

Article

EqSpike: spike-driven equilibrium propagation for neuromorphic implementations

EqSpike: Spike-driven Equilibrium Propagation for Neuromorphic Implementations

1- A spiking version of Equilibrium Propagation

$\frac{du}{dt} = -\gamma_{LIF}u + I(t) - \beta(\hat{\rho}_k - \rho_k)$

$\frac{du}{dt} = -\gamma_{LIF}u + I(t)$

3- Accuracy on MNIST reaches benchmark

Recognition rate: 97.6%

Fully connected architecture 784-300-10

```

Algorithm 1. Spike-based learning procedure for one image.
Inputs: input image, Model (Weights, Biases, Thresholds), Loss function, length free of
Output: phase, Parameters (Weights, Biases, Thresholds, Free of)
for t < T_max
  for each neuron i
    Update membrane potential u_i(t_{cur}, I)
    if u_i > u_{th}
      Emit a spike s_i
      Update u_i(t_{cur}, I)
  for i < T_max
    for each neuron i
      Compute error gradient g_i
      Backpropagate g_i to u_i
      if u_i > u_{th}
        Emit a spike s_i
        Update u_i(t_{cur}, I)
      Compute error g_i = (f(u_i) - f(u_i - 1)) * ... * g_{in}
  for each synapse w_{ij}
    if neuron i emits a spike
      w_{ij} = w_{ij} + \beta * s_i * s_j
    if neuron j emits a spike
      w_{ij} = w_{ij} - \beta * s_i * s_j
  Return: Trained weights for input images W_{ij} and go to next image/next epoch.
        
```

2- A hardware-compatible learning rule

Neuron *i* Synapse *ij* Neuron *j*

4- STDP-like weight updates

$t_{post} - t_{pre} [T_{refract}]$

5- High potential for hardware

Gain in energy consumption vs GPUs:

- Up to x 1000 for inference
- Up to x 100 for training

Erwann Martin,
Maxence Ernoult,
J er mie
Laydevant, Shuai
Li, Damien
Querlioz, Teodora
Petrisor, Julie
Grollier

julie.grollier@cnsr-thales.fr

HIGHLIGHTS

EqSpike is a spiking neural network version of equilibrium propagation

It achieves 97.6% test accuracy on MNIST with a fully connected architecture

Its two-factor local learning rule is compatible with neuromorphic hardware

Its weight updates exhibit a form of spike-timing-dependent plasticity



Article

EqSpike: spike-driven equilibrium propagation for neuromorphic implementations

Erwann Martin,¹ Maxence Ernout,^{2,3} Jérémie Laydevant,² Shuai Li,² Damien Querlioz,³ Teodora Petrisor,¹ and Julie Grollier^{2,*}

SUMMARY

Finding spike-based learning algorithms that can be implemented within the local constraints of neuromorphic systems, while achieving high accuracy, remains a formidable challenge. Equilibrium propagation is a promising alternative to backpropagation as it only involves local computations, but hardware-oriented studies have so far focused on rate-based networks. In this work, we develop a spiking neural network algorithm called EqSpike, compatible with neuromorphic systems, which learns by equilibrium propagation. Through simulations, we obtain a test recognition accuracy of 97.6% on the MNIST handwritten digits dataset (Mixed National Institute of Standards and Technology), similar to rate-based equilibrium propagation, and comparing favorably to alternative learning techniques for spiking neural networks. We show that EqSpike implemented in silicon neuromorphic technology could reduce the energy consumption of inference and training, respectively, by three orders and two orders of magnitude compared to graphics processing units. Finally, we also show that during learning, EqSpike weight updates exhibit a form of spike-timing-dependent plasticity, highlighting a possible connection with biology.

INTRODUCTION

Spike-based neuromorphic systems have, in recent years, demonstrated outstanding energy efficiency on inference tasks (Merolla et al., 2014). Implementing the training of deep neural networks in such systems remains, however, a considerable challenge, as backpropagation does not apply directly to spiking networks and requires spatially non-local computations that go against the principles of neuromorphic systems. A large number of neuromorphic systems use the unsupervised and biologically inspired spike-timing-dependent plasticity (STDP) learning rule because its weight updates, based on the relative timing of pre- and post-synaptic spikes, are spatially local and can be achieved with compact circuits in several technologies (Bi and Poo, 2001; Masquelier and Thorpe, 2007; Bichler et al., 2012; Zamarreño-Ramos et al., 2011; Jo et al., 2010; Pedretti et al., 2017; Serb et al., 2016; Prezioso et al., 2018; Thakur et al., 2018; Feldmann et al., 2019). Unfortunately, STDP weight updates generally do not minimize a global objective function for the network, and the accuracy of STDP-trained neural networks remains below state-of-the-art algorithms based on the error backpropagation (Falez et al., 2019). Important research efforts therefore investigate how the error backpropagation algorithm can be mathematically modified to make it spatially local and appropriate for spiking neural networks (Neftci et al., 2017; Sacramento et al., 2018; Richards et al., 2019; Neftci et al., 2019; Kaiser et al., 2020; Bellec et al., 2020; Payeur et al., 2020). The derived learning rules are composed of three factors. The first two take into account, as usual, the behavior of pre- and post-neurons, and the third allows for the introduction of an additional error factor. This third factor leads to implementations on neuromorphic chips that are less compact, and possibly less energy efficient, than two-factor learning rules such as STDP (Payvand et al., 2020).

In this work, we propose a different approach to training spiking neural networks with high accuracy while using a local, two-factor learning rule compatible with neuromorphic implementations and scalable to complex tasks. Instead of starting from a non-local algorithm such as backpropagation and modifying it to make it local, we start from a rate-based algorithm called equilibrium propagation (Scellier and Bengio,

¹Thales Research and Technology, 91767 Palaiseau, France

²Unité Mixte de Physique, CNRS, Thales, Université Paris-Saclay, 91767 Palaiseau, France

³Université Paris-Saclay, CNRS, Centre de Nanosciences et de Nanotechnologies, 91120 Palaiseau, France

*Correspondence:

julie.grollier@cnrs-thales.fr

<https://doi.org/10.1016/j.isci.2021.102222>



2017) that is intrinsically local in space and features key advantages for neuromorphic implementations (Kendall et al., 2020; Zoppo et al., 2020). Equilibrium propagation theoretically applies to any physical system whose dynamics derive from an energy function. By minimizing the energy of such a system on data patterns, it can be made to relax toward states of minimal error prediction with respect to targets (Scellier and Bengio, 2017). The weight updates of equilibrium propagation match those of back-propagation through time (BPTT) in recurrent neural networks with static inputs (Ernault et al., 2019), and it reaches high accuracy on image benchmarks such as the CIFAR-10 dataset (Canadian Institute For Advanced Research) (Laborieux et al., 2021). Equilibrium propagation uses the same set of weights for the forward and backward pass, a feature that is not biologically plausible, but is interesting for neuromorphic computing as it decreases the number of synaptic devices to update, thus reducing the overall power consumption. Contrarily to backpropagation, equilibrium propagation uses the same computations in the forward and backward phases, which is another highly desirable feature for neuromorphic systems as it greatly simplifies the circuits. Equilibrium propagation is, however, originally a rate-based algorithm.

Here, we design a spiking, hardware-friendly version of equilibrium propagation, called EqSpike, compatible with current neuromorphic technologies achieving online learning (Schemmel et al., 2010; Furber et al., 2014; Qiao et al., 2015; Davies et al., 2018; Frenkel et al., 2019; Ishii et al., 2019; Park et al., 2020). EqSpike is local in space and time: contrarily to backpropagation, neither error gradients nor activations need to be stored in external memories, and synapses can be directly updated through neural events. We simulate a fully connected network based on this architecture on the MNIST handwritten digit database. We obtain a test recognition accuracy of 97.6%, which compares favorably with spiking neural networks learning with backpropagation-derived methods, and on par with rate-based equilibrium propagation. We show that EqSpike can be implemented in silicon neuromorphic technology and thus reduce the energy consumption of inference by up to three orders of magnitude and training by up to two orders of magnitude compared to graphics processing units (GPUs). Finally, we also show that during learning, the weight updates of EqSpike exhibit a form of STDP, yielding insights to its link to biology.

RESULTS

EqSpike: a hardware-friendly spiking version of equilibrium propagation

Equilibrium propagation is an algorithm for training convergent recurrent neural networks. Input neurons are clamped to a static input, and all the other neurons, bidirectionally connected through synapses, evolve dynamically in time to reduce the energy of the network (Scellier and Bengio, 2017). The algorithm functions in two phases: a free phase and a nudging phase. In the free phase, performing inference, the network is let to reach equilibrium (Figure 1A). Once this is done, inputs are kept clamped, and output neurons are nudged toward the desired output (Figure 1B). During this nudging phase, the prediction error at the output layer is converted into a “force” acting upon output neurons and propagating to the rest of the system through time until a second equilibrium is reached. For training, synaptic values are updated by probing the neuron states after (Scellier and Bengio, 2017) or during (Ernault et al., 2020) the nudging phase through a learning rule that has been shown theoretically and numerically to match the updates of BPTT, the state-of-the-art algorithm for such recurrent neural networks (Ernault et al., 2019). It has been shown recently that equilibrium propagation also reaches accuracy within 1% of BPTT with convolutional architectures on the CIFAR-10 data set (Laborieux et al., 2021).

The original version of equilibrium propagation uses a rate-based formulation where dynamical neurons evolve smoothly in time. For a network of leaky integrate and fire neurons described by an Hopfield-like energy function $E(u) = 1/2 \sum_i u_i^2 - 1/2 \sum_{i \neq j} \rho(u_i) \rho(u_j) - \sum_i b_i \rho(u_i)$, where u are the membrane potentials of neurons and ρ their activation function, the equilibrium propagation learning rule is as follows: $\Delta W_{ij} \sim (\rho_i \rho_j)_n - (\rho_i \rho_j)_f$, where the product $\rho_i \rho_j$ is measured at equilibrium, at the end of the nudge phase and the free phase (Scellier and Bengio, 2017). This rule can be extended to the case when weights are continuously updated during the nudging phase (Ernault et al., 2020):

$$\frac{dW_{ij}}{dt} \sim \dot{\rho}_i \rho_j + \rho_i \dot{\rho}_j, \quad (\text{Equation 1})$$

where W_{ij} is the synaptic weight connecting neurons i and j , and ρ_i and ρ_j are the rates of the two neurons. The network dynamics thus directly compute the error derivative, encoded in the rate derivative of the post-neuron $\dot{\rho}$ and multiplied by the activation function of the pre-neuron. The reformulation of this

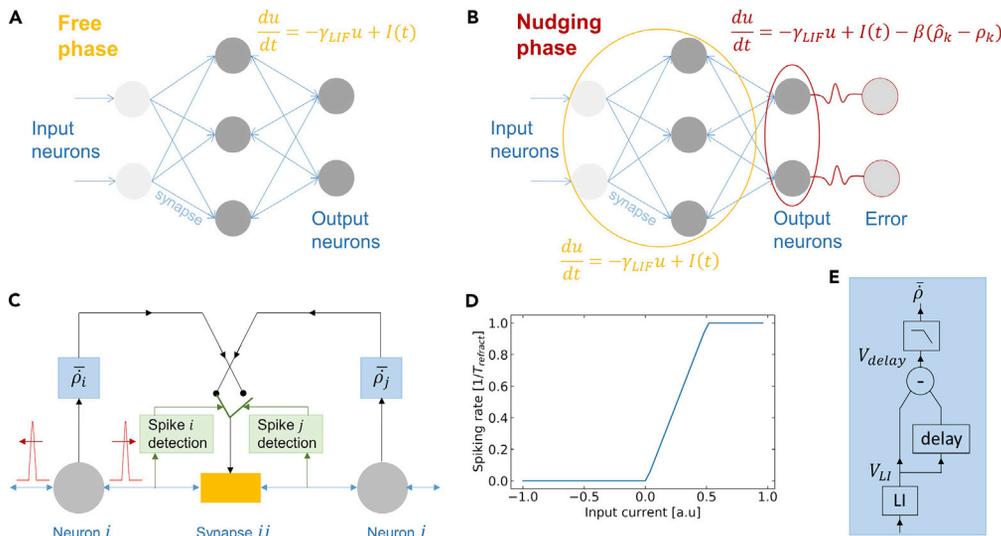


Figure 1. EqSpike: spike-driven equilibrium propagation

- (A) Schematic of the free phase in equilibrium propagation.
- (B) Schematic of the nudging phase in equilibrium propagation.
- (C) Illustration of the weight update implementation in EqSpike.
- (D) Spiking rate of the neuron as a function of the amplitude of the input signal.
- (E) Schematic of the rate acceleration computation.

rate-based learning rule to a spiking neural network is therefore the following: each time neuron i spikes, the weight should be updated by a quantity proportional to the derivative of the rate of neuron j , $\dot{\rho}_j$ (first term in Equation 1), and reciprocally.

We propose here a simple strategy, compatible with current electronic hardware, to implement this learning rule. It is illustrated in Figure 1C in the form of a circuit, including spike detection elements at the output of each neuron, as well as dedicated blocks that extract the rate derivative from the spike trains of each neuron in real time, in order to update synapses accordingly.

We use leaky-integrate-and-fire (LIF) spiking neurons which output spike frequency as a function of input current approximates the hard sigmoid prescribed in the original formulation of equilibrium propagation (Scellier and Bengio, 2017). Their maximum frequency is $f_{\max} = 1/T_{\text{refract}}$, where T_{refract} is the refractory time of the LIF neuron (see supplemental information for details and all parameter values).

The novelty compared to standard spiking neural networks is the scheme that we propose for extracting the rate acceleration $\dot{\rho}$ for each neuron, illustrated in Figure 1E. A leaky integrator with a leak factor γ_{LI} (without reset nor spikes) takes as input the spike train emitted by the neuron to which it is connected and outputs a slowly varying signal proportional to the rate of the neuron spike train: $V_{LI} \sim \rho/\gamma_{LI}$ (Navarro et al., 2020). To take the derivative, we delay this signal by a duration τ and subtract the actual value with the delayed value: $V_{\text{delay}} = V_{LI}(t) - V_{LI}(t-\tau) \cong \tau \frac{\partial V_{LI}}{\partial t} \propto \tau/\gamma_{LI} \dot{\rho}$. We then apply a low-pass filter for smoothing the variations. The filter is simulated using an average over N_{filt} simulation steps: $\bar{x}(t) = \frac{1}{N_{\text{filt}}} \sum_{i=0}^{N_{\text{filt}}-1} x_i(t-i dt)$, where dt is the simulation time step.

The output of the filter that approximates $\frac{\tau}{\gamma_{LI}} \dot{\rho}$ is then multiplied by the coefficient η_r . The corresponding weight updates are $\Delta w_{ij} = \eta_r \frac{\tau}{\gamma_{LI}} \dot{\rho}_i$, which corresponds to an effective learning rate $l_r = \eta_r \frac{\tau}{\gamma_{LI}} = 1.5 \cdot 10^{-3}$.

This approach is hardware compatible as LIF neurons, leaky integrators, delays, and low pass filters are circuit elements that can be efficiently implemented in Complementary Metal Oxide Semiconductor (CMOS) technology (Mead and Ismail, 1989), and bidirectional synapses could be implemented with CMOS compatible emergent nano-devices such as memristors (Ishii et al., 2019; Marković et al., 2020;

Algorithm 1. EqSpike learning procedure for one image

Inputs: input image, model $(n_{inputs}, n_{hidden}, n_{out})$, loss function, length free phase T_{free} , length nudging phase T_{nudge} , parameters $\gamma_{LIF}, \gamma_{LI}, u_{th}, \beta, \eta_r, \tau, N_{filt}, W_{ij}$

for $t < T_{free}$: ■ free phase

 for each neuron j :

 Update membrane potential $u_j(\gamma_{LIF}, I_j)$

 if $u_j > u_{th}$:

 Emit a spike (t_j)

 Update $\rho_j(t_j, \gamma_{LI})$

for $t \in [T_{free}, T_{free} + T_{nudge}]$: ■ nudging phase

 for each output neuron o :

 Compute error gradient ∇e_o

 Nudge neuron: $u_o \leftarrow u_o - \beta \cdot \nabla e_o$

 for each neuron k :

 Update $u_k(\gamma_{LIF}, I_k)$

 if $u_k > u_{th}$:

 Emit a spike (t_k)

 Update $\rho_k(t_k, \gamma_{LI})$

 Compute smoothed: $\bar{\rho}_k((\rho_k(t_k), \rho_k(t_k - \tau)), \dots, N_{filt})$

 for each synapse w_{ij} : ■ Update synapses

 if neuron j emits a spike:

$w_{ij} \leftarrow w_{ij} + \eta_r \cdot \frac{x}{\gamma_{LI}} \bar{\rho}_i$

 if neuron i emits a spike:

$w_{ij} \leftarrow w_{ij} + \eta_r \cdot \frac{x}{\gamma_{LI}} \bar{\rho}_j$

Return: Trained weights for input image: W_{ij} and go to next image/next epoch.

Wan et al., 2020). The corresponding pseudocode is given in Algorithm 1 (see supplemental information for details).

Full network simulations: recognition rate on handwritten digit database

We now evaluate the performance of EqSpike on the Mixed National Institute of Standards and Technology MNIST handwritten digit classification task, using a fully connected network with one hidden layer (see Supplementary Information for details). The obtained train (orange) and test (blue) accuracies are shown Figure 2 as a function of the number of training epochs, with the deviation over six runs in shadow color. Table 1 compares the results to BPTT and the version of equilibrium propagation closest to our implementation, called continual Eq-Prop (Ernault et al., 2020), trained with a batch size of one.

The test accuracy of EqSpike matches closely the accuracy of stochastic gradient descent through BPTT on the same network architecture, given the error margin. With a hidden layer of 300 neurons, EqSpikes reaches a test accuracy of 97.59%. Fully connected spiking neural networks trained on MNIST without conversion from a non-spiking neural network typically achieve recognition rates in the 96–98% range (Nefcici et al., 2017; O’Connor and Welling, 2016; Lee et al., 2016; Mostafa, 2018; Tavanaei and Maida, 2019). EqSpike with its local two-factor online learning rule therefore reaches accuracies on MNIST comparable to those of the latest models investigated for training spiking neural networks on hardware platforms. We have chosen the MNIST data set as a benchmark because it is a standard data set for the neuromorphic community interested in training spiking neural networks online, due to the long simulation times. As EqSpike achieves results on MNIST equivalent to the baseline given by equilibrium propagation, it has the potential, like equilibrium propagation, to adapt to convolutional architectures and perform with good accuracy on more complex image benchmarks (Laborieux et al., 2021), with the additional advantage of being compatible with current neuromorphic technologies.

Inference speed and energy

As EqSpike is derived from a rate-based algorithm, it is interesting for neuromorphic applications to quantify the number of spikes needed to achieve inference and the time needed to reach high accuracy. Operation with fewer spikes is more desirable, as it reduces both execution time and energy consumption.

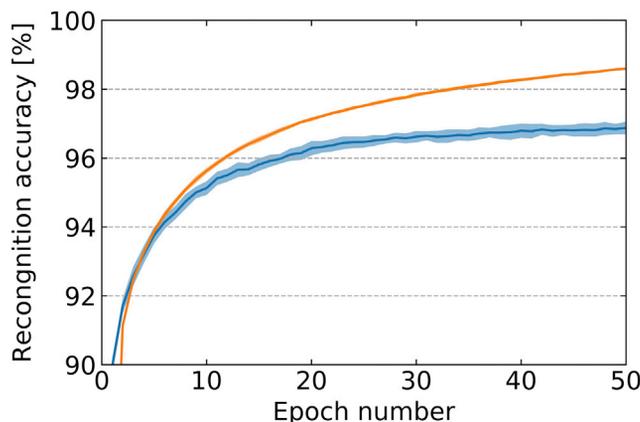


Figure 2. Recognition accuracy during training (orange) and test (blue) as a function of the number of epochs for MNIST, averaged over six runs. The error bars, in light color, correspond to the standard deviation over the six runs.

Figure 3 shows the inference results as a function of the execution time multiplied by the maximum frequency of neurons ($t \times f_{max}$). The orange line is the mean accuracy result over the whole test data set, obtained by computing the spike rate of output neurons (as done for Figure 2 and Table 1), through averaging in a time window $T_{average}$ of 100 simulation time steps ($T_{average} = 50/f_{max}$) and considering that the neuron with highest frequency encodes the output. This method computes the rate accurately at the expense of having to wait the time $T_{average}$ and letting output neurons spike multiple times. Spiking neural networks also offer the possibility to accelerate the computation and reduce the energy consumption by determining the output class from the first output neuron to spike. The blue line in Figure 3 is the accuracy as a function of time, averaged over all images in the test data set, obtained by considering that the first output neuron to spike encodes the output.

The red, vertical dotted line in Figure 3 indicates the average time of the first spike at the output over all presented images, corresponding to $t \cong 3.5/f_{max}$. At $t \cong 10/f_{max}$, the accuracy of single-spike inference (blue curve) reaches $95.11\% \pm 0.78\%$, within 1.4% of the precise rate computation (orange curve). This result shows that even though the algorithm is originally rate based, a single spike at the output suffices in most cases to determine the correct class with good precision, a feature which is highly attractive for energy-efficient inference on neuromorphic chips. It means that inference can be achieved in $100 \mu s$ for electronic neurons with a firing rate of 100 kHz, available in neuromorphic chips working in accelerated time compared to biology (Schemmel et al., 2010), and $1 \mu s$ for electronic neurons with a firing rate of 10 MHz that can be produced, for example, with emerging nanotechnologies (Li et al., 2015). The corresponding throughputs are, respectively, 10k and 1M images/s, on par with current spiking neural network implementations (Pfeiffer and Pfeil, 2018; Park et al., 2020). As the network operations are fully parallel, these orders of magnitudes will be conserved for wider networks. Simulations of rate-based equilibrium propagation on deeper networks indicate that the convergence time increases by a factor of about eight for a network with four hidden layers compared to a network of one hidden layer as here (Laborieux et al., 2021). It should be noted that in the current implementation we present static inputs to the network, which means that input neurons need to integrate these signals before they emit the first spikes that will then propagate to the next layers. The speed of inference could be increased in the future by presenting inputs directly encoded in spikes, for example, sourced from neuromorphic vision sensors (Pfeiffer and Pfeil, 2018).

An estimation of the energy consumption of a spiking neural network on a neuromorphic silicon chip can be performed by counting the number of synaptic operations (SynOps) involved. SynOps are defined as the total number of spikes transiting through synapses of the network. Frenkel et al. show that a SynOp on a neuromorphic chip requires as little as 10 pJ (Frenkel et al., 2019). The total number of SynOps needed for inference depends on the targeted recognition precision and, therefore, on the duration of inference (Figure 3). For EqSpike, the recognition rate saturates at $t \cong 10/f_{max}$. The corresponding measured number of SynOps is about 150,000 in average. This is much less than expected if all neurons spiked. This is also a bit

Table 1. Comparison between BPTT, C-EP, and EqSpike, with the same initialization procedure

Algorithm	BPTT 784-100-10	Continual Eq-Prop 784-100-10	EqSpike 100 784-100-10	EqSpike 300 784-300-10
MNIST	Test: 97.11% ± 0.23% Train: 99.06% ± 0.15%	Test: 96.97% ± 0.12% Train: 99.8% ± 0.04%	Test: 96.87% ± 0.18% Train: 98.59% ± 0.03%	Test: 97.59% ± 0.1% Train: 98.91% ± 0.03%

Batch size = 1, average on 6 runs.

less but comparable to the number of SynOps needed at inference for event-driven random backpropagation (Neftci et al., 2017). Considering 10 pJ/SynOps, each EqSpike inference could potentially consume 1.5 μ J. This means that testing the 10,000 images of the whole MNIST data set could be achieved with a neuromorphic chip while consuming only 15 mJ, in other words, three orders of magnitude less than with a GPU (Joseph and Nagarajan, 2020).

In our current EqSpike implementation, the input layer is the one leading to most spikes and SynOps: with only 16% of illuminated pixels in average in MNIST, the input layer emits 87.5% of all spikes and 98.6% of SynOps occur between the input layer and the hidden layer. In this work, we did not focus on reducing the number of spikes with encoding, but a better encoding of the input may reduce considerably the energy consumption. Kheradpisheh et al. have shown that with a temporal encoding, the total number of spikes in the network before the first output spike can be reduced to 200 with a hidden layer four times larger than our network (Kheradpisheh and Masquelier, 2020). In our case, 678 spikes in total are emitted in average before the first output spike. An adaptation of EqSpike to temporal encoding is not straightforward, but this number could potentially be decreased in the future by reducing the encoding frequency of the input.

Training speed and energy

Training with EqSpike requires performing the free phase and then the nudging phase, during which synaptic weights are updated. A way to speedup the training and reduce the total number of SynOps is to perform the nudging phase only on poorly classified examples and skip the updates when the accuracy is satisfactory. We apply this strategy inspired from (Park et al., 2020) using the criteria that the nudging phase is performed only when the difference between the target rate and the actual rate ($\widehat{\rho}_k - \rho_k$) at the output is above 1%. Figure 4A shows the number of presented examples per epoch as a function of epoch number. In the last 20 epochs, only approximately 15% of the training data set still requires a nudging phase.

For MNIST, we thus perform the free phase on all the data sets (3×10^6 images for the 50 epochs) and the nudging phase on 489,000 images. Given the durations of each phase, we can estimate the training time to $T_{\text{training}} \cong 2.74 \times 10^8 / f_{\text{max}}$. For electronic neurons with a firing rate of 100 kHz (Schemmel et al., 2010), this leads to $T_{\text{training}} \cong 45$ min, and for electronic neurons with a firing rate of 10 MHz (Li et al., 2015), this leads to $T_{\text{training}} \cong 30$ s. As our networks feature a fully parallel nature, these training times would be the same for much wider networks and increased by a factor of about eight only with four hidden layers (Laborieux et al., 2021).

As EqSpike is derived from a rate-based approach, it is interesting to compare the actual spiking rates of neurons in the network during training to their maximum frequency f_{max} . For neuromorphic applications, low overall rates are indeed desirable. Figure 4B shows the average number of spikes emitted by each neuron for an image presentation in the training data set, as a function of the epoch. We found that for the training conditions of Figure 2, there are in average 36 spikes/neurons/images. This means that neurons in the network spike in average with a frequency of the order of 20% of f_{max} , well below f_{max} , which is promising for neuromorphic implementations. Again, this number could be reduced in the future by optimizing the encoding of input at the first layer.

Figure 4C shows the numbers of SynOps needed for training as a function of recognition rate. The total number of SynOps after 50 epochs is of 4.23×10^{12} , which is of the same order of magnitude as event-driven random backpropagation (Neftci et al., 2017) for similar accuracy, and below training MNIST with BP based on (14). With 10pJ per SynOps (Frenkel et al., 2019), the training phase of EqSpike on a neuromorphic

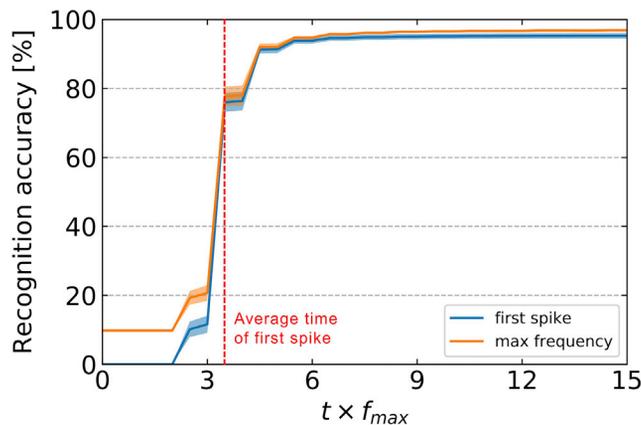


Figure 3. Inference time: recognition accuracy on MNIST on the test data set as a function of time multiplied by the maximum neuron frequency f_{max}

Orange line: recognition accuracy computed from the output neuron showing the highest rate. Blue line: recognition accuracy computed from the output neuron spiking first. Red, vertical dotted line: average time of first output spike. The width of the lines in light colors correspond to the standard deviation over the different runs.

chip could consume as little as 42 J, again, two orders of magnitude less than with a GPU (Joseph and Nagarajan, 2020).

Spike-timing-dependent plasticity

We have shown that EqSpike transforms equilibrium propagation into an efficient algorithm for neuromorphic chips. We now highlight that it also brings equilibrium propagation closer to biological plausibility. Bengio et al. have pointed out a connection between the equilibrium propagation learning rule of Equation 1 and STDP (Bengio et al., 2017). The STDP learning rule, illustrated in Figure 5A, reinforces causality between the spikes of pre- and post-synaptic neurons in networks with unidirectional synapses. If the post-synaptic neuron spikes after the pre-synaptic neurons, causality is observed, and the weight is increased. In the opposite scenario, the weight is decreased.

Let us consider a situation where the pre-synaptic neuron i spikes and the post-synaptic neuron j accelerates, as illustrated in Figure 5B. According to the Eq-Prop learning rule, in a network with unidirectional synapses, $\frac{dW_{ji}}{dt} \propto \dot{\rho}_j \rho_i = \dot{\rho}_{post} \rho_{pre}$, a positive weight update should be applied. Due to the acceleration of the post-neuron, there are less post-neuron spikes before the pre-neuron spike than after. Therefore, $t_{post} - t_{pre}$ is positive in average, yielding a positive weight update through STDP.

We have investigated if STDP-like weight updates did emerge during learning in our simulations. For this purpose, we monitored weight variations in synapses that connect input neurons and the hidden layer neurons. These synapses are unidirectional as input neurons are clamped to the input (see Figure 1): their frequency does not vary. We used 100 images during the first epoch for the MNIST data set. The curve in Figure 5C shows the average weight updates in the first layer as a function of the time difference between post-synaptic neuron spikes and the average time of pre-synaptic neuron spikes in a window of 200 time steps before the post-synaptic neuron spike. The obtained curve, centered on zero, indeed exhibits an STDP-like shape. It has been obtained by filtering out very low frequencies below 0.05. Quiet neurons in the free phase indeed induce large weight updates at the beginning of the nudging phase, due to the sudden acceleration from zero to non-zero frequency, inducing an additional noise in the curve. It should be noted that biological STDP curves being frequency dependent, they are also often obtained by focusing on a given range of frequencies (Kirkwood et al., 1996). Figure 5C shows that the STDP amplitude and time window vary with the strength of nudging β . In equilibrium propagation, weight changes are driven by neuron accelerations. As nudging applies a force that accelerates neurons, for larger β , larger weight changes are obtained, leading to a larger amplitude of the STDP-like curve. As higher accelerations are reached during the nudging phase, noticeable weight modifications occur for larger differences between t_{post} and t_{pre} , corresponding to a wider time window for the STDP-like curve. These results confirm the possible connection between STDP and equilibrium propagation pointed out in (Bengio et al., 2017),

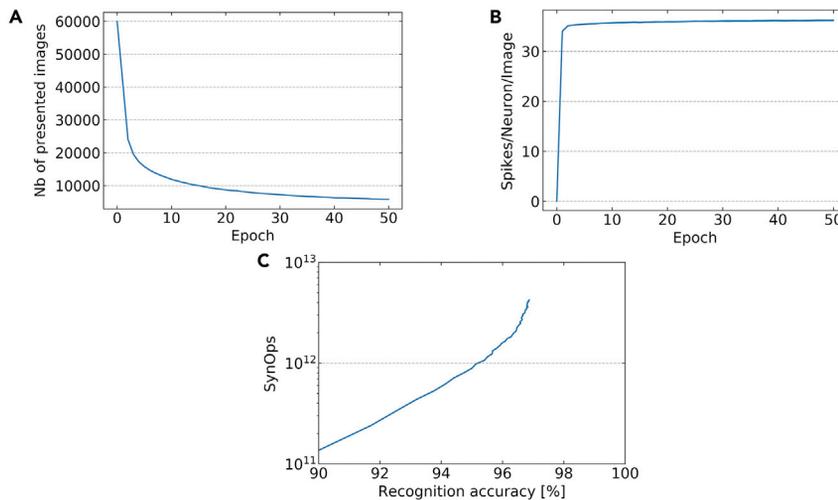


Figure 4. Training performance

(A) Number of presented images in the nudging phase per epoch versus epoch number.
 (B) Number of spikes/neuron/image occurring during the two phases (nudge + free), as a function of the epoch.
 (C) SynOps: number of spikes during both phases (nudge + free) as a function of the recognition accuracy.

despite the fact that individual spike timings are lost through averaging. They show that STDP-like behavior can be obtained during learning in unidirectional synapses without a direct implementation of the original, causality-based rule. They also ask the question whether multilayer networks could be trained with the local STDP learning rule by using the nudging procedure of equilibrium propagation, possibly in the extended version called “vector-field equilibrium propagation” in which synapses are unidirectional (Scellier et al., 2018).

DISCUSSION

Other versions of equilibrium propagation have been specifically designed to train spiking neural networks. O’Connor et al. have designed a spiking network which, by construction, stochastically approximates the dynamics of its rate-based counterpart until reaching the same steady state with a minimal spike communication budget between neurons (O’Connor et al., 2019). Their technique, successfully tested against MNIST, comes at the cost of combining predictive coding, sigma-delta modulation, and adaptive step sizes at the neuron level. In comparison, our technique simply requires LIF neurons, a spike count, and a low-pass filter to implement the learning rule. Mesnard et al. proposed a version of equilibrium propagation to train spiking networks that also makes use of low-pass filters to estimate firing rates. However, they only demonstrate their approach on a non-linear toy problem, and the implemented learning rule is not local in time (Mesnard et al., 2016). In contrast, we demonstrate the effectiveness of our fully event-based implementation on MNIST.

More generally, several spike-based approaches to backpropagation have been proposed. One method consists in smoothing the spikes as a function of time so that gradients can be back-propagated through time (Huh and Sejnowski, 2018; Bellec et al., 2020). Another technique consists in gating the spikes propagating the error signals by surrogate derivatives, as done in SpikeGrad (Thiele et al., 2019). Finally, event-driven random backpropagation uses firing rates in the backward pass, as we do in this paper, and achieves similar performance on MNIST (Neftci et al., 2017). In comparison with these approaches, however, our implementation of equilibrium propagation does not require to know the activation function equation in order to compute its derivatives. One other interesting approach is S4NN, which employs latency coding, where each neuron can spike at most once, and the output is encoded as the first neuron to spike (Kheradpisheh and Masquelier, 2020). In contrast with rate-based coding, latency coding has the potential to save important energy in neuromorphic implementations. However, there is no clear indication of whether S4NN could scale to harder visual tasks, while rate-based equilibrium propagation was shown to train deep ConvNets on CIFAR-10 (Laborieux et al., 2021). One intrinsic limitation of EqSpike, however, may stem from the necessity for the system to reach a steady state before the gradient computation phase. This prevents,

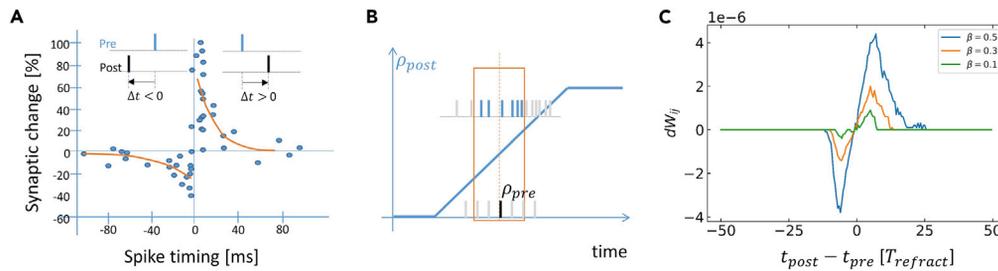


Figure 5. STDP

(A) Illustration of the STDP learning rule, reproduced with data from (Bi and Poo, 2001).

(B) Illustration of the link between Eq-Prop and STDP learning rules; illustration reproduced from (Bengio et al., 2017).

(C) STDP-like curve during EqSpike learning.

for now, the classification of time-varying inputs. Payeur et al. recently proposed a spike-based approach where top-down error signals are encoded as spike bursts and are multiplexed with bottom-up feedforward signals so that the backward pass and the forward pass can occur simultaneously with minimal disruption (Payeur et al., 2020). While they show that the rate-based counterpart of their algorithm works on CIFAR-10 and ImageNet, it comes at the cost of employing dendritic network topologies and specialized synapses. Indeed, as noted in the introduction, most backpropagation-derived local learning rules involve a third factor for supervision (Payvand et al., 2020). In this regard, we believe that our EqSpike implementation of equilibrium propagation, with only two factors, achieves an optimal trade-off between circuitry complexity and performance.

Finally, Zoppo et al. (Zoppo et al., 2020) and Kendall et al. (Kendall et al., 2020) have proposed using equilibrium propagation for training neuromorphic hardware. However, their implementations remain either rate based or current based and are therefore not directly compatible with spiking neuromorphic chips. EqSpike, on the other hand, could be trained directly on reconfigurable neuromorphic systems with online learning such as SpiNNaker (Furber et al., 2014) and Loihi (Davies et al., 2018).

Conclusion

In this work, we present a new algorithm for spiking neural networks, EqSpike, compatible with neuromorphic systems, and achieving good performance on MNIST. We show that EqSpike implements the learning rule of equilibrium propagation locally and autonomously. The gradients are computed by the dynamics of the system, and the weights are modified by a spike and the addition of only one block to the neuron. This can lead to spiking neuromorphic systems that do not need an external circuit to compute the error gradients given by backpropagation and learn autonomously, simply by presenting inputs and nudging the outputs according to errors. Our method obtained results on MNIST close to backpropagation through time and equilibrium propagation, two state-of-the-art algorithms. Moreover, because EqSpike is based on equilibrium propagation, the performance on more complex task, like CIFAR-10, could be similar. The number of SynOps to obtain these results in MNIST shows we can obtain theoretically two orders of magnitude less energy consumption than a GPU for training and the same magnitude of time with high frequency neurons. The inference, after training the network, can be accelerated by waiting for the first output spike rather than compute the highest rate, with only a small loss of accuracy.

Finally, we show that the weight updates of EqSpike share similarity with STDP during learning, raising the question of a possible biological plausibility of the algorithm. In average, the modification of weight is proportional to the spike timing. This could permit to implement synapses in neuromorphic hardware by emergent nano-devices with an STDP-like behavior to obtain a lower power consumption and higher surface density.

Limitations of the study

A limitation of EqSpike is that, with a wrong choice of γ_{\perp} or τ combined with low nudging times, fast or brief rate changes cannot be detected, and the corresponding weight modification cannot be applied. This does not seem to be a problem with our experimentations as demonstrated above but could lead to a lower accuracy on more complex problems.

The circuits to compute the rate derivative could be further miniaturized by extracting $\dot{\rho}$ directly from the membrane potential if neuron models with smoothly varying membrane potential are used (Gerstner et al., 2014).

EqSpike could also be sped up by building on dedicated hardware in analog or digital CMOS (Schemmel et al., 2010; Qiao et al., 2015; Thakur et al., 2018; Frenkel et al., 2019; Park et al., 2020). Emerging nanotechnologies such as memristive synapses and nanoscale spiking oscillators are compelling candidates to scale up neuromorphic hardware due to their small size, their speed, and their low energy consumption (Marković et al., 2020; Milo et al., 2020; Sebastian et al., 2020; Wang et al., 2020; Xi et al., 2020). These technologies are typically prone to imperfections such as the device-to-device variability, cycle-to-cycle variability, or the non-linearity in the conductance-to-voltage response, which are known to considerably jeopardize learning in memristive neural networks (Ishii et al., 2019; Zhang et al., 2020). Our paper implicitly assumes that the underlying memory technology at use would be linear, deterministic, and identical across different synapses. The learning rate l_r in EqSpike is indeed about 10 times smaller than typically used in standard gradient descent. This is due to the fact that the batch size is one and also that a low l_r is needed to avoid too large modifications accumulated over all the nudging phase. For low precision device as synapse, a first possible solution is to update the synaptic devices less frequently (not at each spike) but with a higher value. Another strategy is to adapt to EqSpike the training schemes used for binary neural networks (Hubara et al., 2016; Rastegari et al., 2016; Hirtzlin et al., 2019). Further study should be done to propose a fully end-to-end circuit to implement EqSpike and investigate its resilience to the memristive device imperfections mentioned.

Resource availability

Lead contact

Further information and requests for resources and reagents should be directed to and will be fulfilled by the lead contact, Julie Grollier (julie.grollier@cnsr-thales.fr).

Material availability

N.A.

Data and code availability

The data generated by this study are available on reasonable request.

METHODS

All methods can be found in the accompanying [transparent methods supplemental file](#).

SUPPLEMENTAL INFORMATION

Supplemental information can be found online at <https://doi.org/10.1016/j.isci.2021.102222>.

ACKNOWLEDGMENTS

This publication has received funding from the European Union's Horizon 2020 research innovation program under grant agreement 732642 (ULPEC project) and 876925 (ANDANTE project). E.M. received funding from the ANRT grant 2018/0884.

AUTHOR CONTRIBUTIONS

J.G., T.P., and D.Q. have devised and supervised the study; E.M. has performed the spiking simulations with the help of M.E., J.L., and S.L. M.E. and J.L. have performed the benchmark rate-based simulations. All authors have contributed to writing the article.

DECLARATION OF INTERESTS

The authors declare no competing interests.

Received: October 15, 2020

Revised: January 15, 2021

Accepted: February 18, 2021

Published: March 19, 2021

REFERENCES

- Bellec, G., Scherr, F., Subramoney, A., Hajek, E., Salaj, D., Legenstein, R., and Maass, W. (2020). A solution to the learning dilemma for recurrent networks of spiking neurons. *Nat. Commun.* 11, 3625.
- Bengio, Y., Mesnard, T., Fischer, A., Zhang, S., and Wu, Y. (2017). STDP-compatible approximation of backpropagation in an energy-based model. *Neural Comput.* 29, 555–577.
- Bi, G., and Poo, M. (2001). Synaptic modification by correlated activity: hebb's postulate revisited. *Annu. Rev. Neurosci.* 24, 139–166.
- Bichler, O., Querlioz, D., Thorpe, S., J., Bourgoin, J.-P., and Gamrat, C. (2012). Extraction of temporally correlated features from dynamic vision sensors with spike-timing-dependent plasticity. *Neural Networks* 32, 339–348.
- Davies, M., Srinivasa, N., Lin, T.-H., China, G., Cao, Y., Choday, S., H., Dimou, G., Joshi, P., Imam, N., Jain, S., et al. (2018). Loihi: a neuromorphic manycore processor with on-chip learning. *IEEE Micro* 38, 82–99.
- Ernault, M., Grollier, J., Querlioz, D., Bengio, Y., and Scellier, B. (2019). Updates of equilibrium prop match gradients of backprop through time in an RNN with static input. In *Advances in Neural Information Processing Systems* 32, H. Wallach, et al., eds. (Curran Associates, Inc.), pp. 7081–7091.
- Ernault, M., Grollier, J., Querlioz, D., Bengio, Y., and Scellier, B. (2020). Equilibrium Propagation with Continual Weight Updates. *arXiv:2005.04168*.
- Falez, P., Tirilly, P., Bilasco, Ioan Marius, Devienne, P., and Boulet, P. (2019). Unsupervised visual feature learning with spike-timing-dependent plasticity: how far are we from traditional feature learning approaches? *Pattern Recognit.* 93, 418–429.
- Feldmann, J., Youngblood, N., Wright, C., D., Bhaskaran, H., and Pernice, W., H. P. (2019). All-optical spiking neuromorphic networks with self-learning capabilities. *Nature* 569, 208–214.
- Frenkel, C., Lefebvre, M., Legat, J.-D., and Bol, D. (2019). A 0.086-mm² 12.7-pJ/SOP 64k-synapse 256-neuron online-learning digital spiking neuromorphic processor in 28-nm CMOS. *IEEE Trans. Biomed. Circuits Syst.* 13 (1), 145–158.
- Furber, S., B., Galluppi, F., Temple, S., and Plana, L., A. (2014). The SpiNNaker project. *Proc. IEEE* 102, 652–665.
- Gerstner, W., et al. (2014). *Neuronal Dynamics: From Single Neurons to Networks and Models of Cognition*, First Edition (Cambridge University Press).
- Hirtzlin, T., Bocquet, M., Ernault, M., Klein, J.-O., Nowak, E., Vianello, V., Portal, J. - M., and Querlioz, D. (2019). Hybrid analog-digital learning with differential RRAM synapses. In *2019 IEEE International Electron Devices Meeting (IEDM)*. 2019 IEEE International Electron Devices Meeting (IEDM) (IEEE), pp. 22.6.1–22.6.4.
- Hubara, I., Courbariaux, M., Soudry, D., El-Yaniv, R., and Bengio, Y. (2016). Binarized neural networks. In *Advances in Neural Information Processing Systems* 29, D.D. Lee, et al., eds. (Curran Associates, Inc.), pp. 4107–4115.
- Huh, D., and Sejnowski, T.J. (2018). Gradient descent for spiking neural networks. In *Advances in Neural Information Processing Systems* 31, S. Bengio, et al., eds. (Curran Associates, Inc.), pp. 1433–1443.
- Ishii, M., Kim, S., Lewis, S., Okazaki, A., Okazawa, J., Ito, M., Rasch, M., Kim, W., Nomura, A., Shin, U., et al. (2019). On-chip trainable 1.4M 6T2R PCM synaptic array with 1.6K stochastic LIF neurons for spiking RBM. In *2019 IEEE International Electron Devices Meeting (IEDM)* (2019 IEEE International Electron Devices Meeting (IEDM)), pp. 14.2.1–14.2.4.
- Jo, S.H., Chang, T., Ebong, I., Bhadviya, B.B., Mazumder, P., and Lu, W. (2010). Nanoscale memristor device as synapse in neuromorphic systems. *Nano Lett.* 10, 1297–1301.
- Kaiser, J., Mostafa, H., and Neftci, E. (2020). Synaptic plasticity dynamics for deep continuous local learning (DECOLLE). *Front. Neurosci.* 14, 424.
- Kendall, J., Pantone, R., Manickavasagam, K., Bengio, Y., and Scellier, B. (2020). Training end-to-end analog neural networks with equilibrium propagation. *arXiv:2006.01981*.
- Kheradpisheh, S.R., and Masquelier, T. (2020). Temporal backpropagation for spiking neural networks with one spike per neuron. *Int. J. Neural Syst.* 30, 2050027.
- Kirkwood, A., Rioult, M.G., and Bear, M.F. (1996). Experience-dependent modification of synaptic plasticity in visual cortex. *Nature* 381, 526–528.
- Laborieux, A., Ernault, M., Scellier, B., Bengio, Y., Grollier, J., and Querlioz, D. (2021). Scaling equilibrium propagation to deep ConvNets by drastically reducing its gradient estimator bias. *Frontiers in Neuroscience* 15, 129.
- Lee, J.H., Delbruck, T., and Pfeiffer, M. (2016). Training deep spiking neural networks using backpropagation. *Front. Neurosci.* 10, 508.
- Li, S., Liu, X., Nandi, S.K., Venkatachalam, D.K., and Elliman, R., G. (2015). High-endurance megahertz electrical self-oscillation in Ti/NbOx bilayer structures. *Appl. Phys. Lett.* 106, 212902.
- Marković, D., Mizrahi, A., Querlioz, D., and Grollier, J. (2020). Physics for neuromorphic computing. *Nat. Rev. Phys.* 1–12.
- Masquelier, T., and Thorpe, S.J. (2007). Unsupervised learning of visual features through spike timing dependent plasticity. *PLoS Comput. Biol.* 3, e31.
- C. Mead, and M. Ismail, eds. (1989). *Analog VLSI Implementation of Neural Systems* (Springer US (The Springer International Series in Engineering and Computer Science)).
- Merolla, P.A., Arthur, J.V., Alvarez-Icaza, R., Cassidy, A.S., Sawada, J., Akopyan, F., Jackson, B.L., Imam, N., Guo, C., Nakamura, Y., et al. (2014). A million spiking-neuron integrated circuit with a scalable communication network and interface. *Science* 345, 668–673.
- Mesnard, T., Gerstner, W., and Brea, J. (2016). Towards deep learning with spiking neurons in energy based models with contrastive Hebbian plasticity. *arXiv:1612.03214*.
- Milo, V., Malavena, G., Monzio Compagnoni, C., and Ielmini, D. (2020). Memristive and CMOS devices for neuromorphic computing. *Materials* 13, 166.
- Mostafa, H. (2018). Supervised learning based on temporal coding in spiking neural networks. *IEEE Trans. Neural Networks Learn. Syst.* 29 (7), 3227–3235.
- Navarro, M.A., Salari, A., Lin, J.L., Cowan, L.M., Penington, N.J., Milesco, M., and Milesco, L.M. (2020). Sodium channels implement a molecular leaky integrator that detects action potentials and regulates neuronal firing. *eLife* 9, e54940.
- Neftci, E.O., Augustine, C., Paul, S., and Detorakis, G. (2017). Event-driven random backpropagation: enabling neuromorphic deep learning machines. *Front. Neurosci.* 11, 1621–1671.
- Neftci, E.O., Mostafa, H., and Zenke, F. (2019). Surrogate gradient learning in spiking neural networks: bringing the power of gradient-based optimization to spiking neural networks. *IEEE Signal Process. Mag.* 36, 51–63.
- O'Connor, P., Gavves, E., and Welling, M. (2019). Training a spiking neural network with equilibrium propagation. In *The 22nd International Conference on Artificial Intelligence and Statistics. The 22nd International Conference on Artificial Intelligence and Statistics (PMLR)*, pp. 1516–1523.
- O'Connor, P., and Welling, M. (2016). Deep spiking networks. *arXiv:1602.08323*.
- Park, J., Lee, J., and Jeon, D. (2020). A 65-nm neuromorphic image classification processor with energy-efficient training through direct spike-only feedback. *IEEE J. Solid-State Circuits* 55, 108–119.
- Payeur, A., Guerguiev, J., Zenke, F., Richards, B.A., and Naud, R. (2020). Burst-dependent synaptic plasticity can coordinate learning in hierarchical circuits. *bioRxiv*. <https://doi.org/10.1101/2020.03.30.015511>.
- Payvand, M., Fouda, M.E., Kurdahi, F., Eltawil, A., and Neftci, E.O. (2020). Error-triggered three-factor learning dynamics for crossbar arrays. In *2020 2nd IEEE International Conference on Artificial Intelligence Circuits and Systems (AICAS)* (2020 2nd IEEE International Conference on Artificial Intelligence Circuits and Systems (AICAS)), pp. 218–222.
- Pedretti, G., Milo, V., Ambrogio, S., Carboni, R., Bianchi, S., Calderoni, A., Ramaswamy, N., Spinelli, A.S., and Ielmini, D. (2017). Memristive neural network for on-line learning and tracking with brain-inspired spike timing dependent plasticity. *Sci. Rep.* 7, 5288.

- Pfeiffer, M., and Pfeil, T. (2018). Deep learning with spiking neurons: opportunities and challenges. *Front. Neurosci.* 12, 126.
- Prezioso, M., Mahmoodi, M.R., Merrikh Bayat, F., Nili, H., Kim, H., Vincent, A., and Strukov, D.B. (2018). Spike-timing-dependent plasticity learning of coincidence detection with passively integrated memristive circuits. *Nat. Commun.* 9, 5311.
- Qiao, N., Mostafa, H., Corradi, F., Osswald, M., Stefanini, F., Sumislawska, D., and Indiveri, G. (2015). A reconfigurable on-line learning spiking neuromorphic processor comprising 256 neurons and 128K synapses. *Front. Neurosci.* 9, 141.
- Rastegari, M., Ordonez, V., Redmon, J., and Farhadi, A. (2016). XNOR-net: ImageNet classification using binary convolutional neural networks. In *Computer Vision – ECCV 2016*, B. Leibe, et al., eds. (Springer International Publishing (Lecture Notes in Computer Science)), pp. 525–542.
- Richards, B.A., Lillicrap, T.P., Beaudoin, P., Bengio, Y., Bogacz, R., Christensen, A., Clopath, C., Ponte Costa, R., de Berker, A., Ganguli, S., et al. (2019). A deep learning framework for neuroscience. *Nat. Neurosci.* 22, 1761–1770.
- Sacramento, J., Ponte Costa, R., Bengio, Y., and Senn, W. (2018). Dendritic cortical microcircuits approximate the backpropagation algorithm. In *Advances in Neural Information Processing Systems*, 31, S. Bengio, et al., eds. (Curran Associates, Inc.), pp. 8721–8732.
- Scellier, B., and Bengio, Y. (2017). Equilibrium propagation: bridging the gap between energy-based models and backpropagation. *Front. Comput. Neurosci.* 11, 24.
- Scellier, B., Goyal, A., Binas, J., Mesnard, T., and Bengio, Y. (2018). Generalization of Equilibrium Propagation to Vector Field Dynamics. *arXiv*, arXiv:1808.04873.
- Schemmel, J., Brüderle, D., Grübl, A., Hock, M., Meier, K., and Millner, S. (2010). A wafer-scale neuromorphic hardware system for large-scale neural modeling. In *Proceedings of 2010 IEEE International Symposium on Circuits and Systems (Proceedings of 2010 IEEE International Symposium on Circuits and Systems)*, pp. 1947–1950.
- Sebastian, A., Le Gallo, M., Khaddam-Aljameh, R., and Eleftheriou, E. (2020). Memory devices and applications for in-memory computing. *Nat. Nanotechnology* 15, 1–16.
- Serb, A., Bill, J., Khiat, A., Berdan, R., Legenstein, R., and Prodromakis, T. (2016). Unsupervised learning in probabilistic neural networks with multi-state metal-oxide memristive synapses. *Nat. Commun.* 7, ncomms12611.
- Tavanaei, A., and Maida, A. (2019). BP-STDP: approximating backpropagation using spike timing dependent plasticity. *Neurocomputing* 330, 39–47.
- Thakur, C.S., Lottier Molin, J., Cauwenberghs, G., Indiveri, G., Kumar, K., Qiao, N., Schemmel, J., Wang, R., Chicca, E., and Hasler, J.O. (2018). Large-scale neuromorphic spiking array processors: a quest to mimic the brain. *Front. Neurosci.* 12, 991.
- Thiele, J.C., Bichler, O., and Dupret, A. (2019). SpikeGrad: an ANN-equivalent computation model for implementing backpropagation with spikes. *arXiv*, arXiv:1906.00851.
- Wan, W., Kubendran, R., Burc Eryilmaz, S., Zhang, W., Liao, Y., Wu, D., Deiss, S., Gao, B., Raina, P., Joshi, S., et al. (2020). 33.1 A 74 TMACS/W CMOS-RRAM neurosynaptic core with dynamically reconfigurable dataflow and in-situ transposable weights for probabilistic graphical models. In *2020 IEEE International Solid-State Circuits Conference - (ISSCC). 2020 IEEE International Solid-State Circuits Conference - (ISSCC)*, pp. 498–500.
- Wang, Z., Wu, H., Burr, G.W., Hwang, C.S., Wang, K.L., Xia, Q., and Yang, J.J. (2020). Resistive switching materials for information processing. *Nat. Rev. Mater.* 5, 173–195.
- Xi, Y., Gao, B., Tang, J., Chen, A., Chang, M.-F., Hu, X.S., Van Der Spiegel, J., Qian, H., and Wu, H. (2020). In-memory learning with analog resistive switching memory: a review and perspective. *Proc. IEEE* 109, 1–29.
- Zamarreño-Ramos, C., Camuñas-Mesa, L.A., Pérez-Carrasco, J.A., Masquelier, T., Serrano-Gotarredona, T., and Linares-Barranco, B. (2011). On spike-timing-dependent-plasticity, memristive devices, and building a self-learning visual cortex. *Front. Neurosci.* 5, 26.
- Zhang, W., Gao, B., Tang, J., Yao, P., Yu, S., Chang, M.-F., Yoo, H.-J., Qian, H., and Wu, H. (2020). Neuro-inspired computing chips. *Nat. Electronics* 3, 371–382.
- Zoppo, G., Marrone, F., and Corinto, F. (2020). Equilibrium propagation for memristor-based recurrent neural networks. *Front. Neurosci.* 14, 240.
- Joseph, V. and Nagarajan, C. (2020) MADONNA: A Framework for Energy Measurements and Assistance in Designing Low Power Deep Neural Networks. p. 7. <http://www.cs.utah.edu/~vinu/madonna.pdf>

iScience, Volume 24

Supplemental information

**EqSpike: spike-driven
equilibrium propagation
for neuromorphic implementations**

Erwann Martin, Maxence Ernout, Jérémie Laydevant, Shuai Li, Damien Querlioz, Teodora Petrisor, and Julie Grollier

Supplementary information for:

EqSpike: Spike-driven Equilibrium Propagation for Neuromorphic Implementations

Erwann Martin¹, Maxence Ernoult^{2,3}, Jérémie Laydevant², Shuai Li², Damien Querlioz³, Teodora Petrisor¹, Julie Grollier²

¹ - Thales Research and Technology, 91767 Palaiseau, France

² - Unité Mixte de Physique, CNRS, Thales, Université Paris-Saclay, 91767 Palaiseau, France

³ - Université Paris-Saclay, CNRS, Centre de Nanosciences et de Nanotechnologies, 91120 Palaiseau, France

Transparent Methods:

Spiking neurons

We use leaky-integrate-and-fire (LIF) neurons whose membrane potentials follow the standard equation (34):

$$\frac{du_i}{dt} = -\gamma_{LIF} u_i + I_i(t),$$

where γ_{LIF} is the leak factor and $I_i(t)$ the input signal of neuron i at time t :

$I_i(t) = \sum_{j=1}^{N_{neuron}} W_{ij} \delta(t - t_j)$, where δ is the Dirac function and t_j are the times at which neurons j spike.

The neuron integrates input signals until its membrane potential u_i overcomes a threshold value u_{th} . At that moment, the neuron declares a spike, the threshold value u_{th} is subtracted from u_i , and the neuron undergoes a refractory period of duration $T_{refract}$.

Simulation parameters

The values of the different parameters used in simulations are listed in Table 1 of the, where time is expressed in units of $T_{refract}$.

Simulation timestep, dt ($T_{refract}$)	γ_{LIF}	γ_{LI}	u_{th}	τ ($T_{refract}$)	η_r	N_{filt}	β	T_{free} ($T_{refract}$)	T_{nudge} ($T_{refract}$)
0.5	0.01	0.1	1	50	3×10^{-6}	10	0.5	75	100

Table S1. - Simulation parameters, related to all figures and tables in the article.

Algorithm

In the EqSpike algorithm, the rates need to be locally stored for the duration τ , in order to obtain the rate derivative, with τ smaller than the duration of both the free and nudge phases. This is why the neuron rates, ρ , are computed even during the free phase. Indeed, the computation of $\overline{\rho(t)}$ in the nudging phase requires knowing the values of ρ at $t - \tau$, which may belong to the free phase when t corresponds to the beginning of the nudging phase. In analog hardware, the delay τ would typically be achieved with capacitors comparable to those used for implementing the LIF neurons. In this sense, the EqSpike approach can be considered as local in time as well as in space. Neurons, integrators and filters are initialized to zero at each new image.

Simulations of EqSpike on MNIST

The MNIST dataset contains 60,000 images for training and 10,000 images for test. Pixel values are normalized between 0 and 0.5 before being sent as fixed input signals I to the input neurons. We consider a one-hidden layer bidirectional neural network, with 100 hidden neurons, resulting in the network topology 784-100-10. The loss function that the network attempts to minimize is the mean square error: $L(\hat{y}, y) = \frac{1}{M} \sum_{k=0}^M (\hat{\rho}_k - \rho_k)^2$, where the sum is performed over the output neurons. The target $\hat{\rho}_k$ corresponds to the maximum neuron frequency f_{max} if the class is correct, and zero otherwise. During the nudging phase, the output neurons integrate the error derivative $(\hat{\rho}_k - \rho_k)$ multiplied by the nudging factor β in addition to the currents that they receive from other neurons. In practice ρ_k is the rate evaluated over 100 simulation time steps. The batch size is one. Parameters in Table 1 have been optimized through hyperparameter search.

Training parameters for benchmark

Our rate-based benchmark with C-EP and BPTT use the prototypical models introduced in Ernout et al with rate-based neurons (33). The table below describes the hyperparameters used for the simulations.

	Alg	Topology	Activation	T1	T2	beta	Learning rates
MNIST	BPTT	784-100-10	hardsigm	30	15	NA	0.003 – 0.0015
MNIST	C-EP	784-100-10	hardsigm	30	15	0.5	0.003 – 0.0015

Table S2 - Parameters used for benchmarking EqSpike to BPTT and C-EP, related to Table 1.