Neuro-Symbolic Hierarchical Rule Induction

Claire Glanois[⋆], Zhaohui Jiang[†], Xuening Feng[†], Paul Weng[†], Matthieu Zimmer[*], Dong Li[‡], Wulong Liu[‡], Jianye Hao[‡]

[⋆] ITU Copenhagen,[†] UM SJTU Shanghai [*] Huawei, UK[‡] Huawei Noah's Ark Lab, China

ICML - July 2022

Context

- `Interpretability-Oriented model`
  - Neuro-Logic models; e.g. solutions expressed as first-order logic (FOL) formula
  - `Inductive Logic Programming` (ILP): goal is to find a FOL that explains positive and negative examples given some background knowledge, e.g:

$$\begin{aligned} Even(X) &\leftarrow Zero(X) \\ Even(X) &\leftarrow Even(Y) \wedge Aux(Y, X) \\ Aux(X, Y) &\leftarrow Succ(X, Z) \wedge Succ(Z, Y), \end{aligned} \tag{1}$$

  - `Neuro-symbolic methods`: $\rightsquigarrow$ continuous relaxation of this discrete space of FOL.

Our Model: HRI, Neuro-symbolic hierarchical model

I `Embeddings-based model`: learn embeddings for each predicate [1]
  - ⇝ semantic or visual priors on concepts can be leveraged (via embeddings initialization).

II `New compact yet expressive representation`, through concept of proto-rules, which encompasses multiple meta-rules.[2]
  - ⇝ Generic set templates, not hand-designed for each task.
  - ⇝ Characterize its expressivity (cf. Theorem 1 below).

III `Hierarchical and incremental prior`

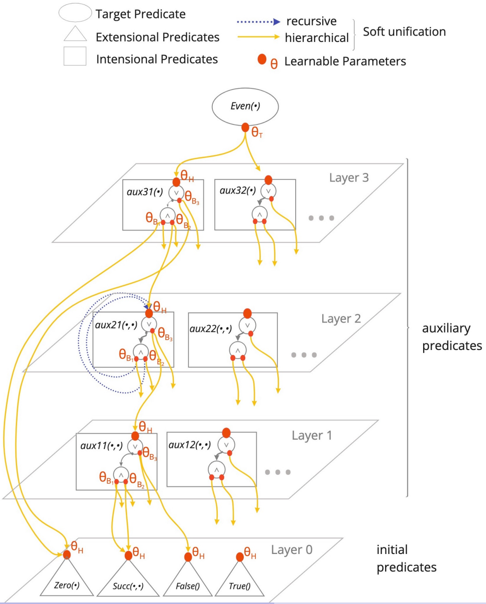
IV `Interpretability-oriented training method`.

V `Rule Induction`. The valuations of predicates are computed via a soft unification between proto-rules and predicates.

[1] Extension of LRI Campero et al. [2018] ; embeddings for both heads and atoms.
[2] Meta-rule corresponds to a second-order clause with predicate variables; e.g.

$$\mathrm{H}(X, Y) \leftarrow \mathrm{B}_1(X, Z) \wedge \mathrm{B}_2(Z, Y)$$

Hierarchical Model



Expressivity Analysis

I Designed an expressive and minimal set of proto-rules $\mathcal{R}_0$:

$$\mathcal{R}_0 := \left\{ \begin{array}{l} \mathfrak{A} : \mathrm{H}(X) \quad\quad \leftarrow \overline{\mathrm{B}}_1(X, Y) \wedge \overline{\mathrm{B}}_2(Y, X) \\ \mathfrak{B} : \mathrm{H}(X, Y) \leftarrow \overline{\mathrm{B}}_1(X, Z) \wedge \overline{\mathrm{B}}_2(Z, Y) \\ \mathfrak{C} : \mathrm{H}(X, Y) \leftarrow \overline{\mathrm{B}}_1(X, Y) \wedge \overline{\mathrm{B}}_2(Y, X) \end{array} \right\}$$

II Characterize its expressivity:

Theorem 1

The hypothesis space generated by $\mathcal{R}_0$ from $\mathcal{P}$ is exactly the set of function-free definite Horn clause fragment $\mathcal{F}_{\mathcal{P},\leq 2}^{\{1,2\}}$ composed of clauses with at most two body atoms involving unary and binary predicates in $\mathcal{P}$.

Experiments

Empirical validation on various domains:

I classical ILP benchmark tasks

II large domain GQA (Hudson and Manning [2019]) extracted from Visual Genome (Krishna et al. [2017])

III RL tasks: block manipulation tasks: Stack, Unstack, and On (cf. Jiang and Luo [2019])

Comparison with various models:

I Neuro-Symbolic Models e.g. dILP (Evans and Grefenstette [2018]), NLM (Dong et al. [2019]), DLM (Zimmer et al. [2021]), NLRL (Jiang and Luo [2019]), NLIL (Yang and Song [2020])

II Traditional ILP Methods e.g. ILASP (Law et al. [2020]) and Popper ([Cropper and Morel, 2021])

Experimental Results 1/2

Table 1. Percentage of successful runs among 10 runs. $|I|$ is the smallest number of intensional predicates needed. Recursive means whether or not the solution needs to learn recursive rules.

Task	$\|I\|$	Recursive	∂ILP	LRI	Ours		
					train	soft evaluation	symbolic evaluation
Undirected Edge	1	No	100	100	100	100	100
Member	1	Yes	100	100	100	100	100
Connectedness	1	Yes	100	100	100	100	100
Grandparent	2	No	96.5	100	100	100	100
Adjacent to Red	2	No	50.5	100	100	100	100
Two Children	2	No	95	0	100	100	100
Graph Coloring	2	Yes	94.5	0	100	100	100
Even-Succ2	2	Yes	48.5	100	40	40	40
Buzz	2	Yes	35	70	100	40	40

Table 2. Comparisons with NLM/DLM in terms of percentage of successful runs and average training times over 10 runs.

Task	#Training constants	% successful runs			Training time (secs)		
		NLM	DLM	Ours	NLM	DLM	Ours
Adjacent to Red	7	100	90	100	163	920	62
	10	90	90	100	334	6629	71
Grandparent	9	90	100	100	402	2047	79
	20	100	100	100	1441	3331	89

Table 3. R@1 and R@5 for 150 objects classification on VG.

Model	Visual Genome	
	R@1	R@5
MLP+RCNN	0.53	0.81
Freq	0.40	0.44
NLIL	0.51	0.52
Ours	0.53	0.60

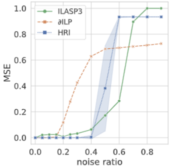
Experimental Results 2/2

Table 4. Comparisons with NLRL/NLM/DLM in terms of rewards on the testing scenarios.

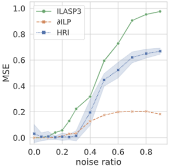
Task	Rewards			
	NLRL	NLM	DLM	Ours
Unstack	0.914	0.920	0.920	0.920
Stack	0.877	0.920	0.920	0.920
On	0.885	0.896	0.896	0.896

Table 5. Performance of different embedding initializations for the single Visual Genome task. Displaying both the soft and the symbolic evaluations (once the symbolic rule has been extracted).

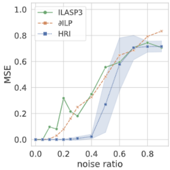
Init.	Accuracy		Precision		R@1	
	soft	symb.	soft	symb.	soft	symb.
Random	0.63	0.49	0.57	0.5	0.23	0.38
NLIL	0.75	0.6	0.87	0.75	0.46	0.58
GPT2	0.65	0.45	0.72	0.66	0.27	0.5



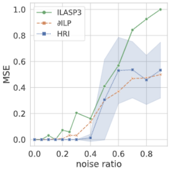
(a) Predecessor



(b) Member



(c) Connectedness



(d) Undirected Edge

Figure 2. Mean Squared Error (MSE) w.r.t. different noise ratios.

Conclusion

HRI, new neuro-symbolic interpretable model performing hierarchical rule induction through soft unification with learned embeddings.

- ⇝ Initialized by a theoretically-backed small-yet-expressive set of proto-rules, able tackle many classical ILP benchmark tasks.
- ⇝ Efficiency and performance in ILP, RL, and richer domain against state-of-the-art baselines: typically one to two orders of magnitude faster to train.
- ⇝ Can leverage semantic or visual priors, and manifest some combinatorial generalisation
- ⇝ Future extensions could extend proto-rule set to broaden model expressivity for further RL domains and continual learning scenario.

References

A. Campero, A. Pareja, T. Klinger, J. Tenenbaum, and S. Riedel. Logical rule induction and theory learning using neural theorem proving. *arXiv preprint arXiv:1809.02193*, 2018.

A. Cropper and R. Morel. Learning programs by learning from failures. *Machine Learning*, 110(4):801–856, 2021.

H. Dong, J. Mao, T. Lin, C. Wang, L. Li, and D. Zhou. Neural Logic Machines. In *International Conference on Learning Representations*. OpenReview.net, 2019.

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

D. A. Hudson and C. D. Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6693–6702, 2019.

Z. Jiang and S. Luo. Neural Logic Reinforcement Learning. In *International Conference on Machine Learning*, Apr. 2019.

R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma, M. S. Bernstein, and L. Fei-Fei. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *Int. J. Comput. Vision*, 123(1):32–73, May 2017.

M. Law, A. Russo, and K. Broda. The ilasp system for inductive learning of answer set programs. *arXiv:2005.00904 [cs]*, May 2020.

Y. Yang and L. Song. Learn to explain efficiently via neural logic inductive learning. In *International Conference on Learning Representations*, 2020.

M. Zimmer, X. Feng, C. Glanois, Z. Jiang, J. Zhang, P. Weng, L. Dong, H. Jianye, and L. Wulong. Differentiable logic machines. *arXiv preprint arXiv:2102.11529*, 2021.

Appendix: Model Inference

The valuations of predicates are computed via a soft unification between proto-rules and predicates using learnable embeddings.
More precisely, one inference step in our model is formulated as follows:

$$\begin{aligned} v_{and} &= \mathrm{POOL}_{\mathsf{P}_1,\mathsf{P}_2}\left(\alpha_{P_1\mathrm{B}_1}\cdot\alpha_{P_2\mathrm{B}_2}\cdot\mathrm{AND}[v_{\mathsf{P}_1},v_{\mathsf{P}_2}]\right) \\ v_{or} &= \mathrm{OR}\left[v_{and},\mathrm{POOL}_{\mathsf{P}_3}\left(\alpha_{P_3\mathrm{B}_3}\cdot v_{\mathsf{P}_3}\right)\right] \\ v &= \mathrm{MERGE}\left(v_{old},v_{or}\right), \end{aligned} \tag{2}$$

where v (resp. v_{old}) denotes the new (resp. old) valuation of a grounded auxiliary predicate P. For an auxiliary predicate at layer ℓ, the POOL operation encompass a max over the groundings compatible to P followed by a pooling performed over both predicates $P_1, P_2, P_3 \in \mathcal{P}_\ell^{\downarrow}$.