A Minimal Spiking Neural Network to Rapidly Train and Classify Handwritten Digits in Binary and 10-Digit Tasks

Amirhossein Tavanaei
The Center for Advanced Computer Studies
University of Louisiana at Lafayette
Lafayette, LA, USA

Anthony S. Maida
The Center for Advanced Computer Studies
University of Louisiana at Lafayette
Lafayette, LA, USA

***Abstract*—This paper reports the results of experiments to develop a minimal neural network for pattern classification. The network uses biologically plausible neural and learning mechanisms and is applied to a subset of the MNIST dataset of handwritten digits. The research goal is to assess the classification power of a very simple biologically motivated mechanism. The network architecture is primarily a feedforward spiking neural network (SNN) composed of Izhikevich regular spiking (RS) neurons and conductance-based synapses. The weights are trained with the spike timing-dependent plasticity (STDP) learning rule. The proposed SNN architecture contains three neuron layers which are connected by both static and adaptive synapses. Visual input signals are processed by the first layer to generate input spike trains. The second and third layers contribute to spike train segmentation and STDP learning, respectively. The network is evaluated by classification accuracy on the handwritten digit images from the MNIST dataset. The simulation results show that although the proposed SNN is trained quickly without error-feedbacks in a few number of iterations, it results in desirable performance (97.6%) in the binary classification (0 and 1). In addition, the proposed SNN gives acceptable recognition accuracy in 10-digit (0-9) classification in comparison with statistical methods such as support vector machine (SVM) and multi-perceptron neural network.**

Keywords—Spiking neural networks; STDP learning; digit recognition; adaptive synapse; classification

I. Introduction

Neural networks that use biologically plausible neurons and learning mechanisms have become the focus of a number of recent pattern recognition studies [1, 2, 3]. Spiking neurons and adaptive synapses between neurons contribute to a new approach in cognition, decision making, and learning [4-8].

Recent examples include the combination of rank order coding (ROC) and spike timing-dependent plasticity (STDP) learning [9], the calculation of temporal radial basis functions (RBFs) in the hidden layer of spiking neural network [10], and linear and non-linear pattern recognition by spiking neurons and firing rate distributions [11]. The studies mentioned utilize spiking neurons, adaptive synapses, and biologically plausible learning for classification.

Learning in the present paper combines STDP with competitive learning. STDP is a learning rule which modifies the synaptic strength (weight) between two neurons as a function of the relative pre- and postsynaptic spike occurrence times [12]. Competitive learning takes the form of a winner-take-all (WTA) policy. This is a computational principle in neural networks which specifies the competition between the neurons in a layer for activation [13]. Learning and competition can be viewed as two building blocks for solving classification problems such as handwritten digit recognition. Nessler et al. (2009) utilized the STDP learning rule in conjunction with a stochastic soft WTA circuit to generate internal models for subclasses of spike patterns [14]. Also, Masquelier and Thorpe (2007) developed a 5-layer spiking neural network (SNN) consisting of edge detectors, subsample mapping, intermediate-complexity visual feature extraction, object scaling and position adjustment, and categorization layers using STDP and WTA for image classification [15].

Auditory and visual signals have special authentication processes in the human brain. Thus, one or more neuron layers are required to model the signal sequences in one and two-dimensional feature vectors in addition to the learning phase. Wysoski et al. (2008 and 2010) proposed a multilayer SNN architecture to classify audiovisual input data using an adaptive online learning procedure [16, 17]. The combination of Izhikevich's neuron firing model, the use of conductance-based synaptic dynamics, and the STDP learning rule can be used for a convenient SNN for pattern recognition. As an example, Beyeler et al. (2013) developed a decision making system using this combination in a large-scale model of a hierarchical SNN to categorize handwritten digits [18]. Their SNN architecture consists of 3136 plastic synapses which are trained and simulated in 500 (ms). They trained the system by 10/100/1000/2000 samples of the MNIST dataset in 100 iterations and achieved a 92% average accuracy rate. In another study, Nessler et al. (2013) showed that Bayesian computation is induced in their proposed neural network through STDP learning [2]. They evaluated the method, which is an unsupervised method for learning a generative model, by MNIST digit classification and achieved an error rate of 19.86% (80.14% correctness). Their proposed neural network for this experiment includes 708 input spike trains and 100 output neurons in a complete-connected feedforward network.

Some previous studies (c.f. [18], [2]) have attempted to develop an autonomous and strong artificial intelligence based on human brain anatomy in a large network of neurons and synapses. However, two inevitable and important aspects of the brain simulation are 1) the size of the network that is, number of the neurons and synapses, and 2) rapid learning and decision making. In some cases, a concise network is needed to be tuned and make a decision quickly in a special environment such as binary classification in the real time robot vision. Although large networks provide convenient circumstances for handling the details and consequently desirable performance, they are resource intensive. Our goal is to develop a fast and small neural network to extract useful features, learn their statistical structure, and make accurate classification decisions quickly.

This paper presents an efficient 3-layer SNN with a small number of neurons and synapses. It learns to classify handwritten MNIST digits. The training and testing algorithms perform weight adaptation and pattern recognition in a time and memory efficient manner while achieving good performance. The proposed SNN provides a robust solution for the mentioned challenge in three steps. First, the digit image is converted to spike trains so that each spike is a discriminative candidate of a row pixel in the image. Second, to reduce the network size and mimic human perception of the image, the spike trains are integrated to a few sections. In this part, each output spike train specifies a special part of the image in the row order. Third, training layer which involves STDP learning, output spike firing, and WTA competition by inhibitory neuron modifies a fast pattern detection strategy. The remarkably simple SNN is implemented for binary (“0, 1” c.f. Fig. 1) and 10-digit task (0-9) handwritten digit recognition problem to illustrate efficiency of the proposed strategy in primitive classifications. Furthermore, the obtained results are compared with statistical machine learning models in the same circumstances (same training/testing data without feature mapping) to depict the trustworthy of our model in similar situations.

II. Spiking Neural Network Architecture

The proposed SNN architecture is shown in Fig. 2. It includes three components: 1) a neural spike generator, 2) image segmentation, and 3) learning session and output pattern creator. Theory and implementation of each component will be explained.

A. First layer: Presynaptic spike train generator

Each row in the 28×28 binary image (c.f. Fig. 3) is transcribed into a spike train in a left-to-right fashion. Fig. 3 shows an example digit “0” with $N \times M$ binary pixels. Rows are converted to spike trains where a pixel value of “1” represents a spike occurrence. To apply the discriminative features of the image in a small network architecture, the digit image is recoded to N presynaptic spike trains with $A \times M$ discrete time points. A controls the interspike spacing and is interpreted as the refractory period of the neuron. In summary, the first layer converts the binary digit image into N rows of spike trains according to the white pixels of the digit foreground.

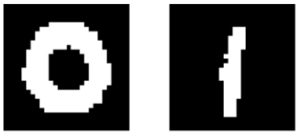


Fig. 1. Sample of handwritten digit images “0” and “1”

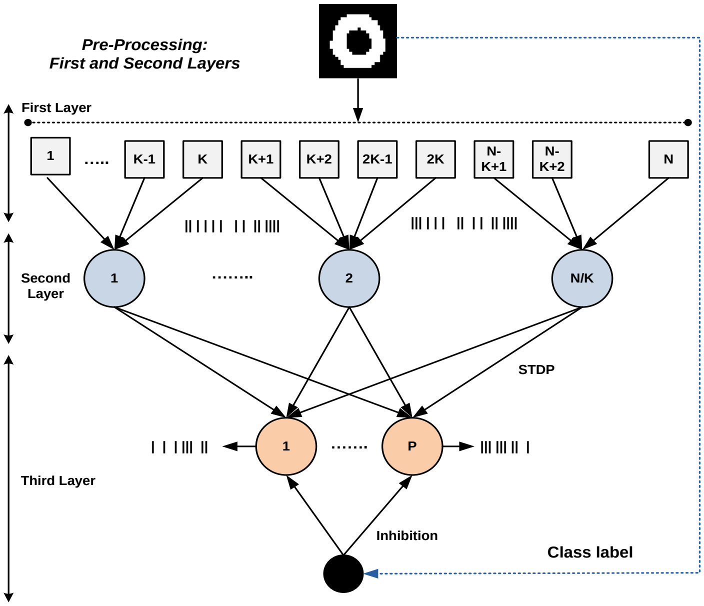


Fig. 2. Supervised SNN architecture containing spike transcription layer, spike train segmentation, STDP learning, output pattern firing, and inhibitory neuron. N: number of rows. K: number of adjacent rows connected to one neuron. P: number of classes. Black circle inhibits all output neurons except the one designated by the class label

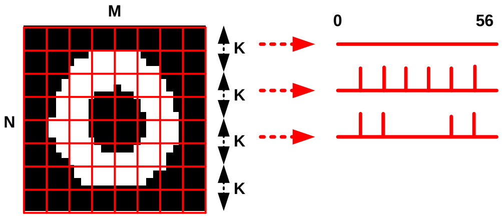


Fig. 3. A digit image with $N \times M$ pixels divided into N/K segments, consisting of k rows per segment

B. Second Layer: Image segmentation

The first layer generates N spike trains, where N is the number of the rows, encoding the image features row by row. However, it does not consider the slight change in orientation and thickness of the digit foreground in comparison with its background. To address this, the second layer illustrated in Fig. 2 merges every K spike trains (rows) onto one neuron. Then the digit image is segmented into N/K parts while preserving the spike train order. This preprocessing layer reduces the number of trainable parameters. Fig. 4 shows three instances of digit “1” (from the MNIST). The second layer converts these different shapes to similar N/K rows of spike trains. In addition, combining the sequential rows increases the network flexibility in pattern classification by decreasing its size. In summary, without the second layer, spike trains are sensitive to noise, outlier data, and diverse writing styles.

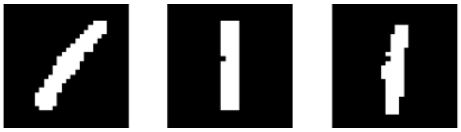


Fig. 4. Three instances of the handwritten digit “1”

To simulate the conductance of the postsynaptic membrane of the second layer units upon receipt of a presynaptic spike, the α-function is used. Equation (1) and Fig. 5 show the formula and graph of the postsynaptic conductance based on the α-function.

$$G_{\text{syn}}(t) = K_{\text{syn}}\, t\, e^{-t/\tau} \tag{1}$$

K_{syn} controls the peak conductance value and τ controls the time at which the peak conductance is reached.

Additionally, the total conductance of N input synapses with $N_{\text{rec},k\,(k=1:N)}$ spikes is calculated by (2)

$$G^{\text{tot}}(t) = \sum_{k=1}^{N} \sum_{j=1}^{N_{\text{rec},k}} K_{\text{syn},k}\,(t - t^{\text{f}}_{k,j})\, e^{-(t-t^{\text{f}}_{k,j})/\tau} \tag{2}$$

where t^{f} is spike firing time. This formula performs linear spatio-temporal summation across the received spike train. The total postsynaptic current is obtained by (3)

$$I_{\text{syn}}(t) = \sum_{k=1}^{N} E_{\text{syn},k} G^{\text{tot}}_{\text{syn},k}(t) - V(t) \sum_{k=1}^{N} G^{\text{tot}}_{\text{syn},k}(t) \tag{3}$$

In this investigation, spike generation in the second and third layer is controlled by Izhikevich's model [19] (4) specified by two coupled differential equations.

$$\begin{aligned} C\frac{dV}{dt} &= k(V - V_{\text{rest}})(V - V_{\text{th}}) - U + I_{\text{inj}} \\ \frac{dU}{dt} &= a\left[b(V - V_{\text{rest}}) - U\right] \end{aligned} \tag{4}$$

There is also a reset condition after a spike is generated, given in (5).

$$\begin{aligned} &if\ \ V \ge V_{\text{peak}} : V = c,\ U = U + d \\ &an\ AP\ is\ emitted \end{aligned} \tag{5}$$

Where, V denotes membrane potential and U specifies the recovery factor preventing the action potential (AP) and keeping the membrane potential close to the resting point. *"a", "b", "k", "c", "d"*, and "V_{peak}" are predefined constants controlling the spike shapes. The time of spike events is taken to occur at reset.

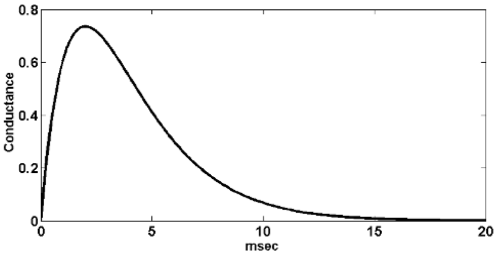


Fig. 5. Conductance graph with K_{syn}=1 (α-function) and τ=2 (msec)

C. Third layer: Learning and output neurons

Third layer of the SNN shown in Fig. 2 learns the input spike patterns and generates output spikes based on the evolving synaptic weights. STDP is controlled by relative pre- and postsynaptic spike times. Equation (6) specifies that postsynaptic spikes which follow presynaptic spikes cause the synaptic weight to be increased (LTP) and in contrast, synapses are weakened when presynaptic spike occurs after postsynaptic spike generation time (LTD).

$$\Delta K_{ji} = \begin{cases} 0.01A_{\text{ltp}} \exp(-(|\,t_j^f - t_i^f\,|)/_{\tau+})\ ,\ t_j^f - t_i^f > 0 \\ 0.01A_{\text{ltd}} \exp(-(|\,t_j^f - t_i^f\,|)/_{\tau-})\ ,\ t_j^f - t_i^f < 0 \end{cases} \tag{6}$$

In (6), A_{ltp} and τ+ (A_{ltd} and τ-) are maximum and time constant strengthening (weakening) constants respectively. In addition, the change in synaptic weights contributes to change in conductance amplitude, K_{syn}, in α-function derivation. The learning strategy used in this investigation is basically derived from the STDP concept. The proposed network in the first layer emits spike trains with maximum M spikes, where M is the number of columns in the image matrix. The second layer presents new information of spike trains at which spikes depict explicit foreground pixel information. In addition, the membrane potential is accumulated based on the received action potentials. Therefore, in the proposed minimal network architecture which models the patterns by exact object coordinates, a modified STDP learning is defined in (7).

$$\Delta K_{ji} = \begin{cases} \dfrac{A_{\text{ltp}}}{K_{ji}} \exp(-(|\,t_j^f - t_i^f\,|)/_{\tau+})\ ,\ \begin{pmatrix} P_j = 1, \\ t_i^f \in [\min(t_j^{f-1}, t_j^f - \sigma), t_j^f\,] \end{pmatrix} \\ -A_{\text{ltd}} \exp(-\beta)\ ,\ \begin{pmatrix} P_j = 1, \\ t_i^f \notin [\min(t_j^{f-1}, t_j^f - \sigma), t_j^f\,] \end{pmatrix} \end{cases} \tag{7}$$

where A_{ltp}, A_{ltd}, β>1, and σ are constant parameters. In (7), if output neuron P_j fires, the synaptic weights can be either increased or decreased. Presence of the presynaptic spikes in the σ time interval before current time strengthens the synaptic conductance. In contrast, absence of the presynaptic spikes reduces the synaptic conductance. To prevent aliasing between σ time interval and previous output spike, presynaptic spikes after the last emitted postsynaptic spike are counted. Also, the inverse value of the conductance amplitude (K_{ji}) controls rate of the LTP in the high conductance conditions.

In addition, output neurons in the third layer receive *N/K* spike trains and generate *P* (as number of the output patterns) output spike trains based on the current synaptic weights, presynaptic spike trains, and Izhikevich's model for the spike generation mechanism. Furthermore, each output neuron specifies one class. The learning strategy in the output layer is supervised. This is implemented by using an inhibitory neuron that imposes a WTA discipline across the output units. Specifically, the inhibitory unit uses the category label for the current training stimulus to inhibit all the output neurons that do not match the label. The net learning effect is that the nonmatching units undergo LTD, while the single matching unit undergoes LTP. Equation (8) specifies the LTD rule for inactive neurons. The synaptic conductance reduction in this formula depends on the presynaptic spikes "y_i" conveyed to the neuron in the time interval [$\min(t_j^{f-1}, t_j^f - \sigma)$, t_j^f].

$$C = \frac{-\gamma}{\sum_{i=\min(t_j^{f-1},t_j^f-\sigma)}^{t_j^f} y_i} \quad , \quad \Delta K'_{ji} = -0.01e^C \tag{8}$$

Where, γ is rate of the inhibition. In the last step, conductance magnitude of the synapses (which can be interpreted as synaptic weights) are updated by (9)

$$K_{ji}^t = K_{ji}^{t-1} + \mu \Delta K_{ji}^t \tag{9}$$

where, μ is learning rate. Finally, the result will be an array of synaptic weights and output spike patterns. Fig. 6 shows pseudocode for the SNN architecture and learning strategy.

D. Justification

Digits belonging to the same categories are not entirely similar due to different handwriting styles, variations in orientation, and variations in line thickness. The second layer converts the various images of one digit into a small number of similar patterns. It combines *K* spike trains to adjust the thickness and presents the image in *N/K* row segments. The slight diversity of the images in a digit category can be manipulated by foreground adjustment in height and width which is implemented by row segmentation and regular spiking (RS) neurons respectively. In addition, *N* input spike trains are mapped to *N/K* spike trains to minimize the network size.

To explain the learning procedure and justify its function in classification, an example consisting of the digits 2, 4, 1, and 9 is described step by step. Fig. 7 shows the digits. They are divided into 4 horizontal segments which are mapped into 4 adaptive synapses. If an output spike occurs, the synapses carrying more frequent and closer presynaptic spikes (white pixels) before the output spike have more casual effects. Thus, their weights are increased based on the LTP rule. For example, the synapses {1,4}, {3}, and {1,2} in digits 2, 4, 9 respectively carry frequent presynaptic spikes, so their weights are increased more than the other synapses in each digit. In digit “1”, all of the synapses have analogous influences onto the output neuron firing. So, the synaptic weights should be almost unbiased. After the first training period including weight augmentation and reduction, in the next iteration, the synaptic weights are tuned better according to the input digit patterns. Additionally, synaptic weights, which are connected to the same neurons in the second layer and different output neurons, are adapted in a competition due to the inhibitory neuron. Therefore, the synaptic weights demonstrate discriminative weight vectors for different digit patterns. In Fig. 7, some nominal synaptic weights (Ex. {0.20, 0.20, 0.45, 0.15} for digit 4) have been shown.

In the test session, if the digit spike trains are matched to the synaptic weights, the target output neuron releases a spike train close to the target pattern. Otherwise, due to discriminative synaptic weights, if the input spike trains are not compatible with the synaptic weights and target pattern, the output neuron might release a spike train either with 0 frequency or arbitrary pattern. Finally, the digit having maximum correlation with training data will be recognized in a small and fast neural network.

III. Experiments and Results

A subset of the MNIST machine learning data set consisting of handwritten digit images was used for evaluation of the proposed method [20]. Digital images in the dataset are 28 pixels in height and 28 pixels in width for a total of 784 pixels composing each grayscale image.

A. Binary classification

In the first experiment, 750 images of the digits “0” and “1” were sampled. Each grayscale image was converted to a binary image by applying the middle point threshold (threshold pixel=128). The 750 digit samples were divided into training and testing sets by 3-fold cross validation to guarantee the generality of the method.

The first layer scans the rows pixel by pixel and generates spikes where the digit points occur. Pixel values equal to 1 denote spike occurrences. In addition, a refractory period, *A*, is assumed to be 2 (ms). Therefore, the spike trains represent a row fall into a 28×2=56 (ms) temporal window. Fig. 3 gives an example of spike train generation for a sample digit “0”. Finally, 28 spike trains with 56 (ms) discrete time points are obtained as presynaptic spikes conveyed to the second layer.

To segment the image into groups of rows, presynaptic spikes are collected to the *N/K* layer-2 neurons where *N*=28 and *K*=4. That is, every 4 consecutive sequential spike trains are connected to one neuron in the second layer. Spike generation of the neuron in this layer is computed by Izhikevich’s RS model with parameters given in Table 1. Spike trains from seven layer-2 neurons submit information to the output neurons in the layer 3.

```
Function OneDataPassTraining(image, & weights):OutputSpike
{
    [N,M]=size(image);
    r=2; % refractory period
    % Layer 1
    for each row of the B/W image
        spikes=generate spikes in r*M time points (1: spike occurrence)
    %Layer 2
    for i=1:N/K
    {
       for j=1:K
            preSpikes{i}.append(spikes{(i-1)*K+j})
        middleSpikes{i}=Izhikevich’s model (preSpikes{i});
    }
    %Layer 3
    for p=1:#classes
    {
        OutputSpike=Izhikevich’s model (middleSpikes);
        STDP learning for target output
        Inhibition for non-target output based on STDP
        weights=update Synaptic weights
    }
```

Fig. 6. SNN pseudocode for handwritten digit classification

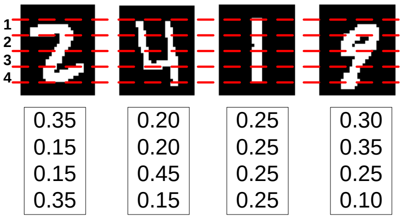


Fig. 7. Synaptic weights for digits 2, 4, 1, and 9. The intervals between dashed lines specify synapses. The numbers below each digit denote the synaptic weights

The output neurons use the same parameters in Izhikevich's model of the second layer to generate the spikes. In the third layer, synaptic weights projecting to the output neurons are initialized uniformly and updated by the STDP rule with parameters of LTP and LTD in Table 2. Furthermore, the inhibition neuron prevents the non-target (0 or 1) neuron to fire while receiving the presynaptic spikes. Hence, synaptic weights are changed according to the relative pre- and postsynaptic spike times.

After one batch of training (500 training samples), 14 synaptic weights (7 synapses for output "0" and 7 synapses for output "1") and a set of output spike patterns for "0" and "1" are obtained. Fig. 8 shows the simulation results (with ΔT=0.1 (msec)) of output spike trains of some handwritten digit images in "0" and "1" categories (each row shows a spike train). The illustrated spike trains in Fig. 8 show 1) specific first spike times and 2) discriminative spike time patterns for class "1" and class "0". Therefore, extracted target patterns are appropriate sources for pattern recognition. The output patterns of the testing samples are compared with the average target patterns for each class. Finally, the similarity measure denotes the objective function for the classification that is shown in (10).

$$Similarity^{pattern} = \frac{1}{\sqrt{\frac{1}{T}\sum_{i \in spike\,train}(Target_i^{pattern} - Output_i)^2}} \qquad (10)$$

Where T is size of the target pattern.

Five different simulation step sizes (ΔT=0.05, 0.1, 0.2, 0.5, 1 (msec)) were studied. Table 3 specifies scaled synaptic weights connected to the output neurons "0" and "1" in the five temporal resolutions. Synaptic weights in Table 3 claim that ΔT in range of 0.05 to 0.5 (msec) give discriminative weight vectors for different classes. On the other hand, in ΔT=1 (msec), the training is biased to "0" because the simulation step size is so large and the learning procedure is not converged.

A subset of disjoint training and testing data was applied to the trained SNN to evaluate the accuracy rate of the proposed method. The results are shown in Table 4. The average accuracy rate is 97.6 for the testing sets. In addition, values of ΔT in the range of 0.05 to 0.5 (msec) give acceptable performance. ΔT=1 (msec), as explained, is not applicable. According to the results in Table 4, ΔT=0.2 (msec) has the best performance. It is also a fast neuron simulation for training and testing sessions.

TABLE I. REGULAR SPIKING NEURON PARAMETERS

Parameter	**Value**	**Parameter**	**Value**
V_{rest}	-60 (mv)	*a*	0.03
$V_{threshold}$	-40 (mv)	*b*	-2
V_{peak}	35 (mv)	*c*	-50
C	100	*d*	100
K	0.7	U_0	0
ΔT	0.05, 0.1, 0.2, 0.5, 1	I_{inj}	0

TABLE II. STDP PARAMETERS

Parameter	**value**
A	103
B	-40
τ+	14
τ-	34
K Default	10

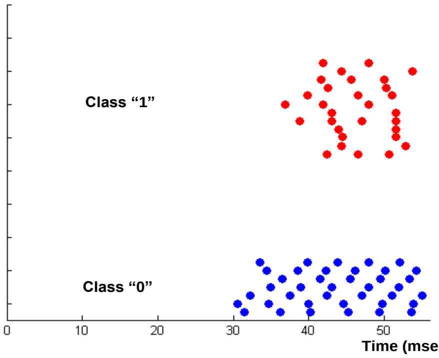


Fig. 8. Example output spike trains for the digits "0" (Blue) and "1" (Red) after learning. ΔT=0.1 (msec)

TABLE III. SYNAPTIC WEIGHTS (AFTER TRAINING) PROJECTING TO THE OUTPUT NEURONS REPRESENTING CATEGORIES "0" AND "1" IN DIFFERENT SIMULATION STEP SIZES (ΔT). EACH COLUMN INDICATES THE IMPORTANCE OF ONE OF THE N/K=7 IMAGE SEGMENTS TO EACH CATEGORY. THE BOLD WEIGHTS SHOW ACCEPTABLE LEARNING. ΔT IMPACTS ON MEMBRANE POTENTIAL COMPUTATION AND SYNAPTIC WEIGHT ALTERNATIONS. THEREFORE, EACH COLUMN SHOWS SOME SLIGHT VARIATIONS IN THE SYNAPTIC WEIGHTS. HOWEVER, THE SYNAPTIC WEIGHTS FOR EACH SIMULATION SHOW SEPARATE CATEGORIES

ΔT	**Digit**	**Syn1**	**Syn2**	**Syn3**	**Syn4**	**Syn5**	**Syn6**	**Syn7**
0.05	0	**0.00**	**1.30**	**6.60**	**5.24**	**0.26**	**6.60**	**0.00**
	1	**0.00**	**0.00**	**0.00**	**10.19**	**4.29**	**5.52**	**0.00**
0.1	0	**0.00**	**2.98**	**5.66**	**4.17**	**1.55**	**5.66**	**0.00**
	1	**0.00**	**0.00**	**0.00**	**10.43**	**4.20**	**5.37**	**0.00**
0.2	0	**0.00**	**3.69**	**5.94**	**4.16**	**0.26**	**5.94**	**0.00**
	1	**0.00**	**0.00**	**0.00**	**9.83**	**7.36**	**2.81**	**0.00**
0.5	0	**0.00**	**2.42**	**7.24**	**1.05**	**1.27**	**8.03**	**0.00**
	1	**0.00**	**0.00**	**0.00**	**17.40**	**2.25**	**0.36**	**0.00**
1	0	0.00	5.06	5.06	4.59	0.24	5.06	0.00
	1	0.03	0.00	0.00	0.00	0.00	0.00	19.97

TABLE IV. ACCURACY RATE OF THE PROPOSED SNN FOR HANDWRITTEN DIGIT CLASSIFICATION OF HANDWRITTEN DIGITS "0" AND "1"

ΔT	Accuracy Rate (Testing Data) %	Accuracy Rate (Training Data) %
0.05	**97.20**	---
0.1	**97.60**	---
0.2	**98.00**	**99.00**
0.5	**97.60**	---
1	56.00	---

B. 10-digit classification task

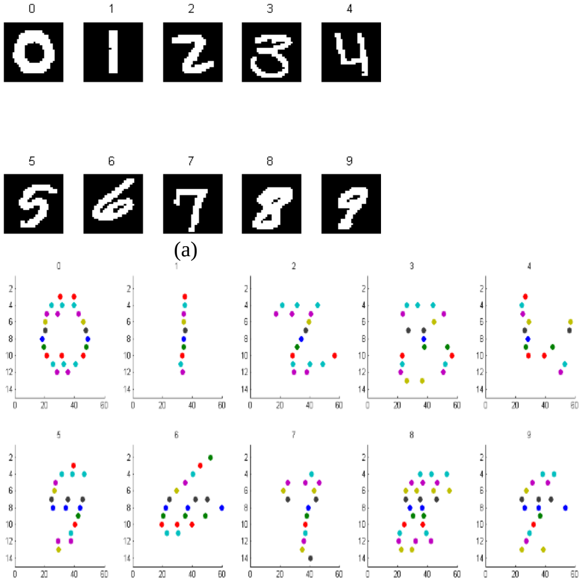
In the second experiment, 320 image samples of the MNIST handwritten digits were randomly selected and converted to binary images. The first and second layers of the SNN are the same as in the binary classification experiment except the segmentation factor, *K*, is set to 2. Therefore, the learning component consists of 28/2=14 adaptive synapses connected to 10 output neurons representing the digits 0 to 9 (140 adaptive synapses total and 24 layer-2 neurons). According to the mentioned theory, the second layer should generate candidate spike trains for a large variety of the input patterns. Fig. 9b illustrates 14 spike trains in the second layer which show a schematic of the input digits in Fig. 9a.

These discriminative spike trains invoke STDP learning in the next layer to adapt the synaptic weights and generate distinguishable spike patterns for digit categories (0-9). Fig. 10 shows the convergence scenario of the training process in 1000 iterations. This chart determines total distance between synaptic weights in sequential trials that is calculated by (11). It is concluded that, the training algorithm converges in 84 iterations and more training trials will not change the synaptic weights considerably. The synaptic weight matrix after 84 training iterations is shown in Table 5.

$$\Delta W^{t}_{Total} = \sum_{i=0}^{9} \left(\sum_{j=1}^{N/K=14} (w^{t}_{ij} - w^{t-1}_{ij})^2 \right)^{\frac{1}{2}} \qquad (11)$$

The adjusted synaptic weights and input digits provide 10 patterns of output spike trains shown in Fig. 11. The membrane potential and spike times in Fig. 11 illustrates discriminative patterns for different digits. For example, spikes in time stream of the digit “1” are close together in the center of the time window because all of the presynaptic spikes are gathered in a small range of simulation time (30-40 (ms)).

However, the proposed method as a minimal SNN architecture with 10 × 14 adaptive synapses (14-D weight vector) is designed for small classification problems such as binary categorization, not optimized for 10 categories, the performance of 10-digit task is 75.93% in average.



(a)

(b)

Fig. 9. a) Handwritten digits 0-9. b) 14 spike patterns emitted from RS neurons of the second layer

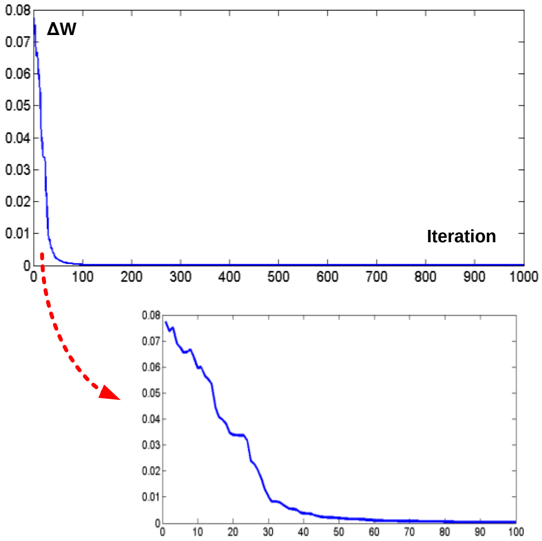


Fig. 10. Convergence plot obtained from 1000 iterations

TABLE V. SYNAPTIC WEIGHTS OF 14 SYNAPSES OF 10 DIGITS 0-9. THE FIRST TWO SYNAPTIC WEIGHTS AND THE LAST ONE ARE SMALLER THAN OTHER SYNAPSES BECAUSE THEY MOSTLY CONVEY BACKGROUND INFORMATION. THUS, RELATIVELY MUCH WEAK SYNAPSES PROVIDE A FAST METHOD OF BACKGROUND ELIMINATION IN THE ROW ORDER

Syn **Digit**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**
1	0.008	0.008	0.041	0.028	0.033	0.070	0.109	0.137	0.158	0.155	0.132	0.105	0.008	0.008
2	0.009	0.010	0.062	0.147	0.100	0.034	0.026	0.051	0.147	0.190	0.142	0.063	0.009	0.009
3	0.009	0.009	0.130	0.190	0.083	0.045	0.076	0.062	0.018	0.026	0.122	0.186	0.033	0.010
4	0.010	0.010	0.010	0.028	0.082	0.138	0.187	0.208	0.159	0.055	0.036	0.038	0.029	0.010
5	0.008	0.008	0.020	0.065	0.092	0.117	0.137	0.152	0.051	0.105	0.144	0.088	0.008	0.008
6	0.009	0.033	0.051	0.043	0.051	0.071	0.132	0.165	0.171	0.151	0.097	0.011	0.009	0.009
7	0.010	0.010	0.010	0.079	0.199	0.114	0.069	0.045	0.051	0.044	0.066	0.091	0.151	0.063
8	0.006	0.006	0.032	0.084	0.114	0.127	0.113	0.055	0.089	0.113	0.113	0.112	0.029	0.006
9	0.008	0.008	0.008	0.059	0.133	0.154	0.152	0.155	0.052	0.027	0.041	0.072	0.124	0.008
0	0.008	0.008	0.021	0.047	0.083	0.092	0.122	0.126	0.114	0.146	0.156	0.061	0.008	0.008

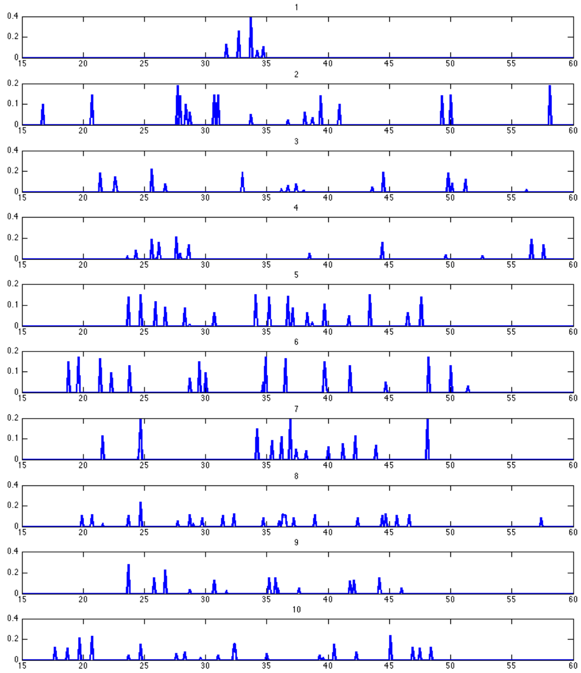


Fig. 11. Membrane potential and spike patterns for digits 0-9 after training

C. Comparison with other models

To compare our model with statistical machine learning strategies, the same training and testing datasets were applied to 1) a support vector machine (SVM) which maximizes the border distances between the classes [21]; and 2) a back propagation multi perceptron artificial neural network (BP-ANN) which learns the synaptic weights using error-feedback adjustment [22]. The obtained results are shown in Table 6. They have been implemented by the R software package [23, 24]. If more data are used in statistical models (with modified parameters) and some preprocessing algorithms such as principle components analysis (PCA) are applied, the performance should be higher than the rates reported in Table 6. However, based on the same situations at which the SNN performs, the SVM and ANN methods show accuracy rates that are slightly lower than the proposed SNN. We claim that the minimal SNN in this investigation has sufficient capacity to be improved more by the required preprocessing and experiments while using the biologically plausible principles.

IV. Conclusion and Discussion

A minimal time and memory efficient SNN architecture for classification was presented. This research shows that phenomenological STDP in a minimal model can support pattern recognition learning. The algorithms and neuron models were chosen to be biologically plausible. The proposed method represented an architecture which specifies a remarkably simple and fast network in comparison with previous investigations.

TABLE VI. Accuracy of 10-Digit Recognition Using Statistical Machine Learning Models. 200 Training Data Without Preprocessing

Method	Acc. %	Description
SVM	73.44	Polynomial kernel n=3.
ANN	70.87	40 hidden neurons, Decay=0.0001, 550 iterations.

Our SNN was applied to handwritten digit classification for a subset of images in the MNIST dataset. First, the initial layer interpreted the image logically based on the exact foreground pixel locations. Therefore, digit image was scanned row by row to generate the spikes as impulse reaction to the object perception. Also, this layer extracted the feature spikes directly from the image and represented a quick and natural image perception without complex computations. Second, every K (Ex. 2 or 4) spike trains were accumulated in a sequential order to provide the segmentation aspect of object detection in order to reduce the working space using Izhikevich's neuron model. This part of the network guaranteed to reduce number of the computational neurons and kept the order of the image segments from top to bottom. This layer provided a structure to produce set of spike trains invariant to diverse handwriting styles, outlier points, and slight changes in foreground orientation and thickness. The extracted sections mimicked digital scanning methods in a fast and implementable manner. Third, STDP learning and inhibitory neuron prepared the required environment for training the network and competition among dissimilar categories. The third layer's algorithm focused on supervised learning of the summarized input patterns. Additionally, the STDP rule was applied in two different sets of the synapses (connected to the target and non-target neurons) simultaneously and the training process converged after a small number of iterations. Thus, The SNN was tuned to categorize the input spike patterns quickly and it did not need many feature spike trains.

In summary, the introduced strategy was implemented in a simple and fast way due to the small number of the neurons and adaptive synapses (totally, {10 and 25} computational neurons and {14 and 140} adaptive synapses for binary and 10-digit classifications respectively). Finally, evaluation of the presented model demonstrated admirable performance of 98.0% maximum and 97.6% average accuracy rates for binary ("0" and "1") handwritten digit recognition. Furthermore, in spite of the minimal architecture of the presented SNN, acceptable performance of 75.93% was obtained in 10-digit recognition. The comparison between accuracy rate of the proposed method and statistical machine learning approaches (basic models without preprocessing and a small number of training data) determined slightly better performance of our SNN in the same and basic situations as well as incremental learning ability of the SNN. The minimal SNN worked much better for binary classification than 10-digit task. However, the reported results showed the potential capability of the SNN to be shrunk and work fast in training and prediction phases.

Therefore, the proposed SNN architecture and learning procedure can be a trustworthy model for classification due to its simple structure, quick feature extraction and learning, robust synaptic adaptation, and feasible implementation on VLSI chips in the future experiments.

References

[1] Maass, Wolfgang. "Networks of spiking neurons: the third generation of neural network models." Neural networks 10.9 (1997): 1659-1671.

[2] Nessler, Bernhard, et al. "Bayesian computation emerges in generic cortical microcircuits through spike-timing-dependent plasticity." PLoS computational biology 9.4 (2013): e1003037.

[3] Kappel, David, Bernhard Nessler, and Wolfgang Maass. "STDP installs in winner-take-all circuits an online approximation to hidden markov model learning." PLoS computational biology 10.3 (2014): e1003511.

[4] Gerstner, Wulfram, and Werner M. Kistler. “Spiking neuron models: Single neurons, populations, plasticity”. Cambridge University Press, 2002.

[5] Rozenberg, Grzegorz, Thomas Back, and Joost N. Kok. “Handbook of natural computing”. Springer Publishing Company, Incorporated, 2011.

[6] Wade, John J., et al. "SWAT: a spiking neural network training algorithm for classification problems." Neural Networks, IEEE Transactions on 21.11 (2010): 1817-1830.

[7] Yousefi, Ali, Alireza A. Dibazar, and Theodore W. Berger. "Synaptic dynamics: Linear model and adaptation algorithm." Neural Networks 56 (2014): 49-68.

[8] Chrol-Cannon, Joseph, and Yaochu Jin. "Computational modeling of neural plasticity for self-organization of neural networks" BioSystems 125 (2014): 43-54.

[9] Kasabov, Nikola, et al. "Dynamic evolving spiking neural networks for on-line spatio-and spectro-temporal pattern recognition." Neural Networks 41 (2013): 188-201.

[10] Wang, Jinling, et al. "An Online Supervised Learning Method for Spiking Neural Networks with Adaptive Structure." Neurocomputing (2014).

[11] Vázquez, Roberto A. "Pattern recognition using spiking neurons and firing rates." Advances in Artificial Intelligence–IBERAMIA 2010. Springer Berlin Heidelberg, 2010. 423-432.

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

[13] Maass, Wolfgang. "On the computational power of winner-take-all." Neural computation 12.11 (2000): 2519-2535.

[14] Nessler Bernhard, Michael Pfeiffer, and Wolfgang Maass. "STDP enables spiking neurons to detect hidden causes of their inputs." Advances in neural information processing systems. 2009.

[15] Masquelier, Timothée, and Simon J. Thorpe. "Unsupervised learning of visual features through spike timing dependent plasticity." PLoS computational biology 3.2 (2007): e31.

[16] Wysoski, Simei Gomes, Lubica Benuskova, and Nikola Kasabov. "Fast and adaptive network of spiking neurons for multi-view visual pattern recognition." Neurocomputing 71.13 (2008): 2563-2575.

[17] Wysoski, Simei Gomes, Lubica Benuskova, and Nikola Kasabov. "Evolving spiking neural networks for audiovisual information processing." Neural Networks 23.7 (2010): 819-835.

[18] Beyeler, Michael, Nikil D. Dutt, and Jeffrey L. Krichmar. "Categorization and decision-making in a neurobiologically plausible spiking network using a STDP-like learning rule." Neural Networks 48 (2013): 109-124.

[19] Izhikevich, Eugene M. "Simple model of spiking neurons." IEEE Transactions on neural networks 14.6 (2003): 1569-1572.

[20] LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). “Gradient-based learning applied to document recognition”. Proceedings of the IEEE, 86, 2278–2324.

[21] Bishop, Christopher M. “Pattern recognition and machine learning”. New York: Springer, 2006.

[22] Cross, Simon S., Robert F. Harrison, and R. Lee Kennedy. "Introduction to neural networks." The Lancet 346.8982, pp. 1075-1079, 1995.

[23] Karatzoglou, Alexandros, David Meyer, and Kurt Hornik. "Support vector machines in R." 2005.

[24] Günther, Frauke, and Stefan Fritsch. "Neuralnet: Training of neural networks." The R Journal 2.1, pp.30-38, 2010.