Which language is best for Artificial Intelligence?

Python is one of the best languages I came across for working on AI projects. Clean easy code, freely typed, saves a lot of time from ridiculing yourself around handling compilation issues and rather focus on the your real problem. Along with that, The python community has developed a lot of libraries and extensions to handle most common tasks used in AI projects like Plotting Graphs, Processing Data etc easily. There are also packages available for almost all AI components ranging for Bayesian Networks to Artificial Neural Networks.

**Some of the helpful packages in python:**

**Numpy (A scientific computing library): @NumPy - Numpy**

**Scypy (Advanced computing): @SciPy.org - SciPy.org**

**Pybrain (Machine learning in python): @PyBrain**

Alternatively, I think JAVA is also a strong language for AI tasks with a nice community support, but the length of code and the complexity is intimidating.

Not one language is best for AI but the three language levels, that you usually need.

Bare metal language level, that is Assembly and other hardware efficient routines, which also includes the usage of GPU and FPGA power. This will cover crucial algorithms and you can do that like at least 5 times faster than on the next level.

Compiler language level, usually “C”, where you get shit done and where you are implementing the bigger algorithms at large. With access to very efficient subroutines in Assembly or hardware, you get an almost optimal way of processing data. Everything you do here is at least 5 times faster than the next level, often you get factor 200 or up to factor 10k. Don’t do any actual work on the next level.

The script level, which is usually any HLL, but a simple and efficient script like Lua would even work. Sometimes you’ll find data base level here or even the use of prototype languages like Prolog or Python. The latter has the great danger, that your average limited HLL freak will start to actually do things in that script and with that will waste like factor 200 of your available hardware by that.

But the script level is important. You have to be able to change the data flow and configuration of the programming of your algorithms from this level, without the need to recompile it.

A prototype language like Prolog or Python is dangerous, because people tend to stay on this level and don’t move their core algorithms out on the second level.

But you will often see Lua and database languages here.

While the use of database languages is one of the limits of current AI systems. Those languages are usually so slow, that they must not be used for this. The data flow inside of AI processes is so high, that you need to keep the data moving. You can’t wait for a base to do its thing. Maybe Sqlite is a solution, because it’s moving the L3 language nearer to the L2 core with that.

All in all it is not a single language. And even on each of those three levels you will most likely not only use one language but some. Like Assembly and Cuda or OpenCL on L1, maybe Verilog for hardware implementation of some of the networks.

There is no single “AI language”, we are engineers and we have a toolbox. We need everything in there to be able to do our thing. And not just the screwdriver and try to drive the nails into walls with that.

There is a place where HLL should be used in those levels and if you are overusing it, you will wreck your project. And most often it is not just one program but many programs working together. And if you start doing the scripting work with your core-language you will fuck up your system, too. Because you have to recompile everything just because you did a little change or did change the configuration.

Same is true for the bare-metal level 1.

The HLL level should be expandable by the core level. Like it is possible with Lua and like it was in the beginning of the age of scripting. With Basic, with Rexx, with Perl. All those languages were expandable and you always were able to add new functionality to them by writing functions in a core language.

The usage of HLL today is wildly grown decadent. Show me how to expand Python with a core language routine or package? That’s hell of complicated to do, if it is possible at all. “Write HLL in HLL” is the credo since, uh, since the Lisp people took over with their toxic credo and poisoned the world of programming with this bullshit.

Lazy.

I call this lazy, to be so inflexible to leave one level and move on the correct level of implementation. And stay on that one level like an oak tree in the storm. It’s usually the tree that splinters and feeds the worms when the storm is over. It’s a good way for an oak to die: inflexibility.

You have to move over those three levels. Whichever languages you might chose to use. But always, really always, be flexible and chose a language that has a smooth, easy, and expandable interface to the next level of languages, up and down.

Always chose a language that plays well with the other kids and that can climb the hierarchy. Negative example: C++ doesn’t work well. You can’t link C++ up to anything. Only down to C. The only way up is over socket connections or something like that. And that’s bad. It doesn’t play well with the other kids.

That’s my advice. You have different problems in AI and with different problems you might want different languages to describe them. Make sure those languages cooperate well with each other. If they don’t: don’t use them.

The best answer often depends on requirements for the AI project. Are you impersonating human intelligence? Crunching huge data sets? Categorizing?

Quite a few people I work with like to use R and Python for regression, decision tree, neurals and random forest analysis. They are quick and easy to load up and use. We take R Studio and tie right into data in SQL or Mongo.

When I wrote a pre-Siri communication tool that could call out external functions I used PHP as I wanted the communication tool web enabled (& weakly typed).

Over at Ford they are cranking out machine learning tasks with CUDA to take advantage of parallel processing.

My good friend built a company with a JAVA engine for his tool that performs fuzzy logic on data from all different sources to determine correlations and connections that are not appropriate. Now, when we tried to feed data into his system commercial data/XML transform systems were about $50-100K, so we wrote our own in Groovy/Grails - XML Slurper !!!. You aren't going to write that integration piece of the data puzzle in LISP (even though LISP my be fine in figuring out the answer)!

Myself, I find that I would like to one day have time to sit down and design a graphical AI programming tool that incorporates a rule based engine. I like the idea of having complete graphical control over my trees & rules for making judgement. Hand coding decisions takes too long ... and letting the system run through attributes and determine rules that can not be reviewed in a graphical mode often produces poor results. Think of having a system return a value that has a great result but does not show you the decision process, and then you run again and get a different result because the system changed it's decision process. I want to see the decision process each time.

Obviously, the end game is for the system to bring external feedback and adjust the rules without my intervention ... but I want to help with teaching and hard-setting some rules. C++ or JAVA may be a good tool to design this. However, it really ends up being dependent on who helps me with a particular project and what I have to bribe them with!

Here we look at the best five programming language for artificial Intelligence development. It is a big concept, so it is very hard to refer to a single programming language.

Python: In artificial intelligence, Python is one of the most widely used programming languages because of its simplicity. The main use is for AI algorithms and data structure. It has a lot of useful libraries that are useful for AI development. For example, for advanced computing, Skype is used. For scientific computation capability, Numpy is used and for machine learning. There are tons of resources available online for AI using Python.

Java: Java is an Object oriented programming language, so it is a great choice. This language provides all high-level features needed work on AI projects. It offers inbuilt garbage collection, and it is a portable language. The plus point with Java community there will be somebody to assist you with your queries and effort. AI is full of the algorithm, so Java is the best choice it provides an easy way to code good algorithms. You can develop algorithms like search algorithm, natural language processing algorithm or neural algorithm. Java has the feature of scalability which best for AI projects. Java is still not as high level as Prolong and Lisp and not faster than C.

Lisp: Lisp is the programming language developed between the 1970s and 1980s. It’s a great programming language used in large AI projects, such as Macsyma, DART, and CYC. Because of its best prototyping capabilities and its support for symbolic expression Lisp is used in AI field. This language is used in Machine Learning/ILP subfield because of its usability and symbolic structure. Lisp is the top programming language in AI field because of its best features. Lisp language has a feature of automatic garbage collection with dynamic creation of new object. Lisp generates efficient code with well development compilers. This language has a macro system that let developers create a domain-specific level of abstraction on which to build the next level. Because of this features, Lisp excels compared to another language.

Prolog: Prolog is the excellent programming language for artificial Intelligence. Some basic features of Prolog which are extremely useful for AI programming. It offers tree based data structuring mechanism, automatic backtracking and pattern matching combining these mechanisms provides a flexible framework to work with artificial intelligence. In the expert system of AI Prolog is extensively used for working on medical projects. Unlike traditional programming language, Prolog is a high-level programming language based on formal logic. It is a language performing sequence of commands and solving logical formulas. As its program consists of list facts and rules, it is rule-based as well as declarative language.

C++: C++ is the greatest object-oriented programming language in the world. For AI project of the time, sensitive C++ is extremely useful. This language can talk at the hardware level and allows developers to progress their program execution time. For statistical AI technique such as neural network, C++ is the preferred language. The search engine can utilize C++ widely. Games in AI mostly coded with C++ for speedy execution and response time.

CONCULSION:

Before deciding a programming language for artificial intelligence makes sure that it can be utilized not partially but extensively. Freelance services are available in all of these programming languages. Also preferring a programming language for you AI project depends upon subfield. Python is well-known due to its flexibility, C++ and Java are also useful because of best features they offer. Lisp and Prolog are always being used extensively because of their productive features.