Data Science for AAE

AAE 718 – Summer 2024

Worksheet 2

Numpy

1 Reading

1.1 Python Data Science Handbook

- Chapter 2 – Particular attention to arrays, broadcasting and masks

2 Daily Goals

- Overview of numpy
  - Arrays
  - Slicing
  - Masks
- Understand vectorization
- Some basics plotting
- Arrays vs Series vs Dataframes

3 numpy

Numpy is a collection of methods for using arrays. An array is like a list except that all the entries of the array must have the same type. This may seem like a limitation, but it turns out to be incredibly important. This limitation allows us to *vectorize* computations which is an extremely fast operation (you'll see how fast in the homework).

Numpy is used widely in industry for numerical computation. It's arguably the most important Python library. Here is a good quickstart, that's more in-depth than this worksheet `https://numpy.org/devdocs/user/quickstart.html`.

The best explanation of Numpy arrays I've read is the second chapter of our book. This chapter is quite long, but it's a fascinating read. It details exactly what an array is, why they are faster, how Python treats variables and lists. You should really consider reading this chapter.

In particular, This section discusses slicing and other array manipulation techniques. In particular you should understand how to build an array, and how array slicing works.

Problem 1 (5 pt) The code `np.random.rand(shape)` returns an array of shape with random values between 0 and 1. Describe how to take this output and modify it so that the values are between a and b for numbers a and b. Your answer should be words, not code. It should have math in it.

Problem 2 (5 pt) Write a function called `problem_02` with three inputs `a, b, c` (all numbers with $a < c < b$) and returns a number.

Create an array with 100 random entries between a and b. Use a mask to return the number of entries in this array less than c.

Problem 3 (5 pt) Write a function called `problem_03` with no inputs and returns rows $0, 2$ and columns $1, 2, 3, 4$ of a random 5×5 array.

Problem 4 (5 pt) Write a function called `problem_04` with a single number input, N, that returns an $N \times N$ array that has ones one the edges and zeros inside.

For example, if $N = 4$ then the returned array is

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

---

4 Vectorization

Loops in Python are slow. Vectorization is a way to avoid using loops to apply operations to arrays. This section describes vectorization quite well..

Problem 5 (5 pt) Write a function called `least_squares_error` that takes two arrays of the same size as inputs and returns the sum of the squares of the differences.

For example, if the inputs are $x = (x_i)$ and $y = (y_i)$ then the returned value will be $\sum(x_i - y_i)^2$.

You should not use any loops in this exercise.

Problem 6 (5 pt) Write a function called `normalized_random` with a single number input and returns a random $N \times N$ array where each row sums to one.

You should use vectorization for this exercise.

---

5 Basic Plotting

For this we use the `matplotlib` module. At it's core, the `plot` function is incredibly simple. If you have two arrays X and Y, where X is the domain and Y is the range and $|X| = |Y|$, then `plot(X,Y)` will create a graph. For example,

```
import numpy as np
import matplotlib.pyplot as plt

X = np.linspace(-10,10,1000)
Y = X**2

# Apparently figsize is not a keyword in base plot
plt.plot(X,Y) # , figsize=(10,10))
plt.savefig(r"tmp.png")
```

You should see a graph of x^2. This works by plotting a sequence of points and connecting the dots. There are a ton of options and additional plotting options which we'll deal with later. This should have also saved a PNG with the figure.

Problem 7 (10 pt) For this problem you'll be making a few graphs. I'll give you domain and range, you plot. Two functions on the same bullet go on the same axis. Bonus points for creating a key on each graph.

- $0 \leq x \leq 10$ $f(x) = x^2$
- $-\pi \leq x \leq \pi$, $f(x) = \sin(x)$, $g(x) = \cos(x)$
- $0 \leq x \leq 5$, $f(x) = \arctan(x)$, $g(x) = \frac{1}{1+e^{-x}}$

Include these graphs in your PDF. Be sure to label them.

Problem 8 (10 pt) Let's analyze the speed of sorting a list vs an array. Call your function `sort_times` you don't need any inputs or outputs.

To time a block of code use the following syntax

```
import time

start = time.time()
CODE TO TIME HERE
end = time.time()

total_time = end-start
```

This will give you the time it takes for `CODE TO TIME HERE` to run.

We are going to repeat the following process 10 times.

1. Create a list containing 10,000,000 random integers.
2. Copy this to an array. You can swap these steps, create an array and copy to a list.
3. Time how long it takes to sort the array, `np.sort(a)`
4. Time how long it takes to sort the list, `L.sort()`
5. Save these times in a list

Create a table showing your results. Discuss the difference in timing.

Repeat everything, except with random numbers from 0 to 1. What do you notice is different?

Discuss your findings using words and sentences. Don't write a novel, be concise.

I recommend starting with 100 random integers rather than 10,000,000. It'll save you so much testing time.