Tutorial 0: Summary and Practice Problems

CS5590: Foundations of Machine Learning

August 2022

Abstract

This document contains a brief summary of the discussion of tutorial #0 (Aug 3, 2022). It also contains some basic coding problems for practice purposes. These will not be counted for evaluation.

1 Summary

1.1 Virtual Environments

In this tutorial session, we briefly touched upon creating virtual environments. Virtual environments help one keep their projects clean, and avoid dependency hell. Anaconda is often used to create virtual environments when working in the field of data-science. Installation instructions can be found here.

1.2 Notebooks

Notebooks are an interactive way of writing code. Notebooks allow you to write small chunks of code in "cells" that can be then executed individually in the same environment.

Jupyter Notebook is one of the most popular widely used and popular Notebook environments. One can use `pip install notebook` to install classic Jupyter Notebook on your local machine. One can also run notebooks on the cloud, with the help of Google Colab. Google Colab hosts Jupyter Notebooks on remote VMs in the cloud and gives access to (limited)GPU and compute resources.

1.3 `tmux` (optional)

When a program is executed on the terminal, exiting the terminal kills the program. This can be a problem since your assignments may require you to keep a terminal open for a long time. `tmux` is a tool to help set up persistent terminals, i.e if you exit the terminal, the program is not killed but continues to run in the background.

This is often useful when you run code in a remote server (or when you forgetfully close the terminal that was running your code on your local machine!). For instance, the ssh connection to a remote server can be broken due to poor internet, and your process can be killed. However, if you run your code under a `tmux` 'session', killing the terminal won't kill your program. Further information can be found under the Terminal Multiplexers section of Command-line Environment of MIT's missing semester course. Here is a youtube video demonstration some `tmux` commands. (There is a `screen` command for a similar purpose.)

2 Some Numpy and Matplotlib Practice Problems

In the tutorial, we demonstrated some functions of `numpy` and `matplotlib` packages. The following questions are intended to showcase use of basic `numpy` and `matplotlib` functions. All of these can be solved in around 4-5 lines of code. Please avoid using native Python Constructs such as for loops, while loops etc. You may want to refer to numpy's guide on indexing to solve some of these problems.

1. Import the `numpy` package under the name np. Print it's version information. Import `pyplot` from `matplotlib` package under the name plt.
2. Create an array of size 10 with all zeros (integer data type).
3. Create an array 'a' with elements [7,12,17,...57]. Reverse this vector and set every 3$^{\text{rd}}$ element of the reversed vector equal to 14.

   Then using matplotlib, plot a curve with x values as [1, 2, ..., 11] and y values as entries of the 'a' vector.

4. Create a random 1D vector of length 500,000 and increase every alternating element by 5 with and without using loops. Compare the time taken in each method.

5. Create a 2D Matrix of shape (5,5) with all ones and Double its Diagonal elements without for loops. Hint: use `np.eye()`

   Then using an appropriate function from matplotlib (like imshow), plot this 2D matrix as an image.

6. Create 2D Matrices A, and B, composed of random elements of size 5*4 and 4*6.

   - Perform matrix multiplication and print the resulting matrix (say C) onto the terminal.
   - Reshape B into size 8*3.
   - Flatten C into a 1D vector without using `np.reshape()`.
   - To every row of A, add the corresponding row's mean.

7. Create a random vector with entries sampled from a standard normal distribution. Find the indices of the maximum and minimum elements without using a `for` loop. Repeat this exercise with entries of the random vector sampled from a uniform distribution between 0 and 1.

   In both these cases, plot a scatter plot and a histogram with the random vector.

8. Create a 2D matrix A of size 7*7. Sort the rows according to increasing order of elements in the 3rd column. Find the trace, and determinant of the resulting matrix. Compute $B = A^T A$ and find its inverse i.e. B^{-1}.