# DSL for ML An Annotated Bibliography

Isaac H. Lopez Diaz

October 1, 2025

## References

[FM23] Daniel P Friedman and Anurag Mendhekar. *The little learner: a straight line to deep learning.* MIT Press, 2023.

This book covers the basics of deep learning using the programming language Racket (Scheme).

[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 discusses the necessity of a first class language for machine learning (ML) . It explains that Python is the metalanguage used to build expressions in TensorFlow's internal language. So, TensorFlow does not provide you with functions and data structures, it provides a graph-based language which you manipulate through it's API. The authors turn to answer the question: Why create a new language? Since ML models are becoming more complex there is a necessity to add more language constructs to TensorFlow's internal language (conditionals, loops, recursions, etc.). Another reason presented is that using a meta-language poses more challenges in order to reason about programs since one needs to reason about the meta-language semantics and the internal language's. This is also a burden for the compiler since it is tracing two executions. One approach presented is "define-by-run" where a program is itself the model, using runtime automatic differentiation (AD). However, ML is computational heavy and this approach falls short. The authors present the idea of programmable semantics because of the possible flexibility it may allow, providing features similar to macros. For example, One could specify where the code should have dataflow

semantics. Another idea for such a language would be to have first-class derivatives, mixing symbolic with runtime techniques.

[Smi84] Brian Cantwell Smith. Reflection and semantics in lisp. In *Proceedings of the 11th ACM SIGACT-SIGPLAN symposium on Principles of programming languages*, pages 23–35, 1984.

[Sto81] Joseph E Stoy. *Denotational semantics: the Scott-Strachey approach to programming language theory.* MIT press, 1981.