 10, so if you score 9.5, we will (sadly) give you 9. 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).

Each lab is worth 5% and there are a number of sections and tasks with their own weighting. 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.

This lab is about understanding Python basics and for using some common Python libraries. At the end of the lab, you will know how to apply your Python knowledge to a maths application. The following are the three sections that you should complete.

- Section 1: Understanding Python basics. This section will explore the fundamentals of Python programming which includes both procedural programming and object orientated programming (OOP)
- Section 2: Using libraries. The Numpy and Matploblib libraries will be used for this section. These are common libraries used for data science within Python to manipulate tensors and for plotting.
- Section 3: Applying your python knowledge. This section will apply what you have learnt so far from the earlier sections in order to complete two tasks. You will investigate how to numerically estimate π and understand the expansion of dimensionality.

The learning outcomes for this lab are:

- Familiarising yourself with Python and Jupyterlab
- Understanding the idea of reproducibility (seed, version requirements)
- Using essential libraries such as Numpy and Matplotlib
- Applying your Python knowledge to a basic mathematical application
- Analysing and describing results
- Understanding the basics of object oriented programming (OOP)

Introduction.

Python will be the programming language used for this unit. There is an installation video of Conda and JupyterLab available on Moodle based on a Windows system. If you use MAC or Linux, the steps should be quite similar, but you will have to select the correct OS / settings when setting up your environment. There is also a video that demonstrates how to use JupyterLab (similar to Jupyter Notebook) which has a Jupyter notebook called Python_Basics.ipynb which you can go through along with the video to understand the Python basics. After being familiar with JupyterLab, you can start on Lab1.ipynb and have it submitted by 7th of August (Sunday) 4:30 PM AEST.

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

1. Begin by reading through this document. This document contains all the relevant information for lab 1
2. Watch the installation video for Conda and JupyterLab
3. Watch both Python Basics video with JupyterLab

In the lab and in your own time, you will be completing the lab 1 notebook by going through this document and the provided notebook

Section 1: Python Basics (out of 25%)

This section is a gentle introduction to the syntax for Python. It contains the following tasks that need to be completed:

- 1.1 Data types
- 1.2 Conditional statements
- 1.3 Loops
- 1.4 Random number generator and writing functions
- 1.5 Classes

Make sure you have a solid understanding of each of these tasks before proceeding to the next section.

Section 2: Using libraries (out of 25%)

In this section, we utilise two basic libraries: Numpy and Matploblib. Numpy is a way for us to manipulate high-dimensional arrays (tensors). There are some caveats with using Numpy which will be explored in the lab tasks itself. Matplotlib is the main plotting library that we will utilise throughout this course, and it is sometimes viewed as a Python version of Matlab's plotting tools.

Section 3: Applying your python knowledge (out of 50%)

In the following tasks, you will apply your knowledge of Python to a basic mathematical problem.

1. In the first task, you will estimate the value of π via uniform random sampling by using the area of a circle equation, namely:

$$\text{Area} = \pi \times r^2 \tag{1}$$

   Using uniform random sampling for pairs of values (x_1, x_2) between [0,1), we are able to calculate the probability that a pair of value lies within the unit circle by using the euclidean distance measure:

$$\text{Euclidean distance measure} = \sqrt{x_1^2 + x_2^2} \tag{2}$$

   By using the euclidean distance, we can see how many points lie within the unit circle - hence giving us the *Area of circle*. We can rearrange equation 1 to solve for π

$$\pi = \frac{Area}{r^2} \times 4 \tag{3}$$

   Note the multiplication by 4 is because we only sampled from one quarter of the circle

2. The second task is to understand the implications of high dimensional space by calculating the volume of a N-dimensional ball. We can calculate the volume of an N-dimensional ball by using the following equation:

$$V_n(r) = \frac{\pi^{\frac{n}{2}}}{\Gamma(\frac{n}{2}+1)} r^n \tag{4}$$

   Note: Γ *is the gamma function which is analogous to factorials but it is used for continuous domains*

   In this task, you will be estimating the volume of a N-dimensional ball (steps prescribed within the Jupyter Notebook), and compare the estimation with equation 4 as above.