DSL for ML An Annotated Bibliography

Isaac H. Lopez Diaz

October 31, 2025

References

[AIM17] Martín Abadi, Michael Isard, and Derek G Murray. A computational model for tensorflow: an introduction. In *Proceedings of the 1st ACM SIGPLAN International Workshop on Machine Learning and Programming Languages*, pages 1–7, 2017.

This paper discusses how TensorFlow works under the hood, by explaining a very limited version of TensorFlow's dataflow computational model. The limited version contains variables, tensors, and read and write operations on variables. The authors explain the semantics of these operations, and how the dataflow graph behaves.

[BM22] Hendrik Pieter Barendregt and Giulio Manzonetto. *A Lambda Calculus Satellite.* College publications, 2022.

[IKS+18] Mike Innes, Stefan Karpinski, Viral Shah, David Barber, PLEPS Saito Stenetorp, Tim Besard, James Bradbury, Valentin Churavy, Simon Danisch, Alan Edelman, et al. On machine learning and programming languages. Association for Computing Machinery (ACM), 2018.

This paper argues that a new language for machine learning is needed. There are various arguments: libraries like TensorFlow are already languages in themselves, current languages work as meta-languages, and a new language could improve certain features. This source is helpful because it lays the basis, informally, as to what is needed for a programming language based on machine learning. Its goal is to explain what features such a language has, what similarities it has with other languages, and how does it differ from conventional programming languages. It changes the approach on how to implement a language or what the target of my language should be.

[Kri17] Shriram Krishnamurthi. *Programming languages: Application and interpretation.* Brown University, 2017.

[Sto77] Joseph E Stoy. Denotational semantics: The scott-strachey approach, 1977.

[VKBR22] Tatiana Castro Vélez, Raffi Khatchadourian, Mehdi Bagherzadeh, and Anita Raja. Challenges in migrating imperative deep learning programs to graph execution: an empirical study. In *Proceedings of the 19th International Conference on Mining Software Repositories*, MSR '22, page 469–481, New York, NY, USA, 2022. Association for Computing Machinery.

> This paper surveys and reviews more than 19 million lines of code to try and identify common bugs on code that migrates imperative style to graph execution programs. This source helps by reinforcing my idea since it presents a large amount of bugs and performance issues. The source can be used as a metric to show how a language can improve the code quality or performance.