

Linear Leakage: Better Robustness for Spiking Neural Network

Jinming Che*, Jian Cao*, Shuo Feng*, Jue Chen*, Yuan Wang**

* School of Software & Microelectronics, Peking University, China

** School of Software & Microelectronics, Peking University, China

{chejinming, fengshuo, chenjue}@stu.pku.edu.cn, caojian@ss.pku.edu.cn, wangyuan@pku.edu.cn

Abstract— As the third-generation neural networks, Spiking Neural Networks (SNNs) have the potential to replace ANNs in noisy input scenarios due to the advantages of low power consumption and high robustness. In this paper, we propose a method that uses linear leakage on Leaky-integrate-fire (LIF) neurons of direct training of SNNs. First, this paper provides an in-depth analysis of the process of integrating and firing spikes on LIF neurons based on the linear leakage method, and finally demonstrates the inverse relationship between the leakage parameters τ_m and spiking rate ($1/T$) in linear leakage. Second, this paper designs experiments on the application of the linear leakage method to three different network models based on MNIST and Cifar10 datasets. The experimental results show that the linear leakage method in this paper has stronger robustness compared to the no-leakage and exponential leakage approaches. This finding provides a new idea with practical significance for the further study of LIF models of SNNs.

Keywords— Spiking neural networks, Spatio-temporal backpropagation, Leaky-integrate-fire neuron, Robustness, Noise attacks

I. INTRODUCTION

In recent years, the mature ANN has been used in many important areas such as finance, security and healthcare, however, the large amount of computational resources consumed makes it computationally expensive for ANN to solve simple problems. In addition, misclassification brought by invisible noise perturbation during ANN training will lead to irreparable consequences. A large number of robust ANN-based experiments [14-16] have emerged to reduce the effects of noise, but they are all accompanied by cumbersome experimental steps, so there is an urgent need for a simple and effective highly robust scheme to emerge as an alternative to ANN in noise input.

Meanwhile, a new generation of neural network, SNN, has been explored to simulate the function of human brain to achieve complex cognitive tasks under low power consumption, with the advantageous features of simulating biological neuron mechanism and propagating spatio-temporal information by spike as signal. In 2015, J. Yang [17] et al. demonstrated that the classification capability of SNNs is robust to sinusoidal and Gaussian noise of the input signal. In 2019, it has been demonstrated that SNNs are more resilient than ANNs under black-box attacks [19]. In 2021, R. El-Allami et al. [6]

investigated the effect of different structural parameters (e.g. threshold voltage, time step) on the robustness of SNN models under white-box attacks based on previous experiments, and the experimental results showed that SNNs improved robustness by 85% over ANNs for specific parameters. The SNN is thus shown to have the potential to replace ANNs in noisy inputs scenarios.

As research progresses, attention has been paid to the effects of the LIF neuron model and the leaky factor on SNN robustness. Among them, the LIF model is the most common SNN neuron model and the leaky factor is the signature parameter of LIF neurons. 2020, S. Sharmin [4] et al. demonstrated the potential of LIF neurons to be noise-resistant, and they pointed out that the leakage parameter is an additional knob to control the noise resistance of the model based on the effect of the inherent structural parameters of LIF neurons on the accuracy of the model. In 2021, S. S. Chowdhury [5] et al. investigated the role played by leakage in SNNs and concluded from frequency domain analysis that leakage can filter high frequency components thus improving model robustness. However, this field still needs a lot of theoretical and experimental support to explore highly robust SNNs and optimize neuronal models.

To address the existing problem, we want to find a way of leakage to make the network perform well in robustness based on direct training of SNNs. The experiments and analysis in this paper revolve around the following three key questions, which are summarised in turn in the conclusion section of the article.

- Can the original robustness of the SNN be improved if the leakage method is changed (linear leakage)?
- Assuming that question 1) is valid, what role does linear leakage play in the LIF neuron model in the SNN?
- Assuming that question 1) is valid, how much robustness is improved by linear leakage? Is this robustness generalisable across different datasets and networks?

For the above problem, this paper conducts a series of SNN robustness experiments around linear leakage. To the best of our knowledge, this paper is the first to propose the use of linear leakage on LIF neurons with directly trained SNNs (explained in Section 4.1). The key contributions can be summarized as follows:

- (1) We propose a method that uses linear leakage on LIF neurons of direct training of SNN.

- (2) We analyze the process of integrating and firing spikes based on linear leakage of LIF neurons, and demonstrates the inverse relationship between the leakage parameters τ_m and spiking rate ($1/T$) in linear leakage.
- (3) In this paper, a linear leakage method is applied on three different network models based on MNIST and Cifar10 datasets. After experimental comparison and analysis, we found that the linear leakage method in this paper has stronger robustness compared to the no-leakage and exponential leakage approaches.

II. SNN FOUNDATION

Spiking neural networks, as the third generation of neural networks, have lower power binary inputs and more bionic neuron models. Among many models, the LIF model visually describes the relationship between the membrane potential U and the input current I , and its equivalent circuit is shown in Figure 1(c). In the equivalent circuit, the resistance R_m and capacitance C_m are connected in parallel. When the presynaptic neuron generates a spike, a current I is generated on the prominence connected to the postsynaptic neuron that partly charging the capacitor C_m and partly leaking on the resistor R_m .

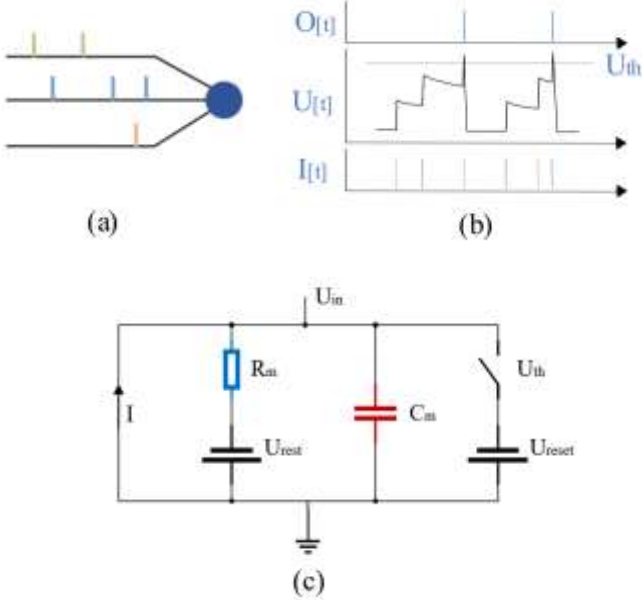


Figure 1. (a) Neuron receiving spike signals from the presynaptic neuron. (b) Process of integrating and firing spike signals on the neuron. (c) Equivalent circuit diagram of LIF

$$\tau_m \frac{dU}{dt} = -[U - U_{rest}] + R_m I, \quad U \leq U_{th} \quad (1)$$

$$U[t] = e^{\frac{-1}{\tau_m}} U[t-1] (1 - O[t]) + W \quad (2)$$

$$O[t] = \begin{cases} 0, & U \leq U_{th} \\ 1, & \text{otherwise} \end{cases} \quad (3)$$

Equation (1) is the first-order differential equation of the LIF model, where $\tau_m = R_m C_m$ is the membrane time constant, U is the membrane voltage, I is the sum of the firing currents of each presynaptic neuron, U_{reset} is the reset voltage, and U_{rest} is the

resting voltage. Figure 1(a) and Figure 1(b) shows the process of neuronal spike sequence input and firing.

We refer to the conventional leakage process as exponential leakage because the membrane voltage leakage term exists in an exponential form. The state update process of the conventional LIF model can be expressed as Equations (2) and (3), where $U[t]$ denotes the membrane voltage at the current moment, $U[t-1]$ represents the membrane voltage at the previous moment, $e^{\frac{-1}{\tau_m}}$ is the membrane voltage leakage term, $O[t]$ represents the pulse emission function and W is the weight. At the membrane potential U does not reach the threshold voltage U_{th} the spike signal accumulates in the time domain and decays according to the membrane time constant τ_m with time step; once the membrane potential U reaches the threshold voltage U_{th} , the neuron immediately fires a spike signal and resets to U_{reset} .

III. LINEAR LEAKAGE

This section details the inspiration, advantages and theoretical support of linear leakage, and also demonstrates the feasibility of linear leakage.

A. Hardware Inspiration

The event-based approach to information propagation of SNNs allows for low power consumption, and as a result, SNNs are receiving industry attention and neuromorphic architecture hardware such as IBM TrueNorth and Intel Loihi are being developed to accommodate itself. This paper proposes a linear leakage method inspired by the method used by Y. Kuang [18] and others in the development of neuromorphic chips. During the operation of the neuromorphic chip PKUNC64C, the voltage leakage within each time step is represented by subtracting a fixed value to accommodate the low power consumption of SNN. Equation (4) represents the way the chip updates the voltage at the hardware level.

$$U_j[t+1] = U_j[t] + \sum_{i=0}^{1023} x_i w_{ij} s_{ij} - \lambda_j \quad (4)$$

Where $U_j[t]$ denotes the membrane voltage of the time step t in the layer j , x_i is the spike signal from the presynaptic of the layer i , w_{ij} is the connection weight of layer i and layer j , s_{ij} is the synaptic connectivity between layer i and layer j , and λ_j is the leakage constant of the layer j . This method reduces the chip computing power and improves the computational robustness.

B. Linear Leakage Advantage

We present a linear leakage method that is more similar to the one used on neuromorphic chips. Starting from the second time step, the linear leakage quantitatively reduces the membrane voltage at each time step. We have two reasons for proposing linear leakage method. First, the leakage term uses subtraction compared to multiplication requiring less computation, which is in line with the low-power characteristics of SNNs. Second, the use of subtraction for the

leakage term is more effective in offsetting noise than multiplication, which can bring better robustness to the SNN. Overall, we propose linear leakage in order to optimize the conventional exponential leakage to obtain lower power consumption and higher robustness. Algorithm 1 explains the LIF neuron state update step base on linear leakage.

Algorithm 1 LIF Neuron State Update based on Linear Leakage

Input: Pixel input $inputs$, total time steps T , total number of network layers L , weights W , neuron membrane voltage U , linear leakage constant τ_m , threshold voltage U_{th} , reset voltage U_{reset} .

Initialization: $U_l[t] = 0, \forall l = 1, \dots, L$.

//Forward propagation state update

for $t = 1$ to T **do**

 // Generate spike inputs from the first convolutional layer as the coding layer

$O_l[t] = \text{Encodinglayer}(inputs)$

for $l = 2$ to $L - 1$ **do**

$U_l[t] = U_l[t - 1](1 - O_{l-1}[t]) + W_l$

if $U_l[t] \geq U_{th}$ **then**

 // Fires a spike and resets the membrane voltage

$O_l[t] = 1, U_l[t] = U_{reset}$

else

 // Membrane voltage decays at each time step

$O_l[t] = 0, U_l[t] = U_l[t - 1] + W_l - \tau_m$

endif

endfor

 //Last layer does not fire spikes

$U_L[t] = U_L[t - 1] + W_L - \tau_m$

endfor

// Error back propagation

for $t = T$ to 1 **do**

for $l = L - 1$ to 1 **do**

 // Evaluate the partial derivatives of the loss with respect to the weights through the time domain

$$\Delta W = \frac{\partial \text{Loss}}{\partial O_l[t]} \frac{\partial O_l[t]}{\partial U_l[t]} \frac{\partial U_l[t]}{\partial W_l[t]}$$

endfor

endfor

C. Theoretical Derivation

This paper explores the intrinsic relationship between linear leakage and pulse emissivity through the derivation of equations. As shown in Figure 2 and Figure 3, assume that a spike signal U_{mem} is received at each time step in a single neuron, the membrane voltage $U_{[t]}$ integrates each spike signal and linear decays by the leaky factor τ_m at each time step. Linear leakage implies that the membrane voltage $U_{[t]}$ decays a fixed value at each time step. In order to understand the leakage method more intuitively, Figure 3 shows the change trend of the LIF neuron membrane voltage signal under different leakage modes when no other signal is received. In terms of bionicity, linear leakage is somewhere between exponential leakage and no leakage.

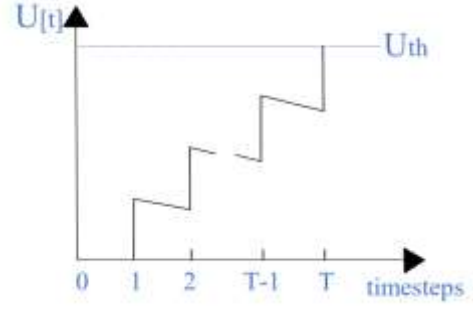


Figure 2. Integrated membrane voltage $U_{[t]}$ reach the threshold voltage U_{th} at the time step T

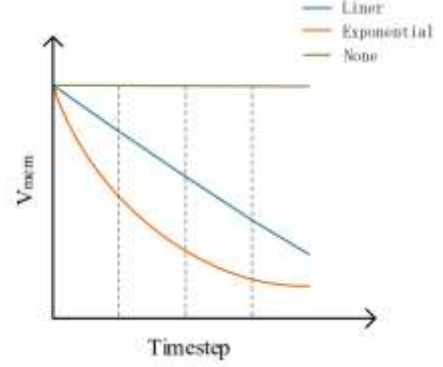


Figure 3. Trends of membrane voltage of LIF neurons with different leakage methods

The time step 1, $U_{[1]} = U_{mem}$

The time step 2, $U_{[2]} = 2U_{mem} - \tau_m$

The time step 3, $U_{[3]} = 3U_{mem} - 2\tau_m$

...

The time step T , $U_{[T]} = TU_{mem} - (T - 1)\tau_m = U_{th}$

At the time step T , the integrated membrane voltage $U_{[t]}$ is equal to the threshold voltage U_{th} . This paper uses the variable R_1 represents the rate of the threshold voltage U_{th} and the spike signal U_{mem} , and the variable R_2 represents the rate of the leaky factor τ_m and spike signal U_{mem} , then we can obtain the relationship between the spiking rate ($1/T$), R_1 and R_2 , which is shown in Equation (5).

$$\frac{1}{T} = \frac{1 - R_2}{R_1 - R_2} \quad (5)$$

Figure 4 is a function plot of the R_1 , R_2 and spiking rate $1/T$ that we have drawn. Under different R_1 , the spiking rate $1/T$ shows a decreasing trend when R_2 varying from 0.05 to 0.25, which means, the spiking rate decreases and the sparsity of the network increases as the leakage increases. In addition, we find that the variation of the linear leakage parameter affects the classification accuracy of the network to some extent. Figure 5 shows the variation of accuracy with different network models when the size of the linear leaky factor is varied.

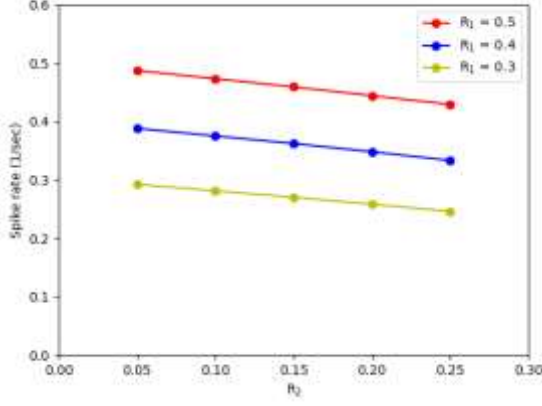


Figure 4. Under different $R_1 = \frac{U_{mem}}{U_{th}}$, the relationship between the spiking rate $1/T$ and $R_2 = \frac{\tau_m}{U_{mem}}$

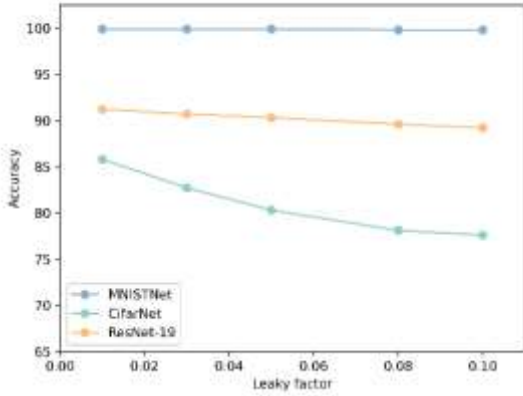


Figure 5. Variation of accuracy with different network models when the size of the linear leaky factor is varied

IV. EXPERIMENTAL DESIGN

A. Baseline

There are two common approaches to train SNN:

- 1) **Pre-training ANN Model and Converting to SNN:** this approach [10-12] usually requires a relatively larger time step, which reduces information loss while also introducing higher latency and energy consumption, while this method ignores the SNN time domain information, which is contrary to the original purpose of SNN.
- 2) **Direct Training of SNN:** this approach can be divided into unsupervised training and supervised training. The former uses the Spike Time Dependent Plasticity (STDP) rule [7], this method is often limited to shallow structures and cannot be applied to complex datasets [3]. The latter uses the gradient substitution method to solve the problem of unsteady SNN gradients.

Wu [1] et al. proposed the STBP method and applied it to deep SNNs [2,9], smaller time steps are used while taking into account temporal and spatial information, thus achieving higher accuracy rates in large data sets. In this experiment, we use the STBP framework as baseline.

B. Experimental Procedure

In this paper, we use a clean dataset for training first and then add noise to test the robustness of the model. The reason for not adding noise to the train dataset are as follows: on the one hand, the robustness of SNN models trained with distinct noise and leaky factors is different, and it is difficult to distinguish between the robustness brought by the linear leakage method and the robustness brought by training with a noise-enhanced dataset. On the other hand, robustness generated under the same noise training does not generalise, and we need to obtain robustness of SNN models with generalisation capability, i.e. models that are robust under different noises. The similar method was used by [5,13]. Figure 6 shows the experimental procedure.

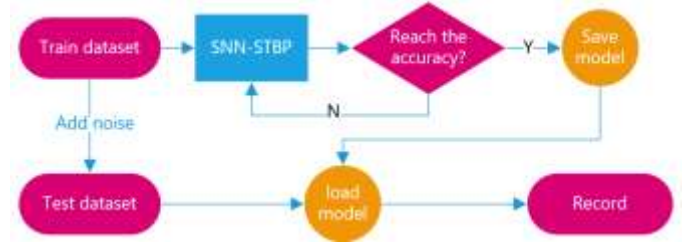


Figure 6. Experimental procedure

C. Network Model

To ensure that the linear leakage approach is robust, we test on the MNIST dataset first. The STBP framework is used in this experiment, and the first convolutional layer is used as the coding layer of the network. The structure of MNISTNet network model we use is represented as (28*28-15C5-2P-40C5-2P-300FC-10FC). Where the input is a 28*28 grayscale map, C5 represents a convolutional layer with a convolutional kernel size of 5*5, 2P represents a mean pooling layer with a kernel of 2*2, and FC represents a fully connected layer. A spiking layer is added to spike the float values after each layer.

We conduct the experiments on the Cifar10 dataset to demonstrate the generalization of linear leakage robustness. The structure of CifarNet model can be shown as (32*32*3-128C3-256C3-2P-512C3-2P-1024C3-512C3-32768FC-100FC-10P). Where the input is a 32*32 color map, C3 represents a convolutional layer with a convolutional kernel size of 3*3, 10P represents a one-dimensional average pooling layer with a kernel of 10*10 and a padding of 10 and FC represents a fully connected layer. Each convolutional layer is followed by the tBN layer [2]. After each network layer, a spiking layer is added to spike the float values.

The structure of ResNet-19 model can be shown as [32*32*3-128C3-(128C3-128C3)*3-(256C3-256C3)*3-(512C3-512C3)*2-1P-256FC-10FC]. Where the input is a 32*32 color map, C3 represents a convolutional layer with a convolutional kernel size of 3*3, the brackets represent a residual block, 1P represents a two-dimensional average pooling layer with a kernel of 1*1 and a padding of 1, and FC represents a fully-connected layer. An spiking layer is added after the last fully connected layer.

D. Experimental Parameters

In this experiment, we uniformly used the parameters: Timestep = 2, $U_{th}=0.2$, $U_{mem}=0.9$. Table 1 shows the specific leaky factors of different leaky methods for different datasets. In order to prevent the phenomenon of overfitting of different leaky methods trained in the model, in this paper, the network model weights of each leakage method are trained to the same accuracy and then the experimental data are saved and tested with a noisy dataset. Among them, the accuracy of MNISTNet is 99%, CifarNet is 80%, and ReaNet-19 is 85%.

TABLE 1. LEAKY FACTORS τ_m FOR DIFFERENT DATASETS

	MNISTNet	CifarNet	ResNet-19
Linear leakage τ_m	0.1	0.01	0.1
Exponential leakage τ_m	0.25	0.25	0.25

V. EXPERIMENTAL RESULTS AND ANALYSIS

We compare the performance of the MNISTNet model accuracy under the influence of Gaussian and impulse noise for the no-leakage (IF), exponential leakage (LIF) and linear leakage methods (as shown in Figure 7 and Figure 8). The experimental results show that the linear leakage is the best in terms of robustness under different noise influences, and in particular, the linear leakage shows extremely high robustness under small noise conditions. The absolute accuracy difference between linear leakage and exponential leakage reaches a maximum of 25% at mark ① when the impulse noise amplitude is 0.2. We visualize the feature map of the MNIST dataset extracted by the network at marker ①. Figure 9 shows the comparison of the feature maps extracted by the network under different leakage methods. From the figure, it can be seen that the feature maps extracted after linear leakage are less noisy compared with those of the other two leakage methods.

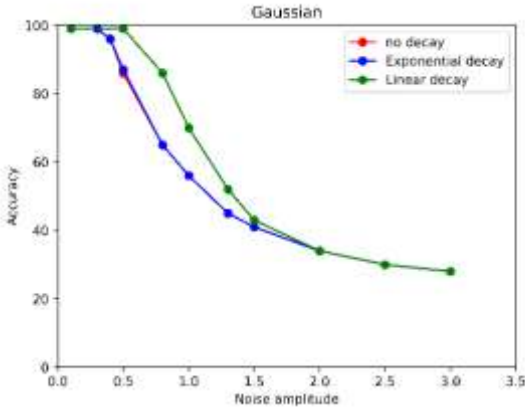


Figure 7. Accuracy of using MNISTNet network under Gaussian noise with no leakage (IF), exponential leakage (LIF) and linear leakage methods

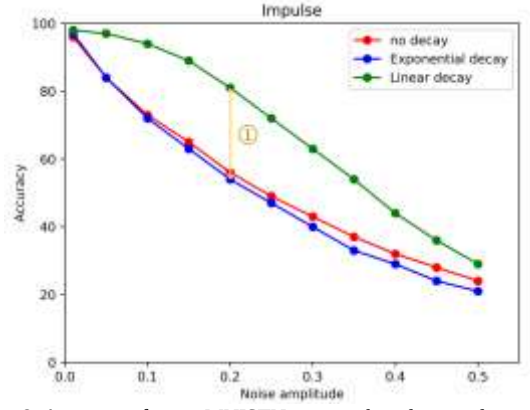


Figure 8. Accuracy of using MNISTNet network under impulsive noise with no leakage (IF), exponential leakage (LIF) and linear leakage methods

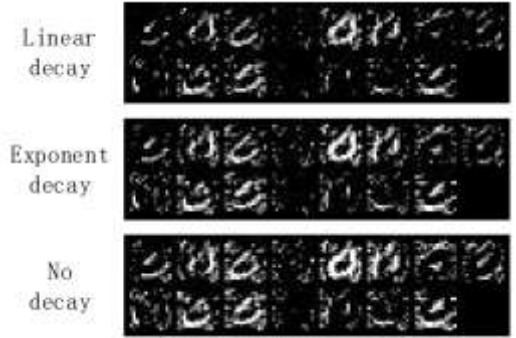


Figure 9. Comparison of feature visualization of MNIST dataset extracted by the network for no leakage (IF), exponential leakage (LIF) and linear leakage approaches at mark ①, with impulse noise amplitude is 0.2

We compare the performance of the CifarNet model accuracy under the influence of Gaussian and impulse noise for the no-leakage (IF), exponential leakage (LIF) and linear leakage methods (see Figure 10 and Figure 11). At mark ②, the absolute accuracy difference between linear and exponential leakage reaches a maximum of 20% when the Gaussian noise amplitude parameter is 18.

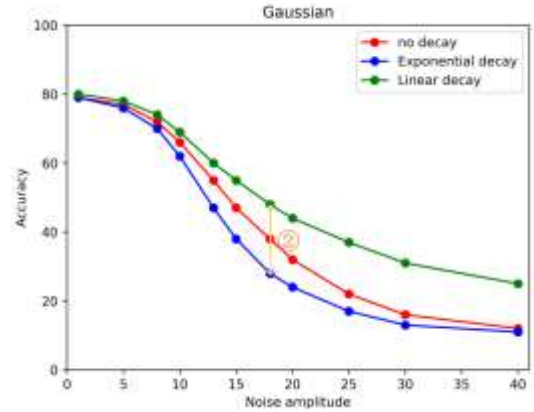


Figure 10. Accuracy of using CifarNet network under Gaussian noise with no leakage (IF), exponential leakage (LIF) and linear leakage methods

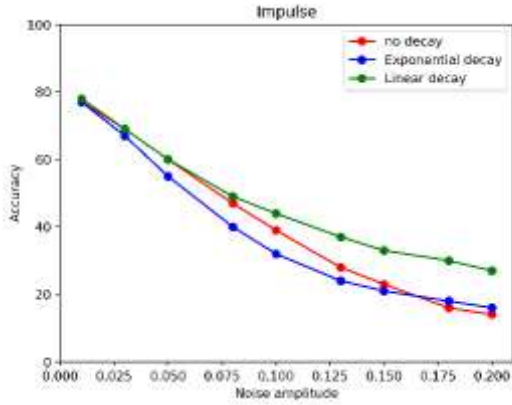


Figure 11. Accuracy of using CifarNet network under impulsive noise with no leakage (IF), exponential leakage (LIF) and linear leakage methods

We compare the performance of the ResNet-19 model accuracy under the influence of Gaussian and impulse noise for the no-leakage (IF), exponential leakage (LIF) and linear leakage methods (see Figure 12 and Figure 13). We find that linear leakage maintains the best robustness in the case of complex datasets and deep networks compared to other leakage methods. The difference in absolute accuracy between linear and exponential leakage reaches a maximum of 17% at mark ③ when the Gaussian noise amplitude parameter is 20.

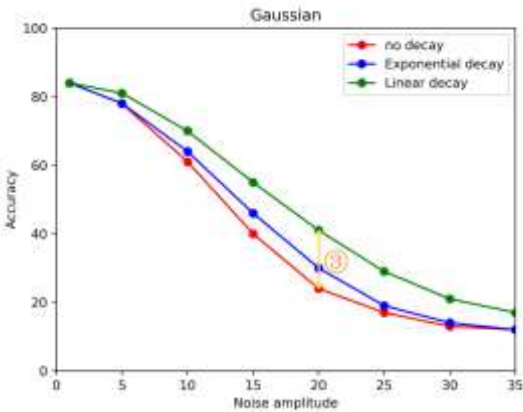


Figure 12. Accuracy of using ResNet-19 network under Gaussian noise with no leakage (IF), exponential leakage (LIF) and linear leakage methods

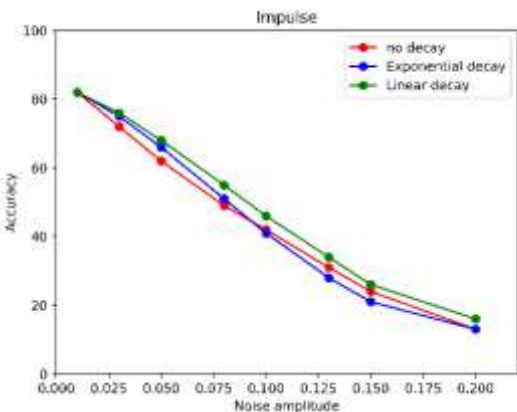


Figure 13. Accuracy of using ResNet-19 network under impulsive noise with no leakage (IF), exponential leakage (LIF) and linear leakage methods

In this paper, a linear leakage approach is applied on MNISTNet, CifarNet and ResNet-19 network models based on MNIST and Cifar10 datasets. The experimental data demonstrate that linear leakage has the best robustness compared to no-leakage and conventional exponential leakage methods in different datasets and different network models. The absolute accuracy difference between linear leakage and exponential leakage could reach a maximum of 25% under specific parameters, which implies that linear leakage has enough prospects and advantages for use in noisy input scenarios. In addition, experimental results in this paper on different datasets and networks demonstrate that the robustness of linear leakage is generalisable.

VI. CONCLUSIONS

The low power consumption and high robustness of SNNs make them a potential replacement for ANNs in noisy input scenarios. In this paper, we propose a method for using linear leakage in neuronal models of directly trained SNNs based on in-depth research in related experiments and articles, and develops a series of experiments around this method. Our answers to the questions raised in the introductory section are given here as follows.

- 1) By changing the leakage of LIF neurons, the linear leakage has better robustness compared to the no-leakage (IF) and conventional exponential leakage (LIF) models.
- 2) We demonstrate that the spiking rate decreases and the sparsity of the network increases when the leakage increases through deriving the process of integrating and firing spikes of LIF neurons based on linear leakage. In addition, we observe that the effect of the linear leakage parameter on the pulse firing rate somewhat affects the classification accuracy of the network.
- 3) It is experimentally demonstrated that linear leakage enhances the robustness of SNN absolute accuracy by 25% compared to conventional exponential leakage under specific parameters. Also, this robustness is generalizable across different datasets and network models.

This paper summarises and describes the advantages of the linear leakage method, in addition to its significant disadvantages. First, the linear leakage dynamical model bionicity is between IF and LIF models, and it can be regarded as a simplified form of LIF, but cannot meet the original intention of SNN to effectively simulate human brain neurons. Secondly, although linear leakage is robust, the error convergence rate and final accuracy are slightly inferior to those of LIF model, which are the problems of linear leakage that need to be solved urgently. Therefore, the subsequent research direction of us is to design experiments to find the balance between robustness and accuracy based on the current work, and to actively explore the influence of the inherent parameters of LIF neurons on the robustness of the model.

ACKNOWLEDGMENT

This work was supported by the National Key Research and Development Program of China (Grant No. 2018YFE0203801).

REFERENCES

- [1] Y. Wu *et al*, "Spatio-temporal backpropagation for training high-performance spiking neural networks," *Frontiers in Neuroscience*, vol. 12, (MAY), pp. 331-331, 2018;2017;.
- [2] H. Zheng *et al*, "Going Deeper With Directly-Trained Larger Spiking Neural Networks," 2020.
- [3] G. Srinivasan and K. Roy, "ReStoCNet: Residual Stochastic Binary Convolutional Spiking Neural Network for Memory-Efficient Neuromorphic Computing," *Frontiers in Neuroscience*, vol. 13, pp. 189-189, 2019.
- [4] S. Sharmin *et al*, "Inherent adversarial robustness of deep spiking neural networks: effects of discrete input encoding and non-linear activations," in *Computer Vision - ECCV 2020* Anonymous Cham: Springer International Publishing, 2020, pp. 399-414.
- [5] S. S. Chowdhury, C. Lee and K. Roy, "Towards understanding the effect of leak in Spiking Neural Networks," *Neurocomputing (Amsterdam)*, vol. 464, pp. 83-94, 2021.
- [6] R. El-Allami *et al*, "Securing deep spiking neural networks against adversarial attacks through inherent structural parameters," in 2021, . DOI: 10.23919/DAT51398.2021.9473981.
- [7] R. Güttig and H. Sompolinsky, "The tempotron: a neuron that learns spike timing-based decisions," *Nature Neuroscience*, vol. 9, (3), pp. 420-428, 2006 .
- [8] P. Dayan and L. F. Abbott, *Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems*. Cambridge, Mass: MIT Press, 2001.
- [9] Y. Wu *et al*, "Direct training for spiking neural networks: Faster, larger, better," in 2019, .
- [10] J. A. Perez-Carrasco *et al*, "Mapping from Frame-Driven to Frame-Free Event-Driven Vision Systems by Low-Rate Rate Coding and Coincidence Processing--Application to Feedforward ConvNets," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, (11), pp. 2706-2719, 2013.
- [11] P. U. Diehl *et al*, "Fast-classifying, high-accuracy spiking deep networks through weight and threshold balancing," in 2015, . DOI: 10.1109/IJCNN.2015.7280696.
- [12] B. Rueckauer *et al*, "Theory and Tools for the Conversion of Analog to Spiking Convolutional Neural Networks," 2016.
- [13] M. Zhang *et al*, "Supervised learning in spiking neural networks with noise-threshold," *Neurocomputing (Amsterdam)*, vol. 219, pp. 333-349, 2017.
- [14] M. Shafique *et al*, "Robust Machine Learning Systems: Challenges, Current Trends, Perspectives, and the Road Ahead," *IEEE Design and Test*, vol. 37, (2), pp. 30-57, 2020. . DOI: 10.1109/MDAT.2020.2971217.
- [15] J. Zhang *et al*, "Building robust machine learning systems: current progress, research challenges, and opportunities," in 2019, . DOI: 10.1145/3316781.3323472.
- [16] M. A. Neggaz *et al*, "Are CNNs Reliable Enough for Critical Applications? An Exploratory Study," *IEEE Design and Test*, vol. 37, (2), pp. 76-83, 2020. DOI: 10.1109/MDAT.2019.2952336.
- [17] J. Yang , P. Zhang and Y. Liu, "Robustness of classification ability of spiking neural networks," *Nonlinear Dynamics*, vol. 82, (1-2), pp. 723-730, 2015.
- [18] Y. Kuang *et al*, "A 64K-Neuron 64M-1b-Synapse 2.64pJ/SOP Neuromorphic Chip With All Memory on Chip for Spike-Based Models in 65nm CMOS," *IEEE Transactions on Circuits and Systems. II, Express Briefs*, vol. 68, (7), pp. 2655-2659, 2021.
- [19] S. Sharmin *et al*, "A comprehensive analysis on adversarial robustness of spiking neural networks," in 2019, . DOI: 10.1109/IJCNN.2019.8851732.



Jinming Che was born in 1998, China. Currently he is a student of School of Software & Microelectronics at Peking University. His current research interests are Spiking Neural network algorithm and Neural SLAM. He joins the Intelligent Hardware and System Laboratory from 2021.



Jian Cao, Doctor of Science, is an Associate Professor at Peking University. In recent years, he has devoted himself to the research of artificial intelligence and robotics. His research areas include: artificial intelligence algorithm optimization and its hardware and software implementation, neural network model compression and hardware acceleration, computer vision, edge computing, intelligent hardware and system design. He has led and participated in many projects of Ministry of Science and Technology, National Natural Science Foundation of China, international cooperation and enterprise cooperation. He has published more than 50 SCI and EI indexed journal and conference papers and applied for more than 20 patents.



Shuo Feng was born in 1996, Beijing. Currently she is an PhD student of School of Software & Microelectronics at Peking University. Her current research interests are Artificial Intelligence and brain-inspired intelligence. She joins Key Laboratory of Microelectronic Devices and Circuits (MoE) from 2021.



Jue Chen was born in 1999, China. Currently he is a master's student of Electronic Information at School of Software and Microelectronics, Peking University. His current research interests are Spiking Neural Networks, Brain-inspired Intelligence and Intelligent System. He joins the Intelligent Hardware and System Laboratory from 2021.



Yuan Wang is a professor and doctoral supervisor at Peking University, and secretary of the Party Committee of Integrated Circuits at Peking University. He is mainly engaged in research on new paradigm intelligent computing and integrated circuit chip design. As the project leader, he has undertaken more than ten important national scientific research tasks such as the National Key Research and Development Project and the Joint Key Fund of National Natural Science Foundation of China, published more than 200 academic papers, and obtained more than 60 patent authorizations. He is currently the director of the Graduate Student Branch of China Electronics Education Society, the expert committee member of China Automotive Chip Industry Innovation Strategy Alliance, the deputy editor-in-chief of New Generation Information Technology, the deputy editor-in-chief of Integrated Circuit Industry Series, the editor of Microelectronics, and has served as the chairman and member of the program committee of international academic conferences for more than 10 times.