idss_selfstudy_numerical_i_ch_0

January 14, 2021

Introduction to Data Science and Systems

Self-study: Arrays, `numpy` and vectorisation

University of Glasgow - material prepared by John H. Williamson* (adapted to IDSS by BSJ).

$*arg\,min$

---

1 Content:

1.1 1: Why use arrays

- what vectorized computation is
- what numerical arrays are and what they are useful for
- the general categories of array operations
- how images and sounds map onto arrays

1.2 2: Typing and shapes of arrays

- the naming of different types of arrays (vector, matrix, tensor)
- what shape and dtype are
- what axes of an array are and how they are named (row, column, etc.)

1.3 3: Creating, indexing, slicing, joining and rotating

- creating new arrays
- slicing and indexing operations and their syntax
- how to rotate, flip and transpose arrays
- how to split and join arrays and the rules governing this
- boolean arrays and fancy indexing
- swapping, adding dimensions, reshaping and adding dimensions

1.4 4: Arithmetic, broadcasting and aggregation

- scalar and elementwise arithmetic on arrays
- broadcasting rules
- basic aggregation operations like summation, mean, cumulative sum
- sorting and selection like argmax, argsort, find

1.5 5: Nummerical aspects

- how IEEE 754 `float32` and `float64` numbers are represented
- how infinity and NaN are represented, how they occur and how they are used
- what roundoff error is and how it tends to be caused
- how to compare floating point numbers
- what machine epsilon is and how it is defined

1.6 6: Vectorisation

- how to vectorise basic algorithms like summations and elementwise operations
- how to mask elements in vectorised operations
- how to write simple equations as vectorised operations

1.7 Appendix: Numpy Reference

- a list of highly releveant `numpy` functions/features

1.8 Extra resources for this self-study:

- From Python to Numpy
- 100 numpy exercises
- NumPy tutorial
- Introduction to NumPy
- Linear algebra cheat sheet *not actually linear algebra!*