**Custom Moving Average Function for NumPy Arrays**

**Aim:** Implement a moving average function for a numpy array with a specified window size.

import numpy as np

def moving\_average(arr, window\_size, mode='valid'):

"""

Compute the moving average of a 1D NumPy array with a specified window size.

Parameters:

arr (numpy.ndarray): Input array.

window\_size (int): Size of the moving window.

mode (str): One of 'valid', 'same', or 'full'. Determines how edges are treated.

'valid' - Only complete windows are used.

'same' - Output has the same length as the input.

'full' - Compute the moving average for all points, including where windows are incomplete.

Returns:

numpy.ndarray: The array containing the moving averages.

"""

if not isinstance(arr, np.ndarray):

raise TypeError("Input must be a numpy array.")

if window\_size < 1:

raise ValueError("Window size must be at least 1.")

if mode not in ['valid', 'same', 'full']:

raise ValueError("Mode must be 'valid', 'same', or 'full'.")

# Compute the moving average using NumPy's convolution function

return np.convolve(arr, np.ones(window\_size), mode) / window\_size

# Example usage

arr = np.array([1, 2, 3, 4, 5, 6, 7, 8, 9, 10])

window\_size = 3

# Compute moving averages in different modes

moving\_avg\_valid = moving\_average(arr, window\_size, mode='valid')

moving\_avg\_same = moving\_average(arr, window\_size, mode='same')

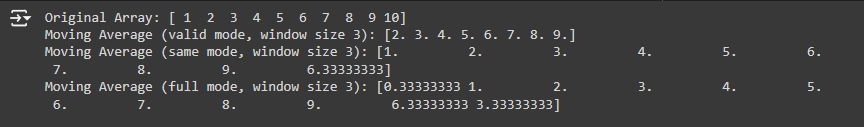
moving\_avg\_full = moving\_average(arr, window\_size, mode='full')

print("Original Array:", arr)

print(f"Moving Average (valid mode, window size {window\_size}):", moving\_avg\_valid)

print(f"Moving Average (same mode, window size {window\_size}):", moving\_avg\_same)

print(f"Moving Average (full mode, window size {window\_size}):", moving\_avg\_full)

****

### Explanation of the mode parameter:

1. 'valid':
   * Only computes the moving average where the window is fully contained within the array.
   * The output will be shorter than the input, specifically by window\_size - 1.
   * Example: If window\_size=3, the output will have len(arr) - window\_size + 1 elements.
2. 'same':
   * The output will have the same length as the input array.
   * It computes the moving average at the edges by partially overlapping windows. For example, at the beginning of the array, it uses fewer values from the window (handling edge cases symmetrically).
3. 'full':
   * Computes the moving average for all possible points, including incomplete windows at the start and end of the array.
   * The output will be larger than the input array, with len(arr) + window\_size - 1 elements.

### Explanation of Outputs:

* **Valid mode**: Only computes averages for full windows.
* **Same mode**: Matches the input array size by averaging even at the edges (where the full window cannot fit).
* **Full mode**: Extends beyond the array's edges, where windows only partially overlap with array elements.