NeuroZoo: A tool for training recurrent neural network models on behavioral tasks

193.10

Alperen Cimen[1,3,5], Mert Yuksekgonul[2], Andrew S. Alexander[3,4], Fatih Dinc[5,6]

[1]Department of Electrical and Electronics Engineering, Ozyegin University, Istanbul, Turkey; [2]Computer Science, Stanford University, San Francisco, CA 94301

[3]Psychological and Brain Sciences, Neuroscience Research Institute, [4]Dynamical Neuroscience, [5]Geometric Intelligence Lab, [6]Kavli Inst. for Theoretical Physics, UC Santa Barbara, Santa Barbara, CA 9311

ÖZYEĞİN UNIVERSITY

Psychological & Brain Sciences
UC SANTA BARBARA

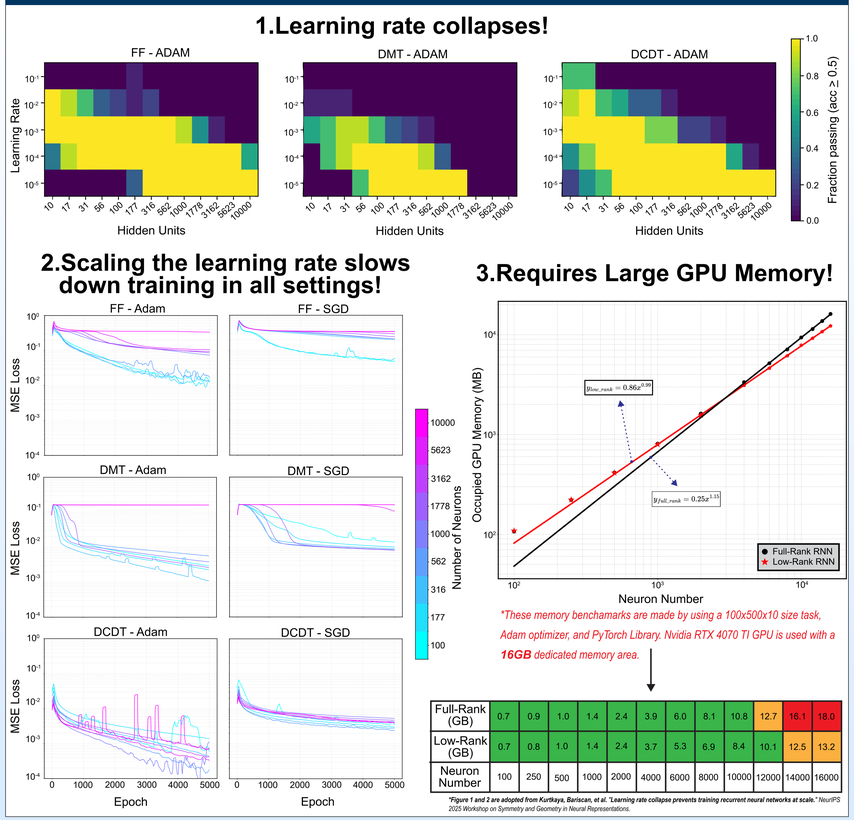
NRI Neuroscience Research Institute

Abstract

System neuroscientists regularly investigate cognitive functions by studying model rodents through various predefined experiments. These experiments reveal diverse neural mechanisms that subserve computation in rodents' brains. Recent advancements in artificial neural networks have enabled neuroscientists to expand these neural studies to silico models. To this end, computational neuroscientists regularly train recurrent neural networks (RNNs) on synthetically design cognitive tasks to gain mechanistic insights. However, even though recent work has shown that even the most trivial changes in task parameters lead to diverse neural circuits learned by RNNs, most existing silico tasks are coded without clear standardization. Moreover, there have been several distinct learning rules developed in the last decade to train RNNs, each of which can favor one neural mechanism over another. To investigate and standardize this process, we designed an extensive dataset for various cognitive model training purposes and benchmarked diverse RNN architectures trained on these tasks using several learning rules. We extracted the learned latent mechanisms in these diverse tasks and compared them across RNN architectures and learning paradigms. We also examined the persistence of learned mechanisms in RNNs trained to perform additional tasks. To facilitate future extensions of these benchmarks and support ongoing research in this field, we outline detailed methodologies for standardizing these tasks and provide them open-source. Our resource presents a standardized and principled approach for benchmarking various RNN models, paving the way for future comparative studies of neural computations in system neuroscience.

1. Training RNNs

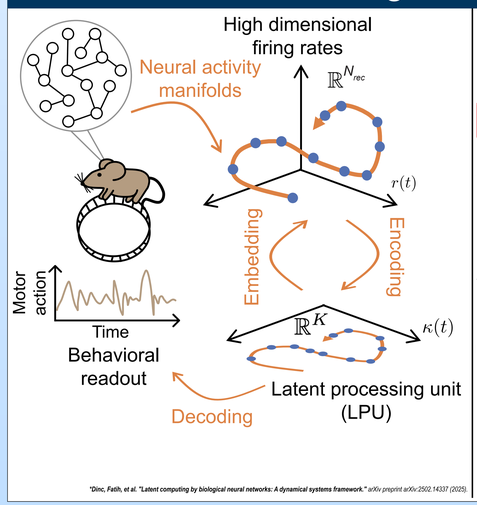
2. Why is it hard to train large RNNs?



Full-Rank (GB)	0.7	0.9	1.0	1.4	2.4	3.9	6.0	8.1	10.8	12.7	16.1	18.0
Low-Rank (GB)	0.7	0.8	1.0	1.4	2.4	3.7	5.3	6.9	8.4	10.1	12.5	13.2
Neuron Number	100	250	500	1000	2000	4000	6000	8000	10000	12000	14000	16000

Figure 1 and 2 are adopted from Kurtkaya, Barıscan, et al. "Learning rate collapse prevents training recurrent neural networks at scale." NeurIPS 2025 Workshop on Symmetry and Geometry in Neural Representations.

3. Latent Processing Unit (LPU) Framework



Equations

$$\tau \dot{r}(t)_{full_rank} = -r(t) + \phi(W_{rec} r(t) + W_{in} u(t))$$

Define

$$W_{\text{rec}} = \sum_{i=1}^{K} m_i n_i^T = MN \quad \text{where} \quad M \in \mathbb{R}^K, N \in \mathbb{R}^K$$

Then, the RNN equation becomes:

$$\tau \dot{r}(t)_{low_rank} = -r(t) + \phi(MNr(t) + W_{in} u(t))$$

Embedding Weight $(\mathbb{R}^K \to \mathbb{R}^N)$ — Encoding Weight $(\mathbb{R}^N \to \mathbb{R}^K)$

Then we define LPU (k) as follows:

$$\kappa(t) = Nr(t)$$

Finally, latent dynamics can be found as:

$$\dot{\kappa}(t) = -\kappa(t) + N\phi(M\kappa(t))$$

Scan the QR code for more details!

Dinc, Fatih, et al. "Latent computing by biological neural networks: A dynamical systems framework." arXiv preprint arXiv:2502.14337 (2025).

3. SSL Framework