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 and interviews with the plagiarism officers at Monash University.

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).

Lab Instructions and the Use of Generative AI

- You may use NumPy for array handling, and vectorizing your code (reducing the number of for-loops) is encouraged.
- 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.

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.

Tasks 1 and 2 are both worth 20% each. Tasks 3 and 4 are both worth 30% each.

The lab quiz will require an understanding of the lab and will only be accessible from Friday morning to Sunday night of the same week the lab is due. More information can be found in the lab tile. Please read the instructions carefully before attempting the lab quiz.

Contents

Lab Structure and Learning Outcomes

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 four tasks that you should complete.

- Task 1: Python Basics. This task will explore the fundamentals of Python programming which includes both procedural programming and object-orientated programming (OOP)
- Task 2: Data Analysis and Visualization. The Numpy and Matploblib libraries will be used for this task. These are common libraries used for data Analysis and visualization within Python to manipulate tensors and for plotting.
- Task 3: Wanna play Darts? This task will apply what you have learnt so far from the earlier tasks in order to estimate the value of π by using the analogy of throwing darts.
- Task 4: Thinking inside the Ball! This task will apply what you have learned so far from the earlier tasks to estimate and calculate the volume of an N-dimensional ball of radius one. You will also discuss the implications of scaling to a higher dimensional space.

There will be two python notebooks. The first notebook will contains tasks 1-2, while the second notebook will contain tasks 3-4.

The learning outcomes for this lab are:

- Familiarising yourself with Python and your coding environment
- 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. The week 0 videos also cover VSCode installation with the JupyterNotebook extension. You may choose to use other environments but it is up to you to test the compatibility between your environment of choice and the ones we have provided.

The second part of the installation videos demonstrates how to use the IDE (*i.e.* JupyterLab/VSCode). Three videos have been recorded to introduce Python which will run through 2 Jupyter notebooks called Python_Basics.ipynb and Numpy_Basics.ipynb. After being familiar with JupyterLab, you can start on Lab1.ipynb and have it submitted by the 4th of August (Friday) at 9:30 AM 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 (or VSCode / Google Colab)
3. Watch the Python Basics and Numpy Basics videos

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

Task 1: Understanding Python Basics (out of 20%)

This task 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 Functions
- 1.5 Classes

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

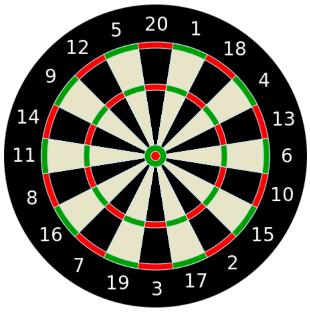
Task 2: Data Analysis and Visualization (out of 20%)

In this task, we utilize two basic libraries:

- 2.1 Numpy
- 2.2 Matplotlib

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 themselves. Matplotlib is the main plotting library that we will utilize throughout this course, and it is sometimes viewed as a Python version of Matlab's plotting tools. This part will cover a handful of ways to display data so the data becomes easily digestable.

Task 3: Wanna play Darts? (out of 30%)



In the following tasks, you will apply your knowledge of Python to a basic mathematical problem, i.e. estimating the value of π by using the analogy of throwing darts.

- The area of a dart board (which is a circle) is given by:

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

- Assuming the dart-hits are uniformly distributed in the square region which encapsulates the dartboard, we can simulate the dart-hits using uniform random sampling, *e.g.*, pairs of values (x_1, x_2) between [0,1) assuming a dart board of unit radius.

- The Euclidean distance from a dart-hit to the center of the dartboard can be found by:

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

- By considering the Euclidean distances, we can count how many points lie within the dartboard and how many lies outside. This can be used to calculate the probability that a dart-hit lies within the dart board hence giving us the *Area of circle*.

- Finally, We can rearrange Equation (1) to solve for π:

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

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

Task 4: Thinking inside the Ball! (out of 30%)

This task is to understand the implications of high-dimensional space by calculating the volume of an 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. We have imported the Γ function for you.

- For dimensions from 2 to 50, estimate the volume of an n-dimensional ball using the method in Task 3. (You can do this by taking the probability of a point being inside the n-dimensional ball of radius one by sampling between [0,1) across the n-dimensions and calculating its Euclidean distance. Afterward, you can calculate the volume by doing $2^n \times$ *fraction of points within a unit circle in high dimensional space*).

- Now, for dimensions from 2 to 50, calculate the actual volume of an n-dimensional ball using Equation (4).

- Finally, compare the two sets of values. Discuss the implications.

Credits

Lab 1 could not have been completed without the following people:

- Mevan and Himashi for writing most of Lab 1
- Grace for assisting with lab 1 write up
- Mehrtash for reviewing
- Yasir for the meme