



Emergent Tech ▶ **Artificial Intelligence**

A bot lingua franca does not exist: Your machine-learning options for walking the talk

Monolingualist or code polymath?

By [Danny Bradbury](#) 25 Apr 2017 at 08:01

10 [SHARE](#) ▼



So, you want to create a hugely successful machine-learning startup? Or you've been asked to start investigating ML for your firm? Well, you'd better get programming – but what language should you use? No languages have been designed specifically with ML in mind, but some do lend themselves to the task.

Developers experimenting with machine learning will spend most of their time processing data sets, running them against a machine-learning

algorithm, and then classifying them again until the results seem right. Languages that can handle a lot of heavy data lifting are therefore near the top of the list for many machine-learning experts.

What are your options?

R

Statistical programming languages are one option, and something that's gained a lot of attention of late is [R](#). Although not developed with artificial intelligence in mind, R can handle many of the tasks that programmers need in a machine-learning environment.

R excels at gathering datasets and cleaning them up. It also includes a variety of functions, either natively or in packages such as [caret](#), that data scientists can use to help model their data sets.

Matlab

Whereas R is a free language, [Matlab](#) is a proprietary commercial product. R comes from a statistical data analysis background while Matlab cut its teeth in numerical computation for engineering and physical sciences. It has a variety of toolboxes that are useful for various machine-learning functions, such as the [Statistics and Machine Learning Toolbox](#), the [Machine Learning Toolbox](#) and [Liblinear](#). The machine-learning toolboxes for Matlab can get very specific. [MeteoLab](#) focuses on machine learning for climate analysis, for example.

You can integrate Matlab models directly into applications written in Java or .Net, and you can also export them as Excel models, making them easier to apply in other systems after you have written them in MatLab.

The bottom line with Matlab is that it offers an excellent programming environment, with debugging capabilities and object browsing, but it'll cost you. It also offers [Simulink](#), which lets you program visually by connecting blocks together.

The alternative to MatLab is [Octave](#), an open-source scientific programming language that is broadly compatible.

Julia

Julia wasn't designed explicitly for machine learning, but is increasingly appropriate for it, says Simon Byrne, a quantitative programmer at Julia Computing. Launched in 2009, this numerical programming language bridges two worlds, he contends, with a programming language that was both easy and fun to use, while also being fast.

Typically, Byrne suggests that languages are one or the other. Python, for example, is an interpreted language, and very permissive. It uses implicit typing, so that developers don't have to explicitly tell the system which type of variable they're declaring (such as an integer or a decimal, say). Programmers pay for this with relatively poor performance.

Conversely, C programs, which move like lightning, must be compiled before they'll run, and typing is explicit, which places more burden on the developer. One doesn't simply hack things around in C.

Whereas R and Matlab morphed into programming languages from tools designed to execute small scripts, Julia has programming rigour baked in from the start. "It's for large numerical work. Lots of floating point stuff and large arrays," Byrne says. "What you write looks like mathematics."

While it's good with heavy maths, it's also made to be programmer-friendly. For example, it uses inference typing.

All of this makes Julia a one-stop shop for the various stages of a complex machine-learning project, says Byrne. Other languages can easily handle simple machine-learning tasks – if you're going to write a basic image recognition service, then you may as well do it in something generic like Python. On the other hand, a more complex machine-learning system with specific needs from a custom framework would be a job for Julia.

One interesting thing about Julia is that it can pull in other language libraries where necessary. Right now, it supports anything that you can run natively in Python, or import into a Python environment.

The Julia community also has its own open-source package effort. Flux is a system that you can use in conjunction with either Google's TensorFlow framework, or with MXNet, the deep-learning platform adopted by Amazon. The team is also exploring integration with Torch, Facebook's deep-learning framework.

"TensorFlow dominates, but there are still a lot of holes and things that it can't do. So there is room for other frameworks," Byrne says. Whereas TensorFlow is effectively a set of wrappers around C++ and Python code, Flux integrates with debuggers, he says. Think of it as a domain-specific language for specifying and representing machine learning models.

Another promising development is [CUDANative](#), a package that will enable Julia programmers to create native code for GPUs. GPUs are a valuable hardware resource for AI, due to their high concurrency and ability to handle floating point calculations with double precision.

Orange

[Orange](#) is a visual programming environment for data visualisation. Developed at the University of Ljubljana, it was released in 1997.

The open-source tool uses Python statistical programming packages under the hood. The GUI-based system allows users to drag data sets on to a work board and connect them with "widgets" – components that carry out mathematical functions such as classification, regression and unsupervised learning. There are also add-on components for functions like text mining.

"R is absolutely great, it's the standard for statistical computation. But you have to speak R," says Blaž Zupan, professor at the university's faculty of computer and information science. "Not many data owners speak R."

By putting a visual front-end on traditional Python packages, Orange's developers created a way to easily manipulate data without having to learn an entirely new language – or indeed, any language at all.

The system can handle tasks like image cluster analysis using an image-embedding widget included out of the box, but don't expect to use it for the kind of heavy lifting in multi-layer neural network analysis, warns Ajda Pretnar, a researcher in the faculty who focuses on Orange for machine learning and data mining.

"Orange is mainly designed for laptops and PCs. It isn't something you'd do on GPUs," she says. This is for interactive data hacking and testing machine learning concepts on data sets, not winning [ImageNet](#).

Other languages with libraries

Given that no single language is explicitly designed for machine learning, it's worth considering some other languages that have useful libraries.

Python is an easy-to-learn language with some excellent libraries, not only for numerical processing ([NumPy](#) and [SciPy](#)), but also specifically for machine learning ([scikit](#), [gensim](#), [PyML](#)).

On the Java side, [Weka](#) is a collection of open-source data-mining libraries with a strong focus on machine learning.

Don't rule out Haskell, a high-level programming language, which has a high-performance machine-learning library called [HLearn](#). This aims to be faster than low-level compiled languages, but more flexible than high-level ones. The aim is to out-Julia Julia, apparently.

Let's also not forget Lisp. One of the oldest computer programming languages, it was created by John McCarthy, known as one of the fathers of modern AI. It was built with list programming and recursion in mind, which makes it great for building and collapsing tree structures, which constitute one machine-learning approach. There are specific machine-learning libraries for Common Lisp, one of the two most popular modern implementations of the language available [here](#).

The rule of thumb for machine learning is to focus on the dedicated statistical and scientific languages if you're a data scientist, and to get into the others, with appropriate libraries, if you're interested in rolling out machine-learning algorithms that manipulate these statistical concepts

while interacting with more broadly functional backends. Julia, with its focus on heavyweight maths and its algorithmic approach, seems to straddle the two. The collection of third-party packages is still pretty thin, though, so you'd best be community-minded and willing to contribute.

Of course, if you're a polymath and enjoy sampling multiple disciplines, you could always dabble with two or more before choosing which to specialise in.

Python is a good place to start because it's easy to get up to speed with the basic concepts, and adding in specialist machine-learning languages is a cinch.®

Tips and corrections

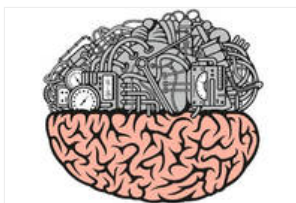
10 Comments


Sign up to our Newsletter - Get IT in your inbox daily

MORE Machine Learning



More from The Register



OpenStack Foundation starts scoping machine learning enhancements

OPENSTACK SYDNEY Users want it and NVIDIA's already sniffing around



Wanna spend a day getting deeply hands-on with machine learning?

EVENTS Two workshops added to MCubed agenda



Machine-learning startup Perspica to be crushed into Cisco's analytics biz

Staff will be folded into January acquisition AppDynamics



Splunk goes native with machine learning, aims to speed up monitoring

Analytics biz eyes up fraud and IoT markets



Need to get up to speed on machine learning, AI?

Just 25 conference tickets left for MCubed



Machine learning? AI? How we learned to relax at MCubed

EVENTS *Reg* conference shows humans how to stay in driving seat

Whitepapers



It's Time to Rethink Your Data Protection

In today's digital world, you must share sensitive information outside your organization. It's imperative that you keep this data safe—wherever it is.



AIIM Industry Watch: Governance and Compliance in 2017

Exploring how organizations are addressing their governance and compliance challenges.



IBM Optimizes Multicloud Strategies for Enterprise Digital Transformation

IDC analyzes how the use of a supply chain management (SCM) approach to optimizing multi-cloud architectures, can simplify and streamline operations.



Mobile: Design to Succeed or Designed to Fail

The mobile ecosystem is constantly evolving to satisfy users' demands for fast, attractive, effective mobile applications and experiences.

About us

[Who we are](#)

[Under the hood](#)

[Contact us](#)

[Advertise with us](#)

More content

[Week's headlines](#)

[Top 20 stories](#)

[Alerts](#)

[Whitepapers](#)

Situation Publishing

[The Next Platform](#)

[Continuous Lifecycle London](#)

[M-cubed](#)

[Webinars](#)



The Register - Independent news and views for the tech community. Part of Situation Publishing

Sign up to our Newsletters

Join our daily or weekly newsletters, subscribe to a specific section or set [News alerts](#)

[Subscribe](#)

Biting the hand that feeds IT © 1998–2017

[Cookies](#)

[Privacy](#)

[Ts&Cs](#)

