Neuro-symbolic approach to reinforcement learning in robotics

Bc. Tomáš Bisták
Supervisor: prof. Ing. Igor Farkaš, Dr.
Consultant: doc. RNDr. Martin Homola, PhD.

Faculty of Mathematics, Physics and Informatics
Comenius University Bratislava

May 17, 2024

Motivation

- successful reinforcement-learning approaches are mostly centered around (deep) neural networks

  Advantages
  - learning from raw inputs (image sequences, sensor readings)
  - acting under uncertainty even in continuous spaces
  - solving relatively complex tasks with little to no prior knowledge

  Disadvantages
  - lacking interpretability (a description of decision processes) and explainability (the reasoning behind choices)
  - limited generalization
  - low data efficiency

- symbolic methods can mitigate the issues with neural networks

Aim

- integrate symbolic and subsymbolic methods into a single neuro-symbolic system for reinforcement learning
  - leverage the learning capabilities of neural networks
  - improve interpretability, explainability, and generalization

Reinforcement Learning (RL)

- informally: an agent interacting with an environment, striving to learn the most optimal behavior
- formulated as a Markov decision process:
  - *state* – a description of the current configuration of the agent in the environment
  - *action* – an atomic interaction of the agent with the environment, usually resulting in a transition to another state
  - *reward* – a real number presented to the agent after each action assessing its quality (may be positive, zero, or negative)
  - *policy* – a (possibly stochastic) mapping from states to actions that the agent takes in the respective states
- goal: learn a policy following which maximizes the expected cumulative reward collected during interaction

Hierarchical Reinforcement Learning (HRL)

- consider episodic tasks, i.e., with a clearly defined goal state
- policies can be learned on multiple levels – at least two
  - higher level – learn to select subgoals leading to the ultimate goal
  - lower level – learn to reach the respective subgoals
- facilitates the integration of symbolic and subsymbolic methods
  - higher level (symbolic) – select abstract actions/subgoals
  - lower level (subsymbolic) – realize the abstract actions

Neuro-Symbolic RL with Templates

Neural Logic Reinforcement Learning (NLRL) [Jiang & Luo, 2019]

- policy = a set of logical rules of the form $action \leftarrow preconditions$
- rules generated in advance via templates
- weights of the rules (importance) optimized through RL

Programmatically Interpretable Reinforcement Learning (PIRL) [Verma et al., 2018]

- also uses templates but learns a program in a functional language
- searches the program space with a pre-trained neural policy

Neurally Guided Differentiable Logic Policies (NUDGE) [Delfosse et al., 2023]

- combines concepts from NLRL and PIRL
- a pre-trained neural policy guides the initial generation of logical rules

Removing Templates in General

- downsides of the mentioned methods: templates (generalization / scalability), neural guidance (transparency)
- solution: learn rules dynamically and from scratch through an entirely differentiable procedure

Differentiable Neural Logic Networks (dNL)
[Payani & Fekri, 2019]

- fuzzy semantics (truth values from $[0, 1]$)
- multi-layer perceptrons (MLP) with special activation functions to represent conjunction (AND) and disjunction (OR)
- weights optimized w.r.t. truth values for input atoms (propositions)
- forward chaining for multi-step reasoning ($\{p \Rightarrow q, p\} \models q$)

Neural Logic Machines (NLM) [Dong et al., 2019]
Logical Neural Networks (LNN) [Riegel et al., 2020]

- more complex, advanced implementations of the idea behind dNL

Removing Templates in RL

- Kimura et al. used LNNs with a simple AND-OR architecture for text-based games [Kimura et al., 2021]
  - for each action: one LNN – outputs a single truth value (certainty)
  - first layer: AND, second layer: OR $\rightsquigarrow$ disjunctive normal form
  - greedy action selection based on all outputs

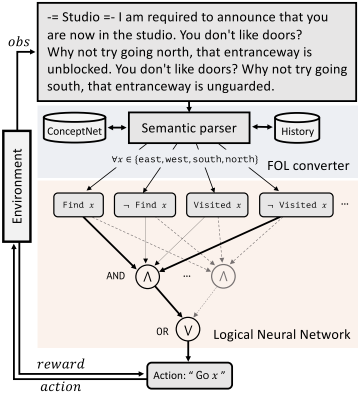

Figure: Approach overview.

Our First Prototype

Environment

- ultimate goal: teach a robotic arm to build a specified structure (e.g., a stack) from blocks on a table
- a purely symbolic environment at first
  - constants – table and blocks (a, b, c, ...)
  - initial state – all blocks on table
  - goal state – all blocks in one stack in the order a, b, c, ...
  - rewards – for building the complete stack ($+1$), building a substack ($+1$), destroying a substack (-1.1), making any other action (-0.1)

Form	Semantics
$top(X)$	block X is on top of a stack
$on(X, Y)$	block X is stacked on block $Y/table$ ($X \neq Y$)
$move(X, Y)$	stack block X onto block $Y/table$ ($X \neq Y$)

Table: State and action atoms.

Our First Prototype

Agent

- for each action: an AND-OR MLP with activation functions from dNL
- outputs from all MLPs $\rightsquigarrow$ probability distribution over actions
- learning: REINFORCE, back-propagation (with gradient ascent)

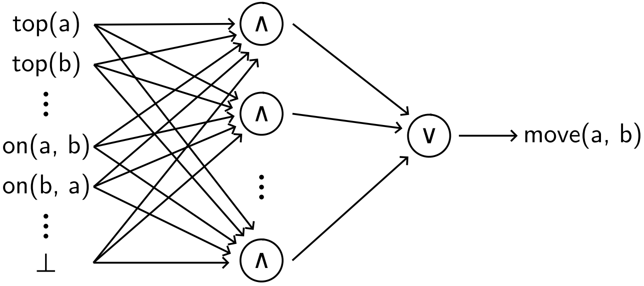

Figure: MLP architecture for a single action.

Our First Prototype

Results

- a nearly optimal policy learned for up to four blocks
  - for $move(b, a)$, the relevant inputs (with weights greater than 0.1) created a rule:
    $$\begin{aligned} move(b,a) \leftarrow\ & top(a)_{0.995}, top(b)_{0.916}, top(c)_{0.629}, top(d)_{0.489}, \\ & on(a, table)_{0.644}, on(b, table)_{0.984}, \\ & on(c, table)_{0.936}, on(d, table)_{0.611} \end{aligned}$$
  - for $move(c, a)$, which is unnecessary, all inputs (even $\perp$) were given weights above 0.5 $\rightsquigarrow$ contradiction
- challenges: generalization and scalability

Generalization and Explainability

- our first prototype cannot generalize to
  - situations other than the one it was trained on
  - environments with more blocks
- an agent with a single MLP for all actions
  - $move(b, c)$ and $move(a, b)$ equally important in the initial state
- more robust generalization requires quantification and actual planning
  - embedding all knowledge into preconditions may lead to arbitrarily complex formulas (questionable viability)
  - a plan is probably the only true explanation

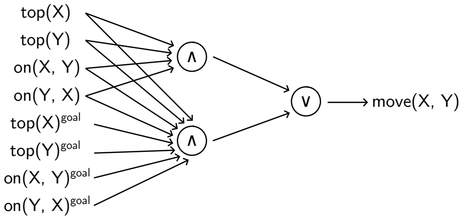

Figure: MLP architecture for a generic action.

Future Work

- learn preconditions and usefulness/ordering of actions separately
  - infer preconditions and effects (transition model) directly from interactions
  - use symbolic planning to construct a reasonable plan (based on the collected rewards)

Symbolic Deep Reinforcement Learning (SDRL)
[Lyu et al., 2019]

- preconditions and effects are given
- a symbolic planner creates a plan
- a neural policy tries to realize the plan

Hierarchical Reinforcement Learning using Inductive Logic Programming [Xu & Fekri, 2021]

- learns a symbolic transition model from experience
- performs a simple value-based selection of the next subtask

Thank You

References I

Jiang, Z. & Luo, S. (2019)
Neural Logic Reinforcement Learning
Proceedings of the 36th International Conference on Machine Learning, PMLR, 97, 3110–3119.

Verma, A., Murali, V., Singh, R., Kohli, P., & Chaudhuri, S. (2018)
Programmatically Interpretable Reinforcement Learning
Proceedings of the 35th International Conference on Machine Learning, PMLR, 80, 5045–5054.

Delfosse, Q., Shindo, H., Dhami, D., & Kersting, K. (2023)
Interpretable and Explainable Logical Policies via Neurally Guided Symbolic Abstraction
Advances in Neural Information Processing Systems, 36, 50838–50858.

Payani, A. & Fekri, F. (2019)
Inductive Logic Programming via Differentiable Deep Neural Logic Networks
arXiv.

References II

- Dong, H., Mao, J., Lin, T., Wang, Ch., Li, L., & Zhou, D. (2019)
  Neural Logic Machines
  International Conference on Learning Representations (ICLR 2019), 4, 2574–2595.
- Riegel, R., Gray, A., Luus, F., Khan, N., Makondo, N., Akhalwaya, I. Y., Qian, H., Fagin, R., Barahona, F., Sharma, U., Ikbal, S., Karanam, H., Neelam, S., Likhyani, A., & Srivastava, S. (2020)
  Logical Neural Networks
  arXiv.
- Kimura, D., Ono, M., Chaudhury, S., Kohita, R., Wachi, A., Agravante, D. J., Tatsubori, M., Munawar, A., & Gray, A. (2021)
  Neuro-Symbolic Reinforcement Learning with First-Order Logic
  Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, 3505–3511.
- Lyu, D., Yang, F., Liu, B., & Gustafson, S. (2019)
  SDRL: Interpretable and Data-Efficient Deep Reinforcement Learning Leveraging Symbolic Planning
  Proceedings of the AAAI Conference on Artificial Intelligence, 33(01), 2970–2977.

References III

Xu, D. & Fekri, F. (2021)
Interpretable Model-based Hierarchical Reinforcement Learning using Inductive Logic Programming
arXiv.