Python Tutorial – Part I: Introduction

Mark A. Austin

University of Maryland

austin@umd.edu
ENCE 201, Fall Semester 2023

September 3, 2023

Overview

Part 3

First Program

(Evaluate and Plot Sigmoid Function)

Problem Desription

Problem Description

In neural network models, the sigmoid function:

$$\sigma(x) = \left[\frac{1}{1+e^{-x}}\right]. \tag{3}$$

serves as an activation. Our first program evaluates and plots $\sigma(x)$ over the range x $\in [-10, 10]$.

Running the Program

From the terminal window, simply type:

```
prompt >> python3 TestSigmoidFunction.py
```

Evaluate and Plot Sigmoid Function

The Python interpreter/compiler will complain if one or more of the required packages (e.g., matplotlib) are missing.

Use pip to install missing Python Packages

The standard package-management system used to install and manage software packages is called pip (or pip3).

Example: And installation is easy!

```
prompt >> pip3 install numpy
prompt >> pip3 install matplotlib
```

To get a list of installed packages:

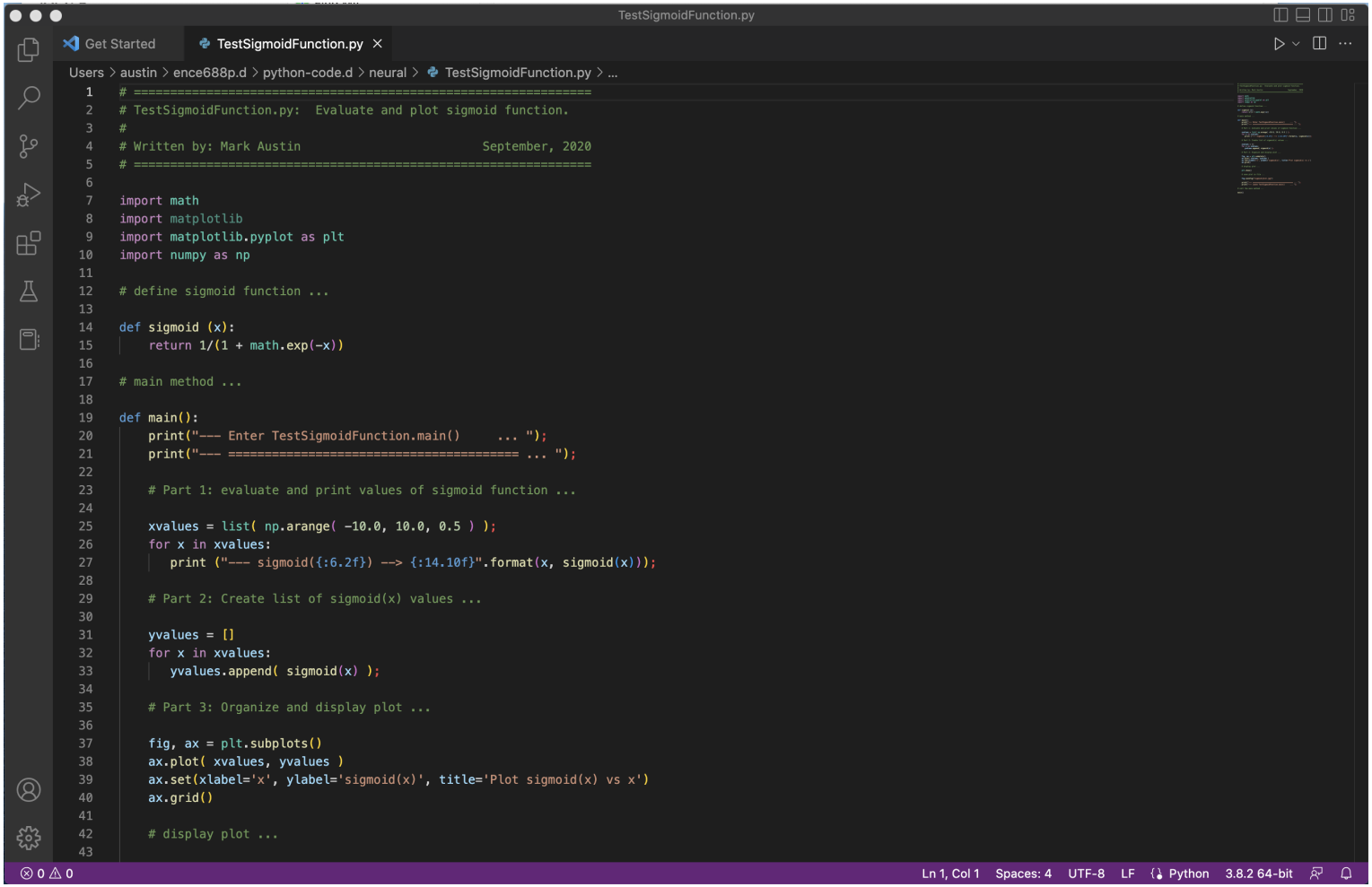
```
prompt >> pip3 list
```

Evaluate and Plot Sigmoid Function

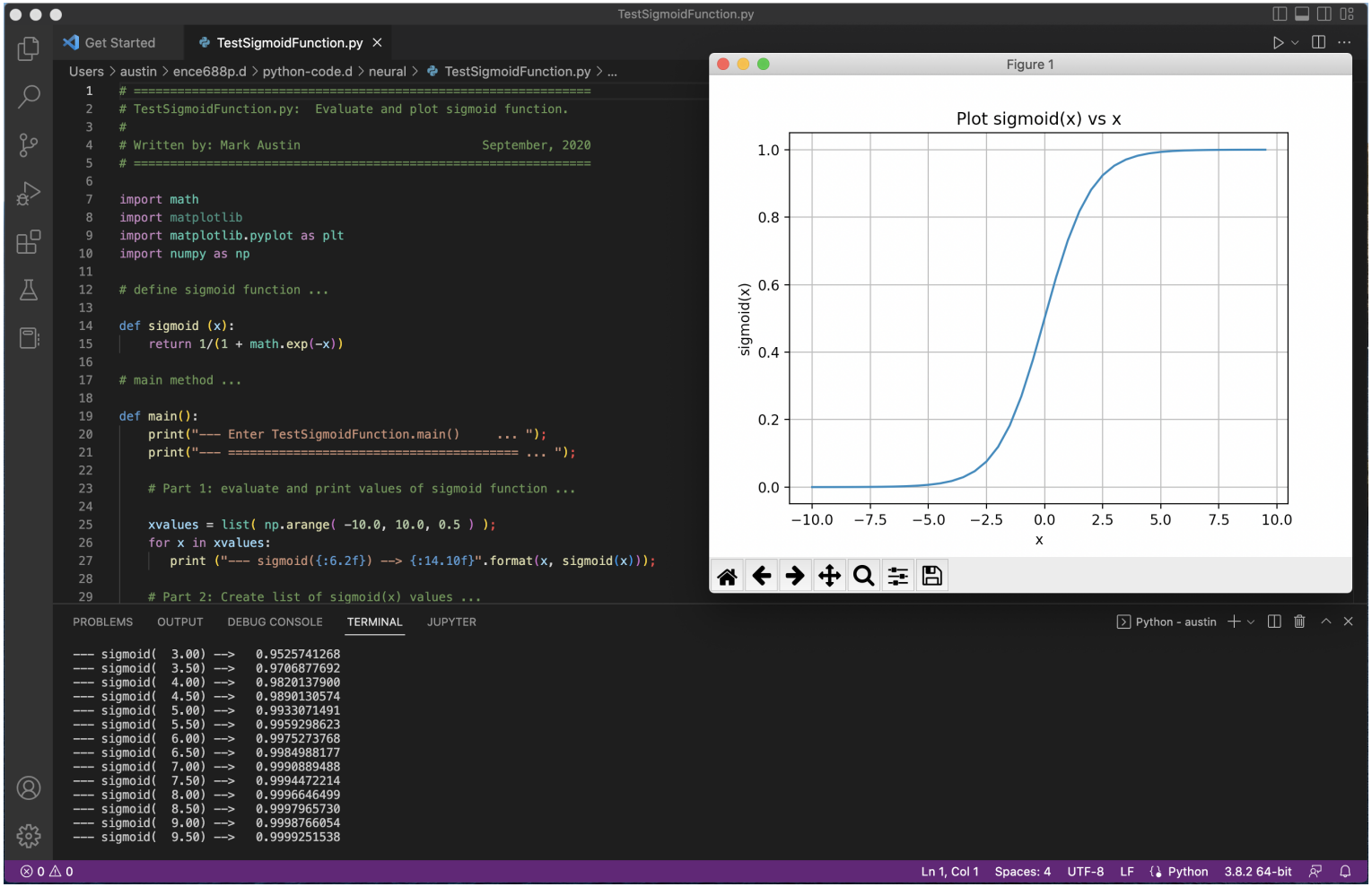
Abbreviated Output:

```
Package                  Version
------------------------ ---------
....
jupyter                  1.0.0
Keras                    2.4.3
....
matplotlib               3.4.1
....
numpy                    1.19.5
....
pandas                   1.1.5
....
scikit-learn             0.24.2
scipy                    1.6.2
....
sklearn                  0.0
```

Program Source Code in Visual Studio Code



Program Source Code + Output in Visual Studio Code



Program Source Code

```
1   # ========================================================
2   # TestSigmoidFunction.py: Evaluate/plot sigmoid function.
3   #
4   # Written by: Mark Austin                    September, 2020
5   # ========================================================
6
7   import math
8   import matplotlib
9   import matplotlib.pyplot as plt
10  import numpy as np
11
12  # define sigmoid function ...
13
14  def sigmoid (x):
15      return 1/(1 + math.exp(-x))
16
17  # main method ...
18
19  def main():
20      print("--- Enter TestSigmoidFunction.main() ...");
21      print("--- ================================ ...");
22
23      # Part 1: Evaluate and print sigmoid function
24
25      xvalues = list( np.arange( -10.0, 10.0, 0.5 ) );
26      for x in xvalues:
27         print ("--- sigmoid({:6.2f}) --> {:14.10f}".format(x, sigmoid(x)));
28
29      # Part 2: Create list of sigmoid(x) values ...
```

Program Source Code

```
29        # Part 2: Create list of sigmoid(x) values ...
30
31        yvalues = []
32        for x in xvalues:
33           yvalues.append( sigmoid(x) );
34
35        # Part 3: Organize and display plot ...
36
37        fig, ax = plt.subplots()
38        ax.plot( xvalues, yvalues )
39        ax.set(xlabel='x', ylabel='sigmoid(x)',
40               title='Plot sigmoid(x) vs x')
41        ax.grid()
42
43        # display and save plot ...
44
45        plt.show()
46
47        fig.savefig("sigmoid-plot.jpg")
48
49        print("--- ================================ ...");
50        print("--- Leave TestSigmoidFunction.main() ...");
51
52    # call the main method ...
53
54    main()
```

Program Source Code

Points to Note:

- Line comment statements begin with the # character.
- Lines 7-10 import the math, matplotlib, matplotlib.pyplot and numpy modules to the program, and make the functions therein available.
- Functions are the primary method of code organization and reuse in Python.
- User-defined functions are declared with the `def` keyword. A function contains a block of code with an optional `return` keyword.
- Lines 13-14 evaluate and return the sigmoid function.
- Use of the second function, `main()`, is a carry over from programming with C, where all programs begin their execution in `main()`. Its use in Python is optional.

Program Source Code

Points to Note (continued):

- Line 25 creates a list of x coordinates. The numpy function `np.arange()` covers $[-10, 10]$ in increments of 0.5.
- Lines 26-27 systematically traverse the elements of `xvalues`, and compute and print the corresponding values of the `sigmoid()` function.
- Line 27 formats and prints the output. The specification `{:6.2}f` means that the output should be printed as a floating point number, six characters wide, and with two decimal places of accuracy to the right of the decimal point.
- Lines 31-33 traverse the elements of `xvalues`, and systematically assemble a list of sigmoid function `yvalues`.
- Lines 37-47 format a plot of `yvalues` vs `xvalues`, and save to `sigmoid-plot.jpg`.