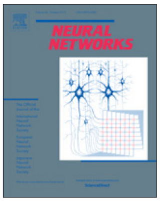
Neural Networks 105 (2018) 294–303

Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet



Representation learning using event-based STDP

Amirhossein Tavanaei [a,*], Timothée Masquelier [b], Anthony Maida [a]

[a] *The School of Computing and Informatics, University of Louisiana at Lafayette, Lafayette, LA 70504, USA*
[b] *CERCO UMR 5549, CNRS-Université de Toulouse 3, F-31300, France*

ARTICLE INFO

Article history:
Received 2 November 2017
Received in revised form 6 March 2018
Accepted 25 May 2018
Available online 1 June 2018

Keywords:
Representation learning
Spiking neural networks
Quantization
STDP
Bio-inspired model

ABSTRACT

Although representation learning methods developed within the framework of traditional neural networks are relatively mature, developing a spiking representation model remains a challenging problem. This paper proposes an event-based method to train a feedforward spiking neural network (SNN) layer for extracting visual features. The method introduces a novel spike-timing-dependent plasticity (STDP) learning rule and a threshold adjustment rule both derived from a vector quantization-like objective function subject to a sparsity constraint. The STDP rule is obtained by the gradient of a vector quantization criterion that is converted to spike-based, spatio-temporally local update rules in a spiking network of leaky, integrate-and-fire (LIF) neurons. Independence and sparsity of the model are achieved by the threshold adjustment rule and by a softmax function implementing inhibition in the representation layer consisting of WTA-thresholded spiking neurons. Together, these mechanisms implement a form of spike-based, competitive learning. Two sets of experiments are performed on the MNIST and natural image datasets. The results demonstrate a sparse spiking visual representation model with low reconstruction loss comparable with state-of-the-art visual coding approaches, yet our rule is local in both time and space, thus biologically plausible and hardware friendly.

1. Introduction

Unsupervised learning approaches using neural networks have frequently been used to extract features from visual inputs (Bhand, Mudur, Suresh, Saxe, & Ng, 2011; Lee, Ekanadham, & Ng, 2008). Single layer networks using distributed representations or autoencoder networks (Bengio, Courville, & Vincent, 2013; Coates, Ng, & Lee, 2011) have offered effective representation platforms. However, the robust, high level, and efficient representation that is obtained by networks in the brain is still not fully understood (Frégnac, Fournier, Gérard-Mercier, Monier, Pananceau, Carelli, & Troncoso, 2016; Landi & Freiwald, 2017; Logothetis & Sheinberg, 1996; Quiroga, Reddy, Kreiman, Koch, & Fried, 2005; Riesenhuber & Poggio, 2002; Wandell, 1995; Young & Yamane, 1992). Understanding the brain's functionality in representation learning can be accomplished by studying spike activity (Self et al., 2016) and bio-inspired spiking neural networks (SNNs) (Ghosh-Dastidar & Adeli, 2009; Izhikevich, 2004; Maass, 1997). SNNs provide a biologically plausible architecture, high computational power, and an efficient neural implementation (Maass, 1996, 2015; Neil, Pfeiffer, & Liu, 2016). The main challenge is to develop a spiking representation learning model that encodes input spike trains to uncorrelated, sparse, output spike trains using spatio-temporally local learning rules.

In this study, we seek to develop representation learning in a network of spiking neurons to address this challenge. Our contribution determines novel spatio-temporally local learning rules embedded in a single layer SNN to code independent features of visual stimuli received as spike trains. Synaptic weights in the proposed model are adjusted based on a novel spike-timing-dependent plasticity (STDP) rule which achieves spatio-temporal locality.

Nonlinear Hebbian learning has played a key role in the development of a unified unsupervised learning approach to represent receptive fields (Brito & Gerstner, 2016). Földiák (1990), influenced by Barlow (1989), was one of the early designers of sparse, weakly distributed representations having low redundancy. Földiák's model introduced a set of three learning rules (Hebbian, anti-Hebbian, and homeostatic) to work in concert to achieve these representations. Zylberberg, Murphy, and DeWeese (2011) showed that Földiák's plasticity rules, in a spiking platform, could be derived from the constraints of reconstructive accuracy, sparsity, and decorrelation. Furthermore, the acquired receptive fields of the representation cells in their model (named SAILnet) qualitatively matched those in primate visual cortex. The representation kernels determining the synaptic weight sets have been successfully utilized by our recent study (Tavanaei & Maida, 2017) for a spiking convolutional neural network to extract primary

* Corresponding author.
E-mail addresses: tavanaei@louisiana.edu (A. Tavanaei), timothee.masquelier@cnrs.fr (T. Masquelier), maida@louisiana.edu (A. Maida).

https://doi.org/10.1016/j.neunet.2018.05.018
0893-6080/© 2018 Elsevier Ltd. All rights reserved.

visual features of the MNIST dataset. Additionally, the learning rules only used information which was locally available at the relevant synapse. Although SAILnet utilized spiking neurons in the representation layer and the plasticity rules were spatially local, the learning rules were not temporally local. The SAILnet plasticity rules use spike counts accumulated over the duration of a stimulus presentation interval. Since the SAILnet rules do not use spike times, the question of training the spiking representation network using a spatio-temporally local, spike-based approach like spike-timing-dependent plasticity (STDP) (Markram, Gerstner, & Sjöström, 2012), which needs neural spike times, remains unresolved. Later work, King, Zylberberg, and DeWeese (2013), extends (Zylberberg et al., 2011) to use both excitatory and inhibitory neurons (obeying Dale's law), but the learning rules still use temporal windows of varying duration to estimate spike rates, rather than the timing of spike events. Our work seeks to develop a learning rule which matches this performance but remains local in both time and space.

In another line of research based on cost functions, Bell and Sejnowski (1997) and Olshausen and Field (1996) showed that the constraints of reconstructive fidelity and sparseness, when applied to natural images, could account for many of the qualitative receptive field (RF) properties of primary visual cortex (area 17, V1). These works were agnostic about the possible learning mechanisms used in visual cortex to achieve these representations. Following Olshausen and Field (1996) and Rehn and Sommer (2007) developed the sparse-set coding (SSC) network which minimizes the number of active neurons instead of the average activity measure. Later, Olshausen, Cadieu, and Warland (2009) introduced an L_1-norm minimization criterion embedded in a highly overcomplete neural framework. Although these models offer great insight into what might be computed when receptive fields are acquired, they do not offer insight into details of the learning rules used to achieve these representations.

Early works that proposed a learning mechanism to explain the emergence of orientation selectivity in visual cortex are those of Bienenstock, Cooper, and Munro (1892) and von der Malsburg (1973). A state-of-the-art model is that of Masquelier (2012). This model blends strong biological detail with signal processing analysis and simulation to establish a proof-of-concept demonstration of the original (Hubel & Wiesel, 1962) feedforward model of orientation selectivity. A key feature of that model, relevant to the present paper, is the use of STDP to account for RF acquisition. STDP is the most popular learning rule in SNNs in which the synaptic weights are adapted according to the relative pre- and postsynaptic spike times (Caporale & Dan, 2008; Markram et al., 2012). Different variations of STDP have shown successful visual feature extraction in layer-wise training of SNNs (Kheradpisheh, Ganjtabesh, & Masquelier, 2016; Kheradpisheh, Ganjtabesh, Thorpe, & Masquelier, 2017; Masquelier & Thorpe, 2007; Tavanaei, Masquelier, & Maida, 2016). In a similar vein, Burbank (2015) has also proposed an STDP-based autoencoder. This autoencoder uses a mirrored pair of Hebbian and anti-Hebbian STDP rules. Its goal is to account for the emergence of symmetric, but physically separate, connections for encoding weights (W) and decoding weights (W^T).

Another component playing a key role in representing uncorrelated visual features in a bio-inspired SNN pertains to the inhibition circuits embedded within a layer. For instance, Savin, Joshi, and Triesch (2010) developed an independent component analysis (ICA) computation within an SNN using STDP and synaptic scaling in which independent neural activities in the representation layer were controlled by lateral inhibition. Lateral inhibition established a winner(s)-take-all (WTA) neural circuit to maintain the independence and sparsity of the neural representation layer. More recent work (Diehl & Cook, 2015) has combined a layer of unsupervised STDP with an explicit layer of non-learning inhibitory neurons. The inhibitory neurons impose a WTA discipline. Their representations were tested on the handwritten MNIST dataset and have been shown to be effective for recognition of such digits. The acquired representations tended to resemble MNIST prototypes, although their reconstructive properties were not directly studied. Shrestha, Ahmed, Wang, and Qiu (2017) also studied a spiking network with stochastic neurons that performs MNIST classification and acquires MNIST prototype representations. Their architecture is a 3-layer network where the hidden layer uses a soft WTA to implement inhibition. Since there is no functional need to introduce an explicit inhibitory layer if there is no learning, our work uses a softmax function (Bishop, 1995; Goodfellow, Bengio, & Courville, 2016) to achieve WTA inhibition. In our work, the standard softmax is adapted to a spiking network. Our acquired representations, when trained on the MNIST dataset, acquires representations resembling V1-like receptive fields, in contrast to the MNIST prototypes of the research described above.

Other works related to spike-based clustering and vector quantization are the evolving SNNs (eSNNs and deSNNs) of Kasabov, Dhoble, Nuntalid, and Indiveri (2013), Schliebs and Kasabov (2013), Soltic and Kasabov (2010), Wysoski, Benuskova, and Kasabov (2008) and Wysoski, Benuskova, and Kasabov (2010) which acquire representations via a recruitment learning paradigm (Grossberg, 2012) where neurons are recruited to participate in the representation of the new pattern (based on similarity or dissimilarity to preexisting representations). In the deSNN framework, if a new online pattern is sufficiently similar to an already represented pattern, the representations are merged to form a cluster. This later work uses a number of bio-plausible mechanisms, including spiking neurons, rank-order coding (Thorpe & Gautrais, 1998), a variant of STDP, and dynamic synapses (Maass & Markram, 2002).

The present research proposes event-based, STDP-type rules embedded in a single layer SNN for spatial feature coding. Specifically, this paper proposes a novel STDP-based representation learning method in the spirit of Burbank (2015), Masquelier (2012) and Zylberberg et al. (2011). Its learning rules are local in time and space and implement an approximation to clustering-based, vector quantization (Coates & Ng, 2012) using the SNN while controlling the sparseness and independence of visual codes. Local in time means that the information to modify the synapse is recent, say within at most a couple of membrane time constant of the postsynaptic spike that triggers the STDP. By local in space, we mean that the information used to modify the synaptic weight is, in principle, available at the presynaptic terminal and the postsynaptic cell membrane. Our derivation uses a continuous-time formulation and takes the limit as the length of the stimulus presentation interval tends to one time step. This leads to STDP-type learning rules, although they differ from the classic rules found in Caporale and Dan (2008) and Masquelier (2012). In this sense, the rules and resulting visual coding model are novel. Independence and sparsity are also maintained by an implicit inhibition and a new threshold adjustment rule implementing a WTA circuit.

2. Background

Földiák (1989) developed a feedforward network with anti-Hebbian interconnections for visual feature extraction. The Hebbian rule in his model, shown in Eq. (1), is inspired from Oja's learning rule (Oja, 1982) that extracts the largest principal component from an input sequence,

$$\Delta w_{ji} \propto (y_j x_i - w_{ji} y_j^2) \tag{1}$$

$$y_j = \sum_i x_i w_{ji} \tag{2}$$

where w_{ji} is the weight associated with the synapse connecting input (presynaptic) neuron i and representation (postsynaptic) unit j. x_i and y_j are input and linear output, respectively. Over repeated trials, the term $y_j x_i$ increases the weight when the input and the output are correlated. The second term $(-w_{ji}y_j^2)$ maintains the learning stability (Földiák, 1989). With respect to binary (or spiking) units, a more appropriate assumption was made by Földiák (1990). He modified the previous feedforward network by incorporating non-linear threshold units in the representation layer. The units are binary neurons with a threshold of 0.5 in which $y_j \in \{0, 1\}$ (Note: $y_j^2 = y_j$). Thus, the Hebbian rule in Eq. (1) is simplified to

$$\Delta w_{ji} \propto y_j(x_i - w_{ji}). \tag{3}$$

$$y_j = \begin{cases} 1, & \sum_i x_i w_{ji} > 0.5 \\ 0, & \text{otherwise.} \end{cases} \tag{4}$$

The weight change rules defined in Eqs. (1) and (3) are based on the input and output correlation. Another interpretation for Eq. (3) can be explained in terms of vector quantization (or clustering in a WTA circuit) (Hammer & Villmann, 2002; Schneider, Biehl, & Hammer, 2009) in which the weights connected to each output neuron represent particular clusters (centroids). The weight change is also affected by the output neuron activation, y_j. In this paper, we utilize the vector quantization concept to define an objective function. The objective function can be adapted to develop a spiking visual representation model equipped with a temporally local learning rule while still maintaining sparsity and independence. Our motivation is to use event-based, STDP-type learning rules. This requires the learning to be temporally local, specifically using spike times between pre- and postsynaptic neurons.

3. Spiking visual representation

The proposed model adopts a constrained optimization approach to develop learning rules that are synaptically local. The spiking representation model is a single layer SNN shown in Fig. 1. The representation layer recodes a $p \times p$ image patch ($p \times p$ spike trains) using D spike trains generated by neurons, z_j, in the representation layer.

We derive plasticity rules that operate over a stimulus presentation interval T (non-local in time) and then take the limit as T tends to one local time step to derive event-based rules. In the case of a linear unit, y_j, the objective function to be minimized is shown below. It uses both the vector quantization criterion and a regularizer that prefers small weight values.

$$F(x_i, w_{ji}) = y_j(x_i - w_{ji})^2 + y_j\lambda w_{ji}^2, \quad y_j = \sum_i x_i w_{ji}. \tag{5}$$

The variables $x_i, y_j, w_{ji} \geq 0$ denote: normalized input pixel intensities in the range [0, 1], the linear output activation, and the excitatory synaptic weight, respectively. The first component shows a vector quantization criterion that is scaled by the output neuron's activity, y_j. The y_j scales the weight update rule according to the neuron's response to the input pattern (x_i). The second component (regularizer) is also scaled by the output neuron's activity to control the weight decay criterion (e.g. if $y_j = 0$, w_{ji} does not undergo learning). We assume that the input and output values can be converted to the spike counts over T ms. The hyperparameter $\lambda \geq 0$ controls the model's relative preference for smaller weights. As $\lambda \to 0$, the objective function emphasizes the vector quantization criterion. In contrast, as $\lambda \to \infty$ the vector quantization component is eliminated and the minimum of the objective function is obtained when the w_{ji}'s $\to 0$.

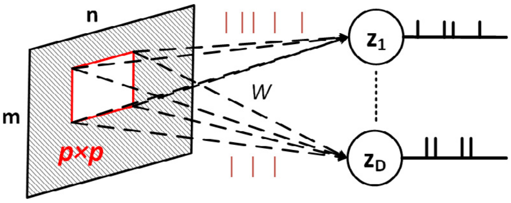


Fig. 1. Spiking representation network. $p \times p$ image patch encoded by D spike trains in the representation layer. W shows the synaptic weight sets corresponding to the D kernels.

In response to a stimulus presentation, a subset of spiking neurons in the representation layer is activated to code the input. To represent the stimuli by uncorrelated codes, the neurons should be activated independently and sparsely. That is, the representation layer demands a WTA neural implementation. This criterion can be achieved by a soft constraint such that

$$g(x) = \sum_j z_j \leq 1 \Rightarrow 1 - \Big(\sum_j z_j\Big) \geq 0 \tag{6}$$

where z_j shows the binary state of unit j after the T ms presentation interval such that $z_j = 1$ if unit j fires at least once. Also, the firing status of a neuron can be controlled by its threshold, θ. Therefore, this constraint can be addressed by a threshold adjustment rule.

The goal is to minimize the objective function (Eq. (5)) while maintaining the constraint (Eq. (6)). This can be achieved by using a Lagrangian function

$$L(x_i, y_j, \mathbf{z}; w_{ji}, \alpha) = \underbrace{y_j(x_i - w_{ji})^2 + y_j\lambda w_{ji}^2}_{\text{Objective Function}} \underbrace{- \alpha(1 - \sum_j z_j)}_{\text{Constraint}} \tag{7}$$

where α is a Lagrange multiplier. Minimizing the first component of Eq. (7) results in a coding module that represents the input by a new feature vector which can cluster the data via the synaptic weights. Minimizing the second component supports the sparsity and independence of the representation to finally (as a special case) end with a winner-take-all network in which exactly one neuron fires upon stimulus presentation. This matter is accomplished by adapting the neuron's threshold, $\theta = -\alpha$. The optimum of the Lagrangian function can be obtained by gradient descent on its derivatives

$$\frac{\partial L}{\partial w_{ji}} = -2y_j(x_i - w_{ji}) + 2y_j\lambda w_{ji} \tag{8}$$

$$\frac{\partial L}{\partial \theta} = -\frac{\partial L}{\partial \alpha} = -\Big(\sum_j z_j - 1\Big). \tag{9}$$

From gradient descent on Eq. (8) (reversing the sign on the derivative), we obtain

$$\Delta w_{ji} \propto y_j(x_i - w_{ji}) - y_j\lambda w_{ji}. \tag{10}$$

However, the information needed in Eq. (10) is not yet temporally local. x_i denotes the rescaled pixel intensity and does not represent the input spike train. To re-encode a pixel intensity, x_i, to a spike train, G_i, we use uniformly distributed spikes (however, each spike train has a different random lag) with rate according to the normalized pixel intensity in the range [0, 1]. The maximum number of spikes (for a completely white pixel) over a $T = 40$ ms interval is 40. Additionally, y_j is a positive value (approximated by spike count) denoting the neuron's activation in response to a stimulus presentation and is not available at synapse, w_{ji}. The value y_j can be reexpressed as H_j representing the output spike train of neuron j.

Spike trains G_i and H_j are formulated by the sum of Dirac functions as shown in Eq. (11). $G_i(t)$ and $H_j(t)$ are either 0 or 1 for a given t.

$$G_i(t)=\sum_{t^{\mathrm{f}}\in S_i^{\mathrm{f}}}\delta(t-t^{\mathrm{f}}),\qquad H_j(t)=\sum_{t^{\mathrm{f}}\in R_j^{\mathrm{f}}}\delta(t-t^{\mathrm{f}}). \tag{11}$$

S_i^{f} and R_j^{f} are the sets of presynaptic and postsynaptic spike times. After coding x_i and y_j by spike trains G_i and H_j, respectively, we propose a local, STDP learning rule following Eq. (10). When x_i and y_j are coded by spike trains over T ms, the synaptic change in continuous time is given by

$$\begin{aligned}\Delta w_{ji} \propto &\left[\int_0^T H_j(t')dt'\right]\left[\frac{1}{K}\int_0^T G_i(t')dt' - w_{ji}\right]\\ &-\lambda w_{ji}\int_0^T H_j(t')dt'.\end{aligned} \tag{12}$$

K is a normalizer denoting the maximum number of presynaptic spikes over the T ms interval. Over a short time period ($t\in[t',t'+\gamma)$, $\gamma<1$ ms, so that $K=1$), the weight adjustment at time t is calculated by

$$\Delta w_{ji}(t) \propto r_j(t)\big(s_i(t)-w_{ji}(t)\big)-\lambda w_{ji}(t)r_j(t). \tag{13}$$

$r_j(t)$ shows the firing status of neuron j at time t ($r_j(t)\in\{0,1\}$). $s_i(t)$ specifies the presence of a presynaptic spike emitted from neuron i at time interval $(t-\epsilon,t]$. In our experiments $\epsilon = 1$ ms. The synaptic weight is changed only when a postsynaptic spike occurs ($r_j(t)=1$). Finally, the learning rule is formulated (upon firing of output neuron j) as follows:

$$\Delta w_{ji}(t) \propto s_i(t)-w_{ji}(t)(1+\lambda) \tag{14}$$

where $w_{ji}\geq 0$. This learning rule is applied to w_{ji} when postsynaptic neuron j fires. The weight change is related to the presynaptic spike times received by the postsynaptic neurons. This scenario resembles spike-timing-dependent plasticity (STDP). In this STDP rule (Eq. (14)), the current synaptic weight affects the magnitude of the weight change. For instance, if $\lambda=0$ and $w_{ji}\in[0,1]$ (it will be proved in Eq. (19)), the smaller weights undergo larger LTP and LTD; and vice versa. It also represents a form of nearest-neighbor spike interaction (Sjöström & Gerstner, 2010).

The second adaptation rule is the threshold learning rule. Eq. (9) is used to implement a learning rule for adjusting the threshold, θ. The threshold learning rule shown in Eq. (15) provides an independent and sparse feature representation. The threshold is the same for all D neurons in the representation layer.

$$\Delta\theta \propto \Big(\sum_{j=1}^{D} z_j\Big)-1. \tag{15}$$

In this section, the theory of the proposed spiking representation learning algorithm was explained. The next section will describe the SNN architecture and the learning rules derived from Eqs. (14) and (15).

4. Network architecture and learning

4.1. Neuron model

The network architecture is shown in Fig. 1 consisting of p^2 and D neurons in the input and representation layers, respectively. Stimuli are converted to spike trains over T ms for both layers. At a given time step, a neuron in the representation layer is allowed to fire only if its firing criterion is met. The firing criterion records the neuron's score in a winners-take-all competition. The WTA score at time step t, given the entire set of incoming weights, W, into the representation layer, is given by

$$WTAscore_j(t;W)=\frac{\exp(\sum_i w_{ji}\zeta_i(t))}{\sum_k \exp(\sum_i w_{ki}\zeta_i(t))} \tag{16}$$

$$\zeta_i(t)=\sum_{t^{\mathrm{f}}} e^{-\frac{(t-t^{\mathrm{f}})}{\tau}} \tag{17}$$

where $\zeta_i(t)$ is the excitatory postsynaptic potential (EPSP) generated by input neuron i and the t^{f}s are the recent spike times of unit i during a small interval $(t-\nu,t]$, where ν is 4 ms. The decay time constant, τ, is set to 0.5 ms.

In our network, the softmax value governs the time at which STDP occurs. If *WTAscore* of a neuron is greater than the adaptive threshold, θ, STDP is triggered and a spike is emitted. The softmax phenomologically implements mutual inhibition among the representation neurons to develop a winners-take-all circuit (Goodfellow et al., 2016; Tavanaei & Maida, 2016) in the representation layer. The neurons in the representation layer are purely excitatory and there is no explicit lateral inhibition between them other than that implicitly implemented by the softmax. When softmax inhibition is imposed within the representation layer, the network implements a form of competitive learning by virtue of STDP being triggered by the firing of postsynaptic neurons. Only neurons that "win the competition" are allowed to learn.

4.2. Learning rules

The synaptic weight change shown in Eq. (14) defines an STDP rule where the current synaptic weight influences the magnitude of the change. STDP events are triggered upon postsynaptic firing yielding two cases corresponding to whether the presynaptic neuron fired within the $(t-\epsilon,t]$ time interval. Eq. (18) shows the final STDP rule derived from Eq. (14). The weights fall in the range [0, 1] and are initialized randomly by sampling from the uniform distribution.

$$\Delta w_{ji}=\begin{cases} a\cdot\big(1-w_{ji}(1+\lambda)\big), & \text{if } s_i=1\\ a\cdot\big(-w_{ji}(1+\lambda)\big), & \text{if } s_i=0\end{cases} \tag{18}$$

a is the learning rate. If $\lambda=0$, the first and second adaptation cases increase and decrease the synaptic weight, respectively (LTP and LTD). If $\lambda\to\infty$, then both cases are negative and decrease the weights down to the minimum value ($w_{ji}=0$). Our experiments study the model's performance using different λ values over a broad range $[0,10^{-4},\ldots,10^4]$. Results are shown in Fig. 2c.

The weight adjustment, at equilibrium, reveals a probabilistic interpretation as follows:

$$E[\Delta w_{ji}]=0 \leftrightarrow \tag{19}$$

$$\begin{aligned}&a\cdot P(s_i=1|r_j=1)(1-w_{ji}(1+\lambda))-\\ &a\cdot P(s_i=0|r_j=1)w_{ji}(1+\lambda)=\\ &a\cdot P(s_i=1|r_j=1)(1-w_{ji}(1+\lambda))-\\ &a\cdot(1-P(s_i=1|r_j=1))w_{ji}(1+\lambda)=0\leftrightarrow\end{aligned}$$

$$(1+\lambda)w_{ji}=P(s_i=1|r_j=1). \tag{20}$$

Therefore, the synaptic weight converges to the $(1+\lambda)$ scaled probability of presynaptic spike occurrence given postsynaptic spike (LTP probability). From Eq. (20), the weights fall in the range $(0,\frac{1}{1+\lambda})$ so that the first case refers to LTP ($\Delta w_{ji}\geq 0$) and the second one refers to LTD ($\Delta w_{ji}\leq 0$), at the equilibrium point.

To show that the STDP rule (Eq. (18)) is consistent with the learning rule in Eq. (10), we rewrite the non-local rule with learning rate, a, as follows:

$$\left(\Delta w_{ji}\right)^{\text{non-local}} = a \cdot y_j\left(x_i - w_{ji} - \lambda w_{ji}\right). \tag{21}$$

As stated earlier, this rule is temporally non-local and shows the weight change over a T ms interval. In contrast, the STDP rule is temporally local, applying the weight change at one time step when the postsynaptic neuron fires. To make Eqs. (21) and (18) (which is derived from Eq. (14)) comparable with each other, we consider a time interval with only one postsynaptic spike where $r_j = 1$. Specifically, we break the T ms interval into subintervals whose boundaries are determined by the event of a postsynaptic spike. It is sufficient to analyze an arbitrary subinterval. Therefore, Eq. (21) at time t simplifies to

$$\left(\Delta w_{ji}\right)^{\text{non-local}} = a\left(x_i - w_{ji} - \lambda w_{ji}\right). \tag{22}$$

Following Eq. (19) for calculating the expected weight change using the proposed STDP rule, where $r_j = 1$, we find that

$$E[\Delta w_{ji}] = a\left(P(s_i = 1) - w_{ji} - \lambda w_{ji}\right) \tag{23}$$

where $P(s_i = 1)$ is the firing probability of presynaptic neuron i. Also, we generated the presynaptic spike trains using the normalized pixel intensities in the range [0, 1] with different random lags. Thus, this probability value is the same as the normalized pixel intensity, x_i, as firing rate. Therefore,

$$E[\Delta w_{ji}] = a\left(x_i - w_{ji} - \lambda w_{ji}\right) = \left(\Delta w_{ji}\right)^{\text{non-local}} \tag{24}$$

which matches the weight change shown in Eq. (22). This shows that the proposed STDP rule is consistent with the non-local rule. Additionally, the STDP weight change is an unbiased estimation for the non-local (non-spike based) learning rule. Over a short time period, the proposed learning rule is also an unbiased estimation for the Hebbian rule of Földiák (1990) (Eq. (3)).

For the threshold adaptation, following Eq. (15), the threshold learning rule can be written as

$$\Delta\theta = b\left(m_z - 1\right) \tag{25}$$

where b is the learning rate. m_z is the number of neurons in the representation layer firing in T ms. This rule adjusts the threshold such that only one neuron fires in response to a stimulus. This criterion provides a framework to extract independent features in a sparse representation. As we used softmax-based neurons in the representation layer, the initial threshold value, θ^{init}, should be in the following range:

$$\frac{1}{D} < \theta^{\text{init}} \ll 0.5 \tag{26}$$

where D is the number of neurons in the representation layer. The upper-bound of 0.5 allows more than one neuron to be active at the initial training steps to capture visual features ($\theta^{\text{init}} \ll 0.5$). On the other hand, the initial threshold should be big enough to stop high synchronization at the beginning ($\theta^{\text{init}} > \frac{1}{D}$). According to the minimum number of neurons we used in the experiments ($D = 8$, $1/D = 0.125$), the initial threshold was set to 0.15.

5. Evaluation metrics

We use the following metrics to judge the quality of the representation acquired in Fig. 1.

5.1. Reconstructed image

We use reconstructed image to qualitatively assess the extent that the representation layer captures the information contained in the image patches. The representation filter set, $W = \{w_1, w_2, \ldots, w_D\}$, is a $p^2 \times D$ weight matrix coding an image patch (p^2 input spike trains) to a vector of D postsynaptic spike trains. To reconstruct the image patch from the coded spike trains, the reconstruction filter set, $W^{\text{rec}} \equiv W^T$, is used to build p^2 spike trains. For this purpose, neurons in the input layer receive spike trains from the neurons in the representation layer via the transposed synaptic weight matrix (like an autoencoder).

5.2. Reconstruction loss

To report the reconstruction loss, we use the correlation measure (Pearson correlation) and the root mean square (RMS) error between the normalized original, $\mathbf{y}_m$, and reconstructed, $\hat{\mathbf{y}}_m$, patches as shown in Eqs. (27) and (28), respectively. A patch stands for p^2 spike rates, $\mathbf{y}$.

$$Corr\ Recon\ Loss = \frac{1}{M}\sum_{m=1}^{M} 1 - Cor(\mathbf{y}_m, \hat{\mathbf{y}}_m) \tag{27}$$

$$RMS = \frac{1}{M}\sum_{m=1}^{M}\sqrt{\frac{1}{p^2}\sum_{i=1}^{p^2}(y_{i,m} - \hat{y}_{i,m})^2}. \tag{28}$$

M is the number of patches extracted from the image.

5.3. Sparsity

To calculate the sparsity, we use average activity and breadth tuning measures. The average activity specifies the density of spikes released from neurons in the representation layer over T time steps given in Eq. (29).

$$Sparsity = \frac{1}{D \cdot T}\sum_{j}\sum_{t} r_j(t). \tag{29}$$

The breadth tuning measure introduced by Rolls and Tovee (1995) specifies the density of neural layer activity (Eq. (30)) calculated by the ratio of mean, μ, and standard deviation, σ, of spike frequencies in the representation layer upon presenting a stimulus. The breadth tuning measures the neural selectivity such that the sparse code distribution concentrates near zero with a heavy tail (Foldiak & Endres, 2008). For a neural layer where most of the neurons fire, the activity distribution is more uniformly spread and *Breadth Tuning* is greater than 0.5. In contrast, in a sparse code where most of the neurons do not fire, the distribution is peaked at zero and *Breadth Tuning* is less than 0.5.

$$Breadth\ Tuning = \frac{1}{C^2 + 1}, \quad C = \frac{\sigma}{\mu}. \tag{30}$$

6. Experiments and results

We ran two experiments using the MNIST (LeCun, Cortes, & Burges, 0000) and the natural image (Olshausen & Field, 1996) datasets to evaluate the proposed local representation learning rules embedded in the single-layer SNN. For both datasets, the intensities of the gray-scale images were normalized to fall in the range of [0, 1] yielding possible spike rates to generate uniformly distributed spike trains for the input layer over $T = 40$ ms. The learning rates for STDP learning, a, (Eq. (18)) and for threshold adjustment, b, (Eq. (25)) were set to 0.0005 and 0.0001, respectively.[1]

We ran a number of experiments with different learning rates and found that changing a and b in the range [0.00005, 0.001] did not change the model's performance significantly. Additionally, as the threshold adjustment rule is not modulated by the current threshold value, we chose a smaller learning rate (0.0001) for it to avoid possible threshold instability.

6.1. Experiment 1: MNIST dataset

Experiments were run using 5×5 patches sampled from 28×28 MNIST digits. We used a random subset of the MNIST digits divided into 15,000 training and 1000 testing images for learning and evaluating the model. The SNN architecture consists of 25 (5×5 image patch) neurons in the input layer and $D = \{2^i,\ i = 3 \ldots 7\}$ neurons in the representation layer. These variations of the network architecture (different D values) determine under-complete to over-complete representations. Trained filters, after 1 through 15,000 iterations, for the network with 32 neurons in the representation layer are shown in Fig. 2a. After 1000 training iterations, the kernels start becoming selective to specific visual patterns (orientations). The filters shown in this image tend to be orientation selective and extract different visual features.

Fig. 2c shows the RMS reconstruction loss and other statistical characteristics (max, min, mean) of the trained weights versus the log regularizer hyperparameter ($\log_{10}\lambda$). For $\lambda \le 0.1$, the RMS loss values reach a near optimal uniformly low plateau.[2] For this reason, λ was set to zero for further experiments. Additionally, Fig. 2c shows that the maximum and minimum synaptic weights after training are $1/(1+\lambda)$ and 0, respectively, as predicted by Eq. (20).

The three performance measures from Section 5 were used to assess the model. These were the reconstructed images, the reconstruction loss, and the sparsity. The reconstructed images of randomly selected digits 0 through 9, acquired by the SNN with $D = 32$ neurons in the representation layer, are shown in Fig. 2b. The reconstructed maps show high quality images comparable with the original images. The reconstruction loss measures for the SNNs with $D = 8$ through 128 filters appear in Fig. 3a and b. The SNNs with $D = 16$ and 32 show the lowest reconstruction loss after training. The sparsity measures reported by the average sparsity and the breadth tuning are shown in Fig. 3c and d. The sparsity measures also show better performance for the networks with $D = \{16, 32, 64\}$ filters. The average sparsity value of 0.09 shows that only 9% of the neurons were active on each trial. The breadth tuning value of 0.23 indicates the sparse stimulus representation. Fig. 3e and f depict the summary of the model's performance after training for $D = \{8, \ldots, 128\}$ and $D = 32$ kernels, respectively.

6.2. Experiment 2: Natural images

This experiment evaluates representations acquired from 16×16 natural image patches (Olshausen & Field, 1996). Fig. 4a shows the trained representation filters for the SNNs with 8, 16, 32, 64, and 128 neurons in the representation layer. For instance, where $D = 32$, except for the filters marked with dotted circles, the other filters have low correlation with each other. For visual assessments, Fig. 4b shows four natural images and their reconstructed maps. Performance of the proposed model in terms of the reconstruction loss and sparsity measures on natural images is shown in Fig. 4c. The lowest reconstruction loss belongs to the networks with $\{16, 32, 64\}$ neurons in the representation layer. The small number of neurons ($D = 8$) is not able to capture the visual codes. On the other hand, using too many neurons ($D > 128$) increases reconstruction loss because a number of neurons cannot be involved in the learning process due to the WTA constraint.

6.3. Comparisons

The proposed spiking representation learning method shows better performance than the traditional K-means clustering (Bishop, 2006) and the restricted Boltzmann machine (RBM) (Hinton, 2010; Le Roux & Bengio, 2008) while introducing local learning in time and space. We implemented these two methods, as traditional quantization-like representation learning examples, using the same training/testing images. The K-means and RBM approaches were applied to the normalized pixel intensities of image patches (not spike trains). Thus, these methods are not temporally local. Table 1 shows this comparison in terms of reconstruction loss (correlation and RMS). Our model outperforms the RBM and K-means methods except for the two cases (natural images) in which the RBM shows slightly better performance. Fig. 5 shows the trained filters obtained by K-means, RBMs, and our model based on the MNIST and natural image patches. K-means, similar to our model, detects different visual orientations for the MNIST and natural image patches, but the filters are highly correlated. The RBM did not perform well for the MNIST dataset but it successfully learned representative visual filters for the natural image patches where $D = 64$. These trained filters (Fig. 5) confirm the reconstruction loss variations reported in Table 1.

Table 2 compares our results with the only (to the best of our knowledge) spike-based representation learning models. The correlation-based reconstruction loss on MNIST and natural images (0.2 and 0.4) shows improvement over the existing spiking autoencoder using mirrored STDP (0.2 and 0.65) proposed by Burbank (2015). The sparse representation introduced by King et al. (2013), which is a modified version of the SAILnet algorithm (Zylberberg et al., 2011), reported an RMS reconstruction loss around 0.74 that is calculated based on the spike rates normalized to unit standard deviation (let us say zRMS). Our model compared favorably with their model with zRMS = 0.67. However, our model did not scale well to a larger number of neurons when $D \geq 128$ in the representation layer. The problem appears to stem from the threshold adjustment rule (Eq. (25)). If we change the rule to $\Delta\theta = b\big(m_z - q\big)$, where q is a proportion of D, the representation layer would be more active and a large number of filters can be trained to reduce the reconstruction loss.

7. Discussion and conclusion

This paper derived a novel STDP-based representation learning method to be embedded in an SNN and evaluated the acquired representations in two experiments to establish the method's initial viability. The derived rules were extremely simple, yet the evaluated reconstruction loss was extremely low. The simplicity of the rules (resulting from the constraint of temporal locality) makes them attractive for hardware implementation.

The learning rules were derived by constrained optimization incorporating a vector quantization-like objective function with regularization and a sparsity constraint. The learning rules included spatio-temporally local STDP-type weight adaptation and a threshold adjustment rule. The STDP rule at equilibrium showed a probabilistic interpretation of the synaptic weights scaled by the regularizer hyperparameter. In addition to the threshold adaptation rule, the WTA-thresholded neurons in the representation

[1] The maximum number of postsynaptic spikes is 40 and the maximum number of patches sampled from an MNIST digit is 25. Our simple strategy for setting the learning rates is $a, b < \frac{1}{25\times 40} = 0.001$.

[2] The average RMS reconstruction loss values for the SNNs with $\lambda \le 0.1$ was reported 0.167 ± 0.001 (Ninety-five percent confidence intervals of the RMS loss values (standard error of the mean; $n = 5$) were calculated).

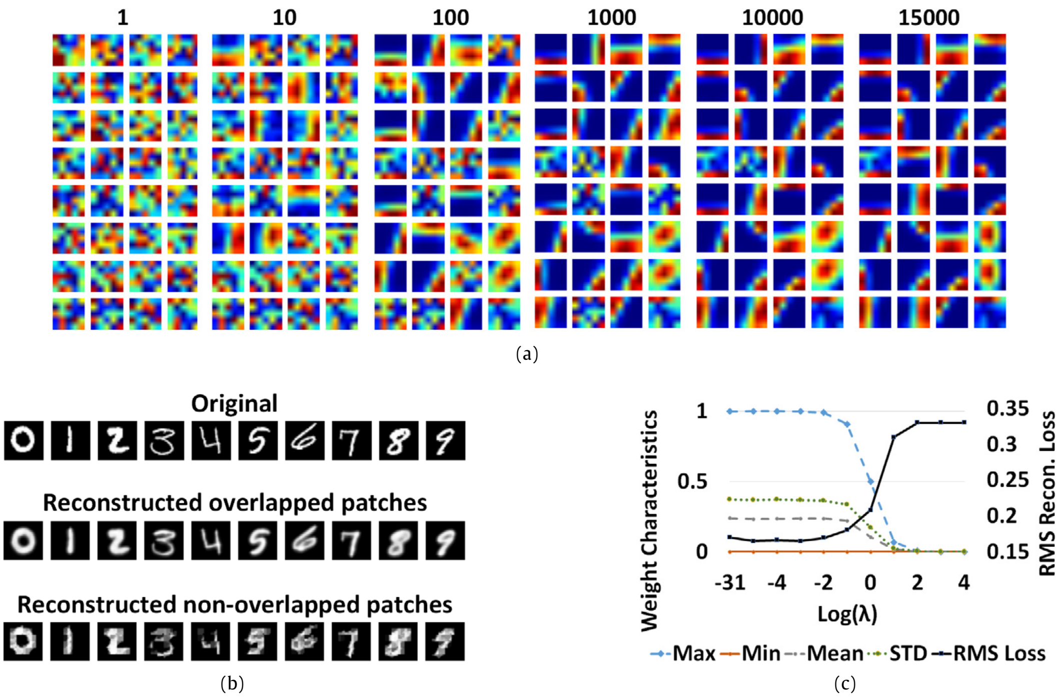


Fig. 2. (a) $D = 32$ trained filters after 1, 10, . . ., 15 000 iterations. The red–blue spectrum denotes the maximum–minimum synaptic weights. (b) Reconstructed images based on overlapped and non-overlapped 5×5 patches. The overlapped patches are selected by 5×5 windows sliding over the image with a stride of 1. The non-overlapped patches slide over the image with a stride of 5. (c) RMS reconstruction loss and synaptic weight ranges for the SNN with $D = 32$ filters versus $\log_{10}$ regularizer hyperparameter, λ. $\lambda = 0$ is approximated by 10^{-31}. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

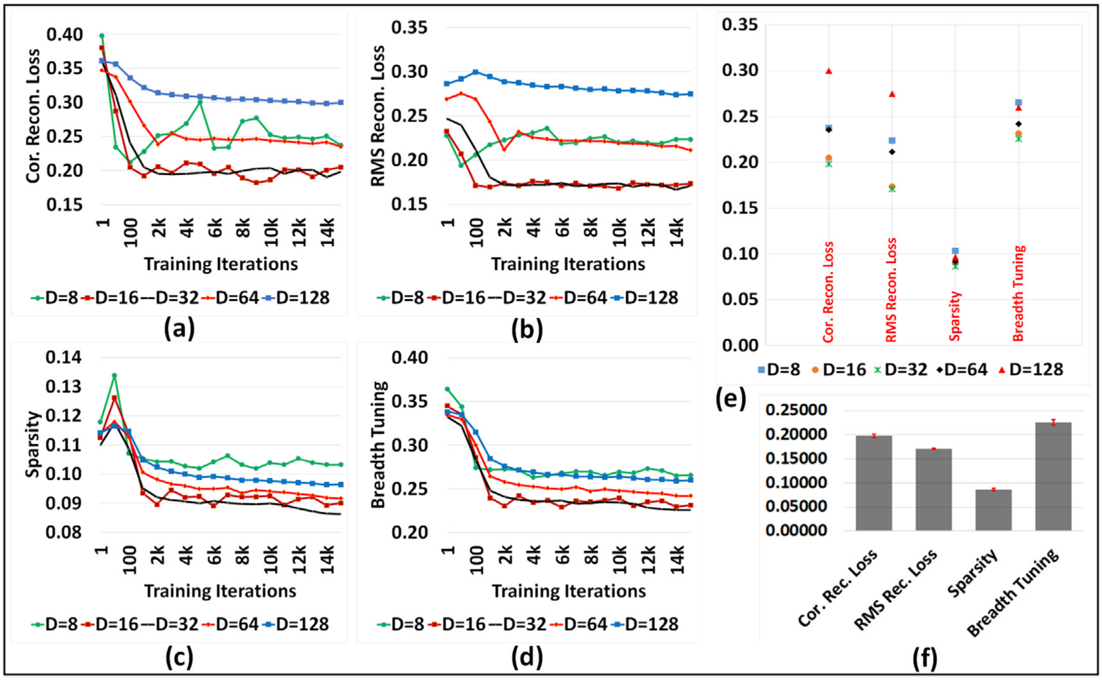
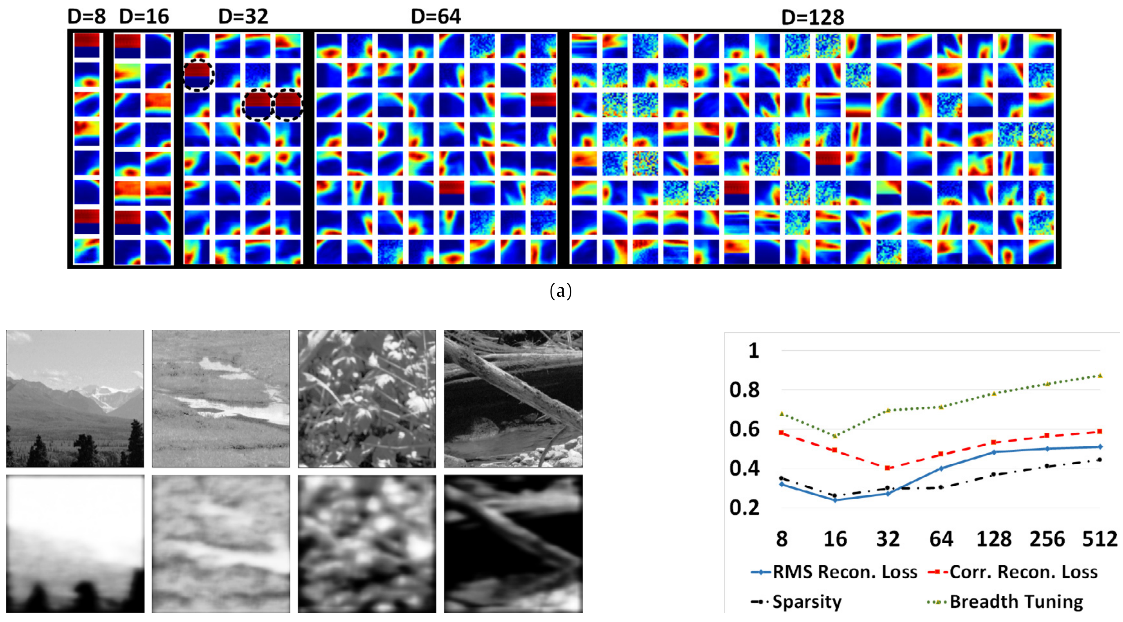


Fig. 3. (a)–(d) Model performance trends on MNIST after 1 through 15 000 training iterations in terms of (a) correlation-based reconstruction loss, (b) RMS reconstruction loss, (c) average sparsity, and (d) breadth tuning. (e) The model's performance after training. (f) The evaluation measures for the trained visual representation model with $D = 32$ kernels. Error bars show standard error of the mean.

layer implemented inhibition (by a novel temporal, spiking softmax function) to represent sparse and independent visual features. Softmax is a standard way to implement a winners-take-all (WTA) circuit and to implement mutual inhibition without using explicit inhibitory neurons in the representation layer Bishop (1995, p. 238), (Goodfellow et al., 2016, p. 181).

The experimental results showed high performance of the proposed model in comparison with spiking and non-spiking approaches. Our model almost outperformed the traditional K-Means and RBM models in representation learning and training of the orientation selective kernels. Also, our method showed better performance (in terms of reconstruction loss) than the



(a)

(b)

(c)

Fig. 4. Model's performance on the natural image patches. (a) Representation filters after training the SNNs with $D = \{8, 16, 32, 64, 128\}$ spiking neurons in the representation layer. (b) Original (first row) and reconstructed (second row) image sections ($D = 32$). (c) Reconstruction loss and sparsity measures of the models with 8 through 512 filters.

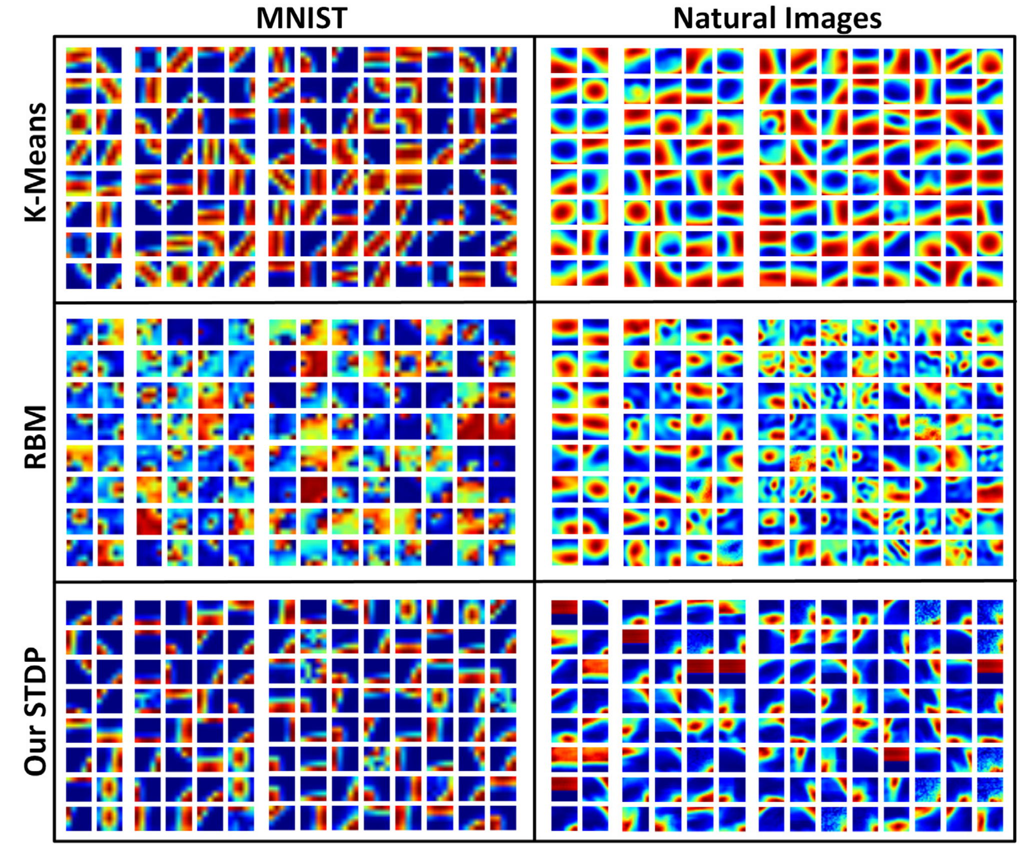


Fig. 5. $D = 16$, 32, and 64 representation filters trained on the MNIST and natural images datasets using K-means, RBM, and our STDP.

state-of-the-art spiking representation learning approaches used by Burbank (2015) (spiking autoencoder) and King et al. (2013) and Zylberberg et al. (2011), (sparse representation).

To obtain the spatio-temporally local learning rules embedded in the SNN, we started from a non-spiking quantization criterion inspired from Földiák (1990). Then, we developed novel rules to

Table 1
Reconstruction loss (correlation and RMS) obtained by K-means and RBM in comparison with our method.

Rec. Loss	MNIST Corr.			MNIST RMS			Natural Corr.			Natural RMS		
D	16	32	64	16	32	64	16	32	64	16	32	64
K-means	0.22	0.23	0.26	0.18	0.21	0.26	**0.45**	0.52	0.57	0.31	0.36	0.40
RBM	0.49	0.49	0.40	0.27	0.27	0.26	0.92	0.41	**0.44**	0.47	**0.27**	**0.26**
Our STDP	**0.20**	**0.20**	**0.24**	**0.17**	**0.17**	**0.21**	0.49	**0.40**	0.47	**0.24**	**0.27**	0.40

Table 2
The reconstruction loss values reported by Burbank (2015) in terms of correlation loss and King et al. (2013) in terms of zRMS in comparison with our results.

Dataset	Burbank (2015)	King et al. (2013)	Our model
Natural images	Corr: 0.65	zRMS: 0.74	Corr: **0.4**, zRMS: **0.67**
MNIST	Corr: **0.2**	–	Corr: **0.2**

implement an STDP based representation learning and a threshold adjustment rule for spiking platforms. The spike-based platform and spatio-temporally local learning rules lead the main difference between our study and well-known, traditional representation learning methods introduced in the literature. Existing spiking representation learning methods in the literature suffer from limitations such as violating Dale's law (Zylberberg et al., 2011), synapses that can change sign (King et al., 2013; Zylberberg et al., 2011), low performance in terms of reconstruction loss (Burbank, 2015), and non-spiking input signals (King et al., 2013; Zylberberg et al., 2011). In this study we proposed an STDP learning rule which updates the synaptic weights falling within the range [0, 1]. The SNN architecture consists of excitatory neurons and an implicit inhibition occurring in the representation layer. The implicit inhibition is analogous to a separate inhibitory layer balancing neural activities in the representation neural layer where Dale's law is maintained. Furthermore, the proposed SNN implements spiking neurons in both the input and representation layers and the neurons only communicate through temporal spike trains.

To the best of our knowledge, our approach is the only high performance (in terms of reconstruction loss) representation learning model implemented on SNNs. There are several studies in the literature developing SNNs equipped with bio-inspired STDP for unsupervised feature extraction through single or multi-layer spike-based architectures. The most recent works of Diehl and Cook (2015), Kheradpisheh et al. (2017), Shrestha et al. (2017) and Tavanaei and Maida (2017) have utilized these features to classify MNIST digits. Although these networks introduce novel spiking network architectures for feature representation, they do not offer a pure representation learning approach with low reconstruction loss.

Although our proposed spiking representation learning was successful for reconstruction, there is a limitation that the spike rate of the presynaptic neurons is higher than biological spiking neurons. Our future work seeks to reduce this spike rate to be more biologically plausible. Using more presynaptic neurons presenting mutual exclusive intensity bands would be a starting point. Additionally, it is a matter of future work to determine how well the acquired representations from our STDP algorithm perform in a pattern recognition context. It can also be tested in future work whether our acquired representations are stackable to afford the ability for multi-layer, STDP-based learning.

References

Barlow, H. B. (1989). Unsupervised learning. *Neural Computation*, *1*(3), 295–311.

Bell, A. J., & Sejnowski, T. J. (1997). The "independent components" of natural scenes are edge filters. *Vision Research*, *37*(23), 3327–3338.

Bengio, Y., Courville, A., & Vincent, P. (2013). Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *35*(8), 1798–1828.

Bhand, M., Mudur, R., Suresh, B., Saxe, A., & Ng, A. Y. (2011). Unsupervised learning models of primary cortical receptive fields and receptive field plasticity. In *Advances in neural information processing systems* (pp. 1971–1979).

Bienenstock, E. L., Cooper, L. N., & Munro, P. W. (1892). Theory for the development of neuron selectivity: Orientation specifity and binocular interaction in visual cortex. *Journal of Neuroscience*, *2*(1), 32–48.

Bishop, C. M. (1995). *Neural networks for pattern recognition*. Oxford University Press.

Bishop, C. M. (2006). *Pattern recognition and machine learning*. Springer.

Brito, C. S., & Gerstner, W. (2016). Nonlinear Hebbian learning as a unifying principle in receptive field formation. *PLoS Computational Biology*, *12*(9), e1005070.

Burbank, K. S. (2015). Mirrored STDP implements autoencoder learning in a network of spiking neurons. *PLoS Computational Biology*, *11*(12), e1004566.

Caporale, N., & Dan, Y. (2008). Spike timing-dependent plasticity: a Hebbian learning rule. *Annual Review of Neuroscience*, *31*, 25–46.

Coates, A., & Ng, A. Y. (2012). Learning feature representations with k-means. In *Neural networks: Tricks of the trade* (pp. 561–580). Springer.

Coates, A., Ng, A., & Lee, H. (2011). An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics* (pp. 215–223).

Diehl, P. U., & Cook, M. (2015). Unsupervised learning of digit recognition using spike-timing-dependent plasticity. *Frontiers in Computational Neuroscience*, *9*, 99.

Földiák, P. (1989). Adaptive network for optimal linear feature extraction. In *1989 international joint conference on neural networks, Vol. 1* (pp. 401–405). IEEE.

Földiák, P. (1990). Forming sparse representations by local anti-Hebbian learning. *Biological Cybernetics*, *64*(2), 165–170.

Foldiak, P., & Endres, D. (2008). Sparse coding. *Scholarpedia*, *3*(1), 2984.

Frégnac, Y., Fournier, J., Gérard-Mercier, F., Monier, C., Pananceau, M., Carelli, P., & Troncoso, X. (2016). The visual brain: Computing through multiscale complexity. In *Micro-, meso-and macro-dynamics of the brain* (pp. 43–57). Springer.

Ghosh-Dastidar, S., & Adeli, H. (2009). Spiking neural networks. *International Journal of Neural Systems*, *19*(04), 295–308.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT press.

Grossberg, S. (2012). Adaptive resonance theory: How a brain learns to consciously attend, learn, and recognize a changing world. *Neural Networks*, *37*, 1–47.

Hammer, B., & Villmann, T. (2002). Generalized relevance learning vector quantization. *Neural Networks*, *15*(8), 1059–1068.

Hinton, G. (2010). A practical guide to training restricted Boltzmann machines. *Momentum*, *9*(1), 926.

Hubel, D. H., & Wiesel, T. N. (1962). Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of Physiology*, *160*(1), 106–154.

Izhikevich, E. M. (2004). Which model to use for cortical spiking neurons? *IEEE Transactions on Neural Networks*, *15*(5), 1063–1070.

Kasabov, N., Dhoble, K., Nuntalid, N., & Indiveri, G. (2013). Dynamic evolving spiking neural networks for on-line spatio-and spectro-temporal pattern recognition. *Neural Networks*, *41*, 188–201.

Kheradpisheh, S. R., Ganjtabesh, M., & Masquelier, T. (2016). Bio-inspired unsupervised learning of visual features leads to robust invariant object recognition. *Neurocomputing*, *205*, 382–392.

Kheradpisheh, S. R., Ganjtabesh, M., Thorpe, S. J., & Masquelier, T. (2017). STDP-based spiking deep convolutional neural networks for object recognition. *Neural Networks*, *99*, 56–67.

King, P. D., Zylberberg, J., & DeWeese, M. R. (2013). Inhibitory interneurons decorrelate excitatory cells to drive sparse code formation in a spiking model of V1. *The Journal of Neuroscience*, *33*(13), 5475–5485.

Landi, S. M., & Freiwald, W. A. (2017). Two areas for familiar face recognition in the primate brain. *Science*, *357*(6351), 591–595.

Le Roux, N., & Bengio, Y. (2008). Representational power of restricted Boltzmann machines and deep belief networks. *Neural Computation*, *20*(6), 1631–1649.

LeCun, Y., Cortes, C., & Burges, C. J. The MNIST database, URL http://yann.lecun.com/exdb/mnist.

Lee, H., Ekanadham, C., & Ng, A. Y. (2008). Sparse deep belief net model for visual area V2. In *Advances in neural information processing systems* (pp. 873–880).

Logothetis, N. K., & Sheinberg, D. L. (1996). Visual object recognition. *Annual Review of Neuroscience*, *19*(1), 577–621.

Maass, W. (1996). On the computational power of noisy spiking neurons. In *Advances in neural information processing systems* (pp. 211–217).

Maass, W. (1997). Networks of spiking neurons: the third generation of neural network models. *Neural Networks*, *10*(9), 1659–1671.

Maass, W. (2015). To spike or not to spike: that is the question. *Proceedings of the IEEE*, *103*(12), 2219–2224.

Maass, W., & Markram, H. (2002). Synapses as dynamic memory buffers. *Neural Networks*, *15*(2), 155–161.

Markram, H., Gerstner, W., & Sjöström, P. J. (2012). Spike-timing-dependent plasticity: a comprehensive overview. *Frontiers in Synaptic Neuroscience*, *4*, 2.

Masquelier, T. (2012). Relative spike time coding and STDP-based orientation selectivity in the early visual system in natural continuous and saccadic vision: a computational model. *Journal of Computational Neuroscience*, *32*(3), 425–441.

Masquelier, T., & Thorpe, S. J. (2007). Unsupervised learning of visual features through spike timing dependent plasticity. *PLoS Computational Biology*, *3*(2), e31.

Neil, D., Pfeiffer, M., & Liu, S.-C. (2016). Learning to be efficient: Algorithms for training low-latency, low-compute deep spiking neural networks. In *Proceedings of the 31st annual ACM symposium on applied computing* (pp. 293–298). ACM.

Oja, E. (1982). Simplified neuron model as a principal component analyzer. *Journal of Mathematical Biology*, *15*(3), 267–273.

Olshausen, B. A., Cadieu, C. F., & Warland, D. K. (2009). Learning real and complex overcomplete representations from the statistics of natural images, in: Proc SPIE, Vol. 7446, pp. 74460S–1.

Olshausen, B. A., & Field, D. J. (1996). Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, *381*(6583), 607–609.

Quiroga, R. Q., Reddy, L., Kreiman, G., Koch, C., & Fried, I. (2005). Invariant visual representation by single neurons in the human brain. *Nature*, *435*(7045), 1102–1107.

Rehn, M., & Sommer, F. T. (2007). A network that uses few active neurones to code visual input predicts the diverse shapes of cortical receptive fields. *Journal of Computational Neuroscience*, *22*(2), 135–146.

Riesenhuber, M., & Poggio, T. (2002). Neural mechanisms of object recognition. *Current Opinion in Neurobiology*, *12*(2), 162–168.

Rolls, E. T., & Tovee, M. J. (1995). Sparseness of the neuronal representation of stimuli in the primate temporal visual cortex. *Journal of Neurophysiology*, *73*(2), 713–726.

Savin, C., Joshi, P., & Triesch, J. (2010). Independent component analysis in spiking neurons. *PLoS Computational Biology*, *6*(4), e1000757.

Schliebs, S., & Kasabov, N. (2013). Evolving spiking neural network a survey. *Evolving Systems*, *4*(2), 87–98.

Schneider, P., Biehl, M., & Hammer, B. (2009). Distance learning in discriminative vector quantization. *Neural Computation*, *21*(10), 2942–2969.

Self, M. W., Peters, J. C., Possel, J. K., Reithler, J., Goebel, R., Ris, P., et al. (2016). The effects of context and attention on spiking activity in human early visual cortex. *PLoS Biology*, *14*(3), e1002420.

Shrestha, A., Ahmed, K., Wang, Y., & Qiu, Q. (2017). Stable spike-timing dependent plasticity rule for multilayer unsupervised and supervised learning. In *2017 international joint conference on neural networks* (pp. 1999–2006). IEEE.

Sjöström, J., & Gerstner, W. (2010). Spike-timing dependent plasticity. *Spike-Timing Dependent Plasticity*, *35*, 35–44.

Soltic, S., & Kasabov, N. (2010). Knowledge extraction from evolving spiking neural networks with rank order population coding. *International Journal of Neural Systems*, *20*(06), 437–445.

Tavanaei, A., & Maida, A. S. (2016). Bio-inspired spiking convolutional neural network using layer-wise sparse coding and STDP learning, arXiv preprint arXiv:1611,03000, pp. 1–16.

Tavanaei, A., & Maida, A. S. (2017). Multi-layer unsupervised learning in a spiking convolutional neural network. In *2017 international joint conference on neural networks* (pp. 2023–2030). IEEE.

Tavanaei, A., Masquelier, T., & Maida, A. S. (2016). Acquisition of visual features through probabilistic spike-timing-dependent plasticity. In *2016 international joint conference on neural networks* (pp. 307–314). IEEE.

Thorpe, S., & Gautrais, J. (1998). Rank order coding. *Computational Neuroscience*, *113*, 113–119.

von der Malsburg, C. (1973). Self-organization of orientation sensitive cells in the striate cortex. *Kybernetik*, *14*, 85–100.

Wandell, B. A. (1995). *Foundations of vision*. Sinauer Associates.

Wysoski, S. G., Benuskova, L., & Kasabov, N. (2008). Fast and adaptive network of spiking neurons for multi-view visual pattern recognition. *Neurocomputing*, *71*(13–15), 2563–2575.

Wysoski, S. G., Benuskova, L., & Kasabov, N. (2010). Evolving spiking neural networks for audiovisual information processing. *Neural Networks*, *23*(7), 819–835.

Young, M. P., & Yamane, S. (1992). Sparse population coding of faces in the inferotemporal cortex. *Science*, *256*(5061), 1327–1331.

Zylberberg, J., Murphy, J. T., & DeWeese, M. R. (2011). A sparse coding model with synaptically local plasticity and spiking neurons can account for the diverse shapes of V1 simple cell receptive fields. *PLoS Computational Biology*, *7*(10), e1002250.