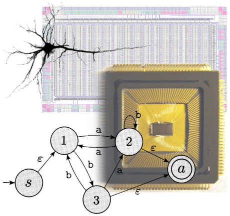
Event-Driven Random Backpropagation: Enabling Neuromorphic Deep Learning Machines

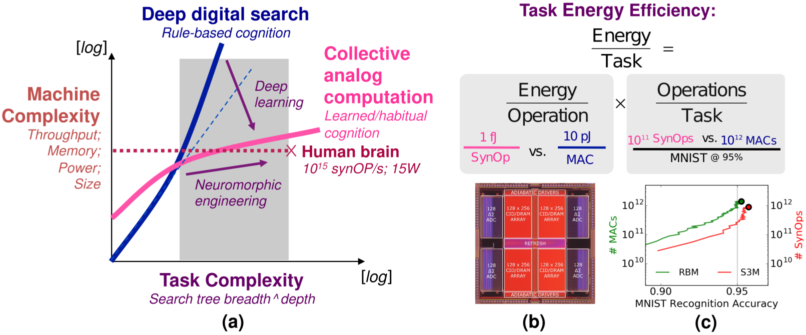
Emre Neftci

Department of Cognitive Sciences, UC Irvine,
Department of Computer Science, UC Irvine,

March 7, 2017



Scalable Event-Driven Learning Machines



Cauwenberghs, *Proceedings of the National Academy of Sciences*, 2013

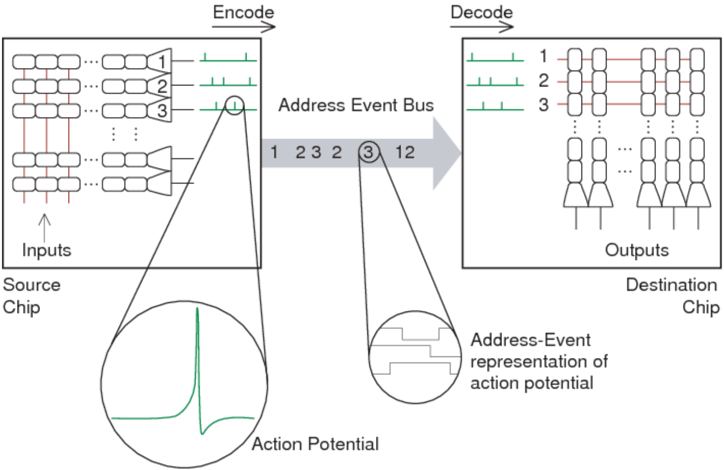
Karakiewicz, Genov, and Cauwenberghs, *IEEE Sensors Journal*, 2012

Neftci, Augustine, Paul, and Detorakis, *arXiv preprint arXiv:1612.05596*, 2016

1000x power improvements compared to future GPU technology through two factors:

- Architecture and device level optimization in event-based computing
- Algorithmic optimization in neurally inspired learning and inference

Neuromorphic Computing Can Enable Low-power, Massively Parallel Computing



- Only spikes are communicated & routed between neurons (weights, internal states are local)
- To use this architecture for practical workloads, we need algorithms that operate on local information

Why Do Embedded Learning?

For many industrial applications involving controlled environments, where existing data is readily available, off-chip/off-line learning is often sufficient.

So why do embedded learning?

Two main use cases:

- Mobile, low-power platform in uncontrolled environments, where adaptive behavior is required.
- Working around device mismatch/non-idealities.

Potentially rules out:

- Self-driving cars
- Data mining
- Fraud Detection

Neuromorphic Learning Machines: Online learning for data-driven autonomy and algorithmic efficiency

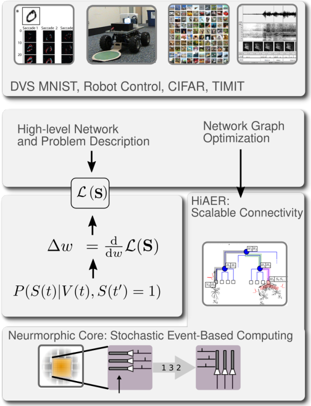
VARIABLE COMPLEXITY TASKS

EVENT-BASED MACHINE LEARNING FRAMEWORK

STOCHASTIC LEARNING

THEORY OF EVENT-BASED STOCHASTIC COMPUTING

NEUROMORPHIC HARDWARE DESIGN



- **Hardware & Architecture:** Scalable Neuromorphic Learning Hardware Design
- **Programmability:** Neuromorphic supervised, unsupervised and reinforcement learning framework

Foundations for Neuromorphic Machine Learning Software Framework & Library

neon_mlp_extract.py

```
# setup model layers
layers = [Affine(nout=100, init=init_norm, activation=Rectlin()),
          Affine(nout=10, init=init_norm, activation=Logistic(shortcut=True))]

# setup cost function as CrossEntropy
cost = GeneralizedCost(costfunc=CrossEntropyBinary())

# setup optimizer
optimizer = GradientDescentMomentum(
        0.1, momentum_coef=0.9, stochastic_round=args.rounding)
```

Can we design a digital neuromorphic learning machine that is flexible and efficient?

- **Leaky Stochastic I&F Neuron (LIF)**

$$V[t+1] = -\alpha V[t] + \sum_{j=1}^{n} \xi_j w_j(t) s_j(t) \tag{1a}$$

$$V[t+1] \geq T : V[t+1] \leftarrow V_{reset} \tag{1b}$$

- **LIF with first order kinetic synapse**

$$V[t+1] = -\alpha V[t] + I_{syn} \tag{2a}$$

$$I_{syn}[t+1] = -a_1 I_{syn}[t] + \sum_{j=1}^{n} w_j(t) s_j(t) \tag{2b}$$

$$V[t+1] \geq T : V[t+1] \leftarrow V_{reset} \tag{2c}$$

- **LIF with second order kinetic synapse**

$$V[t+1] = -\alpha V[t] + I_{syn} + I_{syn}, \tag{3a}$$

$$I_{syn}[t+1] = -a_1 I_{syn}[t] + c_1 I_s[t] + \eta[t] + b \tag{3b}$$

$$I_s[t+1] = -a_2 I_s[t] + \sum_{j=1}^{n} w_j s_j[t] \tag{3c}$$

$$V[t+1] \geq T : V[t+1] \leftarrow V_{reset} \tag{3d}$$

- **Dual-Compartment LIF with synapses**

$$V_1[t+1] = -\alpha V_1[t] + \alpha_{21} V_2[t] \tag{4a}$$

$$V_2[t+1] = -\alpha V_2[t] + \alpha_{12} V_1[t] + I_{syn} \tag{4b}$$

$$I_{syn}[t+1] = -a_1 I_{syn}[t] + \sum_{j=1}^{n} w_j^1(t) s_j(t) + \eta[t] + b \tag{4c}$$

$$V_1[t+1] \geq T : V_1[t+1] \leftarrow V_{reset} \tag{4d}$$

- **Mihalas Niebur Neuron (MNN)**

$$V[t+1] = \alpha V[t] + I_e - G \cdot E_L + \sum_{i=1}^{n} I_i[t] \tag{5a}$$

$$\Theta[t+1] = (1-b)\Theta[t] + aV[t] - aE_L + b \tag{5b}$$

$$I_1[t+1] = -\alpha_1 I_1[t] \tag{5c}$$

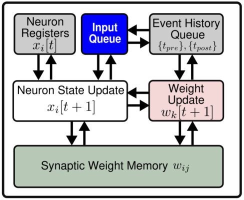
$$I_2[t+1] = -\alpha_2 I_2[t] \tag{5d}$$

$$V[t+1] \geq \Theta[t+1] : Reset(V[t+1], I_1, I_2, \Theta) \tag{5e}$$

MNN can produce a wide variety of spiking behaviors

Mihalas and Niebur, *Neural Computation*, 2009

NSAT Core (2048 Neurons)



Neuron and Synapse Model

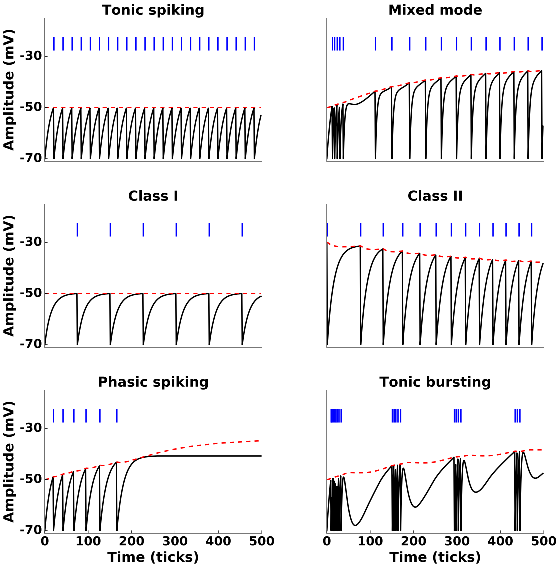
$$
\begin{aligned}
\mathbf{x}[t+1] &= \mathbf{A}\mathbf{x}[t] && \textbf{(Leak \& Coupling)}\\
&+ \ \Xi[t] \otimes \mathbf{W}[t]\mathbf{S}[t] && \textbf{(Synaptic inputs)}\\
&+ \ \boldsymbol{\eta}[t] && \textbf{(Noise)}\\
\mathbf{x}[t+1] &\geq \boldsymbol{\theta}, \mathbf{x}[t+1] \leftarrow \mathbf{X_r} && \textbf{(Thresholds \& Reset)}\\
x_0[t+1] &\geq \theta_0, s_i[t+1] \leftarrow 1 && \textbf{(Spiking Output)}
\end{aligned}
$$

$$
\begin{aligned}
e_k &= x_m[t]\left(K[t-t_k] + K[t_k - t_{last}]\right) && \textbf{(Eligibilty)}\\
w_k[t+1] &= w_k[t] + s_k[t+1]e_k && \textbf{(Weight update)}
\end{aligned}
$$

- Multicompartment generalized integrate-and-fire neurons
- Multiplierless design
- Weight sharing (convnets) at the level of the core

Equivalent software simulations for analyzing fault tolerance, precision, performance, and efficiency trade-offs (available publicly soon!)

NSAT Neural Dynamics Flexibility



Detorakis, Augustine, Paul, Pedroni, Sheik, Cauwenberghs, and Neftci (in preparation)

Flexible Learning Dynamics

$$w_k[t+1] = w_k[t] + s_k[t+1]e_k \qquad \text{(Weight update)}$$

$$e_k = x_m \underbrace{\left(K[t-t_k] + K[t_k - t_{last}]\right)}_{STDP} \qquad \text{(Eligibilty)}$$

$$x_m = \sum_i \gamma_i x_i \qquad \text{(Modulation)}$$

Detorakis, Augustine, Paul, Pedroni, Sheik, Cauwenberghs, and Neftci (in preparation)

$$w_k[t+1] = w_k[t] + s_k[t+1]e_k \qquad \text{(Weight update)}$$

$$e_k = x_m \underbrace{\left(K[t-t_k] + K[t_k - t_{last}]\right)}_{STDP} \qquad \text{(Eligibilty)}$$

$$x_m = \sum_i \gamma_i x_i \qquad \text{(Modulation)}$$

Detorakis, Augustine, Paul, Pedroni, Sheik, Cauwenberghs, and Neftci (in preparation)

Based on two insights:

Causal and acausal STDP weight updates on pre-synaptic spikes only, using only forward lookup access of the synaptic connectivity table

Pedroni et al.,, 2016

"Plasticity involves as a third factor a local dendritic potential, besides pre- and postsynaptic firing times"

Urbanczik and Senn, *Neuron*, 2014

Clopath, Büsing, Vasilaki, and Gerstner, *Nature Neuroscience*, 2010

- **Reinforcement Learning**

$$\Delta w_{ij} = \eta r STDP_{ij}$$

Florian, *Neural Computation*, 2007

- **Unsupervised Representation Learning**

$$\Delta w_{ij} = \eta g(t) STDP_{ij}$$

Neftci, Das, Pedroni, Kreutz-Delgado, and Cauwenberghs, *Frontiers in Neuroscience*, 2014

- **Unsupervised Sequence Learning**

$$\Delta w_{ij} = \eta \left(\Theta(V) - \alpha(\nu_i - C)\right) \nu_j$$

Sheik et al. 2016

- **Supervised Deep Learning**

$$\Delta w_{ij} = \eta(\nu_{tgt} - \nu_i)\phi'(V)\nu_j$$

Neftci, Augustine, Paul, and Detorakis, *arXiv preprint arXiv:1612.05596*, 2016

- **Reinforcement Learning**

$$\Delta w_{ij} = \eta r STDP_{ij}$$

Florian, *Neural Computation*, 2007

- **Unsupervised Representation Learning**

$$\Delta w_{ij} = \eta g(t) STDP_{ij}$$

Neftci, Das, Pedroni, Kreutz-Delgado, and Cauwenberghs, *Frontiers in Neuroscience*, 2014

- **Unsupervised Sequence Learning**

$$\Delta w_{ij} = \eta \left(\Theta(V) - \alpha(\nu_i - C)\right) \nu_j$$

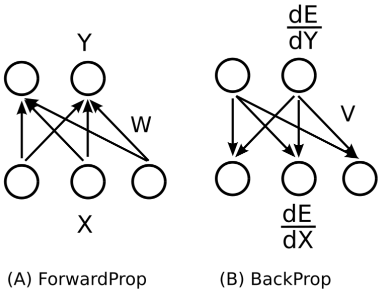
Sheik et al. 2016

- **Supervised Deep Learning**

$$\Delta w_{ij} = \eta(\nu_{tgt} - \nu_i)\phi'(V)\nu_j$$

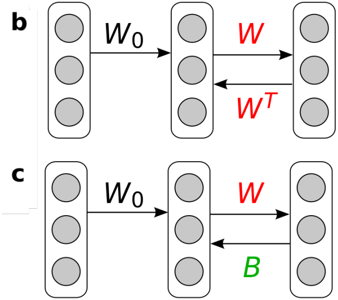
Neftci, Augustine, Paul, and Detorakis, *arXiv preprint arXiv:1612.05596*, 2016

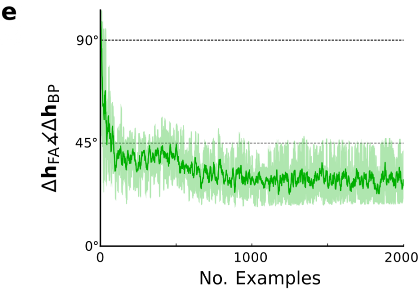
Gradient Backpropagation (BP) is non-local on Neural Substrates



Potential incompatibilities of BP on a neural (neuromorphic) substrate:

1. Symmetric Weights
2. Computing Multiplications and Derivatives
3. Propagating error signals with high precision
4. Precise alternation between forward and backward passes
5. Synaptic weights can change sign
6. Availability of targets





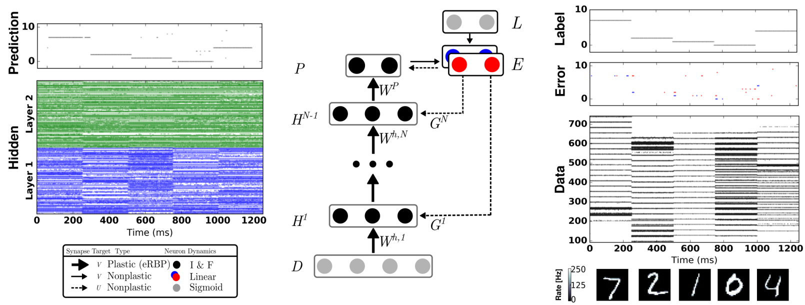
Replace weight matrices in backprop phase with (fixed) random weights

Lillicrap, Cownden, Tweed, and Akerman, *arXiv preprint arXiv:1411.0247*, 2014

Baldi, Sadowski, and Lu, *arXiv preprint arXiv:1612.02734*, 2016

- Event-driven Random Backpropagation Learning Rule: Error-modulated, membrane voltage-gated, event-driven, supervised.

$$\Delta w_{ik} \propto \underbrace{\phi'(I_{syn,i}[t])}_{\textit{Derivative}} S_k[t] \sum_j G_{ij} \underbrace{(L_j[t] - P_j[t])}_{\text{Error}} \qquad \text{(eRBP)}$$



- Event-driven Random Backpropagation Learning Rule: Error-modulated, membrane voltage-gated, event-driven, supervised.

$$\Delta w_{ik} \propto \underbrace{\phi'(I_{syn,i}[t])}_{Derivative} S_k[t] \underbrace{\sum_j G_{ij} \underbrace{(L_j[t] - P_j[t])}_{\text{Error}}}_{T_i} \qquad \text{(eRBP)}$$

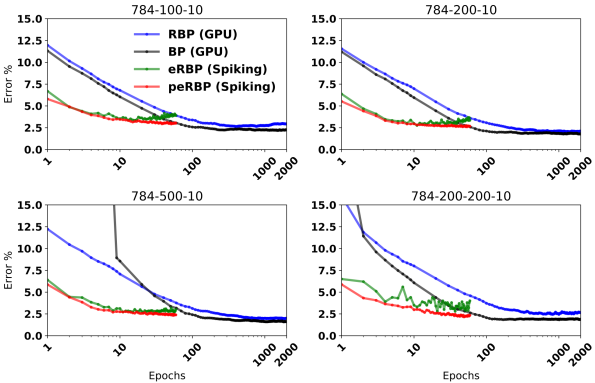
Approximate derivative with a boxcar function:

function ERBP
  for $k \in \{$presynaptic spike addresses $\mathbf{S}^{pre}\}$ **do**
    if $b_{min} < I < b_{max}$ **then** $w_k \leftarrow w_k + T$,
    end if
  end for
end function

Neftci, Augustine, Paul, and Detorakis, *arXiv preprint arXiv:1612.05596*, 2016

One addition and two comparison per synaptic event

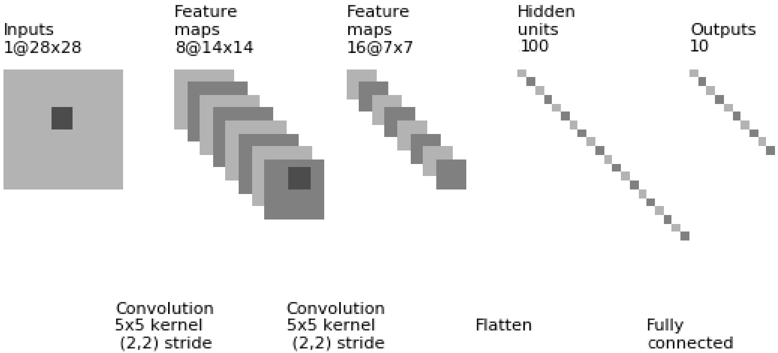
eRBP PI MNIST Benchmarks



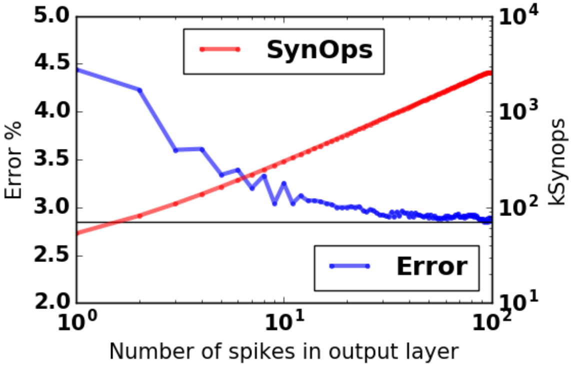
Network	Classification Error			
Dataset	eRBP	peRBP	RBP (GPU)	BP (GPU)
PI MNIST 784-100-10	**3.94%**	**3.02%**	2.74%	2.19%
PI MNIST 784-200-10	**3.53%**	**2.69%**	2.15%	1.81%
PI MNIST 784-500-10	**2.76%**	**2.40%**	2.08%	1.8%
PI MNIST 784-200-200-10	**3.48%**	**2.29%**	2.42%	1.91%
PI MNIST 784-500-500-10		**2.02%**	2.20%	1.90%

peRBP = eRBP with stochastic synapses

peRBP MNIST Benchmarks (Convolutional Neural Net)



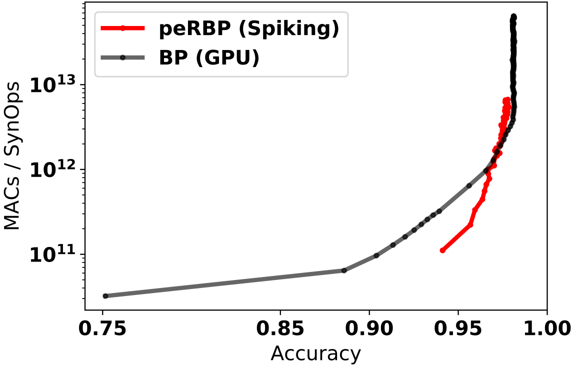
Network	**Classification Error**		
Dataset	peRBP	RBP (GPU)	BP (GPU)
MNIST	**3.8 (5 epochs)%**	1.95%	1.23%



Energy Efficieny During Inference:

- Inference: $\cong 100k$ Synops until first spike: <5% error, $100,000$ SynOps per classification

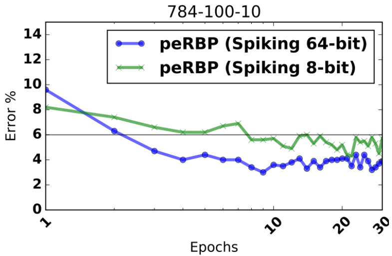
	eRBP	DropConnect (GPU)	Spinnaker	True North
Implementation	(20 pJ/Synop)	CPU/GPU	ASIC	ASIC
Accuracy	95%	99.79%	95%	95%
Energy/classify	2 μJ	1265 μJ	6000 μJ	$4 \mu J$
Technology		28 nm	Unknown	28 nm

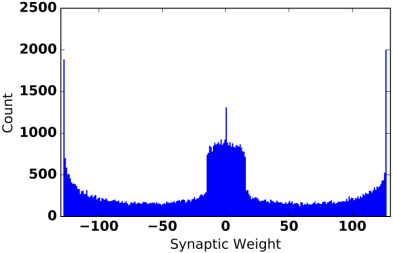


Energy Efficieny During Training:

- Training: SynOp-MAC parity

Embedded local plasticity dynamics for continuous (life-long) learning





- 16 bits neural states
- 8 bits synaptic weights
- $\cong$ 1Mbit Synaptic Weight Memory

All-digital implementation for exploring scalable event-based learning

UCI (Neftci, Krichmar, Dutt), UCSD (Cauwenberghs)

Summary:

1. NSAT: Flexible and efficient neural learning machines
2. Supervised deep learning with event-driven random back-propagation can achieve good learning results at >100x energy improvements

Challenges:

1. Catastrophic Forgetting: Need for Hippocampus, Intrinsic Replay and Neurogenesis
2. Build a neuromorphic library of “deep learning tricks” (Batch normalization, Adam, . . .)

Acknowledgements

Collaborators:

Georgios Detorakis

(UCI)

Somnath Paul (Intel)

Charles Augustine

(Intel)

Support:

NSF

SRC Semiconductor Research Corporation

Nanoelectronics Research Initiative

intel

P. Baldi, P. Sadowski, and Zhiqin Lu. "Learning in the Machine: Random Backpropagation and the Learning Channel". In: *arXiv preprint arXiv:1612.02734* (2016).

Gert Cauwenberghs. "Reverse engineering the cognitive brain". In: *Proceedings of the National Academy of Sciences* 110.39 (2013), pp. 15512–15513.

C. Clopath, L. Büsing, E. Vasilaki, and W. Gerstner. "Connectivity reflects coding: a model of voltage-based STDP with homeostasis". In: *Nature Neuroscience* 13.3 (2010), pp. 344–352.

R.V. Florian. "Reinforcement learning through modulation of spike-timing-dependent synaptic plasticity". In: *Neural Computation* 19.6 (2007), pp. 1468–1502.

R. Karakiewicz, R. Genov, and G. Cauwenberghs. "1.1 TMACS/mW Fine-Grained Stochastic Resonant Charge-Recycling Array Processor". In: *IEEE Sensors Journal* 12.4 (Apr. 2012), pp. 785–792.

Timothy P Lillicrap, Daniel Cownden, Douglas B Tweed, and Colin J Akerman. "Random feedback weights support learning in deep neural networks". In: *arXiv preprint arXiv:1411.0247* (2014).

S. Mihalas and E. Niebur. "A generalized linear integrate-and-fire neural model produces diverse spiking behavior". In: *Neural Computation* 21 (2009), pp. 704–718.

E. Neftci, S. Das, B. Pedroni, K. Kreutz-Delgado, and G. Cauwenberghs. "Event-Driven Contrastive Divergence for Spiking Neuromorphic Systems". In: *Frontiers in Neuroscience* 7.272 (Jan. 2014). ISSN: 1662-453X. DOI: 10.3389/fnins.2013.00272. URL: http://www.frontiersin.org/neuromorphic_engineering/10.3389/fnins.2013.00272/abstract.

Emre Neftci, Charles Augustine, Somnath Paul, and Georgios Detorakis. "Event-driven Random Back-Propagation: Enabling Neuromorphic Deep Learning Machines". In: *arXiv preprint arXiv:1612.05596* (2016).

Bruno U Pedroni, Sadique Sheik, Siddharth Joshi, Georgios Detorakis, Somnath Paul, Charles Augustine, Emre Neftci, and Gert Cauwenberghs. "Forward Table-Based Presynaptic Event-Triggered Spike-Timing-Dependent Plasticity". In: Oct. 2016. URL: %7BIEEE%20Biomedical%20Circuits%20and%20Systems%20Conference%20(BioCAS),%20https://arxiv.org/abs/1607.03070%7D.

Robert Urbanczik and Walter Senn. “Learning by the dendritic prediction of somatic spiking”. In: *Neuron* 81.3 (2014), pp. 521–528.