Rapid spatiotemporal coding in trained multi-layer and recurrent spiking neural networks

Friedemann Zenke & Tim P. Vogels

UNIVERSITY OF OXFORD
Centre for Neural Circuits and Behaviour
Department of Physiology Anatomy and Genetics

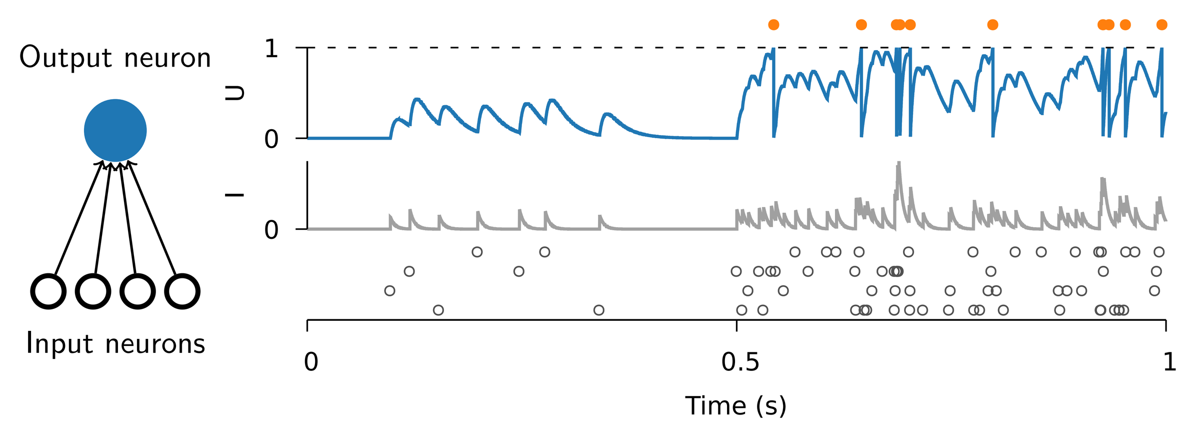
fzenke.net: poster, code, & more...

Aims & Approach

Aim: Build functional spiking neural network models that
1. Use spike timing
2. Compute rapidly
3. Exhibit sparse spiking

Approach: We train spiking neural networks (SNNs) using surrogate gradients to solve a set of classification problems requiring rapid spatiotemporal processing.

Understanding spiking neural networks as RNNs



Update equations:

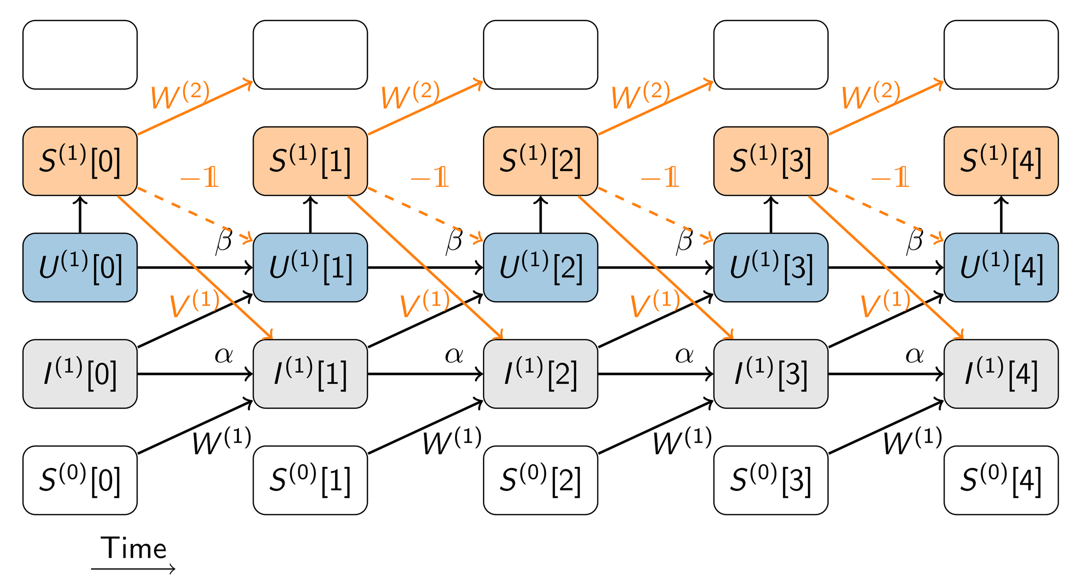
$$I_i^{(1)}[n+1] = \underbrace{\alpha I_i^{(1)}[n]}_{\text{exp. current decay}} + \underbrace{\sum_j W_{ij} S_j^{(0)}[n]}_{\text{feed-forward input}} + \underbrace{\sum_k V_{ik} S_k^{(1)}[n]}_{\text{recurrent input}}$$

$$U_i^{(1)}[n+1] = \underbrace{\beta U_i^{(1)}[n]}_{\text{leak}} + \underbrace{I_i^{(1)}[n]}_{\text{input}} - \underbrace{S_i[n]}_{\text{reset}}$$

$$S_i^{(1)}[n] = \Theta\left(U_i^{(1)}[n] - 1\right)$$

where Θ is the Heaviside function.

Computational graph of a SNN with current-based neurons:



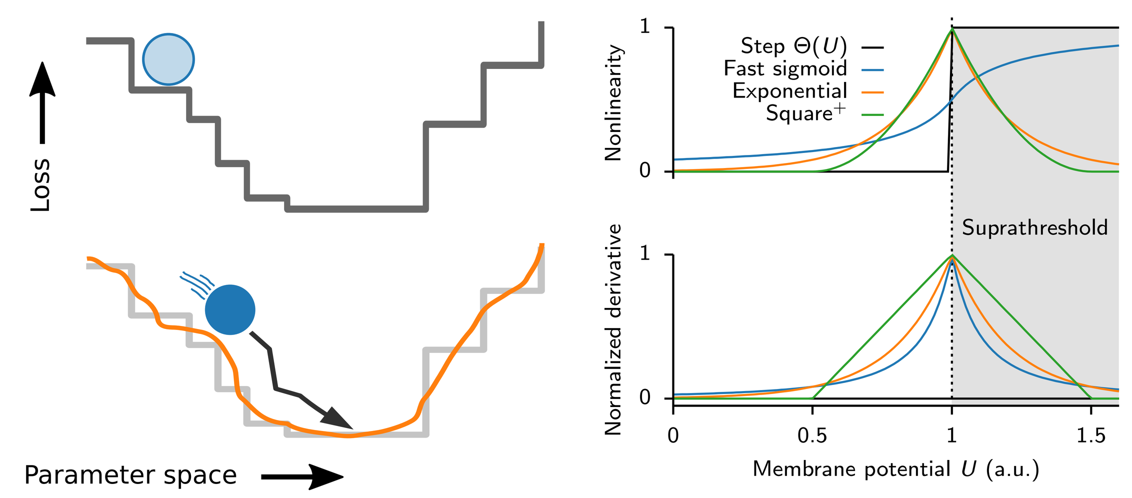
Take-home: A SNN is formally equivalent to a recurrent neural network (RNN) with specific recurrent connectivity (e.g. leak terms). Thus gradients can be computed, for instance, using backpropagation through time (BPTT). See Neftci et al. [1] for details.

Learning with surrogate gradients

Problem: Gradients of SNNs include derivatives of spikes

$$\frac{\mathrm{d}S_i^{(1)}}{\mathrm{d}U} = \Theta'(U) = \begin{cases} \infty & U = 0 \\ 0 & \text{otherwise} \end{cases}$$

Solution: We introduce a surrogate gradient which approximates the original problem [1].

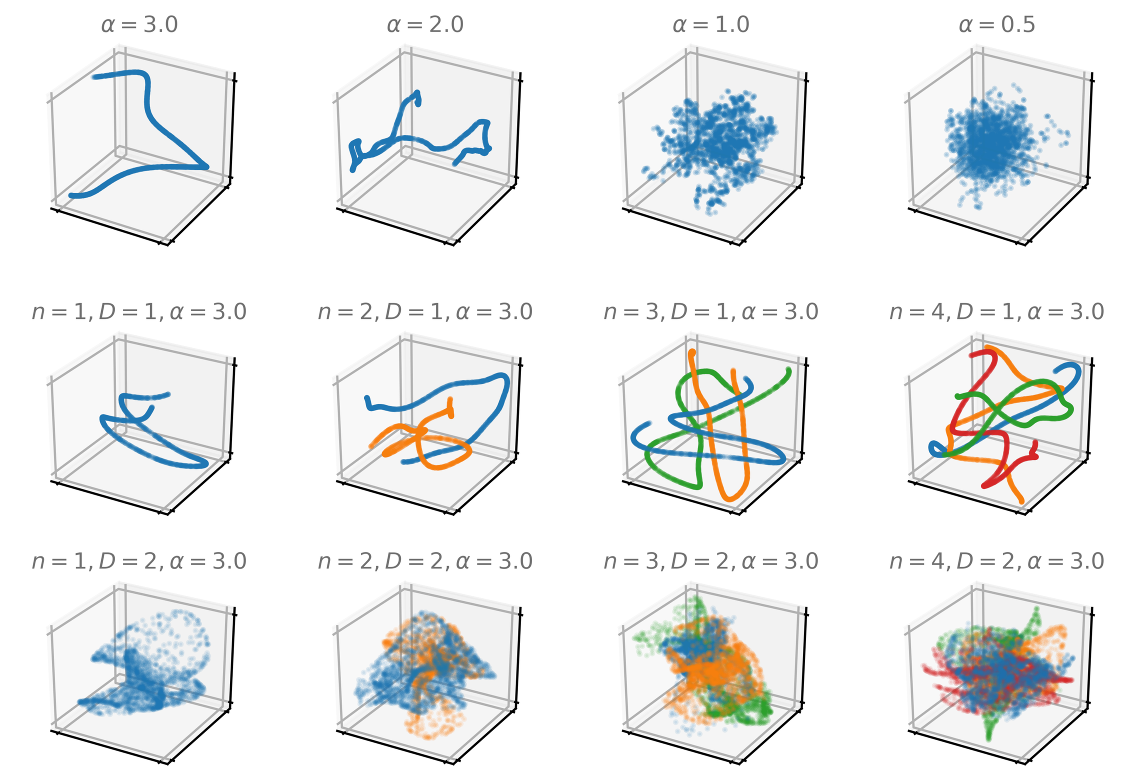


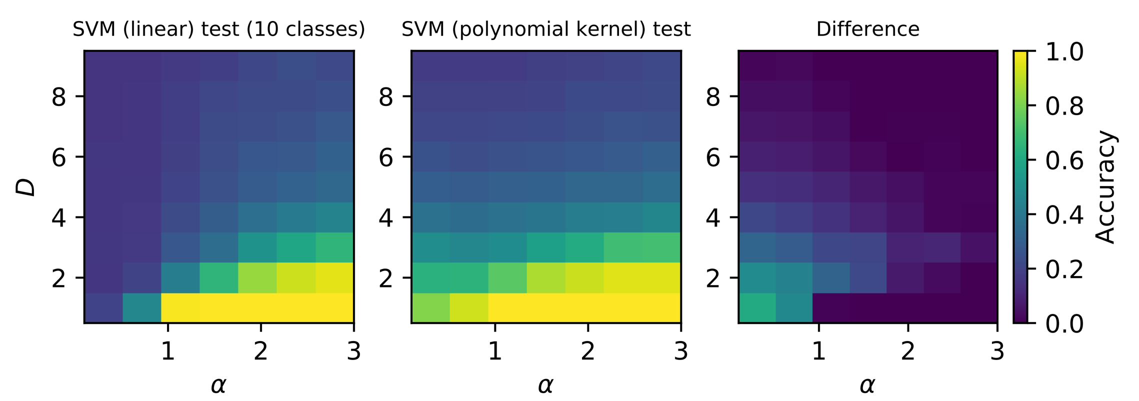
Unlike Huh and Sejnowski [2], spikes in the forward pass are not affected and remain binary. **But also see** [1, 3–5] which all use variations of the above idea.

Synthetic data: Random manifolds

Problem: Simulating SNNs is computationally demanding and training them on large datasets remains challenging. Thus we use synthetic datasets for exploration.

Solution: We generate synthetic data by sampling from smooth random manifolds. These datasets can be kept small, yet difficult (e.g., non-linearly separable). Importantly, this allows to study **generalization**.

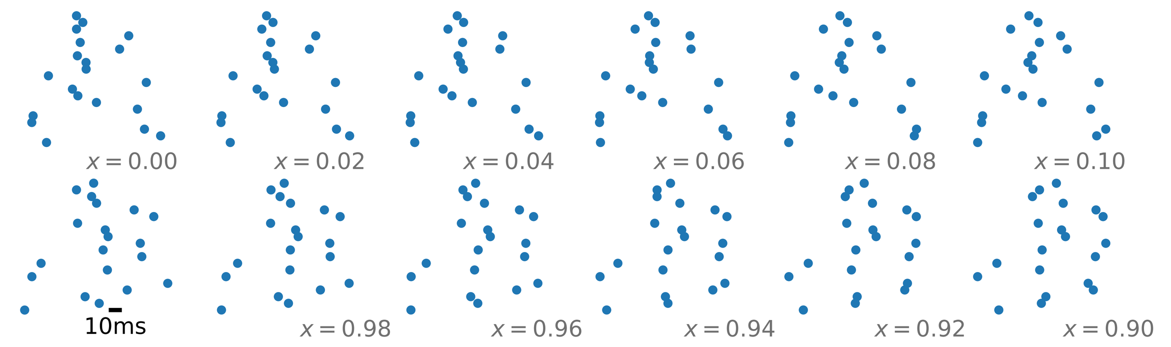




Smoothness $1/\alpha$, manifold dimension D, and number of classes n are freely adjustable which allows us to generate **compact datasets with varying degrees of difficulty**.

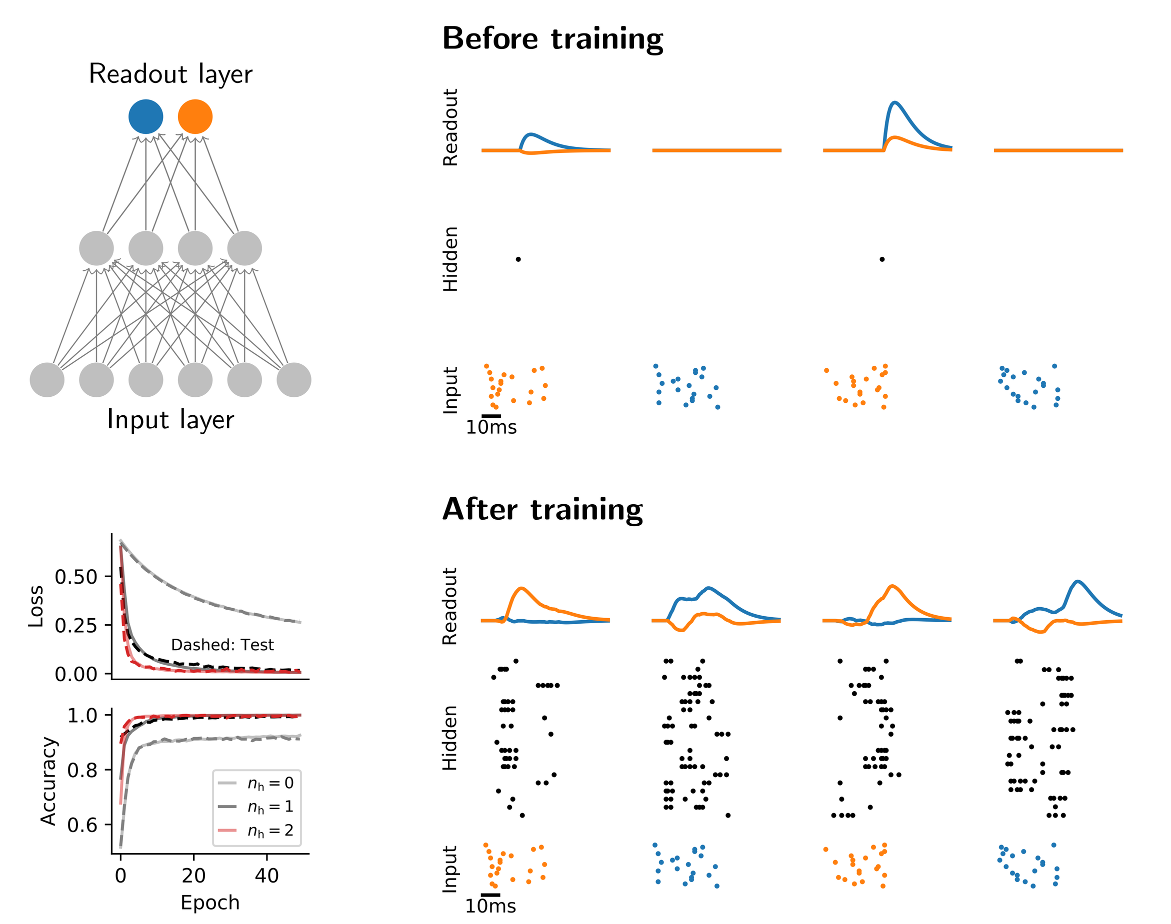
Random manifolds → spikes

To study **temporal coding**, we interpret the embedding space coordinates from a smooth random manifold as firing times. Different manifolds correspond to different classes. In the example below we use a mapping in which each neuron fires precisely one spike. Here we plot **spike rasters** continually sampled along the intrinsic coordinate x $(D = 1)$.



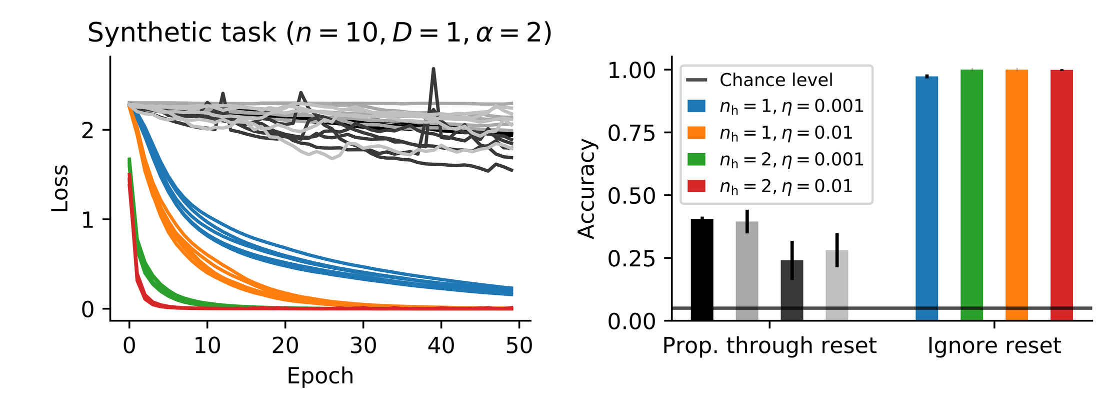
Supervised learning setup

Output and supervised loss: As readout layer we use leaky integrators which do not spike. For training a cross-entropy loss function is defined on this readout by taking the maximum along time (see also [6]).



Input paradigms: We used three distinct input paradigms. (i) Synthetic data with structure imposed from smooth random manifolds. (ii) Standard vision benchmarks [7] converted to a first spike latency code. (iii) Spiking data from a silicon cochlea dataset [8].

Reset — Back-propagate or not?

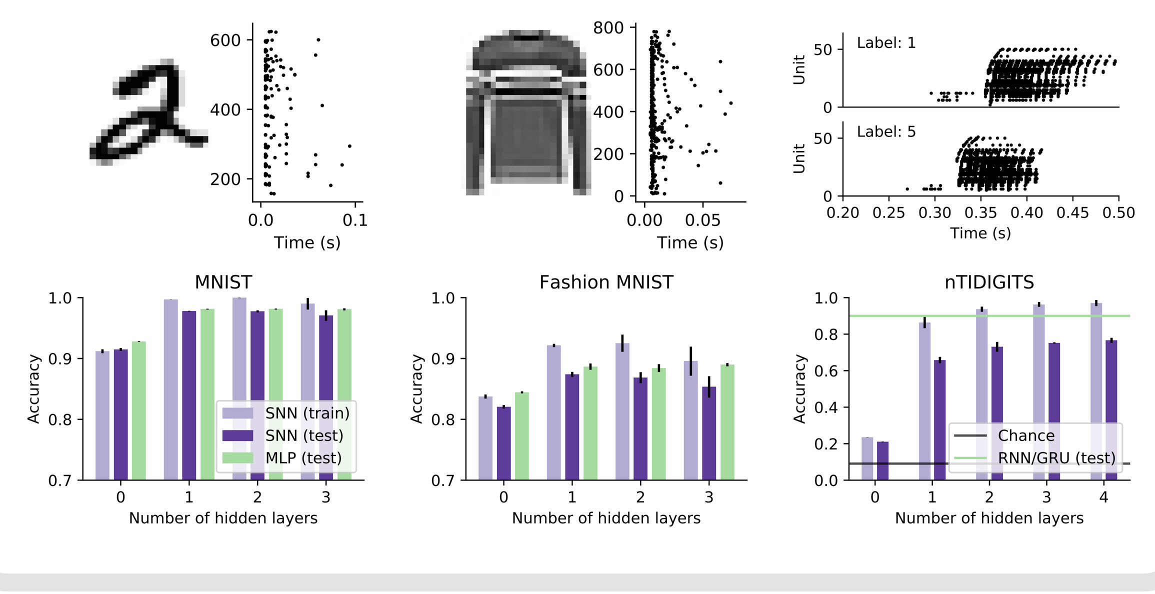


With the number of hidden layers n_h and the η learnig rate.

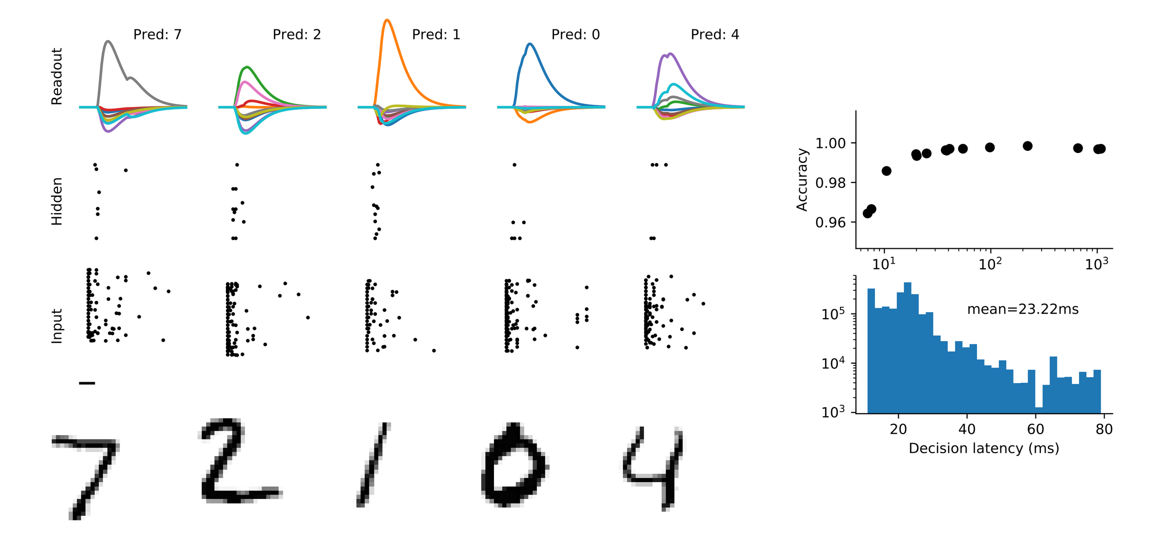
Take-home: Surrogate gradient descent is more efficient if the voltage reset is ignored when propagating gradients.

Performance on benchmarks

MNIST and Fashion MNIST [7]: Input: Each input neuron either remains silent or spikes once. Output: Softmax over maximum value of eacha readout neuron. **nTIDIGITS:** Input: Spoken digits processed by silicon chochlea [8]. Multiple spikes are possible. Output: Softmax readout group at the end of each trial.



Rapid computation & sparse firing



Summary & Outlook

- Surrogate gradients can be used to train SNNs to perform rapid spatiotemporal processing
- SNNs accurately solve classification problems with a small number of spikes
- This opens the door to use SNNs as versatile tools (cf. [9])

[1] Emre O. Neftci, Hesham Mostafa, and Friedemann Zenke. *arXiv:1901.09948 [cs, q-bio]*, January 2019.

[2] Dongsung Huh and Terrence J Sejnowski. *NeurIPS*, pages 1440–1450, 2018.

[3] Steven K. Esser, Paul A. Merolla, John V. Arthur, Andrew S. Cassidy, Rathinakumar Appuswamy, Alexander Andreopoulos, David J. Berg, Jeffrey L. McKinstry, Timothy Melano, Davis R. Barch, Carmelo di Nolfo, Pallab Datta, Arnon Amir, Brian Taba, Myron D. Flickner, and Dharmendra S. Modha. *Proc Natl Acad Sci U S A*, 113(41):11441–11446, October 2016.

[4] Friedemann Zenke and Surya Ganguli. *Neural Computation*, 30(6):1514–1541, April 2018.

[5] Guillaume Bellec, Darjan Salaj, Anand Subramoney, Robert Legenstein, and Wolfgang Maass. *NeurIPS*, pages 795–805, 2018.

[6] Robert Gütig and Haim Sompolinsky. *Nat Neurosci*, 9(3):420–428, March 2006.

[7] Han Xiao, Kashif Rasul, and Roland Vollgraf. *arXiv:1708.07747 [cs, stat]*, August 2017.

[8] Jithendar Anumula, Daniel Neil, Tobi Delbruck, and Shih-Chii Liu. *Front. Neurosci.*, 12, 2018.

[9] Omri Barak. *Current Opinion in Neurobiology*, 46:1–6, October 2017.