



HAL
open science

An attention-based spiking neural network for unsupervised spike-sorting

Marie Bernert, Blaise Yvert

► **To cite this version:**

Marie Bernert, Blaise Yvert. An attention-based spiking neural network for unsupervised spike-sorting. International Journal of Neural Systems, 2019, 29 (8), pp.1850059. 10.1142/S0129065718500594 . cea-02316803

HAL Id: cea-02316803

<https://cea.hal.science/cea-02316803v1>

Submitted on 30 Sep 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

An Attention-Based Spiking Neural Network for Unsupervised Spike-Sorting

Marie Bernert^{*,†,‡} and Blaise Yvert^{*,†,§}

**BrainTech Laboratory U1205, INSERM, 2280 Rue de la Piscine
38400 Saint-Martin-d'Hères, France*

*†BrainTech Laboratory U1205, Université Grenoble Alpes
2280 rue de la piscine, 38400 Saint-Martin-d'Hères, France*

*‡LETI, CEA Grenoble
17 Rue des Martyrs, 38000 Grenoble, France*

§blaise.yvert@inserm.fr

Accepted 15 December 2018

Published Online 18 February 2019

Bio-inspired computing using artificial spiking neural networks promises performances outperforming currently available computational approaches. Yet, the number of applications of such networks remains limited due to the absence of generic training procedures for complex pattern recognition, which require the design of dedicated architectures for each situation. We developed a spike-timing-dependent plasticity (STDP) spiking neural network (SSN) to address spike-sorting, a central pattern recognition problem in neuroscience. This network is designed to process an extracellular neural signal in an online and unsupervised fashion. The signal stream is continuously fed to the network and processed through several layers to output spike trains matching the truth after a short learning period requiring only few data. The network features an attention mechanism to handle the scarcity of action potential occurrences in the signal, and a threshold adaptation mechanism to handle patterns with different sizes. This method outperforms two existing spike-sorting algorithms at low signal-to-noise ratio (SNR) and can be adapted to process several channels simultaneously in the case of tetrode recordings. Such attention-based STDP network applied to spike-sorting opens perspectives to embed neuromorphic processing of neural data in future brain implants.

Keywords: Spike-timing-dependent synaptic plasticity; spiking neural network; spike-sorting; unsupervised learning; attention mechanism.

1. Introduction

Pattern recognition is a fundamental task performed very efficiently by the brain. Artificial intelligence is making quick progress in reproducing these performances with artificial neural networks. In particular, deep neural networks have been successful both for static and sequential patterns recognition in many applications such as image recognition, analysis of

video streams or natural language processing. However, these types of neural networks require extensive training on large datasets and heavy computations. A promising alternative are spiking neural networks (SNNs), also known as the third generation of artificial neural networks,¹ which have in theory more computational efficiency.² Some class of SNNs such as spiking neural P systems have been shown to

[§]Corresponding author.

This is an Open Access article published by World Scientific Publishing Company. It is distributed under the terms of the Creative Commons Attribution 4.0 (CC-BY) License. Further distribution of this work is permitted, provided the original work is properly cited.

be Turing universal.³ Whereas formal neurons output a real value modeling a firing rate, spiking neuron models have a temporal dynamics and output spike trains, thus keeping the spike timing information. Importantly, they offer an opportunity to be implemented in very-low-power analog neuro-morphic hardware that currently undergoes important developments, in particular with memristive devices that are able to mimic synaptic plasticity, such as spike-timing-dependent plasticity (STDP), at a highly miniaturized scale.^{4–11} STDP rules are local learning rules that change the synapse weight depending on the time difference between pre- and post-synaptic spikes. It has been shown that incorporating such rule in SNNs allows to perform unsupervised learning,^{12,13} and this strategy has been successfully applied for visual or auditory pattern recognition.^{14–17} Yet, for now, the range of applications of spiking neural networks in pattern recognition remains limited, most studies being focused on biological modeling.^{18–22} Whereas second-generation networks, and some specific spiking neural network,^{23–25} can be trained using backpropagation, there is no well-established way to train a spiking neural network implementing an STDP rule. New architectures thus remain to be proposed to address a wider range of applications of these networks.

Here, we present an STDP spiking neural network to perform spike-sorting, a central problem in neuroscience. Spike-sorting consists in determining from a common extracellular signal how many cells emit action potentials and, for each cell, when these events take place. Action potentials occur sparsely at discrete time points, and appear in the signal as temporal waveforms whose shapes typically differ from one neuron to another according to the geometry and position of the cells with respect to the recording electrode. Thus, spike-sorting can be seen as an unsupervised pattern recognition task, where the goal is to detect and classify different temporal waveforms, occurring sparsely in a noisy signal. Currently available spike-sorting approaches typically use three separate steps.

- The first step is the detection of action potentials, usually done by thresholding the extracellular signal directly or after suitable filtering, for example using an energy operator²⁶ or templates.²⁷

- The second step is the extraction of features from the action potential waveforms, such as amplitudes,²⁷ wavelet coefficients²⁸ or reduced representations obtained with dimensionality reduction algorithms such as PCA.^{29–31}
- Finally, the last and most computation-demanding step consists in clustering these features. Many clustering algorithms have been proposed, such as expectation-maximization,^{29,30} k -means,³² c -means,³³ mean-shift^{31,34} or superparamagnetic clustering.²⁸ Noticeably, Ref. 35 used an STDP network for this clustering step.

Although some spike-sorting algorithms have been designed to process neural data online,^{26,35,36} most methods remain offline or require an offline preprocessing step, precluding their use in online applications such as brain-computer interfaces (BCIs). Moreover, the current availability of very dense arrays of microelectrodes^{37–42} creates the need for efficient ways to handle the important data flow generated by these devices and in particular to automate spike-sorting processing.^{30,31,34,36,43} Ideally, implantable neural interfaces would strongly benefit from fully automatic spike-sorting algorithms compatible with very-low-power hardware implementation for future embedding at the electrode site. However, no such algorithm is available yet.

In this context, we propose to use a spiking neural network as a radically new way to handle the problem of spike-sorting. We present an STDP network architecture designed for this specific problem. The network processes continuously the stream of data recorded by a microelectrode and directly outputs trains of artificial spikes that correspond, after a short learning period, to the sorted activity of the recorded cells. Thus, the entire processing is done by a single network in a fully unsupervised manner, without explicit implementation of the conventional three-step procedure and without requiring a read-out post-processing step. Beyond the classical STDP learning rule, the network combines an attention mechanism, delayed synapses and threshold adaptation to handle simultaneously the problems of detecting and classifying time-varying patterns with different sizes. This approach was tested on both simulated and real data and compared to two available spike-sorting software, *Osort*²⁶ and *Waveclus*,²⁸

showing better performance at low signal-to-noise ratio (SNR).

2. Materials and Methods

2.1. Network model

We considered an artificial spiking neural network composed of three layers (input, intermediate and output) and one “attention” neuron, all connected by feedforward synapses implementing specific plasticity rules (Fig. 1). This network, composed of “sensory” and Leaky Integrate-and-Fire (LIF) neurons, is bio-inspired rather than realistic. In particular, the characteristic times are much shorter than realistic ones, which allows processing the information contained in the action potential shapes and makes the approach compatible with real-time spike-sorting at the millisecond time scale. After learning, the output spikes of the network correspond to the detected and sorted action potentials present in the input signal. To avoid any confusion in the following, the term “spikes” hereafter denotes action potentials emitted by neurons of the artificial neural network, while “action potentials” denotes those of real neurons embedded in the input signal analyzed by the network.

2.1.1. Input layer

The first layer, called the input layer, encodes the incoming neural signal within a sliding time window preceding the current time point into artificial spikes that are passed on to the next layers. Neurons of this input layer act as “sensory” neurons.

The input neurons are organized into a grid with N_c columns corresponding to different processing delays of the input signal, starting from zero and regularly spaced by Δt_c [see Fig. 2(a)]. Within each

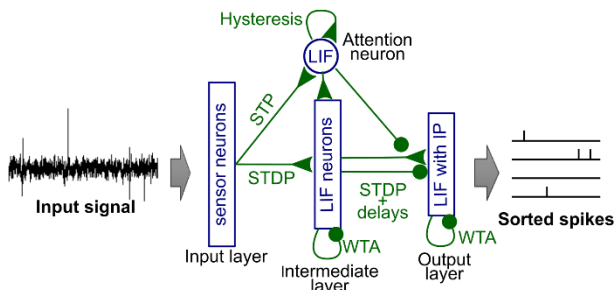


Fig. 1. Overview of the network structure.

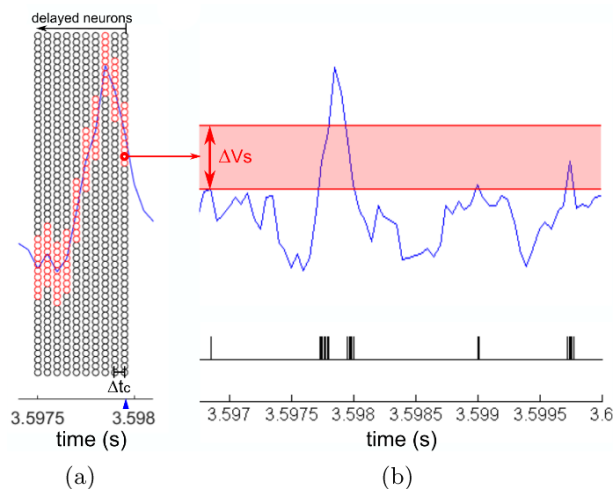


Fig. 2. (Color online) Input layer structure: (a) Input layer activation at a specific time step (blue arrow). (b) Spike train emitted by one specific input neuron, highlighted in bold red in (a).

column, each neuron is sensitive to a given range of signal values and fires at regular time steps Δt_s when the signal falls within this sensitivity range [see Fig. 2(b)]. Thus, at each of these time steps, each input neuron performs a simple computation that consists in comparing the input signal value to the extrema of its sensitivity range and fires accordingly. The size of this range, ΔV_s , is the same for each neuron, and is set proportional to the noise level. The sensitivity ranges of neurons from one column are regularly overlapping so that, for each possible signal value, N_{overlap} neurons of this column fire. Within one column, the step between two consecutive sensitivity ranges is determined by the size of the sensitivity range ΔV_s and the overlapping factor N_{overlap} , and the total number of neurons is determined by the total signal value range to cover. The input layer processes the input signal at regular time steps, Δt_s . This time step is chosen sufficiently short so that the time jitter due to sampling does not impact the performance of the network. As this time step is shorter than the original signal sampling time step, the signal is linearly interpolated for steps falling between samples. Assuming that the timing properties are consistent between different neural recordings, the sensitivity range size is the only parameter of the network that needs to be adjusted depending on the signal properties. The values of the input layer parameters are given in Table 1.

Table 1. Input layer parameters.

Parameter	Description	Value
ΔV_s	Sensitivity range size	$3.5\sigma_{\text{noise}}$
N_{overlap}	Number of neurons active at the same time within one column	10
Δt_s	Delay interval between two activations of the input layer	0.0125 ms
Δt_c	Time interval between two columns	0.05 ms
N_c	Number of columns	10

As a result, the input layer converts continuously the input signal into an artificial spike train encoding at each time step the shape of the signal within a sliding window. The role of the rest of the network is to detect different patterns into this input spike train that reflect the shape of action potentials stemming from different cells.

2.1.2. Attention neuron

The role of the ‘‘attention’’ neuron is to detect every action potential occurrence embedded in the input signal. Each time an action potential is present in the input signal, the attention neuron fires a sequence of spikes from the beginning to the end of the action potential. This spiking activity in turn modulates the intermediate and output layers, thus acting as an attention mechanism gating the learning mechanisms occurring in these two subsequent layers.

The attention neuron receives all spikes emitted by the input layer through excitatory synapses that implement a short-term plasticity (STP) rule, governed by the following equation:

$$\frac{dw}{dt} = \frac{1}{\tau_{\text{stp}}}(1 - w) - \sum_s w * f_d * \delta(t - t_s), \quad (1)$$

where w is the synaptic weight, τ_{stp} a time constant, f_d a depression factor and t_s the pre-synaptic spike times. This STP rule weakens the weights of synapses for which pre-synaptic spikes occur at high frequency.⁴⁴ As a result, the weights of the synapses corresponding to pre-synaptic input neurons encoding signal amplitudes within the range of noise (and thus often activated) are weakened, while synapses corresponding to pre-synaptic input neurons encoding signal amplitudes outside this range remain strong. The time constant τ_{stp} is set long compared to the duration of an action potential,

so that the synaptic weights do not change significantly during an action potential. The depression factor f_d is set so that the asymptotic weight of a synapse stemming from an input neuron firing at each time step is significantly lower ($\times 0.13$ in our network implementation) than that stemming from an input neuron never firing. Thanks to these STP synapses, the excitation received by the attention neuron is much more important for infrequent signal amplitudes than for signal values close to zero (Fig. 3, top and middle).

The attention neuron is an LIF neuron, whose membrane potential V is governed by the following equation:

$$\frac{dV}{dt} = -\frac{1}{\tau_m} * V(t) + \sum_i \sum_s w_i(t_{i,s})\delta(t - t_{i,s}), \quad (2)$$

where V stands for the neuron potential, τ_m for the membrane time constant, i indexes the incoming synapses with synaptic weight w_i , s indexes the received spikes from each synapse i with arrival time $t_{i,s}$ and δ is the Dirac delta function. The membrane time constant of the attention neuron is chosen relatively short, in order to detect an action potential as early as possible. Combined with the STP rule, this LIF dynamic leads to an increased membrane potential when an action potential occurs in the input signal (Fig. 3, bottom). The attention neuron threshold value is then set empirically to obtain a

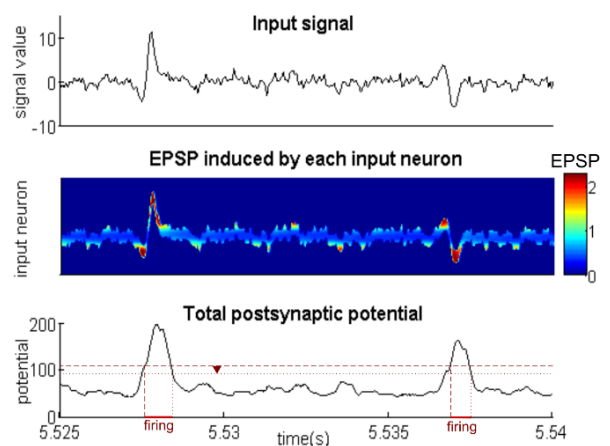


Fig. 3. Short-term plasticity effect on the attention neuron. Top: Example of input signal. Middle: Corresponding EPSP induced on the attention neuron by each input neuron of one input layer column. Bottom: Time evolution of the attention neuron potential and its firing times given its threshold (dashed line) and self-excitation (arrow).

compromise between false negative and false positive detection errors. The neuron has no refractory period and no reset potential is applied. Instead, after firing, the potential is increased through a self-excitatory synapse, which constitutes a hysteresis mechanism. The self-excitatory synapse ensures that, even if the action potential waveform crosses zero, thus lowering temporarily the attention neuron potential, the detection spike train is continuous from the beginning to the end of the action potential. The values of the attention neuron parameters are given in Table 2.

2.1.3. Intermediate layer

The role of each intermediate layer neuron is to learn to recognize specific fragments of action potential waveforms present in the input signal so that the

layer emits different firing sequences for different action potential waveforms. The activity of intermediate layer neurons is gated by the attention neuron through fixed-weight excitatory synapses, ensuring that these neurons fire when the attention neuron fires and remain silent otherwise. Thus, for each action potential occurrence in the signal, the intermediate layer fires a sequence of spikes characterizing the action potential shape, as it is different for different action potential waveforms and stable across occurrences of the same waveform [Fig. 4(a)]. These different sequences are then processed by the output layer, to sort the action potentials.

Neurons of the intermediate layer are LIF neurons following a similar equation as the attention neuron [Eq. (2)]. They receive all spikes emitted by

Table 2. Attention neuron parameters.

Parameters	Description	Value
τ_m	Membrane time constant	$0.5 * \Delta t_c = 0.025$ ms
τ_{refrac}	Refractory period	0
Th	Neurons threshold	$0.43 * N_{\text{overlap}} * N_c / (1 - \exp(-\Delta t_s / \tau_m)) = 108.1$
w_{self}	Weight of the self-excitatory synapse	$0.3 * (1 / \exp(-\Delta t_s / \tau_m) - 1) * \text{Th} = 10.5$
τ_{stp}	Short-term plasticity time constant	$10 * \Delta t_c * N_c = 5$ ms
f_d	Short-term plasticity depression factor	$1 - \exp(-6.5 * \Delta t_s / \tau_{\text{stp}}) = 0.0161$

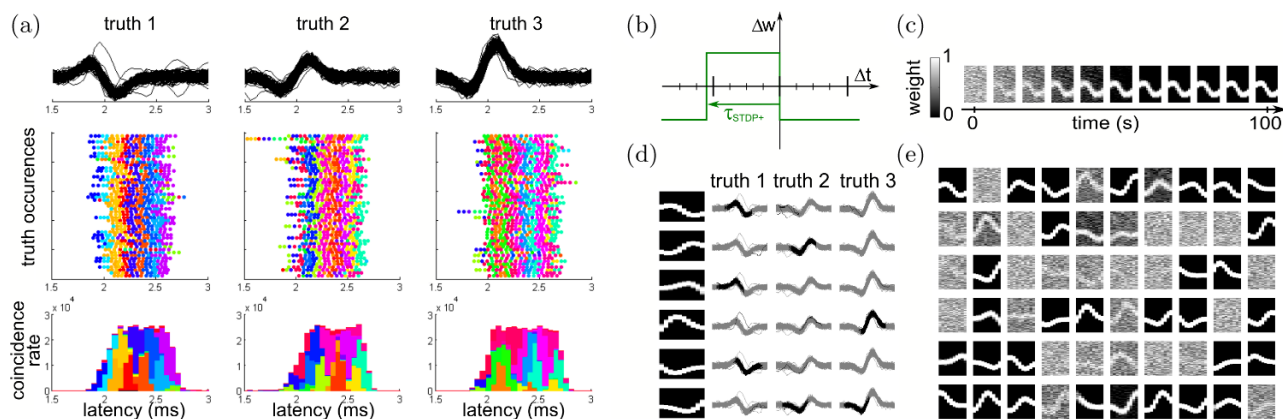


Fig. 4. The intermediate layer learns waveform fragments: (a) Spike sequence emitted by the intermediate layer for each action potential occurrence. Top row: Signal shapes for all action potential occurrences in the last 100 s of simulation. Middle row: Spike trains of the intermediate layer synchronized with 50 different action potential occurrences. Bottom row: Distributions of intermediate spike latencies relative to each action potential occurrence (the histograms are cumulated). The different colors stand for different intermediate neurons. (b) STDP rule applied on the synapses stemming from the input layer. (c) Weight evolution of the synapses projecting on one specific intermediate neuron. (d) Learnt shapes correspond to fragments of waveforms occurring in the signal. The first column shows the learnt weights for six different intermediate neurons. The three other columns show the signal shape at each action potential occurrence, the black lines show the part of the signal encoded by the input layer at the moment when the intermediate neuron fires. (e) Weights learnt by each of the 60 intermediate neurons after a 200-s simulation.

the input layer through excitatory synapses whose weights vary according to an STDP rule. The membrane time constant τ_m of the intermediate neurons is of the same order of magnitude as Δt_s , so they are sensitive to the shape of the signal at current time, which they have to recognize. As part of a winner-take-all (WTA) mechanism, each time an intermediate neuron fires, all intermediate neurons reset their potentials to zero. This ensures that only one neuron fires at a time. Intermediate neurons thus fire one after the other, separated by a time interval that depends on the received excitation and the threshold, and is here adjusted to be approximately equal to Δt_c .

The STDP rule is based on a coincidence time window defined by the value $\tau_{\text{stdp}+}$ [Fig. 4(b)]. A pre-synaptic spike is considered to coincide with a post-synaptic spike if it arrives at most $\tau_{\text{stdp}+}$ before the post-synaptic spike. For each post-synaptic spike occurrence, the synapse weight is decreased by Δw_{post} , and additionally, if a pre-synaptic spike coincides with the post-synaptic spike, the synapse weight is increased by Δw_{pair} (resulting in a total change of $\Delta w_{\text{post}} + \Delta w_{\text{pair}}$). Consequently, the synaptic weight globally increases toward one if the ratio of pre- and post-synaptic spike coincidences over post-synaptic spike occurrences is superior to the ratio $|\Delta w_{\text{post}} / \Delta w_{\text{pair}}|$, and decreases to zero otherwise.

The synapses weights are initialized randomly according to a uniform distribution. The mean weight is chosen so that the neurons potential is just high enough to reach the threshold when the attention neuron also fires. Thus, at the beginning of the signal processing, when an unknown action potential waveform is presented, one intermediate neuron arbitrarily fires first and inhibits the others. This neuron updates its synaptic weights whereas synapses projecting to the other intermediate neurons remain unchanged. The STDP time window being equal to Δt_c , the potentiated synapses reflect the shape of the signal at this specific time with the same time resolution as the input layer. Consequently, when a similar waveform fragment occurs again, this neuron is more likely to fire and thus reinforce the specificity of its response to this particular waveform pattern [Figs. 4(c) and 4(d)]. Once learning has been achieved, several intermediate neurons become specifically responsive to different waveform fragments [Figs. 4(d) and 4(e)]. The intermediate layer is constituted of 60 neurons, which we found enough to learn the different possible input patterns. This was confirmed by the fact that some intermediate neurons did not learn any pattern [Fig. 4(e)]. It is possible to increase this number if a great variety of waveforms is expected in the signal. The values of the intermediate layer parameters are given in Table 3.

Table 3. Intermediate layer parameters.

Parameter	Description	Value
N_{neur}	Number of neurons	60
τ_m	Membrane time constant	$2 * \Delta t_s = 0.025$ ms
τ_{refrac}	Refractory period	$\Delta t_c = 0.05$ ms
V_{reset}	Reset potential	0
V_{inhib}	Post-inhibition potential	0
w_{AN}	Weight of the synapses coming from the attention neuron	$0.6 * N_c * N_{\text{overlap}} = 60$
Th	Neurons threshold	$(w_{\text{AN}} + 0.7 * N_c * N_{\text{overlap}}) * (1 - \exp(-\Delta t_c / \tau_m)) / (1 - \exp(-\Delta t_s / \tau_m)) = 252.7$
$w_{\text{init}-}$	Minimal value for the feedforward synapses weight initialization	0.4
$w_{\text{init}+}$	Maximal value for the feedforward synapses weight initialization	1
$\tau_{\text{stdp}+}$	Positive STDP rule time window	$\Delta t_c + 0.5 * \Delta t_s = 0.0563$ ms
$\tau_{\text{stdp}-}$	Negative STDP rule time window	0
Δw_{pair}	Weight change for a pre-synaptic spike coinciding with a post-synaptic spike	0.005
Δw_{post}	Weight change for each post-synaptic spike	$-0.55 * \Delta w_{\text{pair}}$

2.1.4. Output layer

The purpose of the output layer is to finalize the spike-sorting process, by learning to recognize spike sequences produced by the intermediate layer that correspond to combinations of waveform fragments. If the input signal contains action potentials of N different cells, a successful spike-sorting results in exactly N active neurons of the output layer, each firing once for each occurrence of an action potential of a given cell in the input signal. An important problem to overcome is the ability to learn patterns of different sizes and to differentiate between patterns that may overlap or include one into another. Overlapping patterns can occur for example if two action potential waveforms have the same beginning but different endings.

The output layer is designed here to overcome these problems using four features in addition to the STDP rule and the WTA mechanism.

- First, each pair of intermediate and output neurons are connected through multiple synapses with different transmission delays, as in Refs. 45–47. This gives information to the output layer about the spike times of the intermediate layer [Fig. 5(a)].
- Second, the output layer is inhibited by the attention neuron. This inhibition is applied on the feedforward synapses from the intermediate layer, after the transmission delay (mimicking a biological pre-synaptic inhibition), so that intermediate spikes arriving at the output layer are prevented from being transmitted to the output neuron if

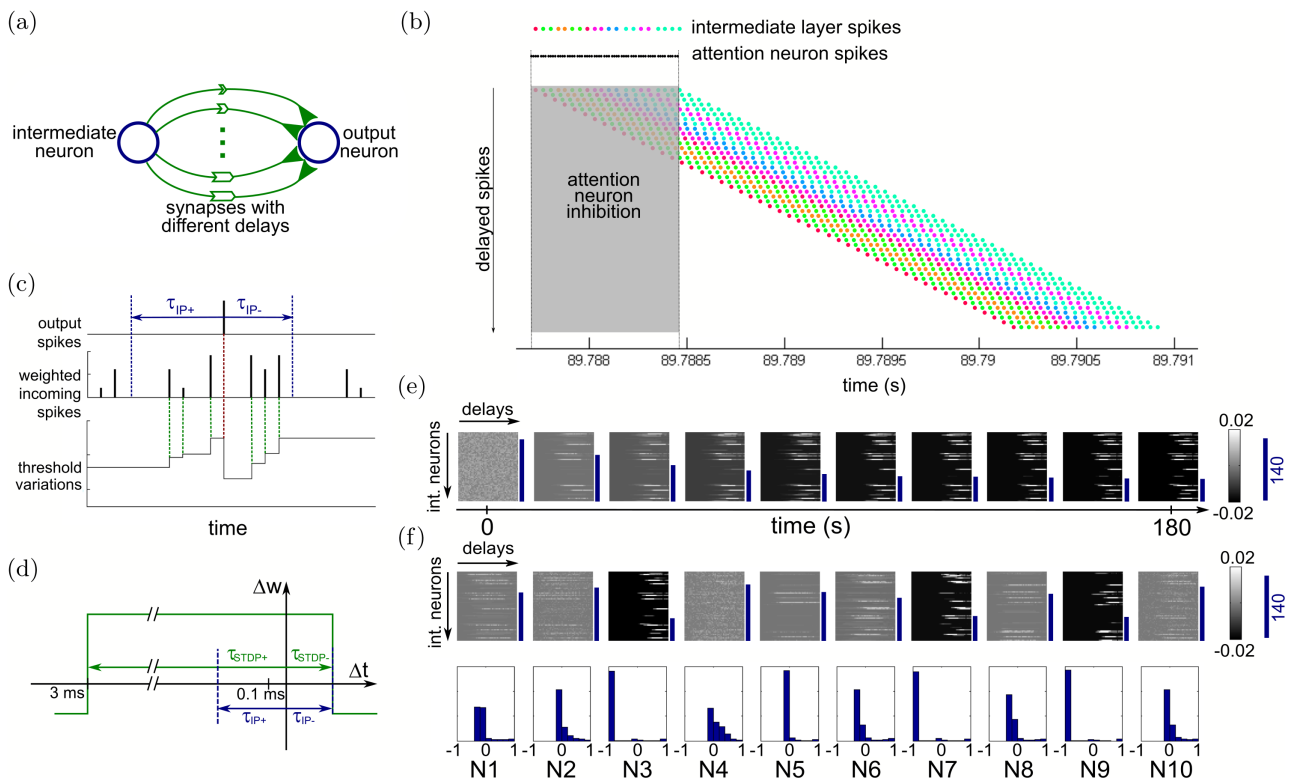


Fig. 5. (Color online) The output layer learns to recognize different intermediate layer patterns. (a) Connection between intermediate and output neurons, through several synapses with different transmission delays. (b) Spike train received by the output layer, depending on the intermediate layer and attention neuron spike train. The different colors stand for different intermediate neurons. (c) Intrinsic plasticity principle. (d) STDP rule applied on the synapses stemming from the intermediate layer. (e) Evolutions of weights (in grayscale) and thresholds (blue bars) of one output neuron. The weights are organized according to their delays and their pre-synaptic intermediate neuron. Shown weights are the sum of the inhibitory and excitatory synapse weights divided by the neuron's threshold. (f) Synaptic weights and thresholds learned by each of the 10 output neurons. Top row: Final synaptic weights and thresholds at the end of a 200-s simulation, represented similarly as in (e). Bottom row: Distributions of final synaptic weights for each output neuron, without threshold normalization.

the attention neuron is firing at the same time [Fig. 5(b)]. These spikes are thus not taken into account for the plasticity rules. This forces the output neurons to wait until the end of a pattern before firing and thus to take into account the entire pattern to adjust the weights of incoming synapses.

- Third, each neuron of the output layer receives the spikes of each neuron of the intermediate layer through both excitatory and inhibitory synapses implementing an STDP rule, allowing the overall weight to take negative or positive values. As a result, a neuron that has learnt a pattern gets excited by spikes belonging to this pattern and inhibited by spikes not belonging to the pattern. Thus, its potential is maximal when the pattern is exactly reproduced, without missing or additional intermediate neuron spikes.
- Fourth, the output neurons implement an intrinsic plasticity (IP) rule [Fig. 5(c)], which enables them to adapt their threshold to the learnt pattern size. This ensures that an output neuron that has learnt to recognize a long pattern has a sufficiently high threshold preventing it to fire if the incoming pattern is incomplete, whereas a neuron that has learnt a short pattern has a low threshold and is still able to detect it. The introduction of this IP rule on the output neurons in conjunction with the use of excitatory and inhibitory synapses is key to solve the problem of recognition of overlapping patterns or patterns that included one into another.

The output neurons, thresholds are initialized at a high value, and the feedforward synapses weights are initialized randomly, according to a uniform distribution, with an average positive weight. Before learning, when an output neuron receives a spike pattern from the intermediate layer, it fires late due to its high threshold. Once a neuron fires, its threshold begins to evolve according to the intrinsic plasticity rule [Fig. 5(c)]. Every time an output neuron emits a post-synaptic spike, its threshold Th is decreased by $\Delta Th_{\text{post}} = F^{\Delta Th_{\text{post}}} * Th$. For each pre-synaptic spike received within a coincidence window around the post-synaptic spike $[-\tau_{IP} + \tau_{IP}]$, the threshold is increased by ΔTh_{pair} multiplied by the synaptic weight. Additionally, the threshold is bounded between a minimum value and a maximum

value (see Table 4). It can be shown that the equilibrium threshold value is proportional by a factor $F^{\Delta Th_{\text{post}}}/\Delta Th_{\text{pair}}$ to the average weighted sum of the pre-synaptic spikes received within the coincidence window around a post-synaptic spike. As a consequence, the learning neuron's threshold, which is initialized at its maximum value, progressively decreases and the neuron fires earlier than the neurons that have not learnt any pattern [Fig. 5(e)].

The synaptic weights evolve in parallel to the threshold according to an STDP rule [Fig. 5(d)]. The rule used is similar to the one used for the intermediate layer, with a different coincidence time window and different weight change values. The coincidence time window is defined by both $\tau_{\text{stdp}+}$, for pre-synaptic spikes occurring before a post-synaptic spike, and $\tau_{\text{stdp}-}$, for pre-synaptic spikes occurring after a post-synaptic spike. $\tau_{\text{stdp}+}$ is chosen quite long to facilitate learning at the beginning, when the neuron fires late due to its high threshold. The inhibitory and excitatory feedforward synapses follow an STDP rule with the same coincidence time window, but with a different $|\Delta w_{\text{post}}/\Delta w_{\text{pair}}|$ ratio, so that after learning, the summed weight converges to either 1, 0 or -1 depending on the ratio of the number of pre- and post-synaptic spike coincidences over the number of post-synaptic spike occurrences. Once an output neuron has learnt a pattern [see neurons 3, 7 and 9 in the example of Fig. 5(f)], most of its incoming synapses have converged to the minimum negative weight value. Only the synapses corresponding to the intermediate neurons and delays constituting the pattern converge to the maximum positive weight, and the few remaining synapses not relevant for discriminating the pattern converge to a null weight [Fig. 5(f)]. By contrast, the incoming synaptic weights of neurons that did not learn any pattern remain distributed near the zero value.

The output layer is constituted of 10 LIF neurons. The number of neurons is chosen higher than the maximum number of expected action potential waveforms in the signal. These neurons have their potentials reset after firing or after a lateral WTA inhibition. The reset potential and the lateral inhibition potential are proportional to the neuron threshold, which vary according to the IP rule. The values of the output layer parameters are given in Table 4.

Overall, the proposed architecture of the network allows the output neurons to learn to recognize

Table 4. Output layer parameters.

Parameter	Description	Value
N_{neur}	Number of neurons	10
τ_m	Membrane time constant	3 ms
τ_{refrac}	Refractory period	3 ms
F_{reset}	Reset potential factor. The reset potential is $V_{\text{reset}} = F_{\text{reset}} * \text{Th}$	-10
F_{inhib}	Lateral inhibition potential factor. The lateral inhibition potential is $V_{\text{inhib}} = F_{\text{inhib}} * \text{Th}$	-10
N_{delays}	Number of synaptic delays	50
Δt_d	Time interval between synaptic delays	$\Delta t_c = 0.05$ ms
$w_{\text{init-}}^{\text{pos}}$	Minimal value for the excitatory feedforward synapses weight initialization	0
$w_{\text{init+}}^{\text{pos}}$	Maximal value for the excitatory feedforward synapses weight initialization	1
$w_{\text{init-}}^{\text{neg}}$	Minimal value for the inhibitory feedforward synapses weight initialization	0
$w_{\text{init+}}^{\text{neg}}$	Minimal value for the inhibitory feedforward synapses weight initialization	0
$\tau_{\text{stdp+}}$	Positive STDP rule time window	3 ms
$\tau_{\text{stdp-}}$	Negative STDP rule time window	$4 * \Delta t_c = 0.2$ ms
$\Delta w_{\text{pair}}^{\text{pos}}$	Excitatory synapses weight change for the pre-synaptic spike coinciding with a post-synaptic spike	0.05
$\Delta w_{\text{post}}^{\text{pos}}$	Excitatory synapses weight change for each post-synaptic spike	$-0.7 * \Delta \Delta w_{\text{pair}}^{\text{pos}}$
$\Delta w_{\text{pair}}^{\text{neg}}$	Inhibitory synapses weight change for the pre-synaptic spike coinciding with a post-synaptic spike	-0.05
$\Delta w_{\text{post}}^{\text{neg}}$	Inhibitory synapses weight change for each post-synaptic spike	$-0.1 * \Delta \Delta w_{\text{pair}}^{\text{neg}}$
Th_{min}	Lower bound of the neuron threshold	8
Th_{max}	Upper bound of the neuron threshold, also used as initialization value	$-0.35 * N_{\text{delays}} * \text{Th}_{\text{min}} = 140$
$\tau_{\text{Ip+}}$	Positive IP rule time window	$6 * \Delta t_c = 0.3$ ms
$\tau_{\text{Ip-}}$	Negative IP rule time window	$4 * \Delta t_c = 0.2$ ms
$F^{\Delta \text{Th}}_{\text{post}}$	Proportional threshold decreases for each post-synaptic spike	0.01
$\Delta \text{Th}_{\text{pair}}$	Threshold increases for each pre-synaptic spike coinciding with a post-synaptic spike	$0.6 * F^{\Delta \text{Th}}_{\text{post}}$

different patterns generated in the intermediate layer and to emit one spike for each occurrence of an action potential waveform in the input neural signal. Thus, each active output neuron directly predicts the activity of one real cell (Fig. 6).

2.2. Simulated data

The spike-sorting methods were first tested on simulated data generated using a method adapted from the literature.⁴⁸ Simulated signals are the superposition of correlated noise and action potentials occurring at known timestamps thus providing a known ground truth. The simulated sampling frequency is 20 kHz. The waveform of each truth neuron action potential is defined according to the following template equation:

$$V(t) = A \cos\left(2\pi \frac{t - t_{\text{ph}}}{\tau_1}\right) \exp\left(-\left(\frac{2.3548t}{\tau_2}\right)^2\right). \quad (3)$$

The parameters of this equation are given in Table 5 for the three neurons that were simulated. The coefficient A is computed so that the maximum value of the template matches the parameter A_{max} . The time occurrences of these waveforms are defined according to a Poisson process, each with a mean firing rate of 3.3 Hz (unless otherwise stated), with a simulation time step of 0.05 ms corresponding to a 20-kHz sampling frequency. Additionally, a 3-ms refractory period is applied for each neuron. The noise added to the signal is a correlated noise generated by a dynamical Ornstein–Uhlenbeck process, with a 0.1-ms time constant and a simulation step of 0.002 ms. Simulated signals were generated with different noise levels to achieve different signal-to-noise ratios defined as $\langle |A_{\text{max}}| \rangle / \sigma_{\text{noise}}$, where $\langle |A_{\text{max}}| \rangle$ is the mean of the peak amplitudes of the three action potentials and σ_{noise} is the standard deviation of the noise. Seven different noise levels were defined (from

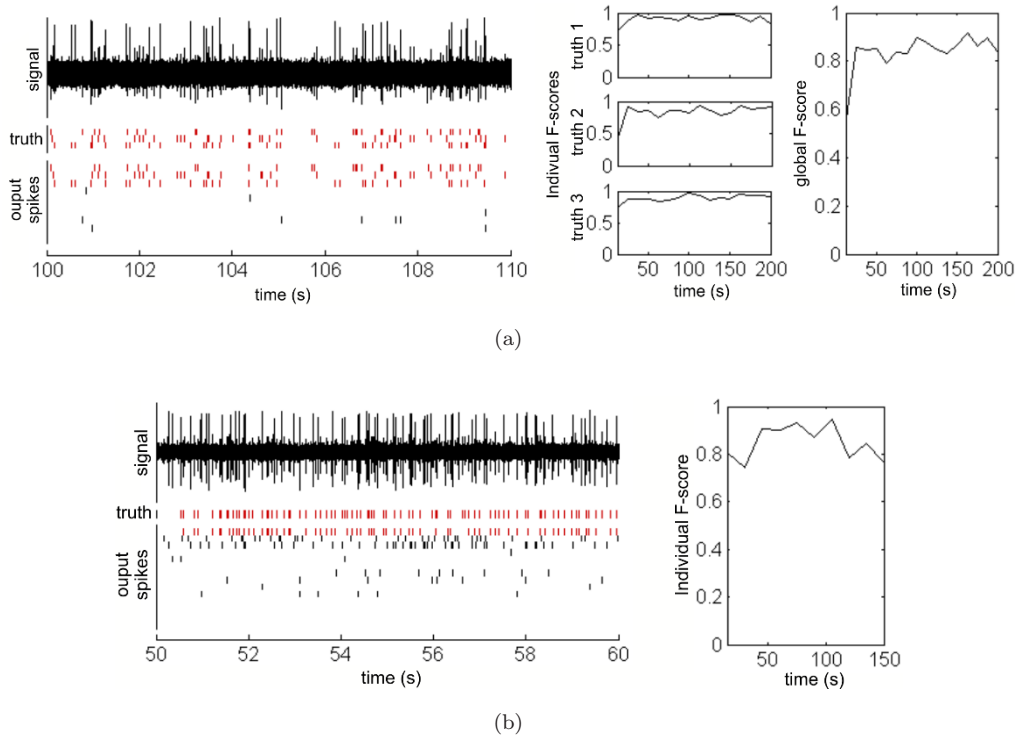


Fig. 6. (Color online) Examples of spike-sorting achieved by the artificial network on simulated and real data. (a) Performance on simulated data. Left: comparison of the output spike trains generated by the 10 output neurons with the truth spike trains generated by the three simulated neurons, on a 10-s segment of input signal. Matching spike trains have been highlighted in red. Right: evolutions of the performance of the network over the 200-s simulation. (b) Same as (a) for a real recording (dataset d553101). Here the truth is only known for one cell thus only the performance relative to this cell can be computed.

Table 5. Waveform template parameters.

	A_{\max}	$\tau_1(\text{ms})$	$\tau_2(\text{ms})$	$t_{\text{ph}}(\text{ms})$
Waveform 1	5	1	0.5	-0.25
Waveform 2	5	1	0.5	0.25
Waveform 3	10	1	0.5	-0.19

0.5 to 2) and 10 200-s datasets were generated for each of them.

2.3. Tetrode data

The spike-sorting methods were also evaluated on real recordings from the hippocampus region CA1 of anesthetized rats, available from the Buzsaki Laboratory^{49,50} (datasets d533101 and d11221.002). Before sorting, the signals were band-pass-filtered using a first-order Butterworth filter (300–3000 Hz). The d533101 dataset, having an original 10-kHz sampling frequency, was up-sampled, for convenience,

at 20 kHz using a Whittaker–Shannon interpolation. However, the signal was always resampled to 80 kHz at the input layer level (see Table 1).

2.4. Spike-sorting performance evaluation

To assess the performance of the spike-sorting method, we computed indices based on an F -score. For each pair of truth neuron i and output neuron j , we computed the F -score as

$$F_{ij} = \frac{2 * H_{ij}}{T_i + O_j}, \quad (4)$$

where T_i is the number of spikes emitted by the i th truth neuron, O_j is the number of spikes emitted by the j th output neuron and H_{ij} is the number of output spikes coinciding with a truth spike within a 3-ms coincidence time window. Note that the F -score combines the recall and the precision of a prediction.

We also computed a global F -score across all truth neurons and all output neurons as

$$F = \frac{2 * H}{T + O}, \quad (5)$$

where T is the total number of truth spikes, O the total number of output spikes and H the total number of hits. To compute H , it is necessary to choose a correspondence between output neurons and truth neurons. Then H is the number of output spikes coinciding with a corresponding truth spike within a 3-ms coincidence time window. The pairing between output neurons and truth neurons was chosen to maximize H .

We also computed the recall as the number of correctly detected action potentials over the number of true action potentials, the precision as the number of correctly detected action potentials over the number of detected action potentials and a clustering score as the number of correctly classified action potentials over the number of correctly detected action potentials.

2.5. Statistical tests

The STDP network was compared to two other spike-sorting methods, Wave_clus²⁸ and Osort,²⁶ on both the simulated and real data. For the simulated data, 10 200-s datasets were generated for each of the seven different noise levels, and each of the three compared methods were run once on each dataset. For the tetrode data, the STDP method was run eight times on each channel, Wave_clus was run eight times on each channel, and Osort was run once on each channel as its result was deterministic. We then used the tetrode data to evaluate the performance of the STDP network when adapted to process all channels simultaneously. In this case, the STDP network was run 40 times both on each channel separately and on all channels simultaneously using two different tetrode network architectures (see Fig. 8). Results obtained by the two tetrode methods were then compared to the best result obtained using only a single channel. As the variances were significantly different for the different groups (as assessed by a Bartlett test), a Welch test was used for two by two comparisons, except for the comparison with Osort on the tetrode data where a one-sample t -test was used. A Bonferroni correction was used for each set

of multiple comparisons. All tests were performed using Matlab R2014a.

3. Results

3.1. Performance of the network on simulated and real extracellular data

Figure 6 shows two examples of spike-sorting results obtained on both artificial data with known ground truth and real extracellular tetrode recordings associated with an intracellular recording providing ground truth for one cell.^{49,50} Different types of error can occur when comparing the ground truth spike trains to the spike trains predicted by the sorting method: false negative, false positive and wrong classifications. The performance of the method was assessed using the F -score, which accounts for all types of error. Figure 6(a) shows an example of 10s of simulated data embedding three different action potential waveforms, each firing at known timestamps according to a Poisson distribution with an average rate of 3.3 Hz. As shown in this figure, the output spiking pattern of the STDP network showed three mainly active output neurons whose spike trains closely resembled those of the three embedded signal waveforms. The network quickly converged to provide high classification rates as assessed over the last 100 s by an overall F -score of 85%, with individual scores of 90%, 89% and 93% for each of the waveforms. Figure 6(b) shows an example of 10 s of real extracellular data for which the activity of one neuron was known from a simultaneous intracellular recording. The output of the STDP network showed three main active neurons, the activity of one being close to that of the intracellularly identified cell. For this cell, the sorting F -score of the network was stable for 150 s with an average value of 89% in the last 100 s.

We further compared our method with two open-source software able to perform unsupervised spike-sorting: Osort²⁶ and Wave_clus.²⁸ Figure 7(a) shows the confusion results for the three methods on a simulated dataset. In this example, the STDP network made only a few false negative errors, while Wave_clus made both false negative and false positive errors and Osort failed to detect one of the waveform and misclassified the two others in the same cluster, with an important number of false negative errors [Fig. 7(a)]. We further considered artificial

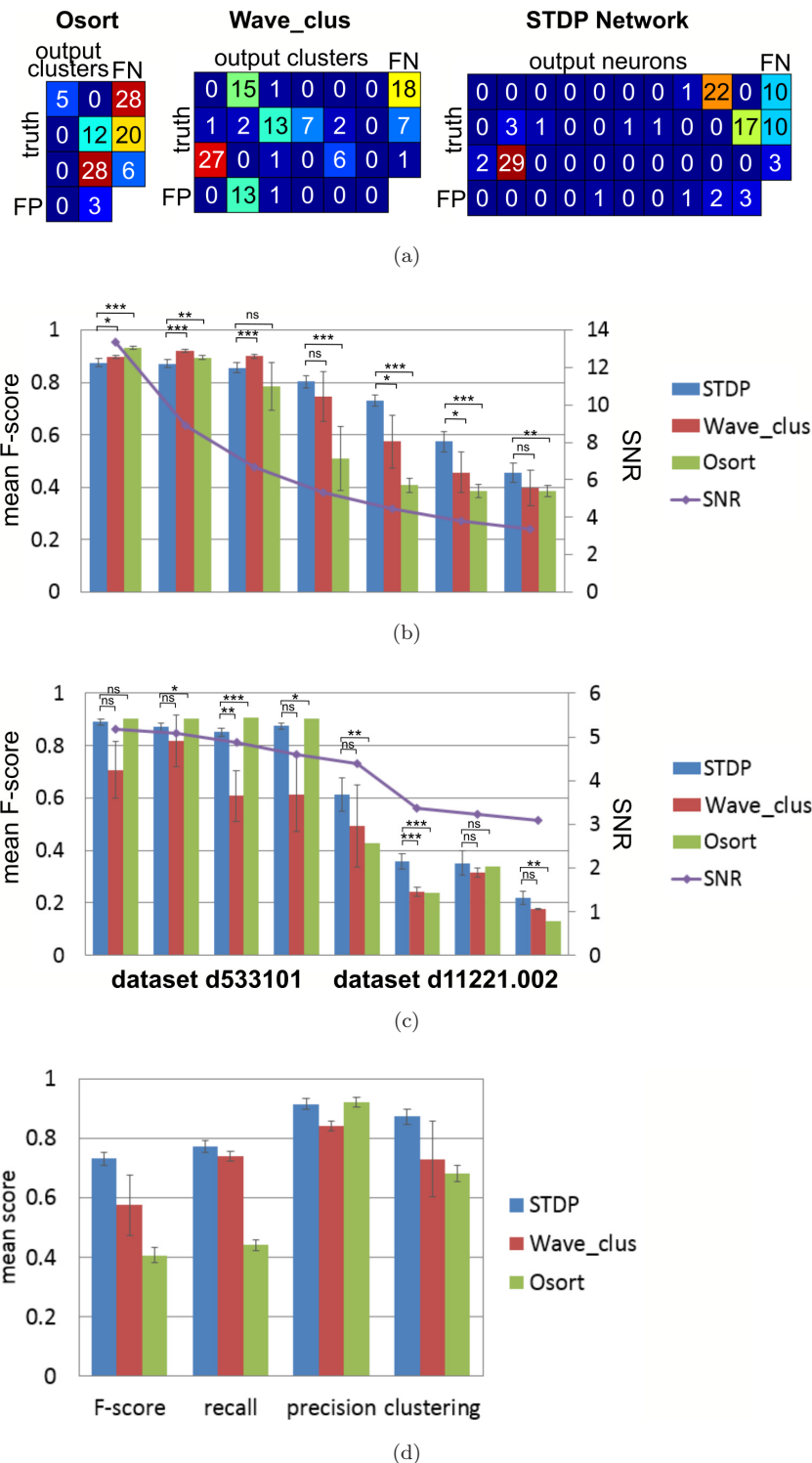


Fig. 7. Comparison of the performance of the STDP network with those of two other algorithms (Osort and Wave_clus). (a) Confusion matrices for the three sorting methods for an SNR = 4.4 simulated dataset. The values are normalized relatively to 100 ground truth spikes, i.e. 33 occurrences of each waveforms. (b) Mean F -score obtained on simulated data. (c) Mean F -score obtained on real tetra recordings. “ns” stands for nonsignificant, * for $p < 0.05$, ** for $p < 0.01$, and *** for $p < 0.001$ (d) Detailed mean scores on the 10 simulated datasets with an SNR of 4.5.

data with various SNRs. At very high SNRs, the STDP method gave slightly lower performance than Osort and Wave_clus, but the F -score differences remained low (<0.06). At lower SNRs more comparable to those encountered in real cortical recordings, the STDP method became more efficient than the two other methods with strong F -score differences up to 0.3 for an SNR of 4.4 [Fig. 7(b)]. These performances were further observed on real neural signals [Fig. 7(c)]. In this case, the STDP approach performed similarly or better than Wave_clus in all cases and performed better than Osort at SNR lower than 4. The superior performance of the STDP approach at low SNR seems to be due to a better recall (with a fewer false negatives), and, more relevantly, a better clustering [Fig. 7(d)].

3.2. Stability of the network

A spike-sorting method should generate robust classification irrespective of the level of activity of the cells. In particular, classification should be correct for highly active neurons as well as for poorly active neurons. We thus tested the STDP-based sorting method on artificial data where three neurons were simulated with different firing rates: 1 Hz for one cell, 3 Hz for another and 9 Hz for the third cell [Fig. 8(a)]. The network successfully classified the three different neurons with an overall $F1$ -score of 84%. Noticeably, the network learned faster the waveform of the most active cell and more slowly the waveform of the least active cell. We found that the network needed about 50 occurrences of a given waveform to reach a correct and stable classification. The proposed method is thus able to correctly classify different waveforms corresponding to neurons having different firing rates, as soon as each waveform has occurred enough times.

A robust spike-sorting method then requires that a classification remains correct when the firing rates of the cells fluctuate. Indeed, the activity of a neuron during a behavioral task can be subject to fluctuations depending on the involvement of the cell into the task. We thus also tested the STDP-based sorting method on artificial data where three neurons had intermittent firing activity [Fig. 8(b)]. All cells had a firing rate of 3 Hz but one became active only 50s after the others and then these two other cells became silent one after the other before all

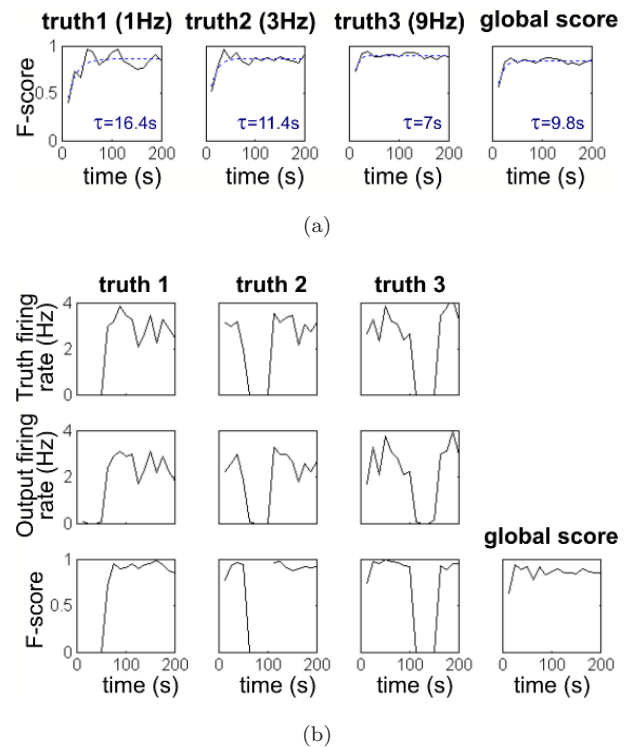


Fig. 8. (Color online) Robustness of the network to different firing rate scenarios. (a) Performance of the network along time for ground truth neurons with different firing rates. Blue dashed lines are an exponential fits. (b) Evolutions of the performance of the network for neurons with varying firing rates.

became active again. We found that these firing rate variations did not impair the algorithm performance [Fig. 8(b)]. In particular, after the network had stabilized learning of two waveforms, it could still learn a new one. Moreover, when a waveform that the network had learnt became transiently absent in the signal, it could still classify it immediately and successfully as soon as it reappeared, meaning that the network keeps memory of the waveforms that have been learnt.

3.3. Extension of the network to tetrode recording

The recent advances of neural interfacing have benefited from novel micro- and nano-technologies allowing the fabrication of high-density multi-electrode systems.^{51–54} Dense arrays of microelectrodes may provide neural recordings displaying partially overlapping information from one electrode site to the

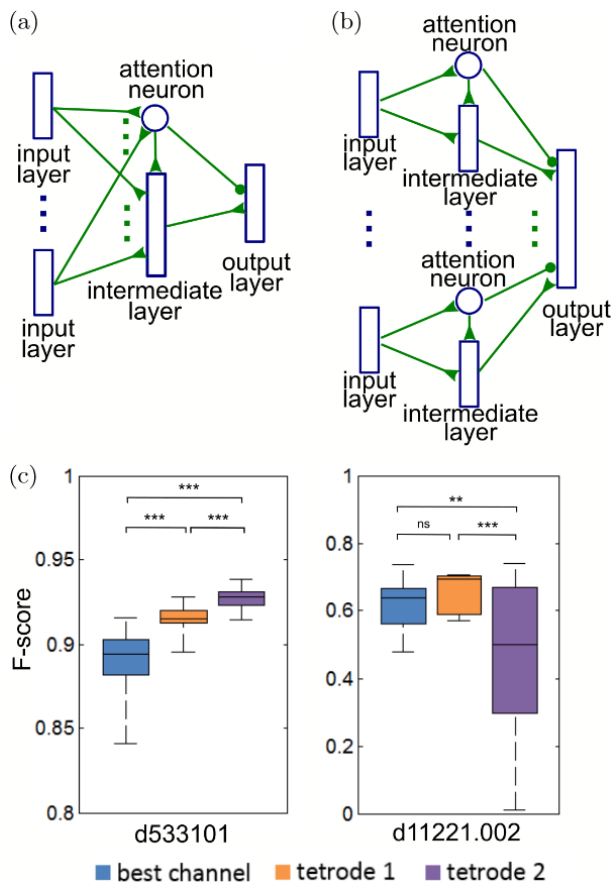


Fig. 9. Adaptation of the network for the processing of tetrode data. (a) First network adaptation to tetrodes (tetrode 1) where one input layer is used independently for each recording site and the intermediate and output layers are further shared. (b) Second adaptation (tetrode 2) where one input and one intermediate layers are used independently for each recording site and only the output layer is further shared. (c) Performance of the two different tetrode architectures compared with the best single electrode performance. “ns” stands for nonsignificant, * for $p < 0.05$, ** for $p < 0.01$ and *** for $p < 0.001$.

next. In particular, when recording sites are separated by less than a few tens of microns, action potentials from a same cell can be recorded on several sites and spike-sorting methods can benefit from combining neighboring electrodes instead of processing recording sites independently.^{30,34,55,56} In this context, we thus adapted our method to the case of tetrode recordings, processing all channels at once instead of separately. To do so, we considered two variations of our network architecture. The first one had four input layers processing in parallel each of

the four input signals and projecting to one common attention neuron and one common intermediate layer [Fig. 9(a)]. In this configuration, the attention neuron’s threshold, the intermediate neurons’ threshold and the weights of the synapses linking the attention neuron and the intermediate layer were multiplied by four. The second configuration had four input layers, four attention neurons and four intermediate layers working in parallel. The four intermediate layers and attention neurons project to one common output layer [Fig. 9(b)]. In this case the network parameters were unchanged. On the two datasets tested, we found that combining the four signals with the first configuration gave better classification than using each electrode separately. The second configuration, requiring a higher number of neurons and thus more computations, was only better for the first dataset having a higher SNR [Fig. 9(c)].

4. Conclusion

We show that a three-layer STDP network with an attention-based mechanism is successful on a spike-sorting task. The network needs only few data to configure in a fully unsupervised manner. Moreover, the parameters of the network did not need to be adjusted between the different tested datasets. Indeed, only the noise level of the input data needs to be known to parameter the network correctly (see Sec. 2.1.1). In a preliminary study,⁵⁷ extracellular action potentials could be successfully sorted in case of a very high SNR using a single-layer STDP network but this architecture failed at SNR corresponding to standard cortical recordings. Here, we propose a multi-layer network architecture incorporating several plasticity rules, which shows performances comparable to existing methods with the advantage to process the stream of neural signal continuously. In contrast to conventional spike-sorting algorithms that use separate steps for action potential detection, feature extraction and clustering, here the entire process is considered as a whole. The neural network directly processes the raw signal to output timestamps of sorted action potentials.

5. Discussion

Bio-inspired computing using artificial spiking neural networks promises performances outperforming

currently available computational approaches. Yet, the number of applications of such networks remains limited due to the lack of generic training method for complex pattern recognition. This study shows encouraging results on the application of STDP networks to spike-sorting, with classification results even better than those obtained with Osort and Wave_clus software. This latter finding yet only applies to the datasets tested in this study and would require more extensive testing on numerous situations to be confirmed. This study proposes a radically new approach to handle spike-sorting offering perspective in the long term for very-low-power embedding of this central neural signal processing step in brain implants. From the point of view of unsupervised pattern recognition using STDP networks, these results also represent significant advances with respect to previous achievements obtained so far. STDP networks have been shown to be successful for unsupervised learning¹³ and pattern recognition from sets of static images such as digits¹⁷ or objects or faces,¹⁴ with performances improved by considering deep architectures.⁵⁸ When considering dynamical inputs such as sounds or videos, pattern recognition using STDP networks has so far been restricted to cases where the time of occurrence of the pattern fragments across the different input units did not need to be taken into account.¹⁵ Indeed, an LIF neuron combined with STDP can act as a coincidence detector, which is not adequate to discriminate between patterns differing by the time ordering of their fragments across the input units. Here we solved this problem using a set of delaying synapses, to take into account time ordering. In our case, two layers with delays were used, but this mechanism could be extended to more layers, the length of the recognized time-series going up in scale at each layer. Another difficulty which had to be overcome to solve the problem of spike-sorting is that the pattern processed by the output layer might vary in size. As different patterns can share common parts, this can lead in the worst case to patterns including one into another. This situation was handled by using an intrinsic plasticity rule which allows the neuron threshold to adapt to the size of the learnt pattern, combined with synapses that can have negative as well as positive weights, which ensures that the maximum excitation is reached when the

presented pattern exactly matches the learnt pattern. It has been shown that STDP network can recognize spike patterns occurring sparsely in a random spike background.¹² However, here, we face a different problem as the background signal fluctuates around zero and thus gets encoded as consistent spike trains that could be learnt by the network. This problem was handled with an attention mechanism, which takes advantage of the fact that the relevant patterns occur only sparsely, whereas background patterns that should be ignored occur repeatedly. This attention mechanism thus detects the relevant parts of the signal and then gates the rest of the network to only learn from these relevant periods.

This approach has been designed to be compatible with a neuromorphic implementation, which opens perspectives for low-power real-time processing, thanks to the development of neuromorphic chips. Real-time processing requires fast computations, especially when hundreds of recording sites need to be processed simultaneously. Artificial STDP neural networks have been implemented on FPGAs to demonstrate embedding neuromorphic computing.^{59–62} However, although the real-time transposition of artificial STDP spike-sorting networks could be envisioned with GPUs or FPGAs, such strategies would still require high power consumptions not compatible with future embedding in implantable devices. Future brain implants embedding spike-sorting at the electrode level will thus need to rely on other types of very-low-power implementations. The approach proposed here was designed specifically in such perspective. It complies with neuromorphic circuits and in particular with scalable non-volatile resistive memory (memristive) devices that can mimic artificial synapses with embedded STDP plasticity and offer the perspective of very-low-power implementation of spiking neural networks in analog hardware for pattern recognition.^{4–10,35,63,64} Indeed, synapses are several orders of magnitude more numerous than neurons in neural networks. Thus, low-power computing based on artificial neural networks should be conceived to be compatible with low-power synapses. Here, the proposed method relies on two types of plasticity rules at the synaptic level, STP and STDP. Both have been modeled into RRAM synapses, making the method compliant with

currently developed nanoscale neuromorphic hardware.^{10,15,65,66} The more power-consuming part of our network is the synaptic connection between the input layer and the intermediate layer with $8 * 10^6$ spikes per second generated by the input layer, each transmitted through 60 synapses. With a raw estimation of about 200 pJ of energy consumed for each weight change,^{57,67} this leads to a total power of less than $10 \mu\text{W}$. This however remains a very rough estimation as, depending on the technology used, the switching power of a synapse can be decreased under 1 pJ,^{68,69} which would result in a power consumption of the network of about $0.5 \mu\text{W}$, hence $10 \mu\text{W}$ for 200 microelectrodes. This approach thus opens new perspectives to achieve very-power-efficient spike-sorting in miniaturized analog neuromorphic circuits, which should benefit to future fully implantable electronics and intelligent and standalone autonomous implants for seamless neural function monitoring and neurorehabilitation.

The present study may further open new ways to build large-scale hybrid neural networks. Hybrid networks connect real neurons to artificial neurons, ideally in a one-to-one bidirectional scheme. This field has been pioneered by dynamic clamp experiments at the single cell level using the patch-clamp technique.⁷⁰ Microelectrode arrays now open the way to build larger hybrid networks connecting multiple living cells to multiple artificial units. Achieving a complete bidirectional hybridization requires two types of transformations: converting real neural network activity into artificial neural activity and, conversely, converting artificial activity into real activity. Artificial-to-real conversion has been achieved using electrical microstimulation at a network level⁷¹ but remains unachieved at the level of one-to-one artificial-real neuron pairs. Here, the spike-sorting STDP network offers a simple solution for a real-to-artificial one-to-one conversion of the spike trains of multiple biological single units into spike trains of an equivalent number of artificial spiking neurons. Such direct real-to-artificial conversion should thus ease the exploitation of neural activity by downstream neuromorphic architectures that have been proposed for neural signal decoding in rehabilitation devices.⁷²

Data Availability

The simulated data used in this work is made available on the Zenodo platform at doi:10.5281/

zenodo.888977. The code used to simulate the STDP network is also made available on the Zenodo platform at doi:10.5281/zenodo.2248525.

Acknowledgments

This work was supported in part by the European Union's Horizon 2020 research and innovation program Under Grant Agreement No. 732032 (BrainCom), by the Fondation pour la Recherche Médicale (FRM) Under Grant Agreement No. DBS20140930785 and by the French National Research Agency Under Grant Agreement No. ANR-16-CE19-0005-01 (BrainSpeak). The authors wish to thank Elisa Vianello and Jean-François Bêche for helpful discussions, and George Malliaras for his feedback on the manuscript.

References

1. S. Ghosh-Dastidar and H. Adeli, Spiking neural networks, *Int. J. Neural Syst.* **19** (2009) 295–308.
2. W. Maass, Networks of spiking neurons: The third generation of neural network models, *Neural Netw.* **10** (1997) 1659–1671.
3. T. Wu, F.-D. Bilbâie, A. Păun, L. Pan and F. Neri, Simplified and yet Turing universal spiking neural P systems with communication on request, *Int. J. Neural Syst.* **28** (2018) 1850013, doi:10.1142/S0129065718500132.
4. S. H. Jo *et al.*, Nanoscale memristor device as synapse in neuromorphic systems, *Nano Lett.* **10** (2010) 1297–1301.
5. G. Indiveri, B. Linares-Barranco, R. Legenstein, G. Deligeorgis and T. Prodromakis, Integration of nanoscale memristor synapses in neuromorphic computing architectures, *Nanotechnology* **24** (2013) 384010.
6. G. Indiveri, F. Corradi and N. Qiao, Neuromorphic architectures for spiking deep neural networks, in *Proc. IEEE Int. Electron Devices Meeting* (2015), pp. 4.2.1–4.2.4.
7. S. Park *et al.*, Electronic system with memristive synapses for pattern recognition, *Sci. Rep.* **5** (2015) 10123.
8. S. Saïghi *et al.*, Plasticity in memristive devices for spiking neural networks, *Front. Neurosci.* **9** (2015) 51.
9. B. Rajendran and F. Alibart, Neuromorphic computing based on emerging memory technologies, *IEEE J. Emerg. Sel. Top. Circuits Syst.* **6** (2016) 198–211.
10. S. La Barbera, A. F. Vincent, D. Vuillaume, D. Querlioz and F. Alibart, Interplay of multiple synaptic plasticity features in filamentary memristive devices

- for neuromorphic computing, *Sci. Rep.* **6** (2016) 39216.
11. I. Sourikopoulos *et al.*, A 4-fJ/spike artificial neuron in 65 nm CMOS technology, *Front. Neurosci.* **11** (2017) 123.
 12. T. Masquelier, R. Guyonneau and S. J. Thorpe, Spike timing dependent plasticity finds the start of repeating patterns in continuous spike trains, *PLoS ONE* **3** (2008) e1377.
 13. T. Masquelier, R. Guyonneau and S. J. Thorpe, Competitive STDP-based spike pattern learning, *Neural Comput.* **21** (2009) 1259–1276.
 14. T. Masquelier and S. J. Thorpe, Unsupervised learning of visual features through spike timing dependent plasticity, *PLoS Comput. Biol.* **3** (2007) e31.
 15. M. Suri *et al.*, Bio-inspired stochastic computing using binary CBRAM synapses, *IEEE Trans. Electron Devices* **60** (2013) 2402–2409.
 16. N. Srinivasa and Y. Cho, Unsupervised discrimination of patterns in spiking neural networks with excitatory and inhibitory synaptic plasticity, *Front. Comput. Neurosci.* **8** (2014) 159.
 17. P. Diehl and M. Cook, Unsupervised learning of digit recognition using spike-timing-dependent plasticity, *Front. Comput. Neurosci.* **9** (2015) 99.
 18. A. Antonietti, J. Monaco, E. D’Angelo, A. Pedrocchi and C. Casellato, Dynamic redistribution of plasticity in a cerebellar spiking neural network reproducing an associative learning task perturbed by TMS, *Int. J. Neural Syst.* **28** (2018) 1850020.
 19. G. Antunes, S. F. Faria da Silva and F. M. Simoes de Souza, Mirror neurons modeled through spike-timing-dependent plasticity are affected by channelopathies associated with autism spectrum disorder, *Int. J. Neural Syst.* **28** (2018) 1750058.
 20. A. Geminiani, C. Casellato, A. Antonietti, E. D’Angelo and A. Pedrocchi, A multiple-plasticity spiking neural network embedded in a closed-loop control system to model cerebellar pathologies, *Int. J. Neural Syst.* **28** (2018) 1750017.
 21. A. Naro, A. Bramanti, A. Leo, P. Bramanti and R. S. Calabrò, Metaplasticity: A promising tool to disentangle chronic disorders of consciousness differential diagnosis, *Int. J. Neural Syst.* **28** (2018) 1750059.
 22. X. Zhang, G. Foderaro, C. Henriquez and S. Ferrari, A scalable weight-free learning algorithm for regulatory control of cell activity in spiking neuronal networks, *Int. J. Neural Syst.* **28** (2018) 1750015.
 23. S. M. Bohte, J. N. Kok and H. La Poutré, Error-backpropagation in temporally encoded networks of spiking neurons, *Neurocomputing* **48** (2002) 17–37.
 24. A. Mohemmed, S. Schliebs, S. Matsuda and N. Kasabov, SPAN: Spike pattern association neuron for learning spatio-temporal spike patterns, *Int. J. Neural Syst.* **22** (2012) 1250012.
 25. L. Guo, Z. Wang, M. Cabrerizo and M. Adjouadi, A cross-correlated delay shift supervised learning method for spiking neurons with application to interictal spike detection in epilepsy, *Int. J. Neural Syst.* **27** (2017) 1750002.
 26. U. Rutishauser, E. M. Schuman and A. N. Mamelak, Online detection and sorting of extracellularly recorded action potentials in human medial temporal lobe recordings, *in vivo*, *J. Neurosci. Methods* **154** (2006) 204–224.
 27. M. Delescluse and C. Pouzat, Efficient spike-sorting of multi-state neurons using inter-spike intervals information, *J. Neurosci. Methods* **150** (2006) 16–29.
 28. R. Q. Quiroga, Z. Nadasdy and Y. Ben-Shaul, Unsupervised spike detection and sorting with wavelets and superparamagnetic clustering, *Neural Comput.* **16** (2004) 1661–1687.
 29. S. Shoham, M. R. Fellows and R. A. Normann, Robust, automatic spike sorting using mixtures of multivariate *t*-distributions, *J. Neurosci. Methods* **127** (2003) 111–122.
 30. C. Rossant *et al.*, Spike sorting for large, dense electrode arrays, *Nat. Neurosci.* **19** (2016) 634–641.
 31. G. Hilgen *et al.*, Unsupervised spike sorting for large-scale, high-density multielectrode arrays, *Cell Rep.* **18** (2017) 2521–2532.
 32. E. Chah *et al.*, Automated spike sorting algorithm based on Laplacian eigenmaps and *k*-means clustering, *J. Neural Eng.* **8** (2011) 016006.
 33. A. Oliynyk, C. Bonifazzi, F. Montani and L. Fadiga, Automatic online spike sorting with singular value decomposition and fuzzy *C*-mean clustering, *BMC Neurosci.* **13** (2012) 96.
 34. O. Marre *et al.*, Mapping a complete neural population in the retina, *J. Neurosci.* **32** (2012) 14859–14873.
 35. B. Zhang, Z. Jiang, Q. Wang, J. Seo and M. Seok, A neuromorphic neural spike clustering processor for deep-brain sensing and stimulation systems, in *Proc. 2015 IEEE/ACM Int. Symp. Low Power Electronics and Design (ISLPED)* (IEEE, 2015), pp. 91–97, doi:10.1109/ISLPED.2015.7273496.
 36. J. E. Chung *et al.*, A fully automated approach to spike sorting, *Neuron* **95** (2017) 1381–1394.
 37. K. Seidl *et al.*, CMOS-based high-density silicon microprobe arrays for electronic depth control in intracortical neural recording-characterization and application, *J. Microelectromech. Syst.* **21** (2012) 1426–1435.
 38. A. P. Alivisatos *et al.*, The brain activity map, *Science* **339** (2013) 1284–1285.
 39. G. N. Angotzi, M. Malerba, S. Zucca and L. Berdondini, A 512-channels, whole array read-out, CMOS implantable probe for acute recordings from the brain, in *Proc. 2015 37th Annu. Int. Conf. IEEE Engineering in Medicine and*

- Biology Society (EMBC)* (IEEE, 2015), pp. 877–880, doi:10.1109/EMBC.2015.7318502.
40. F. Pothof *et al.*, Chronic neural probe for simultaneous recording of single-unit, multi-unit, and local field potential activity from multiple brain sites, *J. Neural Eng.* **13** (2016) 046006.
 41. G. Rios, E. V. Lubenov, D. Chi, M. L. Roukes and A. G. Siapas, Nanofabricated neural probes for dense 3-D recordings of brain activity, *Nano Lett.* **16** (2016) 6857–6862.
 42. C. M. Lopez *et al.*, 22.7 A 966-electrode neural probe with 384 configurable channels in 0.13 μm SOI CMOS, in *Proc. 2016 IEEE Int. Solid-State Circuits Conf. (ISSCC)* (IEEE, 2016), pp. 392–393, doi:10.1109/ISSCC.2016.7418072.
 43. J. Lee *et al.*, YASS: Yet another spike sorter, bioRxiv 151928, doi:10.1101/151928.
 44. L. F. Abbott, J. A. Varela, K. Sen and S. B. Nelson, Synaptic depression and cortical gain control, *Science* **275** (1997) 220–223.
 45. T. Natschläger and B. Ruf, Spatial and temporal pattern analysis via spiking neurons, *Netw. Comput. Neural Syst.* **9** (1998) 319–332.
 46. S. Ghosh-Dastidar and H. Adeli, Improved spiking neural networks for EEG classification and epilepsy and seizure detection, *Integr. Comput.-Aided. Eng.* **14** (2007) 187–212.
 47. S. Ghosh-Dastidar and H. Adeli, A new supervised learning algorithm for multiple spiking neural networks with application in epilepsy and seizure detection, *Neural Netw.* **22** (2009) 1419–1431.
 48. D. A. Adamos, E. K. Kosmidis and G. Theophilidis, Performance evaluation of PCA-based spike sorting algorithms, *Comput. Methods Programs Biomed.* **91** (2008) 232–244.
 49. D. A. Henze *et al.*, Intracellular features predicted by extracellular recordings in the hippocampus *in vivo*, *J. Neurophysiol.* **84** (2000) 390–400.
 50. D. Henze *et al.*, Simultaneous intracellular and extracellular recordings from hippocampus region CA1 of anesthetized rats (2009), <http://crcns.org/data-sets/hc/hc-1>.
 51. L. Berdondini *et al.*, Extracellular recordings from locally dense microelectrode arrays coupled to dissociated cortical cultures, *J. Neurosci. Methods* **177** (2009) 386–396.
 52. G. Charvet *et al.*, BioMEA: A versatile high-density 3D microelectrode array system using integrated electronics, *Biosens. Bioelectron.* **25** (2010) 1889–1896.
 53. J. Vivenzi *et al.*, Flexible, foldable, actively multiplexed, high-density electrode array for mapping brain activity *in vivo*, *Nat. Neurosci.* **14** (2011) 1599–1605.
 54. D. Khodagholy *et al.*, NeuroGrid: Recording action potentials from the surface of the brain, *Nat. Neurosci.* **18** (2014) 310–315.
 55. B. L. McNaughton, J. O’Keefe and C. A. Barnes, The stereotrode: A new technique for simultaneous isolation of several single units in the central nervous system from multiple unit records, *J. Neurosci. Methods* **8** (1983) 391–397.
 56. C. M. Gray, P. E. Maldonado, M. Wilson and B. McNaughton, Tetrodes markedly improve the reliability and yield of multiple single-unit isolation from multi-unit recordings in cat striate cortex, *J. Neurosci. Methods* **63** (1995) 43–54.
 57. T. Werner *et al.*, Spiking neural networks based on OxRAM synapses for real-time unsupervised spike sorting, *Front. Neurosci.* **10** (2016) 474.
 58. S. R. Kheradpisheh, M. Ganjtabesh, S. J. Thorpe and T. Masquelier, STDP-based spiking deep convolutional neural networks for object recognition, *Neural Netw.* **99** (2018) 56–67.
 59. J. Li, Y. Katori and T. Kohno, An FPGA-based silicon neuronal network with selectable excitability silicon neurons, *Front. Neurosci.* **6** (2012) 183.
 60. M. Ambroise, T. Levi, S. Joucla, B. Yvert and S. Saïghi, Real-time biomimetic central pattern generators in an FPGA for hybrid experiments, *Front. Neurosci.* **7** (2013) 215.
 61. L. P. Maguire *et al.*, Challenges for large-scale implementations of spiking neural networks on FPGAs, *Neurocomputing* **71** (2007) 13–29.
 62. J. L. Rossello, V. Canals, A. Morro and A. Oliver, Hardware implementation of stochastic spiking neural networks, *Int. J. Neural Syst.* **22** (2012) 1250014.
 63. P. Yao *et al.*, Face classification using electronic synapses, *Nat. Commun.* **8** (2017) 15199.
 64. M. Al-Shedivat, R. Naous, G. Cauwenberghs and K. N. Salama, Memristors empower spiking neurons with stochasticity, *IEEE J. Emerg. Sel. Top. Circuits Syst.* **5** (2015) 242–253.
 65. T. Ohno *et al.*, Short-term plasticity and long-term potentiation mimicked in single inorganic synapses, *Nat. Mater.* **10** (2011) 591–595.
 66. D. Kuzum, R. G. D. Jeyasingh, B. Lee and H. S. P. Wong, Nanoelectronic programmable synapses based on phase change materials for brain-inspired computing, *Nano Lett.* **12** (2012) 2179–2186.
 67. O. Bichler *et al.*, Visual pattern extraction using energy-efficient ‘2-PCM synapse’ neuromorphic architecture, *IEEE Trans. Electron Devices* **59** (2012) 2206–2214.
 68. F. Xiong, A. D. Liao, D. Estrada and E. Pop, Low-power switching of phase-change materials with carbon nanotube electrodes, *Science* **332** (2011) 568–570.
 69. A. Chin, C. H. Cheng, Y. C. Chiu, Z. W. Zheng and M. Liu, Ultra-low switching power RRAM using hopping conduction mechanism, *ECS Trans.* **50** (2013) 3–8.
 70. A. A. Sharp, M. B. O’Neil, L. F. Abbott and E. Marder, Dynamic clamp: Computer-generated

- conductances in real neurons, *J. Neurophysiol.* **69** (1993) 992–995.
71. S. Joucla *et al.*, Generation of locomotor-like activity in the isolated rat spinal cord using intraspinal electrical microstimulation driven by a digital neuromorphic CPG, *Front. Neurosci.* **10** (2016) 67.
72. J. Dethier *et al.*, A brain-machine interface operating with a real-time spiking neural network control algorithm, *Adv. Neural Inf. Process. Syst.* **2011** (2011) 2213–2221.