

# Learn How to Seamlessly Use Julia for Your Machine Learning Tasks



Dr. Matt Bauman  
Director of Applications Engineering

[www.juliacomputing.com](http://www.juliacomputing.com)

# Learn How to Seamlessly Use Julia for Your Machine Learning Tasks

- Brief introduction to Julia
- Deep dive into a simple machine learning model
- Demo of an advanced ML model

---

## Julia in a Nutshell

---

### Fast

Julia was designed from the beginning for [high performance](#). Julia programs compile to efficient native code for [multiple platforms](#) via LLVM.

### Dynamic

Julia is [dynamically typed](#), feels like a scripting language, and has good support for [interactive](#) use.

### Reproducible

[Reproducible environments](#) make it possible to recreate the same Julia environment every time, across platforms, with [pre-built binaries](#).

### Composable

Julia uses [multiple dispatch](#) as a paradigm, making it easy to express many object-oriented and [functional](#) programming patterns. The talk on the [Unreasonable Effectiveness of Multiple Dispatch](#) explains why it works so well.

### General

Julia provides [asynchronous I/O](#), [metaprogramming](#), [debugging](#), [logging](#), [profiling](#), a [package manager](#), and more. One can build entire [Applications and Microservices](#) in Julia.

### Open source

Julia is an open source project with over 1,000 contributors. It is made available under the [MIT license](#). The [source code](#) is available on GitHub.



Jeff Dean (@ )  
@JeffDean

Julia + TPUs = fast and easily expressible ML computations!



Keno Fischer @KenoFischer · Oct 23, 2018

Our new paper today: [arxiv.org/abs/1810.09868](https://arxiv.org/abs/1810.09868). Compile your #julia code straight to @Google's #CloudTPU. Must go faster! We'll have an (alpha quality) repo up soon for people to start playing with this.

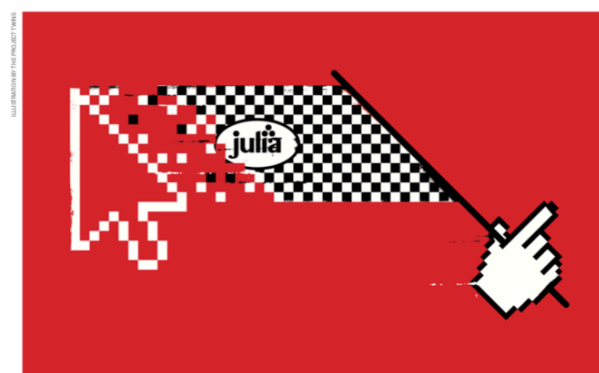
11:23 PM · Oct 23, 2018 · [Twitter for Android](#)

# nature

## TOOLBOX

## JULIA: COME FOR THE SYNTAX, STAY FOR THE SPEED

Researchers often find themselves coding algorithms in one programming language, only to have to rewrite them in a faster one. An up-and-coming language could be the answer.

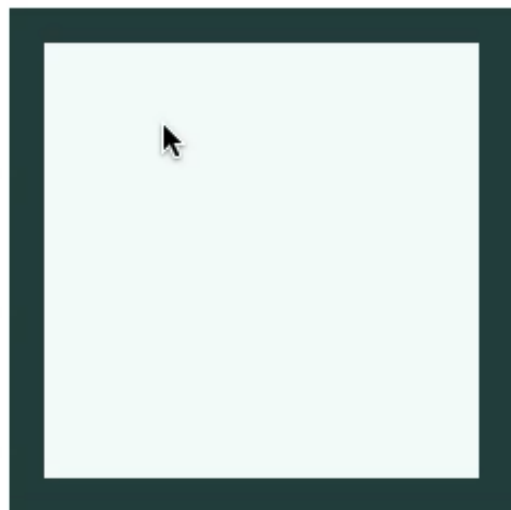


BY JEFFREY M. PERKEL

When it comes to climate modeling, every computational second counts. Designed to account for air, land, sun and sea, and the complicated physics that links them, these models can run to millions of lines of code, which are executed on the world's most powerful computers. So when the coder-climatologists of the Climate Modeling Alliance (CIMA) — a coalition of US-based scientists, engineers and mathematicians — set out to build a model from the ground up, they opted for a language that could handle their needs. They opted for Julia.

Launched in 2012, Julia is an open-source language that combines the interactivity and syntax of 'scripting' languages, such as Python, Matlab and R, with the speed of compiled languages such as Fortran and C. Among climate scientists, the lingua franca is Fortran: speedy, but — with roots dating to the 1950s — not terribly exciting. 'A lot of people, when they hear 'Fortran, are like, 'Oh, my God, I don't want to program in that,' says Frank Giraldo, a mathematician at the Naval Postgraduate School in Monterey, California, and a co-principal investigator on the CIMA project. Younger programmers prefer languages that can accommodate the latest trends in software and hardware design,

Giraldo says, and since adopting Julia he has seen an uptick in interest. 'Some of them are really interested in climate modelling, but others are intrigued by the idea of using Julia for some large-scale application,' he says. Jane Herriman, who is studying materials science at the California Institute of Technology in Pasadena, says that she has seen ten-fold faster runs since rewriting her Python codes in Julia. Michael Stumpf, a systems biologist and self-styled Julia proselytizer at the University of Melbourne, Australia, who has ported computational models from R, has seen an 800-fold improvement. 'You can do things in an hour that would otherwise take weeks or months,' he says.



0 1 2 3 4 5 6 7 8 9

Clear

See code



<https://fluxml.ai/experiments/>

# AlphaZero



LEE SEDOL AND ALPHA GO

**LEE SEDOL**, right, plays the ancient Asian board game Go against Google's artificial intelligence program, AlphaGo. Lead programmer Aja Huang sits at left.

## AI makes its next move

A computer's board game victory marks a milestone

By STEVEN BOROWIEC  
AND TRACEY LIEN

SEOUL — First went checkers, then fell chess. Now, a computer program has defeated the world's top player in the ancient east Asian board game of Go — a major milestone for artificial intelligence that brings to a close the era of board games as benchmarks in computing.

At the Four Seasons Ho-

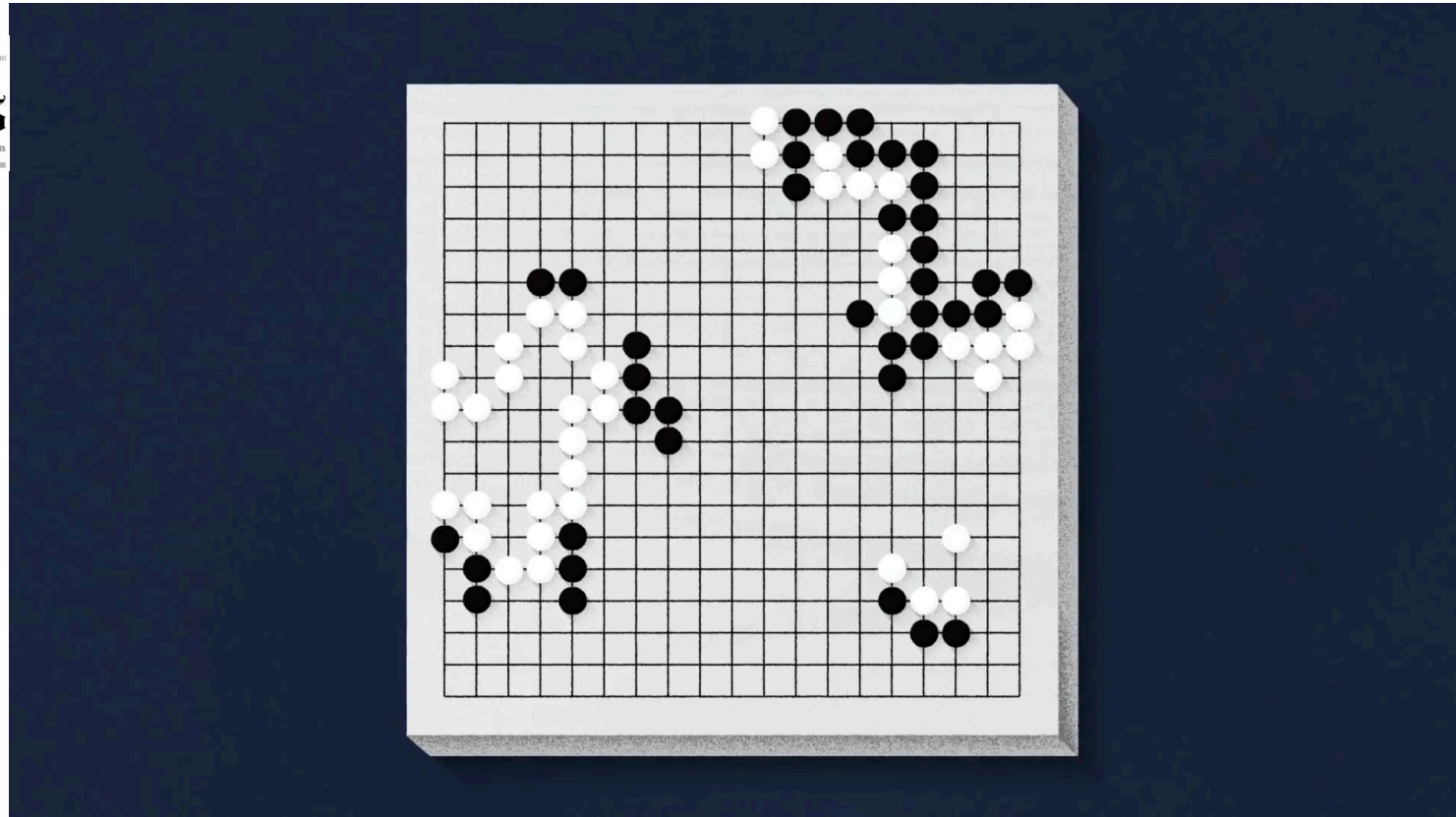
tel in Seoul, Google DeepMind's AlphaGo capped a 3-0 week on Saturday against Lee Sedol, a giant of the game. Lee and AlphaGo were to play again Sunday and Tuesday, but with AlphaGo having already clinched victory in the five-game match, the results are in and history has been made. It was a feat that experts had thought was still years away.

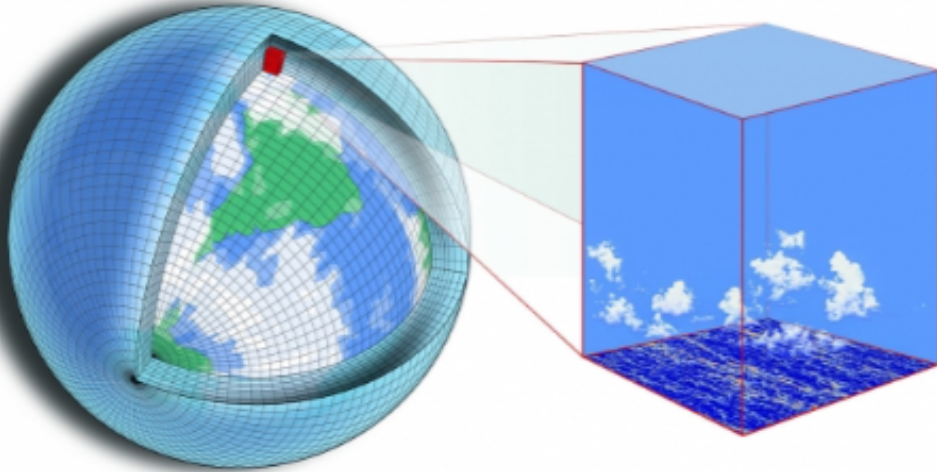
At the postgame news conference, Lee sat bolt up-

right, with a slight tone of resignation in his voice as he tried to explain his failure to get the better of the computer program that has taken from him bragging rights to global Go supremacy.

"I have to express my apologies," Lee said, his voice quivering slightly. He seemed just as sad as after his previous two losses earlier in the week, but this time not so surprised.

"I misjudged the capability," [See AlphaGo, A4]





## New climate model to be built from the ground up

Scientists and engineers will collaborate in a new Climate Modeling Alliance to advance climate modeling and prediction.

**School of Science**  
**December 12, 2018**



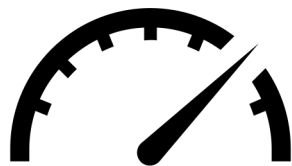




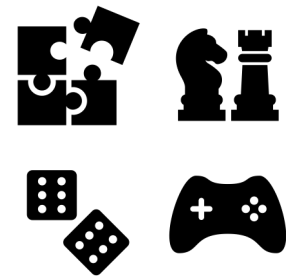
# AlphaZero.jl



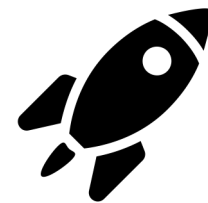
*Simple*



*Fast*



*Extensible*



*Built to scale*



**Jonathan  
Laurent**  
Carnegie Mellon  
University

demo.juliahub.com/ui/Home

Matt Bauman

JuliaHub

beta

Home

Packages

Doc Search

Code Search

Repo Search

Download JuliaPro

Register Packages

Applications

Run in Cloud

Problems? Ideas?  
File an issue here.

Powered by Julia Computing  
JuliaTeam v2.1.0-juliahub

JuliaHub

Track the pulse of the Julia ecosystem, find the packages you use, and discover new packages.

Cloud computing is coming to JuliaHub. [Get notified](#) when it's ready.

Popular Packages

Flux

lJulia

Gadfly

Gen

DifferentialEquations

Knet

JuMP

Plots

Genie

# Popular Tags

image-processing data-structures  
time-series finance neural-networks  
data-science economics visualization  
astronomy deep-learning regression  
differentialequations plotting  
biology math nlp bioinformatics  
mlj mcmc ode statistics queryverse  
machine-learning dae  
data flux optimization sde geo  
gpu dde differential-equations simulation  
automatic-differentiation arrays  
linear-algebra bayesian-inference



- Home
- Packages
- Doc Search
- Code Search
- Repo Search
- Register Packages
- Download JuliaPro
- Applications

Run in Cloud

Problems? Ideas?  
File an issue here.

Powered by Julia Computing  
JuliaTeam v2.1.0-juliahub

## PREVIOUS RUNS

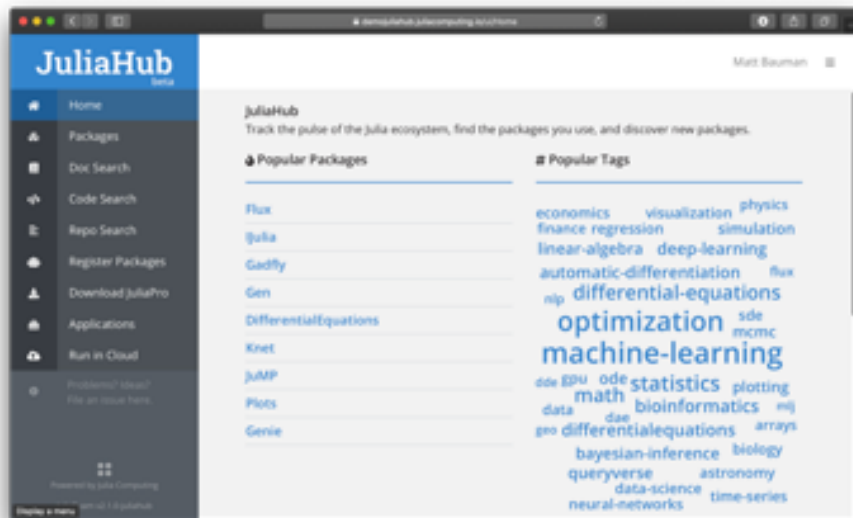
Job Status: Completed Submitted Running Failed Stopped

Filter

Clear

	Timestamp	Job ID	Branch/Tag	Git Tree SHA	Job Status			
<input type="checkbox"/>	7/28/2020, 11:08:26 PM	jr-wpymb69qou	AlphaZero#mb/demo-distributed	Copy		gpu	workers	memory
						1	10	16Gi
<input type="checkbox"/>	7/28/2020, 5:46:59 PM	jr-j0eyx6j1rq	AlphaZero#mb/demo-distributed	Copy		gpu	workers	memory
						1	10	16Gi
<input type="checkbox"/>	7/28/2020, 5:22:58 PM	jr-6qbdpyfxi8	AlphaZero#demo-distributed	Copy		gpu	workers	memory
						1	10	16Gi

Compare



is available for enterprise use

- Private package and registry management
- Deployment behind firewalls and airgaps
- Governance and security
- Deploy into your own AWS account/subnet



10:30 AM GMT

**Training and Deploying ML Models in**



In This Workshop:

- We go over the basics of Julia syntax
- We form a baseline for ML concepts
- We apply them using Flux
- We scale them to real world models
- And demonstrate large novel workflows at scale at high performance