Robustness of Spiking Neural Networks Based on Time-to-First-Spike Encoding Against Adversarial Attacks

Osamu Nomura, Member, IEEE, Yusuke Sakemi, Takeo Hosomi, and Takashi Morie, Member, IEEE

Abstract—Spiking neural networks (SNNs) more closely mimic the human brain than artificial neural networks (ANNs). For SNNs, time-to-first-spike (TTFS) encoding, which represents the output values of neurons based on the timing of a single spike, has been proposed as a promising model to reduce power consumption. Adversarial attacks that can lead ANNs to misrecognize images have been reported in many studies. However, the characteristics of TTFS-based SNNs trained using a backpropagation algorithm against adversarial attacks have not yet been clarified. In particular, the dependence of the robustness against adversarial attacks on spike timings has not been investigated. In this brief, we investigated the robustness of SNNs against adversarial attacks and compared it with that of an ANN. We found that SNNs trained with the appropriate temporal penalty settings are more robust against adversarial images than ANNs.

Index Terms—Spiking neural networks, time-to-first-spike encoding, adversarial attack.

I. INTRODUCTION

ARTIFICIAL neural networks (ANNs) [1] with multilayer structures have achieved a remarkable performance. Here, we define ANNs as neural networks in which input and output signals are expressed by analog values. Although ANNs can achieve a high accuracy, their high energy consumption is a major challenge caused by the execution of a large number of multiply-accumulate (MAC) operations. In addition, ANNs can misclassify images that humans easily recognize. In a previous study [2], perturbations generated based on trained models were shown to have resulted in ANNs misclassifying images that humans can easily recognize. This issue can cause fatal errors in safety-critical applications such as automated driving.

ANNs are primarily mathematical models that implement spike frequencies in biological neurons, whereas spiking neural networks (SNNs) have been proposed as models that implement spike generation directly and more closely mimic the human brain [3]. Although conventional ANNs are suitable for execution on digital computers that perform computations in a clock-synchronous manner, they consume a large amount of power because of the clock operation. By contrast, SNNs can achieve low power consumption by representing computations as asynchronously generated spikes. As a promising model for low-power-consumption operations, an SNN based on time-to-first-spike (TTFS) encoding was investigated in [4]. Unlike rate coding [3], which represents the information based on the frequency of multiple spikes, TTFS encoding represents the output values of neurons based on the timing of a single spike. In TTFS encoding, neurons ignore input spikes after their membrane potentials, corresponding to the results of the MAC operations, reach a threshold, and then generate a spike.

Unlike ANNs, SNNs based on TTFS encoding do not perform correct MAC operations. However, they are considered to be a more appropriate model of the visual system in the brain. In [5], the visual system in the brain was shown to process spikes that arrived earlier than others for image recognition. It was also pointed out that real-time recognition is difficult with rate coding that requires processing time to average the spikes. In [5], images were recognized over time by the SNN implementing the process of the primary visual cortex. Brain-inspired SNNs may exhibit different robustness to adversarial images because their information processing mechanism is different from that of ANNs, as explained above. Recognition processes based on spikes that arrive earlier (i.e., of higher importance) could lead to a faster and more robust recognition without being affected by small perturbations added to the images. Although several studies have been conducted on adversarial attacks on SNNs [6]–[9], adversarial white-box attacks on TTFS-based SNNs trained by using the backpropagation algorithm have not yet been reported. In particular, the effects of the early spikes on robustness against adversarial attacks has not been investigated. In this brief, we investigated the robustness of TTFS-based SNNs against adversarial white-box attacks and compared it with that of an ANN. Furthermore, we examined the dependence of SNN robustness on spike timings by adjusting the temporal penalty term for training.
Vth simplify future hardware implementations.

same neuron in the present input pattern, $w$

neuron model adopted in [4] is given by

A. SNN Model Based on Time-to-First-Spike Encoding

The time evolution of the membrane potential of the spiking neuron model adopted in [4] is given by

$$\frac{dv_{ij}^{(l)}}{dt} = \sum_{j=1}^{N_{ij}^{(l-1)}} w_{ij}^{(l)} \theta(t - t_{ij}^{(l-1)}),$$

$$\theta(x) = \begin{cases} 0 & (x < 0) \\ 1 & (x \geq 0) \end{cases},$$

where $v_{ij}^{(l)}$ is the membrane potential of the $i$th spiking neuron in the $l$th layer, $t_{ij}^{(l)}$ is the spike timing generated by the $i$th neuron in the $(l-1)$th layer to the $i$th neuron in the $l$th layer, and $N_{ij}^{(l)}$ is the number of neurons in the $l$th layer. We use a non-leaky spiking neuron model to simplify future hardware implementations.

The neuron generates a spike when its membrane potential reaches the threshold, $V_{th}$. After a spike is generated, the membrane potential is reset to a fixed voltage ($v_{ij} = 0$) and does not change again, preventing the neuron from firing twice. This feature is common to all neurons in the network. Fig. 1 depicts the evolution of the membrane potential on the timing of the received spikes and connection weights. Because the weights can be positive and negative, the spikes drive the membrane potential upward or downward.

In TTFS encoding, the data of input images are converted into temporal spike sequences. All the elements (pixels) of the vector are normalized into vector $\mathbf{x}$ as $0 \leq x_i \leq 1$. Then, the corresponding spikes are generated as

$$t_{ij}^{(0)} = t_{ij}^{(m)}(1 - x_i), \quad \text{if } x_i \geq 0,$$

where $t_{ij}^{(m)}$ is the maximum time window of the input spikes and set to 5 ms in the experiments. In a previous study [4], a spike was not generated when $x_i = 0$ in TTFS encoding. We modified the model such that a spike is generated when $x_i = 0$, as expressed in (3) without the exclusion of $x_i = 0$, because perturbations must be added to all pixels of an image in the adversarial attack method that we used.

B. Backpropagation Algorithm for the SNN

The following functions are used for the supervised learning of the SNNs [4]:

$$C(y^{(M)}, k; w) := L(y^{(M)}, k; w) + \gamma \frac{1}{2} R(t^{(M)}),$$

$$L(y^{(M)}, k; w) := - \sum_{i=1}^{N_{ij}^{(M)}} k_i \ln(S_i(t^{(M)})),$n

$$S_i(t^{(M)}) := \frac{\exp(-t_{ij}^{(M)})}{\sum_{i=1}^{N_{ij}^{(M)}} \exp(-t_{ij}^{(M)})},$$

$$R(t^{(M)}) := \sum_{i=1}^{N_{ij}^{(M)}} (t_{ij}^{(M)} - t_{ij}^{(Ref)})^2,$$

where $C(\cdot, \cdot, \cdot), L(\cdot, \cdot, \cdot), S_i(\cdot)$, and $R(\cdot)$ are the cost function, loss function, softmax function, and temporal penalty term, respectively.

The teacher labels are shown as vector $\mathbf{k}$, such that $k_i$ is 1 when the $i$th label is given and 0 otherwise. The weights of the entire network are shown as vector $\mathbf{w}$. The temporal penalty term, $R(t^{(M)})$, is defined as the temporal difference between the spike timing of the output neurons, $t^{(M)}$, and the reference spike, $t^{(Ref)}$. Here, $t^{(M)}$ is the vector of the spike timing of the neurons in the output layer. The coefficient $\gamma (> 0)$ controls the effect of the temporal penalty term. The temporal penalty term causes the cost function to specify the timing of the output spikes to approximate the reference timing, $t^{(Ref)}$.

III. METHODS

We compared the robustness of the SNN and the ANN, which have the same convolutional neural network (CNN) structure, against adversarial images. A summary of the network structure is shown in Table I.

The ANN was trained by using a backpropagation (BP) algorithm with Adam optimization and a cross-entropy loss function. The SNN was trained by using the BP algorithms described in Section II-B, Adam optimization, and cross-entropy loss function. For the SNN, we selected the following values of the temporal penalty term, $\gamma$: 10.0, 20.0, or 30.0 [ms], to evaluate the robustness. We set the coefficient $\gamma$ to 10.0, where the learning process is stable [4].

We used a fast gradient sign method (FGSM) attack to generate adversarial images [2]. The FGSM attack is designed to attack neural networks by leveraging the manner in which they

| Layer | Features | Inputs | Outputs | Weights |
|-------|----------|--------|---------|---------|
| Conv1 | 10       | 28x28  | 28x28   | 10x3x10 |
| max-pooling | - | 28x28 | 14x14 | - |
| Conv2 | 20       | 14x14  | 14x14   | 5x5x20 |
| max-pooling | - | 14x14 | 7x7 | - |
| Dense1 | 50       | 7x7x20 | 50     | 7x7x20x50 |
| Dense2 | 10       | 50     | 10     | 50x10  |

learn based on backpropagated gradients in the BP algorithm.
The attack generates perturbed images to maximize the loss
based on backpropagated gradients:

\[ \tilde{x} = x + \varepsilon \text{sign} \nabla_x J(\theta, x, y) \]  

where \( \tilde{x} \) is a perturbed image, \( x \) represents the original
image, \( \varepsilon \) denotes the pixel-wise perturbation coefficient, and
\( \nabla_x J(\theta, x, y) \) is the loss gradient with respect to the input image.
The amount of perturbation is controlled by the perturbation
coefficient, \( \varepsilon \). We set \( \varepsilon \) to the following seven values:

\[ \{0.00, 0.05, 0.10, 0.15, 0.20, 0.25, 0.30\} \]

In our experiments, we used MNIST and Fashion-MNIST as
the original images. Some samples of the perturbed images
generated for MNIST and Fashion-MNIST are presented in
Fig. 2

Random seed selection for weight initialization affects the
training results [10]. Because the perturbations for the images
depend on the trained weight sets, the effects of the adversarial
images can vary among the weight sets initialized with differ-  
ent seeds for training. We compared the robustness of the SNN
and the ANN against the adversarial images by performing a
statistical analysis for 10 different seed values for weight ini-

tialization. Welch’s t-test [11] was conducted to compare the
test accuracies of the SNN and ANN for each perturbation
coefficient.

IV. RESULTS

Fig. 3 shows the test accuracy averaged over 10 differ-

tent seeds of the SNN for each \( t(\text{Ref}) \) and the ANN for the adver-
sarial images of MNIST. Error bars representing standard deviations are shown only for the SNN with \( t(\text{Ref}) = 10 \) [ms] and the
ANN to avoid complicating the graph. Welch’s t-tests between
the SNN and ANN for each \( \varepsilon \) or each \( t(\text{Ref}) \) were performed.
Consequently, the averaged test accuracies of the SNN on
the MNIST images were significantly higher \( (p < 0.01) \) than those of the ANN for \( \varepsilon = \{0.05, 0.10, 0.15, 0.20, 0.25, 0.30\} \)
for \( t(\text{Ref}) = 10 \) [ms] and \( \varepsilon = \{0.10, 0.15, 0.20, 0.25\} \) for \( t(\text{Ref}) = 20 \) [ms].

Fig. 4 shows the test accuracy averaged over 10 differ-

tent seeds of the SNN for each \( t(\text{Ref}) \) and the ANN for the adver-
sarial images of Fashion-MNIST. Error bars representing standard deviations are shown only for the SNN with \( t(\text{Ref}) = 10 \) [ms] and the ANN to avoid complicating the graph. Welch’s t-tests between
the SNN and ANN for each \( \varepsilon \) or each \( t(\text{Ref}) \) were performed. Consequently, the averaged test accuracies of the SNN on
the Fashion-MNIST images were significantly higher \( (p < 0.01) \) than those of the ANN for \( \varepsilon = \{0.05, 0.10, 0.15, 0.20, 0.25, 0.30\} \)
for \( t(\text{Ref}) = 10 \) [ms], \( \varepsilon = \{0.05, 0.10, 0.15, 0.20, 0.25\} \) for \( t(\text{Ref}) = 20 \) [ms], and
\( \varepsilon = \{0.05, 0.10\} \) for \( t(\text{Ref}) = 30 \) [ms].
The results indicate that the robustness of the SNNs to adversarial images depends on average timing of output spikes.

V. DISCUSSION

As shown in Figs. 3 and 4, the averaged test accuracies of the SNN for MNIST and Fashion-MNIST are significantly higher than those of the ANN for most values of $\varepsilon$, particularly when trained with a smaller reference timing, $t_{(\text{Ref})}$. These results indicate that SNNs could be more robust against adversarial images than ANNs when trained with appropriate temporal penalty settings.

We hypothesize that the robustness of the SNN against adversarial images is caused by the TTFS encoding model. In the model, when the membrane voltage reaches the threshold, the neurons send a spike to the next layer. After sending the spike, the neuron ignores spikes from the previous layer. Figs. 5 and 6 show the spike timing of neurons (top) and time evolutions of the membrane potentials (bottom) trained by using the temporal penalty term, $t_{(\text{Ref})} = 10$ [ms] and $t_{(\text{Ref})} = 30$ [ms], for Fashion-MNIST. In the spike timing plots, the horizontal axis represents the time in milliseconds, and the vertical axis represents the index of neurons for each layer. Numerous neurons fire before receiving some spikes from the previous layers, which is a necessary condition for performing nonlinear information processing as explained in [4].

The accuracy of the original image ($\varepsilon = 0.00$) for the ANN is slightly higher than that for the SNN on both MNIST and Fashion-MNIST, as shown in Figs. 3 and 4. This result could have been caused by the effects of some neurons firing before receiving input spikes on TTFS encoding. By contrast, the SNN trained with the smaller reference timing, $t_{(\text{Ref})}$, is more robust against adversarial images than the ANN. In an FGSM attack, adversarial perturbations are distributed among all pixels in an image. TTFS encoding can avoid accumulating the adversarial effects by ignoring some of input spikes. More specifically, SNNs are more likely to focus on only a salient subset of input spikes when a smaller reference timing is chosen. In most cases, the basic recognition accuracy can be improved in the design phase by adjusting the network settings. By contrast, as dealing with adversarial attacks in real time under fixed settings is difficult, SNNs are considered to have an advantage over ANNs in this regard.
In [5], the characteristics of the early visual cortex were represented as the properties of the SNN, and the construction of the images was completed over time in the SNN. The hypothesis that the recognition process in the human brain starts from spikes that fire earlier and determines the pre-conceived notion of the whole image may lead to a robust recognition that is unaffected by small perturbations in the images. In other words, the “time-dependent input,” which does not occur in ANNs, may be the cause of human-like recognition. We suppose that human brains can obtain robust recognition by exploiting these time dependencies.

Many methods to resist adversarial attacks for deep neural networks (DNNs) models have been proposed [12]–[16]. Our study compared TTFS-based SNN models to the ANN models without any resisting methods against adversarial attacks. We set this brief as the first trial to investigate the dependence of the robustness against adversarial white-box attacks on TTFS-based SNNs trained by using the backpropagation algorithm. We have found that SNNs trained with the appropriate temporal penalty settings are more robust against adversarial images than ANNs. However, it is still unclear whether SNNs are more robust even when the resisting methods are introduced into the model. We plan to investigate the dependence of robustness against adversarial attacks on spike timings compared to ANNs that introduce resisting methods against adversarial attacks as the next step.

VI. CONCLUSION

We compared the robustness of TTFS-based SNNs and ANNs against adversarial images. The results indicated that SNNs trained with appropriate temporal penalty settings were more robust to adversarial images than ANNs. This suggests that SNNs will reflect aspects of the characteristics of human recognition.

REFERENCES

[1] B. Richards et al., “A deep learning framework for neuroscience,” Nat. Neurosci., vol. 22, pp. 1761–1770, Oct. 2019.
[2] I. J. Goodfellow, J. Shlens, and C. Szegedy, “Explaining and harnessing adversarial examples,” in Proc. 3rd Int. Conf. Learn. Represent., 2015, pp. 1–11.
[3] A. Daniel, H. Julian, M. Etienne, and K. Alois, “A survey of encoding techniques for signal processing in spiking neural networks,” Neural Process. Lett., vol. 53, pp. 4693–4710, Jul. 2021.
[4] Y. Sakemi, K. Morino, T. Morie, and K. Aihara, “A supervised learning algorithm for multilayer spiking neural networks based on temporal coding toward energy-efficient VLSI processor design,” IEEE Trans. Neural Netw. Learn. Syst., early access, Jul. 19, 2021, doi: 10.1109/TNNLS.2021.3095068.
[5] S. Thorpe and J. Gautrais, “Rapid visual processing using spike asynchrony,” in Advances in Neural Information Processing Systems, M. C. Mozer, M. Jordan, and T. Petsche, Eds., vol. 9, Cambridge, MA, USA: MIT Press, 1997, pp. 901–907.
[6] A. Marchisio, G. Nana, F. Khalid, M. A. Hanif, M. Martina, and M. Shafiique, “Is spiking secure? A comparative study on the security vulnerabilities of spiking and deep neural networks,” in Proc. Int. Joint Conf. Neural Netw. (IJCNN), 2020, pp. 1–8.
[7] S. Sharmin, N. Rathi, P. Panda, and K. Roy, “Inherent adversarial robustness of deep spiking neural networks: Effects of discrete input encoding and non-linear activations,” in Proc. 16th Eur. Conf. Comput. Vis. (ECCV), vol. 12374, 2020, pp. 399–414.
[8] S. Sharmin, P. Panda, S. S. Sarwar, C. Lee, W. Ponghiran, and K. Roy, “A comprehensive analysis on adversarial robustness of spiking neural networks,” in Proc. Int. Joint Conf. Neural Netw. (IJCNN), 2019, pp. 1–8.
[9] L. Liang et al., “Exploring adversarial attack in spiking neural networks with spike-compatible gradient,” IEEE Trans. Neural Netw. Learn. Syst., early access, Sep. 2, 2021, doi: 10.1109/TNNLS.2021.3106961.
[10] D. Picard, “Torch.manual_seed(3407) is all you need: On the influence of random seeds in deep learning architectures for computer vision,” 2021, arXiv:2109.06203.
[11] M. Delacre, D. Lakens, and C. Leys, “Why psychologists should by default use Welch’s t-test instead of student’s t-test,” Int. Rev. Social Psychol., vol. 30, no. 1, pp. 92–101, 2017.
[12] D. Warde-Farley and I. Goodfellow, “Adversarial perturbations of deep neural networks,” in Perturbations, Optimization, and Statistics. Cambridge, MA, USA: MIT Press, 2017, pp. 311–342.
[13] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami, “Distillation as a defense to adversarial perturbations against deep neural networks,” in Proc. IEEE Symp. Security Privacy (SP), 2016, pp. 582–597. [Online]. Available: https://doi.ieeecomputersociety.org/10.1109/SP.2016.41
[14] W. Xu, D. Evans, and Y. Qi, “Feature squeezing: Detecting adversarial examples in deep neural networks,” in Proc. Netw. Distrib. Syst. Security Symp. (NDSS), 2018, pp. 1–15.
[15] R. Theagarajan, M. Chen, B. Bhanu, and J. Zhang, “ShieldNets: Defending against adversarial attacks using probabilistic adversarial robustness,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), 2019, pp. 6981–6989.
[16] X. Jia, X. Wei, X. Cao, and H. Foroosh, “ComDefend: An efficient image compression model to defend adversarial examples,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 6077–6085. [Online]. Available: https://doi.ieeecomputersociety.org/10.1109/CVPR.2019.00624