

UC Irvine

UC Irvine Previously Published Works

Title

Stochastic Synapses Enable Efficient Brain-Inspired Learning Machines.

Permalink

<https://escholarship.org/uc/item/4wf3185b>

Authors

Neftci, Emre O
Pedroni, Bruno U
Joshi, Siddharth
[et al.](#)

Publication Date

2016

DOI

10.3389/fnins.2016.00241

Peer reviewed

Event-driven Random Backpropagation: Enabling Neuromorphic Deep Learning Machines

Emre Neftci^{*†}, Charles Augustine[†], Somnath Paul[†], and, Georgios Detorakis^{*}

^{*}Department of Cognitive Sciences, UC Irvine, Irvine, USA

[†]Intel Corporation, Hillsboro, USA

[‡]eneftci@uci.edu

Abstract

An ongoing challenge in neuromorphic computing is to devise general and computationally efficient models of inference and learning which are compatible with the spatial and temporal constraints of the brain. The gradient descent backpropagation rule is a powerful algorithm that is ubiquitous in deep learning, but it relies on the immediate availability of network-wide information stored with high-precision memory. However, recent work shows that exact backpropagated weights are not essential for learning deep representations. Random backpropagation replaces feedback weights with random ones and encourages the network to adjust its feed-forward weights to learn pseudo-inverses of the (random) feedback weights. Here, we demonstrate an event-driven random backpropagation (eRBP) rule that uses an error-modulated synaptic plasticity for learning deep representations in neuromorphic computing hardware. The rule is very suitable for implementation in neuromorphic hardware using a two-compartment leaky integrate & fire neuron and a membrane-voltage modulated, spike-driven plasticity rule. Our results show that using eRBP, deep representations are rapidly learned without using backpropagated gradients, achieving nearly identical classification accuracies compared to artificial neural network simulations on GPUs, while being robust to neural and synaptic state quantizations during learning.

I. INTRODUCTION

Biological neurons and synapses can provide the blueprint for inference and learning machines that are potentially thousandfold more energy efficient than mainstream computers. However, the breadth of application and scale of present-day neuromorphic hardware remains limited, mainly due to a lack of general and efficient inference and learning algorithms compliant with the spatial and temporal constraints of the brain.

Machine learning and deep learning are well poised for solving a broad set of applications using neuromorphic hardware, thanks to their general-purpose, modular, and fault-tolerant nature [1], [2], [3]. One outstanding question is whether the learning phase in deep neural networks can be efficiently carried out in neuromorphic hardware as well. Performing learning online can confer continuous adaptability in dynamic or less controlled environments (where no prior, representative dataset exists) and more fine-grained context awareness to behaving cognitive agents, while reducing the energetic footprint of learning. However, deep learning usually relies on the immediate availability of network-wide information stored with high-precision memory. The non-locality of deep learning on a brain-like substrate is due to the gradient descent Back-Propagation (BP) rule [4], which requires the transpose of the feedforward matrices in the feedback path. In digital computers, the access to this information funnels through the von Neumann bottleneck, which dictates the fundamental limits of the computing substrate. Distributing computations along multiple cores (such as in GPUs) is a popular solution to mitigate this problem, but even there the scalability of BP is often limited by its memory-intensive operations [5].

Here, we build on the recent advances in approximate forms of the gradient BP rule [6], [7], [8]

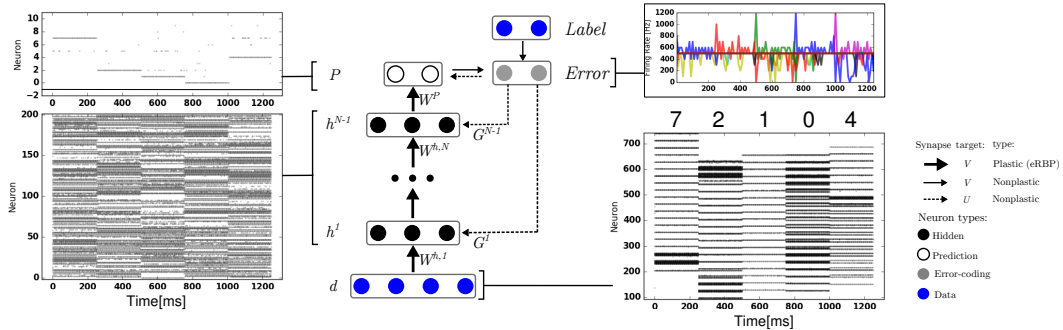


Figure 1: Network Architecture for Event-driven Random Backpropagation (eRBP) and example spiking activity during training. The network consists in feed-forward layers for prediction and feedback layers for supervised training with labels (targets) L . Full arrows indicate synaptic connections, dashed arrows indicate synaptic plasticity modulation. In this example, digits 7,2,1,0,4 were presented in sequence to the network. The digit pixel values were transformed to spike trains using a Spike Response Model. All other neurons in the network implemented two-compartment leaky I&F neurons. The error is the difference between labels (L) and predictions (P), offset by 500Hz.

PI Dataset	Network	Classification Error			
		eRBP	peRBP	RBP (GPU)	BP (GPU)
MNIST	784-100-10	3.89%	3.65%	2.85%	2.19%
MNIST	784-200-10	2.85%	2.68%	2.15%	1.81%
MNIST	784-500-10	2.75%	2.24%	2.08%	1.8%
MNIST	784-200-200-10	3.30%	2.48%	2.42%	1.91%

Table I: Bold indicates online trained spiking network

for training spiking neurons of the type used in neuromorphic hardware to perform supervised learning. These approximations solve the non-locality problem by replacing BP weights with random ones, leading to remarkably little loss in classification performance on benchmark tasks [8], [9]. Although a general theoretical understanding of random BP (RBP) is still lacking, extended simulations and analyses of linear networks show that, during learning, the network adjusts its feed-forward weights to learn an approximation of the pseudo-inverse of the (random) feedback weights, which is at least equally good in communicating gradients.

We describe how an event-driven implementation of RBP can be tightly embedded with the neuron dynamics, and lay out the foundations of neuromorphic deep learning machines. Our results show that MNIST classification performances of the on-line trained spiking network are nearly identical to those of equivalent artificial neural networks running on digital computers. Finally, we show that eRBP is robust in fixed width simulations with limited neural and synaptic state precision.

II. EVENT-DRIVEN RANDOM BACKPROPAGATION

Event-driven RBP (eRBP) is a presynaptic spike-driven rule modulated by top-down errors and gated by the membrane voltage of the postsynaptic neuron. The idea behind this additional modulation factor is similar in spirit to previous biologically plausible models of three-factor plasticity rules [10], and was argued to subserve supervised, unsupervised and reinforcement

learning, an idea that was also reported in [8]. The eRBP learning rule can be summarized as follows:

$$\Delta w = T[t] \Theta(V[t]) S^{\text{pre}}(t) \quad (1)$$

where $S^{\text{pre}}(t)$ is the pre-synaptic spike train, and a Θ is the derivative of the spiking neuron's activation function evaluated at the post-synaptic membrane potential $V[t]$. For the final output (prediction) layer, T is equal to the classification error E , similarly to the delta rule. For hidden layers, the T is equal to the error multiplied by a random vector G_i , i.e. $T = \sum_j G_{ij} E_j$. All G_{ij} are fixed during the learning.

For Integrate & Fire (I&F) neurons, we find that a boxcar function in place of Θ provides very good results, while being more amenable to hardware implementation than computing the exact derivative of the activation function. This choice is motivated by the fact that the activation function of leaky I&F neurons with absolute refractory period can be approximated by a threshold unit with saturation. Using a boxcar function with boundaries V_{\min} and V_{\max} , the eRBP update consist of additions and comparisons only, and can be captured using the following operations for each neuron:

```

function ERBP
  for  $k \in \{\text{presynaptic spike addresses } S^{\text{pre}}(t)\}$  do
    if  $V_{\min} < V < V_{\max}$  then  $w_k(t) \leftarrow w_k(t) + T(t)$ ,
    end if
  end for
end function

```

This rule is reminiscent of membrane voltage-based rules, where spike-driven plasticity is induced only when membrane voltage is inside an eligibility window [11], [12]. Furthermore, ERBP does not require “causal” updates of Spike Timing Dependent Plasticity (STDP), so reverse look-up tables or recently studied approximate forms of STDP are not necessary [13]. The learning rule assumes that the $T(t)$ modulation is available at the pre-synaptic synapses. In the following section, we show how eRBP can be realized in a neural network of Integrate & Fire (I&F) neurons.

A. A Spiking Network for eRBP with Error-Coding Neurons

The network used for eRBP consists of one or two feed-forward layers (Fig. 1), although results can be generalized to any feedforward connectivity between any number of layers. The top layer, labeled P , is the prediction. The feedback from the error population is fed back directly to the hidden layers' neurons. This type of feedback is also called “skipped RBP”[9]. This direct feedback simplifies the delivery of the error modulation to the hidden neurons and improves the performance of the learning on standard benchmarks [9].

The proposed network is composed of three types of neurons:

1) **Error-coding neurons** are non-leaky I&F neurons following the linear dynamics:

$$C \frac{d}{dt} V_i^E = w^L(S_i^P(t) - S_i^L(t)) + b^E \quad (2)$$

$$\text{if } V^E > V_T^E \text{ then } V^E \leftarrow V^E - V_T^E,$$

where $S_i^P(t)$ and $S_i^L(t)$ are spike trains from prediction neurons and labels (teaching signal), respectively and b^E is a fixed common bias. The notation for time dependencies of V and U are omitted to reduce clutter. The firing rate of the error-coding neurons is proportional to a linear rectification of the inputs. When the prediction neurons classify correctly, the two terms on the rhs cancel and the neuron fires at a constant rate $\bar{r} = \frac{b^E}{C^E V_T^E}$. When an error occurs, the firing

rate deviates linearly from \bar{r} down to the rectification point.

2) **Hidden neurons** follow current-based leaky I&F dynamics:

$$C \frac{d}{dt} \begin{pmatrix} V_i^h \\ U_i^h \end{pmatrix} = - \begin{pmatrix} g_V V_i^h \\ g_U U_i^h \end{pmatrix} + \begin{pmatrix} I_i^h + \sigma_w \eta_i^h(t) \\ \sum_{k=1}^{N_L} w_{ik}^E S_k^E(t) \end{pmatrix}$$

$$\tau_{syn} \frac{d}{dt} I_i^h = -I_i^h + \sum_{k=1}^{N_d} W_{ik}^d S_k^d(t) \xi_{ik}^p(t) + \sum_{j=1}^{N_h} W_{ij}^h S_j^h(t) \xi_{ik}^p(t)$$

if $V_i(t) > V_T$ then $V_i^h \leftarrow 0$ during refractory period τ_{refr} .

where $S_k^d(t)$ and $S_j^h(t)$ are the spike trains of the data neurons and the hidden neurons, respectively, I^h are current-based synapse dynamics, $\sigma_w \eta_i^h(t)$ a Poisson process of rate 100Hz, and ξ^p is a stochastic Bernoulli process with probability p . The latter type of stochasticity was previously shown to be beneficial for regularization and decorrelation of spike trains, while being easy to implement in neuromorphic hardware [2]. In this work, we consider feed-forward networks, *i.e.* the weight matrix \mathbf{W}^h is restricted to be upper diagonal. Each neuron is equipped with a separate ‘‘dendritic’’ compartment U_i^h following similar subthreshold dynamics as the membrane potential and where $S^E(t)$ is the spike train of the error-coding neurons and \mathbf{W}^E is a fixed random matrix such that for every hidden neuron i , $\sum_j W_{ij}^E = 0$. This condition ensures that the spontaneous firing rate of the error-coding neurons does not bias the learning. The synaptic weights dynamics follow a dendrite-modulated and neural state-gated rule:

$$\frac{d}{dt} W_{ij}^h = W_{ij}^h + U_i^h \Theta(V_i^h) S_j^h(t). \quad (3)$$

3) **Prediction neurons**, synapses and synaptic weight updates follow the same dynamics as the hidden neurons except for the dendritic compartment:

$$C \frac{d}{dt} \begin{pmatrix} V_i^P \\ U_i^P \end{pmatrix} = - \begin{pmatrix} g_V V_i^P \\ g_U U_i^P \end{pmatrix} + \begin{pmatrix} I_i^P + \sigma_w \eta_i^P(t) \\ w^P S_i^E(t) - \bar{r} \frac{C_d}{g_U} \end{pmatrix}, \quad (4)$$

where the extra bias term $-\bar{r} \frac{C_d}{g_U}$ counteracts the effect of the spontaneous firing rate of the error-coding neurons.

B. Experimental Setup and Software Simulations

We trained fully connected feed-forward networks MNIST hand-written digits, separated in three groups, training, validation, and testing (50000, 10000, 10000 samples respectively). During a training epoch, each of the training diits were presented in sequence during 250ms. The pixel intensities of the samples were transformed into spike trains online using ‘‘sigmoid neurons’’, *i.e.* a Spike Response Model [14] with an exponential activation (hazard) function and a refractory period matching those of the hidden and prediction neurons. We tested eRBP using two configurations: one with additive noise ($\sigma_w > 0$, $p = 1$, labeled eRBP), and one with multiplicative noise implemented as blank-out noise on the connections (blank-out probability $p < 1$ and $\sigma_w = 0$, labeled peRBP). Stochasticity was beneficial in reducing synchronization across hidden and prediction layers. Furthermore, to prevent the network from learning (spurious) transitions between digits, the synaptic weights did not update in the first 50ms window of each digit presentation.

We tested eRBP training on two different implementations: 1) Spiking neural network based on the Auryn simulator [15], and 2) Hardware compatible simulator with quantized neural states and weights. Results are compared against GPU implementations of RBP and standard BP performed in Theano [16] using an equivalent, non-spiking neural network. We focus

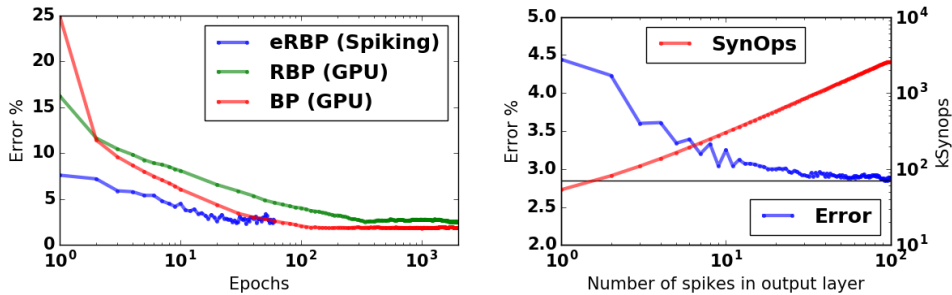


Figure 2: **(Left)** MNIST Classification error on a fully connected 784-200-200-10 network. The spiking network converged after 60 epochs. **(Right)** Classification error and number of synaptic operations using first spikes upon stimulus onset in the 784-200-10 network. Horizontal line is 2.85%.

on Permutation-Invariant (PI) tasks to emphasize that the network was unstructured (*e.g.* no convolutions or poolings).

III. RESULTS

A. Spiking Networks with eRBP Learn with High Accuracy

We tested eRBP in networks consisting of one and two hidden layers (Tab. I), although eRBP can generalize to deeper networks as well. Rather than optimizing for absolute classification performance, we compare to equivalent artificial (non-spiking) neural networks trained with RBP and standard BP, with free parameters fine-tuned to achieve high accuracy on the classification tasks. On most network configurations eRBP achieved performances equivalent to those achieved with RBP in artificial networks. ERBP equipped with probabilistic connections (peRBP) performed better overall, and more so for the deeper, 784-200-200-10, network. This is because, as learning progresses, a significant portion of the neurons tend to fire near their maximum rate and synchronize their spiking activity across layers as a result of large synaptic weights (and thus pre-synaptic inputs). Synchronized spike activity is not well captured by a simple firing rate model, which is inherently assumed by the eRBP formulation. However, probabilistic connections that randomly blank-out pre-synaptic spikes, effectively introduce irregularity in the pre-synaptic spike-trains to improve learning. Additive noise, on the other hand, has relatively small effect when the magnitude of the pre-synaptic input is large. Overall, the learned classification accuracy is comparable with those obtained with offline training of spiking neural networks (*e.g.* GPUs) using exact standard backpropagation [17].

B. Classification with Single Spikes

The response of the 784-200-10 network after stimulus onset is about one synaptic time constant. Using the first spike after $\tau_s = 4\text{ms}$ from the stimulus onset for classification leads to less than 4.5% error (Fig. 2), and improves steadily as the number output layer spikes increase. Classification using the first spike incurred 53407 synaptic operations (averaged over 10000 test samples), which can potentially result in sub-micro joule energy per classification in dedicated neuromorphic hardware [18]. The low latency and accurate response may seem at odds with the inherent firing rate code underlying the network computations. However, a code based on the time

of the first-spike is consistent with a firing rate code, since a neuron with a high firing rate is expected to fire first [14]. In addition, the onset of the stimulus provokes a burst of synchronized activity, which further favors the rapid onset of the response. These results suggest that despite the underlying firing rate code, eRBP takes advantage of the spiking dynamics, comparably to spiking networks trained exclusively for single-spike classification [19]

C. Spiking Networks Equipped with ERBP Learn Rapidly

The spiking neural network eRBP requires fewer epochs to reach the peak classification performance compared to the artificial neural network running RBP. The artificial neural network was trained using minibatches of $n_{batch} = 100$ data samples. Minibatches improve learning speed in conventional hardware thanks to vectorization libraries or leveraging GPUs, and lead to smoother convergence. However, the weight updates are averaged over the minibatch, resulting in n_{batch} times fewer weight updates per epoch compared to the spiking network per epoch where n_{batch} is effectively equal to one. Standard artificial neural networks can be trained using $n_{batch} = 1$, but learning becomes much slower on standard platforms because the operations cannot be vectorized across data samples. Such faster learning of the spiking network is a fortunate by-product of on-line gradient descent, given that potential applications of neuromorphic hardware often involve real-time streaming data.

D. Learning with Low Precision, Fixed Point Representations

The effectiveness of stochastic gradient descent degrades when the precision of the synaptic weights using a fixed point representation is smaller than 16 bits [20]. This is because quantization determines the smallest learning rate and bounds the range of the synaptic weights, thereby preventing averaging the variability across dataset iterations. The tight integration of memory with computing circuits as pursued in neuromorphic chip design is challenging due to space constraints and memory leakage. For this reason, full precision (or even 16 bit) computer simulations of spiking networks may be unrepresentative of performance that can be attained in dedicated neuromorphic chip due to quantization of neural states and parameters, and synaptic weights. Extended simulations suggest that the random BP performances at 10 bits precision is indistinguishable from unquantized weights [9], but whether this is the case for online learning without using minibatches was not yet tested. Here, we hypothesize that 8 bit synaptic weight is a good trade-off between learnability and hardware footprint. To demonstrate robustness to such constraints, we simulate quantized versions of the eRBP network using low precision fixed point representations (8 bits per synaptic weight and 16 bits for neural states). Consistent with existing findings, our simulations of eRBP in a quantized 784-100-10 network show that eRBP still performs reasonably well under these conditions (Fig. 3). While many weights aggregate at the boundaries, a majority of them remain between the boundaries.

IV. CONCLUSION

Our results demonstrate that general purpose deep learning using streaming spike-event data in neuromorphic platforms at artificial neural network proficiencies is realizable. ERBP can be implemented in a very compact fashion, while using local synaptic plasticity rules modulated by global error on a neuron-to-neuron basis. Although our experiments targeted digital neuromorphic implementations, membrane-voltage based learning rules implemented in mixed-signal neuromorphic hardware [12] are compatible with eRBP provided that synaptic weight updates can be modulated by an external signal on a neuron-to-neuron basis. Thus, the recent advances in neuromorphic engineering and emerging nanotechnologies combined with eRBP can become key to ultra low-power processing in space and power constrained platforms.

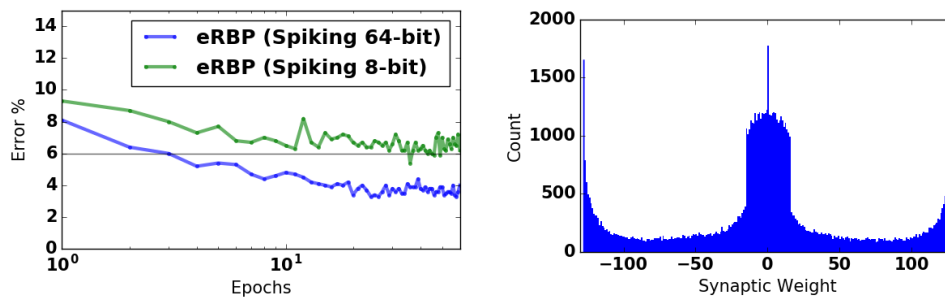


Figure 3: **(Left)** MNIST Classification error using a fully connected 784-100-10 network with quantized synaptic weights and neural states (8 bits and 16 bits respectively) **(Right)** Synaptic weights of the quantized network after training.

In tandem with developments in deep learning and recurrent neural networks, we envision that such RBP techniques will enable the embedded learning of pattern recognition, attention, working memory and action selection mechanisms which promise transformative hardware architectures for embedded computing.

ACKNOWLEDGMENTS

This work was partly supported by the Intel Corporation and by the National Science Foundation under grant 1640081, and the Nanoelectronics Research Corporation (NERC), a wholly-owned subsidiary of the Semiconductor Research Corporation (SRC), through Extremely Energy Efficient Collective Electronics (EXCEL), an SRC-NRI Nanoelectronics Research Initiative under Research Task ID 2698.003.

REFERENCES

- [1] S. K. Esser *et al.*, “Convolutional networks for fast, energy-efficient neuromorphic computing,” *PNAS*, vol. 113, pp. 11 441–11 446, 2016.
- [2] E. O. Neftci *et al.*, “Stochastic synapses enable efficient brain-inspired learning machines,” *Frontiers in Neuroscience*, vol. 10, no. 241, 2016.
- [3] J. H. Lee, T. Delbruck, and M. Pfeiffer, “Training deep spiking neural networks using backpropagation,” *Frontiers in Neuroscience*, vol. 10, 2016.
- [4] D. E. Rumelhart, J. L. McClelland, P. R. Group *et al.*, *Parallel distributed processing*. IEEE, 1988, vol. 1.
- [5] X. Zhu, M. Awatramani, D. Rover, and J. Zambreno, “Onac: Optimal number of active cores detector for energy efficient gpu computing,” in *Computer Design (ICCD), 2016 IEEE 34th International Conference on*. IEEE, 2016, pp. 512–519.
- [6] D.-H. Lee, S. Zhang, A. Biard, and Y. Bengio, “Target propagation,” *arXiv preprint arXiv:1412.7525*, 2014.
- [7] Q. Liao, J. Z. Leibo, and T. Poggio, “How important is weight symmetry in backpropagation?” *arXiv preprint arXiv:1510.05067*, 2015.
- [8] T. P. Lillicrap, D. Cownden, D. B. Tweed, and C. J. Akerman, “Random synaptic feedback weights support error backpropagation for deep learning,” *Nature Communications*, vol. 7, 2016.
- [9] P. Baldi, P. Sadowski, and Z. Lu, “Learning in the machine: Random backpropagation and the learning channel,” *arXiv preprint arXiv:1612.02734*, 2016.
- [10] R. Urbanczik and W. Senn, “Learning by the dendritic prediction of somatic spiking,” *Neuron*, vol. 81, no. 3, pp. 521–528, 2014.
- [11] J. Brader, W. Senn, and S. Fusi, “Learning real world stimuli in a neural network with spike-driven synaptic dynamics,” *Neural Computation*, vol. 19, pp. 2881–2912, 2007.
- [12] E. Chicca, F. Stefanini, and G. Indiveri, “Neuromorphic electronic circuits for building autonomous cognitive systems,” *Proceedings of IEEE*, 2013.
- [13] B. U. Pedroni *et al.*, “Forward table-based presynaptic event-triggered spike-timing-dependent plasticity,” Oct 2016.
- [14] W. Gerstner and W. Kistler, *Spiking Neuron Models. Single Neurons, Populations, Plasticity*. Cambridge University Press, 2002.

- [15] F. Zenke and W. Gerstner, "Limits to high-speed simulations of spiking neural networks using general-purpose computers," *Frontiers in neuroinformatics*, vol. 8, 2014.
- [16] J. Bergstra *et al.*, "Theano: a CPU and GPU math expression compiler," in *Proceedings of the Python for Scientific Computing Conference (SciPy)*, vol. 4, 2010.
- [17] P. U. Diehl *et al.*, "Fast-classifying, high-accuracy spiking deep networks through weight and threshold balancing," in *International Joint Conference on Neural Networks (IJCNN)*,. IEEE, 2015, pp. 1–8.
- [18] J. Park *et al.*, "A 65k-neuron 73-mevents/s 22-pj/event asynchronous micro-pipelined integrate-and-fire array transceiver," in *Biomedical Circuits and Systems Conference (BioCAS)*. IEEE, Oct. 2014.
- [19] H. Mostafa, "Supervised learning based on temporal coding in spiking neural networks," *arXiv preprint arXiv:1606.08165*, 2016.
- [20] M. Courbariaux, Y. Bengio, and J.-P. David, "Low precision arithmetic for deep learning," *arXiv preprint arXiv:1412.7024*, 2014.