January 2024 CSE 220
Online on the Convolution of Linear Time-Invariant Systems
Subsections: A1, A2

October 29, 2024

Suppose you have the stock market prices of MNP company for the last few days. For trend analysis, it is often convenient to work with the moving averages of the stock prices. This process is also called **smoothing**. In stock price analysis, the later days usually have more influence on the future prices. In exponential smoothing, recent data points are given exponentially greater weights than older points. This can be achieved by convolving the data with an impulse response function that decreases exponentially. We can consider only the prices of the last **n** days. Here **n** is called the **window size**. Since the exponentially decaying function rapidly approaches 0, we can neglect the other values or consider as 0. We can define this function as:

$$h[k] = \alpha \times (1 - \alpha)^k \tag{1}$$

We can control the amount of weight we want to assign on the most recent values by controlling the parameter α, where $0 < \alpha < 1$. To be more specific, we put exactly the weight α on the most recent value, then the weight $\alpha \times (1 - \alpha)$ on the value before that and so on. Please refer to the sample I/O for a better understanding.

Your task in this assignment is to perform Exponential Smoothing on stock price data for a given window size **n** and a given value of α. You must make use of convolution to solve this problem. Using Python library functions to determine any of the two averages is strictly prohibited. You should use the classes and functions implemented during your Offline 1 assignment.

I/O Format

You shall need to take stock market prices, the window size and the parameter alpha as inputs and print the results found after performing exponential smoothing on the stock prices.

In the first line, take stock prices as input in the form of **m** comma separated integers. Then in the next line, you should take the moving window size n as the input. Finally take the parameter α as the input.

As for the output, you should print out total **m-n+1** as the outputs. For example, if there are 10 stock prices given and the window size is 3, then you need to print out total $10-3+1 = 8$ values as the output. Please refer to the sample I/O for a better understanding.

A template Python file already containing these codes has been be given. You need not implement I/O.

Sample I/O

Case 1

Input

```
Stock Prices: 10,11,12,9,10,13,15,16,17,18
Window size: 3
Alpha: 0.8
```

Output

$11.68, 9.47, 9.82, 12.29, 14.40, 15.62, 16.64, 17.63$

Explanation

Here we have 10 stock prices and the window size is 3. So we need to determine $10-3+1 = 8$ values. We determine the first output 11.68 by considering the first **3** inputs 10, 11 and 12. Since 12 is the most recent value, we put exactly the weight $\alpha = 0.8$ on it. Then, we put the weight $\alpha \times (1-\alpha) = 0.8 * 0.2 = 0.16$ on 11. The remaining value 10 has the weight $\alpha \times (1-\alpha)^2 = 0.8 * 0.2^2 = 0.032$. By adding these 3 values, we get the first output 11.68.

$$0.8 * 0.2^0 * \mathbf{12} + 0.8 * 0.2^1 * \mathbf{11} + 0.8 * 0.2^2 * \mathbf{10} = 11.68$$

Similarly, to determine the second output 9.47, we need to focus on the next 3 prices 11, 12 and 9.

$$0.8 * 0.2^0 * \mathbf{9} + 0.8 * 0.2^1 * \mathbf{12} + 0.8 * 0.2^2 * \mathbf{11} = 9.472$$

Hints

- Try to relate the following formula of the Convolution Sum to the formula of the moving average. What should be $x[k]$ and how is it related to the stock prices?

$$y[n] = \sum_{k=-\infty}^{\infty} x[k] \times h[n-k]$$

- The impulse response function $h[k]$ is an exponentially decaying function which can be defined as ($0 < \alpha < 1$):

$$h[k] = \alpha \times (1-\alpha)^k$$

  You only need to define $h[k]$ where $k = 0, 1, ..., n-1$ where n is the window size. You can assume $h[k] = 0$ everywhere else.

Marks Distribution

- Taking the input in the given format: **1 Mark**
- Determining the proper **input signal**: **2 Marks**

- Determining the proper **impulse response**: **3 Marks**
- Determining the **results of Exponential Smoothing** correctly: **3 Marks**
- Printing the output in the given format to the console: **1 Mark**