



UNIVERSITY OF LEEDS

This is a repository copy of *Dynamic Spatiotemporal Pattern Recognition with Recurrent Spiking Neural Network*.

White Rose Research Online URL for this paper:

<https://eprints.whiterose.ac.uk/177488/>

Version: Accepted Version

Article:

Shen, J, Liu, J orcid.org/0000-0002-5391-7213 and Wang, Y (2021) Dynamic Spatiotemporal Pattern Recognition with Recurrent Spiking Neural Network. *Neural Computation*. ISSN 0899-7667

https://doi.org/10.1162/neco_a_01432

© 2021 Massachusetts Institute of Technology. This is an author produced version of a paper published in *Neural Computation*. Uploaded in accordance with the publisher's self-archiving policy.

Reuse

Items deposited in White Rose Research Online are protected by copyright, with all rights reserved unless indicated otherwise. They may be downloaded and/or printed for private study, or other acts as permitted by national copyright laws. The publisher or other rights holders may allow further reproduction and re-use of the full text version. This is indicated by the licence information on the White Rose Research Online record for the item.

Takedown

If you consider content in White Rose Research Online to be in breach of UK law, please notify us by emailing eprints@whiterose.ac.uk including the URL of the record and the reason for the withdrawal request.



eprints@whiterose.ac.uk
<https://eprints.whiterose.ac.uk/>

**Dynamic Spatiotemporal Pattern Recognition with Recurrent Spiking Neural
Network**

Jiangrong Shen

jrshen@zju.edu.cn

College of Computer Science and Technology, Zhejiang University, Hangzhou, China;
Qiushi Academy for Advanced Studies, Zhejiang University, Hangzhou, China; Centre
for Systems Neuroscience, Department of Neuroscience, Psychology and Behaviour,
University of Leicester, Leicester, UK

Jian K. Liu¹

jian.liu@leicester.ac.uk

Centre for Systems Neuroscience, Department of Neuroscience, Psychology and Be-
haviour, University of Leicester, Leicester, UK

Yueming Wang¹

yumingwang@zju.edu.cn

College of Computer Science and Technology, Zhejiang University, Hangzhou, China;
Qiushi Academy for Advanced Studies, Zhejiang University, Hangzhou, China; State

¹Corresponding Authors

Key Lab of CAD&CG, Zhejiang University, Hangzhou, China; Zhejiang Lab, Hangzhou, China

Abstract Our real-time actions in everyday life reflect a range of spatiotemporal dynamic brain activity patterns, which are the consequence of neuronal computation with spikes in the brain. Most existing models with spiking neurons aim at solving static pattern recognition tasks such as image classification. Compared with static features, spatiotemporal patterns are more complex due to their dynamics in both space and time domains. Spatiotemporal pattern recognition based on learning algorithms with spiking neurons, therefore, remains challenging. Herein, we propose an end-to-end recurrent spiking neural network model trained with an algorithm based on spike latency and temporal difference backpropagation. Our model is a cascaded network with three layers of spiking neurons whereof the input and output layers are the encoder and decoder, respectively. In the hidden layer, the recurrently connected neurons with transmission delays carry out high-dimensional computation to incorporate the spatiotemporal dynamics of the inputs. The test results based on the datasets of spiking activities of the retinal neurons show that the proposed framework can recognize dynamic spatiotemporal patterns much better than using spike counts. Moreover, for 3D trajectories of human action dataset, the proposed framework achieves the test accuracy of 83.6% on average. Rapid recognition is achieved through the learning methodology based on spike latency and the decoding process using the first spike of the output neurons. Taken together, these results highlight a new model to extract information from activity patterns of neural computation in the brain and provide a novel approach for spike-based

neuromorphic computing.

Keywords: Spiking neurons, spatiotemporal dynamics, spatial-temporal patterns, spike latency, biological plausibility

1 Introduction

We conduct various cognitive behaviours in response to the dynamic environment and make a sequence of decisions every day (Golestani and Moghaddam, 2020). Neurons, the building blocks of the brain, play an important role in the powerful computations that give rise to these reactions. The most basic computation performed by individual or networked neurons is to generate sequences of input-output relations. Mechanistically, individual neurons receive the input stimuli and respond by adjusting their membrane potentials to generate a sequence of fast events, termed spikes (Rieke et al., 1996). Information is then transmitted by a group or network of neurons over time (Buonomano and Maass, 2009; Liu and Buonomano, 2009; Liu, 2011; Buzsáki, 2010; Yuste, 2015; Laje and Buonomano, 2013). Dynamic spatiotemporal patterns of neural spikes thus convey important messages of the brain activities (Buonomano and Maass, 2009). However, proper methods remain missing to effectively extract the rich information from such trajectories of spiking patterns (Quiroga and Panzeri, 2009; dos Santos et al., 2015; Panzeri et al., 2017).

In recent years, the development of machine learning (ML) — in particular, the artificial neural network (ANN) modeling — has made significant progress in analyzing temporal sequences (Lecun et al., 2015; Miranda et al., 2012; Devanne et al., 2013;

Miranda et al., 2014). Unlike most ANNs that employ artificial neurons communicating via continuous variables, the spiking neural networks (SNNs) transfer information through discrete spiking data. Hence, SNNs resemble the biological information processing system in the brain. Moreover, SNNs triggered by discrete inputs have event-driven computational advantages over general ANNs, allowing for the transmission of spatiotemporal data through spike-based encoding, learning rules, and memory mechanisms (Tsukada and Pan, 2005; Ghosh Dastidar and Adeli, 2009; Maass, 1997).

Many existing SNNs aim at solving the static pattern recognition tasks by incorporating additional temporal information into the original two-dimensional system and transforming the stimuli from static inputs, such as images, to dynamic spatiotemporal patterns (Diehl and Cook, 2015; Göltz et al., 2019; Yu et al., 2013; Xu et al., 2020). However, the performance of these SNNs is limited by their intrinsic shallow structures and basic classifying strategies. Hence, they are unable to potently process stimuli complexed with temporal data. In order to address this problem, recurrent SNNs (RSNNs) have been proposed with nodes connected by feedback loops (Soula et al., 2006; Pyle and Rosenbaum, 2017; Arena et al., 2012; Panda and Roy, 2017; Gilra and Gerstner, 2017). However, most of the current RSNNs are unreliable in computation due to the random connections in the recurrent layer (Pyle and Rosenbaum, 2017) in an overall superficial network structure.

Herein, we propose an RSNN model that can be trained end-to-end on GPU efficiently with a flexible network structure. Different from the conventional SNNs, the proposed framework is a multilayer model containing recurrent connections to take full advantage of spike latency. The system is capable of extracting useful features in large

quantities from complex inputs. Moreover, the learnable delays applied to the expandable time steps in the recurrent layers can balance the spiking time distribution of different input sources. Hence, they can enhance the classification capability of the system. Specifically, the dynamic input stimuli are encoded as spatiotemporal patterns, which are then transformed into spike trains by the designed encoding mechanism. The spiking neurons of the fully-connected layers and the recurrent layer could communicate the temporal information through the spiking trains. Such a framework and functional motifs are expected to improve the information processing ability of RSNN, representing a more cognitive neural system with biological plausibility than conventional ANNs.

We tested the RSNN system on dynamic datasets of spike patterns recorded from biological neurons and human motion trajectories to evaluate its learning capabilities under different dynamic input stimuli. The experimental results show that the proposed RSNN not only has a more efficient encoding scheme and more effective network architecture, but also exhibits enhanced ability in processing dynamic stimuli and performing complex classification than many other spike-based models.

2 Related Works

In recent years, ANNs showed upstaging advances in conducting various tasks (Lecun et al., 2015), particularly in classifying static images. However, when processing dynamic videos with additional time information, the large amount of data redundancy can corrupt or alter the integrality of individual image frames. Very few ANN studies operating on spatiotemporal patterns have been reported (Miranda et al., 2012; Devanne

et al., 2013; Miranda et al., 2014).

On the other hand, temporal patterns have been a focus of interest in recent years for SNN modeling (Diehl and Cook, 2015; Göltz et al., 2019; Yu et al., 2013). According to the network structure, the common-used SNNs can be categorized into the feedforward SNN and the recurrent SNNs.

For feedforward SNNs, different directly-trained and indirectly-trained SNN learning models sprang up for static spike pattern recognition. Firstly, the directly-trained SNNs consider more biologically plausible features of spike trains and adapt to the temporal coding mechanism more smoothly than indirectly-trained way. To improve the feature extraction ability of the SNN model, an augmented spike-based framework with perceptron-inception was introduced to combine the feature learning abilities of CNN and SNN (Xu et al., 2020) while using fewer neurons and training samples. Subsequently, a new approach controlling spiking networks with realistic temporal dynamics was proposed, exploring the end-to-end training technique with the gradient descent rule (Mostafa, 2018). With temporal coding, the input-output relations in a network became differential with a piecewise linear topology. Secondly, considering the limited performance of the directly-trained SNNs, the deep neural networks (DNNs) based converted training method is introduced to transfer the fully-trained DNNs into rate-based SNNs. In (Cao et al., 2015), the deep CNNs were converted into SNNs by first tailoring the CNNs structure to fit the requirements of SNNs then mapping the trained CNNs to the derived SNNs. Although these converted SNNs achieve competitive performance as CNNs, there exists information loss during converting process and makes inadequate use of spike characteristics. These feedforward SNNs recognize objects by

transforming the static input patterns to dynamic spatiotemporal ones. However, the patterns which recognized by these SNNs are limited by their feedforward structures. Hence, this kind of feedforward structures are good at recognizing static pattern without temporal information, while taking little advantage of the induced temporal information hidden in the spike trains.

Meanwhile, efforts have been made to improve the learning capability by resorting the recurrent connections. It is suggested that recurrent connections are important for object recognition in the biological visual system (Kar et al., 2019). To explore the variability of network dynamics, a parsimonious model with random recurrent connections of spiking neurons was firstly proposed, although being notoriously unreliable in computations (Soula et al., 2006). By incorporating the well-documented dependence of connection probability on distance, a new reservoir computing framework for SNNs was reported to perform dynamical computations (Pyle and Rosenbaum, 2017). In another recent work (Gilra and Gerstner, 2017), a network of heterogeneous spiking neurons with feedforward and recurrent connections was constructed and the corresponding supervised learning scheme was explored. During the training process, the output errors are fed back through fixed random connections with a negative gain, driving the network to improve in the desired direction. This model was evaluated on the learning of linear, non-linear, and chaotic dynamics, as well as those with a two-link arm. Yet, these recurrent SNN models failed to solve complex pattern recognition problems as a result of limited computation ability.

Except the random recurrent connection, the recurrent SNNs models with learnable recurrent connections are explored further. The first kind of the common-used

method for learning those recurrent connections is by virtue of the unsupervised learning rule, such as spike-timing-dependent plasticity (STDP). Combining the standard spike time correlation-based Hebbian plasticity and a reservoir network-based synaptic decay mechanism, a new SNN model was trained to generate sequences on datasets of word images (Panda and Roy, 2017), where the adaptive decay of synaptic weights introduced by Hebbian plasticity facilitates in learning stable contextual dependencies between temporal sequences, while reducing the strong attractor states that emerge in recurrent models due to feedback loops. However, the structure of this network cannot be scaled to multiple layers easily, limiting the learning ability of the model. Inspired by the insect olfactory system, a multilayer spiking network was proposed (Arena et al., 2012), where each layer is constructed according to the insect brain components mainly involved in olfactory information processing, namely, the Mushroom Bodies, the Lateral Horns, and the Antennal Lobes. The plastic recurrent connections are learned by the unsupervised simple learning mechanism. Thus, that structure is able to realize a top-down modulation at the input level, which leads to the emergence of an attentional loop as well as to the arousal of basic expectation behaviors in case of subsequently presented stimuli. The other kind of methods can learn the recurrent connections in the manner of supervised learning, which contain two typical ways to implement these learnable recurrent connections. The one is like (Wu et al., 2018), the recurrent behavior is implemented in the spiking neuron model, in which each spike neuron is unfolded over time. This kind of recurrence considers the historical information reflecting at the precise time points for each spiking neuron. Taking (Wu et al., 2018) for instance, the spatio-temporal backpropagation (STBP) algorithm was proposed to combine the layer-

by-layer spatio domain and the time-dependent temporal domain in a creative way. The non-differentiable during BP process is solved by approximated derivative for spike activity. The forget gate of the proposed iterative LIF neuron model is employed to control the leaky extent of the potential memory, which means the recurrence is introduced by the self-recurrent connection of individual spiking neurons. Considering the local memory captured by the precise step-by-step time point in the spiking neuron model, this method is more suitable for the tasks needing to memorize the relationships between contiguous time points, such as the image classification based on the dynamic vision sensor (DVS) datasets. While the other kind of learnable recurrent connections method is implemented by unfolding the recurrent layers over time and separating them into time windows. This method considers the global neural dynamics of recurrent connection from the group of neurons and ignores the detailed historical information on each single neuron. In addition, the leaky memory extent is decided by the learned weights in the hidden layer. This is consistent with the main recurrent framework in our recurrent SNN model. These recurrent connections implemented by unfolding the neurons in layer is more suitable for the human activity recognition application. Because the human activity recognition process pays more attention to global information preserved in the neuron group, not in a single neuron, then fast proceeds the recognition process by remitting the historical information in the recurrent layer. Furthermore, to make up the time gap between the input layer and the output of the hidden layer at the last moment, the learnable transmission delay is introduced to control the spike time distribution of the different input sources dynamically. Together with these analyses, our proposed model highlights the importance of using learnable recurrent connections

in SNNs to achieve computation advances.

As demonstrated in natural biological systems (Budd and Kisvárdy, 2012), transmission delays play a pivotal role in neural communication and computation. Several SNN models implement delay learning with different computational approaches. The direct optimization of delays between the internal and output neurons in reservoir models is attained through supervised learning (Paugam-Moisy et al., 2008). Similarly, delays are assigned to the connections between the pre- and post-synaptic neurons and learned by gradient descent rule in the Spikprop model (Bohte et al., 2002). As for the unsupervised learning method, a delay is increased if a post-synaptic pulse arrives before a defined temporal reference point, and decreased if it arrives after (Hüning et al., 1998). Hence, multiple postsynaptic pulses are synchronized. In line with these works, our proposed model utilizes the learnable delays to help discriminate the samples closed in time, hence enhancing the classification of temporal patterns.

In our current work, we show the capability of our model by using two types of datasets. One consists of biological spike patterns recorded from animal retinal neurons, where sequences of binary spikes are naturally triggered by image stimuli. The other dataset contains data from human motion trajectories, represented as a series of continual values sampled in real-life 3D space. Therefore, our framework can handle both analogue and continuous signals. A recent study shows that systems translating input signals into binary spikes greatly outperform those using continuous signals, even with the same neural network model (Zhang et al., 2020), suggesting the schematic advantage of computation with neural spikes (Yu et al., 2020).

Taken together, implementing SNN algorithms in software (Roy et al., 2019) or

processing signals as spikes or events on the neuromorphic chips (Ma et al., 2017; Davies et al., 2018; Pei et al., 2019) will greatly extend our computation capability and potentially give rise to the next generation of computing.

3 Methods

3.1 Overview of RSNN Model

Inspired by the information processing mechanism in the biological neural system, we propose an end-to-end spike-based framework with recurrent connections, named RSNN. As shown in Figure 1, the proposed model systematically incorporates the functional components, including neural encoding, learning, and decision-making layers, which in turn synergistically enable the RSNN to powerfully process complex multi-dimensional information with high accuracy.

3.1 Input Layer

Encoding refers to the mechanism to transfer the information from the outside stimuli to spikes. The existing encoding methods could be categorized as rate-based encoding and temporal encoding roughly. Rate-based methods (Gautrais and Thorpe, 1998) usually encode a value into a series of spikes, termed spike train, by systems such as the Poisson model. However, generating spike trains not only brings too much redundancy, but also ignores spike latency, the spike timing information. Compared to the rate-based encoding method, temporal encoding interprets the activity patterns in a spatiotemporal form with less redundancy where the precise timing of spikes is used for representing

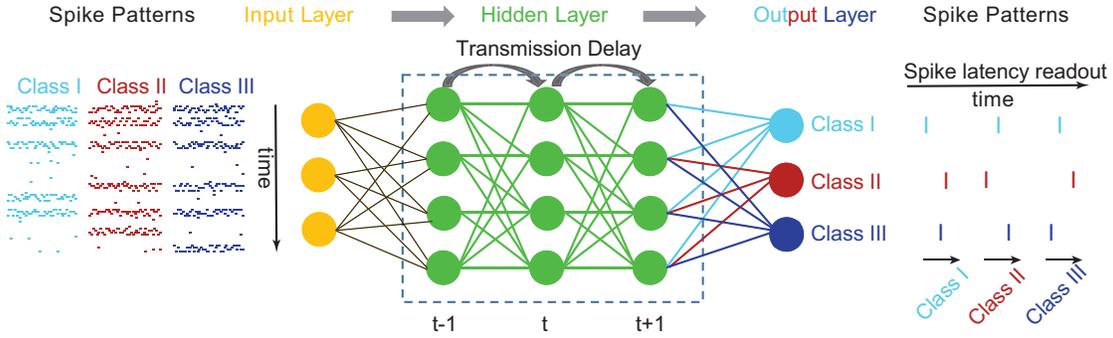


Figure 1: The framework of the RSNN model. Three different classes of spiking stimuli are fed into the input layer. The data are then encoded and passed to the recurrent layer where multiple time steps take place to relay the information. In each time step, information from the connected layer and previous time step is combined. The final readout is computed by the output layer, and the category of the input sample is determined by the first-firing neurons.

information. Mounting evidence suggests that neurons respond to stimulus with different spike timings (Gollisch and Meister, 2008; Egea-Weiss et al., 2018). We hence employed the temporal coding method in the proposed system due to its high precision and biological plausibility.

In our model, the network structure and the number of encoding neurons are fixed. Each neuron is permitted to fire only once. The input real-world floating data is translated into spatiotemporal spike pattern. The linear interpolation is employed to ensure all samples have a uniform time length. The maximum firing time for the encoding neurons is set to be $\ln(6) = 1.79s$ with $z = 6$ in the z -domain, in line with (Mostafa, 2018). This firing time setting guarantees a large enough temporal separation between the spikes interpreted from different real values in time-sequential data. After the en-

coding process, the information contained in the spike trains is transmitted into the hidden layer.

3.2 Recurrent Hidden Layer

The recurrent learning layer of the proposed RSNN, fully connected to the encoding neurons, is composed of recurrently networked hidden neurons and can unfold over time. Most classic training methods, such as Tempotron and SpikeProp, cannot train a recurrent model. Tempotron ignores the precise firing times of post-synaptic neurons and can overburden the training process since it requires the number of sample patterns to be approximately 3 times as many as the number of synapses. On the other hand, SpikeProp (Bohte et al., 2002) can train the multilayer SNNs by adopting an approximate gradient descent method to mimic the backpropagation (BP), which can impair the performance. Furthermore, SpikeProp cannot be implemented in machines with GPU. To overcome these problems, we propose a new training rule based on the previous work (Mostafa, 2018), which can be fitted into the recurrent spiking neural structures. Our formulation establishes an analytical relation between input and spike times with mathematical support. To accelerate the training process, our method can make full use of the existing GPU-based training packages to scale up the network capacity.

In addition, compared with ANN models where connection weight is the only programmable parameter, SNNs contain transmission delays that can be configured and used for achieving more powerful computation. In our RSNN model, we assign different transmission delays for each time step in the hidden layer. These delays are also learnable through the training process. In detail, for the time step $r \geq 1$ in the

recurrent layer, the input spike train T_r should be delayed by D_r , which means that $T_r = T'_r + D_r (r \geq 1)$, where D_r is constrained as $[0, 20]$ ms in line with biological plausibility. The transmission delay can balance the spikes from different input sources. In detail, when computing the output spike of one hidden neuron at r_{th} unfolded time step, there are two input sources with different weights. The One is the spike trains from the input layer, the other is the spike trains from the hidden layer at $r - 1_{th}$ time step. Due to the sequential property of spike trains, the latter spikes are usually more delayed than the former ones. Under the condition of no transmission delay in the recurrent layers, for instance, as illustrated in Fig. 2 (left), when computing the output firing time of this hidden neuron, the spikes from the input layer at r_{th} unfolded time step decide the first firing spike time of hidden neurons in a dominant way. Hence, for the first-time encoding mechanism in this paper, most of the hidden spikes at $r - 1_{th}$ time step could not contribute to the first firing spikes of the hidden neuron. Even worse, the time gap between those two input source could be more serious. In such a case, that situation could not be moderated only through adjusting their network weights. Hence, the learnable delay is introduced to balance the spike time distribution of two input sources. As shown in Fig. 2 (right), the transmission delay avoids the dominance of one input source. The output spike firing time of the hidden neuron is jointly decided by the current input spike from the input layer and the historical information from the hidden layer at the last time step.

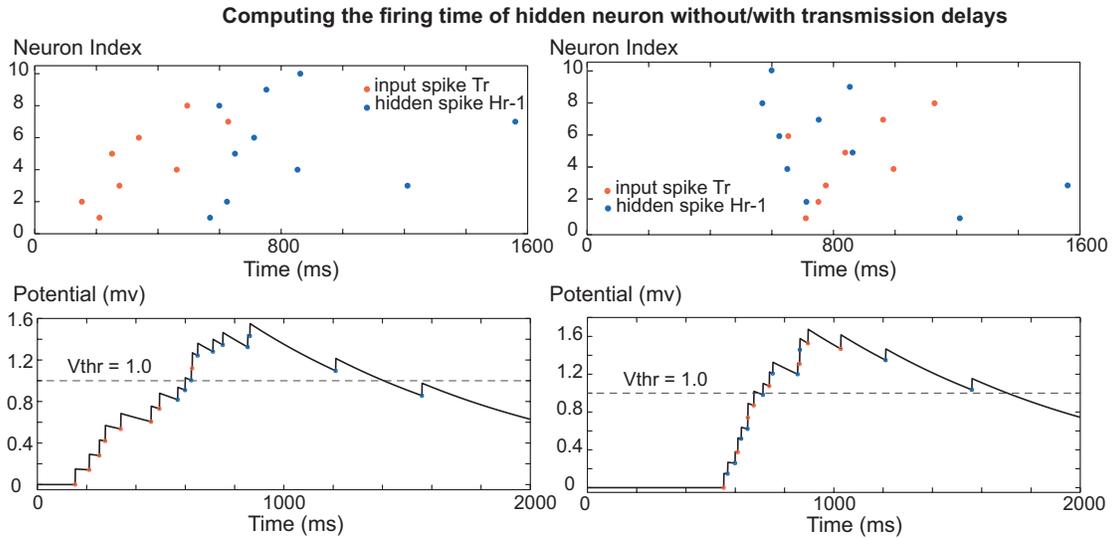


Figure 2: The simulation for computing the output firing time of one hidden neuron with/without transmission delay. (left) Without transmission delay, the output spike firing time of hidden neuron is dominated by input spike trains from input layer but ignoring the historical information on the spikes of hidden layers at the last time step. (right) With the learnable transmission delay, the relationship between the spike times from input layers and the spike times from hidden layers at the last time step becomes balanced.

3.3 Decision-Making Layer

For the decision-making layer, the label is defined by the neuron with the minimum spike timing. With the maximized value of negative spike timing in the z -domain, the neuron of the correct class would fire earlier than those from the incorrect ones, as shown in Equation 4.

3.2 Feedforward and Feedback Process

In this section, we derive the feedforward and feedback processes for non-recurrent layer and recurrent layer, respectively. During the feedforward phase, the output spike trains for each layer are recorded in order, while in the feedback process, all the connection weights are updated by the gradient descent rule.

3.1 The Feedforward and Feedback Processes in the Non-Recurrent Layer

We assume the post-synaptic neuron j receives N_I spikes at time $\{t_1, t_2, \dots, t_{N_I}\}$ with weights $\{w_1, w_2, \dots, w_{N_I}\}$ from N_I pre-synaptic neurons. Each neuron is permitted to fire only once. The membrane potential of neuron j is as follows:

$$V_j(t) = \sum_{i=1}^{N_I} \Theta(t - t_i) w_i (1 - \exp(-(t - t_i))), \quad (1)$$

where Θ is the Heaviside function. Once the value of V_j crosses the threshold $V_{thr} = 1$, the neuron j emits a spike. The causal set $C_j = \{i : t_i < t_j\}$ is defined to collect the pre-synaptic spikes that contribute to the firing of the first spike of the post-synaptic neuron. Hence t_j satisfies:

$$1 = \sum_{i \in C_j} w_i (1 - \exp(-(t_j - t_i))). \quad (2)$$

After completing a transformation for spike times by $\exp(t_x) \rightarrow z_x$, that is, spike time description in z-domain, the first spike of neuron j would be described as:

$$z_j = \frac{\sum_{i \in C_j} w_i z_i}{\sum_{i \in C_j} w_i - 1}. \quad (3)$$

For the feedforward process of fully-connected multilayer SNNs, the firing time of each neuron is calculated according to its causal set. If its causal set is empty, the output spike time is set to be infinite. The cross-entropy loss is used to prepare the neuron of the correct class fire earlier than the others in the output layer. Assuming that the spike time of the output layer is z_o and the target class label is g , the cost is as follows:

$$L(g, z_o) = -\ln \frac{\exp(-z_o[g])}{\sum_k \exp(-z_o[k])}. \quad (4)$$

Moreover, for the feedback process, the derivatives of the first spike time of pre-synaptic neuron in the z-domain and the corresponding weight are given by:

$$\frac{dz_j}{dw_i} = \begin{cases} \frac{z_i - z_j}{\sum_{i \in C_j} w_i - 1} & \text{if } i \in C_j, \\ 0 & \text{Otherwise.} \end{cases} \quad (5)$$

$$\frac{dz_j}{dz_i} = \begin{cases} \frac{w_i}{\sum_{i \in C_j} w_i - 1} & \text{if } i \in C_j, \\ 0 & \text{Otherwise.} \end{cases} \quad (6)$$

3.2 The Feedforward and Feedback Process in the Recurrent Layer

Assuming that there are two time steps in the recurrent hidden layer. For a hidden neuron d , the incoming spike train $T_i = \{t_1, t_2, \dots, t_{N_I/2}\}$ from the input layer has a transmission delay D_0 , that is, $T_i = T_i' + D_0$. The potential of the hidden neuron d can then be defined as:

$$V_d(t) = \sum_{i=1}^{N_I} \Theta(t - t_i) w_i (1 - \exp(-(t - t_i))) + \sum_{h=1}^{N_H} \Theta(t - t_h) w_h (1 - \exp(-(t - t_h))), \quad (7)$$

where w_h is the linking weight between d_{th} neuron in the current time step and the h_{th} neuron in the last time step. N_H denotes the total number of hidden neurons. t_h is the firing time of the h_{th} hidden neuron of the recurrent layer in the last time step, which can be computed by:

$$1 = \sum_{i=1}^{N_I} w_i (1 - \exp(-(t - t_i))) + \sum_{h=1}^{N_H} w_h (1 - \exp(-(t - t_h))). \quad (8)$$

Then z_h that represents the t_h in the z-domain is given by:

$$z_h = \frac{\sum_{i \in C_d} w_i z_i + \sum_{h \in C_d} w_h z_{h-1}}{\sum_{i \in C_d} w_i + \sum_{h \in C_d} w_h - 1}. \quad (9)$$

For the backward process, the partial differential for $\frac{dz_h}{dw_i}$ is:

$$\frac{dz_h}{dw_i} = \begin{cases} \frac{z_i - z_h}{\sum_{i \in C_d} w_i + \sum_{h \in C_d} w_h - 1} & \text{if } i, h \in C_d, \\ 0 & \text{Otherwise.} \end{cases} \quad (10)$$

The partial differential of z_h over z_{h-1} time step in z-domain is as follows:

$$\frac{dz_h}{dz_{h-1}} = \begin{cases} \frac{w_h}{\sum_{i \in C_d} w_i + \sum_{h \in C_d} w_h - 1} & \text{if } i, h \in C_d, \\ 0 & \text{Otherwise.} \end{cases} \quad (11)$$

Finally, the gradient of w_h can be calculated as:

$$\frac{dz_h}{dw_h} = \begin{cases} \frac{z_{h-1} - z_h}{\sum_{i \in C_d} w_i + \sum_{h \in C_d} w_h - 1} & \text{if } i, h \in C_d, \\ 0 & \text{Otherwise.} \end{cases} \quad (12)$$

In addition, transmission delay D_0 satisfies that $z(T_i) = z(T'_i + D_0)$, that is, $\exp(T_i) = \exp(T'_i + D_0)$. Its gradient $\frac{dz_h}{dD_0} = \frac{dz_h}{dz_i} \frac{dz_i}{dD_0}$ can be obtained by:

$$\frac{dz_h}{dD_0} = \begin{cases} \frac{\sum_{i \in C_d} w_i z_i}{\sum_{i \in C_d} w_i + \sum_{h \in C_d} w_h - 1} & \text{if } i, h \in C_d, \\ 0 & \text{Otherwise.} \end{cases} \quad (13)$$

Based on these formulas, the derivatives of other variables in the network can be obtained with backpropagation error through the layers using standard backpropagation technique. Besides, the constrains on synaptic weights and gradient normalization are applied to avoid neurons ceasing spike firing.

3.3 Datasets

3.1 Biological Neural Spike Data

We used two datasets of spiking patterns recorded from biological retinal ganglion cells. Experimental recordings were made in isolated retinas obtained from axolotl salamanders as described previously (Liu and Gollisch, 2015; Onken et al., 2016). The first dataset is for classification of moving direction. We analyzed the neural spike patterns in response to the drifting square-wave gratings, *i.e.*, black and white patterns, moving in eight different directions, with the entire sequence repeated five times, as described priorly (Liu et al., 2017). The second dataset contains flashed grating images with different spatial phases as reported in (Onken et al., 2016), we used a stimulus set of square-wave grating consisted of 60 shifted versions of the same grating images, uniformly covering the complete range of spatial phases of the grating. Images were

presented individually for 200 ms each in a pseudo-random sequence, with an inter-image interval of 800 ms in which a full-field gray stimulus was presented. At least 30 trials per grating image were recorded. For simplicity, we selected a few stimulus images and classified their corresponding neural spike responses by our model.

3.2 Human Action Dataset

We employed the MSRAction3D dataset (Li et al., 2010) collected by an MS Kinect-V1 camera to evaluate the performance of our model. The corresponding 3D joints positions are extracted from the depth sequences using the same real-time skeleton tracking algorithm as in (Shotton et al., 2011). This dataset consists of 20 actions performed by 10 subjects for 2 or 3 times. The categories of actions include “high arm wave”, “horizontal arm wave”, “hammer”, “hand catch”, “forward punch”, “high throw”, “draw X”, “draw tick”, “draw circle”, “hand clap”, “two hand wave”, “side-boxing”, “bend”, “forward kick”, “side kick”, “jogging”, “tennis swing”, “tennis serve”, “golf swing” and “pick up & throw”.

4 Results

The proposed RSNN model was evaluated on two types of datasets, biological neural spike data and 3D human action data. Firstly, we assessed the classification of stimuli in different moving directions using spike patterns of retina neurons, and compared the results with those from experiments using static images. Secondly, we tested our RSNN model on the MSRAction3D dataset for human trajectory activity recognition. Through comparison, our model demonstrated better performance than other SNNs. The effect

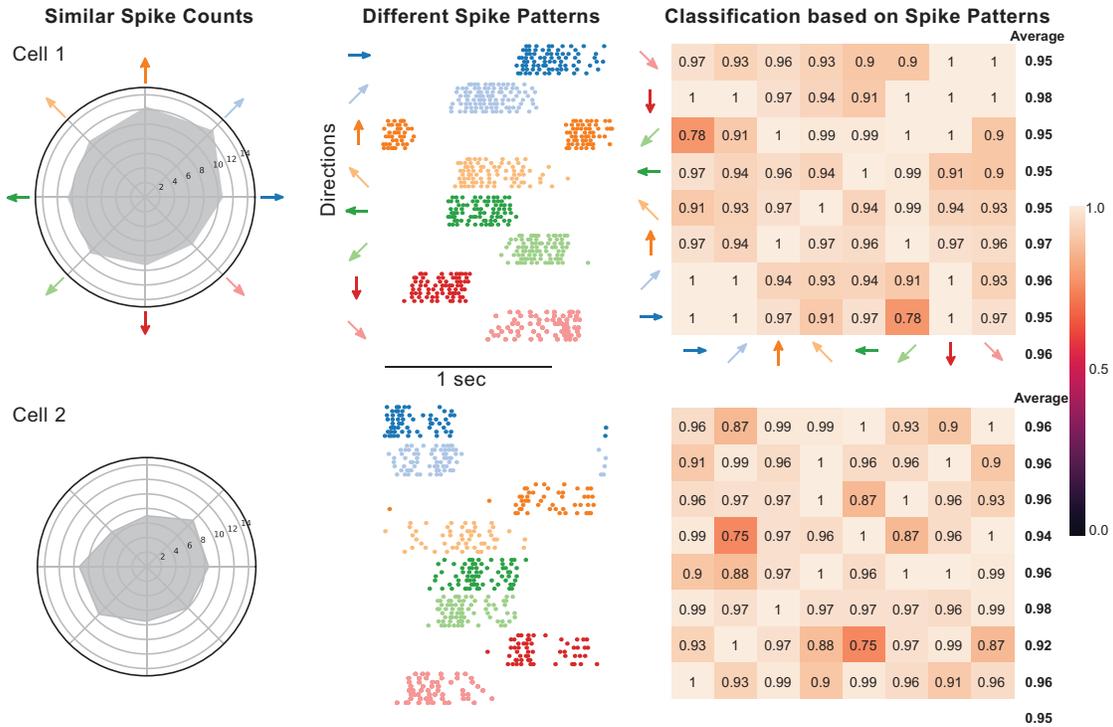


Figure 3: Classification of moving directions of neural spike patterns. (Left) Neural responses are similar among eight different directions as shown by the spike counts for two retinal neurons, cell 1 (top) and cell 2 (bottom). (Middle) Neural responses to eight different directions are distinguished as shown by the spatiotemporal spike patterns. (Right) High accuracy is shown by the confusion matrix of classification based on spike patterns. The average accuracies are shown beside the matrices.

of network structure with different numbers of hidden neurons was also evaluated.

4.1 Classification of Moving Directions using Neural Spike Patterns

We evaluated the performance of our RSNN model on biological retina cell data obtained from the retinal ganglion cells of salamanders in previous studies (Onken et al., 2016; Liu et al., 2017). The collected retina cell data consists of gratings moving in eight different directions as stimuli. We conducted binary classifications for different

directions and computed the accuracy. As shown in Figure 3, the average spike numbers in different directions are quite similar, which means that distinguishing these directions by spike number would be difficult.

We therefore employed the RSNN model to capture the spike timing and sequence distribution of spike trains to improve the classification performance. The RSNN structure was configured as 10 – 50 – 2, which means there are 10 encoding neurons, 50 hidden neurons, and 2 output neurons in the network. Only the first ten spikes were extracted to represent the input spike train of a neuron. If the spike number was less than ten, we would treat the last few missing spikes as not being fired by setting their firing times to be a large enough value. The number of time steps in the recurrent layer was set to be two, making the learnable delay a scalar. The original data was divided into the training set, validation set and test set with 64%, 16%, and 20% of the entire data, respectively. The accuracy on the test dataset was recorded when the validation set achieved the highest performance within 500 epochs.

As illustrated in Figure 3, for cell 1, the spike patterns in different directions have different time latency but similar spike numbers, which means the temporal coding with latency is more suitable for binary classification than the rate coding. Almost all pairs of directions are distinguished with high accuracy for their highly different time latencies. The direction classification of 0 and 225 degrees give a relatively low accuracy of 78%. The direction towards 270 degrees is recognized with the best average accuracy of 97.75%. It coincides with the feature of spike trains, where this direction is the most different from others in the gap between the first and second spikes. However, there are also some directions, such as the one towards 315 degrees, hard to be classi-

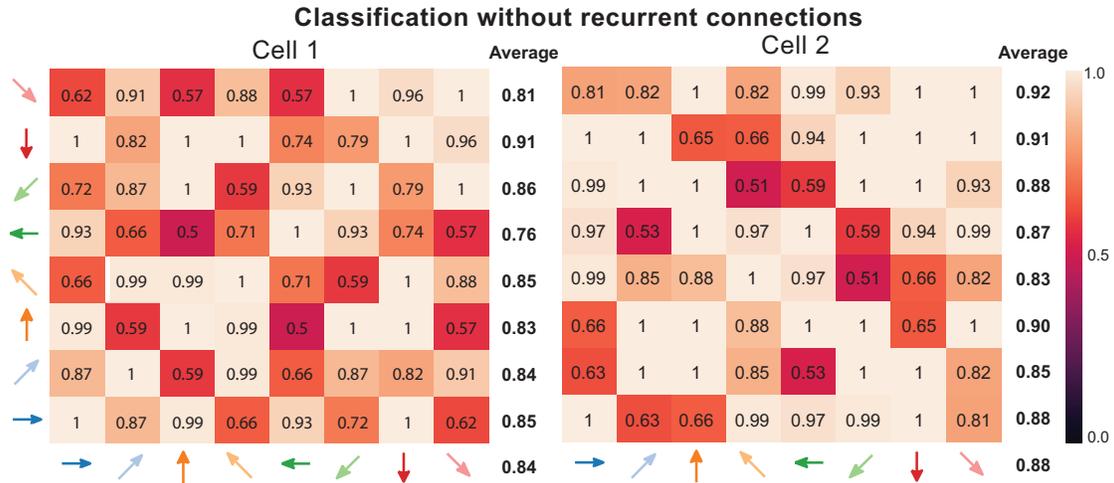


Figure 4: Recurrence in SNNs is necessary for classifying spatiotemporal patterns. Similar to Figure 3 for two example cells, but deleterious average accuracies (Bold font) are shown by confusion matrices obtained from SNNs without recurrent connections.

fied because of the small inter-class distances. For cell 2, the neuron responses of all directions are similar, except for those near 180 degrees. All pairs of directions can be classified by our RSNN model with high accuracy of over 85% except for those between 45 and 135 degrees with a performance of 75%. The direction towards 45 degrees is relatively harder to distinguish from others due to the small distances to other classes. As the results show, the RSNN model with temporal coding with latency information outperforms the rate coding-based models.

Moreover, we compared the performance of the RSNN model with the SNN model of the same $10 - 50 - 2$ network structure on these two retina cell datasets. As shown in Figure 4, the average accuracies of SNN are 83.87% for cell 1 and 88.10% for cell 2, much lower than those of the RSNN model (95.59% and 95.13%, respectively). Although there are more 1s in the matrix obtained by SNN model for cell 2, all the average accuracy of SNN model on each direction is quite lower than that obtained by RSNN

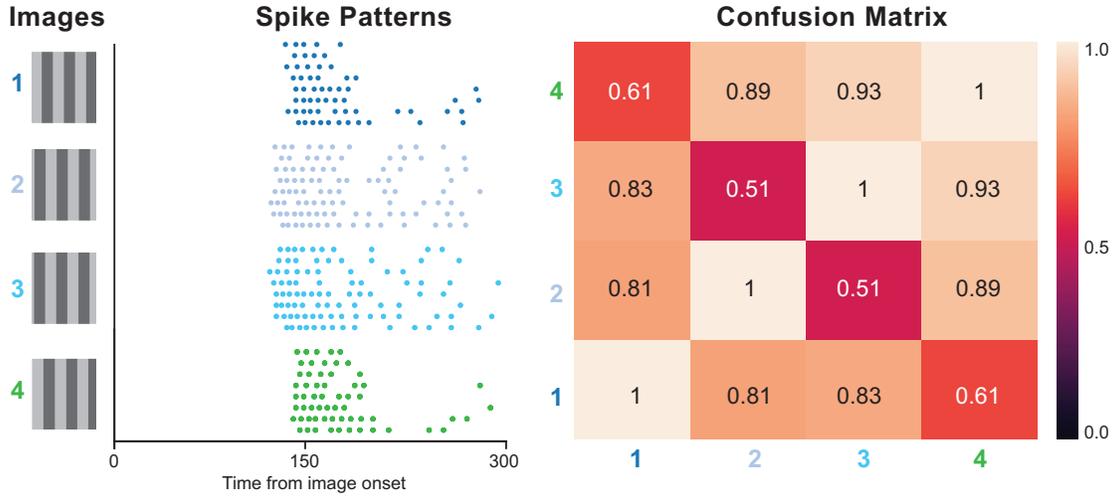


Figure 5: Image classification based on neural spike patterns. (Left) Four stimulus images of gratings with different phases. (Middle) Spike patterns triggered by stimuli in one retinal cell. (Right) Confusion matrix of classification for these four images based on neural spike patterns.

model. Moreover, we compute the variances of these accuracy matrices to observe the model’s stability. The variances of the matrices obtained by the RSNN model in Fig. 3 are 0.002 and 0.003 for cell 1 and cell 2, respectively. The variances of the matrices achieved by the SNN model in Fig. 4 are 0.027 and 0.025 for these two cells. This indicates the stability of our RSNN model compared with the original SNN model for these neural spike patterns classification. Thus, our RSNN model achieves better performance than the SNN model in the binary classification of cell response to directional moving stimuli.

4.2 Classification of Static Images from Neural Spike Patterns

The above results are based on the stimulus images of gratings moving in different directions. Knowing that the retinal neurons are sensitive to static stimulus images by

varying spike timings relative to the onset of the stimulus (Gollisch and Meister, 2008), we proceeded to evaluate the RSNN model on static stimuli by employing a dataset of the retinal ganglion cell response to different static images (Onken et al., 2016). As shown in Figure 5, the neural responses are characterized by spike patterns with different spike time, but not spike count. We performed the binary classification among four different grating images based on temporal spike patterns. Although the cell responses to different grating images exhibit similar spike latency, they are nonetheless well classified by the RSNN model as evidenced by the confusion matrix with an average accuracy of 82.29%. Hence, the RSNN model can also classify spike patterns with quite similar latency.

4.3 Classification of Human Action from Motion Trajectories

Next, we attempted to extend the capacity of our model to real-life applications beyond neural spike patterns. To this end, we employed the MSRAction3D80 dataset (Li et al., 2010) where all body action categories are divided into three subsets as in (Li et al., 2010), and each subset contains 8 actions, as illustrated in Table 1.

The subsets 1 and 2 contain actions of a similar movement, while subset 3 encompasses complex actions. Figure 6 visualized the moving trajectory of 'high throw' with twenty posture frames on x, y, and z axes each. These patterns were encoded into the spike trains which were then fed into the RSNN model. As previously described, the sample class is determined by the index of the output neuron that fires the earliest. The decoding spike trains of the RSNN model for samples of different categories are also shown. The output neuron corresponding to the correct class would generate the first

Subset 1	Subset 2	Subset 3
Horizontal arm wave	High arm wave	High throw
Hammer	Hand catch	Forward kick
Forward punch	Draw x	Side kick
Hand throw	Draw kick	Jogging
Hand clap	Draw circle	Tennis swing
Bend	Two hand wave	Tennis serve
Tennis serve	Forward kick	Golf swing
Pickup & throw	Side boxing	Pickup & throw

Table 1: The divided three subsets for MSR3D dataset of human motion trajectories.

spike the earliest.

We then performed a cross-subject test to compare the performance between our model and other methods. Half of the samples were selected randomly to form the training set and the rest are included in the test set. The task would be repeated 10 times to acquire average accuracy. The following three typical models used previously for this human action dataset were adopted:

- **Key-pose learning method (Miranda et al., 2014).** An approach based on decision forests, in which each forest node is a key pose. The key poses are identified through a multi-class classifier derived from Support Vector Learning machines among the body poses, which are described as a tailored angular representation of the skeleton joints.
- **Feedforward SNN with SRM (Yang et al., 2018).** A plain feedforward SNN model constructed as a two-layer SNNs with single neurons for human motion

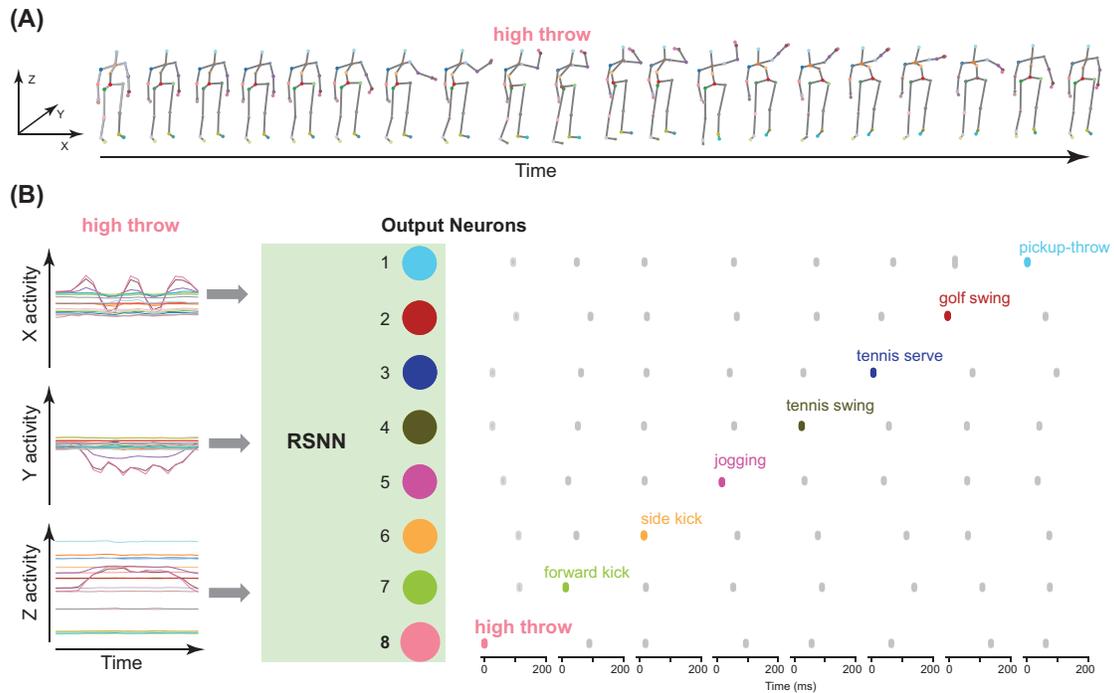


Figure 6: Classification of human motion trajectories by RSNN. (A) Visualization of human motion trajectory of 'high throw' demonstrated by a sequence of 20 recorded human postures in x, y and z direction. (B) Different action readouts by output neurons with different timings of first spikes. (Left) Human trajectories with 20 posture frames in x, y and z directions each. (Middle) Trajectories fed into the RSNN are read out by eight output neurons. (Right) Each action is decoded by an output neuron firing a spike with the shortest latency.

recognition, also described as a spike response model (Bohte et al., 2002). The movement shift in two different directions in the original 3D data is encoded into the spike trains, abandoning the motion movement range and velocity information.

- **Mostafa SNN (1200-800-8) (Mostafa, 2018).** The original SNN model with three layers containing 1200 input neurons, 800 hidden neurons, and 8 output

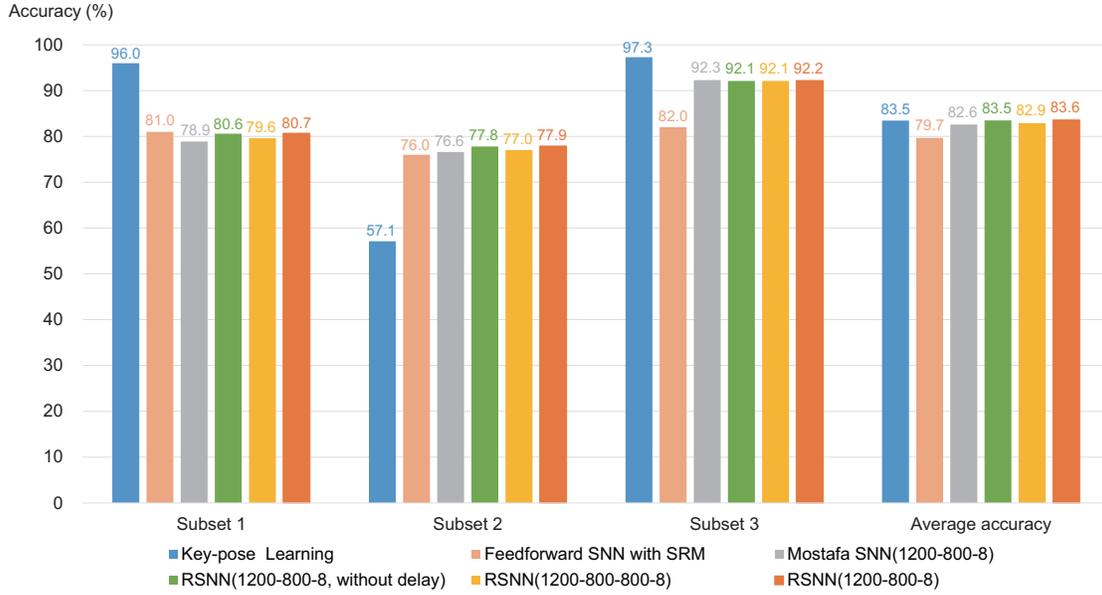


Figure 7: Comparison of model performance. The classification performance for each subset of human actions and the averaged performance over three subsets by RSNNs and three other models: Key-pose learning, Feedforward SNN with SRM, and Mostafa SNN. Our model is as robust as Mostafa SNN but with better performance.

neurons. All neurons in this model are fully connected without recurrence.

And the following three RSNN models with structural variation are used for comparison:

- **RSNN (1200-800-8, without delay).** The proposed RSNN model, consisting of an input layer, an output layer, and a hidden recurrent layer without the transmission delay mechanism.
- **RSNN (1200-800-800-8).** The proposed RSNN model, consisting of an input layer, two hidden recurrent layers, and an output layer.
- **RSNN (1200-800-8).** The proposed RSNN model with only one hidden recurrent layer.

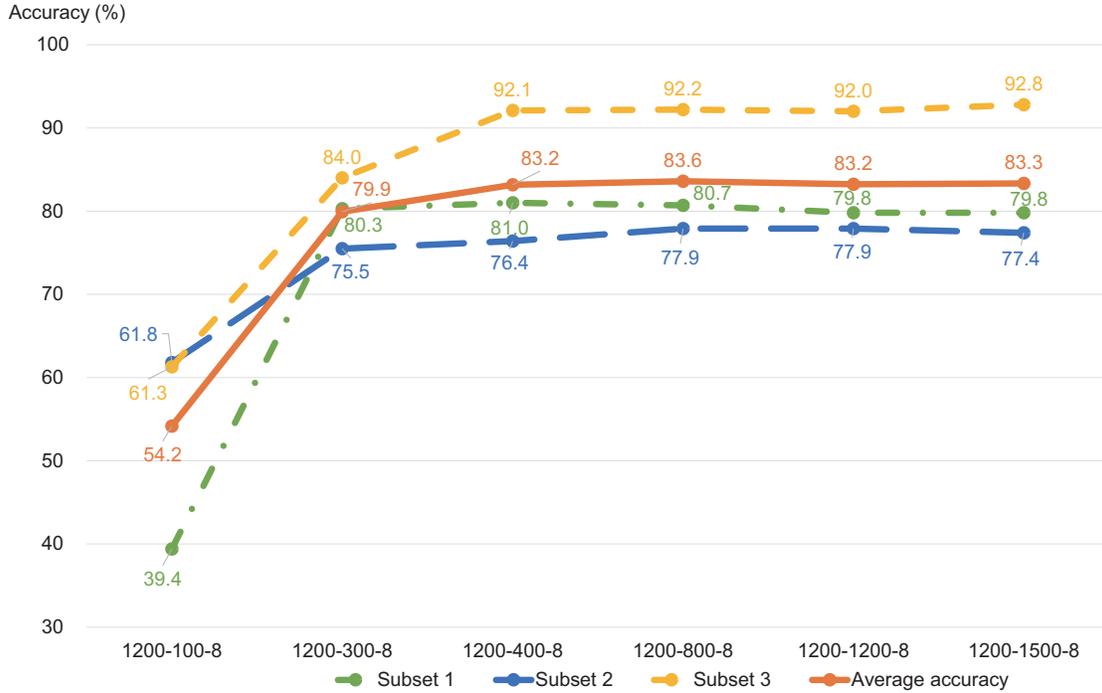


Figure 8: Classification performance of RSNN with different network structures, where the number of neurons in the hidden ranges from 100 to 1500.

As illustrated in Figure 7, the performances of different models are compared. Both the Key-pose Learning and Feedforward SNN with SRM methods perform unstably across the subsets. In contrast, the Mostafa SNN and our RSNN models show robust performance. In particular, our RSNN model with 1200-800-8 network structure performs the best with 83.6% accuracy. Compared with the Mostafa SNN model containing the same number of neurons (82.6% on average), the RSNN models show higher average accuracy. Besides, the accuracy achieved by RSNN model with delay is a bit higher than the model without delay (83.5%). It suggests the effect of learnable transmission delay. In summary, the RSNN model demonstrates competitive capability in solving human activity recognition problems.

Moreover, to explore the effect of network structure on performance, we designed

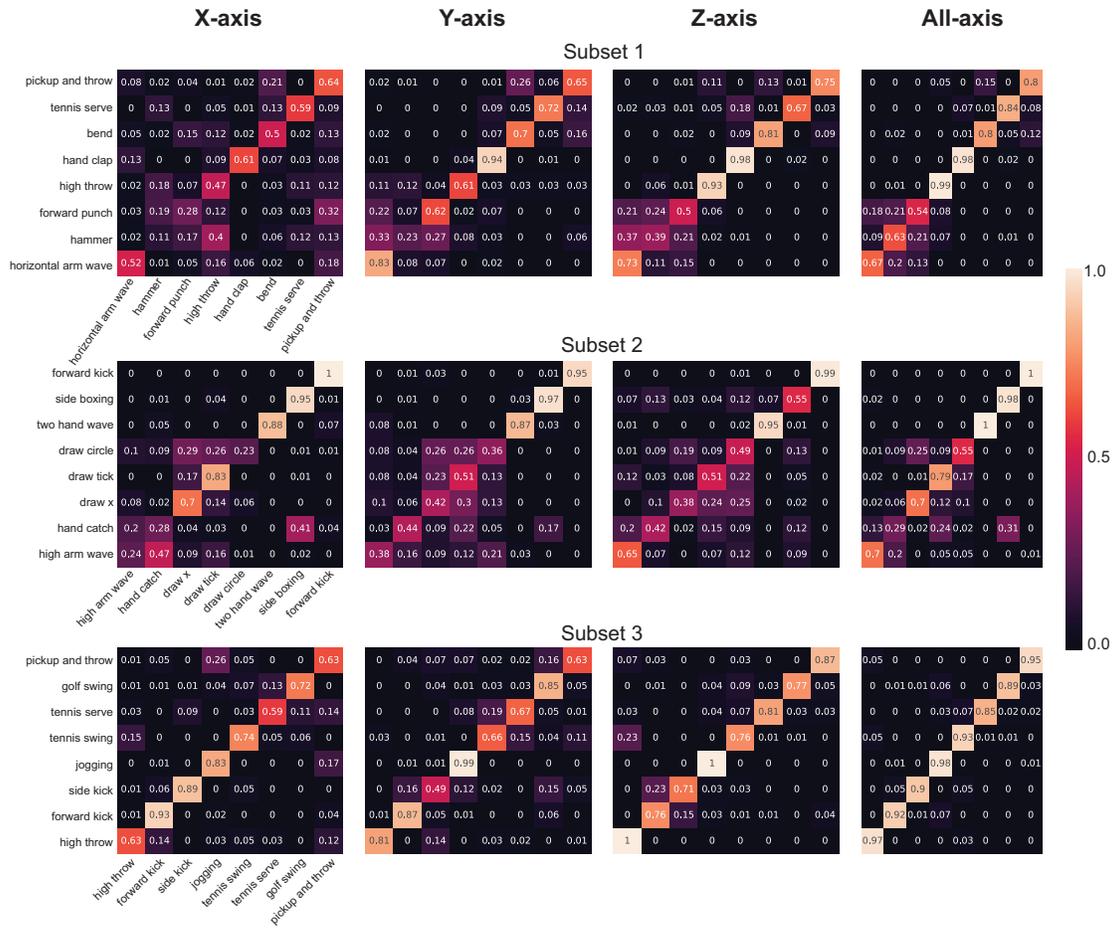


Figure 9: Classification of human actions in different channels. Confusion matrices of model performance on three subsets of actions along x, y or z channel, and all three channels combined.

a range of RSNN networks with different numbers of neurons in the hidden layer. As illustrated in Figure 8, the performance is improved upon increasing the number of neurons in the hidden layers from 100 to 800, indicating that the capability of a network can be enhanced by scaling up the network structure. However, continually increasing the neuron number in the hidden layer to 1500 failed to significantly improve the performance any further, suggesting little overfitting in our networks. Thus, our RSNN is suitable for solving complex tasks with a large scale of network structure.

To get more details of human action information coded in each direction, we performed the classification on each subset of action data along every motion direction, which can be referred to as a channel. In a 3D space, some human actions are more presented in one channel than the others. We explored the influence of different channels on the coding of human actions over time by computing the confusion matrix of accuracy on each channel for each subset of data.

As shown in Figure 9, the performance of the RSNN model in classifying actions along individual x, y, z channel, and all three channels combined are evaluated. The actions taking place on all axes are generally better classified than those on an individual channel. For instance, 'Tennis swing' action with only x or z channel data is hard to distinguish from 'high throw', while with y channel can be easily confused with 'tennis serve' and 'pick up and throw'. However, using data of all the channels, the classification of 'Tennis swing' can be improved by 21% in accuracy. Interestingly, some actions can be distinguished better with single-channel data only. For example, higher accuracy is achieved for 'Forward kick' with only x channel data than those using multiple direction information, presumably due to the fact that 'Forward kick' motion is mainly moving along x channel and not significantly presented in y and z channels.

Moreover, different actions achieve the relatively better accuracy with different preferable channels according to their moving trajectory. For instance, the performance on 'forward kick' and 'side kick' is better with x channel data only, 'golf swing' can be distinguished better with y channel information, while 'pick up and throw' and 'tennis serve' are more sensitive to z channel. These results suggest that the classifier should

be designed based on the nature of the actions, since body motions are not random but often over-represented in certain channels, as evidenced by the neural activities in the brain in response to body actions (Sabbah et al., 2017).

5 Conclusion

In this work, a recurrent spiking neural networks (RSNN) model is proposed to analyze spatiotemporal sequence data. We evaluated the performance of the RSNN model in classifying the biological neural spike patterns and real-life human motion trajectories. The results show that the RSNN model can achieve high performance for spatiotemporal pattern recognition. The learnable delay mechanism was introduced to improve the learning ability of the RSNN model by balancing the spike time distribution between different time steps in the recurrent layer. The efficient learning process of our model is easy to be adapted to GPU and other platforms. Moreover, our model is robust and scalable to perform complex tasks.

Acknowledgments

This work was partly supported by the grants from National Key *R&D* Program of China (2018YFA0701400), Zhejiang Lab (2019KE0AD01), Zhejiang Lab (2019KC0AB03 and 2019KC0AD02), the Royal Society Newton Advanced Fellowship (No. NAF-R1-191082), and the Fundamental Research Funds for the Central Universities.

References

- Arena, P., Patané, L., and Termini, P. S. (2012). Learning Expectation in Insects: a Recurrent Spiking Neural Model for Spatio-temporal Representation. *Neural Networks*, 32:35–45.
- Bohte, S. M., Kok, J. N., and La Poutré, H. (2002). Error-backpropagation in Temporally Encoded Networks of Spiking Neurons. *Neurocomputing*, 48(1–4):17–37.
- Budd, J. M. L. and Kisvárdy, Z. F. (2012). Communication and Wiring in the Cortical Connectome. *Frontiers in Neuroanatomy*, 6:42.
- Buonomano, D. V. and Maass, W. (2009). State-dependent Computations: Spatiotemporal Processing in Cortical Networks. *Nature Reviews Neuroscience*, 10(2):113–125.
- Buzsáki, G. (2010). Neural Syntax: Cell Assemblies, Synapsembles, and Readers. *Neuron*, 68(3):362–385.
- Cao, Y., Chen, Y., and Khosla, D. (2015). Spiking deep convolutional neural networks for energy-efficient object recognition. *International Journal of Computer Vision*, 113(1):54–66.
- Davies, M., Srinivasa, N., Lin, T.-H., Chinya, G., Cao, Y., Choday, S. H., Dimou, G., Joshi, P., Imam, N., Jain, S., et al. (2018). Loihi: A Neuromorphic Manycore Processor with On-chip Learning. *IEEE Micro*, 38(1):82–99.
- Devanne, M., Wannous, H., Berretti, S., Pala, P., Daoudi, M., and Del Bimbo, A. (2013).

- Space-time Pose Representation for 3d Human Action Recognition. In *ICIAP Workshop on Social Behaviour Analysis*, page 1, Naples, France.
- Diehl, P. U. and Cook, M. (2015). Unsupervised Learning of Digit Recognition Using Spike-timing-dependent plasticity. *Frontiers in Computational Neuroscience*, 9:99.
- dos Santos, V. L., Panzeri, S., Kayser, C., Diamond, M. E., and Quiroga, R. Q. (2015). Extracting Information in Spike Time Patterns with Wavelets and Information Theory. *Journal of Neurophysiology*, 113(3):1015–1033.
- Egea-Weiss, A., Renner, A., Kleineidam, C. J., and Szyszka, P. (2018). High Precision of Spike Timing Across Olfactory Receptor Neurons Allows Rapid Odor Coding in *Drosophila*. *iScience (2018)*, 4:76–83.
- Gautrais, J. and Thorpe, S. (1998). Rate Coding versus Temporal Order Coding: a Theoretical Approach. *Biosystems*, 48(1):57–65.
- Ghosh Dastidar, S. and Adeli, H. (2009). Spiking Neural Networks. *International Journal of Neural Systems*, 19(04):295–308.
- Gilra, A. and Gerstner, W. (2017). Predicting Non-linear Dynamics by Stable Local Learning in a Recurrent Spiking Neural Network. *Elife*, 6:e28295.
- Golestani, N. and Moghaddam, M. (2020). Human Activity Recognition Using Magnetic Induction-based Motion Signals and Deep Recurrent Neural Networks. *Nature Communications*, 11(1):1–11.
- Gollisch, T. and Meister, M. (2008). Rapid Neural Coding in the Retina with Relative Spike Latencies. *Science*, 319(5866):1108–11.

- Göltz, J., Baumbach, A., Billaudelle, S., Breitwieser, O., Dold, D., Kriener, L., Kungl, A. F., Senn, W., Schemmel, J., Meier, K., and Petrovici, M. A. (2019). Fast and Deep Neuromorphic Learning with Time-to-first-spike Coding. (2019) *arXiv:1912.11443*.
- Hüning, H., Glünder, H., and Palm, G. (1998). Synaptic Delay Learning in Pulse-Coupled Neurons. *Neural Computation*, 10(3):555–565.
- Kar, K., Kubilius, J., Schmidt, K., Issa, E. B., and DiCarlo, J. J. (2019). Evidence that Recurrent Circuits are Critical to the Ventral Stream’s Execution of Core Object Recognition Behavior. *Nature Neuroscience*, 22(6):974–983.
- Laje, R. and Buonomano, D. V. (2013). Robust Timing and Motor Patterns by Taming Chaos in Recurrent Neural Networks. *Nature Neuroscience*, 16(7):925–933.
- Lecun, Y., Bengio, Y., and Hinton, G. (2015). Deep Learning. *Nature*, 521(7553):436–444.
- Li, W., Zhang, Z., and Liu, Z. (2010). Action Recognition based on a Bag of 3D Points. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops*, pages 9–14.
- Liu, J. K. (2011). Learning Rule of Homeostatic Synaptic Scaling: Presynaptic Dependent or not. *Neural computation*, 23(12):3145–3161.
- Liu, J. K. and Buonomano, D. V. (2009). Embedding Multiple Trajectories in Simulated Recurrent Neural Networks in a Self-organizing Manner. *Journal of Neuroscience*, 29(42):13172–81.

- Liu, J. K. and Gollisch, T. (2015). Spike-Triggered Covariance Analysis Reveals Phenomenological Diversity of Contrast Adaptation in the Retina. *PLOS Computational Biology*, 11(7):e1004425.
- Liu, J. K., Schreyer, H. M., Onken, A., Rozenblit, F., Khani, M. H., Krishnamoorthy, V., Panzeri, S., and Gollisch, T. (2017). Inference of Neuronal Functional Circuitry with Spike-triggered Non-negative Matrix Factorization. *Nature Communications*, 8(1).
- Ma, D., Shen, J., Gu, Z., Zhang, M., Zhu, X., Xu, X., Xu, Q., Shen, Y., and Pan, G. (2017). Darwin: A Neuromorphic Hardware Co-processor based on Spiking Neural Networks. *Journal of Systems Architecture*, 77:43–51.
- Maass, W. (1997). Networks of Spiking Neurons: the Third Generation of Neural Network Models. *Neural Networks*, 10(9):1659–1671.
- Miranda, L., Vieira, T., Martinez, D., Lewiner, T., Vieira, A. W., and Campos, M. F. M. (2012). Real-Time Gesture Recognition from Depth Data through Key Poses Learning and Decision Forests. In *2012 25th SIBGRAPI Conference on Graphics, Patterns and Images*, pages 268–275.
- Miranda, L., Vieira, T., Martínez, D., Lewiner, T., Vieira, A. W., and M. Campos, M. F. (2014). Online Gesture Recognition from Pose Kernel Learning and Decision Forests. *Pattern Recognition Letters*, 39:65–73.
- Mostafa, H. (2018). Supervised Learning Based on Temporal Coding in Spiking

- Neural Networks. *IEEE Transactions on Neural Networks and Learning Systems*, 29(7):3227–3235.
- Onken, A., Liu, J. K., Karunasekara, P. P. C. R., Delis, I., Gollisch, T., and Panzeri, S. (2016). Using Matrix and Tensor Factorizations for the Single-Trial Analysis of Population Spike Trains. *PLOS Computational Biology*, 12(11):e1005189.
- Panda, P. and Roy, K. (2017). Learning to Generate Sequences with Combination of Hebbian and Non-hebbian Plasticity in Recurrent Spiking Neural Networks. *Frontiers in neuroscience*, 11:693.
- Panzeri, S., Harvey, C. D., Piasini, E., Latham, P. E., and Fellin, T. (2017). Cracking the Neural Code for Sensory Perception by Combining Statistics, Intervention, and Behavior. *Neuron*, 93(3):491–507.
- Paugam-Moisy, H., Martinez, R., and Bengio, S. (2008). Delay learning and polychronization for reservoir computing. *Neurocomputing*, 71(7):1143–1158.
- Pei, J., Deng, L., Song, S., Zhao, M., Zhang, Y., Wu, S., Wang, G., Zou, Z., Wu, Z., He, W., Chen, F., Deng, N., Wu, S., Wang, Y., Wu, Y., Yang, Z., Ma, C., Li, G., Han, W., Li, H., Wu, H., Zhao, R., Xie, Y., and Shi, L. (2019). Towards Artificial General Intelligence with Hybrid Tianjic Chip Architecture. *Nature*, 572(7767):106–111.
- Pyle, R. and Rosenbaum, R. (2017). Spatiotemporal Dynamics and Reliable Computations in Recurrent Spiking Neural Networks. *Physical review letters*, 118(1):018103.
- Quiroga, R. Q. and Panzeri, S. (2009). Extracting Information from Neuronal Popula-

- tions: Information Theory and Decoding Approaches. *Nature Reviews Neuroscience*, 10(3):173–185.
- Rieke, F., Warland, D., de Ruyter van Steveninck, R., and Bialek, W. (1996). *Spikes: Exploring the Neural Code (Computational Neuroscience)*, volume 7. The MIT Press.
- Roy, K., Jaiswal, A., and Panda, P. (2019). Towards Spike-based Machine Intelligence with Neuromorphic Computing. *Nature*, 575(7784):607–617.
- Sabbah, S., Gemmer, J. A., Bhatia-Lin, A., Manoff, G., Castro, G., Siegel, J. K., Jeffery, N., and Berson, D. M. (2017). A Retinal Code for Motion along the Gravitational and Body Axes. *Nature*, 546(7659):492–497.
- Shotton, J., Fitzgibbon, A., Cook, M., Sharp, T., Finocchio, M., Moore, R., Kipman, A., and Blake, A. (2011). Real-time Human Pose Recognition in Parts from Single Depth Images. In *CVPR*, pages 1297–1304.
- Soula, H., Beslon, G., and Mazet, O. (2006). Spontaneous Dynamics of Asymmetric Random Recurrent Spiking Neural Networks. *Neural Computation*, 18(1):60–79.
- Tsukada, M. and Pan, X. (2005). The Spatiotemporal Learning Rule and its Efficiency in Separating Spatiotemporal Patterns. *Biological Cybernetics*, 92(2):139–146.
- Wu, Y., Deng, L., Li, G., Zhu, J., and Shi, L. (2018). Spatio-temporal backpropagation for training high-performance spiking neural networks. *Frontiers in neuroscience*, 12:331.

- Xu, Q., Peng, J., Shen, J., Tang, H., and Pan, G. (2020). Deep CovDenseSNN: A hierarchical event-driven dynamic framework with spiking neurons in noisy environment. *Neural Networks*, 121:512–519.
- Yang, J., Wu, Q., Huang, M., and Luo, T. (2018). Real Time Human Motion Recognition via Spiking Neural Network. *IOP Conference Series: Materials Science and Engineering*, 366:012042.
- Yu, Q., Tang, H., Tan, K., and Li, H. (2013). Rapid Feedforward Computation by Temporal Encoding and Learning With Spiking Neurons. *IEEE Transactions on Neural Networks and Learning Systems*, 24(10):1539–1552.
- Yu, Z., Liu, J. K., Jia, S., Zhang, Y., Zheng, Y., Tian, Y., and Huang, T. (2020). Toward the next generation of retinal neuroprosthesis: Visual computation with spikes. *Engineering*, 6(4):449–461.
- Yuste, R. (2015). From the Neuron Doctrine to Neural Networks. *Nature Reviews Neuroscience*, 16(8):487–497.
- Zhang, Y., Jia, S., Zheng, Y., Yu, Z., Tian, Y., Ma, S., Huang, T., and Liu, J. K. (2020). Reconstruction of Natural Visual Scenes from Neural Spikes with Deep Neural Networks. *Neural Networks*, 125:19–30.