Embodied Cognition and Multi-Agent Behavioral Emergence

Paul E. Silvey and Michael D. Norman

The MITRE Corporation[1]
{psilvey, mnorman}@mitre.org

Abstract

Autonomous systems embedded in our physical world need real-world interaction in order to function, but they also depend on it as a means to learn. This is the essence of artificial Embodied Cognition, in which machine intelligence is tightly coupled to sensors and effectors and where learning happens from continually experiencing the dynamic world as time-series data, received and processed from a situated and contextually-relative perspective. From this stream, our engineered agents must perceptually discriminate, deal with noise and uncertainty, recognize the causal influence of their actions (sometimes with significant and variable temporal lag), pursue multiple and changing goals that are often incompatible with each other, and make decisions under time pressure. To further complicate matters, unpredictability caused by the actions of other adaptive agents makes this experiential data stochastic and statistically non-stationary. Reinforcement Learning approaches to these problems often oversimplify many of these aspects, e.g., by assuming stationarity, collapsing multiple goals into a single reward signal, using repetitive discrete training episodes, or removing real-time requirements. Because we are interested in developing dependable and trustworthy autonomy, we have been studying these problems by retaining all these inherent complexities and only simplifying the agent's environmental bandwidth requirements. The Multi-Agent Research Basic Learning Environment (MARBLE) is a computational framework for studying the nuances of cooperative, competitive, and adversarial learning, where emergent behaviors can be better understood through carefully controlled experiments. In particular, we are using MARBLE to evaluate a novel reinforcement learning long-term memory data structure based on probabilistic suffix trees. Here, we describe this research methodology, and report on the results of some early experiments.

Introduction

Autonomous systems embedded in our physical world need to be intellectually competent to perform the tasks they were designed for, but they need not demonstrate human-level abilities with language or reasoning to be effective and valuable to us. They will, however, need to function through real-world interaction and, most likely, will gain the knowledge on which they depend from direct experiential learning. Artificial agents facing adversarial challenges need cognitive capabilities that are tightly integrated with, and dependent on, their embodiment, but today's dominant Machine Learning (ML) methods have limited applicability in these kinds of dynamic situated environments, which are characterized as continuous, on-going, time-pressured, uncertain, and statistically non-stationary [1][2]. In response, we are developing and testing holistic, temporally sensitive machine learning and goal-directed planning methods for embodied cognition and multi-agent artificial intelligence, with the objective of improving our ability to build resilient and trustworthy real-world autonomous systems. To this end, we have built a Java testbed designed to host a variety of progressively more difficult challenge scenarios, in which multiple artificial agents play cooperative, competitive, and adversarial games with and against each other and their environments. While this is primarily intended to offer a means to test and refine our episodic and procedural memory innovations, it naturally presents a dual opportunity to study behavioral emergence and complexity in a bottom-up fashion [3].

The best examples we have of intelligent behavior are biological ones, particularly animate beings with long lifespans of embodied learning in our physical world. Insects, animals, and humans all have multi-modal sensory organs coupled to neural information processing systems, as well as real-world effectors that allow them to both change their perspectives and to alter their local environments. Proponents of Artificial General Intelligence argue that AI has become too compartmentalized and fragmented, where researchers study things like vision, language, reasoning, and planning in isolation. Even Machine Learning is fragmented into significantly diverse sub-schools of technique and thought [4]. Primitive embodied cognition, easily observable in even the lowest life forms, requires that many of these faculties be integrated into what we might call a system of

[1] The authors also wish to specifically acknowledge Jason Kutarnia and Brittany Tracy, for their contributions to this work.

systems. One theme that stands out for such embodied cognitive agents is the importance of time. Data comes to their sensory mechanisms as impressions or signals that vary with time, and the perceived world can change at varying rates, often completely independent of the agent's presence. By directly confronting the temporal aspects of situated learning and decision making, and by integrating these capabilities together, we believe we can influence and improve the body of practice in building trustworthy and autonomous systems.

Our bottom-up methodology is motivated by drawing a behavioral research analogy with the Study of Model Organisms in the Life Sciences, where multiple groundbreaking discoveries have been made. Here, cellular and genomic research is more easily and effectively conducted using simple life forms, such as the fruit fly or nematode worm [5]. Similarly, our artificial agents are simple enough for us to conduct extensive behavioral and learning experiments with them, yet they are embodied in realistically complex worlds they can only partially observe, understand, and control. Coping with limited sensory, memory, and processing time resources is a hallmark of intelligence. Our research approach enables us to study and to test learning under these types of real-world cognitive constraints. We specifically start with low-entropy, low-bandwidth sensory data to allow rapid experimentation using standard compute resources. Our agents use relatively simple data-driven mechanisms to construct problem-specific and accurate Markov Decision Process models and use behavioral learning algorithms to cope with changing temporal sequence data and real-time dynamic pressure.

This paper is organized as follows: first, we review some significant past AI research in embodied cognition and reinforcement learning and discuss emergent behaviors in multi-agent simulations. Then, our own research testbed is described, along with experimental scenarios that we have explored. Next, our holistic agent's cognitive architecture is introduced, and we then share some of the unexpected learned behaviors these agents presented for us to analyze and seek to explain. Finally, we conclude with our plans to continue this work, and some of the various directions we hope to take.

Related Work

The disciplined Computer Science quest to develop true Artificial Intelligence (AI) is over sixty years old, but during most of that time was dominated by disembodied mind research, where a focus on language and knowledge in many ways led researchers to detach aspects of thinking and reasoning from sensing, acting, and real-world interaction. However, there have always been those who believed intelligence depends in non-trivial ways on being embodied, and on learning from the uncertain experiences of living.

For example, by the late 1980s, seminal work at MIT had begun to build simple insect-like robots [6]. These served to demonstrate the power of primitive reactive thinking mechanisms, which could work without deliberative or goal-directed planning or learning. Soon there was great interest in the idea of autonomous agents and distributed AI, often using object-oriented programming concepts to encapsulate and situate artificial agents in simulated or virtual problem environments [7]. A good summary of the development of these views can be found in Anderson's field guide to embodied cognition [8]. At the same time, clearer definitions and better algorithms for Reinforcement Learning as a distinct paradigm were being developed [9], while others were beginning to pursue universal and general cognitive architectures [10].

Along with these developments came a slow shift in AI, from logic and predicate calculus as the dominant paradigm for Knowledge Representation to more probabilistic modeling and data-driven (so-called *weak*) methods. Hand-crafting or Knowledge Engineering of assertions and rules gave way to the use of more statistically-based techniques, which have proven to be extremely powerful in spite of their label. Owing much to this trend, the recent successes of deep artificial neural networks have reinvigorated the fields of AI and machine learning, but they may have unfortunately steered some focus away from the progress that was being made by the early embodied learning agent researchers.

Another area of inspiration for us, where simplicity has shown great power, is complexity science. Here, chaos can be generated by something as simple as the recurrence equations of the logistic map, and patterns that are unpredictably complex can be generated by simple one-dimensional cellular automata [11]. Likewise, the study of the emergence of cooperation, which we discuss more later in this paper, came from simple game-theoretic and simulation analysis [12]. It has now become standard practice to try to understand complex systems using Agent-Based Modeling.

The MARBLE Framework and Testbed

Our Multi-Agent Research Basic Learning Environment (MARBLE) is a discrete modeling framework and simulation testbed for doing realistic embodied cognition research, implemented in the Java programming language. It is constructed around the Model-View-Controller design pattern [13], based on a 2D world model consisting of a hexagonal grid that, by default, wraps around on all six sides. The

wrap-around logic is such that moving forward in any one direction repeatedly will visit each cell in the world exactly once before returning to the starting cell [14]. Agents and other simulated objects may occupy any given cell, and agents have directional orientation with a limited field of view and a small repertoire of possible actions on any given simulation time step. We use a reinforcement learning paradigm, where each agent receives sensory data and reward signals for the results of their prior-step action choice, and all agents attempt their next moves simultaneously. This forces the environment controller to mediate the results of agent actions, such that the agents must discover, through experience, what their actions might or might not accomplish for them in particular discernable situations. Agents receive their sensory data as a *DataFrame* object, which is a bit-string of problem-specific length whose structure remains undisclosed to the agents, encoding experimenter-determined discernable characteristics or perceptual features within the agent's field of view. For example, a simple world might use only 3 bits to encode whether each of the three immediately facing cells contains an object or not. Slightly more information-rich examples might use 3 bits per cell for a total *DataFrame* size of 9 bits, where each cell can distinguish eight different states, including being empty, containing a food object, or containing another agent in one of its six directional orientations. See Figure 1 for some currently established MARBLE elements.

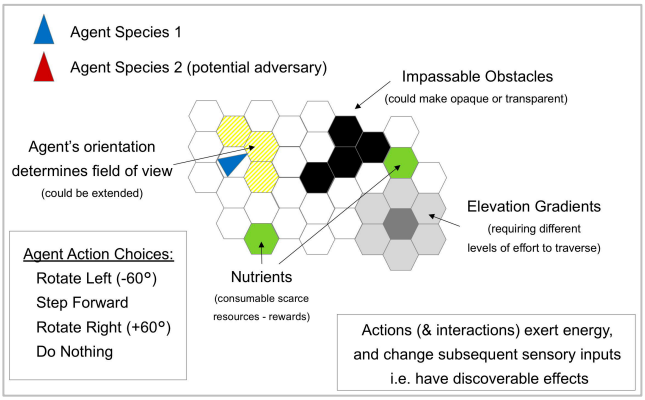


Figure 1. Example MARBLE Elements.

We have implemented two distinct problem scenarios, both related to a simple food gathering learning challenge. The first contains only a single agent, who lives in a world that contains only one other object, which is a green cell representing food. The agent can choose between one of the following four distinct actions per time step: stepping forward one cell, turning 60 degrees right or left, or doing nothing. The reward structure for this game is -1 point for any action that does not result in the agent stepping into the cell containing food, and +1000 points when it consumes the food in this way. When the food is consumed, a new food object immediately appears four cells directly ahead of the agent, based on its orientation at the time. Optimal behavior for this problem is easy to specify, unless the size of the hex grid becomes small enough that the shortest sequence of moves between feedings is not simply to keep walking forward. This problem is completely deterministic, and only poses a simple delayed-gratification challenge. As this problem is extremely simple, the sensory data presented to the agent at each time step is the 3-bit example given above, where an unoccupied cell is perceived as a zero bit, and the existence of the food object in any of the three cells facing the agent (immediately ahead of and to its left and right periphery) is perceived as a one bit (see agent's field of view in Figure 1).

The second variant of the food gathering problem consists of a red agent and a blue agent who co-exist in the same otherwise empty world, also with a single food object that they must compete to find and benefit from consuming. The reward structure is mostly the same, with -1 point for staying put, turning, or stepping into an empty cell, and +1000 points for consuming the food. The food regenerates as before, four steps ahead of the agent who has just consumed it, or one additional step forward if that cell is occupied by the other agent. Our physics rules for this problem prevent the two agents from occupying the same cell at the same time, so conflict resolution logic must be added to the controller. When both agents attempt to step into the same cell at the same time, a random coin flip determines who will succeed, with the losing agent not moving from its cell of origin and receiving a -10 point reward. The winner will receive either the usual -1 point for entering an empty cell, or +1000 points if the cell contains the food object. This problem was intended to study competition, and as such the agents are penalized -50 points for the *aggressive* action of trying to move into a cell that the other agent already occupies (and is not simultaneously vacating). If the two agents are facing one another and both try to move forward, they each receive the -50-point penalty. The addition of another competitive agent changes the stochastics of the world considerably, and, as we will see, creates the opportunity for many complex behaviors to emerge through learning. The sensory data for this problem is also more feature-rich, with each agent being able to discriminate empty cells, cells occupied by the food object, and cells occupied by the other agent (as well as the other agent's orientation relative to itself). This uses a 9-bit *DataFrame* as described earlier.

Embodied Agent Learning

Our current MARBLE agents are built using a long-term episodic and procedural memory data structure which records experienced temporal sequence traces, maintaining event

distribution statistics for observed successor states and expected discounted rewards for actions that have been tried. Our machine learning research is aimed at understanding how well these memory structures perform in a variety of challenge problem environments, and how well certain parameter settings might affect rates of learning, abilities to generalize effectively, and abilities to cope with non-stationarities caused by the presence of, and interactions with, other adaptive agents.

The basis of our learning framework is a variable depth probabilistic suffix tree [15], which produces an environmentally data-driven variable-order Markov Model [16], using focus bits from each data frame in the temporal sequence sensory data. For large data frames, multiple trees using different foci bits can be built in parallel, and their independently recommended actions can be pooled using an ensemble strategy. In low-entropy environments or when the number of focus bits is small, these trees are sparse and can grow deep to discover causal patterns with long temporal lags. Each node in the tree represents a state with probabilistic knowledge of its successors, updated through experience in a Bayesian manner.[2] As in hidden Markov models, these tree nodes have both output variable probabilities for expected rewards, as well as successor (hidden) state transitions within the tree. These internal transitions are computed (vs. explicitly being stored) by walking from a current state leaf node to the root, appending the latest perceptual state as a new suffix to a short term memory buffer, and using that new suffix to walk down the tree to the next leaf node state. The current-to-next leaf node pairs represent logical links within the tree that, when considered separately, form a partial DeBruijn Graph, since the short-term memory buffer acts like a shift register as time progresses.

There are two types of learning currently implemented in our MARBLE agents, and they both occur by updating probability estimates while ascending from the current leaf node to the tree root as just described. The first type can be described from the perspective of model-free design [17], where the expected utility of the most recent action is updated using the Q-Learning technique [18]. It is model-free in the sense that it blindly learns actions that will maximize rewards without any attempt to develop sensory expectations. Learning is completely based on event sequences the agent has actually experienced. The other type of learning is model-based, as it updates the probability of seeing the just-experienced perceptual symbol extracted from the *DataFrame* for each successively shorter suffix state (defined by the tree nodes visited during the climb to the root).

This second form of learning prepares the agent to use its memory in imaginative ways and to envision futures that are probabilistically plausible, whether they in fact have ever actually been experienced. Probabilistic suffix trees have been used in compression algorithms exactly because they construct statistical prediction models (which can inform arithmetic or Huffman encoders based on a symbol's probability of occurrence). This type of learning forms the basis of deliberative planning mechanisms, including ones like Monte Carlo planning [19], which can be implemented to work in time-constrained situations as an Anytime Algorithm [20]. Neuroscientists and AI researchers have hypothesized that temporal sequence prediction is a core foundational capability of the brain's neocortex, common across all sensory data modalities [21]. Similarly, the ability to recognize surprise in the form of expectation failures has been seen as a significant trigger mechanism for attentionally-directed learning of event salience [22]. As a result, this kind of model-based learning is seen as extremely important for inclusion in embodied cognitive agent research.

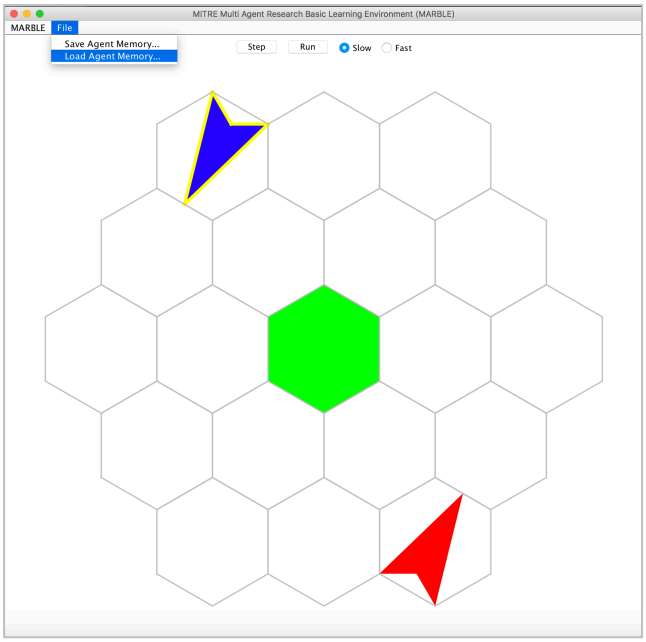


Figure 2. Small-world MARBLE Competition

Figure 2 shows the global view of the MARBLE competitive problem scenario, with the blue agent highlighted and the memory load/store menu item selected. The probabilistic suffix tree memories can thus be saved to disk and reloaded for additional training and/or testing at a later time. The controller buttons at the top include a single step button and a

[2] Posterior probabilities are computed from maximum entropy priors initialized by setting the alpha parameter in a multimodal Dirichlet distribution.

run/stop button with radio controls for two speeds. Clicking on an agent when the simulation is stopped highlights it with a yellow outline and allows the user to 'drive' that agent manually using the arrow cursor keys (which also invoke the step operation). In addition, a right-click on an agent will bring up a parameter dialog box where learning modes and exploration rates and conditions can be set (see Figure 3).

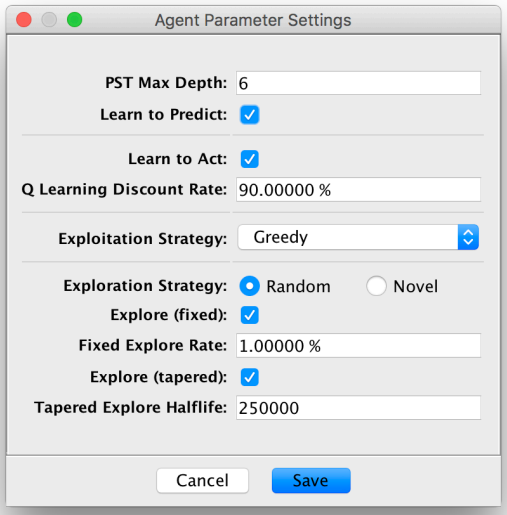


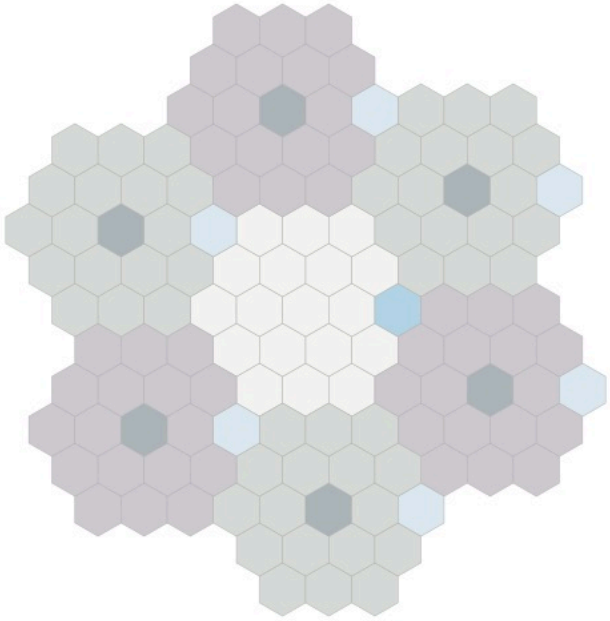
Figure 3. MARBLE Agent Parameter Settings

As noted, cognitive constraints include both limitations to memory space and to thinking time. They are naturally related in MARBLE agents because algorithms that learn through traversal of the memory tree, or creatively plan by using it to generate hypothetical futures, will take more time as the size of the tree grows. Although there are many relatively low-entropy problem domains that an embodied cognitive agent might face, distinguishing all observable state combinations over long time horizons produces combinatoric growth that becomes unbearable without efforts to constrain it. To address this, our research plan includes several methods we want to test to implement controlled remembering, selective forgetting, and generalized abstraction via state combination, but the details of these are beyond the scope of the current paper. As our initial experiments are with limited bandwidth and small-world environments, this potential scaling problem has not yet arisen. Amazingly, rich behavioral patterns are seen to emerge from our agents, despite our attempts at extreme experimental simplification.

Behavioral Emergence

In this section, we describe four examples of emergent behavior that we witnessed from our embodied cognitive agent learning experiments. We use colloquial descriptions of human-like behavior to illustrate how we interpreted what the agents were doing.

One of the first things we observed in our single agent food gathering scenario was that, at least for certain world sizes, the agent developed a preference for turning right vs. left. After some investigation, it became apparent that this was the result of the particular grid alignment used to determine adjacent cells for the wrap-around logic. Figure 4 shows a center hex grid consisting of 19 cells replicated six times along the edges, to highlight which cells are adjacent to each other across the board during wrap-around. Because the shapes interlock like gears, the replica below the center, for example, could equally well have been shifted to the left vs. the right as it is in the figure. The particular choice for this determined that food cells discovered by repetitive walking forward would more often appear in the agent's right field-of-view than in its left, because the continuous traversed path wraps and aligns as adjacent rows which sweep left to right. Thus, when the agent can finally 'see' the food, a right



turn from that state will position it to receive the large reward. As a result, the agent essentially formed a habit of turning right when there was no better alternative available.

Figure 4. Hexagonal grid wrap-around logic.

Our second example comes from being able to empathize with the agents, and thinking of the food not just as a perceptual bit pattern but as an object in the world. It started by asking how we should determine if an agent has learned. The most obvious way is to look at metrics like total accumulated reward or average reward per step (which should improve over time). But observing agent behaviors showed us that this is only part of the story. The reinforcement learning mechanism produces a state-based decision policy, such that some states have clear preferences for actions that are likely to lead to rewards. If action choices are made in a greedy way while learning, the agents quickly learn what to do, but unless they explore sufficiently, they will not learn the best action to take in many cases. Since the world never forced our agents into some of those states and they didn't play or explore enough on their own, their learning turned out to be fragile. We saw this because we had a natural sense of what we would do if we were in the agent's shoes, experiencing the world and learning from it, and when an agent didn't do something we expected it appeared much less intelligent.

Specifically, we observed that agents who explored a lot appeared to learn the concept Psychologists refer to as *Object Permanence* (with respect to the food, which persists in its cell until consumed in the single agent scenario), whereas those who always tried to exploit their acquired knowledge clearly did not. We could see this by manually controlling the agents to put them into test states, from which we let them decide how to act. For example, if the food was in their right periphery but we turned them left (away from the food), smarter agents knew that two successive right turns would return them to facing it, demonstrating their grasp of Object Permanence. Similarly, we could test a number of drive-by or near-miss food encounters, where memory of the recent past were exploited by the most intelligent agents, while others seemed to possess ineffectual or missing short-term memories. Here, two factors were relevant for the smarter agents. First, as noted, they explored more, and second, they had suffix tree memories that were deep enough to distinguish and remember what they had observed multiple time-steps in the immediate past. Cognitive resources such as memory capacity are necessary but not sufficient for intelligence, and they attain their ultimate value for agents by the diversity of experiences that feed them.

Our third example is one of stubbornness. Our competitive learning scenario with two agents was designed to present a non-stationary learning problem that was as simple as possible. The initial learning algorithms we tested, however, were not designed for this, since they assumed that estimated probabilities would converge by *the law of large numbers*. Our training cases usually pitted novice agents against each other, often with the exact same mental resources and parameter settings. The rate of exploration (vs. exploitation) was slowly tapered to vanishing, and the learned greedy mode behaviors were examined at the end of the run. As is common in many machine learning situations, we witnessed numerous cases where behaviors appeared to have settled into a local (vs. global) optimum. When exploration stopped, it became apparent that some of these behaviors led to starvation for one or both agents, even though they both survived reasonably well whenever there were still some occasional random actions taken. The most unexpected of these happened with two agents that had shallow suffix trees of only depth 2 and when they both learned that going forward was the best action in almost all cases. Because their expected rewards had converged through many trials, if they ran into each other they got locked into a head-butting stalemate, which only could be broken by many steps of receiving the -50-point penalty. We believe that faster forgetting, in the form of non-stationary reward estimators, will avoid this behavior or at least permit it to be seen as futile more quickly.

Finally, in several competitive agent experiments conducted using the small 19-cell world, the agents essentially learned to cooperate, and they did it by dancing a kind of waltz (there were three time-steps between feedings). We had hypothesized that some synergistic or win-win behavior might emerge, as opposed to the pathologies just described, but this behavior appeared remarkably clever. The agents alternated in consuming the food, carefully positioning themselves so as to do this most effectively and using the wrap-around nature of the world to continue this way indefinitely. Furthermore, this behavior had become robust in the sense that random perturbations to one or the other agent upset the dance for a few steps, but they were able to settle back into the rhythm fairly soon.

Future Directions

Our MARBLE testbed is designed to support problems that involve many agents, with many more discernable conditions, objects, and interaction rules. We have only begun to scratch the surface of experimental conditions we can configure and test against. However, we don't have to make the environment much more complicated to study the essential problems we have outlined above. A couple of important areas only mentioned in passing that we would like to address soon include the following.

First, we need to test the variety of methods we have planned for dealing with non-stationarities, including fading memories using moving window averages or exponentially decaying weighted ones. In addition, there are numerous techniques that data analysis practitioners have developed for measuring the degree of stationarity in time-series data,

which could likely be employed as adaptive hyperparameter setting mechanisms.

Multi-goal (or multi-task) learning is a real-world problem and a topic that is currently of great interest in machine learning research. It is more complicated than simply training a single classifier to recognize multiple types of objects, or a single neural network to play a variety of games without re-training, though these problems pose significant challenges in their own right. Recognizing that motivations are neither constant nor linear (e.g., consumption affects appetite), and that goals can interact in both synergistic as well as antagonistic ways, are just of few of the concerns here. Sometimes goals can be stacked in a kind of hierarchy, with lower level or immediate need (tactical) goals taking precedence over longer-range (strategic) goals. Making curiosity a motivation to drive exploration, after more primitive needs such as safety and survival have been met, is one example of this type of thinking. On the other hand, it is interesting to speculate how many people would survive their childhood if they didn't have adults overseeing their play, suggesting this problem might best be solved in a social or multi-agent manner. See [23] for a cognitive architecture design that seeks to address motivation as a first-class problem. Receiving multi-goal-indexed reward signals and maintaining separate expected utilities for them can possibly provide a foundation for flexible multi-task behaviors, and a meta-observation process might be able to learn when goals align or diverge in their recommended actions. These are some of the ideas we would like to pursue.

There have been several efforts to create standardized challenge problems for AI researchers to compare their algorithms and implementations, and we are considering an Open Source release of MARBLE for this purpose. At the same time, adapting our agents to other reinforcement learning and game simulation environments is of interest to us. Two of these are the OpenAI Gym environment [24] and the micro-Real-Time-Strategy (μRTS) game platform and competition [25]. Most of the Gym problems are cast as single agent vs. the world learning problems, and true multi-player games are only just emerging in that environment. On the other hand, the RTS-type games are inherently adversarial problems, but many of the competitors use heuristic strategies that are not necessarily improved through machine learning. We would like to see these testbeds used to examine agents learning against other agents (who are themselves simultaneously learning) in order to create the kinds of non-stationary challenges we seek to study.

Embodiment also presents a natural opportunity for comparing our artificial agent's cognitive abilities against those of humans and for exploring ways in which the agents might be taught faster through demonstration via remote human control using Virtual Reality (VR). In a complementary way, a skilled artificial agent could advise a human doing a similar task using Augmented Reality (AR). We plan to explore these ideas by building interfaces between our testbed and the popular Unity game engine [26]. This presents unique and powerful opportunities for bi-directional teaching and learning in human-computer teaming scenarios. If people can "get in the robot's head" to experience its sensory world and control its motor responses, they might guide its learning more quickly and efficaciously on how to deal with important problem situations. Similarly, an artificial cognitive agent who has learned effective ways to solve a problem or get out of trouble could give hints and decision suggestions to humans doing the same task, whether using heads-up displays or other types of Augmented Reality interfaces.

Conclusions

While there has been a recent resurgence of interest in Machine Learning in general, and deep learning neural networks in particular, there remains a kind of compartmentalization where only narrow aspects of intelligence receive careful study.

Supervised machine learning, as widely practiced today, focuses on building classifier systems from expert-curated labeled training data, based on assumptions of statistical stationarity and using off-line processing. On the other hand, research in Reinforcement Learning studies on-line learning that is at least episodic if not continuous, confronting head-on the problems of sensing and goal-directed action in the same manner as embodied cognitive agents in the real world. Modern robots (including industrial manufacturing robots, robo-soccer players, self-driving cars, etc.) can be taught to perform specific functions in reasonably controlled environments, but, once taught, they usually operate in exploitative or performance-only modes.

Learning to act (to exploit) requires (exploratory) acting to learn, and this is only one of many examples where real agents must use attentional focus or goal prioritization to simultaneously balance their potentially oppositional objectives. Model-free RL, such as the technique known as Q-Learning, can learn to optimize behavior around a single reward signal, but because it builds no explicit model of its environment, is unable to realize when something very unexpected happens. Very little autonomous systems research has tried to integrate a full complement of real-world requirements for embodied cognition. These requirements include learning perceptual discriminators, developing sequential expectations from experience, using delayed rewards to hone reactive skills, dealing with variable causal time lags and real-time decision requirements, using multi-

goal attention balancing and planning (when possible), and using surprise as a motivator to adapt and re-learn.

We believe all these aspects and challenges of intelligence through embodied cognition can be studied together; we should instead simplify the problem in other ways, such as using simple discrete simulations with few observable states and few selectable actions. The research we have described here takes this approach.

We conclude with one final anecdote that came from enacting this work. It is easy for us to anthropomorphize our simple artificial intelligent agents, because we have built them in our own image, so to speak. They too are embodied cognitive beings who experience and interact with a reasonably predictable world through a stream of temporal sequence data. The environments we create for them have been sometimes compared to *god games* because there are privileged views of the world and its rules that we understand (and program) which are not known by the agents. However, we sometimes see unintended consequences of our actions when we run the agents, as happened in the following case. Our initial implementation of the competitive controller included the conflict resolution rules described earlier, but accidentally failed to properly consider what to do in one rather unlikely case. This occurred when the food was being consumed by one agent and the location for its regeneration was in the path of the other agent. In the process of moving the food to a new cell, the controller put it back in the world before the second agent's move had been completed. When the second agent had already decided to step forward *and* the food was being relocated to that destination cell, the controller incorrectly credited both agents with the 1000-point reward on the same time-step, effectively moving the food twice in one step. This bug in our code violated the intended physics rules and was only discovered when we observed the agents collecting more reward than was 'possible'. The agents had found a 'glitch in the matrix' and exploited it, thus humbling their creators.

References

[1] Wilson, M. (2002) Six views of embodied cognition, Psychonomic Bulletin & Review, vol. 9, no. 4, pp. 625–636.

[2] Machado, M.C., Bellemare, M.G., Talvitie, E., Veness, J., Hausknecht, M., and Bowling, M. (2018) Revisiting the Arcade Learning Environment: Evaluation protocols and open problems for general agents, Journal of Artificial Intelligence Research, vol. 61, pp. 523–562.

[3] Silvey, P.E. (2017) Leveling Up: Strategies to Achieve Integrated Cognitive Architectures. AAAI 2017 Fall Symposium Series - A Standard Model of Mind: AAAI Technical Report FS-17-05, pp. 460-465.

[4] Domingos, P. (2015) The Master Algorithm: How the quest for the ultimate learning machine will remake our world. Basic Books.

[5] Mukherjee, S. (2017) The Gene: An intimate history. Simon and Schuster.

[6] Brooks, R. (1986) A robust layered control system for a mobile robot. IEEE journal on robotics and automation 2.1, pp. 14-23.

[7] Jennings, N.R., Sycara, K., Wooldridge, M. (1998) A Roadmap of Agent Research and Development. Autonomous Agents and Multi-Agent Systems 1, pp. 7–38.

[8] Sutton, R.S. (1988) Learning to predict by the methods of temporal differences. Machine learning 3, pp. 9–44.

[9] Anderson, M.L. (2003) Embodied Cognition: A field guide. Artificial Intelligence, vol. 149, no. 1, pp. 91–130.

[10] Laird, J., Newell, A., and Rosenbloom, P. (1987). SOAR: an architecture for general intelligence. Artificial Intelligence, 33, pp. 1-64.

[11] Mitchell, M. (2009). Complexity: A guided tour. Oxford University Press.

[12] Axelrod, R., & Hamilton, W.D. (1981). The evolution of cooperation. science, 211(4489), pp. 1390-1396.

[13] Coad, P. (1992). Object-oriented patterns. Communications of the ACM, 35(9), pp. 152-159.

[14] Patel, A. (2013) Red Blob Games, Hexagonal Grid Reference. https://www.redblobgames.com/grids/hexagons/.

[15] Volf, P.A., and Willems, F.M. (1995). A study of the context tree maximizing method. In Proc. 16th Symposium on Information Theory in the Benelux, Nieuwerkerk Ijsel, Netherlands (pp. 3-9).

[16] Begleiter, R., El-Yaniv, R., and Yona, G. (2004) On Prediction Using Variable Order Markov Models, Journal of Artificial Intelligence Research 22, pp 385-421.

[17] Norman, M.D., Koehler, M.T.K., and Pitsko, R. (2018) Applied Complexity Science: Enabling Emergence through Heuristics and Simulations. In S. Mittal & S. Diallo and A. Tolk, Emergent Behavior in Complex Systems Engineering: A Modeling and Simulation Approach (pp. 201-226). Wiley.

[18] Watkins, C.J.C.H. (1989) Learning from delayed rewards. Ph.D. diss., King's College, Cambridge.

[19] Chung, M., Buro, M., and Schaeffer, J. (2005) Monte Carlo Planning in RTS Games. CIG.

[20] Dean, T.L., and Boddy, M.S. (1988) An Analysis of Time-Dependent Planning. AAAI. Vol. 88.

[21] Hawkins, J., and Blakeslee, S. (2007) On intelligence. Macmillan.

[22] Schank, R.C. (1999). Dynamic memory revisited. Cambridge University Press.

[23] Bach J. (2009) Principles of Synthetic Intelligence PSI: An Architecture of Motivated Cognition. Oxford University Press, USA.

[24] Brockman, G. et al., (2016) OpenAI Gym, arXiv:1606.01540.

[25] Ontañón, S., Barriga, N.A., Silva, C.R., Moraes, R.O., and Lelis, L.H. (2018). The First MicroRTS Artificial Intelligence Competition. AI Magazine, 39(1).

[26] Unity 3D Game Engine, https://unity3d.com/public-relations.