Emergence of the Primacy Effect in Structured State-Space Models

Takashi Morita
Academy of Emerging Sciences, Chubu University
1200 Matsumoto-cho
Kasugai, Aichi 487-8501 JAPAN
Institute for Advanced Research, Nagoya University
Furo-cho, Chikusa-ku
Nagoya, Aichi 464-8601 JAPAN

Abstract

Human and animal memory for sequentially presented items is well-documented to be more accurate for those at the beginning and end of the sequence, phenomena known as the *primacy* and *recency* effects, respectively. By contrast, artificial neural network (ANN) models are typically designed with a memory that decays monotonically over time. Accordingly, ANNs are expected to show the *recency* effect but not the *primacy* effect. Contrary to this theoretical expectation, however, the present study reveals a counterintuitive finding: a recently developed ANN architecture, called *structured state-space models*, exhibits the primacy effect when trained and evaluated on a synthetic task that mirrors psychological memory experiments. Given that this model was originally designed for recovering neuronal activity patterns observed in biological brains, this result provides a novel perspective on the psychological primacy effect while also posing a non-trivial puzzle for the current theories in machine learning.

Keywords: primacy effect; state space models; long-term memory; artificial neural networks

Introduction

Human and animal memory for sequentially presented items is well-documented to be more accurate for those appearing at the beginning and end of the sequence—phenomena known as the *primacy* and *recency* effects, respectively (Ebbinghaus, 1913; Glanzer & Cunitz, 1966; Murdock, 1962). For example, when a sequence of random integers such as $49, 75, \ldots, 5, 38$ is presented in that order, the initial $(49, 75)$ and final $(5, 38)$ items are more likely to be recalled accurately at the end of the presentation.

By contrast, artificial neural network (ANN) models are typically designed with a memory that decays monotonically over time (Bengio, Simard, & Frasconi, 1994; Jaeger, 2001; Jaeger & Haas, 2004). Thus, ANNs are expected to show the *recency* effect but not the *primacy* effect.

Contrary to this theoretical expectation, however, the present study reveals a counterintuitive finding: a recently developed ANN architecture—called *structured state-space models* (Gu, Dao, Ermon, Rudra, & Ré, 2020; Gu, Goel, & Ré, 2022)—exhibits the primacy effect when trained and evaluated on a synthetic task that mirrors psychological memory experiments. Given that this model was originally designed for recovering neuronal activity patterns observed in biological brains, this result provides a novel perspective on the psychological primacy effect while also posing a non-trivial puzzle for the current theories in machine learning.

The remainder of this paper is organized as follows. The next section first reviews ANN models for time-series processing. After this preliminary discussion, the Methods section details the task and model specifications of the present study, and the subsequent section presents the results. Finally, the Discussions section summarizes the findings and discusses their implications in relation to previous research.

Methods

Task

The memorization patterns of ANNs were assessed using the binary memory verification task (a.k.a. *serial probe recognition*; Sands & Wright, 1980; Thompson & Herman, 1977; Wickelgren & Norman, 1966; Wright, Santiago, Sands, Kendrick, & Cook, 1985). In this task, the models were first presented with a sequence of randomly generated, non-repeating integers (hereinafter referred to as *study items*). Subsequently, they received another sequence of integer queries and were trained to determine whether each query token was present (labeled as 1) or absent (labeled as 0) in the study items. To construct these queries, the study items were first shuffled, and then, with a probability of $p = 0.5$, each shuffled token was replaced with a randomly sampled integer from the complement set of the study items (termed *distractors*).

The task hyperparameters were manually adjusted to prevent the models from achieving perfect accuracy. Specifically, the input length was set to $L \in \{64, 128, 256\}$, and the vocabulary size was fixed at $K := 4096$. Each model underwent ten independent training runs with different random seeds. For evaluation, 1024 sets of integers were held out as test data, ensuring that these integer combinations never appeared as study items in the training set, regardless of their order.

To build test sequences, the held-out study items were randomly ordered, and queries were generated by first shuffling and then cyclically shifting them (e.g., $(2, 8, 11, 29) \mapsto \{(2, 8, 11, 29), (8, 11, 29, 2), (11, 29, 2, 8), (29, 2, 8, 11)\}$). This design ensured that each study item was queried in all L possible positions. Finally, either the even- or odd-indexed query positions were replaced with random distractors, resulting in a total of $1024 \times L \times 2$ test sequences per trial.

Models

The models used for the binary memory verification task comprised three layers. In the first layer, the input integers were embedded into 256-dimensional real-valued vectors. These embeddings were shared between study items and query tokens. The resulting sequence of vectors was then processed by the SSM/RNN, whose outputs were linearly projected onto binary logits to determine whether each query token was present in the study items.

This study primarily examined the single-layer S4 model as the goldstandard implementation of the SSM (Gu, Goel, & Ré, 2022).[1] The model encoded the channel-wise dynamics of the input embeddings in a complex-valued space, with its outputs subsequently projected back into the real domain by discarding imaginary components. The state and input matrices were initialized to approximate each channel's trajectory using Legendre/Laguerre polynomials of degrees 0–63 (HiPPO-LegS/LagT) or a Fourier basis $\{s_0, c_0, \dots, s_{31}, c_{31}\}$, where $s_n(t) := \sqrt{2}\sin(2\pi nt)$ and $c_n(t) := \sqrt{2}\cos(2\pi nt)$ (HiPPO-Fout, Fourier Recurrent Unit; Gu et al., 2020; Gu, Johnson, Timalsina, Rudra, & Ré, 2023; Zhang, Lin, Song, & Dhillon, 2018). The matrices were discretized by the bilinear method (Tustin, 1947).

For comparison, a single-layer long short-term memory (LSTM) network was also evaluated (Hochreiter & Schmidhuber, 1997). LSTM has been the goldstandard RNN architecture for various time-series processing tasks, including language modeling (Graves, 2013; Sundermeyer, Schlüter, & Ney, 2012). The dimensionality of both hidden and cell states was set to 256.

The models were trained for 300,000 iterations using the Adam optimizer with parameters $(\beta_0, \beta_1) := (0.9, 0.99)$ (Kingma & Ba, 2015). Batch size was set to 512. The learning rate was linearly increased from 0.0 to 0.001 over the first 1,000 iterations (*warmups*) and subsequently decayed according to the cosine annealing schedule Loshchilov and Hutter (2017). To prevent gradient explosion, the gradient norm was clipped at 1.0.

Results

The binary memory verification performance of the SSM model was highest for study items presented at the beginning of the sequence. The model maintained high accuracy across different query timings, provided that the sequence length did not exceed its capacity. In other words, memory for the initial study items exhibited minimal decay over time.

By contrast, the LSTM did not display this primacy effect; its accuracy was uniform across both the memorization and verification phases.

Interestingly, the SSM's accuracy for the most recently presented study items was lowest when they were queried immediately after their initial presentation in the memorization phase. This suggests a temporal delay between the encoding of study items and their effective retrieval.

These findings held true regardless of whether the state and input matrices of the SSM were optimized for the task or remained fixed at their initial values. Moreover, the results remained consistent across different polynomial bases underlying the state and input matrices, including Laguerre, Fourier, and Legendre.

Acknowledgments

This study was supported by JST ACT-X (JPMJAX21AN) and Core Research for Evolutional Science and Technology (JPMJCR22P5); JSPS Grant-in-Aid for Early-Career Scientists (JP21K17805) and for Scientific Research A (JP24H00774), B (JP22H03914), and C (JP24K15087); and Kayamori Foundation of Informational Science Advancement (K35XXVIII620). The author also gratefully acknowledges the support of the ACCMS, Kyoto University, regarding the use of their supercomputer system.

[1] Recent studies have shown that the state matrix (A) of S4 can be simplified into a purely diagonal form without compromising performance (S4D; Gu, Goel, Gupta, & Ré, 2022). By contrast, the original S4 model introduced an additional low-rank component to the diagonal structure (referred to as the Diagonal Plus Low Rank form, or DPLR) to ensure a mathematically well-founded state matrix. Notably, the diagonal variant exhibited a qualitatively similar primacy effect to the DPLR model. Due to the page limitations, results for the diagonal model are omitted from this paper, and all reported findings are based on the DPLR model.

References

Bengio, Y., Simard, P., & Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, *5*(2), 157–166. doi: 10.1109/72.279181

Ebbinghaus, H. (1913). *Memory: A contribution to experimental psychology* (H. A. Ruger & C. E. Bussenius, Eds.). Teachers College Press. doi: 10.1037/10011-000

Glanzer, M., & Cunitz, A. R. (1966). Two storage mechanisms in free recall. *Journal of Verbal Learning and Verbal Behavior*, *5*(4), 351–360. doi: 10.1016/S0022-5371(66)80044-0

Graves, A. (2013). *Generating sequences with recurrent neural networks.* arXiv:1308.0850.

Gu, A., Dao, T., Ermon, S., Rudra, A., & Ré, C. (2020). HiPPO: Recurrent memory with optimal polynomial projections. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, & H. Lin (Eds.), *Advances in neural information processing systems* (Vol. 33, pp. 1474–1487). Curran Associates, Inc.

Gu, A., Goel, K., Gupta, A., & Ré, C. (2022). On the parameterization and initialization of diagonal state space models. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, & A. Oh (Eds.), *Advances in neural information processing systems* (Vol. 35, pp. 35971–35983). Curran Associates, Inc.

Gu, A., Goel, K., & Ré, C. (2022). Efficiently modeling long sequences with structured state spaces. In *Proceedings of the tenth international conference on learning representations (ICLR).* OpenReview.net.

Gu, A., Johnson, I., Timalsina, A., Rudra, A., & Ré, C. (2023). How to train your HIPPO: State space models with generalized orthogonal basis projections. In *Proceedings of the eleventh international conference on learning representations (ICLR).* OpenReview.net.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735–1780. doi: 10.1162/neco.1997.9.8.1735

Jaeger, H. (2001). *Short term memory in echo state networks* (Tech. Rep.). Sankt Augustin: German National Research Center for Information Technology. doi: 10.24406/publica-fhg-291107

Jaeger, H., & Haas, H. (2004). Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication. *Science*, *304*(5667), 78–80. doi: 10.1126/science.1091277

Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. In *Proceedings of 3rd international conference on learning representations (ICLR).* San Diego, California.

Loshchilov, I., & Hutter, F. (2017). SGDR: stochastic gradient descent with warm restarts. In *Proceedings of the 5th international conference on learning representations (ICLR).* OpenReview.net.

Murdock, B. B. (1962, November). The serial position effect of free recall. *Journal of Experimental Psychology*, *64*(5), 482–488. doi: 10.1037/h0045106

Sands, S. F., & Wright, A. A. (1980). Primate memory: Retention of serial list items by a rhesus monkey. *Science*, *209*(4459), 938–940. doi: 10.1126/science.6773143

Sundermeyer, M., Schlüter, R., & Ney, H. (2012). LSTM neural networks for language modeling. In *Proceedings of INTERSPEECH* (pp. 194–197). doi: 10.21437/Interspeech.2012-65

Thompson, R. K. R., & Herman, L. M. (1977). Memory for lists of sounds by the bottle-nosed dolphin: Convergence of memory processes with humans? *Science*, *195*(4277), 501–503. doi: 10.1126/science.835012

Tustin, A. (1947). A method of analysing the behaviour of linear systems in terms of time series. *Journal of the Institution of Electrical Engineers - Part IIA: Automatic Regulators and Servo Mechanisms*, *94*, 130–142. doi: 10.1049/ji-2a.1947.0020

Wickelgren, W. A., & Norman, D. A. (1966). Strength models and serial position in short-term recognition memory. *Journal of Mathematical Psychology*, *3*(2), 316–347. doi: 10.1016/0022-2496(66)90018-6

Wright, A. A., Santiago, H. C., Sands, S. F., Kendrick, D. F., & Cook, R. G. (1985). Memory processing of serial lists by pigeons, monkeys, and people. *Science*, *229*(4710), 287–289. doi: 10.1126/science.9304205

Zhang, J., Lin, Y., Song, Z., & Dhillon, I. (2018, 10–15 Jul). Learning long term dependencies via Fourier recurrent units. In J. Dy & A. Krause (Eds.), *Proceedings of the 35th international conference on machine learning* (Vol. 80, pp. 5815–5823). PMLR.