

# Flux: Elegant machine learning with Julia

Mike Innes<sup>1</sup>

<sup>1</sup> Julia Computing

DOI: [10.21105/joss.00602](https://doi.org/10.21105/joss.00602)

## Software

- [Review](#) ↗
- [Repository](#) ↗
- [Archive](#) ↗

**Submitted:** 16 February 2018

**Published:** 03 May 2018

## Licence

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC-BY](#)).

## Summary

Flux is library for machine learning (ML), written using the numerical computing language Julia (Bezanson et al. 2017). The package allows models to be written using Julia’s simple mathematical syntax, and applies automatic differentiation (AD) to seamlessly calculate derivatives and train the model. Meanwhile, it makes heavy use of Julia’s language and compiler features to carry out code analysis and make optimisations. For example, Julia’s GPU compilation support (Besard, Foket, and De Sutter 2017) can be used to JIT-compile custom GPU kernels for model layers (Innes and others 2017a).

The machine learning community has traditionally been divided between “static” and “dynamic” frameworks that are easy to optimise and easy to use, respectively (Innes and others 2017b). Flux blurs the line between these two approaches, combining a highly intuitive programming model with the compiler techniques needed by ML. This enables research into advanced compiler transforms such as batching (Bradbury 2018) without changing any user code.

Flux has been used heavily for natural language processing, but can also support state-of-the-art research models in areas like computer vision, reinforcement learning and robotics. Many examples of such models can be found in the model zoo (Innes and others 2018).

## References

- Besard, Tim, Christophe Foket, and Bjorn De Sutter. 2017. “Effective Extensible Programming: Unleashing Julia on GPUs.” *arXiv* abs/1712.03112. <http://arxiv.org/abs/1712.03112>.
- Bezanson, Jeff, Alan Edelman, Stefan Karpinski, and Viral B. Shah. 2017. “Julia: A Fresh Approach to Numerical Computing.” *SIAM Review*. [julialang.org/publications/julia-fresh-approach-BEKS.pdf](http://julialang.org/publications/julia-fresh-approach-BEKS.pdf). <https://doi.org/10.1137/141000671>.
- Bradbury, James. 2018. “Minibatch.jl.” 2018. <https://github.com/jekbradbury/Minibatch.jl>.
- Innes, Mike, and others. 2017a. “Generic Gpu Kernels.” 2017. <http://mikeinnes.github.io/2017/08/24/cudanative.html>.
- . 2017b. “On Machine Learning and Programming Languages.” 2017. <https://julialang.org/blog/2017/12/ml&pl>.
- . 2018. “Flux Model Zoo.” 2018. <https://github.com/FluxML/model-zoo/>.