Agenda for the Hands-on Session: 17/1

Julia (1 hour 15 minutes)

- Python/MATLAB like syntax
- Easy syntax; C Speed
- Mitigate two language problem
- Adopted by major Universities and Companies for ML and Computing Research

Installation

Ways to install and use Julia:

1. Easy way: If you have mini/anaconda - Install via `conda install -c conda-forge julia` This would give you Julia v1.1.1
2. Manual Install Download Julia from: https://julialang.org/downloads/ Installation instructions on: https://julialang.org/downloads/platform/ You can use the lastest julia v1.3.1 this way.

- If you like to work on an IDE(MATLAB/PyCharm Style interface), install atom and setup uber-juno from package manager
- Download and install juliapro directly which comes with Juno and commonly used packages.
- If you like to work on Jupyter, get the julia kernel by installing IJulia. Or conda install julia in a conda environment with jupyter.
- For sublime users, install the syntax highlighting for julia from package manager.
- Run julia on colab (experimental): https://stackoverflow.com/questions/58270424/julia-in-google-colab

Intro to Julia

- Variables, Expressions and Statements
  - Variables
    - `x = 42`: Variable Assignment
  - Expressions
    - `x + 1`: Expression
  - Statements
    - `y = x + 1`: Statement
    - `# This is a comment`

    - Unicode variable names: `Δ = 1`
- Operators and Functions
    - Operators: `+, .+`
        * binary: `+, -, *, /, ^`, logicals
        * elementwise: `.` syntax for arrays
        * ternary: `if x==1 ? println("x is one") : println("x is not 1")`
    - Functions: inline, defined and inplace(!)
        * inline: `f(x, y) = x + y`
        * defined:

```
def addFunction(x, y)
  z = x + y
  return z
end
```

        * inplace: `sort!(arr)` - a convention
    - Standard: print(), typeof()
- Control Flow: Conditionals and Loops
    - Conditionals:

```
if x==1
 print("x is one")
else
 print("x is not one")
end
```

    - Loops: for, while, break, continue

```
for i=1:100
  print(i)
end
```

- Arrays and Strings
    - Arrays
        * mutable
        * `arr = [1,2,3]`
        * typeof(), size()
        * indexing, slicing, push, pop
    - Strings
        * immutable
        * Concatenation: `x * y`
    - Dictionaries and Tuples
- Call Python modules from Julia

Some relevant modules

- CSV
    - Read a CSV file into a dataframe and slice/dice `data = CSV.read("~/filename.csv")`
- Plots

    - Make line or scatter plots

  ```
  x = 1:10
  y = rand(10)
  using Plots
  plot(x,y)
  scatter(x,y)
  ```
- Statistics
    - mean, var, std, cor …
- Linear Algebra
    - Identity matrices, trace, determinant …
- Flux
    - Machine learning and neural networks with Julia

References

- Think Julia by Ben Lauwens is a free online book and a comprehensive reference for learning Julia.
- Julia Official Documentation: https://docs.julialang.org/en/v1/

Python (15 minutes)

Installation

Python comes pre-installed with Ubuntu and macOS. For windows, there are installers available online. I advise you to use winpython distribution if you are starting with Python and need it only for this course. Although the best way to build with Python is using the mini/anaconda distribution.

Intro to Python

There is a short and crisp Python tutorial by Stavros Korokithakis here. It is also a useful resource for a quick Python review.

Some relevant modules

- numpy
- matplotlib
- scikit-learn
- pandas