AXONAL DELAY AS A SHORT-TERM MEMORY FOR FEED FORWARD DEEP SPIKING NEURAL NETWORKS

Pengfei Sun[1] *Longwei Zhu*[2*] *Dick Botteldooren*[1*]

[1] Ghent University, Belgium
[2] Institute for Infocomm Research, A*STAR, Singapore

ABSTRACT

The information of spiking neural networks (SNNs) are propagated between the adjacent biological neuron by spikes, which provides a computing paradigm with the promise of simulating the human brain. Recent studies have found that the time delay of neurons plays an important role in the learning process. Therefore, configuring the precise timing of the spike is a promising direction for understanding and improving the transmission process of temporal information in SNNs. However, most of the existing learning methods for spiking neurons are focusing on the adjustment of synaptic weight, while very few research has been working on axonal delay. In this paper, we verify the effectiveness of integrating time delay into supervised learning and propose a module that modulates the axonal delay through short-term memory. To this end, a rectified axonal delay (RAD) module is integrated with the spiking model to align the spike timing and thus improve the characterization learning ability of temporal features. Experiments on three neuromorphic benchmark datasets : NMNIST, DVS Gesture and N-TIDIGITS18 show that the proposed method achieves the state-of-the-art performance while using the fewest parameters.

Index Terms— Axonal Delay, Deep Spiking Neural Network, Supervised Learning

1 Introduction

Spiking neural networks, which are composed of biologically plausible spiking neurons, have been proven to be robust on several tasks for both unsupervised[1, 2, 3] and supervised learning[4, 5]. Not only the visual cognitive tasks, recent researches also showed the enormous potential of spiking neural networks on acoustical tasks[6, 7]. The spiking neuron, which generates spikes when its membrane potential exceeds the threshold usually relies on spiking events in time to propagate the information forward. Thus, SNNs are inherently suited for problems of temporal nature. With the fiery and tremendous growth of artificial intelligence in various research areas, a series of multi-layer spiking neural networks have also been rapidly developed for different learning tasks[8, 9, 10, 11, 12].

However, the randomness in the spiking character of an SNN itself dictates its imperfection. With the inexplicable nature of the inner activation of the multi-layer network, the assignment of temporal credit is a huge problem. Several delay-based learning paradigms have been proposed to solve such discrepancies problem. Knoblauch and Sommer [13] discussed short and long delays effects in Spike Timing Dependent Plasticity (STDP) rule, while Zhang et al. [14] proposed that synaptic delay can be used to learn the precise spiking time by affecting the membrane potential in a multilayer spiking neural network. Nevertheless, these methods completely ignore the possibility of modifying axonal delay in deep spiking neural networks. Neurophysiology evidence suggests that axonal delay modulation can occur as a short-term memory during the learning process, greatly affecting learning performance. Researches also show that the axonal delays could be the fundamental of the different response characteristics of different neuronal populations[15, 16].

In this article, we propose modulation of axonal delay in the deep spiking neural networks based on the supervised learning methods. We describe the updating rule of the rectified axonal delay(RAD) module and show how this module behaves in the spiking neuron. The comprehensive experiments show that deep SNNs with our proposed RAD module could significantly outperform the benchmarks in terms of accuracy and model size.

2 PROPOSED METHOD

In this work, we propose a rectified axonal delay (RAD) module to align the spike firing timing during the pulse transmission along the axon. This module is implemented on top of the spiking neuron model and is expected to improve the representation power especially for time-critical identification tasks.

[*] Corresponding Author

This research received funding from the Flemish Government under the ”Onderzoeksprogramma Artificiele Intelligentie (AI) Vlaanderen” and Programmatic grant no. A1687b0033 from the Singapore Governments Research, Innovation and Enterprise 2020 plan (Advanced Manufacturing and Engineering domain).

978-1-6654-0540-9/22/$31.00 ©2022 IEEE

2.1 Spiking Neuron Model

The spiking neuron is the computational unit of SNNs that communicates with spikes and maintains an interval membrane potential over time. In this paper, we adapt the spike response model (SRM) as the neuron model and formulate our method based on SLAYER-PyTorch [8], which is an effective and powerful training framework.

The sub-threshold membrane potential of the SRM neuron can be described as follows

$$u_i^l(t) = \sum_j (W_{ij}^{l-1}(\varepsilon * s_j^{l-1})(t) + (\nu * s_j^l)(t)) \tag{1}$$

Where $u_i^l(t)$ is the membrane potential of neuron i in layer l at time t and s_j^{l-1} are the incoming spikes. $\varepsilon(t)$ and $\nu(t)$ denote spike response kernel and refractory kernel respectively. The incoming spikes are converted into a response signal by convolving incoming spikes with a spike response kernel. According to Eq.1, the neuron's state is updated by the sum of the postsynaptic potential (PSP) and refractory response, where the PSP is the weighted sum of the response signal from other neurons and the refractory phase describes the brief period before the neuron regains its capacity to make a second response. The neuron generates an output spike when the $u_i(t)$ surpasses the predefined threshold θ_u and transmits the spikes to the subsequent neurons along the axon. This generation process can be formulated by a Heaviside step function Θ as follows

$$s_i^l(t) = \Theta(u_i^l(t) - \theta_u) \tag{2}$$

2.2 Rectified Axonal Delay Module

The axonal delay module of each neuron simulates the delay in the transmission process. We denote N is the number of neurons at layer l, thus, the spike train $s^l(t)$ can be represented as follows

$$s^l(t) = \{s_1^l, ..., s_N^l\} \tag{3}$$

To speed up the training and the response of the networks, we limit the time delay of each neuron to a reasonable range.

$$\hat{s}_d^l(\hat{t}) = \delta(t - \hat{d}^l) * s^l(t) \tag{4}$$

$$\hat{d}^l = \begin{cases} 0 & d < 0 \\ d & 0 \leq d \leq \theta_d \\ \theta_d & \theta_d < d \end{cases} \tag{5}$$

The θ_d refers to the limit delay of the spiking neuron and $\hat{s}_d^l(\hat{t})$ is the shifted spike trains.

During the training, $\widetilde{s}^l(t)$ is the desired spike trains for all output neurons at time t. Then the loss function L is usually defined as

$$L = (\sum_{t=0}^{T} \widetilde{s}^l(t) - \sum_{t=0}^{T} s^l(t))^2 \tag{6}$$

The temporal dimension is discretized with the sampling time T_s such that $t = nT_s$. With N_s denoting the total number of samples, the observation time becomes $T = (N_s - 1)T_s$. Taking into account the temporal dependency, the gradient term of synaptic weight is the same as Shrestha et al. [8] except there is a delay shift of spike. For the axonal delay of layer l, the gradient term is given by

$$\triangledown \hat{d}^l = T_s \sum_{n=0}^{N_s} \frac{\partial L[n]}{\partial \hat{d}^l} \tag{7}$$

Here $L[n]$ is the loss at time instance n. Using the chain rule and understanding the fact that the loss $L[n]$ is dependent on all previous values of shifted spike trains. we get

$$\triangledown \hat{d}^l = T_s \sum_{n=0}^{N_s} \sum_{m=0}^{n} \frac{\partial \hat{s}_d^l[m]}{\partial \hat{d}^l} \frac{\partial L[n]}{\hat{s}_d^l[m]} \tag{8}$$

We use finite difference approximation $\frac{\hat{s}_d^l[m] - \hat{s}_d^l[m-1]}{T_s}$ to numerically estimate the gradient term $\frac{\partial \hat{s}_d^l[m]}{\partial \hat{d}^l}$, where $m-1$ refers to the previous time step. After formulating the forward and backward propagation, the axonal delay and synaptic weight are updated through the gradient descent algorithm.

3 EXPERIMENTS AND RESULTS

In this session, we conduct a series of experiments to validate the performance of our proposed method. In this paper, we will follow the similar notation as Shrestha et al. [8] to define our network architecture. The layer is separated by the $-$, a convolution layer with x channels and y filters are represented by the ${}_x c_y$, an aggregation (pooling) layer with y filters is represented by a_y. The input signal is expressed as $H \times W \times C$, where H and W are the spatial dimensions and C is the input channel.

3.1 IMPLEMENTATION DETAILS

In our experiments, we use the SLAYER-PyTorch as our training framework and implement the rectified axonal delay (RAD) module on top of it. Each network and RAD module are trained with an identical optimizer. We use the number of spikes generated from the last layer as the loss measurement and specify the desired spikes for different tasks respectively. We use the response kernel $\varepsilon(t) = \frac{t}{\tau_s}\exp(1 - \frac{t}{\tau_s})\Theta(t)$ and refractory kernel $\nu(t) = -2\theta_u \frac{t}{\tau_r}\exp(1 - \frac{t}{\tau_r})\Theta(t)$. Here, τ_s and τ_r are the time constant of the kernels. The simulation step time T_s is set as 1 ms. Table 1 lists all the hyper-parameters we used in our three examples. All the experiments are run for 5 independent trials and we report the average performance and deviation for a fair comparison. Our codes and the detailed configuration are made publicly available[1].

[1] The codes is available at:https://github.com/bamsumit/slayerPytorch

Table 1. Detailed hyper-parameter settings for different datasets

Dataset	τ_s	τ_r	θ_d	θ_u
NMNIST	1	1	64	10
DVS Gesture	5	5	64	10
N-TDIDIGITS18	5	5	128	10

Table 2. Comparison with the state-of-the-art in terms of network size and accuracy.

	Method	Params	Accuracy
NMNIST	Tandem learning. [18]	4.63 MB	99.31%
	Wu et al. [19]	17.67 MB	99.53%
	Spike-based BP. [20]	65.36 MB	**99.61%**
	Our method	**2.13 MB**	99.37%
DVSGesture	TrueNorth [21]	1.99 MB	91.77(94.59)%
	DECOLLE [22]	1.25 MB	95.54%
	Ghosh et al. [23]†	2.12 MB	95.94%
	Spike-based BP. [20]	6.48MB	**97.57%**
	Our method	**1.06 MB**	96.97%
NTDIDIGITS	GRU-RNN [24]†	0.11 MB	90.90%
	Phased-LSTM [24]†	0.61 MB	91.25%
	ST-RSBP [25]	0.35 MB	93.63 ± 0.27%
	Our method	**0.08 MB**	**94.45%**

† Non SNN implementation.

3.2 DATASETS AND OVERALL RESULTS

3.2.1 NMNIST

The NMNIST[17] dataset contains 60000 training samples and 10000 testing samples originated from the MNIST image and then processed by the Dynamic Vision Sensor(DVS). Each sample has a 300 ms duration and the spatial dimension is 34×34 pixels. In our experiment, we only use the raw data and don't do any processing to compensate for the saccadic movement.

For this task, the following spiking CNN architecture is used: `34x34x2-16c5-a2-32c3-a2-64c3 -512-10`, wherein the pure numbers refer to the number of neurons at each fully-connected layer. As can be seen from Table 2, our method is very competitive compared with other benchmarks. The performance is a little lower than the best-reported result[20], while the model size is significantly reduced (30X). This kind of small size model will benefit more by combining the mapping technique [26].

3.2.2 DVS Gesture

The DVS-Gesture [21] is the dataset consisting of 29 subjects performing 11 different hand and arm gestures. These gestures are recorded using the DVS camera under 3 illumination conditions. Unlike the NMNIST, this task is not derived from the static image but the real movement of the subject. We use the first 23 subjects for training while the last 6 for testing.

An SNN with architecture `128x128x2-a4-16c5-a2 -32c3-a2-512-11` was trained in this dataset. To speed up training, we only use randomly selected 300ms long sequences. while for the inference, the first 1.5 *s* of action video for each class is used to classify the actions. The results are listed in Table 2. Our method shows the best accuracy of 96.21% on average and our model size is only 1 *MB* .

3.2.3 N-TDIDIGITS18

The N-TDIDIGITS18 [24] dataset is the neuromorphic version of TDIDIGITS [27]. It contains the 11 spoken digits ("oh," and the digits "0" to "9") and 64 response channels. We use the same train-test split of the dataset as [24].

The fully connected architecture `64-256-256-11` is explored. As we can see from Table 2, we report the best performance of 94.45% with a mean of 94.19% and a stand deviation of 0.18%. Our method significantly outperforms the Non-SNN based methods and other SNN based approaches. It is worth noting that our method can classify with fewer parameters and better performance. Compared with nonspiking networks like Phased Long Short Term Memory (Phased LSTM) [24], we can achieve 2.94% performance improvement while using 7*X* fewer parameters.

3.3 DETAILED ANALYSIS OF THE RESULTS

The effect of the proposed RAD module introducing the axonal delay function is further analyzed. As shown in Table 3, it can be observed that: (1) For the NMNIST dataset, which is recorded from static images and does not comprise much temporal information, the performance is only slightly better when combining spiking CNN network with the RAD module, and the improvement of adding a delay is not significant. This indicates that precise spike timing is not important in this dataset [28]. (2) For the event-based video dataset DVS Gesture and neuromorphic audio dataset N-TDIDIGITS18 which are both highly temporal-dependent, the proposed method introduces rapid performance gains with added delay, which demonstrate that the axonal delay contributes to optimizing the spike temporal information that enhances the representation of features. As can be seen from the table, the DVS Gesture classification accuracy is boosted by 0.57% with the axonal delay. The most considerable performance improvement comes from the speech task, and the best accuracy can reach 94.45%, which is more than 15% better.

During the experiments with axonal delay studies, it is found that after a large amount of repetitive training, the spike occurrence can be over-inhibited. That means a spike might be depressed for too long. As a result, for the speech task from the table 3, when the $\theta_d = +\infty$, the classification performance becomes worse. Thus, it is necessary to constrain the shifting range to prevent such over-inhibition.

A simple experiment helps to understand the effectiveness of the RAD module. We take voice '5' in the NTIDIGITS as an input example (Figure 1(a)) and explore the three fully-connected networks with different axonal delay limitations (Figure 1(b)). Figure 1(c) left and right illustrate the cumulative spike count distribution over time and the total spike

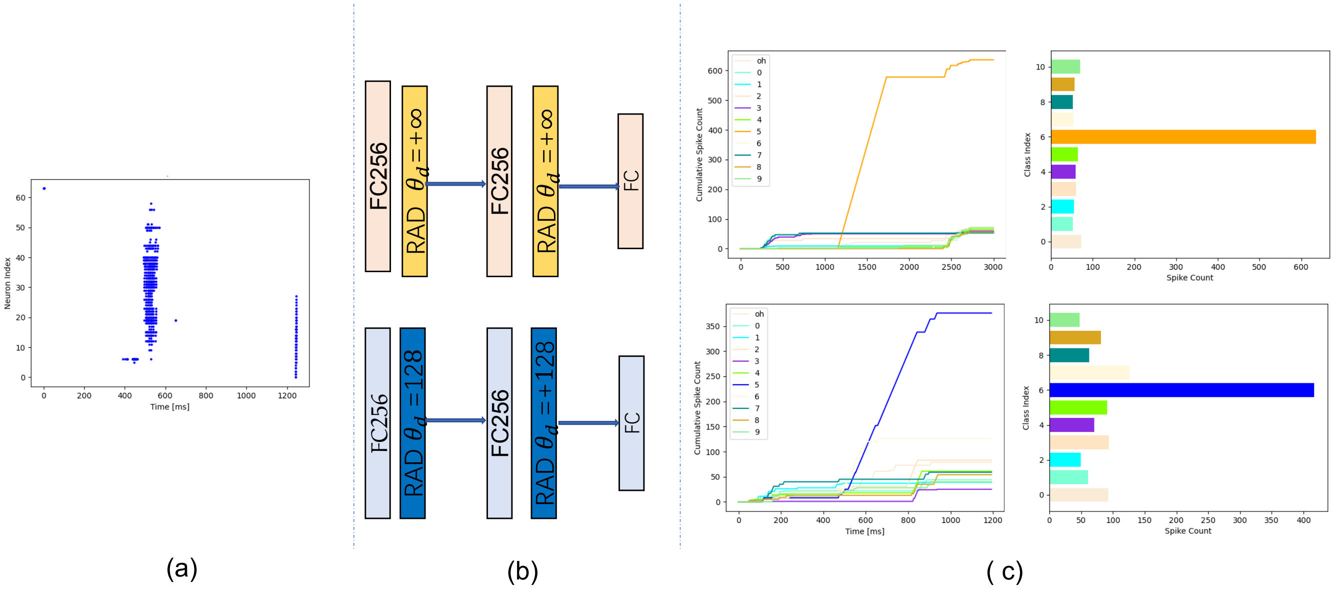

Fig. 1. Illustration of flow chart of the proposed method. (a) Spiketrains of input sample '5'. (b) Three fully-connectd layers with two different threshold scheme of delay. (c) Cumulative Spikecount of last layer (left), Spikecount(right)

Table 3. Ablation studies for different delay threshold θ_d in the RAD module.

Dataset	θ_d	Params	Accuracy
NMNIST	0	2,126,754	99.26 ± 0.02%
	64	2,131,458	**99.33 ± 0.03%**
DVS Gesture	0	1,060,211	95.64 ± 0.65%
	64	1,060,771	**96.21 ± 0.63%**
N-TDIDIGITS18	0	85,259	78.86 ± 0.47%
	+∞	85,771	93.83 ± 0.10%
	128	85,771	**94.19 ± 0.18%**

count for every output neuron respectively. For both cases, the total spike count is used to decide which number was spoken, and both are well classified. While for the system delay, as can be observed from the Figure 1(c) left, the rectified axonal delay could classify the sample as early as 800 *ms* , while the unlimited axonal delay needs 1500 *ms* to vote for the true class.

4 CONCLUSIONS

Spiking neural networks (SNNs) offer the opportunity to include precise timing as part of the solution for classifying objects with a temporal aspect. However, this possibility has been rarely used in its full potential. In this work, we introduce the rectified axonal delay (RAD) as an additional degree of freedom for training that can easily be incorporated into existing SNN frameworks. The comprehensive evaluation results show that the proposed RAD module could significantly outperform several other models on two out of the three benchmarks that were selected. Not surprisingly, the new model performs particularly well on problems where timing matters. We believe that using spiking neuron delay to model short-term memory needed to interpret a spoken word or a gesture, explain the performance increase and the reduction of parameters needed. Biological evidence conceptually supports this statement. In addition, such module can be easily incorporated into the current deep spiking models with very few tunable parameters added.

5 ACKNOWLEDGEMENTS

The authors would express our very great appreciation to Dr. Sumit Bam Shrestha for his valuable and constructive suggestions and technical support during the development of this research work.

6 References

[1] Wulfram Gerstner, Richard Kempter, J Leo Van Hemmen, and Hermann Wagner, "A neuronal learning rule for sub-millisecond temporal coding," *Nature*, vol. 383, no. 6595, pp. 76–78, 1996.

[2] Sen Song, Kenneth D Miller, and Larry F Abbott, "Competitive hebbian learning through spike-timing-dependent synaptic plasticity," *Nature neuroscience*, vol. 3, no. 9, pp. 919–926, 2000.

[3] Malu Zhang, Hong Qu, Ammar Belatreche, Yi Chen, and Zhang Yi, "A highly effective and robust membrane potential-driven supervised learning method for spiking neurons," *IEEE*

transactions on neural networks and learning systems, vol. 30, no. 1, pp. 123–137, 2018.

[4] Sander M Bohte, Joost N Kok, and Han La Poutre, "Error-backpropagation in temporally encoded networks of spiking neurons," *Neurocomputing*, vol. 48, no. 1-4, pp. 17–37, 2002.

[5] Sumit Bam Shrestha and Qing Song, "Robustness to training disturbances in spikeprop learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 7, pp. 3126–3139, 2018.

[6] Zihan Pan, Haizhou Li, Jibin Wu, and Yansong Chua, "An event-based cochlear filter temporal encoding scheme for speech signals," in *2018 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2018, pp. 1–8.

[7] Zihan Pan, Yansong Chua, Jibin Wu, Malu Zhang, Haizhou Li, and Eliathamby Ambikairajah, "An efficient and perceptually motivated auditory neural encoding and decoding algorithm for spiking neural networks," *Frontiers in neuroscience*, vol. 13, pp. 1420, 2020.

[8] Sumit Bam Shrestha and Garrick Orchard, "SLAYER: Spike layer error reassignment in time," in *Advances in Neural Information Processing Systems 31*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds., pp. 1412–1421. Curran Associates, Inc., 2018.

[9] Yan Xu, Xiaoqin Zeng, Lixin Han, and Jing Yang, "A supervised multi-spike learning algorithm based on gradient descent for spiking neural networks," *Neural Networks*, vol. 43, pp. 99–113, 2013.

[10] Sumit Bam Shrestha and Qing Song, "Event based weight update for learning infinite spike train," in *2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE Computer Society, 2016, pp. 333–338.

[11] Saeed Reza Kheradpisheh and Timothée Masquelier, "Temporal backpropagation for spiking neural networks with one spike per neuron," *International Journal of Neural Systems*, vol. 30, no. 06, pp. 2050027, 2020.

[12] 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.

[13] Andreas Knoblauch and Friedrich T Sommer, "Spike-timing-dependent synaptic plasticity can form "zero lag links" for cortical oscillations.," *Neurocomputing*, vol. 58, pp. 185–190, 2004.

[14] Malu Zhang, Jibin Wu, Ammar Belatreche, Zihan Pan, Xiurui Xie, Yansong Chua, Guoqi Li, Hong Qu, and Haizhou Li, "Supervised learning in spiking neural networks with synaptic delay-weight plasticity," *Neurocomputing*, vol. 409, pp. 103–118, 2020.

[15] Catherine E Carr and Masakazu Konishi, "Axonal delay lines for time measurement in the owl's brainstem," *Proceedings of the National Academy of Sciences*, vol. 85, no. 21, pp. 8311–8315, 1988.

[16] Carl R Stoelzel, Yulia Bereshpolova, Jose-Manuel Alonso, and Harvey A Swadlow, "Axonal conduction delays, brain state, and corticogeniculate communication," *Journal of Neuroscience*, vol. 37, no. 26, pp. 6342–6358, 2017.

[17] Garrick Orchard, Ajinkya Jayawant, Gregory K Cohen, and Nitish Thakor, "Converting static image datasets to spiking neuromorphic datasets using saccades," *Frontiers in neuroscience*, vol. 9, pp. 437, 2015.

[18] 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*, 2021.

[19] Yujie Wu, Lei Deng, Guoqi Li, Jun Zhu, Yuan Xie, and Luping Shi, "Direct training for spiking neural networks: Faster, larger, better," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019, vol. 33, pp. 1311–1318.

[20] Wei Fang, Zhaofei Yu, Yanqi Chen, Timothée Masquelier, Tiejun Huang, and Yonghong Tian, "Incorporating learnable membrane time constant to enhance learning of spiking neural networks," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 2661–2671.

[21] Arnon Amir, Brian Taba, David Berg, Timothy Melano, Jeffrey McKinstry, Carmelo Di Nolfo, Tapan Nayak, Alexander Andreopoulos, Guillaume Garreau, Marcela Mendoza, et al., "A low power, fully event-based gesture recognition system," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 7243–7252.

[22] Jacques Kaiser, Hesham Mostafa, and Emre Neftci, "Synaptic plasticity dynamics for deep continuous local learning (decolle)," *Frontiers in Neuroscience*, vol. 14, pp. 424, 2020.

[23] Rohan Ghosh, Anupam Gupta, Andrei Nakagawa, Alcimar Soares, and Nitish Thakor, "Spatiotemporal filtering for event-based action recognition," *arXiv preprint arXiv:1903.07067*, 2019.

[24] Jithendar Anumula, Daniel Neil, Tobi Delbruck, and Shih-Chii Liu, "Feature representations for neuromorphic audio spike streams," *Frontiers in neuroscience*, vol. 12, pp. 23, 2018.

[25] Wenrui Zhang and Peng Li, "Spike-train level backpropagation for training deep recurrent spiking neural networks," *arXiv preprint arXiv:1908.06378*, 2019.

[26] Roshan Gopalakrishnan, Yansong Chua, Pengfei Sun, Ashish Jith Sreejith Kumar, and Arindam Basu, "Hfnet: A cnn architecture co-designed for neuromorphic hardware with a crossbar array of synapses," *Frontiers in neuroscience*, vol. 14, pp. 907, 2020.

[27] R Gary Leonard and George Doddington, "Tidigits speech corpus," *Texas Instruments, Inc*, 1993.

[28] Laxmi R Iyer, Yansong Chua, and Haizhou Li, "Is neuromorphic mnist neuromorphic? analyzing the discriminative power of neuromorphic datasets in the time domain," *Frontiers in neuroscience*, vol. 15, pp. 297, 2021.