Pattern Recognition Computation in A Spiking Neural Network with Temporal Encoding and Learning

Qiang Yu
Department of Electrical and Computer Engineering
National University of Singapore
Singapore 117576
Email: yu.qiang@nus.edu.sg

K.C. Tan
Department of Electrical and Computer Engineering
National University of Singapore
Singapore 117576
Email: eletankc@nus.edu.sg

Huajin Tang
Institute for Infocomm Research
Agency for Science Technology and Research (A*STAR)
Singapore 138632
Email: htang@i2r.a-star.edu.sg

Abstract—Many conventional methods have been widely studied to solve the pattern recognition task, but most of them lack the biological plausibility. This paper presents a spiking neural network of integrate-and-fire neurons to perform pattern recognition. A biologically plausible supervised synaptic learning rule is used so that neurons can efficiently make a decision. The whole system contains encoding, learning and readout. It can classify complex patterns of activities stored in a vector, as well as the real-world stimuli. We test the performance of the network with digital images from the MNIST and images of alphabetic letters. It turns out to be able to classify these patterns correctly. In addition, the synaptic dynamics is shown to be compatible with many experimental observations on induction of long-term modifications, like spike-timing-dependent plasticity (STDP).

I. INTRODUCTION

Pattern recognition is a general task that assigns an output value to a given input pattern. Several conventional methods are used to implement pattern recognition, such as maximum entropy classifier, naive Bayes classifier, decision trees, support vector machines and perceptrons. However, most of these methods lack biological plausibility. In this case, we consider an application to pattern recognition using spiking neurons, which holds biological evidences.

The first step for pattern recognition is to understand how the information is stored in a pattern. How the information is represented in the brain still remains unclear. However, there is a strong evidence showing that using pulses to encode, as a basic means of information transfer, is optimal in terms of information transmission [1]. Two basic and widely studied coding schemes using these pulses are rate coding and temporal coding. In rate coding the number of spikes within a time window is considered while the precise timings of each spike are considered in temporal coding [2]. Neurons, in the retina, the lateral geniculate nucleus (LGN) and the visual cortex as well as in many other sensory systems, are observed to precisely respond to the stimulus on a millisecond timescale [3]. Temporal patterns in the spike train can carry more information than rate-based coding [4], [5], [6]. A simple example of temporal encoding is spike latency code. The capability of encoding information in the timing of single spikes to compute and learn realistic data is demonstrated in [7]. Since this coding utilizes only a single spike to transfer information, it could potentially be beneficial for efficient pulse-stream very large scale integration (VLSI) implementations.

There are numerous ways that human brain has been modeled, but these models are far from reaching comparable performance. The spiking neurons dealing with precise timing spikes improve on the traditional neural models in both precision and accuracy. There are several kinds of spiking neuron models, like the integrate-and-fire (IF) model, the Hodgkin-Huxley-type model and the Izhikevich model. Compared to the other two models, the IF model is simple and computationally effective [8]. Using a biologically plausible model of supervised learning, the tempotron [9], an IF neuron can learn to categorize a broad range of input classes.

Although spiking neural networks (SNNs) show promising capability in playing a similar performance as living brains due to their more faithful similarity to biological neural networks, the big challenge of dealing with spiking neural networks is getting data into and out of them, which requires proper encoding and decoding methods [10]. Some existing SNNs for pattern recognition (as in [11], [12]) based on the rate coding. Different from these SNNs, we focus more on the temporal coding which could potentially carry the same information using less number of spikes than the rate coding. This could largely facilitate the computing speed.

In this paper, we build a whole system containing encoding, learning and readout. Inspired by the local receptive fields of biological neurons, we encode discrete input variables by grouping several input dimensions together. Through computational converting method proposed in this paper, the grouped input dimensions can be encoded to the timing of a single spike. The precise timing scale of the spike is on a millisecond level which is consistent with biological experimental observations. The readout part uses a simple binary presentation as proposed in this paper. In the binary string, the value 1 refers to the fired state of the output neuron and 0 refers to
the non-fired state. Through the encoding and readout, the spiking neural networks can be applied to deal with real data well.

The main contribution of this paper lies in the design of a whole system of spiking neural network, using a temporal coding scheme, for solving pattern recognition. The system contains encoding, learning and readout part. The spiking neural network using tempotron learning rule can classify complex patterns of spike trains. We show that neural networks based on the alternative and biologically more plausible paradigm are effective and efficient for pattern recognition. In addition, the results could show the single time coding is a viable means for neural information processing and learning on real-world data. Using this coding, it could potentially be beneficial for implementations of VLSI.

This paper is organized as follows. Section 2 introduces the temporal learning rule we used. The relationship between this rule and STDP is also introduced. Section 3 presents the architecture of the spiking neural network. Section 4 shows the ability of the network to learn different patterns of activities. More general cases for binary and non-binary patterns with variable coding level are included. Section 5 takes a further step to apply the network to learn real-world stimuli (images). Images of handwritten digits from the MNIST database and also the 26 alphabetic letters are used. Finally, we end up with some comments in the last section.

II. TEMPORAL LEARNING RULE

Temporal learning rule aims on dealing with information encoded by precise timing spikes. One of the most commonly studied rules is the spike-timing-dependent plasticity (STDP) which has emerged in recent years as experimentally most studied form of synaptic plasticity (see [13], [14], [15], [16], [17] for reviews). According to STDP learning rule, the plasticity depends on the intervals between pre- and postsynaptic spikes. The basic mechanisms of plasticity found in STDP are the long term potentiation (LTP) and the long term depression (LTD). However, STDP characterizes synaptic changes solely in terms of the temporal contiguity of the presynaptic spike and the postsynaptic potential or spike. In addition, to get the convergence of learning with STDP, a suitable balance of many parameters is needed [17]. In [9], the tempotron learning rule is presented. In this rule, the synaptic plasticity is governed by the temporal contiguity of a presynaptic spike and a postsynaptic depolarization, and a supervisory signal. The tempotron can make the appropriate decision under a supervisory signal by tuning fewer parameters than STDP. Moreover, the tempotron rule also uses the mechanisms of LTP and LTD to fulfill the synaptic plasticity as in STDP.

The neuron model used here is called tempotron. It is a leaky integrate-and-fire (LIF) neuron driven by exponential decaying synaptic currents generated by its synaptic afferents [9]. The potential of the neuron is a weighted sum of postsynaptic potentials (PSPs) from all incoming spikes:

$$V(t) = \sum_i w_i \sum_{t_i} K(t - t_i) + V_{rest}$$

Here \(w_i\) and \(t_i\) are the synaptic efficacy and the firing time of the \(i^{th}\) afferent. \(V_{rest}\) is the rest potential of the neuron. \(K\) denotes the normalized PSP kernel:

$$K(t - t_i) = V_0(\exp(-\frac{(t - t_i)}{\tau_m}) - \exp(-\frac{(t - t_i)}{\tau_s}))$$

Here \(\tau_m\) and \(\tau_s\) denote decay time constants of membrane integration and synaptic currents. \(V_0\) normalizes PSP so that the maximum value of the kernel is 1. \(K(t - t_i)\) is a causal filter that only considers spikes \(t_i \leq t\). The kernel function is showed in Fig. 1.

A neuron is fired when \(V(t)\) crosses the firing threshold, after which the potential smoothly decreases to \(V_{rest}\) by shutting down all the following spikes from the input afferents; the spikes after firing time do not have effect on the postsynaptic voltage. Fig. 1 shows the dynamic response of the neuron. The blue line indicates a neuron has fired; the green one indicates a neuron has not fired. In this neuron model, the potential boundaries at threshold and rest potential are ignored.

![Dynamic tempotron response](image)

Fig. 1. Dynamic tempotron response. Left top: examples of spiking patterns. There are two patterns (blue and green) and each spike from an input afferent is denoted by a dot. The Y axis is input identification number. Left bottom: neural potential traces. Each colour of lines corresponds to the same colour patterns on left top. In this neuron model, the potential boundaries at threshold and rest potential are ignored. Right: normalized PSP kernel.

In the classification task, each input pattern presented to the neuron belongs to one of two classes (which are labeled by \(P^+\) and \(P^-\)). One neuron can discriminate these patterns by firing or not. When a \(P^+\) pattern is presented to the neuron, it should elicit a spike; when a \(P^-\) pattern is presented to the neuron, it should keep silent by not firing. The tempotron learns patterns by changing the synaptic efficacies \((w_i)\) whenever there is an error. If the neuron fails to fire in response to a pattern \((P^+)\), each synaptic efficacy is increased as the following:

$$\Delta w_i = \lambda \sum_{t_i < t_{max}} K(t_{max} - t_i)$$

Here \(t_{max}\) denotes the time at which the neuron reaches its maximum potential value in the time domain. \(\lambda > 0\) is a constant representing the learning rate. It denotes the maximum change on the synaptic efficacies. If the neuron makes a false fire in response to a pattern \((P^-)\), each synaptic efficacy is decreased the same way as shown in (3).
III. THE SPIKING NEURAL NETWORK

In this section, we describe the whole system architecture for pattern recognition. The system consists of 3 functional parts: the encoding part, the learning part and the readout part. See Fig. 2 for the complete architecture. A stimulus consists of several components. The components are partially connected to the encoding neurons to generate encoded spiking information. The encoding neurons are fully connected to the learning neurons.

Each part performs different functional role in the system: The encoding one generates the set of specific activity patterns that represent the various attributes of external stimuli; the learning one tunes the neurons’ weights making sure particular neurons can respond to certain patterns correctly; the readout part extracts information about the stimulus from a given neural response. Through this architecture, the problem of getting data into and out of the spiking neural network is solved, and the task of pattern recognition could be fulfilled.

A. Encoding

The aim of the encoding part is to generate spiking patterns that represent the input stimuli. The temporal encoding is used over rate-based one when patterns within the encoding window[2] provide information about the stimulus that cannot be obtained from the spike count. The latency code [2] is a simple example of temporal encoding. It encodes information in the timing of response relative to the encoding window, which is usually defined with respect to stimulus onset. The single spike latencies are used to encode stimulus information in our system. Within the encoding window, each input neuron fires only once.

The encoding neuron has M input points (Fig. 2) which are selected from the components of the stimulus. It performs a specific function to convert the input points into latencies within the encoding window. For example, if the stimulus is composed of binary values (0 or 1), the function of the encoding neuron is to convert the binary string into a decimal value. The encoding time window is chosen to be hundreds of milliseconds.

B. Learning

The learning part of network is composed of one layer of tempotron. The encoding neurons are fully connected to the learning neurons. The number of encoding neurons (N_e) is determined by the number of patterns (N_p). The number of synapses to a learning neuron is equal to N_e, according to the structure. The tempotron can perform the classification task as long as the load is less than a critical value, approximately 3 [9]. That is to say, the maximum number of random generated patterns that a tempotron can learn is roughly 3 times the number of synapses connected to it. Therefore, as long as the number of patterns does not exceed the critical load value, the network can perform the task well. If there are too many patterns, the number of encoding neurons should be increased correspondingly. Through simulation, we found that nearly a hundred of encoding neurons are sufficient to fulfill the task considered in this paper.

The tempotron comprises the learning neurons. The learning neuron fires or not when it is presented to a stimulus and then the synaptic efficacies are updated according to the learning rule.

C. Readout

The aim of the readout part is to extract information about the stimulus from the response of learning neurons. In this part, we use a binary sequence to represent a certain class of patterns for the reason that each learning neuron can only discriminate two groups. Each learning neuron responds to a stimulus by firing (1) or not firing (0). So, the total N learning neurons as the output can represent a maximum number of $2^N$ classes of patterns. The number of learning neurons is determined by the number of classes in the recognition task. For example, four readout is sufficient for a group of patterns containing 16 classes.

IV. LEARNING PATTERNS OF NEURAL ACTIVITIES

Many ways of encoding memory patterns in neural networks have been studied. The memory patterns encoded in synaptic weights can be taken to be binary vectors, as well as they can also be taken to be drawn from a distribution with several discrete activity values or from a continuous distribution [18]. In Hopfield network [19], the memory patterns are expressed through the activities of neurons, where the states of neurons have binary values (+1 for active neuron and -1 for inactive one). In some other networks, the non-binary coding schemes [20] are also introduced.

In the previous section, the ability of tempotron to separate temporal patterns is introduced. It can classify a number of spatiotemporal patterns due to the learning rule. However, the following questions arise: Can this method be used to recognize memory patterns mentioned above in this section? If so, how can it perform the task?

The patterns are n-dimensional vectors and the value of each element in the vector refers to the neuron activity which can be drawn from several discrete values. The coding schemes used here are same as the ones in Treves and Rolls [21]. The activity $\eta$ of each neuron follows the probability distribution function $p(\eta)$:

$$p(\eta) = \begin{cases} (1 - c)\delta(\eta - \eta_0) + c\delta(\eta - \eta_1), & \text{(binary)} \\ (1 - \frac{2}{3})\delta(\eta - \eta_0) + \frac{1}{3}\delta(\eta - \eta_1) + \frac{1}{3}\delta(\eta - \eta_2), & \text{(ternary)} \\ (1 - 2c)\delta(\eta) + 4ce^{-2\eta}, & \text{(exponential)} \end{cases}$$

Where $\delta(x)$ is the Dirac’s function: $\delta(x) = \begin{cases} 1 & \text{if } x = 0 \\ 0 & \text{otherwise} \end{cases}$ and $c$ is the coding level which is defined as the mean level of activity of the network [18], [21].

As explorations for the ability of tempotron to classify different patterns of activities, we use binary and ternary patterns as the stimulus. The ternary patterns represent a simple non-binary structure. We also use variable coding levels.
to see the performance. The pattern vectors are generated according to (4). The activity values we choose for binary patterns are $\eta_0 = 0$ and $\eta_1 = 1$, and for ternary patterns are $\eta_0 = 0$, $\eta_1 = 1$ and $\eta_2 = 2$. The examples of binary and ternary patterns are shown in Fig. 3.

The pattern is stored in an n-dimensional vector with discrete values of activity. We use the system architecture for tempotron to classify pattern vectors (see Fig. 2). The layer of encoding neurons performs a function that convert the pattern vector into temporal pattern for tempotron to classify. We only use one learning neuron to test the ability of tempotron learning two groups of patterns.

To test the performance, we generate 100 memory patterns with 1024 elements, and assign half of patterns to a same group and others to another group. We also use different coding levels ($c = 0.2$ and $c = 0.5$) in our simulation. We set the number of input points of the encoding neuron for binary patterns to 8, and 5 for ternary patterns. Each element of pattern vector is connected to only one encoding neuron and the connections between pattern vector and encoding neurons are in order. For example, in binary patterns, the first 8 elements connect to the first encoding neuron and the second 8 elements connect to the second and the last 8 connect to the last encoding neuron.

From Fig. 4, we can see the tempotron also can learn different patterns of activities.

V. LEARNING REAL-WORLD STIMULI

From the previous section, we show the tempotron has the ability to learn different patterns of activities. It can separate different vector patterns successfully. To move forward a step, we apply the tempotron to learn some real-world stimuli (images).

In this section, we show the system architecture to perform visual pattern recognition. The image is composed of $n$ pixels. The pixels are partially connected to the encoding neurons to generate encoded spiking information.

We implement the learning procedure on two kinds of data sets: the handwritten digits from MNIST database and the 26 alphabetic letters. Both of the data sets are composed of binary images. We use these sets to test the performance of our network on real-world stimuli.
A. The Data Sets and The Classification Problem

The stimuli from real world typically have a complex statistical structure. It is quite different from idealized case of random patterns often considered. In the real world, the stimuli hold large variability in a given class and have a high level of correlation between members of different classes. There are two data sets we consider here: the MNIST digits and the alphabetic letters (see Fig. 5).

Fig. 5. Examples of data sets. Top: alphabetic letters. Bottom: handwritten digits from MNIST database

The MNIST set consists of ten classes (digit 0 to 9) and each example is on a grid of 28 × 28 pixels. (The MNIST set is available from http://yann.lecun.com/exdb/mnist on which many classification results from different methods are listed). To get the input data from the example, each image with grayscale values is converted to a binary vector. Each element out of the 784-element vector of an image is set to 1 if its grayscale value is above 128, otherwise set to 0. The MNIST data set provides a good benchmark for our network performance.

The great number of images in the MNIST set are written by different people. As the real-world stimuli, the images are variable in the same class and highly correlated with images in different classes. On one side, images from the same class differ from each other more or less. And on the other side, images from different classes share some similarities with each other. These properties determine the real-world stimuli are quite difficult for learning compared with randomly generated patterns in theoretical studies.

The alphabetic letters set is the second data set we consider. There are 26 classes (from A to Z), with one example in a class. This set is a binary representation of 26 alphabetic letters. Each letter is on a grid of 16 × 16 pixels. Although the set is small, there exist large overlaps between different classes (like “C”, “O” and “Q”). This data set is convenient for investigating the performance of the network on real-world pattern recognition.

B. Encoding

The encoding focuses on converting images into another representations in the form of spikes. A good encoding method should remain adequate information of the original images and facilitate the later learning process.

The input images are 2-dimensional binary pictures. They compose of a number of pixels. Each pixel is either value 0 or 1. The output values of the encoding part are encoded by pulses in the encoding window. The pixels to the encoding neurons are partially connected. Fig. 2 shows the encoding neuron model. It acts as a mapping function that converts a binary string to a decimal value. The number of input points defines the time domain of the encoding window. For example, if there are \( N \) input points, the encoding window length is \( 2^N \) ms.

Fig. 6. A schematic of encoding. Left: successive encoding; Right: random encoding.

The encoding neuron has \( M \) input points (Fig. 2) which are selected from the pixels pool. There are two kinds of selecting methods that we considered: the successive one and the random one (see Fig. 6). The successive encoding chooses input pixels from an image in a row-wise manner. The first \( M \) pixels for the first encoding neuron, and the next \( M \) pixels for the second encoding neuron, and so on. The random encoding chooses input pixels for each encoding neuron in a random manner. After selection, the input points of an encoding neuron combine a binary string which could be used to calculate the output pulse time by converting it to a decimal value. The spiking time is on the millisecond time scale. So, the encoding neurons convert the images to latency patterns in which each neuron only fire once in the time domain.

To simply compare these two encoding ways, we perform the recognition task on the alphabetic data set. We set the number of input points for each encoding neuron to 8. Thus, the encoding window length of the pattern is 256 ms. Fig. 7 illustrates the encoding results of the letter “A”. Each dot and cross denote a spike from the \( i^{th} \) afferent at time \( t \). Through simulation for recognition, we find that the random selection method is better than the other one both in convergent speed and correct rate of classification under the same condition (see Fig. 7). This is because many spikes fire at the same moment in the successive encoding, while in random encoding the spikes distribute randomly in the time domain. The distributed spikes
make full use of the whole encoding window, which results in a better recognition of the patterns.

C. Recognition Results

If there are too many patterns that exceed the load of the system, the number of encoding neurons should be increased by randomly selecting 8 pixels from the image for a new encoding neuron.

To see the performance ability of the network on the recognition task, we use a small data set from the MNIST. After several iterations of training, the network can recognize all the patterns in this data set. Here, we take the recognition results of several digits as an example (Fig. 8). If the potential of the learning neuron crosses the threshold, which is said to fire, the value of this neuron is considered as 1, otherwise it is 0. According to Fig. 8, when image “1” shows up to the network, only neuron 4 fires, so the result is \([0001]_{\text{bin}}\). For image “5”, the result is \([0101]_{\text{bin}}\), and for “8” it’s \([1000]_{\text{bin}}\). This shows the spiking neural network we used could successfully perform the recognition task.

VI. Conclusion

A simple network of spiking neurons to perform pattern recognition using temporal coding has been presented. Since the tempotron is believed to be inherited from the biological neural systems, the biological plausibility of the approach is a main aspect in this study considering machine learning as a main target. The whole system for pattern recognition task goes through encoding, learning and readout. It can perform the recognition task with a number of patterns. The encoding window of the temporal patterns is on a scale of hundreds of milliseconds, which matches the biological evidence [3], [5], [2]. The encoding scheme used in this paper is inspired from receptive fields of biological neurons. Each neuron receives a partial information from the whole external stimuli. In addition, the random encoding plays an important role on classification. Since the real-world stimuli are variable in the same class and highly correlated with patterns in different classes, random encoding focuses on finding the differences by making full use of the encoding window. The encoding schemes used in this paper might be biologically implausible but they provide a practical and effective way helping spiking neurons to read images. Such practical ways might be useful for real-world applications.

Acknowledgment

This work was supported by Agency for Science, Technology, and Research (A*STAR), Singapore under SERC Grant 092 157 0130.

References


