Python Bootcamp

Python 3 and efficient array processing with Numerical Python (NumPy)

Roberto Souza
Assistant Professor
Electrical and Computer Engineering
Schulich School of Engineering

January 2021

UNIVERSITY OF CALGARY

Outline

- Hands-on tutorials
- Python
  - What it is? And why use it?
  - Python 2 versus Python 3
  - Python data types and flow control statements (if/else, for, while)
- Numerical Python (NumPy)
  - NumPy array slicing
- Summary

UNIVERSITY OF CALGARY

Hands-on Python + NumPy tutorials

- https://github.com/rmsouza01/ENEL645
    - **Tutorial 01**: Introduction to Python
    - **Tutorial 02** Introduction to NumPy

What is Python?

- Programming language created in 1991
  - Interpreted:
    - Does not convert code into machine language prior to running
    - Potentially slower performance. Avoid loops at all cost!
  - High-level
    - Strong abstraction from the details of the computer
  - General-purpose
    - Applied on a range of domains: backend web development, scientific computing, data analysis, and **artificial intelligence (AI)**

Why use Python?

- High productivity
- Maintainable code
  - Readable and clean
- Programming paradigms
  - Object-oriented and structured programming
- Compatible with major platforms
- Robust standard and non-standard libraries
  - NumPy, Pandas, OpenCV, Scikit-learn, Scikit-image, etc.
- Large community
  - Fastest growth in past years compared to other languages

Why use Python?

Lasagne

Caffe

TensorFlow

Caffe2

theano

K Keras

PYTORCH

- It is the AI language
- Preferred by data scientists and engineers, which are professions in high-demand

Python 2 versus Python 3

- Python 3 was released in 2008
    - Not backward-compatible with Python 2
    - Supposed to fix design flaws that came with Python 2
- Python 2 was supposed to be discontinued by 2015
    - Companies that had millions of lines written in Python 2 hesitated in making the change right away
    - Support was extended until January 2020, but many still use Python 2

You should use Python 3!

UNIVERSITY OF CALGARY

Python Data Types

- No need to pre-declare variables and their types

Variable type	Description	Syntax example
int	Integer variable	a = 103458
float	Floating point variable	pi = 3.14159265
bool	Boolean variable - True or False	a = False
complex	Complex number variable	c = 2+3j
str	UNICODE string	a = "Example"
list	Heterogeneous list (any type of elements)	lista = [4,'me',1]
tuple	Heterogeneous tuple (values can't change)	tupla = (1,'I',2)
dict	Associative set of values	dic = {'me':1,'you':2}
set	Unordered collection of unique items	a = {5,2,3,1,4}

UNIVERSITY OF CALGARY

Flow Control Statements

- Blocks, such as "if", "for" are delimited by code indentation and not delimiters like "{}" "BEGIN...END“
- Main flow control commands:
  - if/else
  - For
  - While
  - Nested flow control statements (e.g., a loop inside another loop)
- Avoid loops at all cost in Python!

Numerical Python (NumPy)

- *NumPy* is a library that allows efficient multi-dimensional array processing
- It is the basis for scientific computing with Python
  - AI applications are built upon NumPy arrays (*ndarray*)
- NumPy functions are compiled (i.e., faster)

NumPy Slicing

- C/C++ style for:
  - for (i=begin; i < end; i += step) a[i]

- NumPy slicing
  - a[begin:end:step]
    - Begin index is included in the slice
    - End index is not included

1D slicing cheat sheet

Full syntax	Short syntax
a[0:a.size:1]	a[:]
a[0:10:1]	a[:10]
a[0:10:2]	a[:10:2]
a[2:a.size:1]	a[2::]
a[2:a.size:2]	a[2::2]

Summary

- Python is a high productivity language with a rich set of libraries, specially related to AI
- Python 3 is the future
- NumPy array processing style can lead to efficient and elegant code
- Avoid loops at all cost!

Thank you!

UNIVERSITY OF CALGARY