Bridging rate and temporal coding in bio-realistic spiking neural networks with a noise-driven gradient-based learning rule

Vikrant Jaltare[1,2,*], Johannes Leugering[1,2], Ali Safa[2,3], Samira Sebt[1,2], Leif Gibb[2] and Gert Cauwenberghs[1,2]

[1] Department of Bioengineering, University of California San Diego, La Jolla, CA, USA, [2] Institute of Neural Computation, University of California San Diego, La Jolla, CA, USA

[3] IMEC and ESAT, KU Leuven, Belgium

UC San Diego
JACOBS SCHOOL OF ENGINEERING
Shu Chien-Gene Lay Department of Bioengineering

INTEGRATED SYSTEMS NEUROENGINEERING

Abstract

- ***Aim:*** Bridging efficiency/performance gap between rate and spike-timing based models.
- ***Method:*** Unified gradient-based learning rule for two-compartment LIF neuron with noisy current input to the membrane.
- ***Takeaway:*** Continuum between rate and spike time codes emerges as the noise magnitude is varied producing rate code in presence of higher noise and temporal code in presence of lower noise

Introduction

- Developing learning algorithms for SNNs remains an open challenge.
- ***Rate codes:***
  - *Pros:* Error-tolerance and correspondence with Artificial Deep Neural Networks. Training with surrogate gradients.
  - *Cons:* May learn energy inefficient codes.
- ***Timing codes***:
  - *Pros:* Efficient codes that can capture large dynamic range of inputs. Biologically plausible.
  - *Cons:* Limited gradient-based approaches for learning are available. The algorithms often require complex PSP kernel models.
- We aim to bridge the robustness of rate codes and efficiency of timing codes through a probabilistic model of synaptic and neural dynamics embedded in the learning rule.

Methods

- ***Objective:*** Find the optimal weight w^* such that:

$$\mathbf{w}^* = \mathrm{argmax}_{\mathbf{w}}\, \mathcal{L}(\mathbf{q}(t), \mathbf{p}(t))$$

Succinctly, $\mathcal{L}(q(t), p(t)) = \mathcal{L}(t)$

$$\nabla_w \mathcal{L}(t) = \frac{\partial \mathcal{L}}{\partial p} \cdot \frac{\partial p}{\partial V_m} \cdot \frac{\partial V_m}{\partial w} \quad \textbf{Gradient}$$

$$\frac{\partial \mathcal{L}(t)}{\partial p} = \frac{q(t) - p(t)}{p(t)\,(1 - p(t))} \quad \textbf{Error Term}$$

$$\frac{\partial p}{\partial V_m} = k \cdot e^{-\frac{(u - V_m(t))^2}{2\sigma_n^2}} \quad \textbf{Gaussian}$$

$$\frac{\partial V_m}{\partial w} = \varepsilon(t) * s_{in}(t) \quad \textbf{PSP Trace}$$

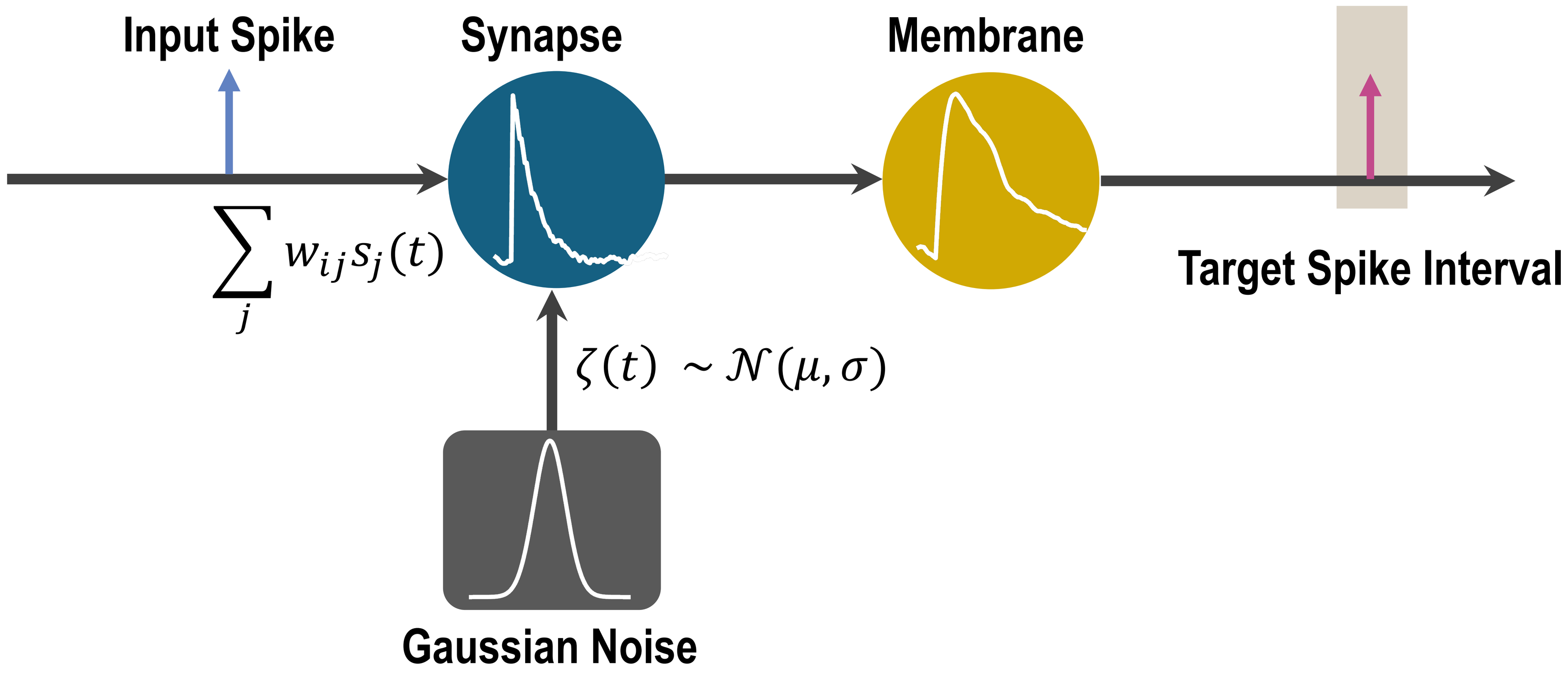
(Assuming low firing rate)

- $\varepsilon(t)$ is the membrane post-synaptic potential (PSP) kernel

$$\nabla_w \mathcal{L} = \frac{1}{T} \sum_{t=1}^{T} \nabla_w \mathcal{L}$$

vjaltare@ucsd.edu

Model Setup



Model Equations

Synapse

$$\tau_r \frac{dr(t)}{dt} = -r(t) + \sum_j w_{ij} s_j(t) + \zeta(t)$$

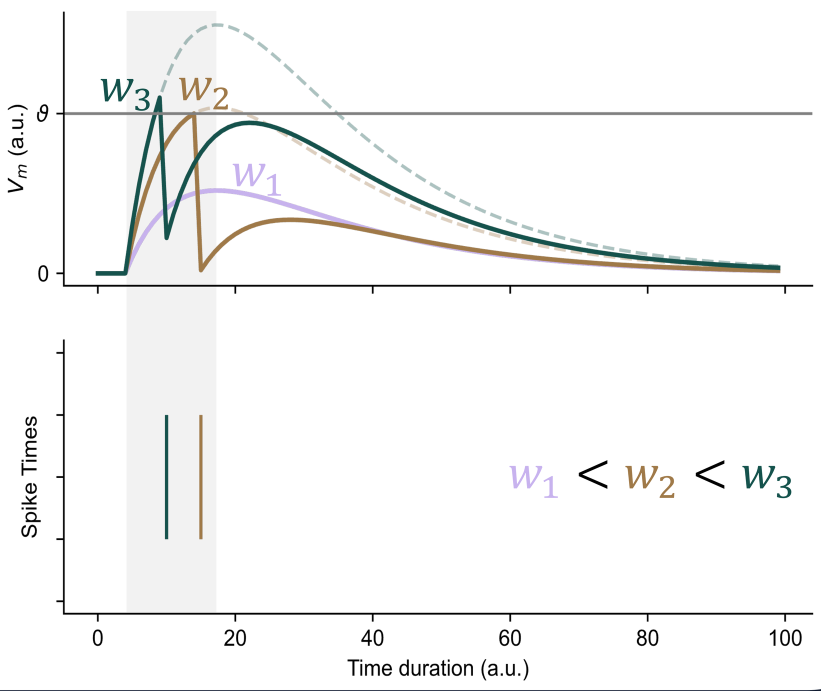
Membrane

$$\tau_m \frac{dV_m(t)}{dt} = -V_m(t) + r(t)$$

Spike generation

$$s_o(t) = \begin{cases} 1, & V_m \geq \vartheta \\ 0, & V_m < \vartheta \end{cases}$$

Scaling the weight shifts the spike time

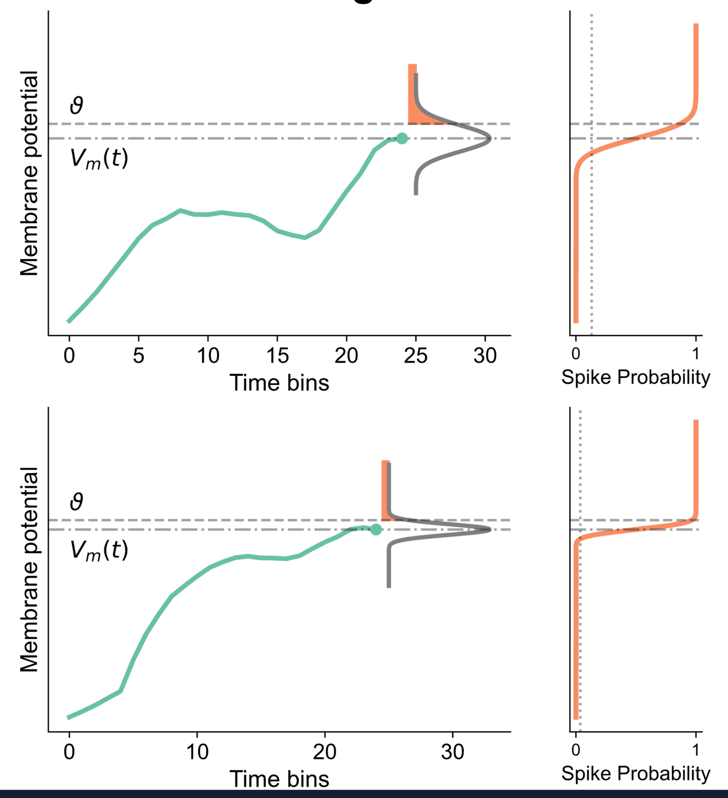


Spike probability and Model Evaluation

Gaussian Tail Probability

$$p_t(s_o(t)|V_m(t)) = \int_{u=\vartheta}^{\infty} \frac{1}{\sqrt{2\pi}\,\sigma_n} e^{-\frac{(u - V_m(t))^2}{2\sigma_n^2}} du$$

Modulating Noise Level



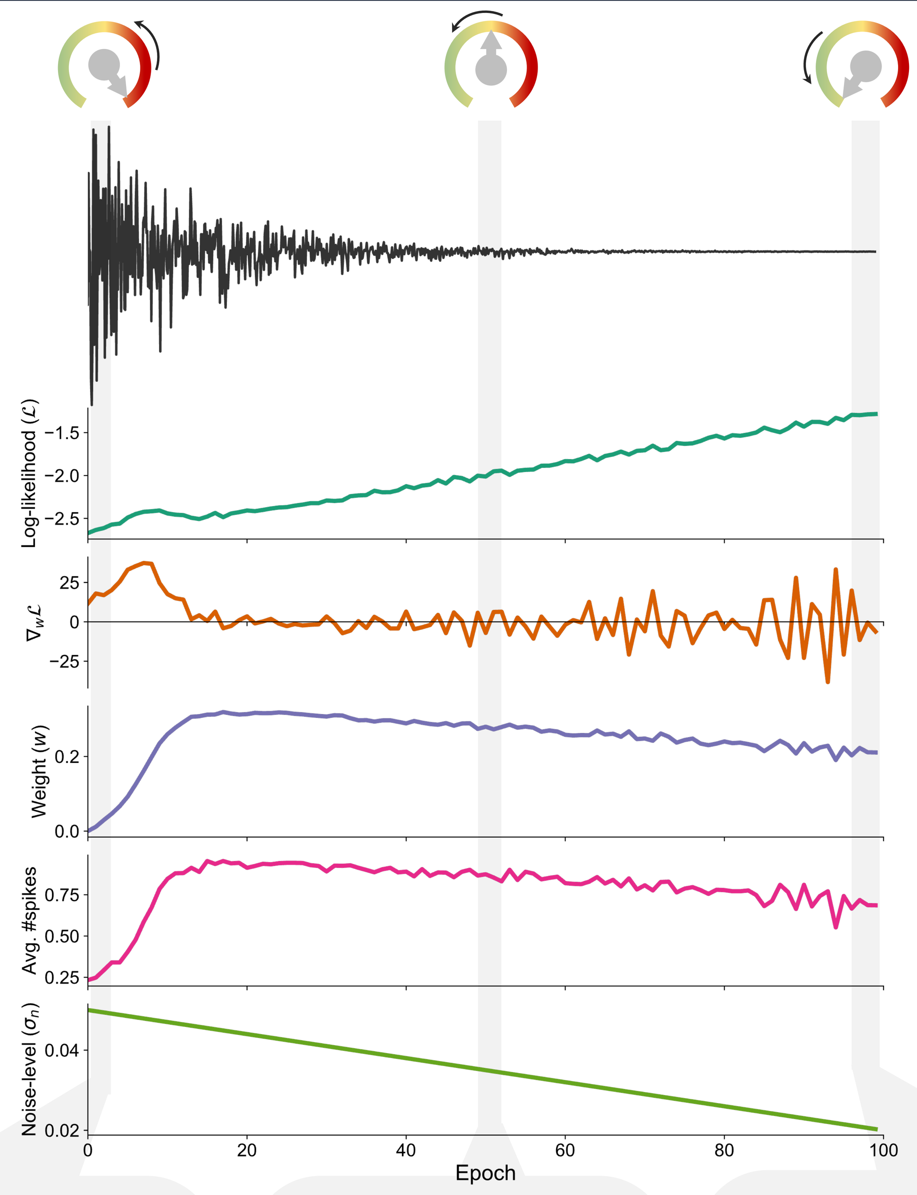
Likelihood of an output spike train

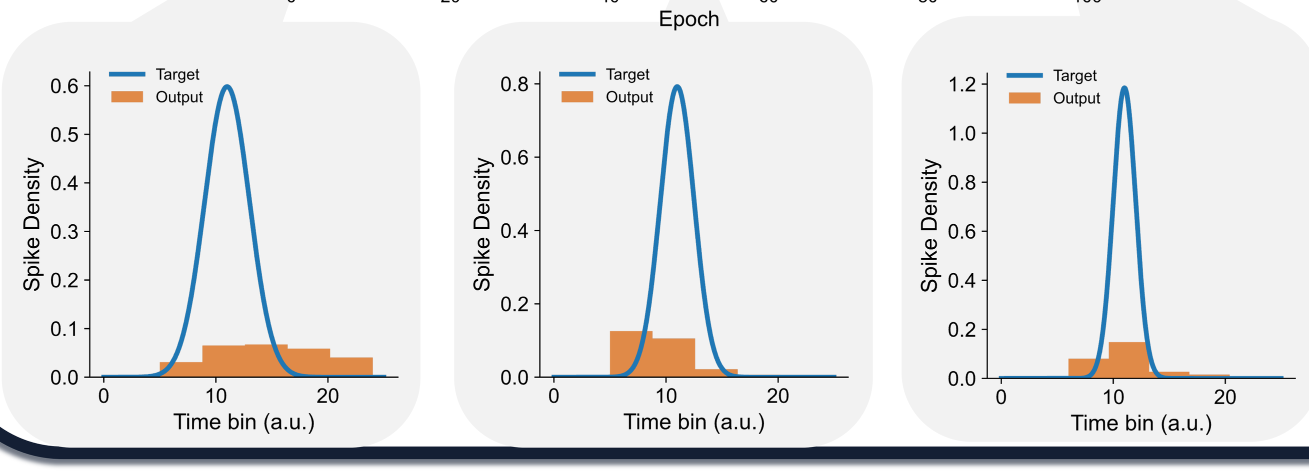
$$\mathcal{L} = \frac{1}{T} \sum_{t=0}^{T-1} q(t) \log p(t) + (1 - q(t)) \log(1 - p(t))$$

$q(t), p(t)$ are desired and observed instantaneous firing probabilities respectively

The objective is to **maximize** the likelihood of the spike train $\sum_t \mathcal{L}(q(t), p(t)) = \sum_t \mathcal{L}(t)$

Training Procedure





Noise Scaling

$$\sigma_n(t) = \sigma_n^0 + \frac{\sigma_n^T - \sigma_n^0}{T} \cdot t$$

Target distribution scaling

$$\sigma_t(t) = \sigma_t^0 + \frac{\sigma_t^T - \sigma_t^0}{T} \cdot t$$

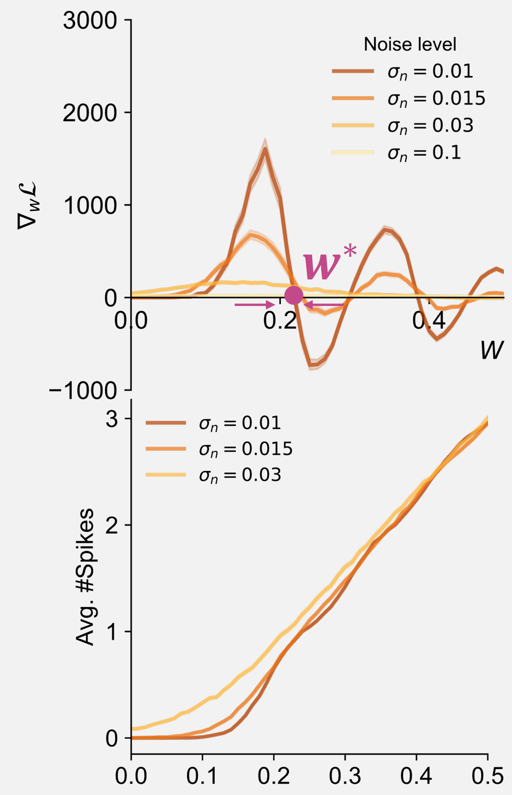
Gradient Ascent

$$w \leftarrow w + \eta \nabla_w \mathcal{L}$$

t: Epoch of training
T: Total duration of training

x

Gradient Landscape



- ***Fixed points:*** The gradient landscape has stable fixed points at the optimal weight (w^*).
- ***Zero-weights:*** For non-zero gradients near zero weights, the noise should be high, whereas for stable fixed points near the optimum, the noise should be low.

This forms the rationale for "cooling" the model for learning.

Future Directions

- ***Scaling up:*** Extending the framework to multilayer networks.
- ***Autodiff:*** Integrating stochastic gradient descent with automatic differentiation tools like PyTorch.
- ***Efficient Coding:*** Studying efficiency of codes learned through this method in the hidden layers.

Conclusions

- ***Noise to the rescue:*** Learning rule leveraging noise in the dynamics of LIF neurons can help learning sparse inputs or zero weight initialization.
- ***Continuum:*** Stochasticity not only smoothens the spike Heaviside function for gradient computation but also provides a "tunable knob" to go from a rate-based to a timing-based learning rule.

Acknowledgements

We acknowledge funding from the Office of Naval Research (ONR) for this project. We also thank Justin Kinney, Omowuyi Olajide, Kenneth Yoshimoto, Steve Deiss and members of the Integrated Systems Neuroengineering Lab at UC San Diego for insightful discussions and help.

References

[1] Jang, H., et al. *IEEE Signal Process. Mag.* **36**, 64–77 (2019).
[2] Gygax, J. & Zenke, F. *arXiv [cs.NE]* (2024).
[3] Eshraghian, J. K. *et al. IEEE* **111**, 1016–1054 (2023).
[4] Kaiser, J., et al. *Front. Neurosci.* **14**, 424 (2020).
[5] Neftci, E. O., et al., F. *IEEE Signal Process. Mag.* **36**, 51–63 (2019).
[6] Zenke, F. & Ganguli, S. *Neural Comput.* **30**, 1514–1541 (2018).
[7] Zenke, F. & Vogels, T. P. *Neural Comput.* **33**, 899–925 (2021).
[8] Zhang, M. *et al. IEEE Trans Neural Netw Learn Syst* **33**, 1947–1958 (2022).
[9] Wang, H.-P et al., *J. Neurosci.* **39**, 7674–7688 (2019).