Dendritic Resonate-and-Fire Neuron for Effective and Efficient Long Sequence Modeling

Dehao Zhang[1], Malu Zhang[1,2✉], Shuai Wang[1], Jingya Wang[1], Wenjie Wei[1], Zeyu Ma[1], Guoqing Wang[1], Yang Yang[1], Haizhou Li[2,3],

[1]University of Electronic Science and Technology of China, [2]Shenzhen Loop Area Institute, [3]The Chinese University of Hong Kong (Shenzhen)

Motivation & Model Structure

Resonate-and-Fire (RF) neurons can efficiently extract frequency components from input signals and encode them into spatiotemporal spike trains, making them well-suited for long sequence modeling. However, RF neurons exhibit **limited effective memory capacity** and a **trade-off between energy efficiency and training speed** on complex temporal tasks.

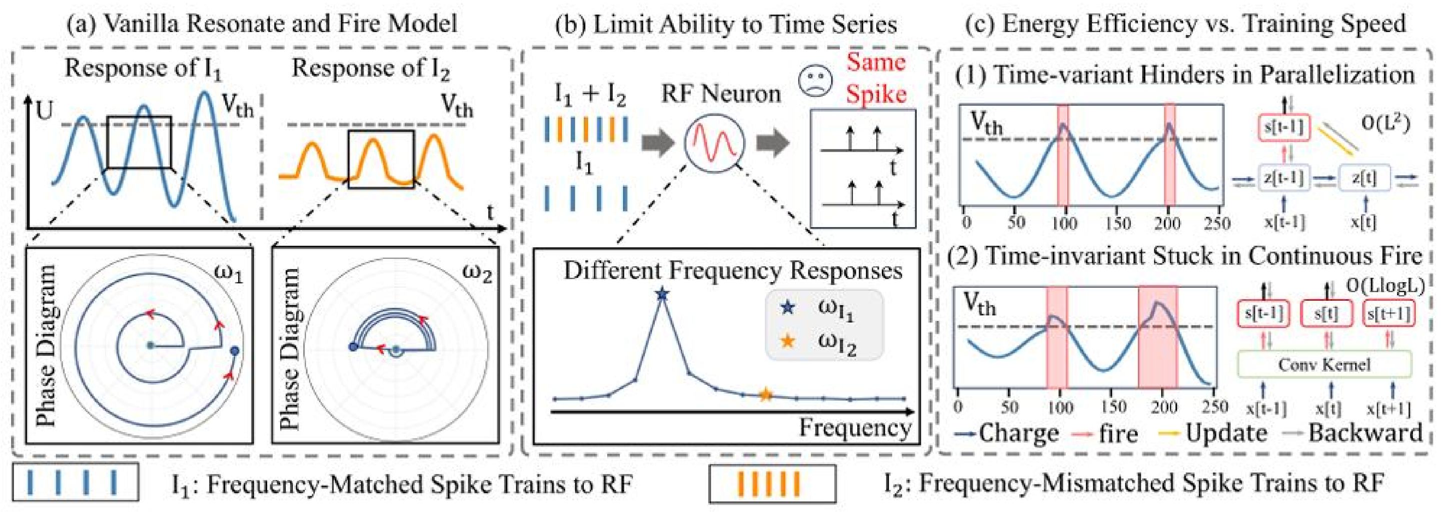

Figure 1. Analysis of RF Neuron in Long-Sequence Modeling

Inspired by **dendritic structure of biological neurons**, we propose Dendritic Resonate-and-Fire (D-RF) models. The contributions are as follows:

- We conduct an analysis of the limitations of RF neurons in long sequence modeling, highlighting their restricted memory capacity and inherent trade-off between energy efficiency and training speed.
- We propose the D-RF neuron. First, dendritic branches exploit RF dynamics to achieve specialized frequency selectivity. Second, the adaptive threshold mechanism in the soma dynamically adjusts thresholds based on history spiking activity, which balances computational cost and energy efficiency while preserving training effectiveness.
- Extensive experiments demonstrate that D-RF achieves competitive performance across various long sequence tasks. Moreover, the method produces sparser spiking activity while maintaining training efficiency.

Methods

High Performance in Complex Tasks

To better capture features across different frequency bands, we propose the D-RF neuron model. D-RF neuron comprises a **soma** and **multiple dendritic branches**. Each dendritic branch extracts state responses corresponding to specific frequency preferences in the input signal I(t).
Its dynamics are defined as follows:

$$\frac{dz_i(t)}{dt} = \{-1/\tau_i + \mathrm{i}\omega_i\} \cdot z_i(t) + \gamma_i \mathcal{I}(t), \quad (1)$$

τ_i and ω_i represent the decay factor and membrane potential oscillation coefficient associated with the i-th dendritic branch, respectively. To enable efficient inference, the Zero-Order Hold(ZOH) method is employed for discretization. The membrane potential dynamics of all dendrites are:

$$\mathcal{Z}[t] = \exp\left\{\begin{bmatrix} -\frac{1}{\tau_1}+\mathrm{i}\omega_1 & 0 & \cdots & 0 \\ 0 & -\frac{1}{\tau_2}+\mathrm{i}\omega_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & -\frac{1}{\tau_n}+\mathrm{i}\omega_n \end{bmatrix} \cdot \delta\right\} \mathcal{Z}[t-1] + \Gamma^l I[t]. \quad (2)$$

where δ denotes the discrete timestep and $Z = [z_1, z_2, \ldots, z_n]^T$ represents the states of individual dendritic branches, with $\Gamma = [\gamma_1, \gamma_2, \ldots, \gamma_n]^T$ denoting their respective time constants. To further enhance the frequency characteristics across dendritic branches, **each branch is assigned an individual importance weight**. The dynamics of soma are defined as follows:

$$H[t] = \mathcal{C}\Re\{Z(t)\}, \qquad S[t] = \Theta\left(H[t] - V_{th}[t]\right), \quad (3)$$

$\mathcal{C}$ denotes the importance weights as signed to each dendritic branch. $\Theta(\cdot)$ represents the Heaviside function.When the presynaptic membrane potential of the soma H[t] exceeds the threshold Vth,a spike S[t] is generated.

Trade-off between Energy Efficiency and Training

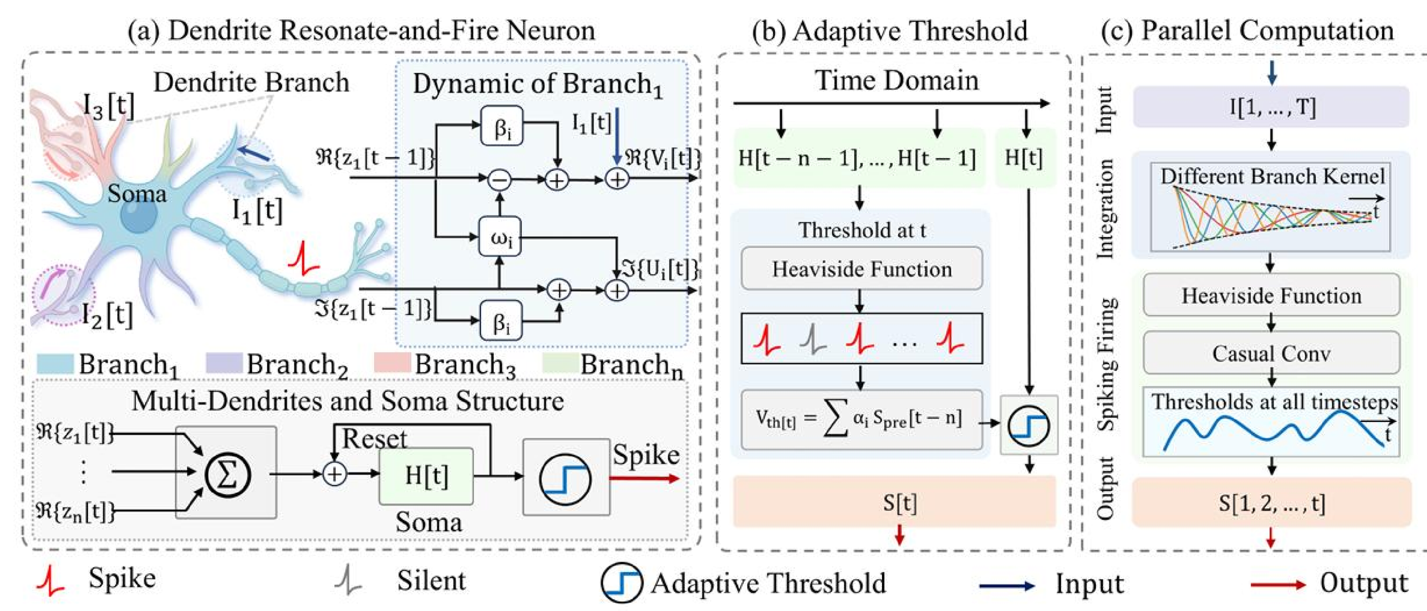

Figure 2. Structure of Dendrite Resonate-and-fire Neuron

To balance training speed and energy efficiency, we propose an adaptive thresholding strategy that dynamically adjusts thethreshold based on the spiking activity from previous timesteps. Specifically, the threshold at timestep t is defined as follows:

$$V_{\mathrm{th}}[t] = \sum_{k=1}^{n} \alpha_k \Theta(\Re\{\mathcal{Z}[t-k-1, \ldots, t-1]\} - V_{\mathrm{pre}}) + V_{\mathrm{pre}}, \quad (4)$$

where V_{pre} denotes the origin threshold (set to1),and $\alpha_k \in (0,1)$ represents the importance of preceding spikes. This process can be interpreted as a one-dimensional causal convolution with kernel size n, where the kernel is defined as A = [α1,...,αn]. The spiking process can be reformulated as:

$$\mathcal{S}[t] = \Theta\{\underbrace{C^l\Re\{\mathcal{Z}\}}_{\text{Dendritic Input}} - \underbrace{(\mathrm{Conv1d}(\Theta(C^l\Re\{\mathcal{Z}\} - V_{\mathrm{pre}}) + V_{\mathrm{pre}})}_{\text{Adaptive Threshold}}\}[t]. \quad (5)$$

Conv1d(·) present causal convolution along the temporal domain, enabling sparser spike activity while preserving the parallelizable nature of training.

Overall Architecture

We demonstrate the effectiveness of this strategy by analyzing both the forward and backward propagation processes of traning processes.

Forward Propagation: the computational complexity of the D-RF neuron is determined by multi-dendritic input and an adaptive threshold mechanism. According to Eq.(2)~Eq.(3), the parallel training can be defined as:

$$\mathcal{Z}[t] = (\mathcal{K} * \mathcal{I})[t] = \mathcal{F}^{-1}\{\mathcal{F}\{\mathcal{K}\} \cdot \mathcal{F}\{\mathcal{I}\}\}[t], \quad \mathcal{K} = \left[\delta\mathcal{D}^1, \delta\mathcal{D}^2, \ldots, \delta\mathcal{D}^n\right], \; L(LogL)$$

D characterizes the oscillatory resonators of individual dendritic branches and is defined as a diagonal matrix: D = Diag$\{-1/\tau_1 + \omega_1, -1/\tau_2 + \omega_2, \ldots, -1/\tau_n + \omega_n\}$. $F(\cdot)$ and $F^{-1}(\cdot)$ denote the forward and inverse Fourier transform operations.

Backward Propagation: In the backward propagation, the adaptive threshold removes temporal dependencies between the gradient, enabling parallelizable training. The gradient of the loss with respect to the weight w:

$$\nabla_{w^l}\mathcal{L} = \underbrace{\sum_{t=0}^{T} \frac{\partial \mathcal{L}}{\partial \mathcal{S}^l[t]} \frac{\partial \mathcal{S}^l[t]}{\partial \mathcal{Z}^l[t]} \frac{\partial \mathcal{Z}^l[t]}{\partial w^l}}_{\text{Sequential Training}} = \underbrace{\left\langle \frac{\partial \mathcal{L}}{\partial \mathcal{S}^l[t]} \frac{\partial \mathcal{S}^l[t]}{\partial \mathcal{Z}^l[t]}, (\mathcal{K} * \mathcal{S}^{l-1})[t] \right\rangle}_{\text{Parallel Training}}, \quad L(LogL)$$

the gradient during backpropagation depends only on the current timestep. Consequently, our method introduces no additional training complexity,while encouraging sparse spike activity.

Experiments & Conclusion

Experiments

We conduct experiments on multiple time-series datasets, including **SHD dataset**, **S-CIFAR10 dataset**, **LRA Benchmark.**

Table 1. Performance of Various Dataset

SHD (250 Timesteps)	LIF [74]	249.0K	Rec	✗	✗	84.00
	ALIF [72]	141.3K	Rec	✗	✗	84.40
	BRF [21]	108.8K	Rec	✗	✗	92.50
	PSN [14]	232.5K	FF	✓	✗	89.75
	TC-LIF [79]	141.8K	Rec	✗	✓	88.91
	DH-LIF [80]	0.5M	Rec	✗	✓	91.34
	PMSN [7]	199.3K	FF	✓	✓	95.10
	Ours	155.1K	FF	✓	✓	**96.20**
S-CIFAR10 (1024 Timesteps)	LIF [74]	0.18M	FF	✗	✗	45.07
	PSN [14]	6.47M	FF	✓	✓	55.24
	SPSN [14]	0.18M	FF	✓	✓	70.23
	PMSN [7]	0.21M	FF	✓	✓	82.14
	Ours	0.21M	FF	✓	✓	**84.30**

Table 2. Comparison of Model on LRA.

Model (Input length)	SNN	ListOps (2,048)	Text (4,096)	Retrieval (4,000)	Image (1,024)	Pathfinder (1,024)	Avg.
Random	-	10.00	50.00	50.00	10.00	50.00	34.00
Transformer [62]	✗	36.37	64.27	57.46	42.44	71.40	54.39
S4 (Bidirectional) [19]	✗	59.60	86.82	90.90	88.65	94.20	84.03
Binary S4D [57]	✓	54.80	82.50	85.03	82.00	79.79	77.39
→ + GSU & GeLU	✓	59.60	86.50	90.22	85.00	91.30	82.52
SpikingSSMs [51]	✓	60.23	80.41	88.77	88.21	93.51	82.23
Spiking LMU [33]	✓	37.30	65.80	79.76	55.65	72.68	62.23
ELM Neuron [56]	✓	44.55	75.40	84.93	49.62	71.15	69.25
SD-TCM [24]	✓	59.20	86.33	89.88	84.77	91.76	82.39
Ours	✓	**60.02**	**86.52**	**90.02**	**85.32**	**92.36**	**82.88**

Compared with other SOTA method, our method maintains competitive accuracy on various dataset.

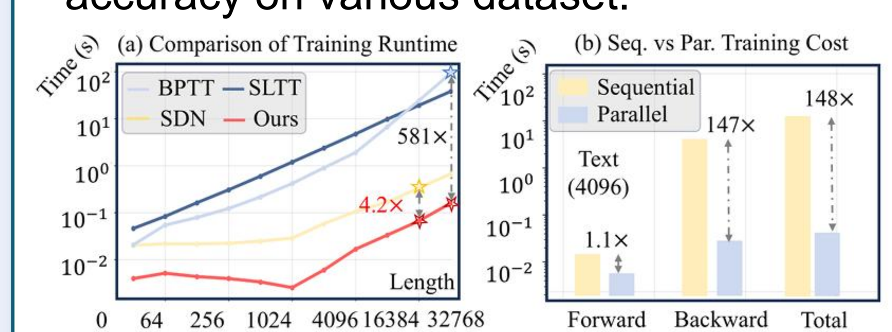

Figure 3. (a) Comparison of Training Runtime. (b) Sequentialvs. Paral. Training Cost

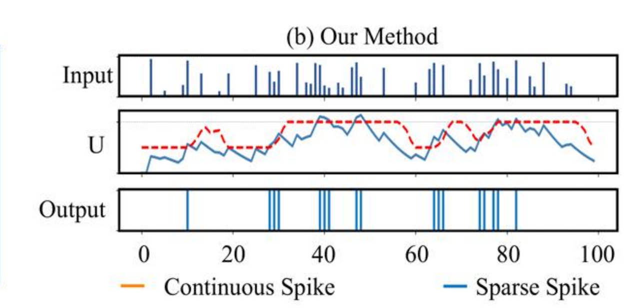

Figure 4. Spiknig Behavior of our D-RF Model

Compared with other SOTA method, our method substantially ensuring sparse spikes without compromising computational efficiency during training

Conclusion

Inspired by the dendritic structure of biological neurons, this study proposes the D-RF model to further enhance the performance of SNNs on time-series signals.The multi-dendritic structure consists of branches with distinct decay factors, enabling the neuron to extract multi-frequency information from the input signal. The soma with an adaptive threshold that ensures sparse spiking while enabling parallelizable computation. These results highlight the strong potential of D-RF method to enable effective and efficient long sequence modeling on edge-computing platforms.