

Chapter 2

Rapid Feedforward Computation by Temporal Encoding and Learning with Spiking Neurons

Abstract As we know, primates perform remarkably well in cognitive tasks such as pattern recognition. Motivated from recent findings in biological systems, a unified and consistent feedforward system network with a proper encoding scheme and supervised temporal rules is built for processing real-world stimuli. The temporal rules are used for processing the spatiotemporal patterns. To utilize these rules on images or sounds, a proper encoding method and a unified computational model with consistent and efficient learning rule are required. Through encoding, external stimuli are converted into sparse representations which also have properties of invariance. These temporal patterns are then learned through biologically derived algorithms in the learning layer, followed by the final decision presented through the readout layer. The performance of the model is also analyzed and discussed. This chapter presents a general structure of SNN for pattern recognition, showing that the SNN has the ability to learn the real-world stimuli.

2.1 Introduction

Primates are remarkably good at cognitive skills such as pattern recognition. Despite decades of engineering effort, the performance of the biological visual system still outperforms the best computer vision systems. Pattern recognition is a general task that assigns an output value to a given input pattern. It is an information-reduction process which aims to classify patterns based on a priori knowledge or statistical information extracted from the patterns. Typical applications of pattern recognition includes automatic speech recognition, handwritten postal codes recognition, face recognition and gesture recognition. There are several conventional methods to implement pattern recognition, such as maximum entropy classifier, Naive Bayes classifier, decision trees, support vector machines (SVM) and perceptrons. We refer these methods as traditional rules since they are less biologically plausible compared to spiking time involved rules described later. Compared to human brain, these methods are far from reaching comparable recognition. Humans can easily discriminate different categories within a very short time. This motivates us to investigate computational models for rapid and robust pattern recognition from a biological point of

view. At the same time, inspired by biological findings, researchers have come up with different theories of encoding and learning. In order to bridge the gap between those independent studies, a unified systematic model with consistent rules is desired.

A simple feedforward architecture might account for rapid recognition as reported recently [1]. Anatomical back projections abundantly appear almost every area in the visual cortex, which puts the feedforward architecture under debate. However, the observation of a quick response appeared in inferotemporal cortex (IT) [2] most directly supports the hypothesis of the feedforward structure. The activity of neurons in monkey IT appears quite soon (around 100 ms) after stimulus onset [3]. For the purpose of rapid recognition, a core feedforward architecture might be a reasonable theory of visual computation.

How information is represented in the brain still remains unclear. However, there are strong reasons to believe that using pulses is the optimal way to encode the information for transmission [4]. Increasing number of observations show that neurons in the brain precisely response to a stimulus. This support the hypothesis of the temporal coding.

There are many temporal learning rules proposed for processing spatiotemporal patterns, including both supervised and unsupervised rules. As opposed to the unsupervised rule, a supervised one could potentially facilitate the learning speed with the help of an instructor signal. Although so far there is no strong experimental confirmation of the supervisory signal, an increasing body of evidence shows that this kind of learning is also exploited by the brain [5].

Learning schemes focusing on processing spatiotemporal spikes in a supervised manner have been widely studied. With proper encoding methods, these schemes could be applied to image categorization. In [6], the spike-driven synaptic plasticity mechanism is used to learn patterns encoded by mean firing rates. A rate coding is used to encode images for categorization. The learning process is supervised and stochastic, in which a teacher signal steers the output neuron to a desired firing rate. According to this algorithm, synaptic weights are modified upon the arrival of pre-synaptic spikes, considering the state of post-synaptic neuron's potential and its recent firing activity. One of the major limitations of this algorithm is that it could not be used to learn patterns presented in the form of precise timing spikes. Different from the spike-driven synaptic plasticity, the tempotron learning rule [7] is efficient to learn spike patterns in which information is embedded in precise timing spikes as well as in mean firing rates. This learning rule modifies the synaptic weights such that a trained neuron fires once for patterns of corresponding category and keeps silent for patterns of other categories. The ReSuMe learning rule [8, 9] is also a supervised rule in which the trained neuron can fire at desired times when corresponding spatiotemporal patterns are presented. It has been demonstrated that the tempotron rule and the ReSuMe rule are equivalent under certain conditions [10].

Although SNNs show promising capabilities in achieving a performance similar to living brains due to their more faithful similarity to biological neural networks, one of the main challenges of dealing with SNNs is getting data into and out of them, which requires proper encoding and decoding methods. The temporal learning algorithms are based on spatiotemporal spike patterns. However, the problem remains

how to represent real-world stimuli (like images) by spatiotemporal spikes for further computation in the spiking network. To deal with these problems, a unified systematic model, with consistent encoding, learning and readout, is required.

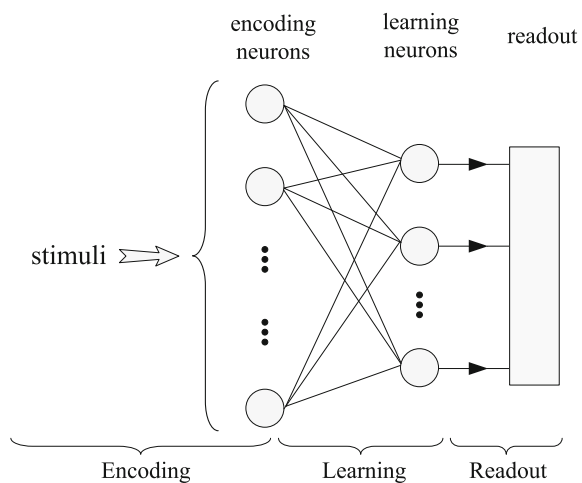
The main contribution of this chapter lies in the design of a unified system model of spiking neural network for solving pattern recognition problems. To the best of our knowledge, this is the first work in which complex classification task is solved through combination of biologically plausible encoding and supervised temporal learning. The system contains consistent encoding, learning and readout parts. Through the network, we fill the gap between real-world problem (image encoding) and theoretical studies of different learning algorithms for spatiotemporal patterns. Finally, our approach suggests a plausibility proof for a class of feedforward models of rapid and robust recognition in the brain.

2.2 The Spiking Neural Network

In this section, the feedforward computational model for pattern recognition is described. The model composes of 3 functional parts: the encoding part, the learning part and the readout part. Figure 2.1 shows the general architecture of the system model.

Considering the encoding, the latency code is a simple example of temporal coding. It encodes information in the timing of response relative to the encoding window, which is usually defined with respect to the stimulus onset. The external stimuli would trigger neurons to fire several spikes in different times. From biological observations, visual system can analyze a new complex scene in less than 150 ms [11, 12]. This period of time is impressive for information processing considering billions of

Fig. 2.1 A schematic of the feedforward computational model for pattern recognition. It contains three functional parts: encoding, learning and readout. A stimulus is converted into spatiotemporal spikes by the encoding neurons. This spiking pattern is passed to the next layer for learning. The final decision is represented by the readout layer



neurons involved. This suggests that neurons exchange only one or few spikes. In addition, it is shown that subsequent brain region may learn more and earlier about the stimuli from the time of first spike than from the firing rate [11].

Therefore, we use single spike code as the encoding mechanism. Within the encoding window, each input neuron fires only once. This code is simple and efficient, and the capability of encoding information in the timing of single spikes to compute and learn realistic data has been shown in [13]. Compared to rate coding as used in [6], this single spike coding would potentially facilitate computing speed since less spikes are involved in the computation.

Our single spike coding is similar to the rank order coding in [14, 15] but taking into consideration of the precise latency of the spikes. In the rank order coding, the rank order of neurons' activations is used to represent the information. This coding scheme is still under research. Taking the actual neurons' activations into consideration but not their rank orders, our proposed encoding method could convey more information than the rank order coding. Since this coding utilizes only a single spike to transmit information, it could also potentially be beneficial for efficient very large scale integration (VLSI) implementations.

In the learning layer, supervised rules are used since they could improve the learning speed with the help of the instructor signal. In this chapter, we investigate the tempotron rule and the ReSuMe rule.

The aim of the readout part is to extract information about the stimulus from the responses of learning neurons. As an example, we could use a binary sequence to represent a certain class of patterns in the case that each learning neuron can only discriminate two groups. Each learning neuron responds to a stimulus by firing (1) or not firing (0). Thus, the total N learning neurons as the output can represent a maximum number of 2^N classes of patterns.

A more suitable scheme for readout would be using population response. In this scheme, several groups are used and each group, containing several neurons, is one particular representation of the external stimuli. Different groups compete with each other by a voting scheme in which the group with the most amount of firing neurons would be the winner. This scheme is more compatible with the real brain since the information is presented by the cooperation of a group of neurons rather than one single neuron [16].

2.3 Single-Spike Temporal Coding

We have mentioned the function of the encoding layer is to convert stimulus into spatiotemporal spikes. In this section, we illustrate our encoding model of single-spike temporal coding, which is inspired from biological agents.

The retina is a particular interesting sensory area to study neural information processing, since its general structure and functional organization are remarkably well known. It is widely believed that information transmitted from retina to brain codes the intensity of the visual stimuli at every place in visual field. The ganglion

cells (GCs) collect the information from their receptive fields which could best drive spiking responses [17]. In addition, different ganglion cells might have overlapped centers of receptive fields [18]. A simple encoding model of retina is described in [14] and is used in [15]. The GCs are used as the first layer in our model to collect information from original stimuli.

Focusing on emulating the processing in visual cortex, a realistic model (HMAX) for recognition has been proposed in [19] and widely studied [1, 20, 21]. It is a hierarchical system that closely follows the organization of visual cortex. The HMAX performs remarkably well with natural images by using alternate simple cells (S) and complex cells (C). Simple cells (S) gain their selectivity from a linear sum operation, while complex cells (C) gain invariance through a nonlinear max pooling operation. Like the HMAX model, in order to obtain an invariant encoding model to some extent, a complex cells (CCs) layer is used in our model. In the brain, equivalents of CCs may be in V1 and V4 (see [22] for more details).

In our model (see Fig. 2.2), the image information (intensity) is transmitted to GCs through photo-receptors. Each GC linearly integrates at its soma the information from its receptive field. Their receptive fields are overlapping and their scales are generally distributed non-uniformly over the visual field. DoG (difference of gaussian) filters are used in the GCs layer since this filter is believed to mimic how neural processing in the retina of the eye extracts details from external stimuli [23, 24]. Several different scales of DoG would construct different GCs images. The CCs unit would oper-

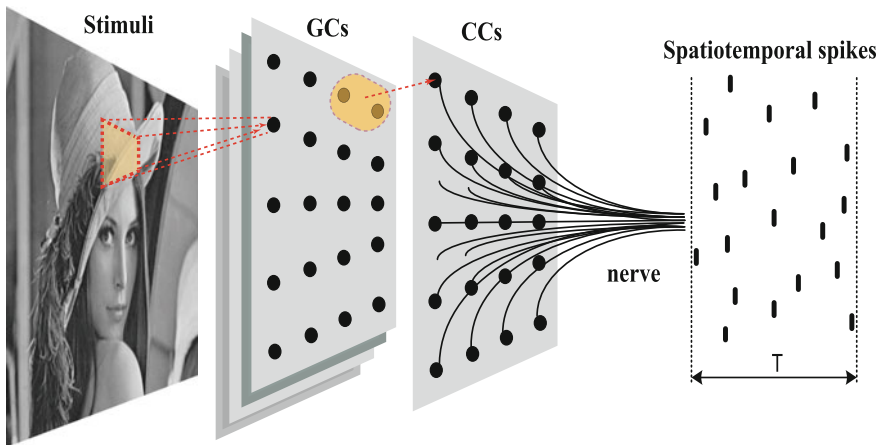


Fig. 2.2 Architecture of the visual encoding model. A gray-scale image (as the stimuli) is presented to the encoding layer. The photo-receptors transmit the image information analogically and linearly to the corresponding ganglion cells (GCs). Each ganglion cell collects information from its receptive field (an example shown as the *red dashed box*). There are several layers of GCs and each has a different scale of receptive field. The complex cells (CCs) collect information from a local position of GCs and a MAX operation among these GCs determines the activation value of CC unit. Each CC neuron would fire a spike according to their activations. These spikes are transmitted to the next layer as the spatiotemporal pattern in particular time window (T)

ate a nonlinear max pooling to obtain an amount of invariance. Max pooling over the two polarities, different scales and different local positions provides contrast reverse invariance, scale invariance and position invariance, respectively. Biophysically plausible implementations of the MAX operation have been proposed in [25], and biological evidences of neuron performing MAX-like behavior have been found in a subclass of complex cells in V1 [26] and cells in V4 [27].

The activation value of CC unit would trigger a firing spike. Strongly activated CCs will fire earlier, whereas weakly activated will fire later or not at all. The activation of the GC is computed through the dot product as:

$$GC_i := \langle I, \phi_i \rangle = \sum_{l \in R_i} I(l) \cdot \phi_i(l) \quad (2.1)$$

where $I(l)$ is the luminance of pixel l which is sensed by the photo-receptor. R_i is the receptive field region of neuron i . ϕ_i is the weight of the filter.

The GCs compute the local contrast intensities at different spatial scales and for two different polarities: ON- and OFF-center filters. We use the simple DoG as our filter where the surround has three times the width of the center. The DoG has the form as:

$$DoG_{\{s, l_c\}}(l) = G_{\sigma(s)}(l - l_c) - G_{3 \cdot \sigma(s)}(l - l_c) \quad (2.2)$$

$$G_{\sigma(s)}(l) = \frac{1}{2\pi \cdot \sigma(s)^2} \cdot \exp\left(-\frac{\|l\|^2}{2 \cdot \sigma(s)^2}\right) \quad (2.3)$$

where $G_{\sigma(s)}$ is the 2D Gaussian function with variance $\sigma(s)$ which depends on the scale s . l_c is the center position of the filter.

An example of the DoG filter is shown in Fig. 2.3. An OFF-center filter is simply an inverted version of an ON-center receptive field. All the filters are sum-normalized to zero and square-normalized to one so that when there is no contrast change in the image the neuron's activation would be zero and when the image is same with the filter the neuron's activation would be 1. Therefore, all the activations of the GCs are scaled to the same range $([-1, 1])$.

The CCs max over different polarities according to their absolute values at same scale and same position. Through this max operation, the model gain a contrast reverse invariance (Fig. 2.4a). From the property of the polar filters, only one could be positive activated for a given image. Similarly, the scale invariance is increased by max pooling over different scales at the same position (Fig. 2.4b). Finally, the position invariance is increased by pooling over different local positions (Fig. 2.4c). The dimension of images is reduced since only the max activated value in a local position is preserved.

Figure 2.5 shows the basic processing procedures in different encoding layers. Through the encoding, the original image is sparsely presented in the CCs (Fig. 2.5f).

The final activations of CCs are used to produce spikes. Strongly activated neurons would fire earlier, whereas weakly activated ones would fire later or not at all. The

Fig. 2.3 Linear filters in retina. **a** is an image of the ON-center DoG filter, whereas **(b)** is an image of the OFF-Center filter. **c** is the one-dimensional show of the DoG weights and **(d)** is the 2-dimensional show

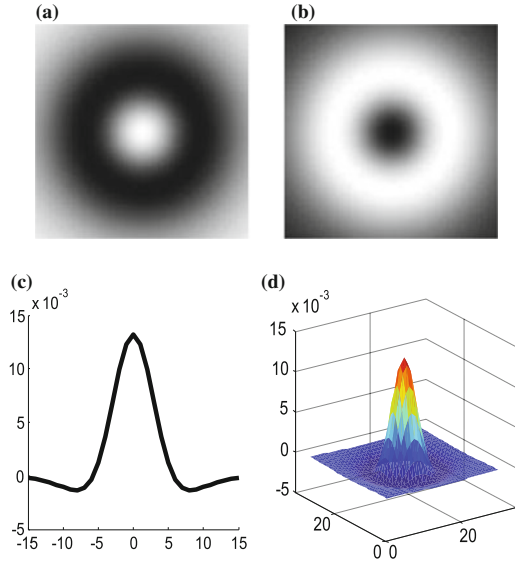
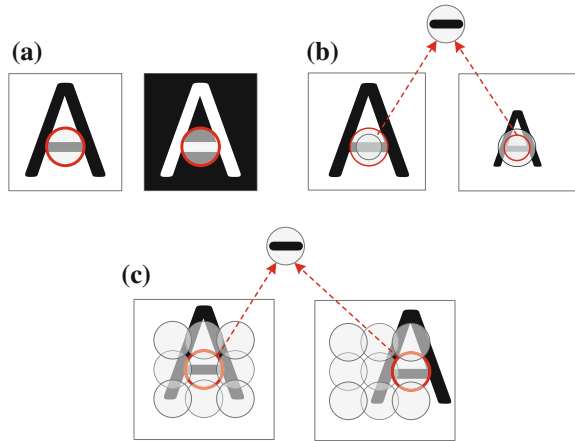


Fig. 2.4 Illustration of invariance gained from max pooling operation. **a** the contrast reverse invariance by max pooling over polarities. **b** the scale invariance by max pooling over different scales. **c** the local position invariance by max pooling over local positions. The *red circle* denotes the maximally activated one



spike latencies are then linearly mapped into a predefined encoding time window. These spatiotemporal spikes are transmitted to the next layer for computation.

In our encoding scheme, we consider the actual values of neurons' activations to generate spikes but not the rank order of these activations as used in [14, 15]. This could carry more information than the rank order coding which only considers the rank order of different activations and ignores the exact differences between different activations. For example, there are 3 neurons (n_1 , n_2 and n_3) having their activations (C_1 , C_2 and C_3) in the range of $[0, 1]$. Pattern P_1 is represented by ($C_1 = 0.1$, $C_2 = 0.3$ and $C_3 = 0.9$); pattern P_2 is represented by ($C_1 = 0.29$, $C_2 = 0.3$ and $C_3 = 0.32$). In rank order coding, it will treat P_1 and P_2 as same patterns since the

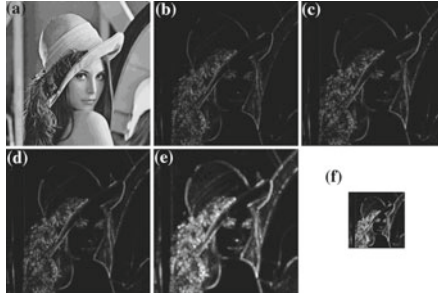


Fig. 2.5 Illustration of the processing results in different encoding procedures. **a** is the original external stimulus. **b** and **c** are the processing results in layer GCs with different scales. **d**, **e** and **f** are the processes in the CCs layer. **d** is the result of max pooling over different scales. **e** is max pooling over different local positions. **f** is the sub-sample from (**e**)

rank orders are same. For our encoding, in contrast, P_1 and P_2 would be treated as totally different patterns. In addition, the rank order coding would be very sensitive to the noise since the encoding time of one neuron depends on other neurons' rank. For example, if the least activated value is changed to a max activated value because of a disturbance, the rank of all the other neurons would be changed. However in our proposed algorithm only the information of the disturbed neuron would be affected.

2.4 Temporal Learning Rule

Temporal learning rule aims at dealing with information encoded by precise timing spikes. In this section, we consider supervised mechanisms like the tempotron rule and the ReSuMe rule that could be used for training neurons to discriminate between different spike patterns. Whether a LTP or LTD process occurs depends on the supervisory signal and the neuron's activity. This kind of supervisory signal can facilitate the learning speed compared to the unsupervised method.

2.4.1 The Tempotron Rule

In [7], the tempotron learning rule is proposed. According to 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 appropriate decisions under a supervisory signal by tuning fewer parameters.

In binary classification problem, each input pattern presented to the neuron belongs to one of two classes (which are labeled by P^+ and P^-). One neuron can make decision by firing or not. When a P^+ pattern is presented to the neuron, it

should elicit a spike; when a P^- pattern is presented, it should keep silent by not firing. The tempotron rule modifies the synaptic weights (w_i) whenever there is an error. This rule performs like gradient-descent rule that minimizes a cost function as:

$$C = \begin{cases} V_{thr} - V_{t_{max}}, & \text{if the presented pattern is } P^+; \\ V_{t_{max}} - V_{thr}, & \text{if the presented pattern is } P^-. \end{cases} \quad (2.4)$$

where $V_{t_{max}}$ is the maximal value of the post-synaptic potential V .

Applying the gradient descent method to minimize the cost leads to the tempotron learning rule:

$$\Delta w_i = \begin{cases} \lambda \sum_{t_i < t_{max}} K(t_{max} - t_i), & \text{if } P^+ \text{ error;} \\ -\lambda \sum_{t_i < t_{max}} K(t_{max} - t_i), & \text{if } P^- \text{ error;} \\ 0, & \text{otherwise.} \end{cases} \quad (2.5)$$

where 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. P^+ error denotes that the neuron should fire but it did not; P^- error denotes that the neuron should not fire but it did. According to the kernel shape, the efficacies of afferents that spike near to t_{max} change more than that are away from it. In this rule, t_{max} is the reference time for updating synaptic weights.

2.4.2 The ReSuMe Rule

The ReSuMe described in [9] is a supervised method that aims to produce desired spike trains in response to the given input sequence. According to this rule, the synaptic weights are modified according to the following equation:

$$\frac{d\omega_i(t)}{dt} = \lambda[S^d(t) - S^{out}(t)][a + \int_0^\infty W(s)S^i(t-s)ds] \quad (2.6)$$

where λ is the learning rate, a is a constant, W is a learning window with an exponential form ($W(s) = Ae^{-s/\tau_E}$). $S^d(t)$, $S^{out}(t)$ and $S^i(t)$ are the target, post- and pre-synaptic spike trains, respectively. Although the shape of learning window is not restricted to exponential form, this shape can result in a better performance of convergence [28]. The spike trains have the following form:

$$S(t) = \sum_{f=1}^n \delta(t - t^f) \quad (2.7)$$

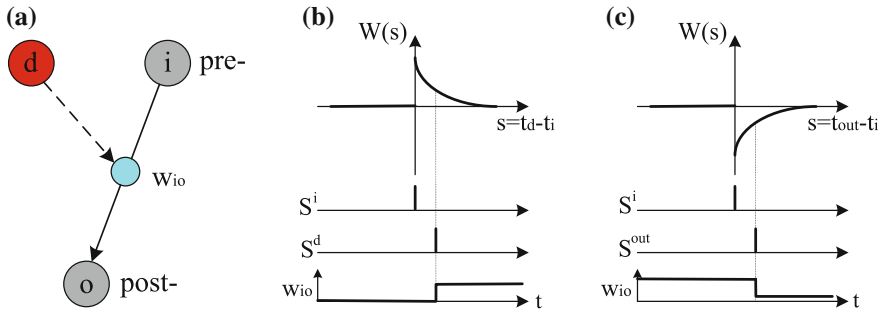


Fig. 2.6 Illustration of the ReSuMe learning rule. **a** demonstrates that the synaptic plasticity depends on the correlation between the pre- and postsynaptic firing times, and on the correlation between pre- and desired firing times. **b** demonstrates that the synaptic weight is potentiated whenever a desired spike is observed. **c** shows that the synaptic weight is depressed whenever the trained neuron fires. This figure is revised from [8]

where t^f denotes the moment of the f -th spike in the train, n denotes the total number of spikes in the train, $\delta(x)$ is the impulse function $\delta(x) = 1$ if $x = 0$ (or 0 otherwise).

Figure 2.6 illustrates the ReSuMe learning rule. The synaptic efficacy depends not only on the correlation between the pre-synaptic and post-synaptic firing times but also on the correlation between the pre-synaptic and desired firing times. A desired spike would result in synaptic potentiation, and a post-synaptic spike would result in synaptic depression.

After a learning trial, the total synaptic change is:

$$\Delta\omega_i = \lambda a(n^d - n^{out}) + \lambda \sum_{t^d} \sum_{t_i \leq t^d} W(t^d - t_i) - \lambda \sum_{t^{out}} \sum_{t_i \leq t^{out}} W(t^{out} - t_i) \quad (2.8)$$

where n^d and n^{out} are the number of spikes from the desired and the actual output spike trains respectively. t_i is the pre-synaptic spike time.

The ReSuMe rule could be used for both the batch learning and the online learning.

2.4.3 The Tempotron-Like ReSuMe Rule

As proposed in [10], the tempotron learning rule is a particular case of ReSuMe rule under certain conditions. The rule discussed here is a connection between the tempotron rule and the ReSuMe rule.

Considering to apply ReSuMe to the tempotron setup, the combined rule can be approached. The neuron is only allowed to fire once or not. After a spike is emitted, the neuron shunts all its incoming spikes immediately. If there is only one spike,

regardless of its time, it is reasonable to consider the neuron firing at t_{max} . This learning rule follows [10]:

$$\Delta w_i = \begin{cases} \lambda a + \lambda \sum_{t_i \leq t_{max}} W(t_{max} - t_i), & \text{if } n^d = 1, n^{out} = 0; \\ -\lambda a - \lambda \sum_{t_i < t_{out}} W(t_{out} - t_i), & \text{if } n^d = 0, n^{out} = 1; \\ 0, & \text{if } n^d = n^{out}. \end{cases} \quad (2.9)$$

When $a = 0$ and $W(s) = K(s)$, the combined rule is equivalent to the tempotron learning rule. This implicates that the tempotron rule is a particular case of the ReSuMe rule.

2.5 Simulation Results

In this section, several simulations are performed to test the performance of the network and different learning rules.

2.5.1 The Data Set 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. The data set we considered here is the MNIST digits (see Fig. 2.7).

The MNIST data set contains a large number of examples of hand-written digits, which consists of ten classes (digits 0 to 9) of examples and each example is an image of 28×28 pixels. The MNIST data set is available from <http://yann.lecun.com/exdb/mnist>, where many classification results from different methods are also listed. All images from this data set are gray-scale.

Fig. 2.7 Examples of handwritten digits from MNIST dataset



2.5.2 Encoding Images

Each image is presented to the encoding layer, and is then converted into spatiotemporal pattern. We use the coding strategy discussed previously through which the output is sparse, as is observed in biological agents [29].

For simplicity of applying the encoding algorithm to the data set, we distribute GCs with different receptive fields all over the image (each pixel). The image size in GCs is same as the input image. Considering examples of 28-by-28 images, we choose two scales for the filters ($\sigma = 1$ for 5×5 pixels as scale 1, and $\sigma = 2$ for 7×7 pixels as scale 2). The CCs layer performs the max pooling operation on the previous GCs layer. For local position operation we choose 6×6 pixels and we set the overlap pixels to be 3 in one axis (x or y) for sub-sampling operation. A detailed process of max operation is described in [19].

The application of all these processes produces a set of analog values, corresponding to the activation levels of our CCs unit. The strongly activated cell will fire earlier, whereas the weakly activated will fire later or not at all. The spike latencies are linearly mapped into a predefined encoding time window (100 ms in this study). The activation values are linearly converted to delay times, associating $t = 0$ with activation value 1 and later times up to 100 ms with lower activation values. The neurons with activation value of 0 (or below a chosen small value) will not fire due to the weak activation.

An illustration of encoding an image is shown in Fig. 2.5. Our scheme is to extract the basic information and encode it to a spatiotemporal spike pattern. Through the whole encoding structure, a sparse representation of the original incoming image is finally obtained. Using this sparse representation to generate the spike pattern would, to some extent, be compatible with biological observations in retina.

2.5.3 Choosing Among Temporal Learning Rules

In the tempotron rule, we specify the following parameters. The ratio between the membrane and the synaptic constants is fixed at $\tau_m/\tau_s = 4$. The threshold V_{thr} is set to 1 and V_{rest} is set to 0. We use $\tau_m = 10$ ms and $\lambda = 0.002$.

For comparison purpose, in the ReSuMe we use the similar neuron model as the one in the tempotron rule. However, the difference is that when the neuron emits a spike, its potential is reset to a rest value (0 here) and is hold there for a refractory period (3 ms here).

Since the ReSuMe rule is based only on the spiking times, it could work independently on the used spiking neuron models [9]. To verify the suitability of this rule for our chosen neuron model, we generate a spike pattern and force the neuron to respond at desired times. We choose 300 afferent synapses and each fires only once in the time window. The timing of each spike is generated randomly with uniform

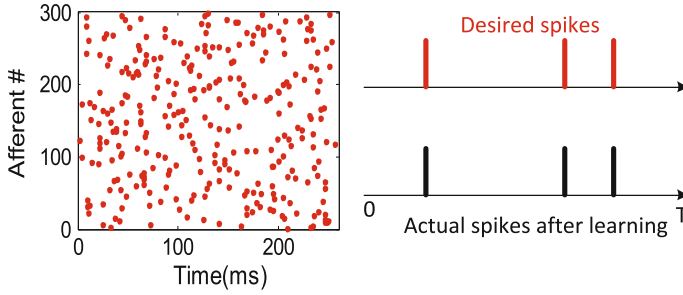


Fig. 2.8 Illustration of the suitability of ReSuMe rule for the chosen neuron model. The input pattern contains 300 afferent synapses and each fires once only. These spikes are generated randomly with uniform distribution. For the desired spike, 3 random spiking times are chosen

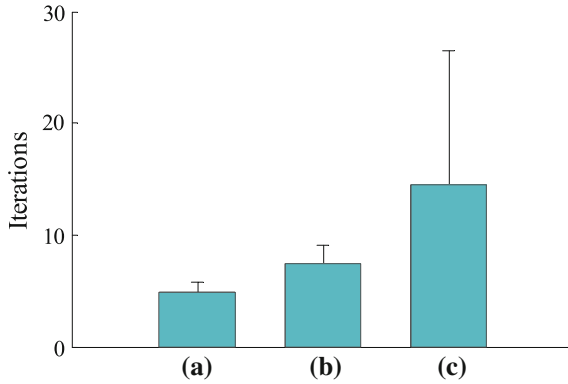


Fig. 2.9 The number of iterations needed for the correct classification of spike patterns, through different learning rules. **a** is the tempotron learning rule. **b** is the tempotron-like ReSuMe rule. **c** is the ReSuMe rule in which if the neuron fires, it should spike at a desired time. Over 100 experiments with different initial conditions, the averages (4.95, 7.36 and 14.48) and standard deviations (0.8454, 1.7438 and 12.014) are obtained for (a), (b) and (c), respectively

distribution between 0 and T . After learning, the neuron could perform as the desired way (see Fig. 2.8).

To compare the learning speed of different learning rules, we generate 30 spatiotemporal patterns and each pattern contains 120 afferent synapses. The spiking times are generated randomly with a uniform distribution between 0 and T . We randomly choose 3 patterns as one category that is needed to be discriminated from others. We record the minimum times of iterations for different rules to learn these patterns correctly. We perform this experiment for 100 times and the results are shown in Fig. 2.9.

According to Fig. 2.9, there is no significant difference of learning speed between the tempotron rule and tempotron-like ReSuMe rule. This is because the only difference between these two rules is the kernel windows which have a similar effect on

the synaptic change. However, compared to the ReSuMe rule, the tempotron rule is much faster (about 3 times as the ReSuMe rule). Besides this, the learning speed of the ReSuMe varies significantly for different initial conditions (such as the number of patterns, the initial weights and the learning rate). For the sake of fast recognition, we choose the tempotron rule as our learning rule.

2.5.4 The Properties of Tempotron Rule

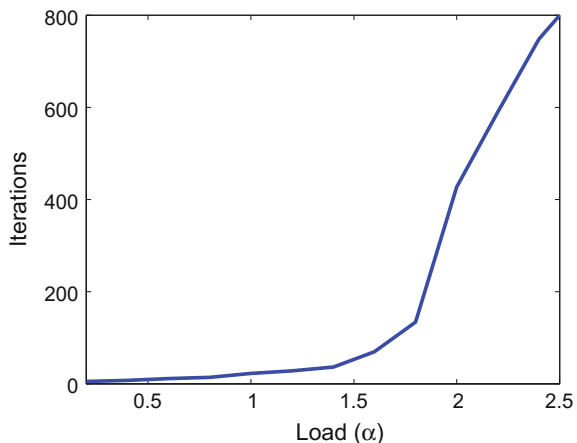
Since the tempotron rule is chosen, a test on its properties is needed.

2.5.4.1 Capacity

As is used for perceptron [30], the ratio of the number of random patterns (N_p) that correctly classified by the neuron over the number of its synapses (N_{in}), $\alpha = N_p/N_{in}$, is used to measure the load of the neuron. An important characteristic of neuron's capacity is the maximum load that it can learn. As studied in [7], the maximum recognition load of a tempotron can reach 3 approximately, which means that the number of patterns the neuron can learn could roughly approach to 3 times the number of synapses connected to it.

For our chosen neuron, a test on its load is shown in Fig. 2.10. We set $N_{in} = 100$ and generate different number of spike patterns within a fixed time window ($T = 100$ ms). Each afferent fires only once and the spiking time is randomly chosen from uniform distribution within T . The mean number of cycles of pattern presentations for error-free classification is shown versus the load (α). Although a more robust estimation of the load is feasible by allowing a small percentage of false alarms, the

Fig. 2.10 The mean number of iterations of pattern presentations for error-free classification versus neuron load. The patterns are randomly generated within the fixed time window (100 ms). The number of synapses is 100. Data are averaged over 20 runs



rigorous condition of error-free classification is useful to testify the neuron’s ability of classifying all assigned patterns successfully.

According to Fig. 2.10, the neuron could successfully learn the patterns within several tens of iterations if the load is not very high (below 1.5), but the number of iterations would increase sharply when the load is over 1.5. This means that under a higher load the neuron needs more time to learn the patterns or the learning process could never converge.

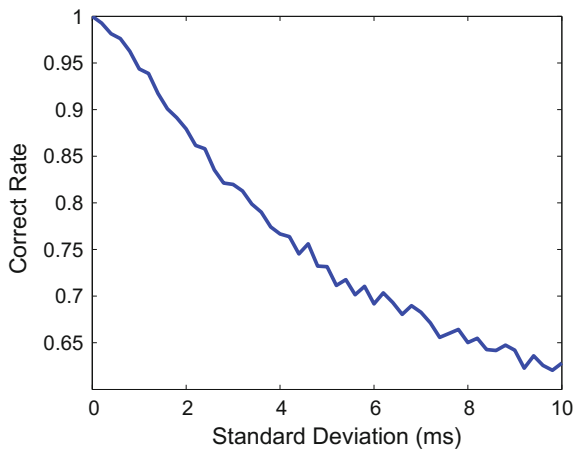
This load test, to some extent, could guarantee the learning convergence when the tempotron neuron is applied to our chosen recognition task. In our task, there are only ten categories and patterns in each category share some common features. Compared to the randomly generated patterns, the neuron’s capacity might be sufficient to learn these real-world stimuli.

2.5.4.2 Robustness

In some cases, the external noise might change the encoded spike patterns more or less. The tempotron rule should hold some level of robustness to tolerate the noise. To assess the robustness of the learning rule, we trained the neuron with a number of patterns ($\alpha = 1$). Then we tested the performance of the neuron when facing with jittered versions of previous learned patterns. The jittered pattern was generated by adding a Gaussian noise to all spike times of a template pattern. The robust performance of the neuron is shown in Fig. 2.11.

According to Fig. 2.11, the performance of correct recognition decreases with increasing jitter. Within a limited jitter range (0–3 ms), the performance stays in a relatively high level (over 0.8). This indicates the learning rule is robust to the presence of temporal noise to some extent.

Fig. 2.11 The mean correct rate of classification on jittered spike patterns. The jittered pattern is generated by adding Gaussian noise with standard deviation to all spike times of a template pattern



2.5.5 Recognition Performance

The combined system is applied to recognize different patterns. To see the ability of our system network on the recognition task, we use a small data set from the MNIST (50 digits and 5 for each category). And we choose four neurons as the readout. We call this readout as the fully distributed scheme with no redundancy (each neuron codes for one bit). 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. 2.12). If the potential of the learning neuron crosses the threshold, namely it fires, the value of this neuron is considered as 1, otherwise it is 0. In Fig. 2.12, when image “0” shows up to the network, none of the learning neurons fire, so the result is [0000]. For image “3”, the result is [0011], and for “9” it’s [1001]. This indicates that the tempotron rule applied in our model could recognize different classes of images successfully.

However, using only four neurons as the readout in a binary format might be very sensitive to changes of input images, especially considering the real-world stimuli in which samples hold large variability in a given class and overlap with members in different classes. If only one neuron misclassified the incoming pattern while others correctly responded, the pattern was still wrongly classified. Researchers have found that neighboring neurons have similar response properties. Depending on this, neural groups are used for assembly computing [16].

Thus, we use several grouped pools as our readout. We firstly consider a distributed code with redundancy: 4 pools of 20 neurons each. Each pool codes for one binary

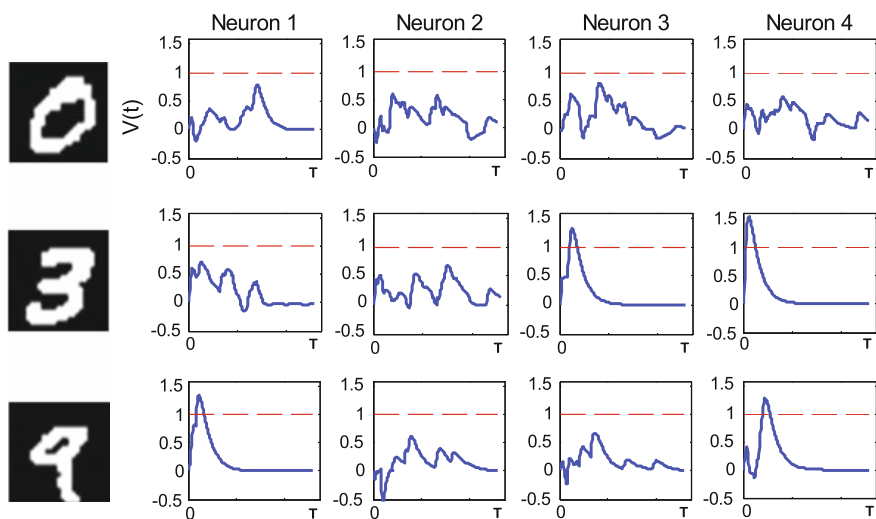


Fig. 2.12 Recognition results of digits by Tempotron learning rule. Here shows 4 learning neurons (Neuron 1–4) and 3 images. The neuron responds to an image by firing (1) or not (0). The results for “0”, “3” and “9” are [0000], [0011] and [1001], respectively

feature as in Fig. 2.12. A voting system decides if the binary feature is 0 or 1 based on the voting majority in this pool. Then we consider a localist scheme with redundancy, where each pool of 20 neurons codes for only one category. For an incoming stimulus, it is classified into a category according to the pool which has the most amount of voting neurons fired. If two or more pools have the same maximal firing number, the incoming stimulus is classified as unknown pattern.

These two schemes of readout with redundancy are used. For cross-validation, we choose 500 digits (50 images for each category) as our training set and randomly choose other 100 images from the MNIST data set as the testing set. In the training phase, each neuron is trained with a sub-training set chosen from the training set. This sub-training set consists of examples randomly chosen from the corresponding category and also other categories. After training, the performance is tested on both training set and testing set. The correct rate on the testing set is around 50% for the distributed code with no redundancy, and is around 79% for the localist code with redundancy. For distributed code, although the robustness for coding one bit feature is improved comparing to single neuron code, it is still not comparable to the localist one. This is due to that in the distributed code the final decision highly depends on correct reaction of each pool, but in the localist code it only depends on a correct major voting of one corresponding pool. Thus, in the localist scheme, the robustness is not only due to the redundancy but also to the localist aspect. This localist scheme is considered in our following experiments.

To make a comparison with the benchmark machine learning method, SVMs are chosen to perform the classification on the CCs activation values. Since SVM also has a binary decision behavior, we set the same classification condition on training and testing as for tempotron. The performances of both the tempotron and SVM on the training set and testing set are shown in Fig. 2.13. The corresponding recognition rates are shown in Table 2.1.

According to Fig. 2.13, our network with the tempotron rule performs at a high correct rate (around 93.7%) on the training set and at an acceptable correct rate (around 79%) on the testing set, especially considering the small data set (500 images) used for training. Comparing with SVM under the same condition of our encoding model, the performances of spiking neurons are better than SVM for the training set and comparable to SVM for the testing set. From a biological point of view, our system attempts to perform robust and rapid recognition with a brain-like architecture.

Table 2.1 The classification performance of tempotron and SVM on MNIST

Percentage(%)	Tempotron rule		SVM	
	Training	Testing	Training	Testing
Correct rate	93.67 \pm 0.67	78.5 \pm 1.85	90.24 \pm 0.98	79.33 \pm 2.03
Wrong rate	4.48 \pm 0.58	18.35 \pm 1.85	6.88 \pm 0.78	18.15 \pm 1.69
Unknown rate	1.86 \pm 0.61	3.15 \pm 1.64	2.89 \pm 0.86	2.53 \pm 2.04

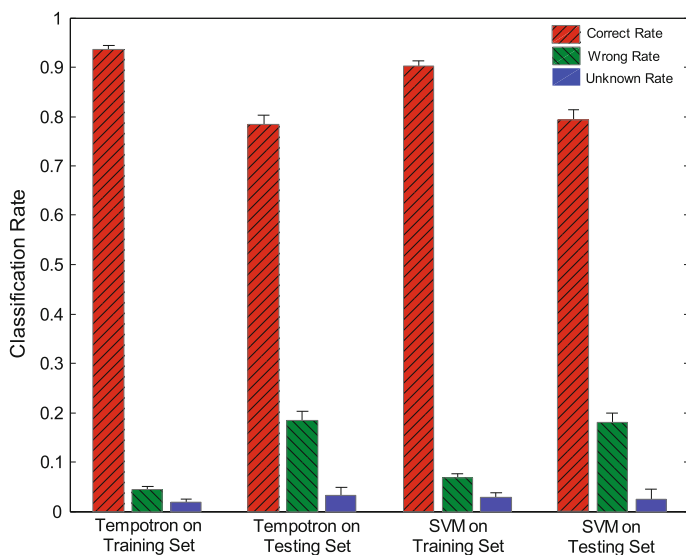


Fig. 2.13 The classification performance of tempotron and SVM. The system is trained 40 times each for tempotron and SVM. After each training time, the generalization is performed on both the training and testing set. The averages and standard deviations are plotted

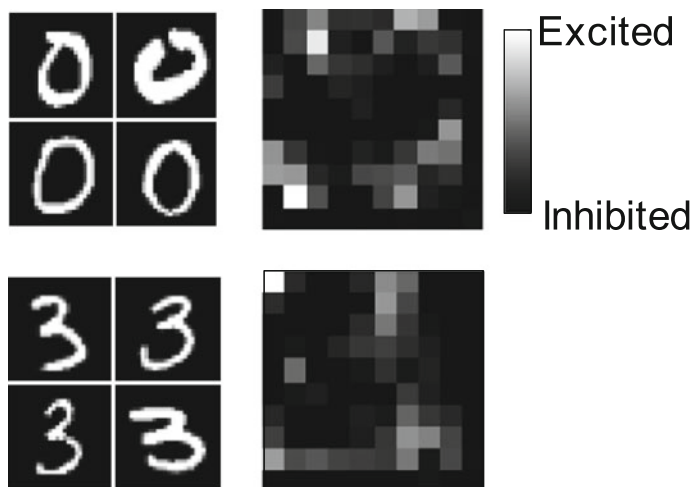


Fig. 2.14 Average weights of the spiking neurons in the pool representing digit 0 and 3. *Left* Image samples of digit 0 and 3 from MNIST are listed. *Right* The picture of the average weight of the spiking neurons in corresponding group. Inhibited afferents are plotted *black*, while excited ones are plotted *gray-scale* according to their weights

To investigate the states of the spiking neurons in one grouped pool after learning, a picture of the average weights is shown in Fig. 2.14. According to Fig. 2.14, the grouped neurons, cooperating together, roughly grab a general and basic feature of the learned category. Taking digit 0 as an example, the center weights are mostly inhibited since these neurons are rarely activated by the incoming stimulus 0 through our encoding model.

2.6 Discussion

Discussions on the proposed system are given as follows.

2.6.1 *Encoding Benefits from Biology*

Through the layers of GCs and CCs the external stimuli are sparsely represented in the activation values of CCs units. These activation values are used to generate spiking patterns in a time domain. It already has been shown that coding schemes based on the firing rates are unlikely to be efficient enough for fast information processing [14, 31]. Considering the rapid processing in the brain and billions of neurons involved, a temporal code which uses single spikes is, in principle, capable of carrying substantial information about the external stimuli [11] and facilitating the computational speed. In several sensory systems, shorter latencies of spikes result from stronger stimulation [32, 33]. In our encoding layer, the strongly activated neurons would fire earlier, whereas the weakly activated neurons would fire later or not at all. The chosen encoding window of the temporal patterns is on a scale of hundreds of milliseconds, which matches the biologically experimental results as mentioned in [34–36]. In addition, our encoding is efficient and the spiking output is sparse as observed in biological retinas [29, 37].

2.6.2 *Types of Synapses*

The types of synapses are determined by the signs of their efficacies, with positive values corresponding to excitatory synapses and negative values to inhibitory synapses. Although this model is far from biological realism, it is proved to be a useful computational approach [9]. In the neuron model, the sign of synapse could change by learning. The learning also works when the signs of synapses are not allowed to change, but the capacity is reduced. For a practical usage for multiple-class problem, changing sign is allowed in the neuron model. This can be realized by altering the balance between excitatory and inhibitory pathways [7].

2.6.3 Schemes of Readout

Using a binary version of readout, the network is shown to be capable to finish a simple recognition task on a small data set. However, this kind of readout would be very sensitive to each neuron's performance in the readout. If only one neuron misclassifies the pattern while others do a correct classification, the final readout would also be wrong since it depends on all the neurons in a binary form. Using grouped pools could effectively compensate this. In nervous systems such as visual cortical areas [38] and hippocampus [39], information is commonly expressed through populations or clusters of cells rather than through single cell [40]. This strategy is robust since damage to a single cell will not have a catastrophic effect on the whole population. Through learning, neurons in the same group try to find the common features discriminating that category, and through voting, the most active group would be chosen. Another meaningful aspect of the readout is that there is an unknown decision. Since some samples in one category are quite similar to other categories (for example the digit "5" in the second row of Fig. 2.7), it is reasonable to label them as unknown rather than wrong. A further processing could be done for these unknown samples.

2.6.4 Extension of the Network for Robust Sound Recognition

In addition to the recognition on images, we also proposed a SNN for recognizing sounds. The general structure remains the same, where functional parts of encoding, learning and readout are involved. The major difference of the two systems is the encoding part. With a proper encoding scheme for sounds, the SNN can perform the

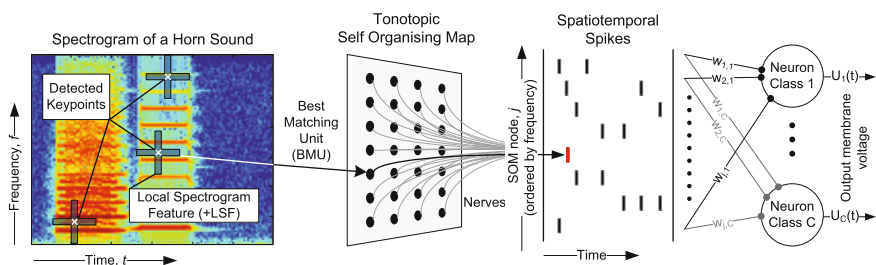


Fig. 2.15 The proposed LSF-SNN system for sound recognition. Firstly the keypoints are detected and the corresponding LSFs are extracted. Then, the SOM map is used to produce the output spatiotemporal spike patterns. These patterns are then learnt by the tempotrons for recognition

recognition well. We propose a novel approach based on the temporal coding of Local Spectrogram Features (LSF) [41], which generates spikes that are used to train the following neurons. The general structure for sound recognition is shown in Fig. 2.15. Our experiments demonstrate the robust performance of this system across a variety of noise conditions, such that it is able to outperform the conventional frame-based baseline methods. More details can be found in [41].

2.7 Conclusion

A systematic computational model by using consistent temporal encoding, learning and readout has been presented to explore brain-based computation especially in the regime of pattern recognition. It is a preliminary attempt to perform rapid and robust pattern recognition from a biological point of view. The schemes used in this model are efficient and biologically plausible. The external stimuli are sparsely represented after our encoding and the representations have properties of selectivity and invariance. Through the network, the temporal learning rules can be applied to processing real-world stimuli.

References

1. Serre, T., Oliva, A., Poggio, T.: A feedforward architecture accounts for rapid categorization. *Proc. Natl. Acad. Sci.* **104**(15), 6424–6429 (2007)
2. Perrett, D.I., Hietanen, J.K., Oram, M.W., Benson, P.J.: Organization and functions of cells responsive to faces in the temporal cortex. *Philos. Trans. R. Soc. Lond. Ser. B* **335**, 23–30 (1992)
3. Hung, C.P., Kreiman, G., Poggio, T., DiCarlo, J.J.: Fast readout of object identity from macaque inferior temporal cortex. *Science* **310**(5749), 863–866 (2005)
4. Tsukada, M., Pan, X.: The spatiotemporal learning rule and its efficiency in separating spatiotemporal patterns. *Biol. Cybern.* **92**, 139–146 (2005)
5. Knudsen, E.I.: Supervised learning in the brain. *J. Neurosci.* **14**(7), 3985–3997 (1994)
6. Brader, J.M., Senn, W., Fusi, S.: Learning real-world stimuli in a neural network with spike-driven synaptic dynamics. *Neural Comput.* **19**(11), 2881–2912 (2007)
7. Güttig, R., Sompolinsky, H.: The tempotron: a neuron that learns spike timing-based decisions. *Nature Neurosci.* **9**(3), 420–428 (2006)
8. Ponulak, F.: ReSuMe-new supervised learning method for spiking neural networks. Institute of Control and Information Engineering, Poznań University of Technology, Technical Report (2005)
9. Ponulak, F., Kasinski, A.: Supervised learning in spiking neural networks with resume: sequence learning, classification, and spike shifting. *Neural Comput.* **22**(2), 467–510 (2010)
10. Florian, R.V.: Tempotron-Like Learning with ReSuMe. In: *Proceedings of the 18th International Conference on Artificial Neural Networks. Part II, ICANN'08*, pp. 368–375. Springer, Heidelberg (2008)
11. Gollisch, T., Meister, M.: Rapid neural coding in the retina with relative spike latencies. *Science* **319**(5866), 1108–1111 (2008)
12. Thorpe, S., Fize, D., Marlot, C.: Speed of processing in the human visual system. *Nature* **381**(6582), 520–522 (1996)

13. Bohte, S.M., Bohte, E.M., Poutr, H.L., Kok, J.N.: Unsupervised clustering with spiking neurons by sparse temporal coding and multi-layer RBF networks. *IEEE Trans. Neural Netw.* **13**, 426–435 (2002)
14. Van Rullen, R., Thorpe, S.J.: Rate coding versus temporal order coding: what the retinal ganglion cells tell the visual cortex. *Neural Comput.* **13**(6), 1255–1283 (2001)
15. Perrinet, L., Samuelides, M., Thorpe, S.J.: Coding static natural images using spiking event times: do neurons cooperate? *IEEE Trans. Neural Netw.* **15**(5), 1164–1175 (2004)
16. Ranhel, J.: Neural assembly computing. *IEEE Trans. Neural Netw. Learn. Syst.* **23**(6), 916–927 (2012)
17. Hubel, D.H., Wiesel, T.N.: Receptive fields and functional architecture of monkey striate cortex. *J. physiol* **195**(1), 215–243 (1968)
18. Burkart, Fischer: Overlap of receptive field centers and representation of the visual field in the cat's optic tract. *Vis. Res.* **13**(11), 2113–2120 (1973)
19. Riesenhuber, M., Poggio, T.: Hierarchical models of object recognition in cortex. *Nature Neurosci.* **2**(11), 1019–1025 (1999)
20. Masquelier, T., Thorpe, S.J.: Unsupervised learning of visual features through spike timing dependent plasticity. *PLoS Comput. Biol.* **3**(2) (2007)
21. Serre, T., Wolf, L., Bileschi, S., Riesenhuber, M., Poggio, T.: Robust object recognition with cortex-like mechanisms. *IEEE Trans. Pattern Anal. Mach. Intell.* **29**, 411–426 (2007)
22. Serre, T., Kouh, M., Cadieu, C., Knoblich, U., Kreiman, G., Poggio, T.: A theory of object recognition: computations and circuits in the feedforward path of the ventral stream in primate visual cortex. In: *AI Memo* (2005)
23. Enroth-Cugell, C., Robson, J.G.: The contrast sensitivity of retinal ganglion cells of the cat. *J. Physiol.* **187**(3), 517–552 (1966)
24. McMahon, M.J., Packer, O.S., Dacey, D.M.: The classical receptive field surround of primate parasol ganglion cells is mediated primarily by a non-GABAergic pathway. *J. Neurosci.* **24**(15), 3736–3745 (2004)
25. Yu, A.J., Giese, M.A., Poggio, T.: Biophysically plausible implementations of the maximum operation. *Neural Comput.* **14**(12), 2857–2881 (2002)
26. Lampl, I., Ferster, D., Poggio, T., Riesenhuber, M.: Intracellular measurements of spatial integration and the MAX operation in complex cells of the cat primary visual cortex. *J. Neurophysiol.* **92**(5), 2704–2713 (2004)
27. Gawne, T.J., Martin, J.M.: Responses of primate visual cortical neurons to stimuli presented by flash, saccade, blink, and external darkening. *J. Neurophysiol.* **88**(5), 2178–2186 (2002)
28. Ponulak, F.: Analysis of the resume learning process for spiking neural networks. *Appl. Math. Comput. Sci.* **18**(2), 117–127 (2008)
29. Olshausen, B.A., Field, D.J.: Sparse coding with an overcomplete basis set: a strategy employed by V1? *Vis. Res.* **37**(23), 3311–3325 (1997)
30. Gardner, E.: The space of interactions in neural networks models. *J. Phys.* **A21**, 257–270 (1988)
31. Gautrais, J., Thorpe, S.: Rate coding versus temporal order coding: a theoretical approach. *Biosystems* **48**(1–3), 57–65 (1998)
32. Reich, D.S., Mechler, F., Victor, J.D.: Independent and redundant information in nearby cortical neurons. *Science* **294**, 2566–2568 (2001)
33. Greschner, M., Thiel, A., Kretzberg, J., Ammermüller, J.: Complex spike-event pattern of transient ON-OFF retinal ganglion cells. *J. Neurophysiol.* **96**(6), 2845–2856 (2006)
34. Panzeri, S., Brunel, N., Logothetis, N.K., Kayser, C.: Sensory neural codes using multiplexed temporal scales. *Trends Neurosci.* **33**(3), 111–120 (2010)
35. Butts, D.A., Weng, C., Jin, J., Yeh, C.I., Lesica, N.A., Alonso, J.M., Stanley, G.B.: Temporal precision in the neural code and the timescales of natural vision. *Nature* **449**(7158), 92–95 (2007)
36. Borst, A., Theunissen, F.E.: Information theory and neural coding. *Nature Neurosci.* **2**(11), 947–957 (1999)
37. Hunt, J.J., Ibbotson, M.R., Goodhill, G.J.: Sparse coding on the spot: spontaneous retinal waves suffice for orientation selectivity. *Neural Comput.* **24**(9), 2422–2433 (2012)

38. Usrey, W., Reid, R.: Synchronous activity in the visual system. *Annu. Rev. Physiol.* **61**(1), 435–456 (1999)
39. Wilson, M., McNaughton, B.: Dynamics of the hippocampal ensemble code for space. *Science* **261**(5124), 1055–1058 (1993)
40. Pouget, A., Dayan, P., Zemel, R.: Information processing with population codes. *Nature Rev. Neurosci.* **1**(2), 125–132 (2000)
41. Dennis, J., Yu, Q., Tang, H., Tran, H.D., Li, H.: Temporal coding of local spectrogram features for robust sound recognition. In: 2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 803–807 (2013)

Neuromorphic Cognitive Systems
A Learning and Memory Centered Approach
Yu, Q.; Tang, H.; Hu, J.; Tan, K.C.
2017, XIV, 172 p., Hardcover
ISBN: 978-3-319-55308-5