Optimizing Recurrent Spiking Neural Networks with Small Time Constants for Temporal Tasks

Yuan Zeng
yuz615@lehigh.edu
Lehigh University
Bethlehem, PA, USA

Edward Jeffs
elj221@lehigh.edu
Lehigh University
Bethlehem, PA, USA

Terrence C. Stewart
terrence.stewart@nrc-cnrc.gc.ca
National Research Council Canada
Ontario, Canada

Yevgeny Berdichevsky
yeb211@lehigh.edu
Lehigh University
Bethlehem, PA, USA

Xiaochen Guo
xig515@lehigh.edu
Lehigh University
Bethlehem, PA, USA

ABSTRACT

Recurrent spiking neural network (RSNN) is a frequently studied model to understand biological neural networks, as well as to develop energy efficient neuromorphic systems. Deep learning optimization approach, such as backpropogation through time (BPTT), equipped with surrogate gradient, can be used as an efficient optimization method for RSNN. Including dynamic properties of biological neurons into the neuron model may improve the network's temporal learning capability. Earlier work only considers the spike frequency adaptation behavior with a large adaptation time constant that may be unsuitable for neuromorphic implementation. Besides adaptation, synapse is also an important structure for information transfer between neurons and its dynamics may influence network performance. In this work, a Leaky Integrate and Fire neuron model with dynamic synapses and spike frequency adaptation is used for temporal tasks. A step-by-step experiment is designed to understand the impact of recurrent connections, synapse model, and adaptation model on the network accuracy. For each step, a hyper-parameters tuning tool is used to find the best set of neuron parameters. In addition, the influence of the synapse and adaptation time constants is studied. Results suggest that, dynamic synapse is more efficient than adaptation in improving the network's learning capability. When incorporating adaptation and synapse model together, the network can achieve a similar accuracy as the sate-of-the-art RSNN works while requiring fewer neurons and smaller time constants.

CCS CONCEPTS

• **Networks** → *Network performance analysis.*

KEYWORDS

Recurrent Spiking Neural Network, BPTT, Synapse, Spike Frequency Adaptation, Time Constant, Dynamics

ICONS 2022, July 27–29, 2022, Knoxville, TN, USA
© 2022 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-9789-6/22/07.
https://doi.org/10.1145/3546790.3546796

ACM Reference Format:
Yuan Zeng, Edward Jeffs, Terrence C. Stewart, Yevgeny Berdichevsky, and Xiaochen Guo. 2022. Optimizing Recurrent Spiking Neural Networks with Small Time Constants for Temporal Tasks. In *International Conference on Neuromorphic Systems (ICONS 2022), July 27–29, 2022, Knoxville, TN, USA.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3546790.3546796

1 INTRODUCTION

Recurrent spiking neural networks (RSNNs) are inspired by brain's dynamics [12] [28] and are used to develop energy efficient neuromorphic systems [19] [8] [6]. Different from artificial recurrent neural networks (ANNs), a set of dynamic functions is used to model the neuron behavior in RSNNs, and the information is processed and propagated through discrete spikes [17]. Although the brain is good at processing certain temporal tasks, RSNN can have a lower accuracy as compared to ANNs due to the lack of an efficient optimization method for training the dynamic features [24].

Recent studies [30] [3] showed that deep learning approaches, such as backpropagation through time (BPTT) [29], can be used as an efficient optimization mechanism for RSNN. With the use of surrogate gradient [21] [5], the discontinuous gradient of a spiking neuron can be estimated with a continuous function. Thus the gradient can be auto-calculated through the well-developed deep learning platform such as Tensorflow [1] and Pytorch [23]. Another finding is that incorporation of spike frequency adaptation can significantly increases the computing and learning capability of a RSNN [15]. RSNN equipped with adaptation and BPTT can achieve similar performance to its ANN counterparts on standard benchmarks such as sequential MNIST [13] and speech recognition [25].

In addition to time-dependent spike integration and frequency adaptation, synapses also contribute to dynamic behaviors of biological neural networks [11]. Synapse is an essential structure that permits a neuron to pass an electrical or chemical signal to another neuron. Dynamic behavior of the synapse can be described by an exponential decay function with a time constant τ_{syn} that is activated by a pre-synaptic spike. In contrast to spike frequency adaptation, which is dependent on a single neuron's firing rate, synapse dynamics of a specific neuron are influenced by the firing pattern of all the pre-synaptic neurons. To the best of our knowledge, the impact of synaptic and somatic dynamics, including spike frequency adaptation, on network's learning capability, has not been studied systematically.

In this work, a step-by-step protocol is designed to understand the impact of recurrent connections, synapse model, and adaptation model on the network accuracy. For each step, a hyper-parameters tuning tool [15] is used to find the best set of neuronal and synaptic parameters. Instead of the adaptive threshold implementation used in earlier works, in this work, spike frequency adaptation is implemented by an adaptation current. The adaptation current model is thought to be better in reproducing the neuron properties of a more realistic conductance-based model [4]. In addition, the influence of time constants on learning capability is also studied in this work. Results suggest that, when incorporating both adaptation and dynamic synapse mechanisms into the model together, smaller time constants ($< 50ms$) can be used to achieve good prediction accuracy. For analog neuromorphic system designs that use advanced CMOS technologies to implement spiking neurons, the rapidly increased leakage current is a limitation factor for achieving a large time constants [22] [18] [7]. More sophisticated device and circuit design is required to support large time constants. Findings in this work may help to design efficient neuromorphic systems. With optimized spike frequency adaptation and dynamic synapse time constants, this work achieves accuracy close to the state-of-the-art neural networks on sequential MNIST and Ti46-Digit speech dataset with fewer neurons and a single recurrent layer.

2 EXPERIMENTAL SETUP

This section describes the neuron model, network structure, input datasets, and hyper-parameters setting used in the experiments.

2.1 Neuron Model

$$\tau_{mi}\frac{du_i(t)}{dt} = -(u_i(t) - V_{resti}) + R_{mi}\sum_k W_{ki}x(t) + R_{syni}I_{syni}(t) - R_{adpi}I_{adpi}(t) \tag{1}$$

$$\tau_{syni}\frac{dI_{syni}(t)}{dt} = -I_{syni}(t) + \sum_j W_{ji}\delta(t-t_0) \tag{2}$$

$$\tau_{adpi}\frac{dI_{adpi}(t)}{dt} = -I_{adpi}(t) + b_i\delta(t-t_0) \tag{3}$$

$$u_i(t) = \begin{cases} V_{resti}, & \text{if } u_i(t-1) > V_{thi} \quad \text{or in refractory period} \\ u_i(t) & \text{otherwise} \end{cases} \tag{4}$$

The adaptive leaky integrate and fire (ALIF) [15] neuron model used in the paper is described by Eq. 1-Eq. 4. Eq. 1 models how membrane potential changes with time. In these equations, u represents a neuron's membrane potential, τ_m represents the membrane time constant, and V_{rest} means resting potential. $\sum_k W_{ki}x(t)$ is weighted sum of external inputs. R_{adp} is the membrane resistance. I_{syn} is the synaptic current that comes from other connected neurons. The term $I_{adp}(t)$ models the spike frequency adaptation behavior and R_{adp} is the adaptation resistance. Synapse dynamics are represented by Eq. 2, where R_{syn} is the synapse resistance, τ_{syn} is the synapse time constant and W_{ji} is the synapse weight between the modeled neuron i and its neighboring neurons j.

The dynamics of the adaptation current are described in Eq. 3. τ_{adp} is the adaptation time constant. The term $b\delta(t-t_0)$ means that, while a neuron is firing, the adaptation current is increased by an amount b when each local spike happens. Due to this effect, a neuron's firing rate will decay when a constant input is given. Different from earlier SNN studies on temporal tasks that implements the adaptation behavior through the dynamic threshold function, in this paper, spike frequency adaptation is implemented via an adaptation current. Although dynamic threshold and adaptation current can both be used to model the adaptation behavior, it has been shown that [4] the two approaches have different effect on the neuron's transfer function. Adding adaptation current to an integrate-and-fire neuron can better reproduce the biological functionality of a neuron.

A neuron fires when the membrane potential reaches a predefined numerical threshold V_{th}. After the spike injection, it will enter a refractory period where no other spike can happen. These are described by Eq. 4. For discrete time implementation, Forward-Euler first-order exponential integrator method is used with $d(t) = 1ms$. An example of Eq. 3 discretization is showing in Eq. 5, where $\alpha = exp(-d(t)/\tau_{adp})$.

$$I_{adpi}(t + d(t)) = \alpha I_{adpi}(t) + (1-\alpha)b_i\delta(t) \tag{5}$$

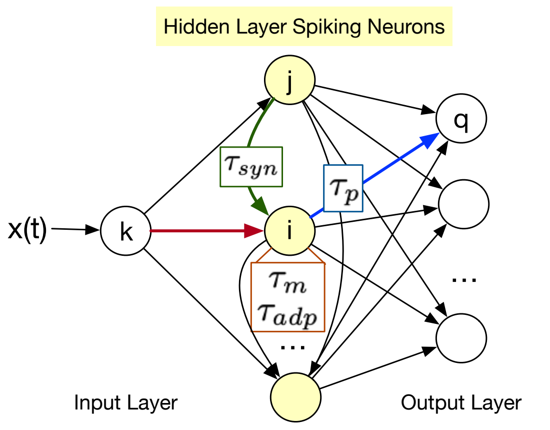

Figure 1: An illustration of the network structure used for the sequential MNIST.

2.2 Temporal Tasks and Network Structure

In this work, tasks are selected to test the network's temporal learning capabilities. For these tasks, inputs span multiple time steps and are given to the network one at a time step. The network needs to have the capability of incorporating the information from the past to make correct prediction.

2.2.1 Sequential MNIST. The MNIST [13] dataset contains 70,000 images for handwritten digits from zero to nine, each image has 28×28 pixels. When used for temporal task, each image is flattened into an one-dimensional pixel array. The input is given to the network one pixel at a time. Therefore, 784 time steps are needed to input one image. In this paper, one input neuron, n hidden layer neurons, and ten output neurons are used for the sequential MNIST task as shown in Fig. 1. Hidden layer neurons are recurrently connected, input to hidden and hidden to output neurons are fully connected. For a specific neuron (e.g., neuron i in Fig. 1), the red line represents the input spike train coming to the neuron, the green line represents the recurrent spike train coming to the neuron, and the blue line represent the spike train going out of the neuron. The membrane and adaptation time constants are internal parameters of a neuron cell, synapse time constant is a property of the connection and the output time constant is related to the output of this neuron.

2.2.2 Ti46-Digits Speech Dataset. Ti46-Digits [25] is the digits subset of the TI46 speech dataset. It contains read utterances from 8 males and 8 females each speaking digits zero through nine, with total 1594 training samples and 2542 testing samples. The corpus was collected at Texas Instruments in a quiet acoustic enclosure using an Electro-Voice RE-16 Dynamic Cardiod microphone at 12.5kHz sample rate with 12-bit quantization. The waveform is pre-processed by LyonPassiveEar model [16], which calculates the probability of firing along the auditory nerve. Then the analog data is encoded to 78 spike trains using Bens spiker algorithm (BSA) [27]. For this Ti46-Digits dataset, 78 input neurons are fully connected with the hidden layer neurons. The other parts of the network structure is the same as the MNIST task.

2.3 Learning Algorithm

$$\tau_{pi}\frac{dI_{qi}(t)}{dt} = -I_{qi}(t) + \delta(t - t_0) \tag{6}$$

$$Output_q = \sum_i W_{iq} I_{qi}(t_{end}) \tag{7}$$

$$\frac{dz_i(t)}{dv_i(t)} := \gamma max\left\{0, 1 - |v_i(t)|\right\} \tag{8}$$

2.3.1 Inference. For temporal tasks, output of the hidden layer neurons are spike trains that span multiple time steps. In this paper, the effect of the spike trains is described by an output current I_q through Eq. 6. τ_p is named output time constant and controls how fast the output current decays. After all the inputs are given, the output current observed at time step t_{end} is used to calculate network output (Eq. 7) and then passed through a softmax function for classification.

2.3.2 Training. Backpropagation through time (BPTT) algorithm is used in this paper. Similar as earlier spiking neural network works that adopt the BPTT algorithm, gradient is estimated for the non-differentiable function between a neuron's spiking output ($dz_i(t)$) and its membrane potential ($du_i(t)$). As described in Eq. 8, $dv_i(t) = \frac{du_i(t)-w}{w}$ is the normalized membrane potential and γ is the damping factor. This is also known as a linear surrogate gradient function as illustrated in Fig. 2.

2.4 Hyper-Parameters Tuning

There are many parameters in the neuron model as well as in the learning algorithm. These parameters can be classified into two categories. One is hyper-parameters that are pre-defined before the learning process starts, another is the parameters learned during the training process, such as input, recurrent, and output weights. In this paper, there are 11 hyper-parameters realted to the neuron model: $\tau_m, \tau_{syn}, \tau_{adp}, \tau_p, R_m, R_{syn}, R_{adp}, b, V_{rest}, Vth, w$. There are also parameters related to the learning algorithm such as learning rate and damping factor. Based on the experimental results,

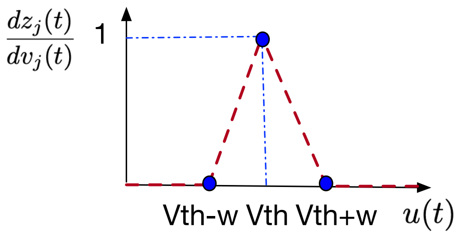

Figure 2: Illustration of the surrogate gradient function.

it is observed that these hyper-parameters can directly influence the learning ability and needs to be optimized. Finding a good set of hyper-parameters by hand-tuning or grid search is unfeasible. In this work, Optuna [2] is used for hyper-parameter tuning. Optuna is an open-source automatic hyperparameter optimization framework written in Python. It can efficiently search large optimization spaces and prune unpromising trials for faster results. The program is also easy to parallelize. By defining the error function, setting the number of trials, parameter tuning range, and optimization direction, Optuna can return the optimized tuning result for each trail.

3 EXPERIMENTS AND RESULTS

3.1 Method

In this work, the influence of each neuron and network component is evaluated with the experiments listed in Table 1: 1). Use leaky integrate and fire (LIF) neuron without dynamic synapse and adaptation in a network without recurrent connections in the hidden layer. Therefore, this network is a feedforward spiking neural network. 2). Add recurrent connections for the hidden layer based on setting 1. The network is a recurrent spiking neural network. 3). Add synapse dynamics to the neuron model based on setting 2. 4). Add adaptation to the neuron model based on setting 2. 5). Add both synapse dynamics and adaptation to the neuron model based on setting 2. 6). Constrain the synapse time constant based on setting 3 to be within 50ms. There is no adaptation for the neuron. 7). Constrain the synapse and adaptation time constant to be within 50ms based on setting 5. For the Ti46-Digit dataset, in order to better show the influence of each component, an additional constraint is added to settings 1-7 to restrict the output time constant to be within 20ms (Eq. 6). The reason behind this setting is introduced in the next subsection. In order to compare the result with and without this constraint, this paper includes a setting 0) for Ti46-Digit dataset, which is setting 1 without output time constant restriction.

For all the model and network settings listed above, 10 output neurons are used to classify the inputs to 10 classes. Sequential MNIST network has a single input neuron and 200 hidden layer neurons. Ti46-digists network has 78 input neurons and 100 hidden layer neurons. For each experiment setting, OPTUNA is used to find the network parameters that give the highest accuracy. Parameters tuned by OPTUNA and the search range are listed in Table 2. Here the maximum time constant is set to 700ms, which is close to the

	Recurrent	Synapse	Adaptation	Constraints
0	✗	✗	✗	/
1	✗	✗	✗	$*\tau_p < 20$
2	✓	✗	✗	$*\tau_p < 20$
3	✓	✗	✓	$*\tau_p < 20$
4	✓	✓	✗	$*\tau_p < 20$
5	✓	✓	✓	$*\tau_p < 20$
6	✓	✓	✗	$\tau_{syn} < 50$, $*\tau_p < 20$
7	✓	✓	✓	τ_{adp}, $\tau_{syn} < 50$, $*\tau_p < 20$

Table 1: Experiment List.* only apply to the Ti46-digit dataset

	τ_m (ms)	τ_{syn} (ms)	τ_{adp} (ms)	R_{adp} $m\Omega$	τ_p (ms)	V_{th} (mv)
baseline	[1,700]	[1,700]	[1,700]	[1,100]	[1,700]	[0.01, 10]
constraint	[1,700]	[1,50]	[1,50]	[1,100]	[1,20]	[0.01, 10]

Table 2: OPTUNA Tuning parameters and search range.

	τ_m (ms)	τ_{syn} (ms)	τ_{adp} (ms)	R_{adp} (mΩ)	τ_p (ms)	V_{th} (mv)	TuneAcc (epoch 1)	FinalAcc (epoch 100)
1	1	/	/	/	50	8.4	43%	42%
2	1	/	/	/	150	1.6	56%	72%
3	1	/	500	40	150	1.3	68%	80%
4	1	600	/	/	350	0.2	67%	86%
5	1	50	350	40	100	5.8	73%	90%
6	1	25	/	/	50	5.3	68%	79%
7	15	50	1	10	5	0.4	73%	95%

Table 3: Tuned parameters and Accuracy for sequential MNIST with Network Structure 1-200-10.

	τ_m (ms)	τ_{syn} (ms)	τ_{adp} (ms)	R_{adp} (mΩ)	τ_p (ms)	V_{th} (mv)	TuneAcc (epoch 3)	FinalAcc (epoch 100)
0	1	/	/	/	350	4	84%	97%
1	1	/	/	/	20	0.6	26%	33%
2	1	/	/	/	20	1.5	38%	37%
3	1	/	650	60	20	0.4	25%	50%
4	1	500	/	/	10	1.2	89%	98%
5	1	500	350	1	10	2.2	85%	98%
6	1	50	/	/	20	2	54%	94%
7	5	50	1	10	15	0.9	60%	97%

Table 4: Tuned parameters and Accuracy for Ti46-Digis with Network Structure 78-100-10.

maximum input time steps for the two datasets used in this paper. Range for R_{adp} and Vth are set empirically. V_{rest} is set to 0, w is set to Vth and Rm, R_{syn}, b, refractory period are set to 1 constantly for simplicity. The network learning rate is set to 0.1 and the damping factor is set to 0.3 based on experience. For each setting, OPTUNA runs for 200 trials to do parameter search. In order to get the result faster, for each trial, the sequential MNIST dataset runs for 1 epoch and the Ti46-digit dataset runs for 3 epoch. The best parameters and highest accuracy (TuneAcc) achieved by using this tool are reported in Table 4 and Table 3. After the parameters are found, they are used for weight training for a longer time, the accuracy at epoch 100 is reported as "FinalAcc" in Table 3 and Table 4. Results are explained in the following sections.

3.2 Understanding the Impact of Recurrent Connections, Synapse Dynamics, and Adaptation through Parameter Auto-Tuning

Comparing the results between setting 1 and setting 2 for the MNIST dataset, one observation is that the recurrent connection can significantly improve the network accuracy from 42% to 72% for sequential MNIST. However, for Ti46-Digit dataset, result of setting 0 suggests that the network can learn well even without the recurrent connection. This is because the speech dataset is a spatial temporal dataset and the spatial information can provide a good accuracy when the output layer integrates temporal information through a large time constant (i.e., τ_p=350ms). With this large output layer time constant, no additional temporal processing through the recurrent layer is required for the network in order to achieve a good accuracy. However, if the output time constant is restricted to a small value, then the temporal processing capability is necessary in order to hold information from earlier time steps. Therefore, to test the temporal processing capability of the network based on the Ti46-Digit dataset, for experiment settings 1-7, the output time constant is restricted to be less than 20ms. The result between setting 0 and setting 1 shows that after adding the constraint, the network accuracy drops from 97% to 33%. This is because the feed forward spiking neural network does not have the temporal processing capability. Interestingly, when adding recurrent connections based on setting 1, the accuracy only improved from 33% to 37%, this suggests that the LIF model without synapse dynamics and adaptation has limited temporal processing capability on the Ti46-digit dataset.

Experiment setting 3-5 are designed to understand the impact of adding synapse and adaptation model. For sequential MNIST dataset, adding adaptation only, adding synapse dynamics only, and adding both achieve 80%, 86%, 90% accuracy respectively. For Ti46-Digit dataset the corresponding experiment ends up with 50%, 98%, 98% accuracy respectively. The results suggest that, adding synapse current or adaptation current on the LIF model can help to improve the network's temporal processing capability. Among these two, synapse current shows greater influence on the accuracy. Adding both mechanisms can help improve the accuracy further for sequential MNIST. Another observation from the OPTUNA tuning results is that, the time constant suggested by the tool always stays at a relatively high value, when only synapse or adaptation current is modeled. The optimized time constant is $>= 500ms$. When both mechanisms are included, one of the time constant can be smaller.

In order to understand the influence of synapse and adaptation current better, two additional experiments are added with a constraint on the synapse and adaptation time constant. A smaller time constant can lead to more efficient analog neuromorphic implementation with spiking neurons. It is also interesting to check how accuracy changes with the reduction of the time constant, which will be introduced in more details in the next subsection. When the time constant is restricted to be within 50ms and the neuron only has synapse model, accuracy drops for both datasets. Comparing setting 4 and setting 6, for sequential MNIST, accuracy reduced from 86% to 79%. For Ti46-Digit, accuracy reduced from 98% to 94%. However, if adaptation current is added together with the dynamic synapse, good accuracy can be reached. For setting 7, sequential

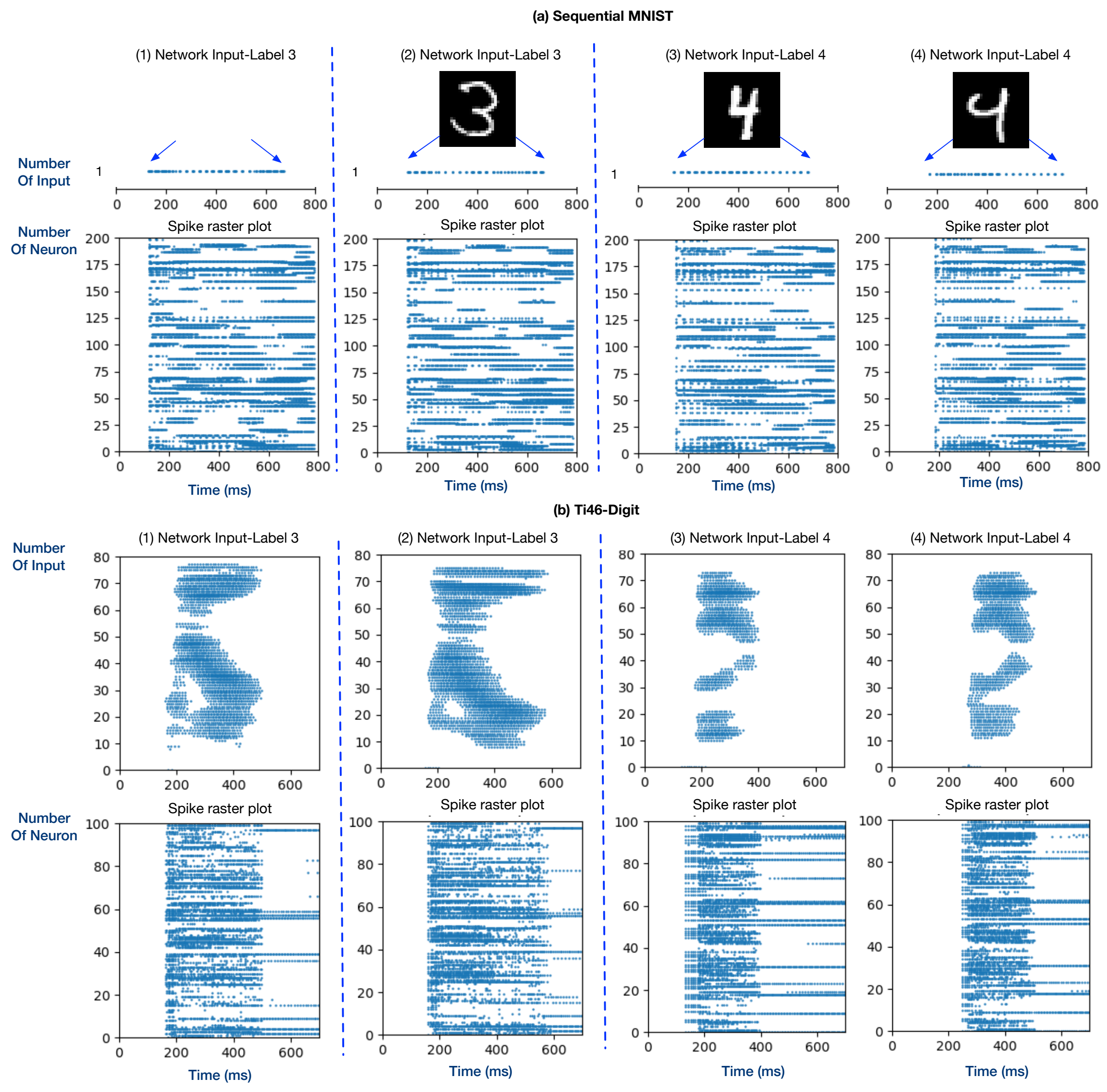

Figure 3: Examples of hidden layer spike raster plot after training with different inputs

MNIST gets 95% accuracy and Ti46-Digit gets 97% accuracy with synapse time constant at 50ms and adaptation time constant at 1ms. Result shows that, spiking neuron model with dynamic synapse and adaptation current does not require large time constants to solve the temporal tasks.

Figure 3 shows examples of the spike raster plots of hidden layer neurons under setting 7 with different network inputs after training. It is observed that, after applying a certain input, some specific neurons in the hidden layer continue to fire, which is enabled by the feedback loops in the recurrent network. Training the weights allows a similar group of neurons to continue firing in response to the same-label inputs, while different sets of neurons to continue firing in response to inputs with different labels. This is why the output layer can make a good prediction for these two tasks.

3.3 Sensitivity Study of Synapse and Adaptation Time Constants

To understand how accuracy changes with different synapse and adaptation time constants, grid search experiment for synapse and adaptation time constant is conducted based on experiment setting 7. In this experiment, both synapse and adaptation time constants are swept from 1ms to 700ms. In Fig.3, when the adaptation time constant is labeled as 0, the adaptation mechanism is turned off. A total of 56 experimental results are reported for each dataset. Each result shows the testing accuracy averaged from three trails. For the sequential MNIST, the accuracy is reported at epoch 20. For the Ti46-Digit dataset, the accuracy is reported at epoch 50.

For both tasks, accuracy drops with reduced synaptic time constant. This trend holds for different adaptation time constant as

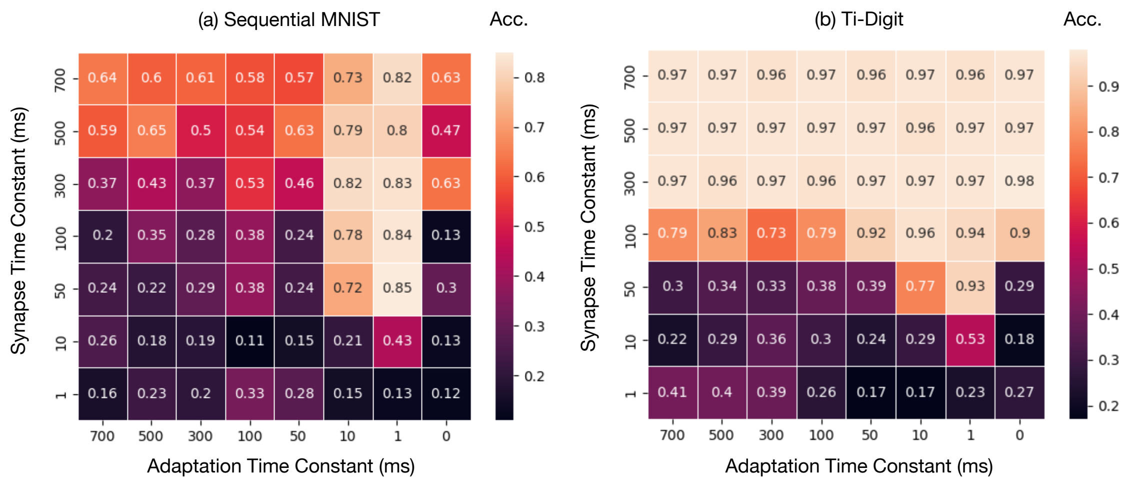

Figure 4: Heatmap for Synapse and Adaptaion Time Constant Change

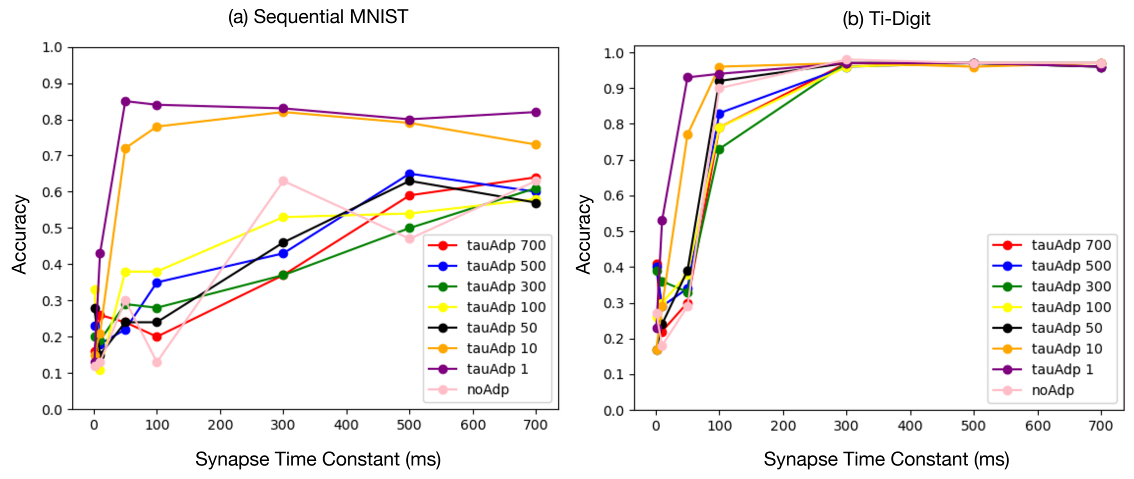

Figure 5: Synapse Time Constant Change with Network Accuracy

shown in Fig. 5. When adaptation time constant is reduced, however, the accuracy changes are not always monotonous. For most cases, the accuracy is significantly higher when adaptation mechanism is included in the model. This trend is most obvious when the synapse time constant is small. For example, when synapse time constant is 50ms, without adaptation, the accuracy is 30% for sequential MNIST and 29% for Ti46-Digit. After adding adaptation with time constant of 1ms, the accuracy improves significantly and reaches 85% and 93%.

Overall, experiment shows that synapse and adaptation time constant significantly influence network accuracy and need to be carefully tuned. Specific combinations of the synapse and adaptation time constant can significantly improve the performance.

3.4 Comparison with Related Work

Prior work on spiking neural networks primarily focused on the spatial tasks such as MNIST and CIFAR [20] [14] [26] [9]. Fewer studies have been published on temporal or spatial temporal tasks. [3], [30] and [31] propose to train recurrent neural network with BPTT on temporal tasks. In these works, an adaptation threshold mechanism is implemented and the network is tested with sequential MNIST. Best accuracy and the corresponding network structure

Work	Task	Model	Network	τ (ms)	Best Acc.
[3]	SMNIST	LIF+adaptive Vth	80-R700	20/-/700	97.1%
[30]	SMNIST	LIF+adaptive Vth	40-R256-R128	20/-/200	97.8%
[31]	SMNIST	LIF+adaptive Vth	64-R256-R256	20/-/200	98.7%
[32]	Ti46-Digit	LIF+synapse	78-200-R200-200	64/8/-	99.4%
[33]	Ti46-Digit	LIF+synapse	78-100-100-100	16/8/-	99.7%
This work	SMNIST	LIF+adaptive current+synapse	1-R400	1/50/1	98%
	Ti46-Digit		78-R100	1/50/1	98%

Table 5: Comparison with Related works. R Means Recurrent Layer, three elements under the τ column are $\tau_{membrane}, \tau_{synapse}, \tau_{adaptation}$

are listed in Table 5. Here for sequential MNIST task, 400 neurons are used for hidden layer to get better performance. In this paper, an adaptive current mechanism is used, which is thought to be closer to a realistic conductance-based neuron model [4]. Dynamic synapse, which is an important component of the biological neural network, is also added to the model together with the adaptation mechanism. Results suggests that model proposed in this paper can achieve a similar best accuracy as compared to prior SNN works with fewer number of neurons and only one recurrent layer. Other SNN works [32] [33] incorporates synapse model and is tested on the Ti46-Digit dataset, however, it does not take adaptive model into account. There is another work [10] that considers both synaptic dynamic and adaptation mechanism. But it was not evaluated on a long-term temporal task and did not study the impact of each dynamic component. That work was evaluated on Ti-digit-short, which converted the audio files into feature vectors through Mel-Frequency Cepstral Coefficients. After conversion, the length of the sequence is 90, which is shorter than the sequence length evaluated in this paper.

In this work, we are not only trying to achieve best accuracy with proposed model, but also to understand the influence of different component and parameters on the network accuracy. Based on the model proposed in this paper, one finding is that, although synapse and adaptation model can both help improving the network's temporal processing capability, when having either synapse or adaption model, synapse model is more efficient. This might because synaptic current is a global current which comes from neighboring neurons while adaptation current is local. Another finding is that, when both synapse and adaptation mechanism are incorporated in the network, a smaller synapse and adaptation time constants can be find to achieve a good accuracy. This is beneficial because a smaller time constant may lead to more efficient analog neuoromorphic implementation with spiking neurons. In [30], $\tau_{adp} = 700ms$ is used, which is similar as the dataset input length in the time domain. In [3], τ_{adp} with mean 200 ms, standard deviation 50 is used. In out work, a good accuracy can be achieved with synaptic time constant 50ms and adaptation time constant 1ms.

This work also shows that different neuron and network hyper-parameters can have significant impact on the result and hyper-parameters tuning is necessary prior to training. Different from [30], which takes the time constant as a trainable parameter, in this work, time constants are considered as the hyper-parameters and pre-tuned before the training starts. This is inspired from biological neurons, which have a fixed time constant.

4 CONCLUSION

This work studies how recurrent connections, adaptation model, and dynamic synapse model influence a spiking neural network's learning capability for temporal tasks. An automatic hyper-parameters tuning tool is used to find the best-achievable accuracy for different neuron and network settings. Results suggests that dynamic synapse is more efficient in improving the network's learning capability than adaptation. However, when incorporating both mechanisms into the neuron model, a set of smaller time constants can be found to achieve a good accuracy. This may help to simplify analog neuromorphic system implementations. This work achieves an accuracy close to the state-of-the-art on the pixel-by-pixel MNIST and Ti46-Digit speech dataset with fewer neurons and a single recurrent layer. Code of this work is avaliable at: https://github.com/yuanzenggit/RSNN-SmallTimeConstants.

REFERENCES

[1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. 2016. {TensorFlow}: A System for {Large-Scale} Machine Learning. In *12th USENIX symposium on operating systems design and implementation (OSDI 16)*. 265–283.

[2] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. 2019. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*. 2623–2631.

[3] Guillaume Bellec, Darjan Salaj, Anand Subramoney, Robert Legenstein, and Wolfgang Maass. 2018. Long short-term memory and learning-to-learn in networks of spiking neurons. *Advances in neural information processing systems* 31 (2018).

[4] Jan Benda, Leonard Maler, and André Longtin. 2010. Linear versus nonlinear signal transmission in neuron models with adaptation currents or dynamic thresholds. *Journal of Neurophysiology* 104, 5 (2010), 2806–2820.

[5] Sander M Bohte, Joost N Kok, and Johannes A La Poutré. 2000. SpikeProp: backpropagation for networks of spiking neurons.. In *ESANN*, Vol. 48. Bruges, 419–424.

[6] Maxence Bouvier, Alexandre Valentian, Thomas Mesquida, Francois Rummens, Marina Reyboz, Elisa Vianello, and Edith Beigne. 2019. Spiking neural networks hardware implementations and challenges: A survey. *ACM Journal on Emerging Technologies in Computing Systems (JETC)* 15, 2 (2019), 1–35.

[7] Thomas Dalgaty, Melika Payvand, Filippo Moro, Denys RB Ly, Florian Pebay-Peyroula, Jerome Casas, Giacomo Indiveri, and Elisa Vianello. 2019. Hybrid neuromorphic circuits exploiting non-conventional properties of RRAM for massively parallel local plasticity mechanisms. *APL Materials* 7, 8 (2019), 081125.

[8] Peter U Diehl, Guido Zarrella, Andrew Cassidy, Bruno U Pedroni, and Emre Neftci. 2016. Conversion of artificial recurrent neural networks to spiking neural networks for low-power neuromorphic hardware. In *2016 IEEE International Conference on Rebooting Computing (ICRC)*. IEEE, 1–8.

[9] Steven K Esser, Paul A Merolla, John V Arthur, Andrew S Cassidy, Rathinakumar Appuswamy, Alexander Andreopoulos, David J Berg, Jeffrey L McKinstry, Timothy Melano, Davis R Barch, et al. 2016. From the cover: Convolutional networks for fast, energy-efficient neuromorphic computing. *Proceedings of the National Academy of Sciences of the United States of America* 113, 41 (2016), 11441.

[10] Haowen Fang, Amar Shrestha, Ziyi Zhao, and Qinru Qiu. 2020. Exploiting neuron and synapse filter dynamics in spatial temporal learning of deep spiking neural network. *arXiv preprint arXiv:2003.02944* (2020).

[11] Wulfram Gerstner, Werner M Kistler, Richard Naud, and Liam Paninski. 2014. *Neuronal dynamics: From single neurons to networks and models of cognition*. Cambridge University Press.

[12] Umut Güçlü and Marcel AJ Van Gerven. 2017. Modeling the dynamics of human brain activity with recurrent neural networks. *Frontiers in computational neuroscience* 11 (2017), 7.

[13] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-based learning applied to document recognition. *Proc. IEEE* 86, 11 (1998), 2278–2324.

[14] Jun Haeng Lee, Tobi Delbruck, and Michael Pfeiffer. 2016. Training deep spiking neural networks using backpropagation. *Frontiers in neuroscience* 10 (2016), 508.

[15] Ying-Hui Liu and Xiao-Jing Wang. 2001. Spike-frequency adaptation of a generalized leaky integrate-and-fire model neuron. *Journal of computational neuroscience* 10, 1 (2001), 25–45.

[16] Richard Lyon. 1982. A computational model of filtering, detection, and compression in the cochlea. In *ICASSP'82. IEEE International Conference on Acoustics, Speech, and Signal Processing*, Vol. 7. IEEE, 1282–1285.

[17] Wolfgang Maass. 1997. Networks of spiking neurons: the third generation of neural network models. *Neural networks* 10, 9 (1997), 1659–1671.

[18] Christian Mayr, Johannes Partzsch, Marko Noack, Stefan Hänzsche, Stefan Scholze, Sebastian Höppner, Georg Ellguth, and Rene Schüffny. 2015. A biological-realtime neuromorphic system in 28 nm CMOS using low-leakage switched capacitor circuits. *IEEE transactions on biomedical circuits and systems* 10, 1 (2015), 243–254.

[19] Alexander Neckar, Sam Fok, Ben V Benjamin, Terrence C Stewart, Nick N Oza, Aaron R Voelker, Chris Eliasmith, Rajit Manohar, and Kwabena Boahen. 2018. Braindrop: A mixed-signal neuromorphic architecture with a dynamical systems-based programming model. *Proc. IEEE* 107, 1 (2018), 144–164.

[20] Emre O Neftci, Charles Augustine, Somnath Paul, and Georgios Detorakis. 2017. Event-driven random back-propagation: Enabling neuromorphic deep learning machines. *Frontiers in neuroscience* 11 (2017), 324.

[21] Emre O Neftci, Hesham Mostafa, and Friedemann Zenke. 2019. Surrogate gradient learning in spiking neural networks: Bringing the power of gradient-based optimization to spiking neural networks. *IEEE Signal Processing Magazine* 36, 6 (2019), 51–63.

[22] Marko Noack, Johannes Partzsch, Christian G Mayr, Stefan Hänzsche, Stefan Scholze, Sebastian Höppner, Georg Ellguth, and Rene Schüffny. 2015. Switched-capacitor realization of presynaptic short-term-plasticity and stop-learning synapses in 28 nm CMOS. *Frontiers in neuroscience* 9 (2015), 10.

[23] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems* 32 (2019).

[24] Michael Pfeiffer and Thomas Pfeil. 2018. Deep learning with spiking neurons: opportunities and challenges. *Frontiers in neuroscience* (2018), 774.

[25] Leonard R. Gary and Doddington George. 1993. TIDIGITS LDC93S10. https://catalog.ldc.upenn.edu/LDC93S10. Web Download, Philadelphia Linguistic Data Consortium.

[26] Bodo Rueckauer, Iulia-Alexandra Lungu, Yuhuang Hu, Michael Pfeiffer, and Shih-Chii Liu. 2017. Conversion of continuous-valued deep networks to efficient event-driven networks for image classification. *Frontiers in neuroscience* 11 (2017), 682.

[27] Benjamin Schrauwen and Jan Van Campenhout. 2003. BSA, a fast and accurate spike train encoding scheme. In *Proceedings of the International Joint Conference on Neural Networks, 2003.*, Vol. 4. IEEE, 2825–2830.

[28] Han Wang, Shijie Zhao, Qinglin Dong, Yan Cui, Yaowu Chen, Junwei Han, Li Xie, and Tianming Liu. 2018. Recognizing brain states using deep sparse recurrent neural network. *IEEE transactions on medical imaging* 38, 4 (2018), 1058–1068.

[29] Paul J Werbos. 1990. Backpropagation through time: what it does and how to do it. *Proc. IEEE* 78, 10 (1990), 1550–1560.

[30] Bojian Yin, Federico Corradi, and Sander M Bohté. 2020. Effective and efficient computation with multiple-timescale spiking recurrent neural networks. In *International Conference on Neuromorphic Systems 2020.* 1–8.

[31] Bojian Yin, Federico Corradi, and Sander M Bohté. 2021. Accurate and efficient time-domain classification with adaptive spiking recurrent neural networks. *Nature Machine Intelligence* 3, 10 (2021), 905–913.

[32] Wenrui Zhang and Peng Li. 2019. Spike-train level backpropagation for training deep recurrent spiking neural networks. *Advances in neural information processing systems* 32 (2019).

[33] Wenrui Zhang and Peng Li. 2021. Skip-connected self-recurrent spiking neural networks with joint intrinsic parameter and synaptic weight training. *Neural Computation* 33, 7 (2021), 1886–1913.