Report
on

Python and Deep Learning
08-Feb-2019

Submitted by

Love Prakash

DronaMaps Private Limited

Contents

1 What is Python:

Python is a high-level general purpose programming language that offers multiple paradigms like object-orientation, and structural and functional programming for software development. It works on cross-platform operating systems and can be used across to develop a wide range of applications including those intended for image processing, text processing, web, and enterprise level using scientific, numeric and data from network. BitTorrent, YouTube, Dropbox, Deluge, Cinema 4D and Bazaar are a few globally-used applications based on Python.

Figure 1: Python

2 Applications of Python in the Real World:

2.1 GUI-Based Desktop Applications:

Python has simple syntax, modular architecture, rich text processing tools and the ability to work on multiple operating systems which make it a desirable choice for developing desktop-based applications. There are various GUI toolkits like wxPython, PyQt or PyGtk available which help developers create highly functional Graphical User Interface (GUI). The various applications developed using Python includes:

2.1.1 Image Processing and Graphic Design Applications:

Python has been used to make 2D imaging software such as Inkscape, GIMP, Paint Shop Pro and Scribus. Further, 3D animation packages, like Blender, 3ds Max, Cinema 4D,

Houdini, Lightwave and Maya, also use Python in variable proportions.

2.1.2 Scientific and Computational Applications:

The higher speeds, productivity and availability of tools, such as Scientific Python and Numeric Python, have resulted in Python becoming an integral part of applications involved in computation and processing of scientific data. 3D modeling software, such as FreeCAD, and finite element method software, such as Abaqus, are coded in Python.

2.1.3 Games:

Python has various modules, libraries and platforms that support development of games. For example, PySoy is a 3D game engine supporting Python 3, and PyGame provides functionality and a library for game development. There have been numerous games built using Python including Civilization-IV, Disney's Toontown Online, Vega Strike etc.

2.2 Web Frameworks and Web Applications:

Python has been used to create a variety of web-frameworks including CherryPy, Django, TurboGears, Bottle, Flask etc. These frameworks provide standard libraries and modules which simplify tasks related to content management, interaction with database and interfacing with different internet protocols such as HTTP, SMTP, XML-RPC, FTP and POP. Plone, a content management system; ERP5, an open source ERP which is used in aerospace, apparel and banking; Odoo – a consolidated suite of business applications; and Google App engine are a few of the popular web applications based on Python.

2.3 Enterprise and Business Applications:

With features that include special libraries, extensibility, scalability and easily readable syntax, Python is a suitable coding language for customizing larger applications. Reddit, which was originally written in Common Lips, was rewritten in Python in 2005. Python also contributed in a large part to functionality in YouTube.

2.4 Operating Systems:

Python is often an integral part of Linux distributions. For instance, Ubuntu's Ubiquity Installer, and Fedora's and Red Hat Enterprise Linux's Anaconda Installer are written in Python. Gentoo Linux makes use of Python for Portage, its package management system.

2.5 Language Development:

Python's design and module architecture has influenced development of numerous languages. Boo language uses an object model, syntax and indentation, similar to Python. Further, syntax of languages like Apple's Swift, CoffeeScript, Cobra, and OCaml all share similarity with Python.

2.5.1 Prototyping:

Besides being quick and easy to learn, Python also has the open source advantage of being free with the support of a large community. This makes it the preferred choice for prototype development. Further, the agility, extensibility and scalability and ease of refactoring code associated with Python allow faster development from initial prototype.

Since its origin in 1989, Python has grown to become part of a plethora of web-based, desktop-based, graphic design, scientific, and computational applications. With Python available for Windows, Mac OS X and Linux / UNIX, it offers ease of development for enterprises. Additionally, the latest release Python 3.4.3 builds on the existing strengths of the language, with drastic improvement in Unicode support, among other new features.

3 Some Advanced Python Concepts:

Following are some advanced Python Concepts:

- **Introduction into the sys module:**

  If there are questions about the import statement, we recommend the introductory chapter of our basic course concerning this topic Modular Programming and Modules

  The sys module provides information about constants, functions and methods of the Python interpreter. dir(system) gives a summary of the available constants, functions and methods. Another possibility is the help() function. Using help(sys) provides valuable detail information.

- **Python and the Shell:**

  Shell is a term, which is often used and often misunderstood. Like the shell of an egg, either hen or Python snake, or a mussel, the shell in computer science is generally seen as a piece of software that provides an interface for a user to some other software or the operating system. So the shell can be an interface between the operating system and the services of the kernel of this operating system. But a web browser or a program functioning as an email client can be seen as shell as well

- **Forks and Forking in Python:**

  A tree as an example for forking Long before biologists started their research of cloning, computer scientists had a successful history of cloning. They cloned processes, though they didn't call it cloning but forking. Forking is one of the most important aspects of Unix and Linux. When a process forks, it creates a copy of itself. More generally, a fork in a multithreading environment means that a thread of execution is duplicated, creating a child thread from the parent thread. they are identical but can be told apart. The fork operation creates a separate address space for the child. The child process has an exact copy of all the memory of the parent process. The execution of the parent and child process is independent of each other.

- **Introduction into Threads:**

  A Thread or a Thread of Execution is defined in computer science as the smallest unit that can be scheduled in an operating system. Threads are normally created by a fork of a computer script or program in two or more parallel (which is implemented on a single processor by multitasking) tasks. Threads are usually contained in processes. More than one thread can exist within the same process. These threads share the memory and the state of the process. In other words: They share the code or instructions and the values of its variables.

- **Pipe, Pipes and "99 Bottles of Beer":**

  Unix or Linux without pipes is unthinkable, or at least, pipelines are a very important part of Unix and Linux applications. Small elements are put together by using pipes. Processes are chained together by their standard streams, i.e. the output of one process is used as the input of another process. To chain processes like this, so-called anonomymous pipes are used. The concept of pipes and pipelines was introduced by Douglas McIlroy, one of the authors of the early command shells, after he noticed that much of the time they were processing the output of one program as the input to another. Ken Thompson added the concept of pipes to the UNIX operating system in 1973. Pipelines have later been ported to other operating systems like DOS, OS/2 and Microsoft Windows as well.

- **Python Network Scanner**

  When it comes to penetration testing or just a simple analysis the network scanner is one major tool for analysing which hosts are available on the local network. Today there exist lots of tools, but depending on what you want to do it is a good idea to write your own analysis and penetration testing tools.

  In this case we start with a simple network scanner, which lists the available hosts on your network. To get this done there are two (basically some more but we stick to the two simplest ones) possibilities:

  – ICMP Echo Request
  – TCP Scan

- **Graph Theory and Graphs in Python**

  Before we start our treatize on possible Python representations of graphs, we want to present some general definitions of graphs and its components. A "graph"1 in mathematics and computer science consists of "nodes", also known as "vertices". Nodes may or may not be connected with one another. In our illustration, - which is a pictorial representation of a graph, - the node "a" is connected with the node "c", but "a" is not connected with "b". The connecting line between two nodes is called an edge. If the edges between the nodes are undirected, the graph is called an

undirected graph. If an edge is directed from one vertex (node) to another, a graph is called a directed graph. An directed edge is called an arc. Though graphs may look very theoretical, many practical problems can be represented by graphs. They are often used to model problems or situations in physics, biology, psychology and above all in computer science. In computer science, graphs are used to represent networks of communication, data organization, computational devices, the flow of computation, In the latter case, the are used to represent the data organisation, like the file system of an operating system, or communication networks. The link structure of websites can be seen as a graph as well, i.e. a directed graph, because a link is a directed edge or an arc. Python has no built-in data type or class for graphs, but it is easy to implement them in Python. One data type is ideal for representing graphs in Python.

- **Currying in Python:**

  In mathematics and computer science, currying is the technique of breaking down the evaluation of a function that takes multiple arguments into evaluating a sequence of single-argument functions.f Currying is also used in theoretical computer science, because it is often easier to transform multiple argument models into single argument models.

- **A Python Class for Polynomial Functions:**

  There is a lot of beaty in polynomials and above all in how they can be implemented as a Python class. We like to say thanks to Drew Shanon who granted us permission to use his great picture, treating math as art!

- **Recursive Programming: Towers of Hanoi:**

  Why do we present a Python implementation of the "Towers of Hanoi"? The hello-world of recursion is the Factorial. This means, you will hardly find any book or tutorial about programming languages which doesn't deal with the first and introductory example about recursive functions. Another one is the calculation of the n-th Fibonacci number. Both are well suited for a tutorial because of their simplicity but they can be easily written in an iterative way as well.

  If you have problems in understanding recursion, we recommend that you go through the chapter "Recursive Functions" of our tutorial.

  That's different with the "Towers of Hanoi". A recursive solution almost forces itself on the programmer, while the iterative solution of the game is hard to find and to grasp. So, with the Towers of Hanoi we present a recursive Python program, which is hard to program in an iterative way.

- **Finite State Machine in Python:**

  A "Finite State Machine" (abbreviated FSM), also called "State Machine" or "Finite State Automaton" is an abstract machine which consists of a set of states (including the initial state and one or more end states), a set of input events, a set of output

events, and a state transition function. A transition function takes the current state and an input event as an input and returns the new set of output events and the next (new) state. Some of the states are used as "terminal states".

The operation of an FSM begins with a special state, called the start state, proceeds through transitions depending on input to different states and normally ends in terminal or end states. A state which marks a successful flow of operation is known as an accept state.

4 Getting Started with Deep Learning in Python:

Deep Learning, a prominent topic in Artificial Intelligence domain, has been in the spotlight for quite some time now. It is especially known for its breakthroughs in fields like Computer Vision and Game playing (Alpha GO), surpassing human ability. Since the last survey, there has been a drastic increase in the trends

4.1 Google Trend shows:

This is how the interest of people growing over time in Deep Learning:-

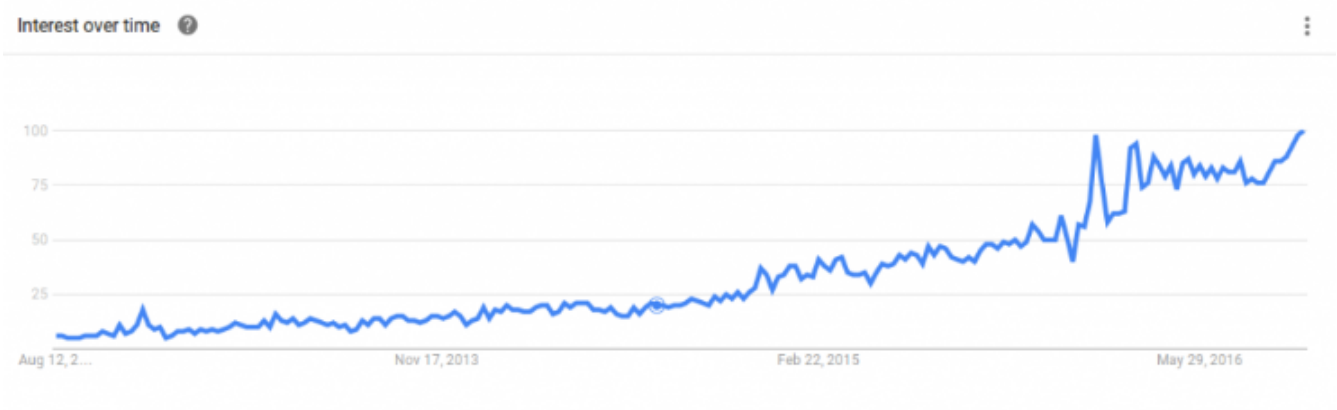

Figure 2: Python

4.2 Pre-requisites for Deep Learning:

It is recommended that before jumping [1]on to Deep Learning, you should know the basics of Machine Learning. The Learning Path on Machine Learning is a complete resource to get you started in the field.

If you want a shorter version, here it is:

- Basics of Math
- Basics of Python
- Basics of Statistics
- Basics of Machine Learning

4.2.1 Setup your Machine:

Before going on to the next step, make sure you have the supported hardware. It is generally recommended that you should have atleast

- A good enough GPU (4+ GB), preferably Nvidia
- An OK CPU (eg. Intel Core i3 is ok, Intel Pentium may not be)
- 4 GB RAM or depending upon the dataset.

If you don't have the required specifications, you could either buy it or lease an Amazon Web Service instance. Here's a good guide for using AWS for deep learning.

4.2.2 A Shallow Dive:

Now that you have a good enough knowledge of pre-requisites, you should go on further into understanding Deep Learning.

As per your preference you could follow:

- Blog Approach
- Video Approach
- TextBook Approach

Along with the pre-requisites, you should get to know the popular deep learning libraries and the languages for running them. Here's a (non-comprehensive) list

- Caffe
- DeepLearning4j
- Tensorflow
- Theano
- Torch

4.2.3 Choose your own Adventure!

Now comes the interesting part! Deep[1] Learning has been applied in various fields with state-of-the-art results. To get a taste of this side of the moon, you, the reader, gets to choose which path to take. This should be a hands-on experience, so that you get a proper foundation.

4.2.4 Deep Learning for Computer Vision:

- Primer : "DL for Computer Vision" blog.
- Project : "Facial Keypoint Detection" Tutorial
- Required libraries : Nolearn
- Associated Course : "CS231n: Convolutional Neural Networks for Visual Recognition"

4.2.5 Deep Learning for Natural Language Processing:

- Primer : "Deep Learning, NLP, and Representations" blog.
- Project : "Deep Learning for Chatbots": "Part 1", "Part 2"
- Required library : Tensorflow
- Associated Course : "CS224d: Deep Learning for Natural Language Processing"

4.2.6 Deep Learning for Speech/Audio

- Primer : "Deep Speech: Lessons from Deep Learning" news article and corresponding video.
- Project : "Music Generation using Magenta (Tensorflow)"
- Required library : Magenta
- Associated Course : "Deep Learning (Spring 2016), CILVR Lab@NYU"

4.2.7 Deep Learning for Speech/Audio

- Primer and Project : "Deep Reinforcement Learning: Pong from Pixels"
- Required Library : No deep learning library required. Although you do require openAI gym to test your model.
- Associated Course : "CS294: Deep Reinforcement Learning"

4.3 Deep Dive into Deep Learning:

Now you are (almost) ready to make a dent in Deep Learning Hall of Fame! The path ahead is long and deep (pun intended) and mostly unexplored. Now it is upto you to make use of this newly acquired skill as efficiently as you can. Here are some tips you should do to hone your skill.

- Reiterate the above step with a different adventure.
- Deep Learning for none of the above! (eg. DL for trading, DL for optimizing energy efficiency)
- Use your newly learned skills to build something (Remember, with great power; comes great responsibility)
- Test your Deep Learning skills (eg. kaggle)
- Participate in Deep Learning community. (eg. Google Group, DL Subreddit)
- Follow recent researches / researchers. (eg. "RE.WORK DL Summit")

5 Conclusion:

I hope this learning path was helpful to you. I have tried to make it as comprehensive as possible. Now, it's time for you to practice and read as much as you can. To gain expertise in working in neural network try out our deep learning practice problem – Identify the Digits.

Once you have an understanding of Deep Learning and its associated concepts, take the Deep Learning Skill test. The way Deep learning is gaining recognition it is important to be familiar with it.

References

[1] Joe Lemley, Shabab Bazrafkan, and Peter Corcoran. Deep learning for consumer devices and services: Pushing the limits for machine learning, artificial intelligence, and computer vision. *IEEE Consumer Electronics Magazine*, 6(2):48–56, 2017.