SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

January 2024, Vol. 67, Iss. 1, 112403:1–112403:10
https://doi.org/10.1007/s11432-022-3699-2

Conversion of a single-layer ANN to photonic SNN for pattern recognition

Yanan HAN[1], Shuiying XIANG[1*], Tianrui ZHANG[1], Yahui ZHANG[1], Xingxing GUO[1] & Yuechun SHI[2]

[1]*State Key Laboratory of Integrated Service Networks, State Key Discipline Laboratory of Wide Bandgap Semiconductor Technology, Xidian University, Xi'an* 710071*, China;*
[2]*Yongjiang Laboratory, Ningbo* 315202*, China*

Received 26 September 2022/Revised 15 November 2022/Accepted 8 February 2023/Published online 28 November 2023

Abstract This work presents a complete conversion scheme for photonic spiking neural networks (SNNs). We verified that the output of an artificial neural network (ANN) trained with the simulated optical activation function can be directly converted into the spike rate of a photonic spiking neuron model. To reveal the feasibility of hardware implementation, we considered the effects of different bit precisions of data and weight, noise level, and bias current mismatch on the converted results. The proposed scheme was evaluated using the Deterding vowel, IRIS, TIDIGITS, and MNIST datasets for pattern recognition, and achieved mean accuracies of 95.80%, 98.67%, 96.19%, and 92.33%, respectively. The proposed scheme can convert an ANN into a photonic SNN with almost no precision loss, and the performance was comparable to that of an ANN trained with the rectified linear unit function. The proposed scheme can enable the high-performance implementation of photonic SNNs.

Keywords photonic SNN, conversion, optical computing, pattern recognition, artificial neural network

1 Introduction

Artificial neural networks exhibit excellent performance in several practical applications, including pattern classification, detection, and sequence recognition. However, in recent years, spiking neural networks (SNNs) have received increasing attention, as spikes are transmitted in SNNs just as in real neural networks [1]. In addition to being more biologically plausible, SNNs feature low power consumption and low latency [2–4]. However, obtaining the high precision of artificial neural networks (ANN) and specific features of SNN has been a longstanding challenge.

Moreover, training an SNN is difficult mainly because the spike signals in SNN are intrinsically not differentiable, and the derivative is 0 when there is no spike, which limits the backpropagation (BP) of the precision error. There exist three mainstream SNN training approaches. The first approach is to train the network directly with the stochastic gradient descent either by modifying the neural membrane potential to a differentiable form, known as Spikeprop [5], or by avoiding the derivative of the membrane potential. To avoid calculating the derivative of the membrane potential, the layer-to-layer response model can be derived [2] to establish a direct expression between the spike rate or spike time and the synaptic weight. Additionally, the replacement of the derivative of spike with surrogate gradient functions during backpropagation has been proposed, and this method has shown good performance [4, 6, 7]. All of these methods have relatively high precision owing to the backpropagation of the global error. The second approach is to directly train the network according to the biologically plausible Hebbian learning rule, such as ReSuMe [8], Tempotron [9], and other spike-timing dependent plasticity (STDP)-based algorithms [10]. These methods update the synaptic weight according to the time difference between the pre- and postsynaptic spikes. Here, no global error information is propagated, resulting in relatively poor performance. Nevertheless, some studies have tried combining backpropagation and the STDP rule [3,11],

* Corresponding author (email: jxxsy@126.com)

and have achieved good performance. The third approach, that is, to previously train an ANN and then convert it into a rate-based SNN [12–15], is more straightforward. A conversion method involving the direct mapping of the connection weights to their SNN counterparts has been proposed by O'Connor et al. [12] and Cao et al. [13]. The spike rate of the integrate-and-fire (IF) neuron is mathematically equivalent to the well-known rectified linear unit (ReLU) function. According to the oversimplified IF neurons, the layer response model of the input and the output spike rates can be approximately derived, and the conversion scheme can therefore be extended to multilayer neural networks.

However, the IF neuron is an oversimplified theoretical model that lacks a basic spiking mechanism. Because the advantages of the SNN are most prominent in neuromorphic circuits, numerous studies have focused on the possibility of hardware implementation [1, 16–25]. The photonic platform has attracted much attention as a potential candidate for neuromorphic computing owing to its ultrahigh bandwidth, ultrafast response, and high energy efficiency [17, 19, 21, 25–27]. Several studies have focused on photonic neurons based on vertical-cavity surface-emitting lasers (VCSELs) [28–30], two-section lasers with an integrated or embedded saturable absorber [25,31–34], such as VCSELs, Fabry-Pérot laser and micropillar laser, micro-resonators [24, 35, 36], and phase-change materials [21, 37]. Several related photonic SNN architectures have also been proposed [10, 25, 35, 38–40]. In the relevant studies, the photonic SNN was directly trained using local learning algorithms, such as STDP, which can be applied to any neuron model according to the time difference of spikes; however, in most of the studies, the STDP protocols focused on simple patterns and will suffer from low learning precision for relatively complex problems. Because of the complexity of the photonic neuron model, it is difficult to directly differentiate the spike signals or derive a layer-to-layer response model as in IF neuron-based SNNs. To enhance the performance of SNNs based on hardware neurons, one study adopted a conversion method to replace the analog neurons in the last layer of an ANN with oscillation neurons, and experimentally demonstrated the method by converting the weighted sum of a memristor synapse array into the oscillation frequency of a neuron [20].

In this work, we propose to convert an ANN trained with an optical activation function into a photonic SNN, by mapping the analog output of the ANN into the spike rate of optical spiking neurons using a two-section photonic spiking neuron model. The neuron is demonstrated to be a physically realized leaky-integrate-and-fire model [33], which is characterized by low power consumption and can have high integration capacity [28]. Our main contributions are as follows: we propose the use of laser-based photonic neurons for converting a well-trained ANN into a photonic SNN, which provides theoretical support for the hardware implementation of photonic SNNs. The optical activation function is calculated and adopted to train a neural network, and the optical confinement is considered in the weight update. Additionally, four benchmark datasets are used to evaluate the network performance for pattern recognition. We also consider the influences of data/weight precision, and the noise and parameter mismatch on the accuracy of the network to verify the feasibility of hardware implementation.

2 Method

2.1 Principle of converting ANN to photonic SNN

Self-sustained pulsations, also known as passive Q-switching, have been observed in many kinds of laser systems [34]. We adopt a two-section optical neuron model based on VCSEL [33, 34], a simple structure of a two-section laser and the schematic diagram of the two-section VCSEL are illustrated in Figures 1(a) and (b), respectively. The adopted spiking neuron model could exhibit self-pulsations when subject to both optical and electronic injection [34, 41], the mechanisms of which have been clearly clarified in [34].

The basic principle of converting an ANN into SNN is to match the spike rate of spiking neurons and the activation of analog neurons [14, 20], inspired by which, the spike rate of a photonic neuron can be used as an index for pattern classification. The rate equations of the adopted spiking neuron model can be described as [10, 33, 41]

$$\frac{\mathrm{d}S_i(t)}{\mathrm{d}t} = \Gamma_a g_a(n_{a,i} - n_{0a})S_i + \Gamma_s g_s(n_{s,i} - n_{0s})S_i - \frac{S_i}{\tau_{ph}} + \beta B_r n_{a,i}^2, \tag{1}$$

$$\frac{\mathrm{d}n_{a,i}(t)}{\mathrm{d}t} = -\Gamma_a g_a(n_{a,i} - n_{0a})(S_i - \Phi_i) - \frac{n_{a,i}}{\tau_a} + \frac{I_{a,i}}{eV_a}, \tag{2}$$

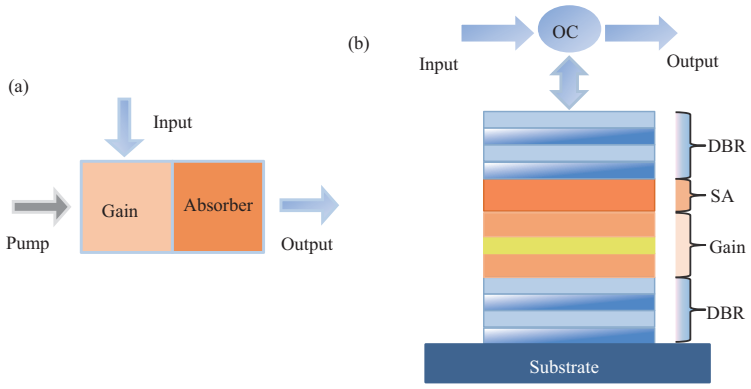

Figure 1 (Color online) (a) Simple structure of a two-section laser; (b) schematic diagram of VCSEL-SA [33, 34].

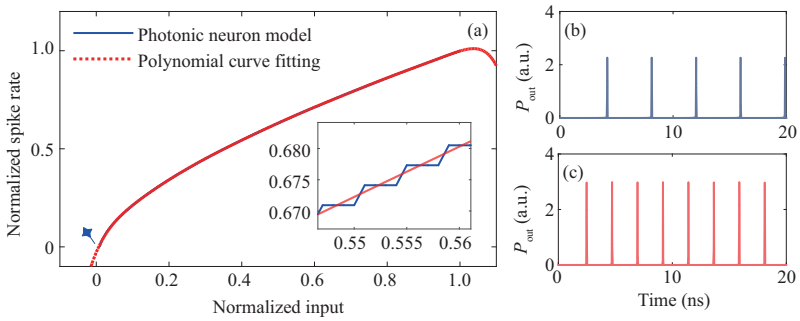

Figure 2 (Color online) (a) Normalized spike rate of the photonic spiking neuron as a function of normalized optical injected power (blue solid line) and the corresponding polynomial fitting (red dotted line); (b) and (c) outputs of the photonic spiking neuron with $P = 0.01$ and $P = 0.05$, respectively.

$$\frac{\mathrm{d}n_{s,i}(t)}{\mathrm{d}t} = -\Gamma_s g_a(n_{s,i} - n_{0a})S_i - \frac{n_{s,i}}{\tau_s} + \frac{I_{s,i}}{eV_s}, \tag{3}$$

$$\Phi_i = \frac{\tau_{ph}\lambda_i}{hcV_a} + \sum_{j=1}^{N} w_{ij}P_j, \tag{4}$$

$$P_i(t) \approx \frac{hc\eta_c\Gamma_a S_i(t)V_a}{\tau_{ph}\lambda_i}, \tag{5}$$

where the subscripts a and s stand for the gain and absorber regions, respectively. S represents the photonic density in the cavity of the i-th neuron. $n_{a,i}$ ($n_{s,i}$) describes the carrier density in the gain (absorber) region. $w_{i,j}$ describes the weighted coupling from neuron j to i. $I_{a,i} = 2.3$ mA is the bias current of gain region, $I_{s,i}$ is the bias current in the SA region, which is 0 mA. P is the output power. The basic parameters can be found in [10].

To match the activation to an optical spiking neuron, we calculate the spike rate of the photonic spiking neuron model as a function of optical injected power. Here, due to the time scale of the generated pulse, the spike rate is calculated as the total number of spikes divided by the simulation time T, and the normalized result is shown in Figure 2(a) (blue solid line). The simulation time T is set to 200 ns. It can be seen that the spike rate increases with increasing input power. As examples, the output time series of the optical spiking neuron with $P = 0.01$ and $P = 0.05$ are given in Figures 2(b) and (c), respectively. It should be noted that such a neuron model acts like class 2 neuron, the spike rate of which is not continuous. The normalized input power at the onset of self-pulsation is about 0.01, as marked by the blue star. The red solid line in Figure 2(a) is the corresponding polynomial fitting with degree 10, which is performed via MATLAB.

High order fitting is chosen to ensure the precision of fitting as much as possible. However, in the inset of Figure 2(a) we can see that the simulated curve has a stepwise form, which may cause misclassification when the weighted sums of samples from different classes are located on the same step. This could essentially be improved by increasing the simulation time T. Next, the 10th order polynomial is utilized as the activation function of the neural network.

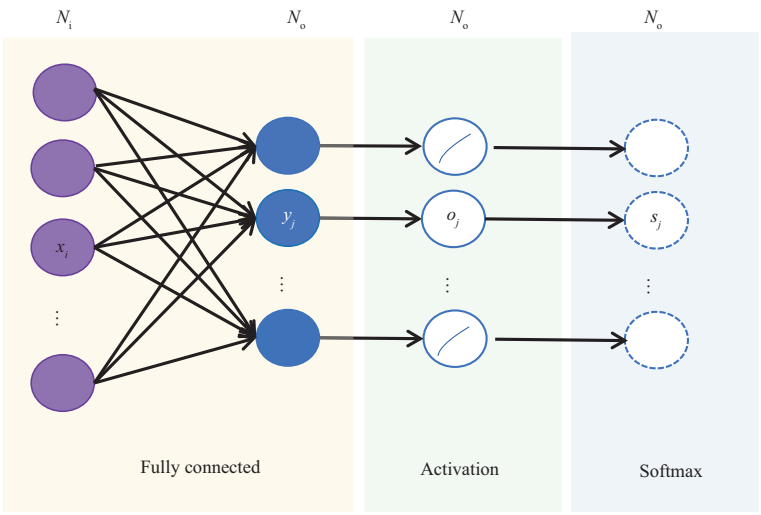

Figure 3 (Color online) Network structure of a single-layer fully connected neural network.

2.2 Training method

We consider a single-layer fully connected neural network consisting of N_i input neurons and N_o output neurons, as shown in Figure 3. Here, the cross-entropy-based method is selected for training, and the total loss of the network is calculated as

$$L = -\sum_{j=1}^{N_o} \log(s_j - d_j), \tag{6}$$

where d_j stands for the one hot label of the j-th output neuron, $s_j = e^{o_j} / \sum_{i=1}^{N_\mathrm{o}} e^{o_i}$ is the output of Softmax, $o_j = h(y_j)$ is the output of the j-th output neuron after activation, $y_j = \sum_{i=1}^{N_\mathrm{i}} w_{i,j} x_i$ is the weighted sum of the input data x_i to neuron j, and $h(\cdot)$ denotes the fitted optical activation function, as illustrated in Figure 2(a). According to the BP rule, the synaptic weight between neurons i and j can be updated by

$$\Delta w_{i,j} = (s_j - d_j) \cdot \partial h(y_j) \cdot x_i, \tag{7}$$

$$w_{i,j} = w_{i,j} + \eta \cdot \Delta w_{i,j}, \tag{8}$$

where η is the learning rate. Note that in the optical domain, it is naturally difficult to express negative pulse and negative weight. Here, to directly match the optical constraint, we limit the weights to non-negative values during training by setting the negative value to a positive random number p:

$$w_{i,j}(w_{i,j} < 0) = p \cdot \kappa, \tag{9}$$

where κ describes the level of noise added to the negative weight. We will demonstrate in the following that, via setting the negative weights to random numbers instead of 0, the network performance can be effectively improved to some extent.

2.3 Normalization

The input x_i is encoded into the power of optical injection, which needs to be positive. Hence, the input needs to be properly normalized. To fit well with the input range of the simulated optical activation function in Figure 2, before training, the input data is firstly normalized into the range [0.01, 1] via

$$\hat{x_i} = a_1 + \kappa \cdot (x_i - x_{\min}), \tag{10}$$

where $k = (\alpha_2 - \alpha_1)/(x_{\max} - x_{\min})$. $\alpha_1 = 0.01$ is the lower bound of data normalization, and $\alpha_2 = 1$ denotes the upper bound. $x_{\max}$ (x_{mim}) is the maximum (minimum) value of data. In this way, the optical

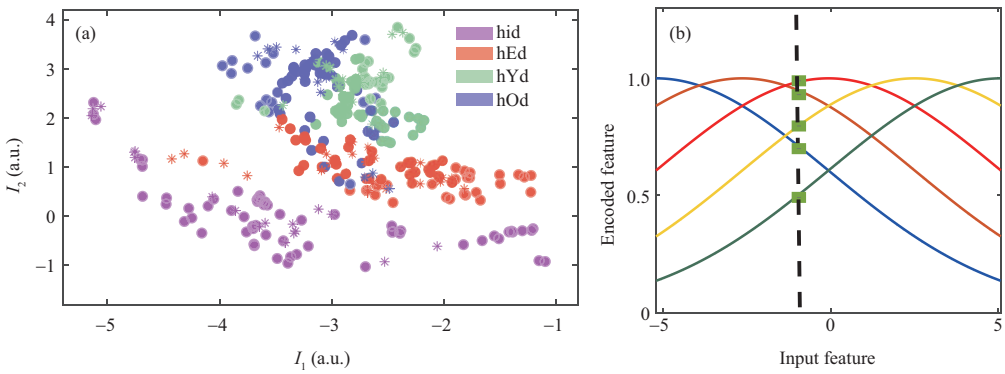

Figure 4 (Color online) (a) Two-dimensional projection (the first two features) of the training dataset (circles) and test dataset (stars); (b) encoding scheme.

input power in an actual test can be determined by multiplying the normalized value by the normalized factor.

In addition, the randomly set initial weights could also be normalized via the number of input neurons as $\hat{W} = W/N_{\mathrm{i}}$. In this way, the weighted sum will be confined within a proper range.

3 Result

3.1 Training with optical activation function

The proposed method is firstly evaluated in a single-layer network for vowel recognition. The Deterding dataset of 11 steady state vowels is used for demonstration [42], which consists of 990 samples of 15 different speakers, each containing 10 linear predictive coding derived log area ratios. We select 4 classes out of a total of 11, and randomly select 60 samples from each class as the training dataset and the rest as the test dataset. The normalized two-dimension features of both training set (circles) and test set (stars) are presented in Figure 4(a). It can be seen that 'hYd' and 'hOd' are highly mixed.

The Deterding dataset has 10 log area ratios as input features. We adopt Gaussian encoding to expand the input dimension, and each feature can be encoded via several Gaussian functions [40]. The center of the i-th Gaussian function is given by

$$\mu_i = I_{\min} + \frac{2i-3}{2} \cdot \frac{I_{\max} - I_{\min}}{m-2}, \ i = 1, 2, \ldots, N_f, \tag{11}$$

where N_f is the number of Gaussian functions, $I_{\min}$ ($I_{\max}$) is the minimum (maximum) value of input data. The variance of the Gaussian function is calculated as

$$\sigma = \frac{1}{\theta} \cdot \frac{I_{\max} - I_{\min}}{m-2}, \tag{12}$$

where θ is set to 1. Figure 4(b) illustrates the encoding process, where the abscissa represents the range of the original feature and the ordinate represents that of the encoded feature. The vector formed by the intersection of a certain input feature (black dotted line) and all Gaussian functions (green square) constitutes the encoded feature.

After encoding, the network could be trained according to (7)–(9). To minimize the conversion loss of the network trained with the previously obtained optical activation function, it is necessary to confine the weighted sums of all samples within the range of the simulated input during training. In addition to the normalization approach described in Section 2, we can further achieve this via pre-training the network with ReLU activation function, and rescale the normalized range of spike rate to the max activation value of ReLU. Here we choose $N_f = 20$, $\kappa = 0$, and $\eta = 0.006$, and repeat the experiment for 6 times with different data distributions and different initial weights. The network loss trained with ReLU (yellow) and the optical activation function (blue) is given in Figure 5(a), where the bars indicate the standard error. Figure 5(b) shows the corresponding test accuracy. The fluctuations are mainly caused by the distribution difference of the training set and test set which are randomly chosen for each trial. We can

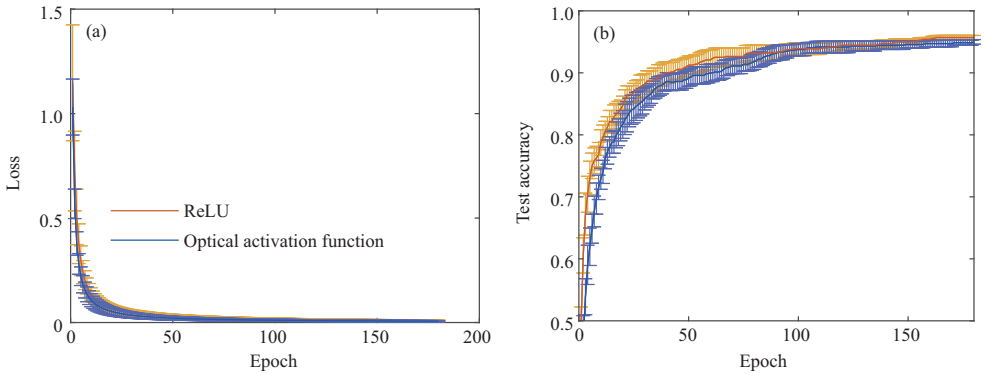

Figure 5 (Color online) (a) and (b) Loss and test accuracy of Deterding vowel trained with ReLU (yellow) and optical activation function (blue) for 6 trials. The error bars indicate standard error ($\eta = 0.006$).

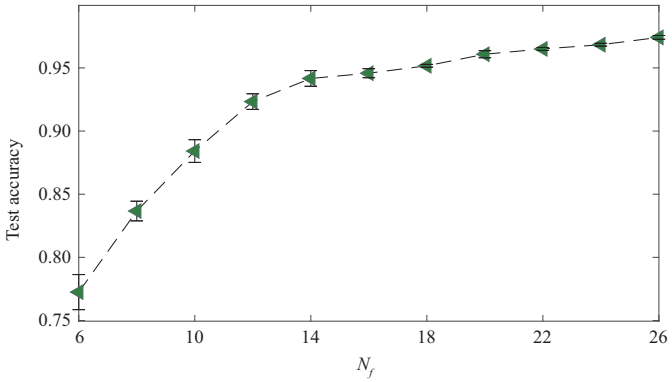

Figure 6 (Color online) Test accuracy trained with the fitted optical activation function as a function of N_f.

see that the mean test accuracy trained with the optical activation function is very close to that trained with ReLU.

Figure 6 shows the mean test accuracy trained with the optical activation function as a function of the number of Gaussian functions N_f, where the error bars represent the standard error. The test accuracy shows an obviously increased average value and decreased fluctuation with the increase of N_f.

Besides, the impact of noise introduced to the negative weight during training is also analyzed. In Figure 7, we present the average train and test accuracy of 6 different trials trained under different noise levels κ. To some extent, this introduced noise can help the network jump out of the local optimum, thus improving network performance ($\kappa = 0.001, 0.01, 0.1$). But on the other hand, it may also cause damage to network performance, especially in the case of relatively high noise intensity ($\kappa = 1$). For a clearer comparison, we show the training and testing process at two different noise levels, namely $\kappa = 0.001$ and 0.01, in Figures 7(c) and (d). It is also not surprising to see from the error bar that larger noise can lead to unstable performance.

3.2 Conversion

Then we try to convert a trained network into a photonic SNN. Here we choose $N_f = 20$, $\kappa = 0.01$, and achieve a test accuracy of 95.80% when trained with the optical activation function. The distribution of all 200 coupling weights to each of the four optical output neurons is presented in Figures 8(a1)–(a4), from where we can see that most of the weights are in the range of 0 to 1. With the normalized synaptic weight, optical inference could be directly implemented in the network. Theoretically, the weight can be adjusted between 0–1 via a variable optical attenuator, MRR weight bank or MZI array, etc. And in practice, the weights are usually calibrated by measuring input and output power.

Figures 8(b1)–(b4) show examples of the output of photonic neurons. The first output neuron obviously has the highest spike rate; therefore the input sample will be classified as class 'hid'. The confusion matrix of the converted result is presented in Figure 8(c), the mean accuracy of which is 95.80%, indicating that there is no loss after conversion.

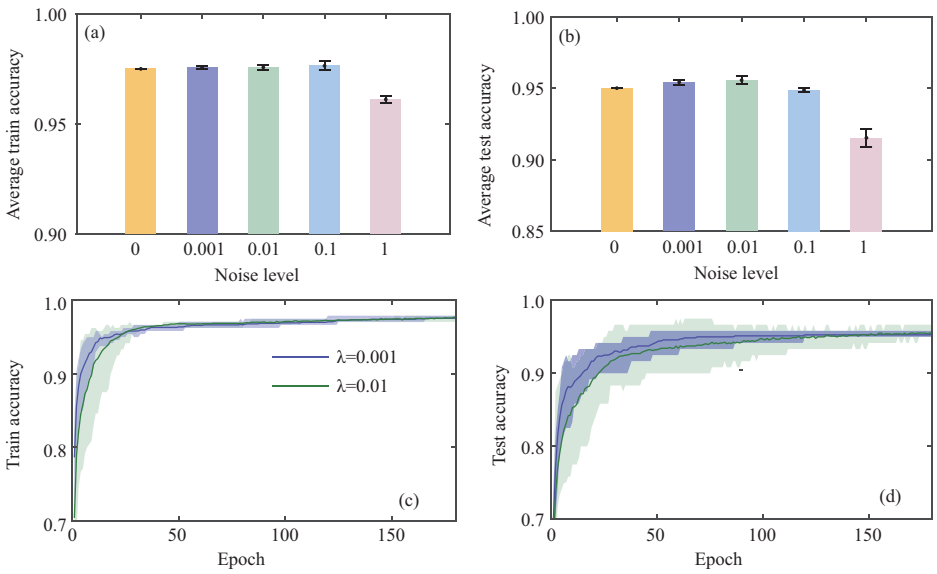

Figure 7 (Color online) Training accuracy (a) and test accuracy (b) trained with the optical activation function under different levels of noise on the negative weight; (c) and (d) the training and testing process with $\kappa = 0.001$ and 0.01, respectively, where the patches indicate the boundary of accuracy.

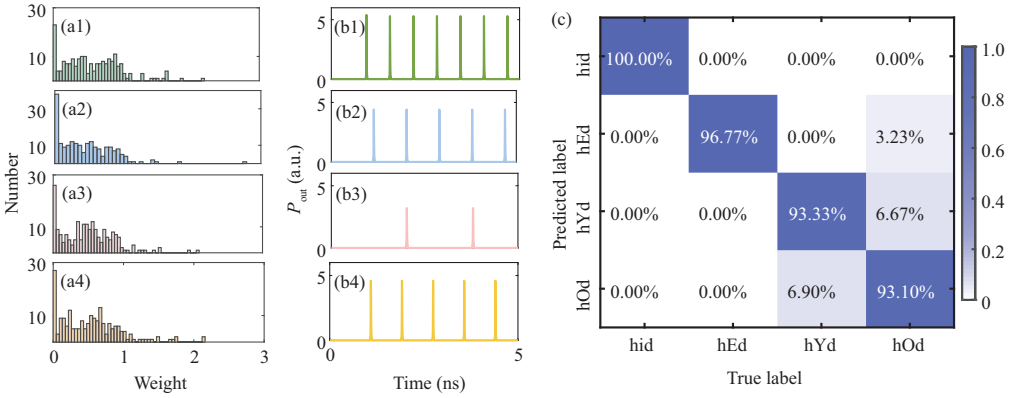

Figure 8 (Color online) (a1)–(a4) Distribution of all 200 coupling weights to each of the four optical output neurons; (b1)–(b4) outputs of 4 output photonic neurons as examples; (c) confusion matrix of the converted result.

To obtain statistical results, we repeat the training for 6 times. The dataset distribution and initial parameters are the same as those adopted in Figure 7(b), and the corresponding converted results with different κ are shown in Figure 9(a). An acceptable conversion loss may occur when $\kappa = 0.01$, most possibly due to the deviation between the calculated spike rate and the fitted function. In addition to this reason, the converted error could also be caused by the bit precision of data and weight during conversion, the noise, and the mismatch of device parameters.

The bit precision of the input signal and weight is a key factor to be considered in hardware implementation. We calculate the converted accuracy with different bit precision of data and weight, and the results are presented in Figure 9(b). We can see that with 1 bit precision of data and weight, the converted accuracy is relatively low and is much more unstable. However, when the data precision is greater than 1 bit the mean converted test accuracy remains stable. Additionally, the accuracy increases when the weight precision is more than 1 bit and remains constant when the weight accuracy is greater than 3 bit. The result indicates that in theory only 2 bit accuracy is required to achieve decent performance on hardware.

Moreover, we also consider the impact of noise on data and weight during conversion. The noise is directly introduced to the normalized data $\hat{x}$. Since the input data is firstly normalized into the range of 0.01 to 1, we consider a noise level from 0.1 to 0.6. In Figure 9(c), we present the conversion result with different noise levels. The converted accuracy remains above 90% when the noise level is less than 0.3,

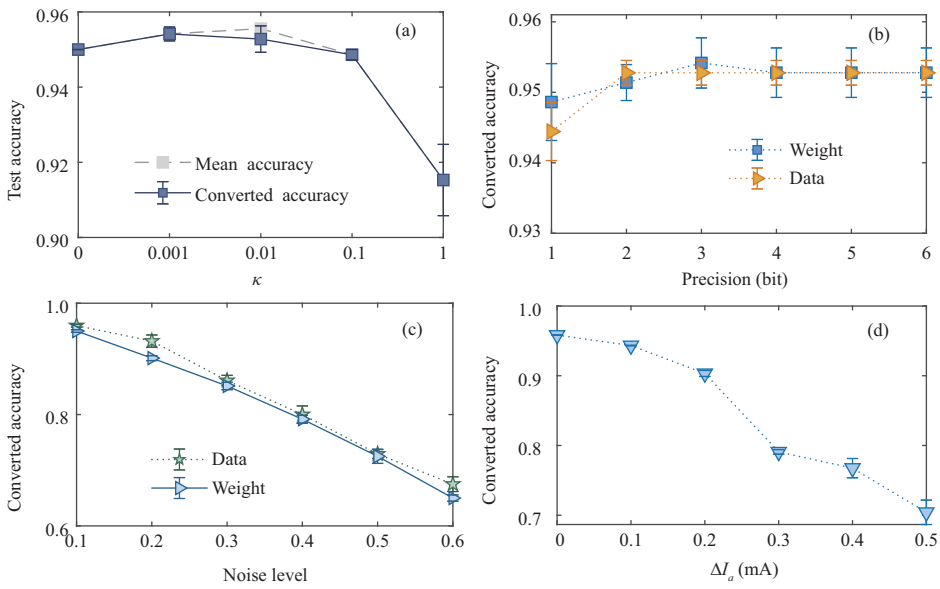

Figure 9 (Color online) (a) Converted results with different κ corresponding with those in Figure 7(b); (b) mean test accuracy after conversion as a function of data precision and the weight precision of 6 trials; (c) mean test accuracy after conversion as a function of the noise level of input data and weight; (d) mean test accuracy after conversion as a function of current mismatch of 10 trials.

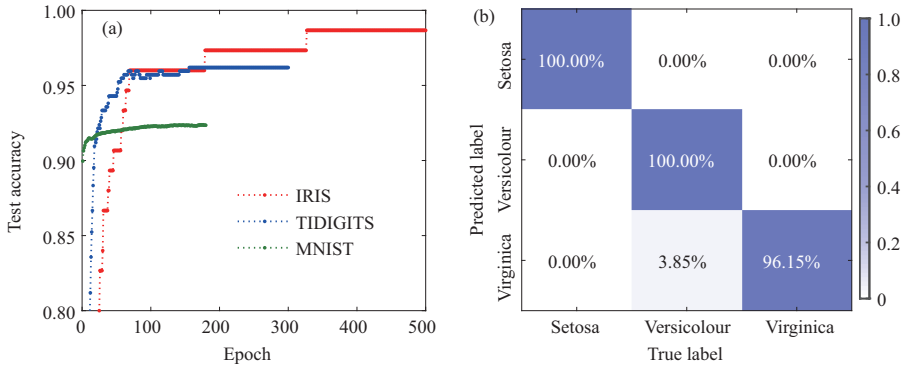

Figure 10 (Color online) (a) The test accuracy trained with optical activation function for IRIS dataset (red line), TIDIGITS (blue line), and MNIST (green line); (b) the converted test accuracy of IRIS dataset, with a mean accuracy of 98.67%.

and then decreases almost linearly with the increase of noise strength.

Parameter mismatch of devices is also critical to the performance of hardware systems. We further consider the effect of bias current I_a mismatch of different photonic neurons on network performance. The results of 10 repeated trials are shown in Figure 9(d). It can be seen that the mean converted accuracy can be above 90% when the current mismatch Δ_a is less than 0.2 mA, but drops quickly as Δ_a further increases.

Based on training of ANN, this method is naturally applicable to other classification tasks. We also evaluated the method through tests on the IRIS dataset (Fisher, 1936), TIDIGITS [43], and MNIST[1)] dataset. The IRIS dataset is a simple three-category dataset consisting of 150 samples, each of which has 4 features. To improve classification precision, we also use Gaussian coding with $N_f = 10$. Half of the data is randomly chosen for training and the rest for testing. Figure 10(a) shows the test accuracy of the IRIS dataset trained with the optical activation function (red line), and the converted confusion matrix is given in Figure 10(b). The mean accuracy is 98.67%, which is the same as that trained via the optical activation function, indicating that there is no conversion loss. The TIDIGITS dataset we used consists of speech digits (0–9), 3480 samples are used for training, and 420 samples are used for testing. We extract 41×40 filter-bank-based mel-frequency cepstral coefficients (MFCCs) [44] and the dimension is reduced to 95 via principal component analysis. The mean converted accuracy is 96.19%, also consistent

1) MNIST. http://yann.lecun.com/exdb/mnist/.

with the trained result, as shown by the solid blue line in Figure 10(a). As for the MNIST dataset, we use all 60000 samples for training and 10000 samples for test. The three-channel image is firstly converted to grayscale, and then all the pixel values are subtracted from 255 to encode the information into larger pixel values. Then, the 28×28 pixels are directly used for training. The result is also present in Figure 10(a) (green line), and the converted accuracy is about 92.33%. Note that the accuracy of MNIST is relatively low, which is limited via the single-layer network structure. Usually, a two-layer network is adopted for the MNIST dataset to achieve high performance, which will be the focus of further consideration.

4 Conclusion

We proposed a method for converting a single-layer ANN trained with an optical activation function into a photonic SNN, aiming to enhance the learning precision of photonic SNNs. The effects of the bit precision and the noise and parameter mismatch on the converted results were explored to verify the feasibility of hardware implementation. We adopted the Deterding vowel, IRIS, TIDIGITS, and MNIST datasets to evaluate the performance of the conversion scheme, and the results demonstrated that the trained network could be converted into a photonic SNN with almost no precision loss.

However, only a single-layer network can be converted, because the relation between the input power and spike rate of a photonic neuron only holds for the output layer and inputs of analog power instead of spike trains. Owing to the numerical complexity of the photonic neuron model, deriving the layer-to-layer rate response model, as realized in the case of IF neurons is difficult. Nonetheless, the converted method exhibited comparable performance to an ANN trained with ReLU in relatively complex tasks, demonstrating its potential as a high-performance scheme for the hardware implementation of photonic SNNs.

Acknowledgements This work was supported in part by National Key Research and Development Program of China (Grant Nos. 2021YFB2801900, 2021YFB2801901, 2021YFB2801902, 2021YFB2801904), National Natural Science Foundation of China (Grant Nos. 61974177, 61674119), National Outstanding Youth Science Fund Project of National Natural Science Foundation of China (Grant No. 62022062), and Fundamental Research Funds for the Central Universities (Grant No. JB210114).

References

1 Schuman C D, Potok T E, Patton R M, et al. A survey of neuromorphic computing and neural networks in hardware. 2017. ArXiv:1705.06963
2 Zhou S B, Li X H, Chen Y, et al. Temporal-coded deep spiking neural network with easy training and robust performance. In: Proceedings of Association for the Advancement of Artificial Intelligence Conference on Artificial Intelligence, Palo Alto, 2021. 11143–11151
3 Tavanaei A, Maida A. BP-STDP: approximating backpropagation using spike timing dependent plasticity. Neurocomputing, 2019, 330: 39–47
4 Wu Y J, Deng L, Li G Q, et al. Spatio-temporal backpropagation for training high-performance spiking neural networks. Front Neurosci, 2018, 12: 331
5 Bohté S M, Kok J N, Poutré H L. Error-backpropagation in temporally encoded networks of spiking neurons. Neurocomputing, 2002, 48: 17–37
6 Wu Y J, Deng L, Li G Q, et al. Direct training for spiking neural networks: faster, larger, better. In: Proceedings of Association for the Advancement of Artificial Intelligence, 2019. 1311–1318
7 Xiang S Y, Jiang S Q, Liu X S, et al. Spiking VGG7: deep convolutional spiking neural network with direct training for object recognition. Electronics, 2022, 11: 2097
8 Ponulak F, Kasiński A. Supervised learning in spiking neural networks with ReSuMe: sequence learning, classification, and spike shifting. Neural Computation, 2010, 22: 467–510
9 Gütig R, Sompolinsky H. The Tempotron: a neuron that learns spike timing-based decisions. Nat Neurosci, 2006, 9: 420–428
10 Han Y N, Xiang S Y, Ren Z X, et al. Delay-weight plasticity-based supervised learning in optical spiking neural networks. Photon Res, 2021, 9: B119
11 Liu F X, Zhao W B, Chen Y B, et al. SSTDP: supervised spike timing dependent plasticity for efficient spiking neural network training. Front Neurosci, 2021, 15: 756876
12 O'Connor P, Neil D, Liu S C, et al. Real-time classification and sensor fusion with a spiking deep belief network. Front Neurosci, 2013, 7: 178
13 Cao Y Q, Chen Y, Khosla D. Spiking deep convolutional neural networks for energy-efficient object recognition. Int J Comput Vis, 2015, 113: 54–66
14 Rueckauer B, Lungu I A, Hu Y H, et al. Conversion of continuous-valued deep networks to efficient event-driven networks for image classification. Front Neurosci, 2017, 11: 682
15 Ding J H, Yu Z F, Tian Y H, et al. Optimal ANN-SNN conversion for fast and accurate inference in deep spiking neural networks. In: Proceedings of the 30th International Joint Conference on Artificial Intelligence Main Track, 2021. 2328–2336
16 Woods D, Naughton T J. Photonic neural networks. Nat Phys, 2012, 8: 257–259
17 Yao P, Wu H Q, Gao B, et al. Face classification using electronic synapses. Nat Commun, 2017, 8: 15199
18 Xiang S Y, Gong J K, Zhang Y H, et al. Numerical implementation of wavelength-dependent Photonic spike timing dependent plasticity based on VCSOA. IEEE J Quantum Electron, 2018, 54: 1–7
19 Boybat I, Gallo M L, Nandakumar S R, et al. Neuromorphic computing with multi-memristive synapses. Nat Commun, 2018, 9: 2514

20 Midya R, Wang Z R, Asapu S, et al. Artificial neural network (ANN) to spiking neural network (SNN) converters based on diffusive memristors. Adv Electron Mater, 2019, 5: 1900060
21 Feldmann J, Youngblood N, Wright C D, et al. All-optical spiking neurosynaptic networks with self-learning capabilities. Nature, 2019, 569: 208–214
22 Zhang J Y, Dai S L, Zhao Y W, et al. Recent progress in photonic synapses for neuromorphic systems. Adv Intell Syst, 2020, 2: 1900136
23 Xiang S Y, Han Y N, Song Z W, et al. A review: photonics devices, architectures, and algorithms for optical neural computing. J Semicond, 2021, 42: 023105
24 Xiang J L, Zhang Y J, Zhao Y T, et al. All-optical silicon microring spiking neuron. Photon Res, 2022, 10: 939
25 Xiang S Y, Shi Y C, Guo X X, et al. Hardware-algorithm collaborative computing with photonic spiking neuron chip based on an integrated Fabry-Perot laser with a saturable absorber. Optica, 2023, 10: 162–171
26 Shen Y C, Harris N C, Skirlo S, et al. Deep learning with coherent nanophotonic circuits. Nat Photon, 2017, 11: 441–446
27 Zhao X M, Lv H B, Chen C, et al. On-chip reconfigurable optical neural networks. Research Square, 2021. doi: 10.21203/rs.3.rs-155560/v1
28 Xiang S Y, Zhang H, Guo X X, et al. Cascadable neuron-like spiking dynamics in coupled VCSELs subject to orthogonally polarized optical pulse injection. IEEE J Sel Top Quantum Electron, 2017, 23: 1–7
29 Deng T, Robertson J, Hurtado A. Controlled propagation of spiking dynamics in vertical-cavity surface-emitting lasers: towards neuromorphic photonic networks. IEEE J Sel Top Quantum Electron, 2017, 23: 1–8
30 Robertson J, Wade E, Kopp Y, et al. Toward neuromorphic photonic networks of ultrafast spiking laser neurons. IEEE J Sel Top Quantum Electron, 2020, 26: 1–15
31 Pammi V A, Alfaro-Bittner K, Clerc M G, et al. Photonic computing with single and coupled spiking micropillar lasers. IEEE J Sel Top Quantum Electron, 2020, 26: 1–7
32 Chlouverakis K E, Adams M J. Two-section semiconductor lasers subject to optical injection. IEEE J Sel Top Quantum Electron, 2004, 10: 982–990
33 Nahmias M A, Shastri B J, Tait A N, et al. A leaky integrate-and-fire laser neuron for ultrafast cognitive computing. IEEE J Sel Top Quantum Electron, 2013, 19: 1–12
34 Dubbeldam J L A, Krauskopf B. Self-pulsations of lasers with saturable absorber: dynamics and bifurcations. Optics Commun, 1999, 159: 325–338
35 Han Y N, Xiang S Y, Zhang Y N, et al. An all-MRR-based photonic spiking neural network for spike sequence learning. Photonics, 2022, 9: 120
36 Xiang J L, Torchy A, Guo X H, et al. All-optical spiking neuron based on passive microresonator. J Lightwave Technol, 2020, 38: 4019–4029
37 Chakraborty I, Saha G, Sengupta A, et al. Toward fast neural computing using all-photonic phase change spiking neurons. Sci Rep, 2018, 8: 12980
38 Xiang S Y, Zhang Y H, Gong J K, et al. STDP-based unsupervised spike pattern learning in a photonic spiking neural network with VCSELs and VCSOAs. IEEE J Sel Top Quantum Electron, 2019, 25: 1–9
39 Xiang S Y, Ren Z X, Song Z W, et al. Computing primitive of fully VCSEL-based all-optical spiking neural network for supervised learning and pattern classification. IEEE Trans Neural Netw Learn Syst, 2021, 32: 2494–2505
40 Fu C T, Xiang S Y, Han Y N, et al. Multilayer photonic spiking neural networks: generalized supervised learning algorithm and network optimization. Photonics, 2022, 9: 217
41 Han Y N, Xiang S Y, Song Z W, et al. Spiking dynamics and synchronization properties of optical neurons based on VCSEL-SAs. Nonlinear Dyn, 2021, 105: 2665–2675
42 Deterding D H. Speaker normalisation for automatic speech recognition. Dissertation for Ph.D. Degree. Cambridge: University of Cambridge, 1990
43 Leonard R G. A database for speaker-independent digit recognition. In: Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing, San Diego, 1984. 328–331
44 Davis S, Mermelstein P. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. IEEE Trans Acoust Speech Signal Process, 1980, 28: 357–366