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) for extracting visual features. The method introduces a novel spike-timing-dependent plasticity (STDP) rule derived from a vector quantization-like objective function subject to a sparsity constraint. Independence and sparsity of the model are achieved by a threshold adjustment rule and 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 been frequently employed to extract weakly correlated visual features. Single layer networks such as distributed representations and autoencoder networks have offered promising and efficient representation platforms. Földiák [3], influenced by Barlow [4], was one of the early designers of sparse, weakly distributed representations with low redundancy. The Földiák model introduced a set of three learning rules (Hebbian, anti-Hebbian, and homeostatic) to work in concert to achieve these representations. Zylberberg et al. [5] showed that Földiák’s plasticity rules 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 SAIL.net) qualitatively matched those in primate visual cortex and the learning rules only used information which was locally available at the relevant synapse. Although they utilized spiking neurons in the representation layer and their plasticity rules were spatially local, the learning rules were not temporally local. The SAIL.net plasticity rules use spike counts accumulated over the duration of a stimulus presentation interval. Since the SAIL.net rules do not use spike times, the question of training the spiking representation network using an STDP-based approach, which needs neural spike times, remains unresolved. Later work, [6], extends their model to use both...
excitatory and inhibitory neurons (obeying the constraints of 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. In another line of research based on cost functions, Olshausen and Field [7] and Bell and Sejnowski [8] established 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.

Early works that proposed a learning mechanism to explain the emergence of orientation selectivity in visual cortex are those of von der Malsburg [9] and Bienenstock et al. [10]. A state-of-the-art model is that of Masquelier [11]. This model blends strong biological detail with signal processing analysis and simulation to establish a proof-of-concept demonstration of the original Hubel and Wiesel [12] feedforward model of orientation selectivity. A key feature of that model, relevant to the present paper, is the use of spike-timing-dependent plasticity (STDP) [13] to account for RF acquisition. In a similar vein, Burbank [14] 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 et al. [15] 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.

The present research seeks to use event-based, STDP-type rules in the spirit of [11][13][14]. Specifically, this paper proposes a novel STDP-based representation learning method. Its learning rules are local in time and space to implement an approximation to clustering-based, vector quantization [16] using the SNN while controlling the sparseness and independence of visual codes. 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 [11] and [13]. In this sense, the rules and resulting visual coding model are novel.

2 Background

Földiák [17] 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 [18] extracting the largest principal component from an input sequence,

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

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 $x_i y_j$ increases the weight when the input and the output are correlated. The second term ($-w_{ji} y_j^2$) maintains the learning stability [17]. A more consistent assumption for binary (or spiking) units has been made by Földiák [3]. He modified the previous feedforward network by incorporating non-linear 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

$$\Delta w_{ji} \propto y_j (x_i - w_{ji}) \quad (2)$$

The weight change rules defined in Eqs. 1 and 2 are based on the input and output correlation. Another interpretation for Eq. 2 can be explained in terms of vector quantization (or clustering) [19][20] in which the weights connected to each output neuron represent particular clusters (centroids). The weight change also is 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) and then take the limit as $T$ tends to one local time step to derive event-based rules. The objective function using both the vector quantization criterion and a regularizer that prefers small weight values is shown below.

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

(3)

The parameters $x_i, y_j, w_{ji} \geq 0$ are normalized input pixel intensities in the range [0, 1], the linear output activation, and the synaptic weight respectively. 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 $\to 0$.

Since each stimulus must generate some kind of representation in the representation layer, one or more neurons must become activated. A function capturing this constraint can be written as

$$g(x) = \sum_j z_j \geq 1 \Rightarrow (\sum_j z_j) - 1 \geq 0.$$  

(4)

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. The firing status can be controlled by a threshold parameter.

The goal is to minimize the objective function (Eq. 3) while maintaining the constraint (Eq. 4). This can be achieved by using a Lagrangian function

$$L(x_i, y_j, z; w_{ji}, \theta) = y_j(x_i - w_{ji})^2 + y_j \lambda w_{ji}^2 - \theta \left( \sum_j z_j - 1 \right)$$

(5)

where, $\theta$ is a Lagrange multiplier. Minimizing the first component of Eq. 5 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 one and only one neuron fires upon stimulus presentation. The optimum of the Lagrangian function can be obtained by gradient descent on its derivatives

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

(6)

$$\frac{\partial L}{\partial \theta} = -\left( \sum_j z_j - 1 \right)$$

(7)
From gradient descent on Eq. 5 (reversing the sign on the derivative), we obtain
\[
\Delta w_{ji} \propto y_j (x_i - w_{ji}) - y_j \lambda w_{ji}.
\] (8)

However, the information needed in Eq. 8 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 of normalized pixel intensity in the range \([0, 1]\). The maximum number of spikes (for a completely white pixel) for a \( T = 40 \) ms interval is 40. Additionally, \( y_j \) is a positive value (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. 9.
\[
G_i(t) = \sum_{t' \in S^t_i} \delta(t - t'), \quad H_j(t) = \sum_{t' \in R^t_j} \delta(t - t')
\] (9)

\( S^t_i \) and \( R^t_j \) 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 seek to propose a local, STDP learning rule following Eq. 8. When, \( x_i \) and \( y_j \) are coded by spike trains over \( T \) ms, the synaptic change in continuous time is given by
\[
\Delta w_{ji} \propto \left[ \int_0^T H_j(t') dt' \right] \left[ \frac{1}{K} \int_0^T G_i(t') dt' - w_{ji} \right] - \lambda w_{ji} \int_0^T H_j(t') dt'
\] (10)

\( K \) is a normalizer denoting the maximum number of presynaptic spikes in \( T \) ms interval. Over a short time period \((t \in [t', t' + \gamma], \gamma < 1 \text{ ms}, \text{ so that } K = 1)\), the weight adjustment at time \( t \) is calculated by
\[
\Delta w_{ji}(t) \propto r_j(t)(s_i(t) - w_{ji}(t)) - \lambda w_{ji}(t)r_j(t)
\] (11)

\( r_j(t) \) shows the firing status of neuron \( j \) at time \( t \) \((r_j(t) \in \{0, 1\})\). \( s_i(t) \) specifies the presynaptic spike emitted from neuron \( i \) at time interval \((t - \epsilon, t]\). In our experiments \( \epsilon = 1 \text{ ms} \). 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
\[
\Delta w_{ji}(t) \propto s_i(t) - w_{ji}(t)(1 + \lambda).
\] (12)

Where, \( w_{ji} \geq 0 \). This learning rule is applied when an output neuron fires. The weight change is related to the presynaptic spike times received by the output neurons. This scenario reminds us of a popular local learning rule in a biologically plausible spiking neural network named spike-timing-dependent plasticity (STDP). In this STDP rule, the current synaptic weight controls the weight change.

Eq. 7 is used to implement a learning rule for adjusting the threshold, \( \theta \) (the Lagrange multiplier is considered as the threshold). The threshold learning rule shown in Eq. 13 provides an independent and sparse feature representation. The threshold is the same for all \( D \) neurons in the representation layer.
\[
\Delta \theta \propto \left( \sum_{j=1}^{D} z_j \right) - 1
\] (13)

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 presented by 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 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
\[
\text{WTA score}_j(t; W) = \frac{\exp(\sum_i w_{ji} \zeta_i(t))}{\sum_k \exp(\sum_i w_{ki} \zeta_i(t))}
\] (14)

\[
\zeta_i(t) = \sum_{t'} \frac{e^{-(t-t')}}{\tau}
\] (15)
where, $\zeta_i(t)$ is the excitatory postsynaptic potential (EPSP) generated by input neuron $i$ and the $t^i$s are the recent spike times of unit $i$ during a small interval $(t - \nu, t]$, where $\nu$ is 4 ms. The decay time constant, $\tau$, is set to 0.5 ms. In our network, the softmax value governs the time at which STDP occurs. If $\text{WTAscore}$ of a neuron is greater than the adaptive threshold, $\theta$, STDP is triggered and a spike is emitted. The softmax phenomenon is implemented to develop a winners-take-all circuit 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 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. [12] defines an STDP rule where the current synaptic weight controls the magnitude of the change. STDP events are triggered upon postsynaptic firing. Eq. [16] shows the final STDP rule derived from Eq. [12]. The weights fall in the range $[0, 1]$ and are initialized randomly between 0 and 1.

$$\Delta w_{ji} = \left\{ \begin{array}{ll} a \cdot (1 - w_{ji}(1 + \lambda)), & \text{if } s_i = 1 \\ a \cdot (-w_{ji}(1 + \lambda)), & \text{if } s_i = 0 \end{array} \right.$$  \hspace{1cm} (16)

$a$ is the learning rate. If $\lambda = 0$, the first and second adaptation cases increase and decrease the synaptic weight respectively (LTP and LTD). If $\lambda \rightarrow \infty$, then both cases are negative and decrease the weights down to the minimum value ($w_{ji} = 0$). The weight adjustment, at equilibrium, demonstrates a probabilistic interpretation as follows

$$E[\Delta w_{ji}] = 0 \leftrightarrow a \cdot P(s_i = 1|r_j = 1)(1 - w_{ji}(1 + \lambda)) - a \cdot P(s_i = 0|r_j = 1)w_{ji}(1 + \lambda) = a \cdot P(s_i = 1)r_j = 1(1 - w_{ji}(1 + \lambda)) - a \cdot (1 - P(s_i = 1)r_j = 1)w_{ji}(1 + \lambda) = 0 \leftrightarrow (1 + \lambda)w_{ji} = P(s_i = 1|r_j = 1)$$ \hspace{1cm} (17)

Therefore, the synaptic weight converges to the scaled probability of presynaptic spike occurrence given postsynaptic spike (LTP probability). From Eq. [18], 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 equilibrium point.

To show that the STDP rule (Eq. [16]) is consistent with the learning rule in Eq. [8], we rewrite the non-local rule with learning rate, $a$, as follows

$$\Delta w_{ji} = a \cdot y_j(x_i - w_{ji} - \lambda w_{ji})$$ \hspace{1cm} (19)

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 Eq. [19] and Eq. [16] (which is derived from Eq. [12]) 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. [19] at time $t$ is simplified to

$$\Delta w_{ji} = a(x_i - w_{ji} - \lambda w_{ji})$$ \hspace{1cm} (20)

Following Eq. [17] for calculating the expected weight change using the proposed STDP rule, where $r_j = 1$, we find that

$$E[\Delta w_{ji}] = a(P(s_i = 1) - w_{ji} - \lambda w_{ji})$$ \hspace{1cm} (21)

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,

$$E[\Delta w_{ji}] = a(x_i - w_{ji} - \lambda w_{ji}) = \Delta w_{ji}$$ \hspace{1cm} (22)

Which matches the weight change shown in Eq. [20]. 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 provided by Földiák [3] (Eq. 2).

For the threshold adaptation, following Eq. [13], the threshold learning rule can be written as

$$\Delta \theta = b(m_z - 1)$$

(23)

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. In the experiments, the initial threshold is set to 0.15.

5 Evaluation Metrics

5.1 Reconstructed image

The representation filter set, $W = \{w_1, w_2, ..., w_D\}$, 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, $W_{rec} = W^T$, 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) between the normalized original, $y_m$, and reconstructed, $\hat{y}_m$, patches as shown in Eqs. 24 and 25 respectively. A patch stands for $p^2$ spike train frequencies, $y_i$.

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

(24)

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

(25)

Where, $M$ 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. 26

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

(26)

The breadth tuning measure introduced by Rolls and Tovee [21] specifies the density of neural layer activity (Eq. 27) calculated by the ratio of mean, $\mu$, and standard deviation, $\sigma$, 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 heavy tail [22]. 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.

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

(27)

6 Experiments and Results

We ran two experiments using the MNIST [23] and the natural image [7] datasets to evaluate the proposed local representation learning rules embedded in the single-layer SNN. The intensities of
Experiment 1: MNIST dataset

Experiments were run using 5×5 patches from 28×28 MNIST digits. We used a random subset of the MNIST dataset divided into 15,000 training and 1000 testing images for learning and evaluating the model. The SNN consists of 25 (5×5 image patch) neurons in the input layer and $D = \{2^i, i = 3..7\}$ neurons in the representation layer. These variations of the network architecture (different $D$ values) determine under-complete to over-complete representations. Trained filters, after 100 and 15,000 iterations, for the network with 32 neurons in the representation layer are shown in Fig. 2a. The filters shown in this image tend to be orientation selective and extract different visual features. Fig. 2b shows the RMS reconstruction loss and statistical characteristics of the trained weights versus the log regularizer hyperparameter ($\log_{10}\lambda$). The RMS reconstruction loss values are less than 0.18 for $\lambda \leq 0.1$. The maximum and minimum synaptic weights after training are $1/(1 + \lambda)$ and 0 respectively as predicted by Eq. 18.

The three performance measures from the previous section 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. 2c. The reconstructed maps show high quality images comparable with the original images. The reconstruction loss measures for the SNNs with $D = 8$ through 128...
6.2 Experiment 2: Natural images

This experiment evaluates the proposed spiking representation model on $16 \times 16$ natural image patches [7]. Fig. 4a shows the trained representation filters. Except for the marked filters, the other filters have low correlation with each other. For visual assessments, Fig. 4b shows three 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. Minimum reconstruction loss belongs to the networks with $\{16, 32, 64\}$ neurons in the representation layer.

6.3 Comparisons

The proposed spiking representation learning method shows better performance than the traditional K-means clustering and the restricted Boltzman machine (RBM) while introducing local learning in time and space. Table 1 shows this comparison in terms of reconstruction loss (correlation and
Figure 4: Model’s performance on the natural image patches. (a): 32 filters after training. (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.

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

| $D$ | MNIST Corr. | MNIST RMS | Natural Corr. | Natural RMS |
|-----|-------------|-----------|---------------|-------------|
| 8   | 0.22        | 0.23      | 0.45          | 0.31        |
| 16  | 0.23        | 0.26      | 0.52          | 0.36        |
| 32  | 0.18        | 0.21      | 0.57          | 0.40        |
| 64  | 0.26        | 0.26      | 0.49          | 0.27        |
| 16  | 0.49        | 0.49      | 0.41          | 0.47        |
| 32  | 0.40        | 0.40      | 0.44          | 0.27        |
| 64  | 0.24        | 0.24      | 0.47          | 0.26        |

Our model outperforms the RBM and K-means methods except for the two cases (natural images) in which the RBM shows slightly better performance. Additionally, 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 [14]. The sparse representation introduced by King et al [6], which is a modified version of the SAILnet algorithm [5], reported an RMS reconstruction loss around 0.74 that is calculated based on the spike rates normalized to unit standard deviation (let’s 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. 23). If we change the rule to $\Delta \theta = b(m_z - q)$, 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 Conclusion

This paper proposed a novel STDP-based representation learning method embedded in an SNN and evaluated in two experiments to establish its initial viability. The learning rules were derived by constrained optimization incorporating a vector quantization-like objective function with regularization and a firing constraint. The learning rules include 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 layer implemented inhibition to represent sparse and independent visual features. The experimental results showed high performance of the proposed model such that it outperformed the K-means clustering and the RBM model and it compares favorably with the state-of-the-art spike-based representation learning approaches.

Although the proposed spiking representation learning was successful, 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.

References

[1] Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In Proceedings of the fourteenth international conference on artificial intelligence and statistics,
REFERENCES

[2] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. IEEE transactions on pattern analysis and machine intelligence, 35(8):1798–1828, 2013.

[3] Peter Földiák. Forming sparse representations by local anti-Hebbian learning. Biological Cybernetics, 64(2):165–170, 1990.

[4] H B Barlow. Unsupervised learning. Neural Computation, 1(3):295–311, 1989.

[5] Joel Zylberberg, Jason Timothy Murphy, and Michael Robert DeWeese. A sparse coding model with synaptically local plasticity and spiking neurons can account for the diverse shapes of V1 simple cell receptive fields. PLoS Comput Biol, 7(10):e1002250, 2011.

[6] Paul D King, Joel Zylberberg, and Michael R DeWeese. Inhibitory interneurons decorrelate excitatory cells to drive sparse code formation in a spiking model of V1. The Journal of Neuroscience, 33(13):5475–5485, 2013.

[7] Bruno A Olshausen and David J Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. Nature, 381(6583):607–609, 1996.

[8] Anthony J Bell and Terrence J Sejnowski. The “independent components” of natural scenes are edge filters. Vision Research, 37(23):3327–3338, 1997.

[9] Chr. von der Malsburg. Self-organization of orientation sensitive cells in the striate cortex. Kybernetik, 14:85–100, 1973.

[10] E. L. Bienenstock, L. N. Cooper, and Paul W. Munro. Theory for the development of neuron selectivity: Orientation specificity and binocular interaction in visual cortex. Journal of Neuroscience, 2(1):32–48, 1892.

[11] Timothee Masquelier. 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, 2012.

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

[13] Natalia Caporale and Yang Dan. Spike timing-dependent plasticity: a Hebbian learning rule. Annu. Rev. Neurosci., 31:25–46, 2008.

[14] Kendra S Burbank. Mirrored STDP implements autoencoder learning in a network of spiking neurons. PLoS Comput Biol, 11(12):e1004566, 2015.

[15] Cristina Savin, Prashant Joshi, and Jochen Triesch. Independent component analysis in spiking neurons. PLoS Comput Biol, 6(4):e1000757, 2010.

[16] Adam Coates and Andrew Y Ng. Learning feature representations with k-means. In Neural networks: Tricks of the trade, pages 561–580. Springer, 2012.

[17] P. Foldiak and D. Endres. Sparse coding. Scholarpedia, 3(1):2984, 2008.

[18] Barbara Hammer and Thomas Villmann. Generalized relevance learning vector quantization. Neural Networks, 15(8):1059–1068, 2002.

[19] Petra Schneider, Michael Biehl, and Barbara Hammer. Distance learning in discriminative vector quantization. Neural computation, 21(10):2942–2969, 2009.

[20] Edmund T Rolls and Martin J Tovee. Sparserness of the neuronal representation of stimuli in the primate temporal visual cortex. Journal of Neurophysiology, 73(2):713–726, 1995.

[21] Yann LeCun, Corinna Cortes, and Christopher J C Burges. The MNIST database. URL http://yann.lecun.com/exdb/mnist, 1998.

[22] Edmund T Rolls and Martin J Tovee. Sparserness of the neuronal representation of stimuli in the primate temporal visual cortex. Journal of Neurophysiology, 73(2):713–726, 1995.

[23] P. Foldiak and D. Endres. Sparse coding. Scholarpedia, 3(1):2984, 2008.

[24] Yann LeCun, Corinna Cortes, and Christopher J C Burges. The MNIST database. URL http://yann.lecun.com/exdb/mnist, 1998.