Towards Learning Abstractions via Reinforcement Learning

Erik Jergéus[1,*,†], Leo Karlsson Oinonen[1,*,†], Emil Carlsson[1] and Moa Johansson[1]

[1]*Chalmers University of Technology, Gothenburg, Sweden*

Abstract

In this paper we take the first steps in studying a new approach to synthesis of efficient communication schemes in multi-agent systems, trained via reinforcement learning. We combine symbolic methods with machine learning, in what is referred to as a neuro-symbolic system. The agents are not restricted to only use initial primitives: reinforcement learning is interleaved with steps to extend the current language with novel higher-level concepts, allowing generalisation and more informative communication via shorter messages. We demonstrate that this approach allow agents to converge more quickly on a small collaborative construction task.

Keywords

Reinforcement learning, Multi Agent Systems, Neuro-Symbolic Systems, Emergent Communication

1. Introduction

Learning to communicate and coordinate efficiently via interactions, rather than relying on solely supervised learning, is often viewed as a prerequisite for developing artificial agents able to do complex machine-to-machine and machine-to-human communication [1]. The field of language learning and emergent communication has a long history [2, 3, 4, 5, 6], and is now a vibrant field of research also in the deep learning community [7, 8, 9, 10]. Recent work has focused on developing agents with single message communication [11, 12, 13], variable length communication [14] and compositional language [15, 16], via interactions and reinforcement learning. However, a striking characteristic of human communication that has been overlooked in the literature is the ability to derive novel concepts and abstractions from primitives, via interaction.

In this paper, we investigate how artificial agents can develop linguistic abstractions via interaction and reinforcement learning, starting from a small set of primitive concepts and gradually increasing the size and efficiency of their language over time. Our motivation is the builder-architect experiment in [17], investigating how humans develop communicative abstractions. Here, the architect is given a drawing of a shape, and has to instruct the builder

AIC 2022, 8th International Workshop on Artificial Intelligence and Cognition

*Corresponding author.

†These authors contributed equally.

erikjer.student@chalmers.se (E. Jergéus); leoo.student@chalmers.se (L. K. Oinonen); caremil@chalmers.se (E. Carlsson); jomoa@chalmers.se (M. Johansson)

0000-0003-2231-6869 (E. Jergéus); 0000-0003-4117-5096 (L. K. Oinonen); 0000-0002-0170-7898 (E. Carlsson); 0000-0002-1097-8278 (M. Johansson)

CEUR Workshop Proceedings (CEUR-WS.org)

how to construct it from small blocks. As the experiment progressed, participants developed more concise instructions after repeated attempts. Instead of talking about the positions of individual blocks, they started using abstractions describing commonly seen shapes, such as *L-shape* or *upside-down U*, see Figure 1. Our contribution here is an initial feasibility study of a neuro-symbolic multi-agent reinforcement learning framework for this task. Inspired by neuro-symbolic program synthesis [18], the agent interleave reinforcement learning to train their neural network, with symbolic reflection to introduce new concepts for common action sequences. We show that agents learn to reconstruct the given shapes faster when allowed the capability to introduce abstractions.

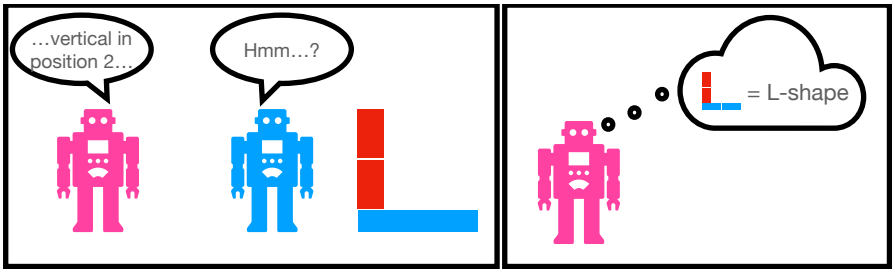

Figure 1: Agents should periodically reflect on their experience and consider introducing abstractions, allowing shorter utterances for constructing commonly occurring shapes.

2. Implementation

Our setup mimics the one from McCarty et al. [17] where two agents, the architect and the builder, communicate about a set of geometric shapes. The agents iterate between two learning phases, one where the agents use reinforcement learning to develop the meaning of each message, followed by an abstraction phase, where the architect may introduce new instructions for commonly seen structures. This will allow the agents to potentially solve the tasks using fewer messages, which gives them a higher reward as shorter interactions are preferred.

2.1. The Environment

The architect's input is a picture of the goal state alongside the current state, each of which is represented by binary 6x6 matrices, see Figure 2. Locations where there are blocks are represented as 1's and empty locations by 0. This is passed through a feed forward neural network, which outputs a message with instructions to the builder. The work described here focus on the learning of the architect, with the builder assumed to understand the architect's messages perfectly. In the future, the builder will also be represented with a neural-network, learning to map messages to the corresponding (sequence of) actions. Initially, the architect's message-space consists of 12 messages, simply the six possible positions of vertical (2 x 1) and horizontal (1 x 2) blocks respectively. We denote the set of messages as $\mathcal{M} = \{V_1, H_1, ..., V_6, H_6\}$. These initial messages have a one-to-one correspondence to the basic actions the builder can perform. Note

Figure 2: The architect sees both the goal and the current state and decides to instruct the builder to place a vertical block in position 4.

that the architect is allowed to introduce new messages, abstractions, during the abstraction phase, formally introduced in later sections.

Reward Function The agents receive a reward R at each time step t when performing an action a, either a larger reward if the new state matches the goal exactly, or a smaller reward if the most recently placed block partially matches the goal. This reward function is given in Equation 1, where *partial_match* denote the number of new grid squares covered by the most recently placed block matching the goal.

$$R_t(s, g, a) = (0.1 * partial\_match + 1 * (s == g)) * 0.9^t \tag{1}$$

The architect becomes better at generalising what placing a single block entails when receiving an intermediate reward based on how much of that block contributes to the final goal. The larger reward from completing the whole structure biases the architect to always aim for a perfect completion of the goal. Finally, we encourage the architect to always use as few messages as possible (i.e. using abstractions) by discounting the reward based on the number of time-steps further.

2.2. Deep Reinforcement Learning

The architect is modelled as a Deep Q-Network (DQN) with experience replay [19]. In short, this means that the architect, for each state and message, (s, m), estimates the Q-value, or expected cumulative reward, for conveying message m given state s. We consider a neural network with layers of sizes $[72, 576, 576, 576, 36, |\mathcal{M}_{max}|]$, where $\mathcal{M}_{max}$ is the maximum allowed size of the message space, and we use ReLU activation between each layer. See GitHub[1] for the other hyper-parameters and implementation.

2.3. Abstraction Phase

In order to learn abstractions, we implement a version of the wake-sleep-dream framework used in the neuro-symbolic system DreamCoder [18]. After a *wake-phase* where the agents have engaged in reinforcement learning to learn to communicate using the current set of messages, the architect enters the *sleep-phase*, where it is allowed to invent new messages. During the

[1]https://github.com/jerge/MARL/tree/Communicative-Abstractions

sleep phase, the architect searches for the longest common sub-sequence(s) of messages from the previous reinforcement learning phase. The sub-sequence's are rated based on their length and frequency and the top-rated sequence will turn into a new abstraction.

Next follows a *dream phase*, which aims to quickly train the agents to use the new abstraction generated from the sleep phase. This is done by letting the agent re-experience the examples from the replay buffer, but now with the new abstraction instead of the corresponding sub-sequence of messages. This will lead the new abstraction towards the appropriate Q-value before starting the next reinforcement learning phase.

3. Experimental Results

We conducted an initial feasibility experiment for our framework: Given a set of three re-occurring shapes from McCarthy's human experiment (Fig. 3) [17], does the agents learn to reconstruct them faster if allowed to introduce abstractions?

We hypothesise that:

a) Having a language with messages also corresponding to common sequences of actions will facilitate the reinforcement learning construction task.

b) Our neuro-symbolic agent can discover and learn to use such concepts.

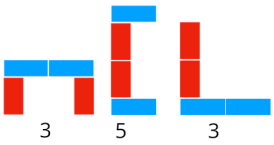

Figure 3: Shapes in our experimental set: 3 *upside-down U*, 5 *C*-shapes and 3 *L*-shapes in all possible different locations in the 6x6 grid.

Agents were first pre-trained on simple randomly generated shapes to learn the basic message/action pairs. To establish an upper and lower bound on the agents' performance, we evaluated one instance without abstraction capabilities (worst case, Fig. 4), and one instance where optimal abstractions for the shapes were already given upfront (best case, Fig. 5). The differences are large: in the worst case the agents required 350 000 epochs to successfully learn to construct all shapes, compared to 17 500 epochs in the best case. There is a clear advantage in having a richer language.

Next, we evaluated the complete architect-agent with ability to introduce abstractions, as shown in Fig. 6. The agents learned to solve the construction task after 160 000 epochs, which is still considerably faster than the worst-case scenario. Note that the architect choose to introduce only two abstractions, the *upside-down U* very early on, and the *C*-shape towards the end. Investigating this behaviour in more detail is further work.

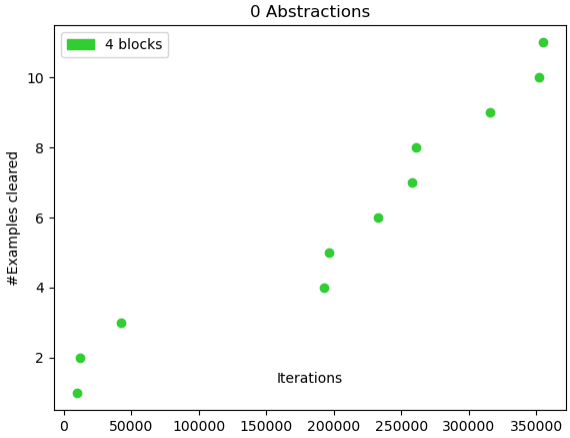

Figure 4: Worst-case bound: without the capability to create abstractions, using only initial primitives, learning to build all shapes requires over 350 000 epochs.

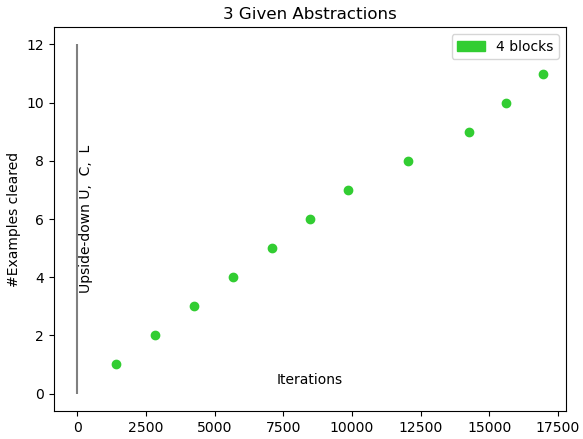

Figure 5: Best-case bound: If agents are given the relevant abstractions upfront, the task can be learned in 17 500 epochs.

4. Conclusion and Future Work

In this work, we have introduced a neuro-symbolic framework for learning linguistic abstractions via a combination of reinforcement learning, symbolic reasoning and interactions between agents. Our initial results on a small collaborative building task suggest that it is feasible for reinforcement learning agents to develop useful abstractions by alternating between neural learning, and symbolic abstraction phases introducing new concepts. These introduced abstract concepts also greatly improve the performance of the agents.

This is just a first step and we would like to further explore how to learn abstractions via reinforcement learning. One interesting direction is to extend our work to more complex

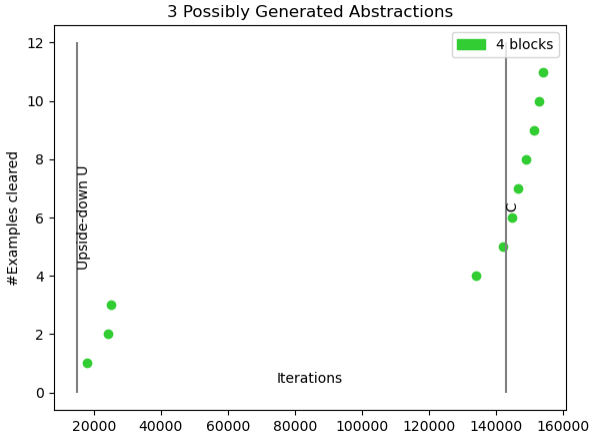

Figure 6: With abstraction-invention, the agents need 160 000 epochs to learn to build all shapes. The horizontal lines mark when the abstraction *upside-down U* and *C*-shape were introduced.

environments. One issue that might arise in such scenarios is that the agents might need to first develop several intermediate abstractions, before being able to construct abstractions that greatly improves the reward. Solving this type of exploration-exploitation dilemma seems like a fundamental problem for the agents, and might require new exploration techniques.

Another interesting future direction is to explore scenarios where agents do not share exactly the same understanding of a message, and are required to reason about each other in a recursive fashion.

Acknowledgement Erik and Leo received a Lars Pareto Travel Grant from Chalmers University of Technology to present this work. Emil Carlsson was supported by CHAIR (Chalmers AI Research), and Moa Johansson was supported by the Wallenberg Al, Autonomous Systems and Software Program - Humanities and Society (WASP-HS) funded by the Marianne and Marcus Wallenberg Foundation and the Marcus and Amalia Wallenberg Foundation.

References

[1] T. Mikolov, A. Joulin, M. Baroni, A roadmap towards machine intelligence, in: Computational Linguistics and Intelligent Text Processing, 2018, pp. 29–61.

[2] T. Hashimoto, T. Ikegami, Emergence of net-grammar in communicating agents., Bio Systems 38 1 (1996) 1–14.

[3] S. Kirby, J. R. Hurford, The Emergence of Linguistic Structure: An Overview of the Iterated Learning Model, Springer-Verlag, Berlin, Heidelberg, 2002, pp. 121–148.

[4] K. Smith, S. Kirby, H. Brighton, Iterated learning: A framework for the emergence of language, Artificial life 9 (2003) 371–86.

[5] L. Steels, The emergence and evolution of linguistic structure: from lexical to grammatical communication systems, Connection science 17 (2005) 213–230.

[6] L. L. Steels, The Talking Heads experiment, number 1 in Computational Models of Language Evolution, Language Science Press, Berlin, 2015.
[7] J. Foerster, I. A. Assael, N. de Freitas, S. Whiteson, Learning to communicate with deep multi-agent reinforcement learning, in: Advances in Neural Information Processing Systems, volume 29, 2016.
[8] A. Lazaridou, M. Baroni, Emergent multi-agent communication in the deep learning era, 2020.
[9] F. Hill, A. K. Lampinen, R. Schneider, S. Clark, M. Botvinick, J. L. McClelland, A. Santoro, Environmental drivers of systematicity and generalization in a situated agent, in: 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020, 2020.
[10] F. Hill, O. Tieleman, T. von Glehn, N. Wong, H. Merzic, S. Clark, Grounded language learning fast and slow, 2020. URL: https://arxiv.org/abs/2009.01719.
[11] E. Jorge, M. Kågebäck, F. D. Johansson, E. Gustavsson, Learning to Play Guess Who? and Inventing a Grounded Language as a Consequence (2016).
[12] A. Lazaridou, A. Peysakhovich, M. Baroni, Multi-agent cooperation and the emergence of (natural) language, in: 5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings, 2017, pp. 1–11.
[13] M. Kågebäck, E. Carlsson, D. Dubhashi, A. Sayeed, A reinforcement-learning approach to efficient communication, PLoS ONE 15 (2020) 1–26.
[14] S. Havrylov, I. Titov, Emergence of language with multi-agent games: Learning to communicate with sequences of symbols, in: Advances in Neural Information Processing Systems, volume 30, 2017.
[15] I. Mordatch, P. Abbeel, Emergence of grounded compositional language in multi-agent populations, 2018.
[16] J. Mu, N. Goodman, Emergent communication of generalizations, in: Advances in Neural Information Processing Systems, 2021.
[17] W. McCarthy, R. Hawkins, C. Holdaway, H. Wang, J. Fan, Learning to communicate about shared procedural abstractions, in: Proceedings of the 43rd Annual Conference of the Cognitive Science Society, 2021.
[18] K. Ellis, C. Wong, M. Nye, M. Sablé-Meyer, L. Morales, L. Hewitt, L. Cary, A. Solar-Lezama, J. B. Tenenbaum, DreamCoder: Bootstrapping inductive program synthesis with wake-sleep library learning, in: Proceedings of the 42nd ACM SIGPLAN International Conference on Programming Language Design and Implementation, 2021, p. 835–850.
[19] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, M. Riedmiller, Playing atari with deep reinforcement learning (2013).