Assignment I: Latex and Python Basics

Exercises in Machine Learning (190.013), SS2022
Stefan Nehl[1]

[1]*stefan-christopher.nehl@stud.unileoben.ac.at, MNr: 00935188,* Montanuniversität Leoben, Austria

February 23, 2022

In the first assignment, I had to create an algorithm for calculating the Fibonacci sequence and plotting the values from 1 to 30. Furthermore, I made a report in Latex, using the template provided by the Institute and submitted the results.

1 Introduction

The Fibonacci sequence was developed by the Italian mathematician Leonardo Pisano Bigollo, also known as Leonardo Fibonacci (1170 - 1250). He created a number sequence where each number is the sum of the two previous numbers (Canaan, Garai, and Daya, 2011). The sequence starts with zero and goes on to a specified number, in my assignment to 30. Example of the Fibonacci sequence: 0, 1, 2, 3, 5, 8, 13, ... The mathematical equation for the Fibonacci sequence is displayed here:

$$f(y) = \begin{cases} 0 & n = 0 \\ 1 & n = 1 \\ f(n) = f(n-1) + f(n-2) & n > 1 \end{cases}$$

(Canaan, Garai, and Daya, 2011)

2 Methods

For the results, I followed the following steps:

- Installed PyCharm
- Created a project with a virtual environment
- Installed packages, *matplotlib* and *numpy*
- Developed the algorithm
- Plotted the results

PyCharm is already installed on my pc, and the creation of a project worked without any issues. Furthermore, installing the packages *matplotlib* and *numpy* with PIP also worked. Because of this, I will focus in this report on the development of the algorithm

2.1 Development of the algorithm

First, I created a class *Fibonacci* to implement the needed calculations. I defined a function called *get_fibonacci_numbers(self, n)*. The parameter *self* is the object of the class and *n* the range of the Fibonacci calculation. The next step was to initialize the array, *fibonacci_numbers* of size *n* with zeros. The package *Numpy* provides a function for this operation called *.zeros(n)*, where *n* defines the size of the array. Next, I handled the base case for $n < 0$, $n = 0$ and $n = 1$ with *if* and *elif*. The calculation of the Fibonacci $n > 1$ happens in the *for-loop*. I iterated over the complete *fibonacci_numbers* array with the variable *x* and implemented different cases. If $x = 0$, the calculated value is 1, if $x = 1$ the calculated value is 1. I need this to calculate the Fibonacci for the third value. Otherwise, the access to the array position *x - 2* would fail. The last step in the *for-loop* is to handle the Fibonacci calculation itself. This calculation happens by accessing the values *x - 1* and *x - 2* in the *fibonacci_numbers* array and summing them up. The newly calculated value is then set to the position *x* in the *fibonacci_numbers* array. The *print* function in line number 26 only displays the values in the console. This function is not part of this assignment. After the loop, I return the *fibonacci_numbers* array.

2.2 Plotting the results

For plotting the results, I created a plot with the *.figure()* function of *matplotlib*. First, I set the subtitle of this plot to *Fibonacci*, the x-axis label to *Number* and the y-axis label to *Value*. I created an array with the *range()* function from python which contains the values from 1 to 30 and is needed for the x-axis values. Next, is set y-axis values to the results of the Fibonacci calculation, added the values to the plot with the *.plot()* function and displayed the plot with *.show()*.

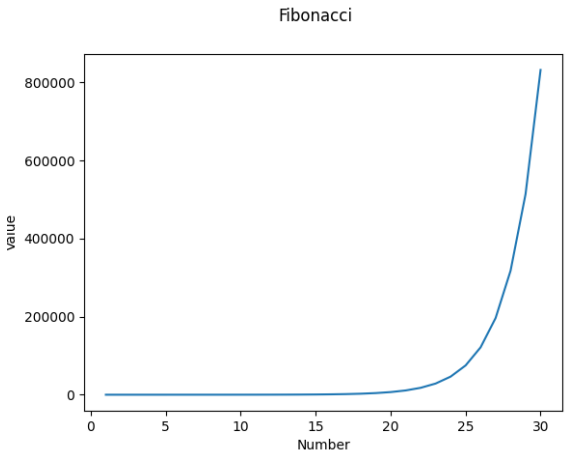

Figure 1: *Illustrates the calculated values for the Fibonacci sequence from 1 to 30.*

3 Results

Figure 1 displays the result of the Fibonacci calculation. The calculated values grow very fast as the numbers for the calculation rise. For example, the value for the number 15 is 610, and for 30, 832040. Table 1 displays the first and last three values of the Fibonacci calculation.

Table 1: *Fibonacci Values*

Number	Calculated Value
1	1
2	1
3	2
...	...
28	317811
29	514229
30	832040

4 Conclusion

The setup for the assignment was straightforward, and no issues appeared in the process. Also, the implementation of the algorithm happened without any problem. However, I had some trouble with creating the report because of the wrong build options in *Texmark*. Changing the build options to *PDFLaTex* solved this issue.

APPENDIX

Fibonacci class:

```
1  import numpy as np
2
3  class Fibonacci:
4
5      def get_fibonacci_numbers(self, n)
           :
6
7          if n < 0:
8              print("Input can't be
                   lower then zero")
9              return 0
10         elif n == 0:
11             return 0
12         elif n == 1:
13             return 1
14
15         fibonacci_numbers = np.zeros(n
               )
16         for x in range(0, n):
17             calculatedValue = -1
18             if x == 0:
19                 calculatedValue = 1
20             elif x == 1:
21                 calculatedValue = 1
22             else:
23                 calculatedValue =
                       fibonacci_numbers[x
                       -1] +
                       fibonacci_numbers[x
                       -2]
24
25             fibonacci_numbers[x] =
                   calculatedValue
26             print('it: ' + str(x + 1)
                   + ' v: ' + str(
                   calculatedValue))
27
28         return fibonacci_numbers
```

Plotting of the values:

```
#imports
from Fibonacci import  Fibonacci
import matplotlib.pyplot as plt


number_of_calculations = 30

fibonacci = Fibonacci()
result = fibonacci.
    get_fibonacci_numbers(
    number_of_calculations)

if type(result) == int:
     print('wrong values')
     quit(0)

fig = plt.figure()
fig.suptitle('Fibonacci')
plt.xlabel('Number')
plt.ylabel('Value')

x = range(1, number_of_calculations+1)
y = result
plt.plot(x, y)
plt.show()

print("Finished")
```

Bibliography

Canaan, C, MS Garai, and M Daya (2011). “All about Fibonacci: A python approach”. In: *World Applied Programming journal, no. April*, p. 72.