PRL: Can Parallel Reservoir Computing with Linear Modules Support DDM?

Ruiran Yang* and Zhiqiu Zhang*

Summary

The DDM model, as a popular and easily analyzable mathematical model, has been shown to effectively characterize animal decision-making processes and widely used in behavioral decision research[1, 2]. By simulating the decision-making process with DDM, we can study the complex underlying biological neural network mechanisms of decision-making. In recent years, probabilistic reasoning tasks have been shown to be modelable using GRU models[3]. However, deep learning models suffer from poor interpretability and inefficiency. We notice that reservoir computing, another form of recurrent neural network[4, 5], only requires training the output layer. Its representational ability is highly dynamic, related to the internal reservoir, and exhibits dynamic characteristics similar to attractor models[6, 7]. Therefore, we argue that external events can be represented by the dynamic characteristics of the reservoir, with no need for learning. The model only needs to learn the decision-making process, thereby achieving the effect of the DDM model. We propose the **PRL model** proved feasible through extensive experiments and analysis. Our findings provide a possible explanation for this efficient learning approach, as well as generalization of biological neural networks. Code is available at https://github.com/yrrIsYourFather/PRL.

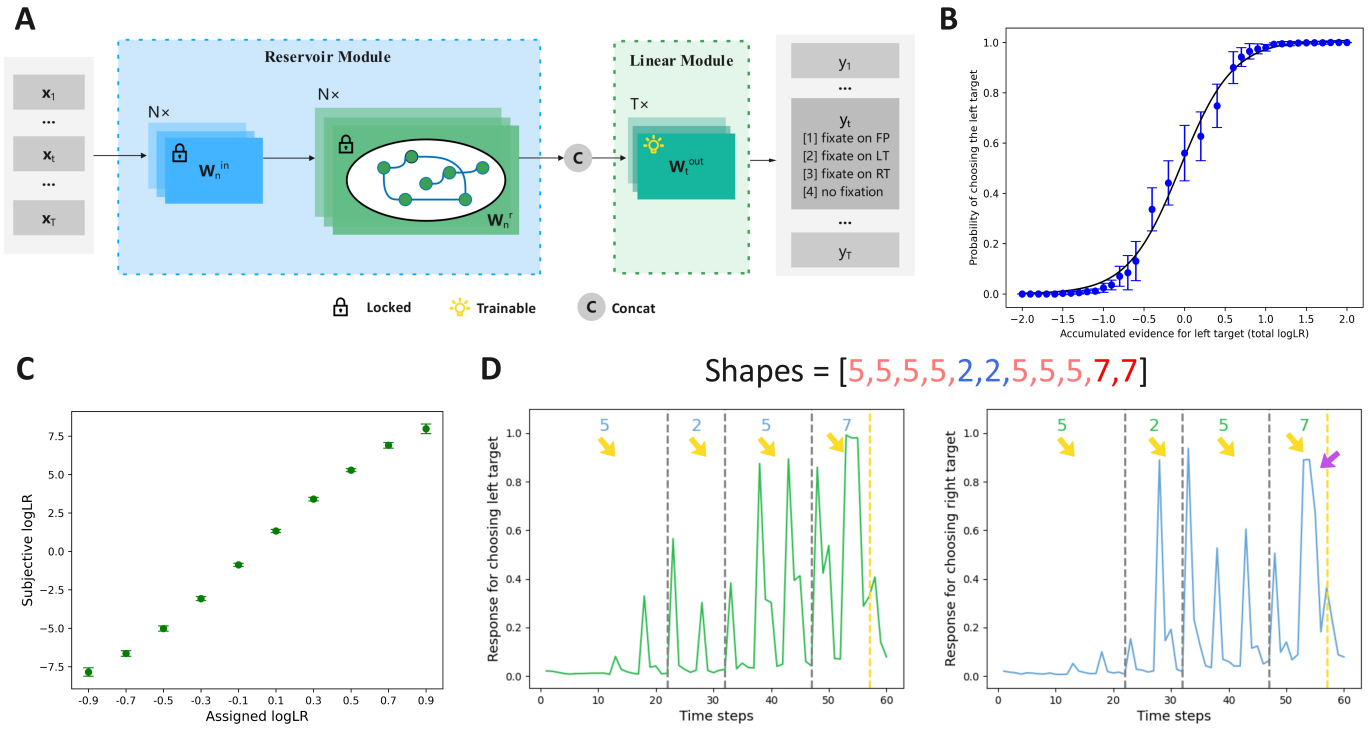

Figure 1: **(A)**: PRL model consists of several reservoir layers with randomly-initialized and frozen parameters to extract features, and several trainable simple linear modules to make predictions. **(B)**: The impact of cumulative log LR on the probability of selection. **(C)**: The influence of different shapes on target choosing. **(D)**: Qualitative results of a particular shape sequence on the model output.

PROBLEM FORMULATION

We follow the first task described in [3] to study a sequential decision-making task in [8]. Similarly, we assign logLR weights to each shape, and the model needs to learn these weights from the sequence of shapes and integrate the information to make the correct decision.

We consider the problem as a classification task of generic V-dimensional multivariate time series (MTS) with T time instants, whose observation at time t is denoted as $\mathbf{x}(t) \in \mathbb{R}^V$. The model needs to output $y(t)$ based on $\mathbf{x}(t)$, which serves as the decision for the next time step.

PRL MODEL

Model architecture Our **P**arallel **R**eservoir-**L**inear (PRL) model (see fig. 1A) consists of N reservoir layers, each taking in the same input sequence and processing in parallel. At time t, an overall feature $\mathbf{h}(t)$ is generated by concatenating all partial features $\mathbf{h}_i(t)$ of the i-th($i = 1, \ldots, N, N = 6$) recurrent layer. The reservoir updating formulation reads:

$\mathbf{h}_i(t) = \tanh(\mathbf{W}_i^{\text{in}}\mathbf{x}(t) + \mathbf{W}_i^{\text{r}}\mathbf{h}_i(t-1))$, where matrices $\mathbf{W}_i^{\text{in}}$ and $\mathbf{W}_i^{\text{r}}$ are the i-th layer's weights of the input and recurrent connections, respectively. To avoid costly operation of back propagation through time (BPTT), RC takes a radical different direction; parameters $\theta = \{\mathbf{W}_{\text{in}}, \mathbf{W}_{\text{r}}\}$ are *randomly generated and left untrained*. To compensate for this lack of adaptability, a large number of recurrent layers, the reservoir generates a rich pool of features useful to solve many different tasks. Then, a learnable linear module is utilized to output predictions of the decisions. We need to mention that the linear modules at different time t are trained *separately*, which means that they do not share parameters.

Loss function Let us denote by $y(t) \in \mathbb{R}$ the ground truth decision *at time* t and $\hat{y}(t)$ the prediction. We further denote by $d \in \mathbb{R}$ the ground truth decision *of the whole sequence*, and $\hat{d}$ the prediction. Loss function is made up of two components. The first loss $\mathcal{L}_{\text{pred}}$ uses the cross entropy between $\hat{y}(t)$ and $y(t)$; while the second $\mathcal{L}_{\text{decision}}$ emphasizes decisions of the whole sequence, w.r.t $\hat{d}$ and d, which forces the model to make the same decision as ground truth. Overall, our loss is a linear combination of both defined as $\mathcal{L} = \mathcal{L}_{\text{pred}} + \lambda\mathcal{L}_{\text{decision}}$ where hyper-parameter λ is typically set to 10.

EXPERIMENTS

Details We follow the settings in [3] and generate a synthetic dataset based on the principles of DDM. However, we split and modify the dataset to adapt it to a multi-class classification problem. For the input sequence $\mathbf{X}$, only the shape sequences and irrelevant stimuli are retained, with each time step represented as a 14-channel input. For output sequence $\mathbf{y}$, there are four options: the fixed point(FP), the left-side target(LT), the right-side target(RT), and no fixation. Our synthetic dataset is made up of 5000 sample sequences where T is set to 70, and the PRL model is trained with default AdamW optimizer on a single 4090 GPU for about 12 minutes.

Analysis We follow previous work[8, 3, 9] to analyze the model's behavior and the responses of its internal units.

First, we examine the impact of cumulative log LR on the probability of model selection (fig. 1B). As the total log LR increases, the model tends to favor selecting the left-side target, generally exhibiting a perfect S-shaped curve.

Next, we use regression (eq. (1)) to analyze how different shapes influence the selection process. Through ridge regression, we derive the subjective weights learned by the model (fig. 1C). It reveals that the log LR assigned to each shape correlates well with its leverage on the choice.

$$P(\text{choice=left}) = \frac{1}{1+10^{-Q}}, \quad Q = \beta_0 + \sum_{m=1}^{10}\beta_m N_m \tag{1}$$

We construct special shape sequences to observe the impact of logLR fluctuations on the model's output (fig. 1D). Shape 5 and 7 increase the performance for choosing the left target, while shape 2 decreases the performance; the effect of shapes on the performance for choosing the right target is the opposite. Our findings can be summarized in 2 points: (1) Changes in logLR and its accumulation affect the model's output, and this influence is typically delayed until the next shape appears (**yellow arrow**); (2) Time, as an urgency signal, affects the model's output, with model's response increasing as time progresses, even leading to behaviors opposite to the effect of logLR (**purple arrow**).

Finally, we examined the impact of log LR accumulation on reservoirs' cell responses using another regression formula (eq. (2)) and t-test to study the significance of cell reactions, where u_t represents the urgency and is quantified as the number of epochs. The results indicate that 25.37% of the cells exhibit significant responses ($p<0.05$) to the time-step accumulation of log LR. These cells are likely to play a key role in encoding evidence accumulation.

$$\mathbf{h}_t = \beta_0 + \beta_1\sum_{n=1}^{t}\text{logLR}_n + \beta_2\left|\sum_{n=1}^{t}\text{logLR}_n\right| + \beta_3 u_t \tag{2}$$

Ablation studies To demonstrate the effectiveness of our model structure and training strategy, we conduct ablation studies on the reservoir design (parallel or non-parallel, equal parameter counts), optimizer selection (SGD or AdamW), and the structure of the output layer (time-dependent or shared). Results are shown in table 1. It can be observed that the parallel reservoirs design and the choice of the AdamW optimizer significantly enhance model performance. Results also show a sharp decline for output layer with shared weights compared to our time-dependent layers, even if a more complex structure (Linear $\rightarrow$ MLP) is applied. Since the time series in this experiment is relatively short and reservoirs do not require training, we argue that the extra cost introduced by the time-dependent output layer parameters is acceptable.

Method	SGD	Non-parallel	Shared Linear	Shared MLP	**Ours**
Accuracy	55.8%	89.6%	50.1%	48.4%	**95.7%**

Table 1: Comparative study on training strategy (the optimizer) and model architecture (parallel or not; time-dependent linear modules or a shared module).

AUTHOR CONTRIBUTIONS

R.Y. and Z.Z conducted surveys and designed the study. R.Y. developed and trained the model. Z.Z. generated the synthetic dataset and analyzed the experimental data. R.Y. and Z.Z. wrote the paper. The whole study is of equal contribution.

References

[1] M. E. Mazurek, J. D. Roitman, J. Ditterich, and M. N. Shadlen, "A role for neural integrators in perceptual decision making," *Cerebral Cortex*, vol. 13, no. 11, pp. 1257–1269, 2003.

[2] R. Ratcliff, A. Cherian, and M. Segraves, "A comparison of macaque behavior and superior colliculus neuronal activity to predictions from models of two-choice decisions," *Journal of Neurophysiology*, vol. 90, no. 3, pp. 1392–1407, 2003.

[3] Z. Zhang, H. Cheng, and T. Yang, "A recurrent neural network framework for flexible and adaptive decision making based on sequence learning," *PLOS Computational Biology*, vol. 16, no. 11, p. e1008342, 2020.

[4] W. Maass, T. Natschläger, and H. Markram, "A model for real-time computation in generic neural microcircuits," in *Proceedings of the 15th International Conference on Neural Information Processing Systems*, NIPS'02, (Cambridge, MA, USA), p. 229–236, MIT Press, 2002.

[5] H. Jaeger, "Adaptive nonlinear system identification with echo state networks," in *Proceedings of the 15th International Conference on Neural Information Processing Systems*, NIPS'02, (Cambridge, MA, USA), p. 609–616, MIT Press, 2002.

[6] K. Wong and X. Wang, "A recurrent network mechanism of time integration in perceptual decisions," *Journal of Neuroscience*, vol. 26, pp. 1314–1328, Jan. 2006.

[7] J. Vitay, "Reservoir computing," 2024. Accessed: 2024-11-29.

[8] S. Kira, T. Yang, and M. N. Shadlen, "A neural implementation of wald's sequential probability ratio test," *Neuron*, vol. 85, no. 4, pp. 861–873, 2015.

[9] T. Yang and M. N. Shadlen, "Probabilistic reasoning by neurons," *Nature*, vol. 447, no. 7148, pp. 1075–1080, 2007.