ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 978-1-6654-0540-9/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICASSP43922.2022.9746792

A HYBRID LEARNING FRAMEWORK FOR DEEP SPIKING NEURAL NETWORKS WITH ONE-SPIKE TEMPORAL CODING

Jiadong Wang[1], *Jibin Wu*[1†], *Malu Zhang*[1], *Qi Liu*[1], *Haizhou Li*[1,2]

[1]National University of Singapore, Singapore [2]The Chinese University of Hong Kong, Shenzhen, China

ABSTRACT

Bio-inspired spiking neural networks (SNNs) are compelling candidates for spatio-temporal information processing on ultra-low power neuromorphic computing chips. However, the existing SNN training methods have not fully exploited the temporal information of spikes that plays a critical role in sparse information representation and communication. Hereby, we present a hybrid learning framework for deep SNNs with one-spike temporal coding to make full utilization of the spike timing. We first propose a novel ANN-to-SNN conversion method based on forward propagation mechanisms of ANNs and SNNs to offer a good initialization for SNNs. The performance of the converted SNN is further improved by training with a timing-based backpropagation (BP) method. Experimental results demonstrate that the proposed hybrid learning framework can achieve competitive accuracies on both visual and audio recognition tasks with significantly improved training efficiency over direct SNN BP methods.

Index Terms— Hybrid Learning, Spiking Neural Networks, Temporal Coding

1. INTRODUCTION

We have witnessed the success of conventional artificial neural networks (ANNs) in many pattern classification tasks. However, it remains a challenge to deploy ANNs in low-resource edge computing platforms. Unlike the ANNs, the spiking neural networks (SNNs) compute and communicate information through discrete spikes that can significantly reduce energy consumption on neuromorphic hardware. While theoretical evidences suggest that SNNs have many computational advantages over ANNs, the SNNs have yet to match the accuracy of their ANNs counterparts in pattern classification tasks. This is mainly due to their complex spatio-temporal dynamics and non-differentiable spiking activation function.

This work is supported by (1) A*STAR under its RIE2020 Advanced Manufacturing and Engineering Domain (AME) Programmatic Grant (Grant No. A1687b0033, Project Title: Spiking Neural Networks) (2)IAF, A*STAR, SOITEC, NXP and National University of Singapore under FD-fAbrICS: Joint Lab for FD-SOI Always-on Intelligent Connected Systems (Award I2001E0053).

† corresponding author

The development of efficient learning algorithms for deep SNNs is an on-going research challenge. One successful strategy is to convert a pre-trained ANN to SNN, namely ANN-to-SNN conversion [1, 2, 3, 4, 5], to benefit from effective ANN training and run-time efficiency of SNN. With a well-balanced neuronal firing threshold and weights, ANN-to-SNN conversion methods show good performance even on large-scale benchmarks [6]. As most of the conversion methods are for rate-based spiking neurons, they don't utilize spike-timing information. This inevitably leads to high energy consumption and inference latency. Temporal coding represents an energy efficient alternative to rate-based coding. However, there have been few studies on ANN-to-SNN conversion for SNN with temporal coding [7]. Existing studies are either showing low accuracy [8] or high inference latency [9]. One of the particular challenges is the discrepancy of neuronal functions and communications between analog neurons and spiking neurons. An effective ANN-to-SNN conversion has to compensate for such discrepancy.

Error back-propagation is the training technique for deep network architecture. Two types of back-propagation methods have been studied that allow direct optimization of SNN parameters. One is to compute the gradients with respect to the change in the membrane potential, wherein surrogate gradients are employed to overcome the discontinuity at the spike generation. This technique involves spatial-temporal credit assignment that requires more memory and training time than the ANN-to-SNN conversion methods. Another line of thought is to compute the gradients with respect to the change in spike timing [10, 11, 12, 13, 14]. These timing-based methods train SNNs in an event-driven manner and are compatible with temporal coding. Therefore, they hold a great potential for neuromorphic computing. Nevertheless, the performance of existing spike timing-based methods is still limited and generally confined to shallow networks on small-scale benchmarks.

Hybrid learning is an idea to leverage both the ANN-to-SNN conversion strategy, and the SNN-enabled back-propagation to embrace the best of two worlds [15, 16, 17, 18, 19]. For example, the activation- and timing-based learning rule (ANTLR) [17] tries to make use of each individual spike more effectively. However, the prior studies are yet to take full advantage of temporal coding. The memory and

Authorized licensed use limited to: IEEE Xplore. Downloaded on May 03,2022 at 07:00:56 UTC from IEEE Xplore. Restrictions apply.

computation efficiency, convergence issue on large-scale network architectures, such as VGG and ResNet, remain the topics of study for the training process.

In this work, we study a novel hybrid learning framework for SNNs with temporal coding. The learning framework takes place in two stages. At the first stage, a novel ANN-to-SNN conversion method is proposed to bridge the disparity between ANN and temporal-encoded SNN. Then, a state-of-the-art SNN BP method [14] is applied to fine-tune the network, so as to recover the discrepancy between the computational dynamics of ANNs and SNNs. We validate our method on both audio and visual recognition tasks.

2. HYBRID LEARNING FRAMEWORK

Suppose $\{X, Y\}$ is an input-label pair. We would like to train a SNN to predict Y given X, i.e., $Y = f_s(\text{Enc(X)}; \theta_s)$, where $\text{Enc}(\cdot)$ denotes the temporal encoding and θ_s are learnable parameters of the SNN.

As shown Fig.1, the proposed hybrid learning containstwo stages. At the first stage, we train an ANN, $Y = f_a(X; \theta_a)$ to classify X into Y, and the ANN employs the same network structure with that of the SNN. We then use θ_a to initialize θ_s, and fine-tune the SNN using a timing-based error backpropagation algorithm at the second stage.

2.1. Analog neuron vs spiking neuron

The analog neuronal function, which has been used in the ANN, can be defined as follows,

$$u_j^l = \sum_i^N \omega_{ij}^l h_i^{l-1}, \quad h_j^l = ReLU(u_j^l) \tag{1}$$

where $l \in [1, 2, ..., L]$ denotes the layer, $j \in [1, 2, ..., J]$ denotes a neuron in the current layer, while $i \in [1, 2, ..., N]$ denotes a neuron in the previous layer which connects to j or an input index if the current layer is the first layer.

For spiking neurons in the SNN, we adopt the Rectified Linear Postsynaptic Potential (ReL-PSP), that seeks to address the exploding gradients and dead neurons problems arising from temporal coding [14], as the neuronal function.

$$V_j^l(t) = \sum_i^N \omega_{ij}^l K(t - t_i^{l-1}) \tag{2}$$

$$t_j^l = \mathcal{F}\left\{t | V_j^l(t) = \vartheta, t \geq 0\right\} \tag{3}$$

where $V_j^l(t)$ is the membrane potential of j-th neuron O_j^l in the l-th layer at time t, ω^l is the learnable weight connecting layer l and $l-1$. O_j^l fires a spike once $V_j^l(t)$ cross the firing threshold ϑ, and the spike time is recorded as t_j^l. $K(\cdot)$ is the PSP kernel function, which is defined as

$$K(t - t_i^{l-1}) = \begin{cases} t - t_i^{l-1} & \text{if} \quad t > t_i^{l-1} \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

2.2. ANN-to-SNN conversion

For SNN, we adopt temporal coding. As shown in Fig.1, we first preprocess and normalize input X into I that fit within the range $\in [0, 1]$. We further encode the normalized inputs I with time-to-first-spike encoding. This temporal encoding mechanism considers a sensory neuron that stimulated with a stronger stimulus triggers an earlier input spike. This behaviour can be simplified and described as $t^0 = 1 - I$, as such the input spikes are generated within the time window $[0, 1]$. After this temporal encoding layer, the information is represented and processed in the spike domain.

To ensure a mathematical equivalence between the activation of an analog neuron and a spiking neuron in Eq.8, we impose two constraints during the ANN training process, that are activation constraint and weight sum constraint.

Activation constraint: to limit the activation of an analog neuron h^l into range of [0, 1] (counterpart of t^l in a spiking neuron). With the $ReLU$ function, we only need to make sure $u^l \leq 1$ to keep h^l within the range.

Weight sum constraint: to ensure sums of weights to one, satisfy $\sum_i^N \omega_{ij} = 1$.

Having the two constraints satisfied and ϑ set to 1, it's of high probability that $t_i^l > t_j^{l-1}$. We define this inequality as **full contribution property**. Let's take the first layer as an example,

$$V_j^1(1) = \sum_i^N \omega_{ij}^1 (1 - t_i^0) = \sum_i^N \omega_{ij} I_i = u_j^1 < 1 = \vartheta \tag{5}$$

where O_j^1 has yet to fire a spike when t=1s. As the PSP function (Eq.4) is a monotonic increasing function, V_j^1 will keep increasing over time, and a spike is eventually fired after 1s ($> t^0$), i.e. $t^1 > t^0$.

Now, we can analytically infer t^1 and find mathematical relationship between u^1 and t^1 as Eq.7.

$$1 = t_j^1 \sum_i^N \omega_{ij}^1 - \sum_i^N \omega_{ij}^1 (1 - I_i) \tag{6}$$

$$t_j^1 = 2 - \sum_i^N \omega_{ij}^1 I_i = 2 - u_j^1 \tag{7}$$

Force firing: To further establish the mathematical relationship between t^1 and h^1, We apply a force firing process as $\mathcal{FF}$ in Eq.8, which forces O^1 to fire the latest by 2s. Similarly and recursively, t^2 will locate at the range $[2, 3]$ and then t^3 will be in the range $[3, 4]$, etc.

$$\mathcal{FF}(t_j^1) = \textbf{min}(t_j^1, 2) = 2 - \textbf{ReLU}(u_j^1) = 2 - h_j^1 \tag{8}$$

The total objective function $\mathcal{L}$ is composed of three items,

$$\mathcal{L} = \mathcal{L}_{ce} + \lambda_w \mathcal{L}_{wsum} + \lambda_a \mathcal{L}_{acti} \tag{9}$$

where $\mathcal{L}_{ce}$ is the traditional Cross-Entropy (CE) loss to draw $f_a(X; \theta_a)$ close to Y, λ_w and λ_a are the contributing factors for $\mathcal{L}_{wsum}$ and $\mathcal{L}_{acti}$, which are designed to impose the weight sum constraint and the activation constraint.

Authorized licensed use limited to: IEEE Xplore. Downloaded on May 03,2022 at 07:00:56 UTC from IEEE Xplore. Restrictions apply.

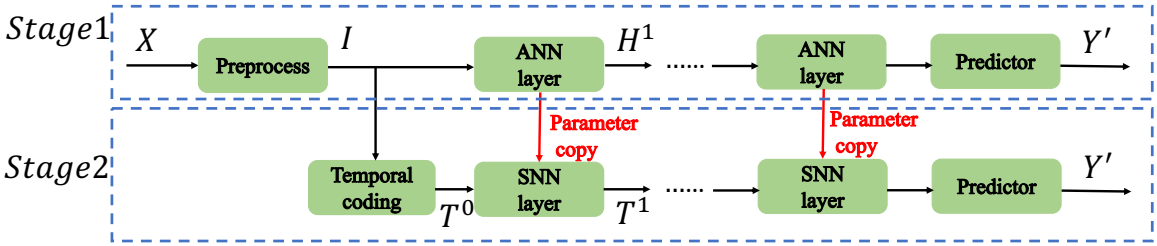

Fig. 1. Illustration of the proposed hybrid learning framework.

$$\mathcal{L}_{wsum} = \sum_{l}^{L}\sum_{j}^{J}|\sum_{i}^{N}\omega_{ij}^{l} - 1| \tag{10}$$

$$\mathcal{L}_{acti} = \sum_{l}^{L}\sum_{j}^{J}|u_j^l - u_j'^l| \tag{11}$$

where $|\cdot|$ is an $L1$ loss and $\mathbf{U}'^l \sim \mathcal{N}(0, 1/9)$ is a vector or matrix normalized from $\mathbf{U}^l$. Thus, 99.73% $u_j'^l$ are located at $[-1, 1]$ according to statistical property.

2.3. SNN error back-propagation

With the weight sum constraint, activation constraint, and full contribution property, we expect that the neuronal behaviour of a spiking neuron in SNN is similar to that of an analog neuron in ANN. However, these constraints and property can't be prefectly satisfied during the ANN-to-SNN conversion. Even if they are perfectly satisfied, there still exist discrepancies between analog and spiking neurons in terms of neuronal functions and communications. We further employ the STDBP algorithm with surrogate gradients [14] to perform error back-propagation at the second stage, to recover the discrepancy between ANN and SNN.

3. EXPERIMENTS

We conduct experiments on both audio and visual tasks: (a) Image classification (b) Keyword spotting. We consider STDBP learning algorithm [14] as our baseline which achieves the state-of-the-art accuracy with temporal-coding SNNs on MNIST [20], Fashion-MNIST [21], and Caltech101 face/motorbike datasets. The difference between our method and the baseline is that STDBP [14] only pretrain ANN with $\mathcal{L}_{ce}$ loss to initialize the SNN model.

3.1. Image classification

In this section, we first evaluate the effectiveness of $\mathcal{L}_{wsum}$ and $\mathcal{L}_{acti}$ in satisfying the weight sum constraint and the activation constraint on the **MNIST** [20] dataset. Furthermore, we compare the proposed hybrid learning framework to the baseline STDBP algorithm in terms of the training speed and recognition accuracy. We conduct experiments with two network structures as follows:

1) **Fully Connected Network (FCN)** Here, we employ two fully-connected layers with 800 and 10 neurons, respectively (FC2: 784-800-10). We provide the distribution of ANN hidden layer activation values on the test set in Fig. 2, where the mean and standard deviation (std) take a value of 2.3e^{-3} and 0.3075 respectively, and 99.26% of the activation values are located within the range of $[-1, 1]$. It hence suggests the $\mathcal{L}_{acti}$ loss term is capable of satisfying the activation constraint. Besides, we also observe that the 800 sums of weights $\sum_i^N \omega_{ij}$ for 800 hidden neurons (O_j^1) generally fits well into the range of $[0.99, 1.01]$ which suggests that the $\mathcal{L}_{wsum}$ loss term takes effect.

2) **Convolutional Neural Network (CNN)** Here, we employ a CNN with 2 convolutional layers followed by 3 fully-connected layers (C2F3: 28*28-16C5-P2-32C5-P2-800-128-10), where 16C5 denotes a convolutional layer with 16 filters and kernel size of each filter is 5×5. P2 denotes a max pooling layer in ANN (min pooling in SNN) with non-overlapping kernels that has a size of 2×2. In practice, we find that the activation constraint and the weight sum constraint are hard to be satisfied simultaneously on the first convolutional layer due to the limited entries in the kernel (1×5×5). Hence, we keep the first convolutional layer to work in the analog domain as in the ANN, which can be absorbed into the preprocess step that generates I as shown in Fig 1. We provide the maximum and minimum values of sums of weights, and the percentage of activation values that satisfying the constraint in the Table 1. It is obvious that both constraints are well satisfied for all the layers.

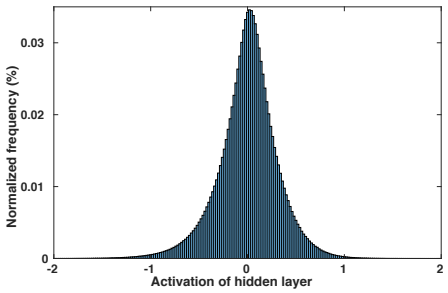

Fig. 2. The distribution of hidden layer activation values.

We compare the classification accuracy of our model to other state-of-the-art temporal-coding SNNs in Table.2. It is obvious that performance gaps between $f_s(\cdot;\theta_a)$ and $f_a(\cdot;\theta_a)$ are small on both FC3 and C2F3 architectures. Notably, the accuracy drop is only 0.2% on the C2F3 with the force firing module. Meanwhile, our method achieves the same accuracy

Authorized licensed use limited to: IEEE Xplore. Downloaded on May 03,2022 at 07:00:56 UTC from IEEE Xplore. Restrictions apply.

Table 1. Maximum and minimum of weight sums and proportion of satisfying activation constraint of different layers.

MNIST layer	32C5	800	128	10
Maximum	**1.01**	**1.01**	**1.01**	**1**
Minimum	**0.99**	**0.99**	**0.99**	**1**
$h^l < 1$	**99.32%**	**99.97%**	**99.32%**	**-**
Cifar10 layer	128C3	128C3	256	10
Maximum	**1.09**	**1.07**	**1.17**	**1.01**
Minimum	**0.92**	**0.93**	**0.82**	**0.98**
$h^l < 1$	**99.53%**	**98.89%**	**84.75%**	**-**

Table 2. The classification accuracy of different temporal-coding SNN algorithms on the MNIST dataset. FF: force firing. The entries in the column of "Init acc." indicates performance of $f_s(\cdot;\theta_a)[f_a(\cdot;\theta_a)]$

Model	Architecture	Init acc. (%)	Final (%)
Comsa[12]	784-340-10	-	97.9
Masquelier[22]	784-400-10	-	97.4
Mostafa[11]	FC2	-	97.5
STDBP[14]	FC2	-	98.5
Ours (FF)	FC2	94.7 [98.3]	98.5
Ours (w/o FF)	FC2	94.7 [98.3]	98.5
STDBP[14]	C2F3	71.6 [99.6]	**99.4**
Ours (FF)	C2F3	**99.3** [99.5]	**99.4**
Ours (w/o FF)	C2F3	98.3 [99.5]	**99.4**

as the state-of-the-art STDBP algorithm on the C2F3 architecture. Moreover, as demonstrated in Fig.3, our second stage SNN training converges within less than 5 epochs, which is much faster than the STDBP algorithm.

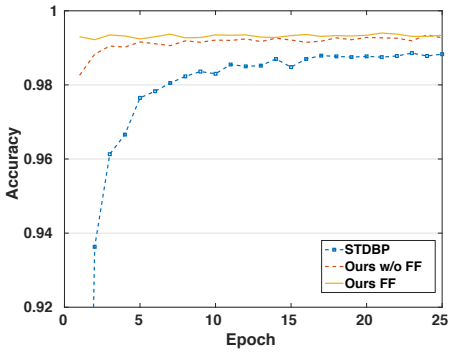

Fig. 3. Training speed comparison

We further evaluate our method on the more challenging **Cifar10** [23] dataset with a CNN (64C3-P2-128C3-128C3-P2-256-10). As shown in Table 1, compared to the ANN trained on the MNIST dataset, it is more challenging to satisfy both constraints on the Cifar10 dataset, especially when layers are close to the output. It can be explained by the higher classification difficulty on the Cifar10, resulting in ANN to has limited room to adjust its parameters to simultaneously satisfying both constraints and achieving high accuracy. Nevertheless, the constraints only slightly deviate from expectations. As shown in Table. 3, the gap between $f_s(\cdot;\theta_a)$ and $f_a(\cdot;\theta_a)$ is small (accuracy drop by less than 5% with force firing), suggesting the impact of such deviation is limited. Moreover, our method outperforms other methods even without performing the fine-tuning with SNN BP, especially improved by 35.7% over the STDBP algorithm [14]. This can be largely credit to the proposed conversion method.

Table 3. The classification accuracy of different temporal-coding SNN algorithms on the Cifar10 dataset. For fair comparison, we use the same network architecture across all the experiments. FF: force firing.

Model	Method	Init. acc. (%)	Final (%)
Sengupta et al.[2]	conversion	-	76.8
Deng et al.[24]	SNN BP	-	74.2
STDBP[14]	SNN BP	11.8 [86.1]	45.9
Ours (FF)	Hybrid	**81.3** [85.6]	**81.6**
Ours (w/o FF)	Hybrid	76.3 [85.6]	77.2

Table 4. Comparison on the keyword spotting task. FF: force firing

Model	Coding	Init. acc. (%)	Final(%)
Yilmaz et al.[26]	Rate	-	75.2
NLIF[27]	Rate	-	87.9
STDBP[14]	Temporal	57.7 [88.2]	75.9
Ours (FF)	Temporal	**84.3** [91.0]	**88.5**
Ours (w/o FF)	Temporal	83.4 [91.0]	87.6

3.2. Keyword spotting

To further validate the applicability of the proposed hybrid learning framework to other data modality, we performed a keyword spotting task on the Google Speech Command dataset [25]. Following the same experimental set-up and network architecture in [26], we classify the input audio samples into 12 classes. As shown in Table 4, the proposed hybrid learning framework outperforms the STDBP algorithm by 12.6%, which validates the idea of applying ANN-to-SNN conversion as the first stage. Moreover, the proposed learning framework also outperform other two rate-based baselines by at least 0.6%.

4. CONCLUSION

We have proposed a hybrid learning framework for SNN with temporal coding. Experiments show that the proposed learning framework is effective in audio and visual classification tasks. Moreover, the SNN BP method can quickly recover the accuracy drop with less than 5 training epochs. Notably, in many cases, the performance after network conversion is already very competitive without further fine-tuning. The learning framework opens up opportunities for actual implementation of spiking neural network on neuromorphic chips.

Authorized licensed use limited to: IEEE Xplore. Downloaded on May 03,2022 at 07:00:56 UTC from IEEE Xplore. Restrictions apply.

5. REFERENCES

[1] Peter U Diehl, Daniel Neil, Jonathan Binas, Matthew Cook, Shih-Chii Liu, and Michael Pfeiffer. Fast-classifying, high-accuracy spiking deep networks through weight and threshold balancing. In *2015 International joint conference on neural networks (IJCNN)*, pages 1–8. ieee, 2015.

[2] Abhronil Sengupta, Yuting Ye, Robert Wang, Chiao Liu, and Kaushik Roy. Going deeper in spiking neural networks: Vgg and residual architectures. *Frontiers in neuroscience*, 13:95, 2019.

[3] Bing Han, Gopalakrishnan Srinivasan, and Kaushik Roy. Rmp-snn: Residual membrane potential neuron for enabling deeper high-accuracy and low-latency spiking neural network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13558–13567, 2020.

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

[5] Shikuang Deng and Shi Gu. Optimal conversion of conventional artificial neural networks to spiking neural networks. 2021.

[6] Youngeun Kim and Priyadarshini Panda. Visual explanations from spiking neural networks using interspike intervals. *arXiv preprint arXiv:2103.14441*, 2021.

[7] Christoph Stöckl and Wolfgang Maass. Optimized spiking neurons can classify images with high accuracy through temporal coding with two spikes. *Nature Machine Intelligence*, pages 1–9, 2021.

[8] Bodo Rueckauer and Shih-Chii Liu. Conversion of analog to spiking neural networks using sparse temporal coding. In *2018 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1–5. IEEE, 2018.

[9] Lei Zhang, Shengyuan Zhou, Tian Zhi, Zidong Du, and Yunji Chen. Tdsnn: From deep neural networks to deep spike neural networks with temporal-coding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 1319–1326, 2019.

[10] Sander M Bohte, Joost N Kok, and Han La Poutre. Error-backpropagation in temporally encoded networks of spiking neurons. *Neurocomputing*, 48(1-4):17–37, 2002.

[11] Hesham Mostafa. Supervised learning based on temporal coding in spiking neural networks. *IEEE transactions on neural networks and learning systems*, 29(7):3227–3235, 2017.

[12] Iulia M Comsa, Krzysztof Potempa, Luca Versari, Thomas Fischbacher, Andrea Gesmundo, and Jyrki Alakuijala. Temporal coding in spiking neural networks with alpha synaptic function. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8529–8533. IEEE, 2020.

[13] Saeed Reza Kheradpisheh and Timothée Masquelier. Temporal backpropagation for spiking neural networks with one spike per neuron. *International Journal of Neural Systems*, 30(06):2050027, 2020.

[14] Malu Zhang, Jiadong Wang, Jibin Wu, Ammar Belatreche, Burin Amornpaisannon, Zhixuan Zhang, Venkata Pavan Kumar Miriyala, Hong Qu, Yansong Chua, Trevor E Carlson, et al. Rectified linear postsynaptic potential function for backpropagation in deep spiking neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

[15] Nitin Rathi, Gopalakrishnan Srinivasan, Priyadarshini Panda, and Kaushik Roy. Enabling deep spiking neural networks with hybrid conversion and spike timing dependent backpropagation. In *International Conference on Learning Representations*, 2019.

[16] Nitin Rathi and Kaushik Roy. Diet-snn: Direct input encoding with leakage and threshold optimization in deep spiking neural networks. *arXiv preprint arXiv:2008.03658*, 2020.

[17] Jinseok Kim, Kyungsu Kim, and Jae-Joon Kim. Unifying activation-and timing-based learning rules for spiking neural networks. *Advances in Neural Information Processing Systems*, 33, 2020.

[18] Jibin Wu, Chenglin Xu, Xiao Han, Daquan Zhou, Malu Zhang, Haizhou Li, and Kay Chen Tan. Progressive tandem learning for pattern recognition with deep spiking neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2021.

[19] Jibin Wu, Yansong Chua, Malu Zhang, Guoqi Li, Haizhou Li, and Kay Chen Tan. A tandem learning rule for effective training and rapid inference of deep spiking neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–15, 2021.

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

[21] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

[22] Saeed Reza Kheradpisheh, Maryam Mirsadeghi, and Timothée Masquelier. Bs4nn: Binarized spiking neural networks with temporal coding and learning. *arXiv preprint arXiv:2007.04039*, 2020.

[23] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[24] Lei Deng, Yujie Wu, Xing Hu, Ling Liang, Yufei Ding, Guoqi Li, Guangshe Zhao, Peng Li, and Yuan Xie. Rethinking the performance comparison between snns and anns. *Neural Networks*, 121:294–307, 2020.

[25] Pete Warden. Speech commands: A dataset for limited-vocabulary speech recognition. *arXiv preprint arXiv:1804.03209*, 2018.

[26] Emre Yılmaz, Ozgür Bora Gevrek, Jibin Wu, Yuxiang Chen, Xuanbo Meng, and Haizhou Li. Deep convolutional spiking neural networks for keyword spotting. In *Proceedings of Interspeech*, pages 2557–2561, 2020.

[27] Thomas Pellegrini, Romain Zimmer, and Timothée Masquelier. Low-activity supervised convolutional spiking neural networks applied to speech commands recognition. In *2021 IEEE Spoken Language Technology Workshop (SLT)*, pages 97–103. IEEE, 2021.

Authorized licensed use limited to: IEEE Xplore. Downloaded on May 03,2022 at 07:00:56 UTC from IEEE Xplore. Restrictions apply.