Theory Learning and Logical Rule Induction with Neural Theorem Proving

Andres Campero [1] **Aldo Pareja** [2] **Tim Klinger** [2] **Josh Tenenbaum** [1] **Sebastian Riedel** [3]

1. Introduction

A hallmark of human cognition is the ability to continually acquire and compress observations of the world into meaningful, predictive theories without explicit supervision. This allows us to quickly understand new concepts and make useful predictions about them. For example, we might represent our knowledge of animals in a taxonomic hierarchy. Using such a hierarchy allows us to infer a whole range of new facts about an individual, observing that a Harpy Eagle is a type of Eagle allows us to immediately deduce that a Harpy eagle can fly and breathe. How such representations can be learned from raw observations has been a key problem in semantic knowledge acquisition going back at least to the 1960's in the work of (Collins & Quillian, 1969), with symbolic, bayesian, and neural approaches proposed (Rogers & McClelland, 2004; Hinton, 1986; Yarden et al., 2008). We follow (Yarden et al., 2008) in proposing Theory Learning as a way to address three questions in the development of a solution: (1) how can we induce logical rules from the observations? (2) how can we learn a small set of core facts from which we can infer the observations (and more), and (3) how can this be done without explicit supervision?

Symbolic and neural solutions each have complementary strengths and weaknesses. Symbolic models can learn from very little data and generalize well but are brittle and prone to failure when the observations are noisy as they inevitably are in the real world. They also provide little insight into how their symbolic structure might be learned. Neural models are generally robust to such noise but prone to over-fitting and require large amounts of data to train. They are also difficult to interpret. There is a long history of research in neural-symbolic systems which try to get the best of both worlds, recent examples include (Yang et al., 2017) and (Serafini & Garcez, 2016), for a survey see (Besold et al., 2017). Two relevant recent examples for logical rule induction include (Evans & Grefenstette, 2018) and (Rocktäschel & Riedel, 2017). Both of these approaches offer differentiable models which can be trained using gradient descent, but are interpretable and generalize well with little data. But both suffer scalability issues: (Evans & Grefenstette, 2018) because they must enumerate all pairs of possible rules and (Rocktäschel & Riedel, 2017) because they must build a proof tree which grows exponentially in the depth.

In this paper we present a new rule induction network for logical theory acquisition which can solve Inductive Logic Programming (ILP) tasks but can also take a set of observed facts and learn to compress them into a small set of core facts in addition to the logical rules. The network is neuro-symbolic in the sense that it can learn the logical structure underlying a set of observed facts using dense vector representations for both the atoms of the rules and for the predicates of the facts. The network implements forward chaining and soft unification to recover the observations from the facts under application of the rules. After K steps of forward inference, the consequences are compared to the initial observations and the rules and core facts are encouraged towards representations that more faithfully generate the observations through inference. By encouraging sparsity in the set of core learned facts it can be trained to both induce and compress. Importantly, this gives the model a capability lacking in many ILP approaches but present in the Bayesian literature, to perform inductive inferences of facts. For example, when observing that salmon can swim, and have fins and gills, the model can learn the core fact that salmon are fish even though that is not deducible directly. The learned rule and core fact representations are interpretable and can involve predicate invention. We demonstrate the efficacy of our approach on a variety of ILP rule induction and domain theory learning datasets.

2. Model

In this section we describe the inference network model which is trained using stochastic gradient descent to do rule induction in a standard ILP setting but can also do theory learning through the induction of both a set of core facts and a set of logical rules. By learning a core set of facts from which the observed knowledge can be recovered through inference, we can compress at the same time we generalize. Compression is achieved through a loss term which penalizes the number of initial core facts.

[1]Brain and Cognitive Sciences, MIT, Cambridge, USA. [2]IBM, Yorktown Heights, NY USA. [3]Computer Science, UCL, London, UK. . Correspondence to: A. Campero <campero@mit.edu>.

Published at the ICML workshop Neural Abstract Machines & Program Induction v2 (NAMPI) — Extended Abstract, Stockholm, Sweden, 2018.

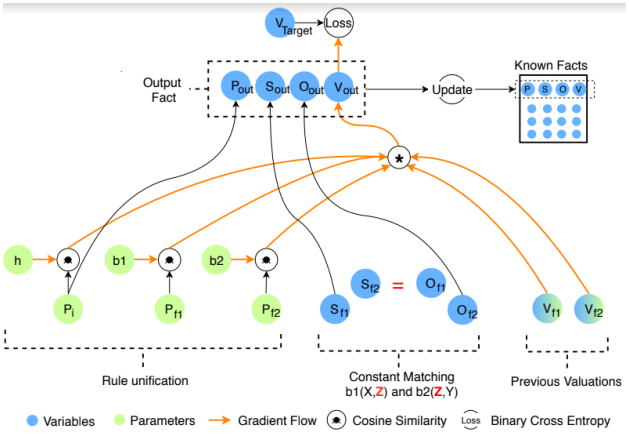

Figure 1. Overview of the Model. Parameters are represented with green balls and constitute the trainable embeddings, orange arrows indicate paths on which gradients flow (in the opposite direction).

Facts are triples (p, s, o), where $p \in \mathbb{N}$ is an index into a dictionary of predicate embeddings P, and $s, o \in \mathbb{C}$ (subject and object) are indices for constants that form the arguments of the facts. To allow for predicate invention, P is provided with auxiliary predicates that might become useful in the logical rules. The embeddings that form the fact predicates can be kept fixed or can be parametrized and modified through learning. A valuation $V : \mathbb{N} \times \mathbb{C}^2 \to [0, 1]$ is a mapping intended to capture the algorithm's belief in the truth of the facts. Logical rules are of the form $h \leftarrow b1, b2, \ldots, b_n$ where h and b_i are atoms with variables as arguments, such as $grandfather(X, Y)$. The head of the rule may have either one or two universally quantified variables. The body must use any variables used in the head and may in addition have existential variables. For example, the body may be of the form $b1(x, z), b2(z, y)$ where x and y are universal and z is existential. We use simple templates which considerably constrain the form of the rules[1]. Every atom of a rule is associated with an embedding that is randomly initialized. During learning, rules acquire their meaning by becoming similar to the dictionary predicates.

The operation of the network is illustrated Figure 1. The network starts with the initial valuations which are updated through K steps of forward inference using the logical rules. To update, the algorithm loops over each rule and sequence of facts making sure that the constants of the facts and the variables of the rules can be matched. For example the rule body $b1(x, z), b2(z, y)$ will unify with $p1(a, b), p2(b, c)$ but not with $p1(a, b), p2(c, d)$. If the fact sequence matches, we iterate over fact predicates to compute new valuations. Multiplication between the unification score for the body, the unification score for the head and the implied valuation is used as a soft form of AND. The unification score of the body is the product of the cosine distance of each of the body atoms from their corresponding facts. The implied valuation is computed using the current valuations for each of the facts being unified with the body. The unification score for a particular predicate is taken with the head. If the implied fact was in the valuation already, it is updated with the max of the previous and the new values (implementing an OR), if it is not, the new fact is appended to the valuation. In this way the valuation is dynamically extended at each step of inference. To train, the K steps of the inference network are composed and the valuation of the final consequences are compared to the valuations of the target using the binary cross entropy loss. The loss gradients are back-propagated to update the predicate embeddings for the rules and for the facts (the predicates of the facts can also be fixed, i.e as one-hot vectors). The rule and fact predicate embeddings are the parameters of the network.

When a set of background facts is given, as in the case of ILP tasks, we initialize the current valuation for the background facts set to 1.0. In the case of Theory Learning, the task additionally includes learning the small set of initial core facts that underly the structure of the observations. Unlike in the ILP setting, we parameterize the valuations by initializing them to 0.5 and train them towards values which allow the model to faithfully recover the observed facts.

3. Experiments and Results

3.1. Predicate Learning ILP Tasks

We test a selection of the ILP problems from (Evans & Grefenstette, 2018) where the task is to learn a target relation from a set of background knowledge facts. Table 1 gives a performance comparison. We see that our algorithm performs considerably better. When the embeddings of the predicate dictionary are fixed as one-hot vectors, our procedure is very similar to theirs, where embedding weights are associated to predicates and search happens at the more compositional level of atoms. The more general case with trainable dense embeddings opens the interesting direction of studying the vector embedding semantic space.

Table 1. Predicate learning tasks. Percentage of successful random weight initializations, $|I|$ is the number of intentional predicates

Task	$\|I\|$	Recursive	∂ILP	Ours
Even	2	Yes	48.5	100
Fizz	3	Yes	10	10
Buzz	2	Yes	35	70
Grandparent	2	No	96.5	100
Cyclic	2	Yes	100	100

[1]These templates constitute the most limiting factor of the current version of our network

3.2. Countries

We are not focused specifically on knowledge base completion but use the COUNTRIES dataset (Bouchard et al., 2015) to evaluate the scalability of our algorithm. The dataset contains 272 constants, 2 predicates and 1158 true facts and we compare with the 3 tasks described in (Rocktäschel & Riedel, 2017) (S1,S2,S3 in table 2). At each training epoch we randomly sample from a section of the knowledge graph both the targets and a set of facts that form the background knowledge.

Table 2. Performance on COUNTRIES dataset

Task	NTP	NTP-λ	Ours
S1	90.83 ± 15.4	100.00 ± 0.0	91.15 ± 15.4
S2	87.40 ± 11.7	94.04 ± 0.4	86.87 ± 3.2
S3	56.68 ± 17.6	77.26 ± 17.0	63.08 ± 28.2

3.3. Learning Theories

We test the capability of our network to compress a set of observations in the form of a theory by learning both a set of core facts in addition to the logical rules. We take the two examples considered by (Yarden et al., 2008). A Taxonomy is a a tree structure where all the observed facts can be recovered from inheritance rules such as $IS(x, y) \leftarrow IS(x, z), IS(z, y)$, and from a small set of direct relations which form the core facts. We report performance on the harder tree from (Rogers & McClelland, 2004, p. 17). The Kinship theory consists on the compression of 6 observed predicates (*mother*, *father*, *daughter*, *wife*, *husband*) into 4 new core predicates (*female*, *male*, *spouse*, *child*) which acquire their meaning through their learned extensions and logical rules (i.e $mother(X, Y) \leftarrow female(X), child(Y, X)$).Table 3 shows the statistics for the observed data and for the target compressed theory as well as the algorithm performance quantified as the percentage of initializations where the rules are successfully learned, the accuracy of the recovered data and the number of learned core facts.

References

Besold, T., Garcez, A., Bader, S., Bowman, H., Domingos, P., Hitzler, P., Kühnberger, K., Lamb, L., Lowd, D., Lima, P., et al. Neural-symbolic learning and reasoning: A survey and interpretation. *arXiv:1711.03902*, 2017.

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

Collins, A.M. and Quillian, M.R. Retrieval time from semantic memory. *Journal of Verbal Learning and Verbal Behavior*, 8:240–248, 1969.

Evans, R. and Grefenstette, E. Learning explanatory rules from noisy data. *Journal of Artificial Intelligence Research*, 61:1–64, 2018.

Hinton, G. Learning distributed representations of concepts. In *Proceedings of the eighth annual conference of the cognitive science society*, 1986.

Rocktäschel, T. and Riedel, S. End-to-end differentiable proving. In *Advances in Neural Information Processing Systems*, pp. 3791–3803, 2017.

Rogers, T. and McClelland, J. *Semantic cognition: A parallel distributed processing approach*. MIT Press, Cambridge, MA, 2004.

Serafini, L. and Garcez, A. Logic tensor networks: Deep learning and logical reasoning from data and knowledge. *arXiv preprint arXiv:1606.04422*, 2016.

Yang, F., Yang, Z., and Cohen, W. Differentiable learning of logical rules for knowledge base reasoning. In *Advances in Neural Information Processing Systems*, pp. 2316–2325, 2017.

Yarden, K., Goodman, N., Kersting, K., and Kemp, C. *Proceedings of the Annual Meeting of the Cognitive Science Society*, 2008.

Table 3. Theory Learning Results

	Taxonomy			Family		
	# Preds	# Const	# Facts	# Preds	# Const	# Facts
Observed	4	36	145	6	10	30
Target Theory	4	36	40	4	10	28
	% Succ	%Acc.	#Facts	% Succ	%Acc.	#Facts
Network	70	99	69	100	96	30.8