First-Spike-Based Visual Categorization Using Reward-Modulated STDP

Milad Mozafari, Saeed Reza Kheradpisheh, Timothée Masquelier, Abbas Nowzari-Dalini, and Mohammad Ganjtabesh

***Abstract*— Reinforcement learning (RL) has recently regained popularity with major achievements such as beating the European game of Go champion. Here, for the first time, we show that RL can be used efficiently to train a spiking neural network (SNN) to perform object recognition in natural images without using an external classifier. We used a feedforward convolutional SNN and a temporal coding scheme where the most strongly activated neurons fire first, while less activated ones fire later, or not at all. In the highest layers, each neuron was assigned to an object category, and it was assumed that the stimulus category was the category of the first neuron to fire. If this assumption was correct, the neuron was rewarded, i.e., spike-timing-dependent plasticity (STDP) was applied, which reinforced the neuron's selectivity. Otherwise, anti-STDP was applied, which encouraged the neuron to learn something else. As demonstrated on various image data sets (Caltech, ETH-80, and NORB), this reward-modulated STDP (R-STDP) approach has extracted particularly discriminative visual features, whereas classic unsupervised STDP extracts any feature that consistently repeats. As a result, R-STDP has outperformed STDP on these data sets. Furthermore, R-STDP is suitable for online learning and can adapt to drastic changes such as label permutations. Finally, it is worth mentioning that both feature extraction and classification were done with spikes, using at most one spike per neuron. Thus, the network is hardware friendly and energy efficient.**

***Index Terms*— First-spike-based categorization, reinforcement learning (RL), reward-modulated spike-timing-dependent plasticity (R-STDP), spiking neural networks (SNNs), temporal coding, visual object recognition.**

Manuscript received May 25, 2017; revised November 8, 2017 and January 28, 2018; accepted April 9, 2018. This work was supported in part by the European Research Council through the European Union's 7th Framework Program (FP/20072013)/ERC under Grant 323711 (M4 project), in part by the Iran National Science Foundation under Grant 96005286, and in part by the Institute for Research in Fundamental Sciences, Tehran, Iran, under Grant BS-1396-02-02. *(Corresponding author: Mohammad Ganjtabesh.)*

M. Mozafari, S. R. Kheradpisheh, and M. Ganjtabesh are with the Department of Computer Science, School of Mathematics, Statistics, and Computer Science, University of Tehran, Tehran 1417614411, Iran, and also with the School of Biological Sciences, Institute for Research in Fundamental Sciences, Tehran, Iran (e-mail: milad.mozafari@ut.ac.ir; kheradpisheh@ut.ac.ir; mgtabesh@ut.ac.ir).

T. Masquelier is with CerCo UMR 5549, Centre national de la recherche scientifique, Université Toulouse 3, 31062 Toulouse Cedex 9, France (e-mail: timothee.masquelier@cnrs.fr).

A. Nowzari-Dalini is with the Department of Computer Science, School of Mathematics, Statistics, and Computer Science, University of Tehran, Tehran 1417614411, Iran (e-mail: nowzari@ut.ac.ir).

This paper has supplementary downloadable material available at http://ieeexplore.ieee.org, provided by the author.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNNLS.2018.2826721

I. Introduction

NEURONS in the brain are connected by synapses that can be strengthened or weakened over time. The neural mechanisms behind long-term synaptic plasticity, which is crucial for learning, have been under investigation for many years. Spike-timing-dependent plasticity (STDP) is an unsupervised form of synaptic plasticity, observed in different brain areas [1]–[4], in particular in the visual cortex [5]–[7]. STDP works by considering the time difference between presynaptic and postsynaptic spikes. According to this rule, if the presynaptic neuron fires earlier (later) than the postsynaptic one, the synapse is strengthened (weakened). Studies have shown that STDP results in coincidence detectors, by which a neuron gets selective to a frequent input spike pattern leading to an action potential whenever the pattern is presented [8]–[11]. STDP works well in finding statistically frequent features; however, as any unsupervised learning algorithm, it faces with difficulties in detecting rare but diagnostic features for important functionalities such as decision-making.

Several studies suggest that the brain's reward system plays a vital role in decision-making and forming behaviors. This is also known as reinforcement learning (RL), by which the learner is encouraged to repeat rewarding behaviors and avoid those leading to punishments [12]–[18]. It is found that dopamine, as a neuromodulator, is one of the important chemical substances involved in the reward system [19], where its release is proportional to the expected future reward [17], [20], [21]. It is also shown that dopamine as well as some other neuromodulators influences the synaptic plasticity, such as changing the polarity [22] or adjusting the time window of STDP [23]–[27].

One of the well-studied ideas to model the role of the reward system is to modulate or even reverse the weight change determined by STDP, which is called reward-modulated STDP (R-STDP) [28]. R-STDP stores the trace of synapses that are eligible for STDP and applies the modulated weight changes at the time of receiving a modulatory signal: a reward or punishment (negative reward).

In 2007, Izhikevich [29] proposed an R-STDP rule to solve the distal reward problem, where the reward is not immediately received. He solved the problem using a decaying eligibility trace by which the recent activities are considered to be more important. He showed that his model can solve both classical and instrumental conditionings [30], [31]. In the same year, Farries and Fairhall [32] employed R-STDP to train neurons

for generating particular spike patterns. They measured the difference between the output and target spike trains to compute the value of the reward. Also, Florian [33] showed that R-STDP is able to solve the XOR task by either rate or temporal input coding and learning a target firing rate. A year later, Legenstein *et al.* [34] investigated conditions, under which R-STDP achieves a desired learning effect. They demonstrated the advantages of R-STDP by a theoretical analysis, as well as practical applications to biofeedbacks and a two-class isolated spoken digit recognition task. Vasilaki *et al.* [35] examined the idea of R-STDP on problems with continuous space. They showed that their model is able to solve the Morris water maze quite fast, while the standard policy gradient rule is failed. Investigating capabilities of R-STDP continued by research from Frémaux *et al.* [36], in which conditions for a successful learning is theoretically discussed. They showed that a prediction of the expected reward is necessary for R-STDP to learn multiple tasks simultaneously. Studying the RL mechanism in the brain has gathered attention in recent years, and researchers try to solve more practical tasks by reward-modulated synaptic plasticity [37]–[39].

Visual object recognition is a sophisticated task, at which humans are expert. This task requires both feature extraction that is done by the brain's visual cortex and decision-making on the category of the object, for which higher brain areas are involved. Spiking neural networks (SNNs) have been widely used in computational object recognition models. In terms of network architecture, there are several models with shallow [40]–[43], deep [44]–[46], recurrent [47], fully connected [48], and convolutional structures [40], [46], [49], [50]. Some use rate-based coding [51]–[53], while others use the temporal coding [40], [43], [46], [48], [54]. Various kinds of learning techniques are also applied to SNNs, from backpropagation [49], [55], tempotron [43], [56], and other supervised techniques [52], [53], [57], [58], to unsupervised STDP and STDP-variants [42], [48], [59]. Although STDP-enabled networks provide a more biological plausible means of visual feature extraction, they need an external readout, e.g., support vector machines (SVM) [46], [60], to classify input stimuli. In addition, STDP tends to extract frequent features that are not necessarily suitable for the desired task. In this paper, we present a hierarchical SNN equipped with R-STDP to solve the visual object recognition in natural images without using any external classifier. Instead, we put class-specific neurons that are reinforced to fire as early as possible if their target stimulus is presented to the network. Thus, the input stimuli are classified solely based on the first-spike latencies in a fast and biologically plausible way. R-STDP enables our network to find task-specific diagnostic features, therefore decreasing the computational cost of the final recognition system.

Our network is based on Masquelier and Thorpe's model [40] with four layers. The first layer of the network converts the input image into spike latencies based on the saliency of its oriented edges. This spike train goes under a local pooling operation in the second layer. The third layer of the network includes several grids of integrate-and-fire (IF) neurons that combine the received information of oriented edges and extract complex features. This is the only trainable layer in our network which employs R-STDP for synaptic plasticity. The signal (reward/punishment) for modulation of synaptic plasticity is provided by the fourth layer, in which the decision of the network is made. Our network only uses the earliest spike emitted by the neurons in the third layer to make a decision without using any external classifier. If its decision is correct (incorrect), a global reward (punishment) signal is generated. Besides, in order to increase the computational efficiency, each cell in the network is allowed to spike only once per image. The motivation for at most one spike per neuron is not only computational efficiency, but it is also biological realism [61], [62]. Decision-making without any classifiers with at most one spike per neuron turns the proposed method into a well-suited candidate for the hardware implementation.

We performed two toy experiments to illustrate the abilities of R-STDP. We showed that the network employing R-STDP finds informative features using fewer computational resources than STDP. We also showed that R-STDP can change the behavior of a neuron, if needed, by encouraging it to unlearn what it has learned before. Thus, reusing a computational resource is no longer useful. Moreover, we evaluated the proposed network on object recognition in natural images, using three different benchmarks, that are Caltech face/motorbike (two classes), ETH-80 (eight classes), and NORB (five classes). The results of the experiments demonstrate the advantage of employing R-STDP over STDP in finding task-specific discriminative features. Our network has reached the performances (recognition accuracies) of 98.9% on Caltech face/motorbike, 89.5% on ETH-80, and 88.4% on NORB data sets.

The rest of this paper is organized as follows. A precise description of the proposed network is provided in Section II. Then, in Section III, the results of the experiments are presented. Finally, in Section IV, the proposed network is discussed from different points of view and the possible future works are highlighted.

II. Materials and Methods

In this section, we first describe the structure of the proposed network and the functionality of each layer. We then explain R-STDP, by which the neurons achieve reinforced selectivity to a specific group of input stimuli. Finally, we give a detailed description on the classification strategy that is used to evaluate the network's performance.

A. Overall Structure

Similar to Masquelier and Thorpe's model [40], our network consists of two simple and two complex layers that are alternately arranged in a feedforward manner (see Fig. 1).

The first layer of the network ($S1$) is a simple layer whose cells detect oriented edges in the input image. These cells emit a spike with a latency that is inversely proportional to the saliency of the edge. After $S1$, there is a complex layer ($C1$), which introduces some degrees of position invariance by applying local pooling operation. A $C1$ neuron propagates the earliest spike in its input window.

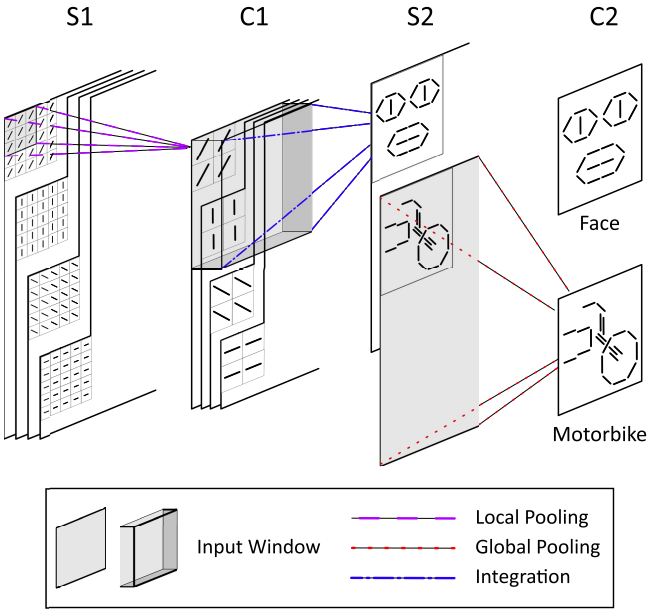


Fig. 1. Overall structure of the proposed network with four retinotopically organized layers. The first layer ($S1$) extracts oriented edges from the input image by applying Gabor filters. A local max-pooling operation is applied by the cells in the subsequent layer ($C1$) to gain some degrees of position invariance. From here, spikes are propagated by the latencies that are inversely proportional to the maximum values. These spikes are the inputs for the IF neurons in the layer $S2$ that are equipped with the R-STDP learning rule. These neurons are encouraged/punished to learn/unlearn complex features. The activity of $S2$ neurons is used by $C2$ neurons for decision-making. These neurons are associated with class labels and the decision is made based on the neuron with the earliest spike.

The second simple layer ($S2$) is made of IF neurons. A neuron in this layer, which detects a complex feature, receives its inputs from $C1$ neurons and generates a spike once its membrane potential reaches the threshold. For synaptic plasticity, we use a learning rule based on three factors: 1) presynaptic spike time; 2) postsynaptic spike time; and 3) a reward/punishment signal. This kind of synaptic plasticity provides the ability to control the behavior of the neurons in terms of their selectivity to input patterns.

The second complex layer ($C2$) of our network is the decision-making layer. Each neuron in this layer is assigned to a category and performs a global pooling operation over $S2$ neurons in a particular grid. Using a rank-order decoding scheme, the neuron that fires first indicates the network's decision about the input image. According to the decision made by the network, a reward/punishment signal is then generated, which drives in the synaptic plasticity of $S2$ neurons.

Implementation of the network is mainly done with C# and the code is available on ModelDB.[1]

B. Layer S1

The goal of this layer is to extract oriented edges from the grayscaled input image and turn them into spike latencies. To this end, the input image is convolved with Gabor filters of four different orientations. Thus, this layer includes four feature maps, each representing the saliency of edges in a particularly preferred orientation.

[1]https://senselab.med.yale.edu/ModelDB/

Let I be the grayscaled input image and $G(\theta)$ represent a Gabor filter (convolution kernel) with window size 5×5, wavelength 2.5, effective width 2, and orientation θ. Then, the lth feature map of layer $S1$ is generated using the following equations:

$$S_1^l = |I \otimes G(\theta_l)|$$
$$\theta_l = \frac{(l-1) \times \pi}{4} + \frac{\pi}{8} \tag{1}$$

where $\otimes$ is the convolution operator and $l \in \{1, 2, 3, 4\}$. In order to introduce invariance to image negative operation, the absolute value of the convolution is used. Also, since vertical and horizontal edges are very common in natural images, a $(\pi/8)$ offset is applied to relax this bias [40].

For each of the feature maps (orientations), we put a 2-D grid of the same size containing dummy neurons to propagate spikes. Using an intensity-to-latency encoding scheme, the obtained feature maps are converted to the spike latencies that are inversely proportional to the saliency of edges. In other words, the more salient the edge is, the earlier the corresponding spike is propagated.

We implemented the proposed network in an event-based manner, where the spikes are sorted by their latencies in the ascending order and propagated sequentially (i.e., the first spike is propagated in time step $t = 1$, the second one in $t = 2$, and so on).

C. Layer C1

Our first complex layer is a local pooling layer over the spikes coming from layer $S1$. Here, there are four 2-D neuronal grids corresponding to each of the orientations. Each $C1$ neuron performs a local pooling operation over a window of size $\omega_{c1} \times \omega_{c1}$ and stride r_{c1} (here we set $r_{c1} = \omega_{c1} - 1$) on $S1$ neurons in a particular grid, after which it emits a spike immediately after receiving its earliest input spike. This pooling operation decreases the redundancy of layer $S1$, and shrinks the number of required neurons, which consequently increases the computational efficiency. It also adds a local invariance to the position of oriented edges.

Let $\mathcal{P}_{c1}(i)$ be the set of all presynaptic neurons of the ith neuron in layer $C1$. Then, the firing time of this neuron is computed as follows:

$$t_{c1}^f(i) = \min_{j \in \mathcal{P}_{c1}(i)} \left\{ t_{s1}^f(j) \right\} \tag{2}$$

where $t_{s1}^f(j)$ denotes the firing time of the jth neuron in $\mathcal{P}_{c1}(i)$.

In addition, two kinds of lateral inhibition mechanisms are employed, which help the network to propagate more salient information. If a neuron located at position (x, y) of the ith grid (orientation) fires: 1) the other neurons at the same position, but in other grids, are prevented from firing and 2) the latencies of the nearby neurons in the same grid are increased by a factor relative to their mutual Euclidean distance. In our experiments, inhibition is done for distances from 1 to 5 pixel(s) (floating-point distances are truncated to integer values) with inhibition factors 15%, 12%, 10%, 7%, and 5%, respectively.

D. Layer $S2$

This layer combines the incoming information about oriented edges and turns them into meaningful complex features. Here, there are n 2-D grids of IF neurons with the threshold $\mathcal{T}$. Each neuron receives its inputs from a $\omega_{s2} \times \omega_{s2} \times 4$ window of $C1$ neurons through the plastic synapses. A weight sharing mechanism is also applied to the neurons belonging to the same grid. This mechanism provides the ability of detecting a particular feature over the entire spatial positions. To be precise, let $\mathcal{P}_{s2}(i)$ be the set of all presynaptic neurons corresponding to the ith neuron. Then, the membrane potential of this neuron at time step t is updated by the following equation:

$$v_i(t) = v_i(t-1) + \sum_{j \in \mathcal{P}_{s2}(i)} W_{ij} \times \delta\big(t - t_{c1}^f(j)\big) \tag{3}$$

where W_{ij} denotes the synaptic weight, δ is the Kronecker delta function, and $t_{c1}^f(j)$ is the firing time of the jth cell in layer $C1$. For each input image, a neuron in $S2$ fires if its membrane potential reaches the threshold $\mathcal{T}$. Also, these neurons have no leakage and are allowed to fire at most once, while an image is being presented.

As the neurons fire, their synaptic weights—the feature they are detecting—are being updated based on the order of presynaptic and postsynaptic spikes, as well as a reward/punishment signal (see Section II-F). This signal is derived from the activity of the next layer, which indicates the network's decision. Besides, initial weights of the synapses are randomly generated, with mean 0.8 and standard deviation 0.05. Note that choosing small or midrange values for mean results in inactive, thus untrained, neurons. Moreover, large values for variance increase the impact of network's initial state. Accordingly, a high mean value with small variance is a suitable choice [46].

E. Layer $C2$

This layer contains exactly n neurons, and each is assigned to one of the $S2$ neuronal grids. A $C2$ neuron only propagates the first spike that is received from its corresponding neuronal grid. To put it differently, let $\mathcal{P}_{c2}(i)$ define the set of $S2$ neurons in the ith neuronal grid (for $i \in \{1, 2, \ldots, n\}$). Then, the firing time of the ith $C2$ neuron is computed as follows:

$$t_{c2}^f(i) = \min_{j \in \mathcal{P}_{c2}(i)} \big\{t_{s2}^f(j)\big\} \tag{4}$$

where $t_{s2}^f(j)$ denotes the firing time of the jth neuron in layer $S2$.

As mentioned before, the activity of $C2$ neurons indicates the decision of the network. To this end, we divide $C2$ neurons into several groups and assign each group to a particular category of input stimuli. Then, the network's decision on the category of the input stimulus is assumed to be the one whose group propagates the earliest spike among other $C2$ groups.

Assume that there are m distinct categories for the input stimuli, labeled from 1 to m, and n neuronal grids in layer $S2$. Accordingly, there are exactly n neurons in layer $C2$ that are divided into m groups. Let $g : \{1, 2, \ldots, n\} \mapsto \{1, 2, \ldots, m\}$ denote a function that returns the group's index of a $C2$ neuron, and let $t_{c2}^f(i)$ denote the firing time of the ith neuron in layer $C2$. Then, the network's decision $\mathcal{D}$ is made by

$$\begin{aligned} \mathcal{F} &= \arg\min_i \big\{t_{c2}^f(i) | 1 \le i \le n\big\} \\ \mathcal{D} &= g(\mathcal{F}) \end{aligned} \tag{5}$$

where $\mathcal{F}$ is the index of a $C2$ neuron which fires first. The network receives reward (punishment) if its decision matches (does not match) the correct category of the input stimulus. If none of the $C2$ neurons fire, no reward/punishment signal is generated, and thus, no weight change is applied. Moreover, if more than one neuron fire early (with the minimum spike time), the one with the minimum index (i) is selected.

F. Reward-Modulated STDP

We propose an RL mechanism to update the presynaptic weights of $S2$ neurons. Here, the magnitude of weight change is modulated by a reward/punishment signal, which is received according to the correctness/incorrectness of the network's decision. We also applied a one-winner-takes-all learning competition among the $S2$ neurons, by which the one with the earliest spike is the winner and the only one which updates its synaptic weights. Note that this neuron is the one determining the network's decision.

To formulate our R-STDP learning rule, if a reward signal is received, then

$$\Delta W_{ij} = \begin{cases} a_r^+ \times W_{ij} \times (1 - W_{ij}) & \text{if } t_{c1}^f(j) - t_{s2}^f(i) \le 0 \\ a_r^- \times W_{ij} \times (1 - W_{ij}) & \text{if } t_{c1}^f(j) - t_{s2}^f(i) > 0 \\ & \text{or the } j\text{th cell is silent} \end{cases} \tag{6}$$

and in case of receiving a punishment signal, we have

$$\Delta W_{ij} = \begin{cases} a_p^+ \times W_{ij} \times (1 - W_{ij}) & \text{if } t_{c1}^f(j) - t_{s2}^f(i) > 0 \\ & \text{or the } j\text{th cell is silent} \\ a_p^- \times W_{ij} \times (1 - W_{ij}) & \text{if } t_{c1}^f(j) - t_{s2}^f(i) \le 0 \end{cases} \tag{7}$$

where i and j refer to the postsynaptic and presynaptic cells, respectively, ΔW_{ij} is the amount of weight change for the synapse connecting the two neurons, and a_r^+, a_r^-, a_p^+, and a_p^- scale the magnitude of weight change. Furthermore, to specify the direction of weight change, we set $a_r^+, a_p^+ > 0$ and $a_r^-, a_p^- < 0$. Here, our learning rule does not take into account the exact spike time difference and uses an infinite time window. According to this learning rule, the punishment signal reverses the polarity of STDP (also known as anti-STDP). In other words, it swaps long-term-depression (LTD) with long-term potentiation (LTP), which is done to conduct the effect of aversion (avoid repeating a bad behavior), and a_p^+ is there to encourage the neuron to learn something else.

G. Overfitting Avoidance

In RL problems, there is a chance of being trapped into local optima or overfitting to acquiring the maximum possible reward over the training examples. In order to help the network, exploring other possible solutions that are more general to cover both seen and unseen examples, we apply two additional mechanisms during the training phase. These techniques are only used for object recognition tasks.

1) Adaptive Learning Rate: Since the initial weights of the neurons are randomly set, the number of misclassified samples is relatively high at the beginning of the training phase (i.e., the performance is at the chance level). As training trials go on, the ratio of correctly classified samples to the misclassified ones increases. In the case of high rate of misclassification, the network receives more punishment signals, which rapidly weakens synaptic weights and generates dead or highly selective neurons that cover a small number of inputs. Similarly, when the rate of correct classification gets higher, the rate of reward acquisition increases as well. In this case, the network prefers to exclude misclassified samples by getting more and more selective to correct ones and remain silent for the others. In either case, the overfitting happens due to the unbalanced impact of reward and punishment.

To tackle this problem, we multiply an adjustment factor to the amount of weight modification, by which the impact of correct and incorrect training samples is balanced over the trials. Assume that the network sees all of the training samples on each training iteration and let N_{hit} and N_{miss} denote the number of samples that are classified correctly and incorrectly in the last training iteration, respectively. If N is the number of all training samples, then the weight changes for the current training trial are modified as follows:

$$W_{ij} = W_{ij} + \begin{cases} \left(\frac{N_{\text{miss}}}{N}\right) \Delta W_{ij} & \text{if a reward is received} \\ \left(\frac{N_{\text{hit}}}{N}\right) \Delta W_{ij} & \text{otherwise.} \end{cases} \tag{8}$$

Note that $N_{\text{hit}} + N_{\text{miss}} \leq N$, since there may be some samples for which none of the $S2$ neurons is active.

2) Dropout: In an RL scenario, the goal of the learner is to maximize the expected value of reward acquisition. In our case, since the network only sees the training samples, it may find a few number of features that are sufficient to correctly classify almost all of the training samples. This issue appears to cause severe overfitting in face of complex problems and the network prefers to leave some of the neurons untrained. These neurons decrease the hit rate of the network over the testing samples, as they blindly fire for almost all of the stimuli.

Here, we employ the dropout technique [63], which causes a $C2$ neuron to be temporary turned OFF with the probability of p_{drop}. This technique gives rise to the overall involvement rate of the neurons, which in turn, not only increases the chance of finding more discriminative features but also decreases the rate of blind firings (see Dropout in the Supplementary Material).

H. Classification

As mentioned before, the activity of the last layer, particularly the earliest spike in layer $C2$, is the only information that our network uses to make its final decision on the input stimuli. In this way, we do not need external classifiers and increase the biological plausibility of the network at the same time.

To set up the network for a classification task with m categories, we put $n = k \times m$ neuronal grids in layer $S2$, where k is the number of features associated with each category. Then, we assign each $C2$ neurons to a category by the association function $g : \{1, 2, \ldots, n\} \mapsto \{1, 2, \ldots, m\}$ defined as follows:

$$g(i) = \lfloor (i-1)/k \rfloor + 1. \tag{9}$$

Then, the network uses (5) to classify the input stimuli. During the training phase, each network's decision is compared with the label of stimulus and a reward (punishment) signal is generated, if the decision matches (mismatches) the label.

I. Comparison of R-STDP and STDP

In object recognition tasks, we make a comparison between our model, SNN with R-STDP, and the one that uses STDP. To this end, we first train the network using STDP and let the network extract features in an unsupervised manner. Next, we compute three kinds of feature vectors of length n from layer $S2$.

1) *First-Spike Vector:* This is a binary vector, in which all the values are zeros except the one corresponding to the neuronal grid with earliest spike.
2) *Spike-Count Vector:* This vector saves the total number of spikes emitted by neurons in each grid.
3) *Potential Vector:* This vector contains the maximum membrane potential among the neurons in each grid by ignoring the threshold.

After extracting feature vectors for both training and testing sets, K-nearest neighbors (KNN) and SVM classifiers are used to evaluate the performance of the network. Moreover, the learning strategy and the STDP formula are the same as in [40], and to make a fair comparison, we use the same values for parameters in both the models. The only parameters that are explored for the STDP are the magnitudes of LTP and LTD.

III. Results

To evaluate the proposed network and learning strategy, we performed two types of experiments. First, we used a series of hand-made problems to show the superiority of R-STDP over STDP. Second, we assessed the proposed network on several object recognition benchmarks.

A. R-STDP Increases Computational Efficiency

Using STDP, when a neuron is exposed to input spike patterns, it tends to find the earliest repetitive subpattern by which the neuron reaches its threshold and fires [8], [11], [64], [65]. This tendency to favor early input spikes can be troublesome in case of distinguishing spike patterns that have temporal differences in their late parts.

Assume that there are several categories of input stimuli that possess the same spatial configuration [Fig. 2(a)]. They also have identical early spikes. These patterns are repetitively presented to a group of IF neurons, for which the synaptic plasticity is governed by STDP and the one-winner-takes-all mechanism. If the neurons have low thresholds, one of them gets selective to the early common part of the input stimuli and inhibits the other neurons. Since the early parts are spatio-temporally the same among all of the input stimuli, there is

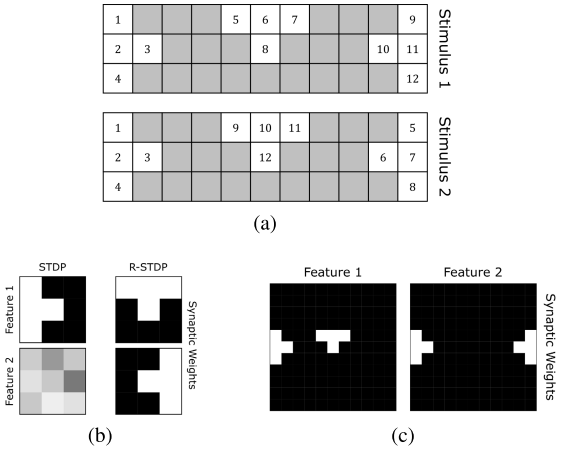


Fig. 2. Temporal discrimination task. (a) Two input stimuli including a temporally different subpattern. Spikes are propagated from the white squares with the order written on them. (b) Synaptic weights (features) that are learned by the neurons with STDP (left column) and R-STDP (right column). The higher the weight, the lighter is the gray level. (c) Synaptic weights when we used STDP-enabled neurons with large receptive fields and high thresholds.

no chance for the other neurons to fire and win the synaptic plasticity. Consequently, the overall activity of the neuronal group is the same for all of the input stimuli and classifies them into a single category.

As we will see in Fig. 2(c), there are also some STDP-based solutions for this problem, and however, they are inefficient in using computational resources. For example, if we increase the size of receptive fields along with the thresholds, neurons gain the opportunity to receive the last spikes as well as the early ones. Another possible solution is to use many neurons that locally inhibit each other and drop the one-winner-takes-all constraint. In this way, regarding the initial random weights, there is a chance for the neurons to learn other parts of the input stimuli.

Here, we show that the R-STDP learning rule solves this issue in a more efficient way than STDP. For this purpose, we designed an experiment containing two 3×11 input stimuli. The inputs are spatially similar, which means that spikes are propagated from similar locations of both inputs. As illustrated in Fig. 2(a), each input is a 2-D grid of white and gray squares. By white (gray) squares, we denote locations, from which a spike is (is not) propagated. At the time of presenting any of these patterns to the network, spikes are propagated with a temporal order that is defined by the numbers written on the squares. According to this ordering, spikes with lower numbers are propagated earlier.

Since the input stimuli are artificial spike patterns, there was no need to apply Gabor filters, and thus, they were fed directly into layer $S2$. There, we put two neuronal grids with parameters $\omega_{s2} = 3$ and $\mathcal{T} = 3$. Therefore, each grid has contained 1×9 neurons to cover the entire input stimuli. We also set $a_r^+ = 0.05, a_r^- = -0.05, a_p^+ = 0.1$, and $a_p^- = -0.1$. The goal of the task was that the first (second) $C2$ neuron fires earlier for the first (second) pattern. We examined both STDP and R-STDP learning rules to see if the network finds discriminative features or not.

As shown in Fig. 2(b), using STDP, the network has extracted a nondiscriminative feature, the shared one between both input stimuli. On the other side, the proposed RL mechanism has guided the neurons to extract features whose temporal order of appearance is the only thing leading to a successful pattern discrimination. We repeated this experiment for 100 times using different random initial weights. The results showed that our network has succeeded in 98% of the times, while there were no chance for STDP to find the discriminative features. When we increased the threshold to 4 (requiring at least two subpatterns) and the size of the receptive fields to 11×11 (covering the entire pattern), the network employing the STDP could also find discriminative features [see Fig. 2(c)] in 80% of the times.

B. Plastic Neurons

As mentioned earlier, the brain reward system plays an important role in the emergence of a particular behavior. In this section, we demonstrate the R-STDP's capability of readjusting neurons' behavior in an online manner.

We designed an experiment, in which the predefined desired behavior of the neurons is changed during the simulation. The experimental setup is very similar to the "temporal discrimination" task with similar input stimuli and parameter values, except that we swapped the target input stimuli during the training iterations [see Tasks 1 and 2 in Fig. 3(a)]. As shown in Fig. 3(b), at the beginning of the simulation, the desired behavior was that the neurons belonging to the first grid respond to the first stimulus earlier than those in the second grid, and vice versa. After 200 iterations, when the convergence is fulfilled, we swapped the target stimuli. At this stage, since the neurons were exclusively sensitive to the previous target stimuli, they began to generate false alarms. Consequently, the network was receiving high rates of punishments for around 80 iterations [see iterations 200 to 280 in Fig. 3(b)], which in turn swapped LTD and LTP (see Sections II-F). As the network has received punishments, the previously weakened (strengthened) synapses got stronger (weaker). Therefore, the sensitivity diminished for a while, and the neurons have regained the possibility of learning something new. After iteration 300, neurons found their new target stimulus and, once again, converged to the discriminative features [see the plots of synaptic weights in the top two rows in Fig. 3(b)].

In summary, R-STDP enables the neurons to unlearn what they have learned so far. This ability results in neurons with a flexible behavior (plastic neurons) that are able to learn rewarding behavior in changing environments. This ability also helps the neurons to forget and escape from the local optima in order to learn something that earns more reward. Applying STDP in such a scenario does not work at all, since there is no difference between Tasks 1 and 2 from an unsupervised point of view.

C. Object Recognition

In this section, the performance of our network on categorization of natural images is evaluated. We begin with

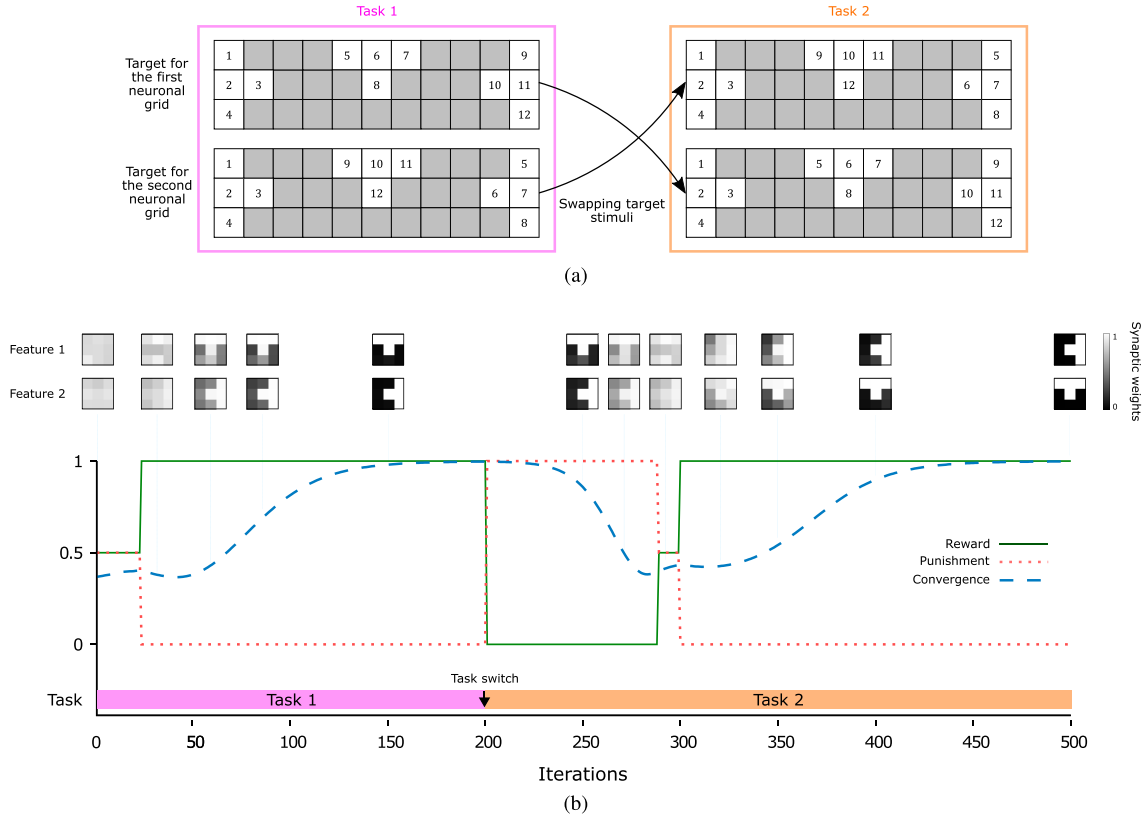


Fig. 3. Neurons with a flexible behavior. (a) Target stimuli for each neuronal grid. In Task 2, the target stimuli of Task 1 are swapped. (b) Flexibility of the network in a changing environment. Top two rows represent changes of the synaptic weights. Bottom row illustrates changes in the rate of receiving reward (green solid line), punishment (red dotted line), and the convergence of the synaptic weights (blue dashed line) over 500 iterations. Convergence is measured by $1-(\sum W_{ij}(1-W_{ij})/18)$, where the summation is over the 18 $(3\times 3+3\times 3)$ synaptic weights that are shared between $S2$ neurons. As this value gets closer to one, the synaptic weights have more binarylike values.

a description of the data sets that are used in our experiments. Then, we show how the network benefits from the RL mechanism to extract features from natural images, followed by comparing R-STDP and STDP in object recognition tasks. Finally, we illustrate how the dropout and adaptive learning techniques reduce the chance of overfitting to the training samples.

1) Data Sets: We used three well-known object recognition benchmarks to evaluate the performance of the proposed network. The first and easiest one is Caltech face/motorbike which is mainly used for demonstration purposes. The next two that are used to evaluate the proposed network are ETH-80 and small NORB. These data sets contain images of objects from different viewpoints which make the task harder (see Fig. S1 in the Supplementary Material).

2) Reinforced Selectivity: The previous experiments showed that R-STDP enables the network to find informative and discriminative features, both spatially and temporally. Here, we show that R-STDP encourages the neurons to become selective to a particular category of natural images. To this end, we trained and examined the network on images from two categories of face and motorbike from the Caltech data set.

In this experiment, we put 10 neuronal grids for each category that were reinforced to win the first-spike competition in response to the images from their target categories. Therefore, the desired behavior of the network was that the neurons of the first 10 grids get selective to the face category, while those in the other grids get selective to the motorbikes.

Fig. 4 shows the behavior of the network over the training iterations. Since the early iterations have contained rapid changes, they are plotted wider. During early iterations, strong synaptic weights (see Section II-D) and 50% dropout probability are resulted in an unstable network whose neurons are responded to random input stimuli. This chaotic behavior can be easily spotted on early iterations in the middle plot [see Fig. 4(b)]. As the network continues training iterations, reward/punishment signals made neurons more and more selective to their target categories. As shown in Fig. 4(b), after 200 iterations, a quite robust selectivity is appeared for the training samples, while on the testing samples, it is elongated for 300 more iterations. This quick convergence on training samples is due to the fact that the network is relatively fast in finding features that successfully discriminate seen samples [see Fig. 4(a)]. These primary features need to converge more to be applicable on testing samples, which requires even more iterations because of the adaptive learning rates. Moreover, we do not let the learning rate that drops below 20% of the values of parameters a_r^+, a_r^-, a_p^+, and a_p^-. This allows the network to continue convergence with a constant rate even if all of the training samples are correctly categorized [see Fig. 4(c)].

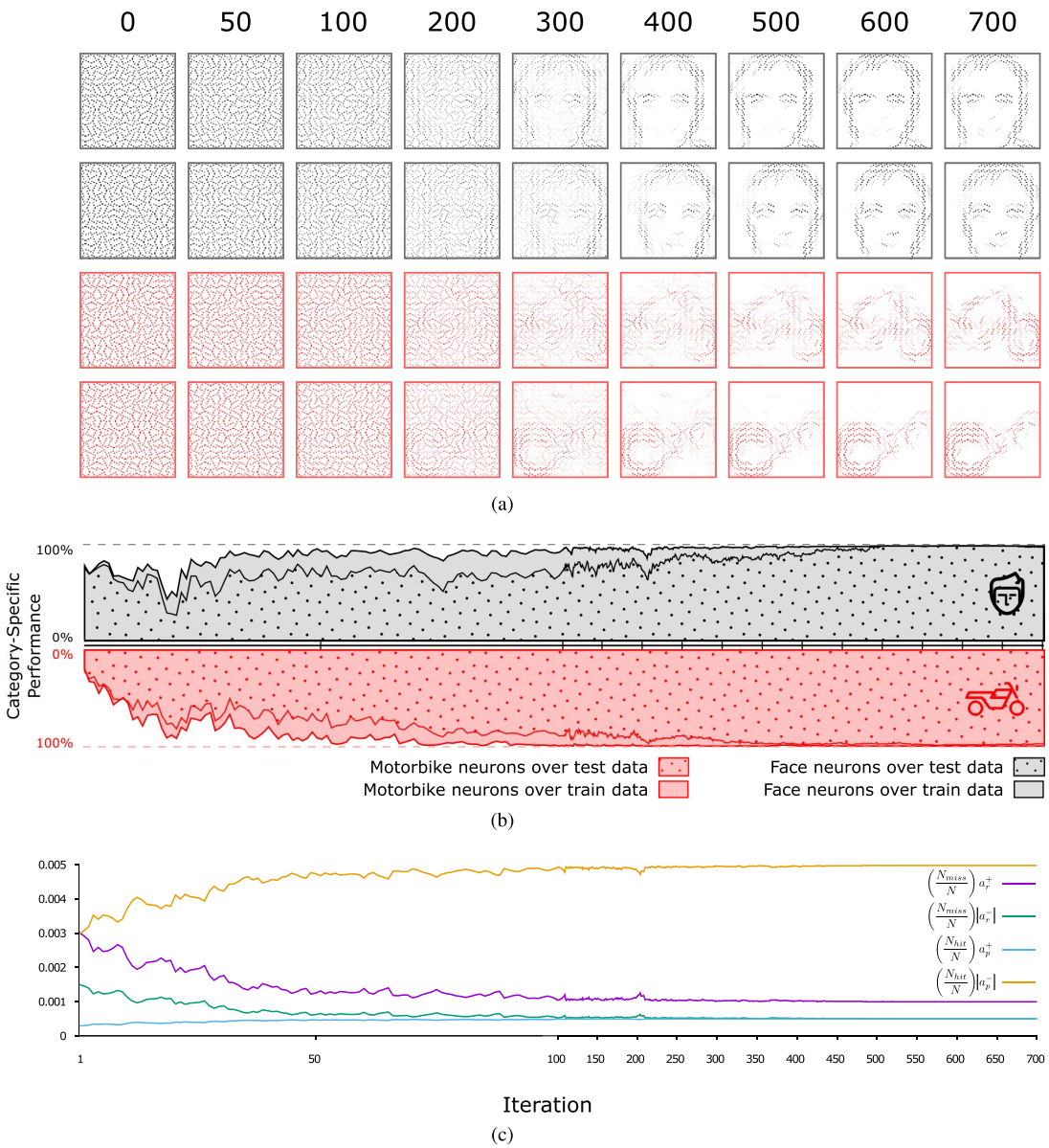


Fig. 4. Training the network on the Caltech face/motorbike data set. (a) Evolution of four different features (out of 20) extracted by the network. The black and red plots correspond to the face and motorbike neurons, respectively. (b) Hit rate for neurons of each category. The gray (pink) filled curves depict the percentage of the times that the face (motorbike) neurons emit the earliest spike in response to their target stimulus. Notice that curves for motorbike neurons are mirrored vertically for the sake of better illustration, and hit rates over testing set are indicated by dot patterns. (c) Trajectory of changes in learning rate with respect to the number of correct (N_{hit}) and incorrect (N_{miss}) categorizations.

We repeated the experiment 30 times with random initial weights and different training and testing samples and the performance achieved by the proposed network is $98.9 \pm 0.4\%$ (mean $\pm$ std). When we tried the same network structure with STDP, 97.2% was its best achievement (see Table I).

3) Performance: We have shown how the proposed network has successfully classified faces from motorbikes with high accuracy. Here, we examined the performance of the proposed network on the ETH-80 and NORB data sets that are more challenging (see Datasets in the Supplementary Material). The performance of the network is tested over the entire testing set after each training iteration, in which the network receives all of the training samples in the random order.

For ETH-80 data set, we configured the network to extract 10 features per category, which have resulted in $8 \times 10 = 80$ features in total. The receptive field of each neuron in layer $S2$ was set in a way that it covered the whole input image. Here, nine instances of each category were presented to the network as the training samples, and the remaining were employed in the test phase. After performing 250 training and testing iterations, the best testing performance of the network was reported.

Again, we repeated this experiment 30 times, each time using a different training and testing set. As before, the network successfully has extracted discriminative features (see Fig. 2 in the SupplementaryMaterial) and reached the performance of $89.5 \pm 1.9\%$ (mean $\pm$ std). We also applied STDP to a network with the same structure. To examine the STDP performance, we used SVMs with linear kernel and KNNs (K was changed from 1 to 10). According to

TABLE I
COMPARISON OF THE NETWORK'S PERFORMANCE WHEN USING R-STDP AND STDP

Dataset	R-STDP	STDP						Shallow CNN
		First-Spike		Spike-Count		Max-Potential		
		SVM	KNN	SVM	KNN	SVM	KNN	
Caltech (Face/Motorbike)	98.9	96.4	96.4	96.9	93.4	96.6	97.2	99.3
ETH-80	89.5	72.9	69.8	74	70.4	79.9	84.5	87.1
NORB	88.4	62.7	58.6	61.7	55.3	66	65.9	85.5

the results, the accuracy achieved by this network is 84.5%, when the maximum potentials were used as the feature vectors and the classifier was KNN. Considering that the proposed network classifies input patterns solely based on the first-spike information, R-STDP definitely outperforms STDP. Table I provides the details of the comparison made between R-STDP and STDP.

By looking at confusion matrices [see Fig. 3(a) in the Supplementary Material], we found that both R-STDP and STDP agree on the most confusing categories that are cow, dog, and horse. However, thanks to the RL, R-STDP not only has decreased the confusion error but also provided a more balanced error distribution.

The same experiment was also performed on the NORB data set. Again, we put 10 neuronal grids for each of the five categories, whose neurons are able to see the entire incoming stimuli. The proposed network with R-STDP has reached the performance of $88.4 \pm 0.5\%$ (mean $\pm$ std) on testing samples, whereas STDP has achieved 66% at most. By reviewing confusion matrices of both methods, we found that both networks have encountered difficulties mostly in distinguishing four-leg animals from humans, as well as cars from trucks [see Fig. 3(b) in the Supplementary Material]. As before, R-STDP has resulted in a more balanced error distribution.

In addition, we compared the proposed network to convolutional neural networks (CNNs). Although the proposed network is not able to beat pretrained deep CNNs (DCNN) such as VGG16 [66] (see Comparison with Deep Convolutional Neural Networks in the Supplementary Material), comparing it to a shallow CNN with a similar network structure and the same input would be a fair point. We repeated all of the object categorization experiments using a shallow CNN implemented with Keras neural networks API and Tensorflow as its backend. As shown in Table I, the proposed network has successfully outperformed the supervised CNN in both the ETH-80 and NORB data sets.

4) Overfitting Problem: Overfitting is one of the most common issues in supervised or RL scenarios. This problem got even worse by the emergence of deep learning algorithms. There are many studies focused on developing techniques that increase the generalization power of the learning algorithms. One of the mechanism that has shown promising empirical results on deep neural networks is the dropout technique [63]. This technique temporarily reduces the complexity of the network by suppressing the activity of a specific number of neurons. This reduction in neuronal resources forces the network to generalize more in order to reduce the prediction error.

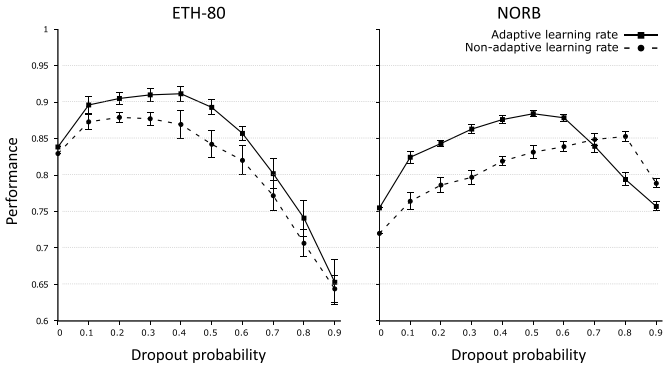


Fig. 5. Impact of the dropout and the adaptive learning rate techniques. The plot on the left (right) demonstrates the result for ETH-80 (NORB) data set. In these plots, the solid (dashed) lines illustrate the performance of the network with different dropout probabilities when the adaptive learning rate is on (off).

The proposed network is not an exception and has shown tendencies to overfit on the training samples through our examinations. Therefore, we adopted the dropout technique in our experiments. We also found that a steady learning rate does increase the chance of overfitting. Thus, we made use of dynamic learning rates with respect to the performance of the network (see Section II-G).

To show the impact of the aforementioned mechanisms, we repeated the object recognition experiments with different dropout probabilities and steady learning rates. Fig. 5 simultaneously shows the impact of both mentioned mechanisms on categorization of test samples. It is clear that when the adaptive learning rate mechanism is applied, the network has achieved higher performances (solid lines). It is also shown that the dropout probability must be chosen according to the complexity of the data set as well as the network. Since the NORB data set contains more complex samples than the ETH-80, it tends more to overfitting on training samples. As a consequence, it needs more dropout rate to overcome this issue. The magnitude of this tendency is even clearer when the steady learning rates are used. To put it differently, faster convergence rate along with the complexity of the samples induces more overfitting, which in turn needs more dropout rate.

IV. DISCUSSION

Mammals are fast and accurate at visual object recognition. Their visual cortex processes the incoming data in a hierarchical manner, through which the complexity of neuronal preference is gradually increased. This hierarchical processing provides a robust and invariant object recognition [67]–[71]. Computational modeling of the mammalian visual cortex has

been under investigation for many years. Developing a biologically plausible model not only enables scientists to examine their hypotheses with low cost but also provides a humanlike vision for artificially intelligent machines [40], [72]–[75].

DCNNs are the most successful works in this area [63], [66], [76]–[78]. The idea behind these networks is inspired by the hierarchical structure of the visual cortex. Despite the promising results obtain by DCNNs, they are not biologically plausible because of using supervised learning rules. In addition, they employ rate-based encoding scheme, which is both energy and resource consuming. There is another group of studies trying to use spiking neurons along with the unsupervised STDP learning rule [40], [42], [46], [48], [59]. These models are more biologically plausible, but they cannot beat DCNNs in terms of accuracy. In theory, SNNs have more computational power than DCNNs, and however, they are harder to control because of the complex dynamics and high-dimensional space of effective parameter. Furthermore, since most of them are trained in an unsupervised manner, the classification step is done by an external classifier or statistical methods.

Here, we solved the object recognition task using a hierarchical SNN equipped with an RL rule called R-STDP [28]. There are several studies showing that the brain uses RL to solve the problem of decision-making [15]–[18]. Therefore, it is a suitable choice for training class-specific neurons that are able to decide on the class of the input image. Therefore, we put one step further developing a more biologically plausible model which is able to perform the visual categorization totally on its own. The proposed network functions in the temporal domain, where the information is encoded by spike times. The input image is first convolved with oriented Gabor filters and a spike train is generated based on a latency-to-intensity coding scheme. The resulting spikes are then propagated toward the feature extraction layer. Using R-STDP, the proposed network successfully found task-specific diagnostic features using neurons that were preassigned to the class labels. In other words, each neuron was assigned to a class *a priori*, where its desired behavior was to respond early for the instances belonging to the specified class. To decrease the computational cost even more, neurons were forced to fire at most once for an input image and the latency of their spike is considered as the measure of stimulus preference. Therefore, if a neuron fired earlier than the others, it would have received its preferred stimulus. This measure of preference served as an indicator for the network's decision. That is to say, when a neuron belonging to a particular class fired earlier, the network's decision was considered to be that class.

Through our experiments, we compared R-STDP to STDP from different aspects. We showed that R-STDP can save computational resources. This was clarified by a hand-designed discrimination task, in which the order of spikes was the only discriminative feature. R-STDP has solved the problem using a minimal number of neurons, synapses, and threshold, whereas STDP has needed more neurons, more synapses, and higher thresholds. This drawback for STDP is due to the fact that it tends to find statistically frequent features [8]–[11], which are not necessarily the diagnostic ones. As a consequence, one needs to use either more neurons or more synapses to ensure that the diagnostic features will be eventually found. On the other hand, since R-STDP informs the neurons about their outcomes, they can function better using minimal resources.

After having demonstrated the advantages of R-STDP in finding diagnostic features, we investigated how well it can be combined with a hierarchical SNN for solving both visual feature extraction and object categorization in a biologically plausible manner. We evaluated the proposed network and a similar network which uses STDP, as well as a CNN with the same structure, on three data sets of natural images Caltech Face/Motorbike, ETH-80, and NORB. The last two contain images of objects from different viewpoints, which made the task harder. When we compared the performances obtained by the networks, we found that R-STDP strongly outperforms STDP and the CNN with the same structure. An even more interesting point is that the proposed network has achieved this superiority decisions solely based on the first-spikes, while in the case of the others, even the powerful classifiers like SVMs and error backpropagation were not of any help.

To compare R-STDP with STDP, both networks have used the same values for parameters except the learning rate (see Section II-I). However, one can use STDP with a higher number of neurons and tuned thresholds to compensate the blind unsupervised feature extraction and achieve better performances [60]. Again, we conclude that R-STDP helps the network to act more efficiently in consuming computational resources.

Putting everything together, the proposed network has the following prominent features.

1) Robust object recognition in natural images.
2) Each neuron is allowed to spike only once per image. This results in a huge reduction of energy consumption.
3) Decision-making (classification) is performed using the first-spike latencies instead of powerful classifiers. Therefore, the biological plausibility of the model is increased.
4) Synaptic plasticity is governed by RL (the R-STDP rule), for which supporting biological evidence can be found [28], and which allows to extract highly diagnostic features.

Our network can be interesting for neuromorphic engineering [79], since it is both biologically plausible and hardware friendly. Although hardware implementation and efficiency is out of the scope of this paper, we believe that the proposed network can be implemented in hardware in an energy-efficient manner for several reasons. First, SNNs are more hardware friendly than classic artificial neural networks, because the energy-consuming "multiply-accumulator" units can be replaced by more energy-efficient "accumulator" units. For this reason, studies on training deep convolutional SNNs (DCSNNs) [44], [46] and converting DCNNs into DCSNNs [80] as well as restricted DCNNs [81]–[83] have gained interests in recent years. Second, most SNN hardwares use event-driven approaches by considering spikes as events. In this way, energy consumption increases with the number of spikes. Thus, by allowing at most one spike per neuron, the proposed model is as efficient as possible.

Finally, the proposed learning rule is more suitable for online, on-chip learning than error backpropagation in deep networks, where updating weights based on high-precision gradients brings difficulties for hardware implementation.

To date, we could not find any other works possessing the aforementioned features. To mention one of the closest attempts, Gardner *et al.* [84] tried to classify Poisson-distributed spike trains by a readout neuron equipped with R-STDP. Although their method is working, it cannot be applied on natural images as it is, because of their time-based encoding and target labeling. There is another related work by Huerta and Nowotny [85], where the authors designed a model of the RL mechanism which occurs in the mushroom body. They applied their RL mechanism on a pool of randomly connected neurons with 10 readout neurons to classify handwritten digits. This paper is different from theirs in several aspects. First, we used a hierarchical structure based on the mammalian visual cortex, while they used randomly connected neurons. Second, we used the R-STDP learning rule, whereas they employed a probabilistic approach for the synaptic plasticity. Third, the input of our network was natural images using intensity-to-latency encoding, while they used binary encoding with a threshold on artificial images.

Although the results of the proposed network were significantly better than the network employing STDP with external classifiers, they are still not competitive to the state-of-the-art deep learning approaches. One of the limitations to the current method is using only one trainable layer. Besides, the receptive field of the neurons in the last layer is set to be large enough to cover an informative portion of the image. As a result, the network cannot resist high rates of variations in the object, unless using more and more number of neurons. Extending the number of layers in the current network is one of the directions for future research. Going deeper seems to improve the performance by providing a gradual simple to complex feature extraction. However, deeper structure needs more parameter tuning and a suitable multi-layer synaptic plasticity rule. Recent studies have also shown that combining deep networks and RL can lead to outstanding results [86], [87].

Another direction for the future research is to use the RL for learning semantic associations. For example, STDP is able to extract features for different kinds of animals in different viewpoints, but it is not able of relating all of them into the category of "animal," because different animals have no reason to co-occur. Or, it can extract features for the frontal and profile face, but it cannot generate an association putting both in the general category of "face." On the other hand, by a reinforcement signal and using learning rules like R-STDP, neurons are not only able to extract diagnostic features but also learn relative connections between categories and create supercategories.

ACKNOWLEDGMENT

The authors would like to thank Dr. J.-P. Jaffrézou for proofreading this paper and A. Yousefzadeh and Dr. B. Linares-Barranco for providing useful hardware-related information.

REFERENCES

[1] W. Gerstner, R. Kempter, J. L. van Hemmen, and H. Wagner, "A neuronal learning rule for sub-millisecond temporal coding," *Nature*, vol. 383, no. 6595, pp. 76–78, 1996.
[2] H. Markram, J. Lübke, M. Frotscher, and B. Sakmann, "Regulation of synaptic efficacy by coincidence of postsynaptic APs and EPSPs," *Science*, vol. 275, no. 5297, pp. 213–215, 1997.
[3] G.-Q. Bi and M.-M. Poo, "Synaptic modifications in cultured hippocampal neurons: Dependence on spike timing, synaptic strength, and postsynaptic cell type," *J. Neurosci.*, vol. 18, no. 24, pp. 10464–10472, 1998.
[4] P. J. Sjöström, G. G. Turrigiano, and S. B. Nelson, "Rate, timing, and cooperativity jointly determine cortical synaptic plasticity," *Neuron*, vol. 32, no. 6, pp. 1149–1164, 2001.
[5] C. D. Meliza and Y. Dan, "Receptive-field modification in rat visual cortex induced by paired visual stimulation and single-cell spiking," *Neuron*, vol. 49, no. 2, pp. 183–189, 2006.
[6] S. Huang *et al.*, "Associative Hebbian synaptic plasticity in primate visual cortex," *J. Neurosci.*, vol. 34, no. 22, pp. 7575–7579, 2014.
[7] Y. Guo *et al.*, "Timing-dependent LTP and LTD in mouse primary visual cortex following different visual deprivation models," *PLoS ONE*, vol. 12, no. 5, p. e0176603, 2017.
[8] T. Masquelier, R. Guyonneau, and S. J. Thorpe, "Spike timing dependent plasticity finds the start of repeating patterns in continuous spike trains," *PLoS ONE*, vol. 3, no. 1, p. e1377, 2008.
[9] M. Gilson, T. Masquelier, and E. Hugues, "STDP allows fast rate-modulated coding with poisson-like spike trains," *PLoS Comput. Biol.*, vol. 7, no. 10, p. e1002231, 2011.
[10] R. Brette, "Computing with neural synchrony," *PLoS Comput. Biol.*, vol. 8, no. 6, p. e1002561, 2012.
[11] T. Masquelier, "STDP allows close-to-optimal spatiotemporal spike pattern detection by single coincidence detector neurons," *Neuroscience*, in press, 2017, doi: 10.1016/j.neuroscience.2017.06.032.
[12] R. S. Sutton and A. G. Barto, *Introduction to Reinforcement Learning*, vol. 135. Cambridge, MA, USA: MIT Press, 1998.
[13] P. Dayan and B. W. Balleine, "Reward, motivation, and reinforcement learning," *Neuron*, vol. 36, no. 2, pp. 285–298, 2002.
[14] N. D. Daw and K. Doya, "The computational neurobiology of learning and reward," *Current Opinion Neurobiol.*, vol. 16, no. 2, pp. 199–204, 2006.
[15] Y. Niv, "Reinforcement learning in the brain," *J. Math. Psychol.*, vol. 53, no. 3, pp. 139–154, 2009.
[16] D. Lee, H. Seo, and M. W. Jung, "Neural basis of reinforcement learning and decision making," *Annu. Rev. Neurosci.*, vol. 35, pp. 287–308, Mar. 2012.
[17] E. E. Steinberg, R. Keiflin, J. R. Boivin, I. B. Witten, K. Deisseroth, and P. H. Janak, "A causal link between prediction errors, dopamine neurons and learning," *Nature Neurosci.*, vol. 16, no. 7, pp. 966–973, 2013.
[18] W. Schultz, "Neuronal reward and decision signals: From theories to data," *Physiol. Rev.*, vol. 95, no. 3, pp. 853–951, 2015.
[19] W. Schultz, "Getting formal with dopamine and reward," *Neuron*, vol. 36, no. 2, pp. 241–263, 2002.
[20] W. Schultz, "Predictive reward signal of dopamine neurons," *J. Neurophysiol.*, vol. 80, no. 1, pp. 1–27, 1998.
[21] P. W. Glimcher, "Understanding dopamine and reinforcement learning: The dopamine reward prediction error hypothesis," *Proc. Nat. Acad. Sci. USA*, vol. 108, pp. 15647–15654, Sep. 2011.
[22] G. H. Seol *et al.*, "Neuromodulators control the polarity of spike-timing-dependent synaptic plasticity," *Neuron*, vol. 55, no. 6, pp. 919–929, 2007.
[23] Q. Gu, "Neuromodulatory transmitter systems in the cortex and their role in cortical plasticity," *Neuroscience*, vol. 111, no. 4, pp. 815–835, 2002.
[24] J. N. Reynolds and J. R. Wickens, "Dopamine-dependent plasticity of corticostriatal synapses," *Neural Netw.*, vol. 15, no. 4, pp. 507–521, 2002.
[25] J.-C. Zhang, P.-M. Lau, and G.-Q. Bi, "Gain in sensitivity and loss in temporal contrast of STDP by dopaminergic modulation at hippocampal synapses," *Proc. Nat. Acad. Sci. USA*, vol. 106, no. 31, p. 13028–13033, 2009.
[26] E. Marder, "Neuromodulation of neuronal circuits: Back to the future," *Neuron*, vol. 76, no. 1, pp. 1–11, 2012.
[27] F. Nadim and D. Bucher, "Neuromodulation of neurons and synapses," *Current Opinion Neurobiol.*, vol. 29, pp. 48–56, Dec. 2014.
[28] N. Frémaux and W. Gerstner, "Neuromodulated spike-timing-dependent plasticity, and theory of three-factor learning rules," *Frontiers Neural Circuits*, vol. 9, p. 85, Jan. 2016.

[29] E. M. Izhikevich, "Solving the distal reward problem through linkage of STDP and dopamine signaling," *Cerebral Cortex*, vol. 17, no. 10, pp. 2443–2452, 2007.

[30] I. P. Pavlov and G. V. Anrep, *Conditioned Reflexes*. North Chelmsford, MA, USA: Courier Corporation, 2003.

[31] E. L. Thorndike, "Review of animal intelligence: An experimental study of the associative processes in animals," *Psychol. Rev.*, vol. 5, no. 5, pp. 551–553, 1898.

[32] M. A. Farries and A. L. Fairhall, "Reinforcement learning with modulated spike timing–dependent synaptic plasticity," *J. Neurophysiol.*, vol. 98, no. 6, pp. 3648–3665, 2007.

[33] R. V. Florian, "Reinforcement learning through modulation of spike-timing-dependent synaptic plasticity," *Neural Comput.*, vol. 19, no. 6, pp. 1468–1502, 2007.

[34] R. Legenstein, D. Pecevski, and W. Maass, "A learning theory for reward-modulated spike-timing-dependent plasticity with application to biofeedback," *PLoS Comput. Biol.*, vol. 4, no. 10, p. e1000180, 2008.

[35] E. Vasilaki, N. Frémaux, R. Urbanczik, W. Senn, and W. Gerstner, "Spike-based reinforcement learning in continuous state and action space: When policy gradient methods fail," *PLoS Comput. Biol.*, vol. 5, no. 12, p. e1000586, 2009.

[36] N. Frémaux, H. Sprekeler, and W. Gerstner, "Functional requirements for reward-modulated spike-timing-dependent plasticity," *J. Neurosci.*, vol. 30, no. 40, pp. 13326–13337, 2010.

[37] J. Friedrich, R. Urbanczik, and W. Senn, "Spatio-temporal credit assignment in neuronal population learning," *PLoS Comput. Biol.*, vol. 7, no. 6, p. e1002092, 2011.

[38] N. Frémaux, H. Sprekeler, and W. Gerstner, "Reinforcement learning using a continuous time actor-critic framework with spiking neurons," *PLoS Comput. Biol.*, vol. 9, no. 4, p. e1003024, 2013.

[39] G. M. Hoerzer, R. Legenstein, and W. Maass, "Emergence of complex computational structures from chaotic neural networks through reward-modulated Hebbian learning," *Cerebral Cortex*, vol. 24, no. 3, pp. 677–690, 2014.

[40] T. Masquelier and S. J. Thorpe, "Unsupervised learning of visual features through spike timing dependent plasticity," *PLoS Comput. Biol.*, vol. 3, no. 2, p. e31, 2007.

[41] J. Brader, W. Senn, and S. Fusi, "Learning real-world stimuli in a neural network with spike-driven synaptic dynamics," *Neural Comput.*, vol. 19, no. 11, pp. 2881–2912, 2007.

[42] D. Querlioz, O. Bichler, P. Dollfus, and C. Gamrat, "Immunity to device variations in a spiking neural network with memristive nanodevices," *IEEE Trans. Nanotechnol.*, vol. 12, no. 3, pp. 288–295, May 2013.

[43] Q. Yu, H. Tang, K. C. Tan, and H. Li, "Rapid feedforward computation by temporal encoding and learning with spiking neurons," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 10, pp. 1539–1552, Oct. 2013.

[44] J. H. Lee, T. Delbruck, and M. Pfeiffer, "Training deep spiking neural networks using backpropagation," *Frontiers Neurosci.*, vol. 10, p. 508, Nov. 2016.

[45] P. O'Connor and M. Welling. (2016). "Deep spiking networks." [Online]. Available: https://arxiv.org/abs/1602.08323

[46] S. R. Kheradpisheh, M. Ganjtabesh, S. J. Thorpe, and T. Masquelier, "Stdp-based spiking deep convolutional neural networks for object recognition," *Neural Netw.*, vol. 99, pp. 56–67, Mar. 2017.

[47] J. Thiele, P. U. Diehl, and M. Cook. (2017). "A wake-sleep algorithm for recurrent, spiking neural networks." [Online]. Available: https://arxiv.org/abs/1703.06290

[48] P. U. Diehl and M. Cook, "Unsupervised learning of digit recognition using spike-timing-dependent plasticity," *Frontiers Comput. Neurosci.*, vol. 9, p. 99, Aug. 2015.

[49] Y. Cao, Y. Chen, and D. Khosla, "Spiking deep convolutional neural networks for energy-efficient object recognition," *Int. J. Comput. Vis.*, vol. 113, no. 1, pp. 54–66, 2015.

[50] A. Tavanaei and A. S. Maida. (2016). "Bio-inspired spiking convolutional neural network using layer-wise sparse coding and STDP learning." [Online]. Available: https://arxiv.org/abs/1611.03000

[51] P. Merolla, J. Arthur, F. Akopyan, N. Imam, R. Manohar, and D. S. Modha, "A digital neurosynaptic core using embedded crossbar memory with 45pJ per spike in 45 nm," in *Proc. IEEE Custom Integr. Circuits Conf. (CICC)*, Sep. 2011, pp. 1–4.

[52] S. Hussain, S.-C. Liu, and A. Basu, "Improved margin multi-class classification using dendritic neurons with morphological learning," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, Jun. 2014, pp. 2640–2643.

[53] P. O'Connor, D. Neil, S.-C. Liu, T. Delbruck, and M. Pfeiffer, "Real-time classification and sensor fusion with a spiking deep belief network," *Frontiers Neurosci.*, vol. 7, p. 178, Oct. 2013.

[54] M. Beyeler, N. D. Dutt, and J. L. Krichmar, "Categorization and decision-making in a neurobiologically plausible spiking network using a STDP-like learning rule," *Neural Netw.*, vol. 48, pp. 109–124, Dec. 2013.

[55] P. U. Diehl, D. Neil, J. Binas, M. Cook, S.-C. Liu, and M. Pfeiffer, "Fast-classifying, high-accuracy spiking deep networks through weight and threshold balancing," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2015, pp. 1–8.

[56] B. Zhao, R. Ding, S. Chen, B. Linares-Barranco, and H. Tang, "Feedforward categorization on AER motion events using cortex-like features in a spiking neural network," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 9, pp. 1963–1978, Sep. 2015.

[57] F. Ponulak and A. Kasiński, "Supervised learning in spiking neural networks with ReSuMe: Sequence learning, classification, and spike shifting," *Neural Comput.*, vol. 22, no. 2, pp. 467–510, 2010.

[58] E. Neftci, S. Das, B. Pedroni, K. Kreutz-Delgado, and G. Cauwenberghs, "Event-driven contrastive divergence for spiking neuromorphic systems," *Frontiers Neurosci.*, vol. 7, p. 272, Jan. 2013.

[59] A. Tavanaei, T. Masquelier, and A. S. Maida, "Acquisition of visual features through probabilistic spike-timing-dependent plasticity," in *Proc. IEEE Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2016, pp. 307–314.

[60] S. R. Kheradpisheh, M. Ganjtabesh, and T. Masquelier, "Bio-inspired unsupervised learning of visual features leads to robust invariant object recognition," *Neurocomputing*, vol. 205, pp. 382–392, Sep. 2016.

[61] S. J. Thorpe and M. Imbert, "Biological constraints on connectionist models," in *Connectionism in Perspective*, R. Pfeifer, Z. Schreter, F. Fogelman-Soulié, and L. Steels, Eds. New York, NY, USA: Elsevier, 1989, pp. 63–92.

[62] R. VanRullen and S. J. Thorpe, "Surfing a spike wave down the ventral stream," *Vis. Res.*, vol. 42, no. 23, pp. 2593–2615, 2002.

[63] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[64] S. Song, K. D. Miller, and L. F. Abbott, "Competitive Hebbian learning through spike-timing-dependent synaptic plasticity," *Nature Neurosci.*, vol. 3, no. 9, pp. 919–926, 2000.

[65] R. Guyonneau, R. VanRullen, and S. Thorpe, "Neurons tune to the earliest spikes through STDP," *Neural Comput.*, vol. 17, no. 4, pp. 859–879, Apr. 2005.

[66] K. Simonyan and A. Zisserman. (2014). "Very deep convolutional networks for large-scale image recognition." [Online]. Available: https://arxiv.org/abs/1409.1556

[67] S. Thorpe, D. Fize, and C. Marlot, "Speed of processing in the human visual system," *Nature*, vol. 381, no. 6582, p. 520, 1996.

[68] C. Hung, G. Kreiman, T. Poggio, and J. DiCarlo, "Fast readout of object identity from macaque inferior temporal cortex," *Science*, vol. 310, no. 5749, pp. 863–866, 2005.

[69] J. J. DiCarlo and D. D. Cox, "Untangling invariant object recognition," *Trends Cognit. Sci.*, vol. 11, no. 8, pp. 333–341, 2007.

[70] H. Liu, Y. Agam, J. R. Madsen, and G. Kreiman, "Timing, timing, timing: Fast decoding of object information from intracranial field potentials in human visual cortex," *Neuron*, vol. 62, no. 2, pp. 281–290, 2009.

[71] J. J. DiCarlo, D. Zoccolan, and N. C. Rust, "How does the brain solve visual object recognition?" *Neuron*, vol. 73, no. 3, pp. 415–434, 2012.

[72] K. Fukushima and S. Miyake, "Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition," in *Competition Cooperation Neural Nets*. New York, NY, USA: Springer, 1982, pp. 267–285.

[73] Y. LeCun and Y. Bengio, "Convolutional networks for images, speech, and time series," in *The Handbook of Brain Theory and Neural Networks*, M. A. Arbib, Ed. Cambridge, MA, USA: MIT Press, 1998, pp. 255–258. [Online]. Available: http://dl.acm.org/citation.cfm?id=303568.303704

[74] T. Serre, L. Wolf, S. Bileschi, M. Riesenhuber, and T. Poggio, "Robust object recognition with cortex-like mechanisms," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 3, pp. 411–426, Mar. 2007.

[75] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng, "Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations," in *Proc. 26th Annu. Int. Conf. Mach. Learn.*, 2009, pp. 609–616.

[76] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 818–833.
[77] S. R. Kheradpisheh, M. Ghodrati, M. Ganjtabesh, and T. Masquelier, "Humans and deep networks largely agree on which kinds of variation make object recognition harder," *Frontiers Comput. Neurosci.*, vol. 10, no. 74, p. 92, 2016.
[78] S. R. Kheradpisheh, M. Ghodrati, M. Ganjtabesh, and T. Masquelier, "Deep networks can resemble human feed-forward vision in invariant object recognition," *Sci. Rep.*, vol. 6, p. 32672, Sep. 2016.
[79] S. Furber, "Large-scale neuromorphic computing systems," *J. Neural Eng.*, vol. 13, no. 5, p. 051001, 2016.
[80] B. Rueckauer, I.-A. Lungu, Y. Hu, M. Pfeiffer, and S.-C. Liu, "Conversion of continuous-valued deep networks to efficient event-driven networks for image classification," *Frontiers Neurosci.*, vol. 11, p. 682, 2017. [Online]. Available: https://www.frontiersin.org/article/10.3389/fnins.2017.00682, doi: 10.3389/fnins.2017.00682.
[81] M. Courbariaux, Y. Bengio, and J.-P. David, "Binaryconnect: Training deep neural networks with binary weights during propagations," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 3123–3131.
[82] J. Binas, G. Indiveri, and M. Pfeiffer. (2016). "Deep counter networks for asynchronous event-based processing." [Online]. Available: https://arxiv.org/abs/1611.00710
[83] S. K. Esser *et al.*, "Convolutional networks for fast, energy-efficient neuromorphic computing," *Proc. Nat. Acad. Sci. USA*, vol. 113, no. 41, pp. 11441–11446, 2016.
[84] B. Gardner, I. Sporea, and A. Grüning, "Classifying spike patterns by reward-modulated STDP," in *Proc. Int. Conf. Artif. Neural Netw.*, 2014, pp. 749–756.
[85] R. Huerta and T. Nowotny, "Fast and robust learning by reinforcement signals: Explorations in the insect brain," *Neural Comput.*, vol. 21, no. 8, pp. 2123–2151, 2009.
[86] V. Mnih *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, pp. 529–533, 2015.
[87] D. Silver *et al.*, "Mastering the game of Go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484–489, 2016.

Milad Mozafari received the B.Sc. degree from Shahed University, Tehran, Iran, and the M.Sc. degree from the Amirkabir University of Technology, Tehran. He is currently pursuing the Ph.D. degree in computer science with the University of Tehran, Tehran.

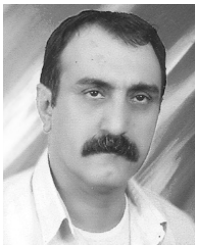
His current research interests include computational neuroscience and specifically investigates vision and reinforcement learning in the human brain.

Saeed Reza Kheradpisheh received the Ph.D. degree in computer science from the University of Tehran, Tehran, Iran, in 2017.

His current research interests include computational neuroscience, spiking neural networks, and deep learning.

Timothée Masquelier received the joint Ph.D. degree from the Centrale Paris, Gif-sur-Yvette, France, the Massachusetts Institute of Technology, Cambridge, MA, USA, and Université Toulouse 3, Toulouse, France, in 2008.

He joined the Centre national de la recherche scientifique, Paris, France, in 2012. He is currently a Computational Neuroscientist. He uses simulations and calculations to understand how neurons compute with spikes. His current research interests include bioinspired computer vision and neuromorphic engineering.



Abbas Nowzari-Dalini is currently an Associate Professor with the School of Mathematics, Statistics and Computer Science, University of Tehran, Tehran, Iran. His current research interests include bioinformatics, combinatorial algorithm, parallel algorithms, DNA computing, neural networks, and computer networks.

Mohammad Ganjtabesh received the B.Sc. degree in pure mathematics and the M.Sc. and Ph.D. degrees in computer science from the University of Tehran, Tehran, Iran, in 2001, 2003, 2008, respectively.

He is currently a Professor with the University of Tehran since 2008. His current research interests include computational neuroscience (vision) and bioinformatics (RNA structures).