BEYOND BIOLOGICALLY PLAUSIBLE SPIKING NETWORKS FOR NEUROMORPHIC COMPUTING

Anand Subramoney[1], Khaleelulla Khan Nazeer[2], Mark Schöne[2], Christian Mayr[2], David Kappel[1]

[1]Institute for Neural Computation, Ruhr University Bochum, Germany. [2]Faculty of Electrical and Computer Engineering, TU Dresden, Dresden, Germany.

Spiking neural networks

Spiking neurons and spiking neural networks (SNNs) developed as models of biological neurons. They are currently the canonical neuron architecture on neuromorphic devices. But, the focus of neuromorphic devices is shifting towards machine learning applications.

Spiking neural networks?

PROs: Common spiking models are theoretically and computationally simple or tractable. The dynamics of these networks is **well understood, or well studied**.

CONs: SNNs **lag significantly behind state-of-the-art deep learning models in task-performance**. In addition, the constraints for computing in the brain are different from that of neuromorphic devices. Therefore, SNNs may not be the ideal models for specific neuromorphic devices.

Building models for neuromorphic hardware from first principles can lead to better models – both in terms of task-performance and energy-efficiency.

BEYOND Spiking neural networks

To demonstrate this paradigm, we build a model by extracting the principles of **event-based** computing and **activity-sparsity** from SNNs.

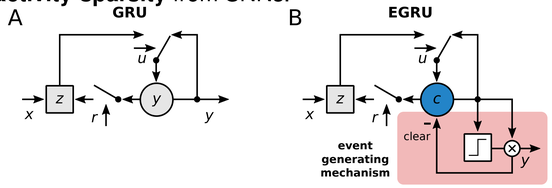

Figure 1: **A.** The standard GRU **B.** Our model extends GRU.

This model is derived from the **Gated Recurrent Unit** (GRU) by adding an event-generation mechanism.

Note that the complexity of biological neurons can be subsumed by such models.

Event-based GRU (EGRU)

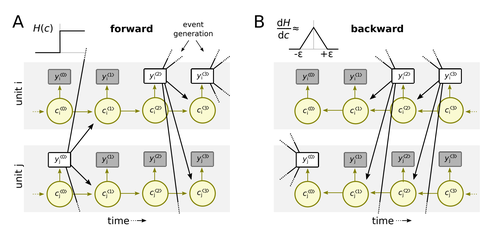

Figure 2: Illustrates the discrete time state dynamics for two EGRU units (i and j). **A.** Forward dynamics. **B.** Activity-sparse backward dynamics. Insets show threshold function H(c) and pseudo derivative thereof.

The dynamics of a layer of GRU units is given by

$$\mathbf{u}^{\langle t\rangle} = \sigma\left(\mathbf{W}_u \hat{\mathbf{x}}^{\langle t\rangle} + \mathbf{b}_u\right) \;, \quad \mathbf{r}^{\langle t\rangle} = \sigma\left(\mathbf{W}_r \hat{\mathbf{x}}^{\langle t\rangle} + \mathbf{b}_r\right) \;,$$

$$\mathbf{z}^{\langle t\rangle} = g\left(\mathbf{W}_z \left[\mathbf{x}^{\langle t\rangle},\;\; \mathbf{r}^{\langle t\rangle} \circ \mathbf{y}^{\langle t-1\rangle}\right] + \mathbf{b}_z\right) \;,$$

$$\mathbf{y}^{\langle t\rangle} = \mathbf{u}^{\langle t\rangle} \circ \mathbf{z}^{\langle t\rangle} + (1 - \mathbf{u}^{\langle t\rangle}) \circ \mathbf{y}^{\langle t-1\rangle} \;.$$

where $\hat{\mathbf{x}}^{\langle t\rangle} = \left[\mathbf{x}^{\langle t\rangle},\; \mathbf{y}^{\langle t-1\rangle}\right]$, $\mathbf{W}_{u/r/z}$, $\mathbf{b}_{u/r/z}$ denote network weights and biases, $\circ$ denotes the element-wise product, and $\sigma(\cdot)$ is the vectorized sigmoid function. The notation $\left[\mathbf{x}^{\langle t\rangle}, \mathbf{y}^{\langle t-1\rangle}\right]$ denotes vector concatenation. The function $g(\cdot)$ is an element-wise nonlinearity (typically the hyperbolic tangent function).

The **GRU is augmented with an event-generator** consisting of a rectifier and a clearing mechanism.

$$y_i^{\langle t\rangle} \;=\; c_i^{\langle t\rangle}\, H\left(c_i^{\langle t\rangle} - \vartheta_i\right)$$

$$\text{with} \quad c_i^{\langle t\rangle} \;=\; u_i^{\langle t\rangle} z_i^{\langle t\rangle} + (1 - u_i^{\langle t\rangle}) c_i^{\langle t-1\rangle} - y_i^{\langle t-1\rangle}$$

where $H(\cdot)$ is the Heaviside step function and $\vartheta_i > 0$ is a threshold parameter.

Sparse backward pass

Since $H(c)$ is not differentiable at the threshold ϑ_i, we define a pseudo-derivative at that point for calculating the backpropagated gradients as shown in the inset in Fig. 2B. **Beyond the support of the pseudo-derivative, gradients are not backpropagated, which makes the backward pass sparse.**

Results: DVS gesture recognition

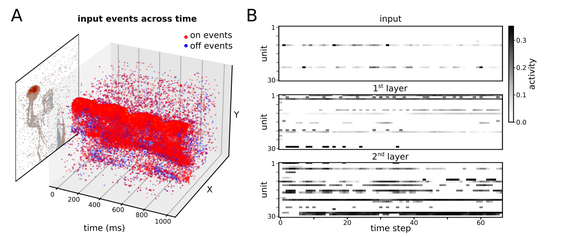

Figure 3: DVS Gesture classification. **A.** Illustration of data for an example class (right hand wave). On (red) and off (blue) events are shown over time. **B.** Sparse activity of input and EGRU units (random subset).

reference	architecture (# units)	parameters	effective MAC	accuracy	activity sparsity	backward sparsity
He et al.	LSTM (512)	7.35M	7.34M	86.81%	-	-
Innocenti et al.	AlexNet+LSTM+DA	9.99M	638.25M	97.73%	-	-
ours	GRU (1024)	15.75M	15.73M	88.07%	0%	-
ours	**EGRU** (512)	5.51M	4.19M	88.02%	83.79%	53.55%
ours	**EGRU** (1024)	15.75M	10.54M	90.22%	82.53%	56.63%
ours	**EGRU**+DA (1024)	15.75M	10.77M	97.13%	78.77%	58.20%

Table 1: Model performance over 5 runs for the DVS Gesture recognition task.

Results: Language modelling on PTB

reference	architecture (# units)	parameters	effective MAC *	validation	test	activity sparsity
Gal et al.	Variational LSTM	24M	-	77.3	75.0	-
Merity et al.	AWD-LSTM	24M	24M	60.0	57.3	-
ours	GRU (1350)	24M	24M	71.2	68.8	-
ours	**EGRU** (1350)	24M	4.7M	67.4	64.5	88.0%
ours	**EGRU** (2000)	45M	6.6M	66.5	63.7	90.4%
ours	**EGRU** (2700)	77M	8.1M	66.4	63.5	93.2%

Table 2: Model comparison on PennTreebank.

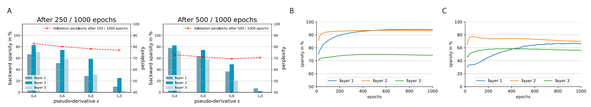

Figure 4: Backward sparsity for EGRU with 2000 hidden units on the Penn Treebank language modeling task with varying pseudo-derivative support ϵ.

EGRU in continuous time

EGRU can also be written in continuous time, since the GRU itself has a natural continuous-time formulation. To handle **discrete inputs** as events, we formulate our model as a hybrid discrete/continuous system. We introduce unit activations $\mathbf{a}_u(t)$, $\mathbf{a}_r(t)$ and $\mathbf{a}_z(t)$, with

$$\mathbf{u}(t) \;=\; \sigma\left(\mathbf{a}_u(t)\right) \;, \quad \mathbf{r}(t) \;=\; \sigma\left(\mathbf{a}_r(t)\right) \;, \quad \mathbf{z}(t) \;=\; g\left(\mathbf{a}_z(t)\right) \;.$$

The model follows the continuous dynamics of the ODE between events:

$$f_c \;\equiv\; \tau_m\, \dot{\mathbf{c}}(t) + \mathbf{u}(t) \circ \left(\mathbf{c}(t) - \mathbf{z}(t)\right) \;,$$

$$f_{a_\mathsf{X}} \;\equiv\; \tau_s\, \dot{\mathbf{a}}_\mathsf{X}(t) + \mathbf{a}_\mathsf{X}(t) \;+ \mathbf{b}_\mathsf{X} \;=\; 0 \;, \quad \mathsf{X} \in \{u,\; r,\; z\} \;,$$

where τ_s and τ_m are time constants and the initial conditions are $\mathbf{a}_\mathsf{X}(0) = \mathbf{c}(0) = \mathbf{0}$.

Discrete state transitions occur at event times s_k triggered by external events when inputs $x_i(s_k) \neq 0$, or internal events when any cell state reach a threshold ϑ.

Event-based gradient updates

The adjoint sensitivity method can be used to calculate gradients of a loss functional with adjoint variables. The weight updates use only quantities calculated at events, **making them event-based** (like in event-prop).

Summary

- We look beyond biologically-plausible spiking neural networks for neuromorphic hardware.
- We propose the EGRU "spiking" network that is based on the popular GRU architecture, rather than on biological details.
- We show state-of-the-art results for DVS gesture recognition and language modelling tasks.
- The EGRU retains the advantages of activity-sparsity and event-based computing of SNNs for both inference and training, with improved task-performance.

A. Subramoney, K. K. Nazeer, M. Schöne, C. Mayr, and D. Kappel, 'EGRU: Event-based GRU for activity-sparse inference and learning'. arXiv, Jun. 13, 2022. http://arxiv.org/abs/2206.06178

RUHR UNIVERSITÄT BOCHUM

RUB

TECHNISCHE UNIVERSITÄT DRESDEN