Neural Theorem Provers

Tim Rocktäschel
University of Oxford
Oxford, UK
tim.rocktaschel@cs.ox.ac.uk

Sebastian Riedel
University College London
London, UK
s.riedel@cs.ucl.ac.uk

Current state-of-the-art methods for automated Knowledge Base (KB) completion learn distributed representations of fact triples using neural link prediction models (Nickel et al. 2016). Neural networks can learn to generalize well when observing many input-output examples, but lack interpretability and straightforward ways of incorporating domain-specific knowledge. Theorem provers, on the other hand, provide effective ways to reason with logical knowledge. However, by operating on discrete symbols they do not make use of similarities between predicates or constants in training data (e.g., `lecturerAt` ∼ `professorAt`, ORANGE ∼ LEMON, etc).

Recent neural network architectures such as Neural Turing Machines (Graves, Wayne, and Danihelka 2014) replace discrete functions and data structures by end-to-end differentiable counterparts. As such, they can learn complex behaviour from raw input-output examples via gradient-based optimization. In the same spirit, we introduce Neural Theorem Provers (NTPs): end-to-end differentiable automated theorem provers working with subsymbolic representations (Rocktäschel and Riedel 2017).

Specifically, we use Prolog's backward chaining algorithm as a recipe for recursively constructing neural networks that are capable of proving facts in a KB. The success score of such proofs is differentiable with respect to vector representations of symbols, which enables us to learn such representations for predicates and constants in ground atoms, as well as parameters of function-free first-order logic rules of predefined structure. NTPs learn to place representations of similar symbols in close proximity in a vector space and can induce rules given prior assumptions about the structure of logical relationships in a KB such as transitivity. Furthermore, NTPs can seamlessly reason with provided domain-specific rules. As NTPs operate on distributed representations of symbols, a single hand-crafted rule can be leveraged for many proofs of queries with symbols that have a similar representation. Finally, NTPs allow for a high degree of interpretability as they induce latent rules that we can decode to human-readable symbolic rules.

In our research we have made the following steps towards a full and scalable implementation of NTPs: (i) we developed an NTP architecture based on differentiable backward chaining and unification of symbol representations, (ii) we developed optimizations to this architecture based on batch proving, approximate gradient calculation and joint training with neural link prediction models, and (iii) we experimentally showed that NTPs can learn representations of symbols and function-free first-order rules of predefined structure, enabling them to perform complex multi-hop reasoning on the Countries KB (Bouchard, Singh, and Trouillon 2015). and the Kinship, UMLS and Nations datasets (Kok and Domingos 2007). The results can be seen in Table 1 and show PR-AUC on the Countries dataset and HITS@1 and HITS@10 on the other datasets for the NTP alone, ComplEx, and an NTP implementation that uses the ComplEx loss as a regularizer on its symbol representations (NTP**λ**).

Corpus		Metric	Model		
			ComplEx	**NTP**	**NTPλ**
	S1	AUC-PR	99.37 ± 0.4	90.83 ± 15.4	$\mathbf{100.00} \pm 0.0$
Countries	S2	AUC-PR	87.95 ± 2.8	87.40 ± 11.7	$\mathbf{93.04} \pm 0.4$
	S3	AUC-PR	48.44 ± 6.3	56.68 ± 17.6	$\mathbf{77.26} \pm 17.0$
Kinship		HITS@1	0.34	0.24	**0.39**
		HITS@10	**0.74**	0.60	0.71
Nations		HITS@1	0.46	**0.48**	0.45
		HITS@10	0.97	0.98	**0.99**
UMLS		HITS@1	0.47	0.47	**0.51**
		HITS@10	0.80	0.79	**0.81**

Table 1: Results on four benchmark knowledge bases. Results on countries are averaged over ten runs with the standard deviation shown next to the AUC.

In future work we plan to scale up the NTP further, and operate directly on natural language.

References

Bouchard, G.; Singh, S.; and Trouillon, T. 2015. On approximate reasoning capabilities of low-rank vector spaces. In *Spring Symposium on Knowledge Representation and Reasoning (KRR)*. Citeseer.

Graves, A.; Wayne, G.; and Danihelka, I. 2014. Neural turing machines. *CoRR* abs/1410.5401.

Kok, S., and Domingos, P. M. 2007. Statistical predicate invention. In *ICML*, 433–440.

Nickel, M.; Murphy, K.; Tresp, V.; and Gabrilovich, E. 2016. A review of relational machine learning for knowledge graphs. *Proceedings of the IEEE* 104(1):11–33.

Rocktäschel, T., and Riedel, S. 2017. End-to-end differentiable proving. *CoRR* abs/1705.11040.