Reinforcement Learning for Production-based Cognitive Models

Adrian Brasoveanu (UC Santa Cruz), Jakub Dotlačil (Utrecht University)
abrsvn@gmail.com, j.dotlacil@gmail.com

I. Learnability of Mechanistic Processing Models. We introduce a framework in which we can start exploring in a computationally explicit way how complex, mechanistically specified cognitive models of linguistic skills (e.g., the parsers in Lewis and Vasishth 2005; Hale 2011; Engelmann 2016) can be acquired. Linguistic cognitive model learnability is an understudied issue, primarily because computationally explicit cognitive models are only starting to be more widely used in psycholinguistics. Cognitive models for linguistic skills pose this learnability problem much more starkly than models for other 'high-level' cognitive processes, since cognitive models that use theoretically-grounded linguistic representations and processes call for richly structured representations and complex rules that require a significant amount of hand-coding.

The learnability problem for production-rule based models can be divided into 2 parts: (*i*) rule acquisition: forming complex rules out of simpler ones, and (*ii*) rule ordering: deciding which rule to fire when. ACT-R's (Anderson and Lebiere, 1998) partial answers to these problems are production compilation for (*i*), and rule-utility estimation for (*ii*). Neither solution has been systematically applied to the complex models for linguistic skills. We focus here on the easier problem (*ii*) and show how advances in the machine learning subfield of Reinforcement Learning (RL, Sutton and Barto 2018) can be leveraged to solve it. RL and ACT-R have very close connections (Fu and Anderson, 2006), but they have remained largely unexplored.

II. Example: Learning goal-conditioned rules for lexical decision (LD). We show how a Q-learning agent (a model-free off-policy learning algorithm; Watkins 1989; Watkins and Dayan 1992; Mnih and al 2015) can be used to learn goal-conditioned rules in an ACT-R based cognitive model of LD tasks. LD tasks can be modeled in ACT-R with a small number of rules (Brasoveanu and Dotlačil, 2019), so they are a good starting example. We model **three LD tasks** of increasing length (hence difficulty): 1 stimulus (the word *elephant*), 2 stimuli (the word *elephant* and a non-word), and 4 stimuli (*elephant*, non-word, *dog*, another non-word).

The model components are split between declarative memory, which stores the lexical knowledge of an English speaker, and procedural memory, which stores rules that enable the model to carry out the LD task. The rules are conditionalized actions: they fire/execute actions when their conditions are satisfied by the cognitive state of the ACT-R mind (the buffers). We assume 4 rules, provided in standard ACT-R format on the next page. These 4 rule were originally hand-coded to fire serially. Assume the initial goal state of the ACT-R model is `retrieving`, and the word *elephant* appears on the virtual screen of the model, which is automatically stored in the value slot of the visual buffer. **Rule 1** fires, attempting to retrieve a word with the form *elephant* from declarative memory, and the goal state is updated to `retrieval-done`. When the word is successfully retrieved, **Rule 2** fires and the `J` key is pressed. At that point: (*i*) in the 1-stimulus task, the text *FINISHED* is displayed on the screen, then **Rule 4** fires and ends the task; (*ii*) in the 2/4-stimuli tasks, a non-word is displayed, then **Rule 1** fires; the retrieval attempt fails (cannot retrieve a non-word), so **Rule 3** fires and the `F` key is pressed, after which the next text (*FINISHED* or *dog*) is displayed, etc.

Instead of hand-coding the rule conditions, we only specify the actions – that's the reason for striking out the goal specifications – and let the Q-learning agent learn to successfully carry out the LD task. We give the agent a reward of 1 if it reaches the final goal-state `done`; for any intermediate rule firing, we give it a small negative reward of -0.15 to encourage it to finish the task asap. The agent learns by trial and error to successfully carry out the LD tasks, i.e., to properly order the rules to efficiently complete LD tasks. **Learning is faster and better for shorter tasks** (fewer stimuli). In the presentation, we will conclude with a discussion of possible ways to model learning from instructions, curriculum learning (progressing from simpler tasks to difficult ones), other RL algorithms, and limitations of RL approaches to more realistic cognitive tasks, e.g., LDs task with hundreds of stimuli (presented in random order) or parsing tasks.

(1) **Rule 1: Retrieving**

~~goal>~~ | ~~state:~~ ~~`retrieving`~~ | $\Longrightarrow$ goal> | state: `retrieval-done` |

[stricken out b/c the agent learns goal conditions]

visual> | value: =val | +retrieval> | isa: word |
| value: ∼FINISHED | | form: =val |

(2) **Rule 2: Lexeme Retrieved**

~~goal>~~ | ~~state:~~ ~~`retrieval-done`~~ | $\Longrightarrow$ goal> | state: `retrieving` |

retrieval> | buffer: full | +manual> | cmd: press-key |
| state: free | | key: J |

(3) **Rule 3: No Lexeme Found**

~~goal>~~ | ~~state:~~ ~~`retrieval-done`~~ | $\Longrightarrow$ goal> | state: `retrieving` |

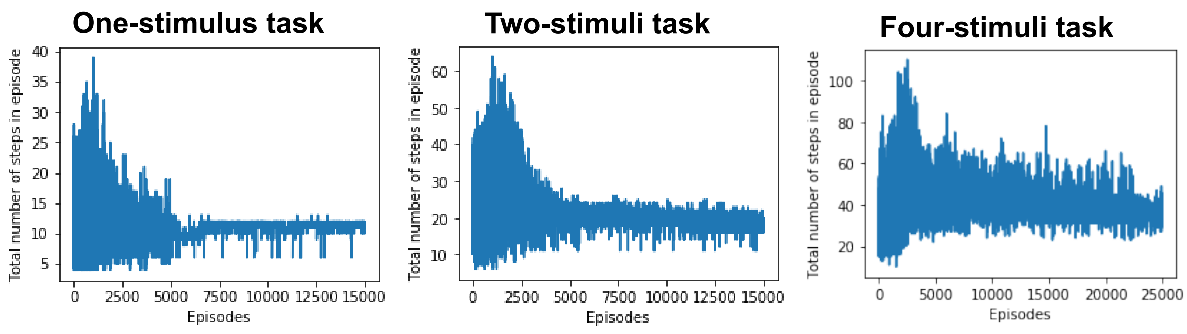
retrieval> | buffer: empty | +manual> | cmd: press-key |
| state: error | | key: F |

(4) **Rule 4: Finished**

~~goal>~~ | ~~state:~~ ~~`retrieving`~~ | $\Longrightarrow$ goal> | state: `done` |

visual> | value: FINISHED |

More details about the ACT-R and RL models



One-stimulus task. We simulate $15{,}000$ episodes, i.e., $15{,}000$ lexical decision tasks consisting of 1 stimulus only (the word 'elephant'), from which the Q-learning agent learns. The leftmost plot above shows that, after about $5{,}000$ episodes, the task is completed in ≈ 12 steps, which is the expected length for this task with fully specified, hand-coded rules. A close examination of the agent's final Q-value table, which stores the agent's rule-firing preferences for any given goal state, indicates that the agent has learned goal-conditioned rules perfectly: there is no need to hand-code goal states in the conditions of a rule to deterministically guide the cognitive process. The Q-learning agent learns by trial-and-error interaction with the environment when to fire which rule, and when to choose to wait and not fire any rule. The agent learns all this from a minimally specified, but fairly carefully designed, reward structure.

The problem of learnability of production-based cognitive models can be systematically formulated and computationally addressed as a reinforcement learning problem, but this is merely a first inroad into what promises to be a very rich nexus of learnability questions, for example:

(*i*) Is it cognitively realistic to require such a high number of episodes (trial-and-error interactions) for learning to happen? What specifically in the human cognitive architecture enables us to learn from much fewer interactions?

(*ii*) There are other value-based tabular learning algorithms (Sarsa, Expected Sarsa), as well as non-tabular approaches to reinforcement learning (both value and policy based), e.g., linear or non-linear (e.g., neural network) function-approximation approaches. How do they perform on lexical decision tasks?

(*iii*) How do all these different approaches perform on a variety of production-based cognitive models, whether linguistic, e.g., syntactic or semantic parsing, or non-linguistic?

Two-stimuli task. We simulate $15{,}000$ episodes, i.e., $15{,}000$ lexical decision tasks consisting of 2 stimuli only (the word 'elephant' and the non-word 'not_a_word'), from which the Q-learning agent learns. The middle plot above shows that, after about $5{,}000$ episodes, the task is completed in ≈ 18 steps, which is the expected length for this task with fully specified, hand-coded rules. A close examination of the agent's final Q-value table indicates that the agent has learned goal-conditioned rules almost perfectly. Unlike in the 1-stimulus task, there is one state-action pair that is not optimal, and is simply a reflection of the trial-and-error learning process that takes longer is and more error prone than for the simpler, 1-stimulus task.

Four-stimuli task. We simulate $25{,}000$ episodes, i.e., $25{,}000$ lexical decision tasks consisting of 4 stimuli (the word 'elephant', the non-word 'not_a_word', the word 'dog' and the non-word 'not_a_word_again'), from which the Q-learning agent learns. We need more episodes for this task because it is longer, hence more complex, than the one-stimulus or the two-stimuli tasks. The rightmost plot above shows that it takes about $22{,}000$ episodes for the task to be reliably completed in less than 40 steps. The task takes 34 steps with fully specified, hand-coded rules.

A close examination of the agent's final Q-value table indicates that the agent has learned goal-conditioned rules fairly well, but there still is a fairly large amount of noise associated with multiple goal states. This noise is a reflection of the trial-and-error learning process that becomes increasingly difficult for tasks requiring large numbers of steps. In the 4-stimuli task, we see that even after $25{,}000$ episodes, the agent still wastes time every now and then trying incorrect rules or just waiting (selecting no action) for no good reason.

References:

Anderson, John R., and Christian Lebiere. 1998. *The atomic components of thought*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Brasoveanu, Adrian, and Jakub Dotlačil. 2019. Quantitative comparison for generative theories. In *Proceedings of the 2018 Berkeley Linguistic Society 44*.

Engelmann, Felix. 2016. Toward an integrated model of sentence processing in reading. Doctoral Dissertation, University of Potsdam, Potsdam.

Fu, Wai-Tat, and John R. Anderson. 2006. From recurrent choice to skill learning: A reinforcement-learning model. *Journal of Experimental Psychology: General* 135:184–206.

Hale, John. 2011. What a rational parser would do. *Cognitive Science* 35:399–443.

Lewis, Richard, and Shravan Vasishth. 2005. An activation-based model of sentence processing as skilled memory retrieval. *Cognitive Science* 29:1–45.

Mnih, Volodymyr, and al. 2015. Human-level control through deep reinforcement learning. *Nature* 518:529–533.

Sutton, Richard S, and Andrew G Barto. 2018. *Reinforcement learning: An introduction*. MIT press.

Watkins, Christopher J. C. H., and Peter Dayan. 1992. Q-learning. *Machine Learning* 8:279–292. URL https://doi.org/10.1007/BF00992698.

Watkins, Christopher John Cornish Hellaby. 1989. Learning from delayed rewards. Doctoral Dissertation, King's College, Cambridge, UK. URL http://www.cs.rhul.ac.uk/~chrisw/new_thesis.pdf.