A Memory-based Multiagent Framework for Adaptive Decision Making

Extended Abstract

Shauharda Khadka
Oregon State University
khadkas@oregonstate.edu

Connor Yates
Oregon State University
yatesco@oregonstate.edu

Kagan Tumer
Oregon State University
kagan.tumer@oregonstate.edu

ABSTRACT

Rapid adaptation to dynamically change one's policy based on a singular observation is a complex problem. This is especially difficult in multiagent systems where the global behavior emerges from inter-agent interactions. In this paper, we introduce a memory-based learning framework called Distributed Modular Memory Unit (DMMU) which enables rapid and adaptive decision making. In DMMU, a shared external memory is selectively accessed by agents acting independently and in parallel. Each agent processes its own stream of sequential information independently while interacting with the shared external memory to identify, retain, and propagate salient information. This enables DMMU to rapidly assimilate task features from a group of distributed agents, consolidate it into a reconfigurable external memory, and use it for one-shot multiagent learning. We compare the performance of the DMMU framework on a simulated cybersecurity task with traditional feedforward ensembles, LSTM based agents, and a centralized framework. Results demonstrate that DMMU significantly outperforms the best LSTM based method by a factor of two and exhibits adaptive decision making to effectively solve this complex task.

KEYWORDS

Memory; Distributed Learning; RNNs, Multi-agent Coordination

ACM Reference Format:
Shauharda Khadka, Connor Yates, and Kagan Tumer. 2018. A Memory-based Multiagent Framework for Adaptive Decision Making. In *Proc. of the 17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2018), Stockholm, Sweden, July 10–15, 2018,* IFAAMAS, 3 pages.

1 INTRODUCTION

Multiagent systems (MAS) can accomplish complex tasks in highly dynamic and stochastic environments improving on both speed and effectiveness over single agent approaches. However, multiagent coordination is a complex control problem especially when the task **requires rapid adaptive behaviors** from the group of coordinating agents. The difficulty of the task is further exacerbated when the features relevant to decision making are distributed over the agent's sequence of observations. In such tasks, it is critical for each agent to dynamically capture relevant features from its own set of observations, while the multiagent team consolidates these features across each other in making rapid adaptive decisions.

Proc. of the 17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2018), M. Dastani, G. Sukthankar, E. André, S. Koenig (eds.), July 10–15, 2018, Stockholm, Sweden.

A characteristic component of rapid adaptation is the ability to dynamically change one's behavior (policy) after a singular observation. One approach to facilitate adaptive behaviors is to incorporate memory which can be used to remember salient observations and recall them during future decision making [8, 9, 13]. Adaptive behaviors for single agent systems have been explored in the past largely with the tools of memory [1, 2, 11, 13]. However, the increased complexity from multiple agents acting concurrently is widely unexplored for tasks that require rapid adaptive behaviors.

Long Short Term Memory (LSTM) [5, 6] is one popular way of incorporating memory in a learning system, and is the state of the art in many sequence processing tasks [3, 4, 12]. A characteristic feature of LSTM is the intertwining of the memory operations and the feedforward computation of the network where the memory stored in the cell states are updated with each feedforward operation. However, in a multiagent setting where multiple agents share memory, this architecture leads to homogenization among the joint action set. Additionally, concurrent access to the same memory cell leads to high degrees of interference and memory corruption.

2 DISTRIBUTED MODULAR MEMORY UNIT

In order to enable distributed adaptive decision making, we introduce Distributed Modular Memory Unit (DMMU) leveraging the modular architecture of GRU-MB [7]. Figure 1 depicts the organization of the DMMU framework. Unlike GRU-MB, DMMU is designed to process streams of sequential observations from multiple agents concurrently. The principal component of the DMMU framework is its modular and flexible integration between the external memory and the agents that interact with it. DMMU is an open system where an agent can join or leave dynamically during execution. This allows for a high degree of reconfigurability such that a variable number of agents (possibly heterogeneous), acting on their own unique set of observations, can interact with the shared external memory in parallel.

Each agent operates with its own unique policy and is defined by a standard feedforward neural network with three key additions:

(1) **Input Gate:** An input gate filters the flow of information that comes from the environment. This serves to shield the agent from the noisy portions of each incoming observation and allows it to focus its attention on relevant features within its observation set.
(2) **Curated Memory Feed:** The agent's input is augmented with content that is selectively read from an external memory. The agent has an independently learnable read gate which filters the content read from external memory. This serves to shape the contents of memory as per the needs of the agent, protecting it from being overwhelmed.

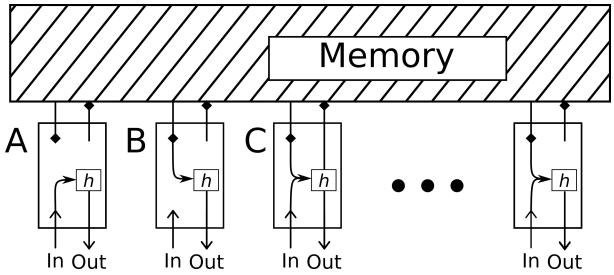

Figure 1: High level schematic of the DMMU framework. Each agent is comprised of a neural network with connections to the world (input/output) and memory (read/write) connections. Agents A, B, and C highlight the modularity of the framework. At this time, agent A ignores memory, and acts reactively based on its input. Agent B ignores its input and acts exclusively from memory. Agent C leverages all available information, combining the contents within memory and its immediate input in making decisions. Agent C also updates memory based on its decision. An agent can choose to perform any subset of these actions at any time.

(3) **Selective Memory Update:** The agent uses a learnable write gate that allows it to selectively update the contents of the external memory. This gate enables the agent to report salient features from its observations to the external memory at any given time step. The write gate that filters this channel of information flow serves to shield the external memory from being overwhelmed by updates from the agent.

We test DMMU in a simulated cybersecurity task which can be thought of as a multiagent extension of the season task [9] requiring coordinated adaptive behaviors across multiple agents. Here, a web server employing a distributed network of proxy servers has to remain operational while withstanding a DDoS attack. The web server receives multiple requests originating from multiple devices. A portion of these devices are conscripted by a nefarious botnet while the rest are genuine. To successfully solve this task, the proxy servers have to coordinate in exploring the devices' categorization by sampling their requests in parallel, and selectively serve requests originating from the genuine ones. The core difficulty here is that the categorization of devices (nefarious/genuine) changes across task instances. This prevents our agents from simply remembering action-value functions and forces it to dynamically sample and determine the nefarious/genuine categorization of each device for each new instance of the task.

3 RESULTS

We use neuroevolution to train our agents and compare DMMU with four baselines spread across the centralized/decentralized and memoried/reactive axes. Feedforward Neural Ensemble **(FFNE)** and LSTM Neural Ensemble **(LSTMNE)** represents each proxy server as a feedforward neural network and a LSTM, respectively. Centralized Neural Framework **(CNF)** represents the entire server fleet as a single feedforward neural network with centralized access to all the information. LSTM with Shared Memory **(LSTMSM)** represents the server fleet as a group of LSTM network with access to an external shared memory similar to the DMMU setup.

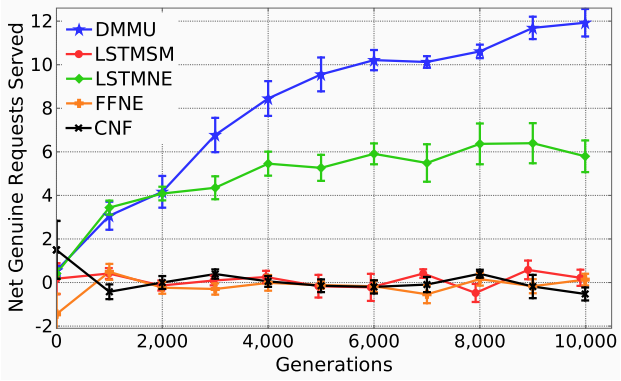

Figure 2: Perfomrance of DMMU alongside other baselines in the cybersecurity task. The server fleet consists of 10 proxy servers handling a request volume of 100 from 20 distinct devices. A random number of devices between {7,13} out of 20 chosen at the start of each task instance are nefarious while the rest are genuine.

Figure 2 shows the comparative performance of DMMU with other baselines in controlling the server fleets where 10 proxy servers handle a request volume of 100 from 20 distinct devices. DMMU significantly outperforms all other baselines achieving a net of 11.93 ± 0.63 genuine requests served. Both approaches without memory (CNF and FFNE) fail to learn in either of the tasks, and converge to an equilibrium state with the net genuine request served centered at 0. This is unsurprising as these methods lack memory and cannot associate actions with rewards over time. Even with centralized access to information and actions (CNF), the lack of memory is a principal limitation.

Surprisingly, LSTMSM fails to learn in both variants of the task despite sharing an external memory similar to DMMU. This is because, unlike DMMU where agents can selectively use memory's contents, LSTMSM's agents are constricted to condition their actions as a strict function of the shared external memory. This greatly limits their flexibility and leads them to fail in the task. This highlights the importance of DMMU's modularity which allows agents to read from, add to, or ignore the external memory at will. This enables DMMU to employ multiple agents with diverse policies working together in sampling devices concurrently, and encoding associations onto the shared external memory.

4 CONCLUSION

Adaptive decision making in a multiagent system where agents can dynamically change their behavior based on a singular observation by one of the agents is a complex problem. However, as most real world systems increasingly move towards decentralization and become more distributed [10, 14], it is an important challenge to tackle. In this work, we introduced the DMMU framework that leverages an external shared memory as a form of 'knowledge base' that can be collectively read and updated by a team of independent agents. Collectively, this facilitates rapid assimilation of dynamic features, and enables adaptive decision making based on a singular observation.

REFERENCES

[1] Bram Bakker. 2002. Reinforcement Learning Memory. *Neural Information Processing Systems 14* (2002), 1475–1782.

[2] Justin Bayer, Daan Wierstra, Julian Togelius, and Jürgen Schmidhuber. 2009. Evolving memory cell structures for sequence learning. *Artificial Neural Networks–ICANN 2009* (2009), 755–764.

[3] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. MIT Press.

[4] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. 2013. Speech recognition with deep recurrent neural networks. In *Acoustics, speech and signal processing (icassp), 2013 ieee international conference on*. IEEE, 6645–6649.

[5] Alex Graves and Jürgen Schmidhuber. 2005. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Networks* 18, 5 (2005), 602–610.

[6] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation* 9, 8 (1997), 1735–1780.

[7] Shauharda Khadka, Jen Jen Chung, and Kagan Tumer. 2017. Evolving Memory-Augmented Neural Architecture for Deep Memory Problems. In *In Proceedings of the Genetic and Evolutionary Computation Conference 2017*. ACM.

[8] Shauharda Khadka, Jen Jen Chung, and Kagan Tumer. 2017. Memory-augmented multi-robot teams that learn to adapt. In *Multi-Robot and Multi-Agent Systems (MRS), 2017 International Symposium on*. IEEE, 128–134.

[9] Benno Lüders, Mikkel Schläger, and Sebastian Risi. 2016. Continual learning through evolvable neural turing machines. In *NIPS 2016 Workshop on Continual Learning and Deep Networks (CLDL 2016)*.

[10] Arvind Narayanan, Joseph Bonneau, Edward Felten, Andrew Miller, and Steven Goldfeder. 2016. *Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction*. Princeton University Press.

[11] Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy Lillicrap. 2016. Meta-learning with memory-augmented neural networks. In *International conference on machine learning*. 1842–1850.

[12] Jurgen Schmidhuber. 2015. Deep learning in neural networks: An overview. *Neural networks* 61 (2015), 85–117.

[13] Kenneth O Stanley, Bobby D Bryant, and Risto Miikkulainen. 2003. Evolving adaptive neural networks with and without adaptive synapses. In *Evolutionary Computation, 2003. CEC'03. The 2003 Congress on*, Vol. 4. IEEE, 2557–2564.

[14] Kagan Tumer and Adrian Agogino. 2007. Distributed agent-based air traffic flow management. In *Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems*. ACM, 255.