Ch1 : NumPy QuickStart

December 30, 2013

[1] sudo apt-get install python-numpy

1 Vector Example

[2] more efficient than Python lists, when it comes to numerical operations. less explicit loops than equivalent Python code. The arange function (imported from NumPy)creates a NumPy array with integers 0 to n. By using datetime to create a time recorder, we see that NumPy is faster thaan Python lists. Note that the string representation is different. NumPy array delimited by space (think of it as a matrix)

2 MatPlotLib

sudo apt-get install python-matplotlib

3 IPython

- Interactive shell created by scientists for doing experiments
- IPython authors only request that you cite IPython in scientific work where IPython was used.
- Pylab switch imports all the Scipy, NumPy, and Matplotlib packages
- sudo apt-get install ipython-notebook
- Installing additional scientific computing packages [3]
  sudo apt-get install python-matplotlib python-scipy python-pandas python-sympy python-nose
- quit( ) or Ctrl + D quits the IPython shell
- Save current IPython shell by `%logstart`
- IPython shell is like UNIX shell, can use ls...etc
- This runs `%run -i vector.py 1000`
- ipdb is IPython Debugger
- -d option for `%run` starts an ipdb debugger with 'c' the script is started. 'n' steps through the code.
  `%run -d vector.py 1000`

[1] NumPy 1.5 Beginner Guide by Ivan Idris

[2] see vector.py

[3] http://ipython.org/install.html

- Enter c at the ipdb> prompt to start your script.

- -p option:(profile) we can identify which parts of our program is taking most of the computing time. This prints out a list of all the methods and function called and then prints out the time it took to compute them.Z

- `%hist` command shows the command history

- we can use ¡tab¿ for autocomplete

- IPython don't need to call "print" to display variable values

- Support: There is an IRC channel on irc.freenode.net. The channel is called #scipy, but you can also ask NumPy questions since SciPy users also have knowledge of NumPy, as SciPy is based on NumPy. There are at least 50 members on the scipy channel at all times.

3.1 Notes on IPython Shell Documentation

- ipython -h :shows command line options available

- help : standard Python help

- %magic : more information on magic subsystem

- %alias :system command aliases

- ?word or word?: brief info about an object

- ??word or word??: fill info about an object

- %pdoc : show docstring

- TAB: completion in local namespace

- TAB: show all available command

- Ctrl + p: match history items that match what you've type so far

- Simmilarly for Ctrl+n (next, down)

- Ctrl +r :search box

- %hist : search history by index

- exec In[3]: re-execute Input line 3

- Be careful not to overwrite global variables, dont name starting with _ or multiple _

- auto_parenthesis and auto-quoting : but they don't seem much helpful

3.2 Logging your session for later use

- logstate show state of the logger (on or off)
- logstart start logging (default log file is ipython_log.py, in the present working directory
- logstart filename store history up to this point, and continue logging history, in filename
- logstart -r filename same as above, but use the raw input: don't put the _ip.magic() wrapper around magic commands
- logon start logging after stopping
- logoff stop logging after starting
- runlog log1 log2 run the log file log1, then run the log file log2 (this executes the logged histories)
- logstart -o :stores output
- logstart -o /home/Desktop/mylog.log
- If start ipython this way, you can name the logfile yourself: ipython –logfile=logfile.txt