Neural Architectures for Tactic-Based Automated Theorem Proving

Christian Szegedy, Sarah M. Loos, Aditya Paliwal, Markus Rabe, and Kshitij Bansal

Google Research

1 Introduction

In this talk, we compare various neural network architectures for tactic-based neurally guided proof search for higher order logic in HOL-Light [4] interactive theorem prover. It was first demonstrated in the TacticToe [3] prover that learned guidence for tactic based interactive proof search could yield superior results for automated higher order theorem proving compared to hammers based on first order logic based ATPs [5]. Here we focus on a deep learning based solution. We will be addressing two kinds of tasks: the selection of a tactic out of 41 possible tactics and the ranking of tactic arguments from all the usable tactic arguments from a theorem database.

Our experiments are conducted on the HOList [2] benchmark, which comprises a standardized set of theorems sorted such that later theorems can be proved solely by earlier theorems and definitions in the database. Our main metric is the number of proofs successfully closed on a held out set of theorems. In our imitation learning setup, we train models using our database of human proofs, logged from the HOL-Light libraries. We also experiment with a reinforcement learning setup, allowing the model to control the proof search with tactic and tactic argument selection, while simultaneously training on human proofs. Finally, we perform reinforcement learning without imitation learning (i.e. "from zero" human proofs); in this setting we additionally measure the cumulative number of proofs closed over a fixed number of proof attempts.

2 Architectures Tested

Our theorem prover is based on a simple breadth first search based backward prover augmented by a neural network for premise selection and tactic prediction. The neural network is a two-tower architecture without weight sharing. The two towers produce a fixed dimensional embedding, one for the goal and one for the premise. The two embeddings are combined by a cheap three-layer network to produce a ranking score for the premise. This architectural choice is essential for fast ranking of a large number of premises in relatively short time, since the embeddings for the potential premises can be shared. However, we have a lot of freedom for choosing the architecture for the individual embedding towers that incur the most computational cost. Here we worked with two types of networks: those that consider the input as a sequence of tokens and those that take a graph representation of the formulas. In the latter case we also employ subexpression sharing.

Our experiments focus on various base neural network architectures which all share the common feature that they produce a feature vector for each input token. In order to use the produced features efficiently for ranking the premises, this set (or sequence) of output feature vectors needs to be reduced to a single, relatively short, fixed dimensional feature vector that can be used in a nearest neighbor look up. The choice of this reduction method is also explored in detail here. For the base architectures, we have evaluated the following variants:

- simple convolutional networks,
- dilated convolutional networks (a.k.a WaveNets [7]),
- transformer network architectures [8],
- graph neural networks (GNNs [6]),
- graph attention networks [9].

We additionally evaluate a variety of pooling mechanisms:

- maximum pooling,
- average pooling,
- expanding the dimension of output features before pooling,
- attention based pooling
- and transformer layers (with self-attention).

3 Evaluation Methodologies and Metrics

These architectures were trained with imitation learning (learning from human proof-logs) and the best models were tested in the context of the reinforcement learning from scratch without utilizing any of the human proofs (in the context of DeepHOL-Zero [1]). We also report several proxy metrics for tactic selection and premise ranking and their evolution during the training process. We study which metrics are most indicative of the final end-to-end prover performance.

References

[1] Kshitij Bansal, Sarah M Loos, Markus N Rabe, and Christian Szegedy. Learning to reason in large theories without imitation. *arXiv preprint arXiv:1905.10501*, 2019.

[2] Kshitij Bansal, Sarah M Loos, Markus N Rabe, Christian Szegedy, and Stewart Wilcox. Holist: An environment for machine learning of higher-order theorem proving. *ICML 2019. International Conference on Machine Learning*, 2019.

[3] Thibault Gauthier, Cezary Kaliszyk, and Josef Urban. Tactictoe: Learning to reason with hol4 tactics. In *LPAR-21. 21st International Conference on Logic for Programming, Artificial Intelligence and Reasoning*, volume 46, pages 125–143, 2017.

[4] John Harrison. HOL Light: A tutorial introduction. In *FMCAD*, pages 265–269, 1996.

[5] Cezary Kaliszyk and Josef Urban. Learning-assisted automated reasoning with flyspeck. *Journal of Automated Reasoning*, 53(2):173–213, 2014.

[6] Aditya Paliwal, Sarah Loos, Markus Rabe, Kshitij Bansal, and Christian Szegedy. Graph representations for higher-order logic and theorem proving. *arXiv preprint arXiv:1905.10006*, 2019.

[7] Aäron Van Den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. *CoRR abs/1609.03499*, 2016.

[8] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[9] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.