Multiple Sequential Learning Tasks Represented in Recurrent Neural Networks

Shaonan Wang
National Laboratory of Pattern Recognition,
Institute of Automation Chinese Academy of Sciences
shaonan.wang@nlpr.ia.ac.cn

Bingyu Liu
CAS Center for Excellence in Brain Science and Intelligence Technology,
Key Laboratory of Primate Neurobiology,
Institute of Neuroscience Chinese Academy of Sciences
byliu@ion.ac.cn

Abstract

Our brain can flexibly perform a variety of sequential learning tasks including music, language, and mathematics, but the underlying mechanism hasn't been elucidated in traditional experimental and modeling studies which were designed for only one task at a time. From the computational perspective, we hypothesize that the working mechanism of a multitask model can provide a possible solution to that of brains. Therefore, we trained a single recurrent neural network to perform 8 sequential learning tasks that depend on working memory, structure extraction, categorization, and other cognitive processes. After training, the model can learn sophisticated information holding and erasing strategies to perform multitasks simultaneously. More interestingly, the model learns to reuse neurons to encode similar task features. Hopefully, this work can provide a computational platform to investigate the neural representations of cognitive sequential learning ability.

1 Introduction

Sequential learning is essential to the daily activities of animals and especially humans, such as music, speech, language, and mathematics. In recent years, researchers have explored the abilities of sequential learning in both primates and humans based on visual and auditory cues, elucidating the crucial role of attention, timing encoding, working memory in different types of sequential tasks. Previous work has found that different brain areas play different roles in general sequential learning. For instance, ordinal numbers are ubiquitously represented in ventral intraparietal cortex (VIP) neurons in macaque parietal cortex [12], while prefrontal cortex (PFC) is related to working memory [8, 3], and the hippocampal area stores timing information [10, 4]. There were also studies showing that the perirhinal cortex (PRC) can integrate item signals from corresponding sensory cortices and temporal order information from the hippocampus, which will be transmitted to PFC for motor planning [10, 11]. However, although there are so many studies in this field, none of them has ever been able to explore the multiple sequential learning abilities mostly because it's very hard to train the animals in the electrophysiological experiment[14].

To explore the potential mechanisms, we took the approach of training recurrent neural network (RNN) models. Our premise hypothesis is that if computational models can perform multiple sequential learning tasks, then the internal working mechanism provides a possible explanation for animal sequential learning ability. Specifically, we investigated the following questions:

NeurIPS 2021 AI for Science Workshop.

1. Can one RNN model accomplish multiple sequential learning tasks simultaneously?
2. What is the internal mechanism of the above multitask RNN model?
3. How are the different features stored at the neuronal level for each task?
4. If two tasks share similar structures, what would be the relationship between their neural representations?

2 Setups

We first designed 8 sequential learning tasks which were simplified from previous physiological studies[4, 6, 7, 13]. There has been a proliferation of research work on sequential learning both in humans and non-human primates in recent years. They either let the humans or monkeys repeat the sequence of visual or auditory items in forward or inverted order, or the subjects needed to report whether the sequence order of certain pictures is the same as the one that appeared before some delay time [7, 1]. As shown in Figure 1, we clarified the tasks as Repeat (repeat the sequence); Mirror (report the sequence in inverse order); Pre (report the symbol before a specified position); Pos (report the symbol after a specified position); Add first (repeat the sequence and add the first symbol in the beginning); Add last (repeat the sequence and add the last symbol in the end); Shift (shift the position of the first two symbols and report); and Swap (swap every two symbols according to the order. If the sequence is odd, then just report the symbol at the end).

Repeat:	$x_1, x_2, \dots, x_n$	$\longrightarrow$	$x_1, x_2, \dots, x_n$
Mirror:	$x_1, x_2, \dots, x_n$	$\longrightarrow$	$x_n, x_{n-1}, \dots, x_1$
Pre (the second):	$x_1, x_2, \dots, x_n$	$\longrightarrow$	x_1
Pos (the second):	$x_1, x_2, \dots, x_n$	$\longrightarrow$	x_3
Add first:	$x_1, x_2, \dots, x_n$	$\longrightarrow$	$x_1, x_1, x_2, \dots, x_n$
Add end:	$x_1, x_2, \dots, x_n$	$\longrightarrow$	$x_1, x_2, \dots, x_n, x_n$
Shift (the first two) :	$x_1, x_2, \dots, x_n$	$\longrightarrow$	$x_2, x_1, \dots, x_n$
Swap (each pair) :	Even $x_1, x_2, \dots, x_{n-1}, x_n$	$\longrightarrow$	$x_2, x_1, \dots, x_{n-1}, x_n$
	Odd $x_1, x_2, \dots, x_n$	$\longrightarrow$	$x_2, x_1, \dots, x_n$

Figure 1: Definition of 8 sequential learning tasks. The same color denotes a similar task structure.

To accomplish these tasks, we adopt a sequence tagging paradigm. Figure 2 shows an example of the Repeat task. As shown, model inputs are x_1, x_2, x_3, followed by a *cue* to indicate which task to perform, and it needs to predict x_1, x_2, x_3 at the last three-time steps.

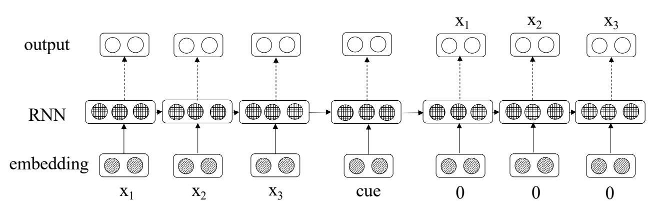

Figure 2: Architecture of sequence tagging for the Repeat task.

Miller and Cowan have summarized that the capacity of working memory is limited to about seven or four chunks according to a variety of experimental data [9, 2]. Therefore, we used the same range of sequence numbers in our experiments. Specifically, the training data comprises 20,000 sequences with a length of 4, 5, or 7 (and each symbol is randomly selected from 1 to 6). To successfully perform sequential learning tasks, the model should have the ability to generalize, thus, the testing data set is randomly selected from a length of 3 to 8 (each length consists of 2,000 sequences). Besides, the multitask RNN model is trained by randomly pick a batch of data from one task at a time. For comparison, we also trained single task RNN with the same settings. In all experiments, we used an advanced RNN, called Long-short-term memory (LSTM) network [5], with one layer comprising 50 hidden neurons and a batch size of 32. We have also tried a simple RNN model without gating and memory cell mechanism, but it failed on the generalization tests.

3 Results

3.1 Can one RNN model accomplish multiple sequential learning tasks simultaneously?

We first tested the ability of the RNN model to solve all tasks together or individually. As shown in Figure 3, the multitask RNN model achieves excellent performance on all 8 tasks with different sequence lengths (The same results were found for single task RNN). An intriguing result is that the performance of the model is not linearly decreased with the ordinal position of each item, instead shows a "U"-shape-curve, which is similar to that in humans and animals [7]. The reason behind this would be an interesting future work to explore.

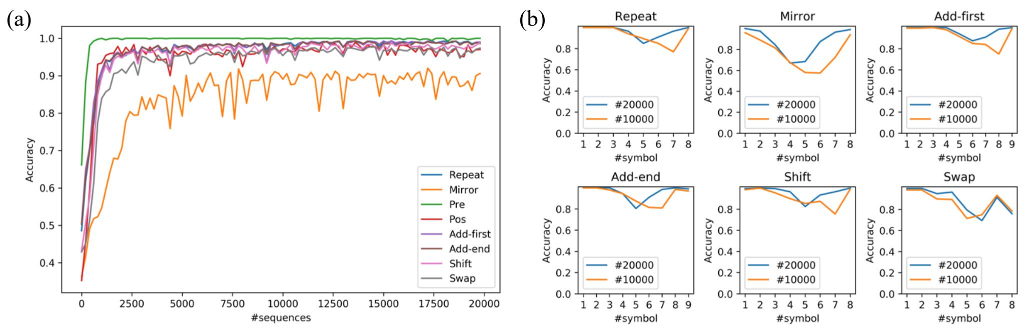

Figure 3: Multitask RNN accuracy, where (a) shows the average accuracy over all time points and (b) shows the accuracy at each time point with a sequence length of 8. Task Pre and Pos are not shown since they only have one output.

3.2 What is the internal mechanism of the above multitask RNN model?

To explore the model's working mechanism, we proposed a label-clustering (LC) method to measure how much information of the target input is maintained in every time step. Taking the repeat task for instance, we have a set of repeat sequences x_0, x_1, x_2, x_3, x_4 in which each symbol appears in a specific time point from t_0 to t_4. At each time point t_i, the network needs to receive a new input symbol x_i and maintain the historical information of symbol $x_{1:i-1}$ in the meantime. For instance, we want to calculate how much information of symbol x_1 is maintained at time t_2. To achieve this goal, the LC method first marks the value of symbols at time t_2 as the value of symbols at time t_1. Then, it extracts the hidden embedding at t_2, resulting in a set of 50-dimensional vectors each with a corresponding label. Finally, the LC method calculates the aggregation degree (Davies-Bouldin index, DBI) of the same label. In short, the greater the difference between intra-class and inter-class aggregation, the more information the symbol x_1 contains at time point t_2.

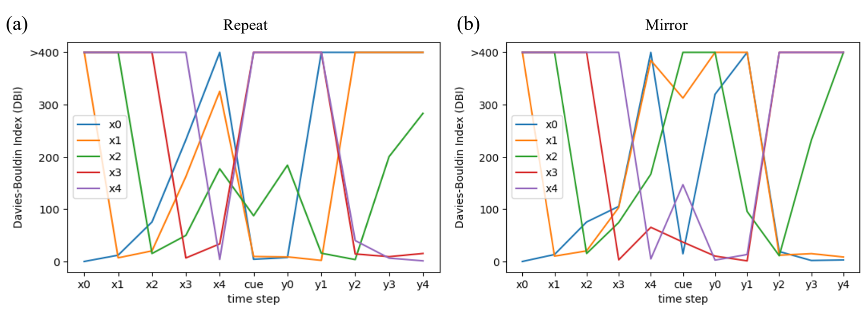

Figure 4: LC results for (a) Repeat task (b) Mirror task with the sequence input length of 4.

Figure 4 shows that the RNN model learns to hold and erase information at the right time (Similar results were found on all 8 tasks). In the encoding phase ($x_0 - x_4$), some information is maintained (low DBI) and others are lost (higher DBI) over time. In the decoding phase ($y_0 - y_4$), most of the corresponding information is erased after the symbols are output. Interestingly, we found that single and multiple task RNN models seem to learn different strategies and the latter one can learn more precise sequential learning rules, e.g., predict the next symbol at the cue time (low DBI of the blue line at cue in the Repeat task).

3.3 How are the different features stored at the neuronal level for each task?

To successfully perform the 8 tasks, the RNN model should: (1) encode and store input symbols along with their appearance time, and (2) extract and output the right symbols at the right time. Therefore, the symbol and time features are crucial to accomplish these tasks. Next, we analyzed relations between the two features with individual neurons in the hidden layer of the RNNs model.

We conducted an ablation test by predicting these features with one neuron shielded. The hypothesis is that if the network uses a local or sparse coding, then there should be a small set of neurons that encode a specific feature, thus ablating them would lead to a drastic performance drop (or error rise).

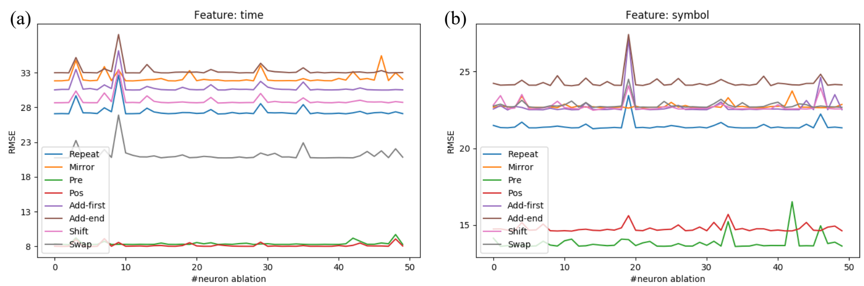

Figure 5: Root-mean-square error (RMSE) of feature (a) time and (b) symbol when masking one neuron at a time.

As shown in Figure 5, there are single neurons that are more important to one feature than others, such as neuron 9 for time and neuron 19 for symbol. We then masked the top 10 neurons that are sensitive to each feature and found that the left neurons can still predict the two features pretty well. Therefore, we conclude that no single neuron is responsible for all behaviors of any features, thus, task features are stored in RNN in a distributed-represented way rather than a local-represented way.

3.4 If two tasks share similar structures, what would be the relationship between their neural representations?

From Figure 5, we can also find that multitask RNN learns to use the same neurons when performing similar tasks. For instance, Pre and Pos tasks have similar error rises; Repeat, Add-first, and Add-end have similar error rises. Note that this phenomenon has not been found on single-task RNN, which indicates that the RNN model can learn more physiologically meaningful representations only with simultaneous training on different tasks.

4 Conclusions and Future Work

We trained a single RNN to perform multiple sequential learning tasks at the same time, hoping to provide a computational platform to investigate the neural representations of cognitive sequential learning ability. An interesting future work is to explore complicated sequential learning tasks such as sequences with a nested structure which is important to animals' sequential learning [4]. To explore that, we need to first simplify and formalize these tasks to a unified architecture. Then we may need more powerful computational models than RNN such as the Neural Turing Machine (NTM) model which has an extra memory module to flexibly manipulate information flow. The purpose of the computational models is to help us understand the mechanism of the brain and to figure out how the real brain truly solves these problems. Therefore, we also need to compare the results of computational models with real neural data such as the electrophysiological data recorded by experimental scientists.

Furthermore, we can also build a brain-inspired computational model to do these sequential learning tasks. For instance, as we proposed before, for the visual sequential learning task, the visual items are originated from the anterior-inferior temporal cortex (area TE); the timing and structure information of the sequence are represented in the posterior parietal cortex (PPC) and hippocampus. All these will then be integrated into PFC as working memory. Accordingly, RNN with biology-inspired working memory and multimodal integration modules may be more intelligent in accomplishing multitask sequential tasks and more similar to the brain. In this way, computational models and the brain assisted each other, driving both artificial intelligence and neuroscience forward.

Acknowledgments and Disclosure of Funding

This work was conducted while both authors were at the 2021 Computational and Cognitive Neuroscience (CCN) summer school. Thanks all the organizers, lecturers, TAs, and classmates for this wonderful summer. Moreover, this work is supported by the Natural Science Foundation of China under Grant 61906189 and 62036001.

References

[1] Adam F Carpenter, Gabriel Baud-Bovy, Apostolos P Georgopoulos, and Giuseppe Pellizzer. Encoding of serial order in working memory: neuronal activity in motor, premotor, and prefrontal cortex during a memory scanning task. *Journal of Neuroscience*, 38(21):4912–4933, 2018.

[2] Nelson Cowan. The magical number 4 in short-term memory: A reconsideration of mental storage capacity. *Behavioral and brain sciences*, 24(1):87–114, 2001.

[3] Clayton E Curtis and Mark D'Esposito. Persistent activity in the prefrontal cortex during working memory. *Trends in cognitive sciences*, 7(9):415–423, 2003.

[4] Stanislas Dehaene, Florent Meyniel, Catherine Wacongne, Liping Wang, and Christophe Pallier. The neural representation of sequences: from transition probabilities to algebraic patterns and linguistic trees. *Neuron*, 88(1):2–19, 2015.

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

[6] Dieuwke Hupkes, Verna Dankers, Mathijs Mul, and Elia Bruni. Compositionality decomposed: how do neural networks generalise? *Journal of Artificial Intelligence Research*, 67:757–795, 2020.

[7] Xinjian Jiang, Tenghai Long, Weicong Cao, Junru Li, Stanislas Dehaene, and Liping Wang. Production of supra-regular spatial sequences by macaque monkeys. *Current Biology*, 28(12):1851–1859, 2018.

[8] Matthew I Leon and Michael N Shadlen. Representation of time by neurons in the posterior parietal cortex of the macaque. *Neuron*, 38(2):317–327, 2003.

[9] George A Miller. The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological review*, 63(2):81, 1956.

[10] Yuji Naya, He Chen, Cen Yang, and Wendy A Suzuki. Contributions of primate prefrontal cortex and medial temporal lobe to temporal-order memory. *Proceedings of the National Academy of Sciences*, 114(51):13555–13560, 2017.

[11] Yuji Naya and Wendy A Suzuki. Integrating what and when across the primate medial temporal lobe. *Science*, 333(6043):773–776, 2011.

[12] Christopher Summerfield, Fabrice Luyckx, and Hannah Sheahan. Structure learning and the posterior parietal cortex. *Progress in neurobiology*, 184:101717, 2020.

[13] Melissa R Warden and Earl K Miller. The representation of multiple objects in prefrontal neuronal delay activity. *Cerebral Cortex*, 17(suppl_1):i41–i50, 2007.

[14] Guangyu Robert Yang, Madhura R Joglekar, H Francis Song, William T Newsome, and Xiao-Jing Wang. Task representations in neural networks trained to perform many cognitive tasks. *Nature neuroscience*, 22(2):297–306, 2019.