Machine Learning for Molecular Engineering

Problem Set 0

Date: January 31, 2022
Due: February 11, 2022 (suggested)

Instructions Please complete this problem set using the following template. This is an ungraded problem set and you do not need to submit anything for it. You are encouraged to ask for clarification on Piazza if you find any of the question statements unclear or ambiguous; chances are, another student feels the same way!

Background

This problem set introduces you to some of the most important computational tools that we will use in this course. We will show you how to use Google Colab to run your code on the cloud and to save your work. We will also show you the package `numpy` which is particularly important for array manipulation and vector operations early in this course. This problem set is ungraded and mostly for you to practice using these tools.

Part 1: Introduction to Colab

Google Colab is a freely available tool that allows you to write and execute Python code in your browser, using a CPU or a GPU from Google Cloud compute as the computational engine. It is easier to configure and use than setting up your own local environment and running your code locally, but potentially more powerful since you have access to computational resources beyond those on your laptop. For a brief overview of how to use Colab, work through Google's official tutorial here.

To use Google Colab, open the template notebook from here and select "Save a copy in Drive" from the File menu. This will automatically save the notebook in your Google Drive. If you want to download the notebook, you can download the `*.ipynb` file from the File menu to upload to Canvas.

Once you have successfully opened the Colab notebook, click the Connect button on the right-hand side to connect to a CPU. You can then run any cell in the notebbok by pressing Shift+Enter. Follow the instructions in Part 1 of the notebook to familiarize yourself with Colab's functionality.

We strongly recommend you use Colab for all your computing needs for this class. However, you are also allowed to complete problem sets on your local compute environment. Just make sure any notebook you develop locally can also run on Colab.

Part 2: Introduction to numpy

This portion of the problem set introduces you to the library `numpy`, one of the most popular libraries for scientific computing and array manipulation in Python. The central object of `numpy` is the `numpy.ndarray`, which allows you to store multidimensional lists as vectorized arrays. `numpy` also has a wide variety of built-in functions to perform computations on these arrays or on regular

Python lists; these range from simple operations like transposing and reshaping arrays to more complicated operations like dot products, eigenvalue computation, exponentials and logistics, and more.

There are a number of important differences between `numpy` arrays and Python lists to keep in mind:

1. `numpy` arrays have fixed length or shape, while Python lists can change length. There are a number of operations that allow you to generate a new `numpy` array, but they are slower and more expensive than the corresponding Python operations.

2. In contrast, `numpy` vectorized operations (i.e. operations that use `numpy` functions as opposed to for loops or indexing) are implemented behind-the-scenes in optimized and pre-compiled C code. As a result, using a `numpy` function will almost always be the fastest way to execute a given computation, as compared to any operation on Python lists. It is important to become familiar with standard `numpy` functions so you can most effectively use the library.

3. `numpy` requires that all elements of a given list have the same type. However, this type can be as general as an object, so this is not a very strong limitation in practice.

Part 2 of the template notebook will now walk you through a tutorial of using some basic important `numpy` functionality. If you would like more information about `numpy`, you can look at the quickstart documentation here or at the full API reference here. Please ensure you understand all the functions in the tutorial, since many of them will be useful for your problem sets.