Memory based Multiagent One Shot Learning*

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

One shot learning is particularly difficult in multiagent systems where the relevant information is distributed across agents, and inter-agent interactions shape global emergent behavior. This paper introduces a distributed learning framework called Distributed Modular Memory Unit (DMMU) that creates a shared external memory to enable one shot adaptive learning in multiagent systems. In DMMU, a shared external memory is selectively accessed by agents acting asynchronously 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 other methods and exhibits distributed one shot learning.

KEYWORDS

RNNs; Multiagent coordination; One shot learning; Emergent Learning

ACM Reference Format:
Shauharda Khadka, Connor Yates, and Kagan Tumer. 2019. Memory based Multiagent One Shot Learning. In *Proc. of the 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2019), Montreal, Canada, May 13–17, 2019*, IFAAMAS, 3 pages.

1 INTRODUCTION

An open challenge in Artificial Intelligence is to determine how an agent can learn to dynamically adapt its policy based on a singular observation. This ability for rapid adaptation is a hallmark of human cognition, and is most closely related to one shot learning. While much progress has been recently made in realizing one shot learning [3, 10, 13, 18], these have been limited to single agent systems. This is in contrast to most real world tasks where decision making is often distributed among multiple agents interacting across time and space. One shot learning on such systems is particularly challenging as local changes in agent policies can have unpredictable consequences in emergent system behavior.

One approach to facilitate one shot learning is the incorporation of memory which can be used to remember salient observations and recall them in the future [15]. For example, in the season task the agent can incorporate a working memory to remember whether a food is poisonous/nutritious after sampling it once. The agent can then consider these alongside its input to make decisions [11]. Adaptive one shot decision making for single agent systems have been explored in the past largely with the tools of memory [1, 2]. However, the increased complexity from multiple agents acting concurrently is widely unexplored for this class of tasks.

The standard medium of incorporating memory within learning is Long Short Term Memory (LSTM) [5, 6]. LSTM is a type of Recurrent Neural Network and is the state of the art in many sequence processing tasks [4, 14]. A central feature within LSTM is its continuous blending of observations (new information) with salient past information stored within its cell (memory). However, in a multiagent domain where many agents concurrently observe new information and seek to update a shared memory, this leads to high degrees of interference and information degradation.

2 DISTRIBUTED MODULAR MEMORY UNIT

We introduce a memory-based multiagent framework called Distributed Modular Memory Unit (DMMU), exploiting the modularity and gating structures of the MMU network [7, 8] and building on [9]. DMMU (depicted in 1) is capable of concurrently processing sequential sets of observations from multiple agents.

Each information stream originates from an actor with individual agency. Each actor can be considered as an agent with its own unique policy, observing the environment and acting within it independently. Each of these agents is defined by a standard feedforward neural network with three major learnable components.

- **Input Gate:** The input gate filters the flow of information that comes from the environment to the agent. 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.
- **Curated Memory Feed:** The agent's input is augmented with content 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 with extraneous information it might not need at a particular time.
- **Selective Memory Update:** The agent is augmented with a learnable write gate that allows it to selectively update the contents of the external memory. This gate allows the agent to report salient features from its observations to the external memory. The write gate that filters this channel of

*Titlenote — e.g., used for things like "This article extends an earlier paper titled XYZ", and "equal contribution by the first two authors".

Proc. of the 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2019), N. Agmon, M. E. Taylor, E. Elkind, M. Veloso (eds.), May 13–17, 2019, Montreal, Canada.

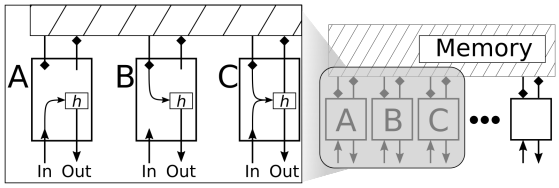

Figure 1: High level schematic of the DMMU framework. Each agent (emphasized in bubble) is comprised of an augmented feedforward neural network with connections to the world (input/output) and memory (read/write) connections. The modularity of the framework is highlighted by the differences in agents A, B, and C. At this time step, agent A ignores memory, and acts reactively based on its input. Agent B meanwhile ignores its input and acts exclusively from memory. Agent C is leveraging all available information, combining the contents within memory and its immediate input to make a decision. Agent C is also updating memory based on its decision.

information flow serves to shield the external memory from being overwhelmed by updates from the agent.

We test DMMU in a simulated cybersecurity task [9]. This task can be thought of as a multiagent extension of the season task [11] where a distributed multiagent system has to adapt its global policy based on a singular observation by one of its agent. In this task, a web server which operates via a distributed set of proxy servers has to withstand a DDoS attack. The web server receives multiple requests for fulfillment originating from multiple devices. A portion of these devices are conscripted by a botnet while the rest represnet genuine users. To succeed in this task, the proxy servers have to coordinate in sampling the incoming requests in parallel, determine which devices are nefarious, and selectively serve requests originating from the genuine ones. The core difficulty here is that the nefarious/genuine categorization of devices changes across task instances. This prevents the servers from simply memorizing action-value functions and forces it to dynamically assess the nefarious/genuine categorization for each new instance of the task. Additionally, unlike [9] the servers take variable number of timesteps (determined randomly) to fulfill requests. This is termed the **busy period** and adds an extra layer of difficulty originating from the asynchronisity across agents in accessing the shared memory.

3 RESULTS

We compared DMMU in the cybersecurity task with four baselines spanning the centralized/decentralized and memoried/reactive characteristics. Feedforward Neural Ensemble **(FFNE)** and LSTM Neural Ensemble **(LSTMNE)** represents each proxy server as a standard feedforward neural network and a LSTM, respectively. Centralized Neural Framework **(CNF)** represents the entire set of servers as a single feedforward neural network that has full access to all the information and makes centralized decisions. LSTM with Shared memory **(LSTMSM)** represents each server as an LSTM with access to an external shared memory mimicking the DMMU framework.

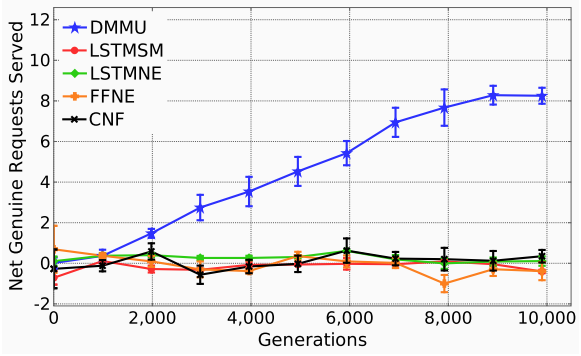

Figure 2: Comparative performance in the cybersecurity task. The server fleet consists of 10 proxy servers handling a request volume of 100 from 20 distinct devices. A random number of devices between {7,13} out of 20 are nefarious while the rest are genuine. Asynchronous memory access by agents with the busy period following each action randomly set between {0,2} timesteps.

Figure 2 shows the comparative performance of DMMU with other baselines. DMMU significantly outperforms other baselines achieving a net of 8.25 ± 0.39 genuine requests served. CNF and FFNE fail to learn entirely. This is unsurprising as these approaches lack memory and are unable to associate actions with rewards over time. Even with centralized access to information and centralized joint action (CNF), the lack of memory is a principal limitation.

Interestingly, LSTMNE also fail to learn despite having access to memory. However, unlike DMMU, LSTMNE does not have a shared memory which is essential in learning to consolidate observations between agents and is thus limited to greedy behaviors.

The most surprising perhaps is the failure to learn for LSTMSM agents which has an external shared memory similar to DMMU. However, unlike DMMU where agents can selectively read and write to memory, LSTMSM agents are constrained to base their actions as a function of the memory content. This limits their flexibility and leads to homogenization of their joint action. This emphasizes the importance of DMMU's modular design which allows agents to read from, add to, or ignore the shared memory dynamically. This enables DMMU to learn effective coordination strategies among a diverse set of agents to jointly sample observations, and share pertinent information leveraging the external memory.

4 CONCLUSION

As most real world systems move towards decentralization [12, 16, 17], multiagent one shot learning where agents can dynamically change their behavior based on an observation by one of the agents is an important challenge to tackle. In this work, we formulated DMMU that leverages a shared memory as a form of 'knowledge base' that can be collectively read from and updated by a team of autonomous agents. This enables the collective to adapt instantaneously based on a singular observation.

ACKNOWLEDGMENTS

Acknowledgments: This work was partially supported by the Electric Power Research Institute under Grant No. 10008781.

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] Yan Duan, Marcin Andrychowicz, Bradly Stadie, OpenAI Jonathan Ho, Jonas Schneider, Ilya Sutskever, Pieter Abbeel, and Wojciech Zaremba. 2017. One-shot imitation learning. In *Advances in neural information processing systems*. 1087–1098.

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

[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] Jen Jen Chung Khadka, Shauharda and Kagan Tumer. 2019. Neuroevolution of a Modular Memory-Augmented Neural Network for Deep Memory Problems. *Evolutionary computation* (2019).

[8] 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.

[9] Shauharda Khadka, Connor Yates, and Kagan Tumer. 2018. A Memory-based Multiagent Framework for Adaptive Decision Making. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*. International Foundation for Autonomous Agents and Multiagent Systems, 1977–1979.

[10] Brenden M Lake, Ruslan R Salakhutdinov, and Josh Tenenbaum. 2013. One-shot learning by inverting a compositional causal process. In *Advances in neural information processing systems*. 2526–2534.

[11] 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)*.

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

[13] Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy Lillicrap. 2016. One-shot learning with memory-augmented neural networks. *arXiv preprint arXiv:1605.06065* (2016).

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

[15] 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.

[16] Melanie Swan. 2015. *Blockchain: Blueprint for a new economy.* " O'Reilly Media, Inc.".

[17] 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.

[18] Oriol Vinyals, Charles Blundell, Tim Lillicrap, Daan Wierstra, et al. 2016. Matching networks for one shot learning. In *Advances in Neural Information Processing Systems*. 3630–3638.