Revision Notes: NumPy - Shallow vs Deep Copy and Array Manipulation

Introduction

This session focuses on understanding the concepts of shallow and deep copies in NumPy arrays, and the operations of array splitting and stacking. These techniques are crucial for efficient data manipulation and transformation in scientific computing and data science.

Key Concepts

Shallow vs Deep Copy

In array and data manipulation, understanding how copies of data are made is crucial due to the implications on memory usage and data integrity.

Shallow Copy

- A shallow copy means creating a new array that references the same data block as the original. Thus, changes in one array reflect in the other.
- In NumPy, whenever you use slicing, it creates a shallow copy of an array. This means both the source and the derived array will share the same data【4:11†source】.

Deep Copy

- A deep copy, in contrast, creates a completely new, independent copy of the data. Changes in the new array do not affect the original array.
- Methods to create deep copies in NumPy include:
  - `A.copy()` : Utilizes NumPy's `copy` method to create a new copy of the data【4:10†source】.
  - `numpy.array(A)` : This function call with the original array will create a deep copy【4:1†source】.

- When manipulating large datasets, especially in image processing, it's efficient to use deep copies to avoid altering the original dataset unintentionally.
- Deep copies are necessary in situations where dataset immutability is important, and shallow copies are used mainly to share data without redundancy【4:7†source】【4:11†source】.

Array Splitting

Array splitting involves dividing an array into multiple sub-arrays. This operation can be vital in data preprocessing or parallel processing tasks.

- **Vertical and Horizontal Splits**: Using functions like `np.vsplit()` and `np.hsplit()`, arrays can be split across different dimensions. This is particularly used in image manipulation where one might want to split images into multiple smaller segments for analysis【4:17†source】.
- **Splitting into Unequal Parts**: Although splitting is generally done equally, customization allows arrays to be divided in user-defined proportions【4:13†source】.

Array Stacking

Array stacking refers to joining arrays along new axes and is used to aggregate data from multiple sources.

- **Stacking Methods**:
  - `np.hstack()`, `np.vstack()`: These functions are used to stack arrays horizontally or vertically. They are equivalent to concatenating along existing axes【4:17†source】.

Application Example: Image Manipulation

During the class, a practical example was discussed involving image manipulation to make two images of birds face each other using these array operations. This real-world application

Conclusion

Understanding NumPy's methodologies for data handling through copying, splitting, and stacking is fundamental for advanced data manipulation in scientific computing. Mastery of these techniques ensures efficient memory usage and the integrity of data across various operations.

This session equipped learners with practical skills essential for high-performance computing tasks using NumPy, demonstrating the importance of array manipulation in big data and image processing contexts【4:5†source】【4:6†source】.