Improving the robustness of binarized neural network using the EFAT method

Dinh Van Ngoc[1], Bui Ngoc My[1], Trinh Quang Kien[2*]

[1]Academy of Military Science and Technology;
[2]Le Quy Don Technical University.
*Corresponding author: kien.trinh@lqdtu.edu.vn.
Received 9 September 2021; Revised 10 December 2021; Accepted 15 December 2021.
DOI: https://doi.org/10.54939/1859-1043.j.mst.csce5.2021.14-23

ABSTRACT

In recent years with the explosion of research in artificial intelligence, deep learning models based on convolutional neural networks (CNNs) are one of the promising architectures for practical applications thanks to their reasonably good achievable accuracy. However, CNNs characterized by convolutional layers often have a large number of parameters and computational workload, leading to large energy consumption for training and network inference. The binarized neural network (BNN) model has been recently proposed to overcome that drawback. The BNNs use binary representation for the inputs and weights, which inherently reduces memory requirements and simplifies computations while still maintaining acceptable accuracy. BNN thereby is very suited for the practical realization of Edge-AI application on resource- and energy-constrained devices such as embedded or mobile devices. As CNN and BNN both compose linear transformations layers, they can be fooled by adversarial attack patterns. This topic has been actively studied recently but most of them are for CNN. In this work, we examine the impact of the adversarial attack on BNNs and propose a solution to improve the accuracy of BNN against this type of attack. Specifically, we use an Enhanced Fast Adversarial Training (EFAT) method to train the network that helps the BNN be more robust against major adversarial attack models with a very short training time. Experimental results with Fast Gradient Sign Method (FGSM) and Projected Gradient Descent (PGD) attack models on our trained BNN network with MNIST dataset increased accuracy from 31.34% and 0.18% to 96.96% and 85.08%, respectively.

Keywords: BNNs; Binarized Neural Networks; Adversarial Attack; Adversarial Defence; Fast Adversarial Training.

1. INTRODUCTION

Nowadays, image and speech processing tasks have been applied to deep neural networks (DNNs) to achieve better results than traditional algorithms. With practical problems such as object detection, image classification, object segmentation, CNN can be performed with high accuracy. However, deep learning network models have the disadvantage that a large number of parameters (up to hundreds of millions of parameters) require storage memory and large computational power requirements. These models mostly use GPUs for deployment but at the expense of large power consumption. This is not suitable if not impossible when deployed on embedded devices such as robots, wearable devices, unmanned devices, ..., with limited power sources. Several studies have proposed BNN to solve two big problems of deep learning neural networks: saving memory and reducing computational requirements, thereby reducing energy consumption.

Besides, when deploying deep learning networks with visual objects such as images and sounds, the question arises whether deep learning neural networks are fooled or give false results while still being easily recognized by a human. This type of adversarial

attack was first proposed by Szegedy et al. [4], where adversarial patterns are generated by adding a small amount of noise to the input to fool the neural network. Deep learning causes the network to make incorrect predictions. Opposing patterns by images can be randomly encountered in nature, such as a camera image blurred by dust, an acquired image changed due to sunny, rainy weather,etc. Adversarial attack research then has two purposes. The first is to find the methods to generate the best adversarial patterns to fool the deep learning neural network, and the second is to study the defensive methods to strengthen the network. Adversarial attack research also helps to power deep learning neural networks that are needed for real-life applications.

Similar to other convolutional neural networks, the conventional BNNs are easily fooled by adversarial patterns because of the same high linearity [4]. Therefore, to apply BNN in practical applications, it is necessary to study the impacts of adversarial patterns on the network. Authors in [10] and [11] have investigated BNN networks with some adversarial attacks without proposing the defending methods. This work studies the adversarial attacks on BNNs and corresponding countermeasures based on methods proposed in [7]-[9]. These methods focus on improving the robustness of convolutional neural networks in general, and the results are evaluated by FGSM [6] and PGD [5] adversarial attack models. The results of the three basic methods are equivalent in terms of attack resistance, but the proposed fast adversarial training method (FAT) in [9] has the shortest training time of 12 minutes while that in [7] and [8] take 80 and 10 hours, respectively. In this study, we proposed the enhanced FAT method (EFAT) in which the disturbance coefficient is dynamically distributed instead of a constant to train the BNN network to increase the strength of the network against adversarial attacks.

The remaining of the paper are presented as follows: Section 2 of the paper will present the binary neural network, adversarial attack; Section 3 will present the FAT training method and propose Enhanced-FAT (EFAT) training scheme for the binary neural network; Section 4 will present the experimental results, conclusions are drawn in Section 5 followed by the future work discussion.

2. BACKGROUND

2.1. Binarized Neural Network

Deep learning neural network currently draws significant attention from the research community. The CNN network is the most popular network architecture applied to computer vision problems such as image classification, object detection. The CNN network often has a large number of parameters and calculations. The number of parameters ranges from a few hundred thousand to several hundred million for a very complex network. Performing calculations in such networks are also very power-consuming, thus, it is not be suitable for embedded devices such as mobile devices, wearable devices, robots, etc., with very tight constraints on resource and power budget.

BNN networks are proposed to enable EdgeAI implementation. The idea of BNN is to binarize both the input and the weight parameters of the convolutional layers and fully connected layers. It brings twofold benefits: reducing the memory requirement for storing the weights and simplifying the network calculations. Indeed, as the results of binarization, the multiplication and addition operation now could be simplified to be

only the bitwise operations of XNOR and POPCOUNT. It reduces memory by approximately 32 times and computation by approximately 58 times [1].

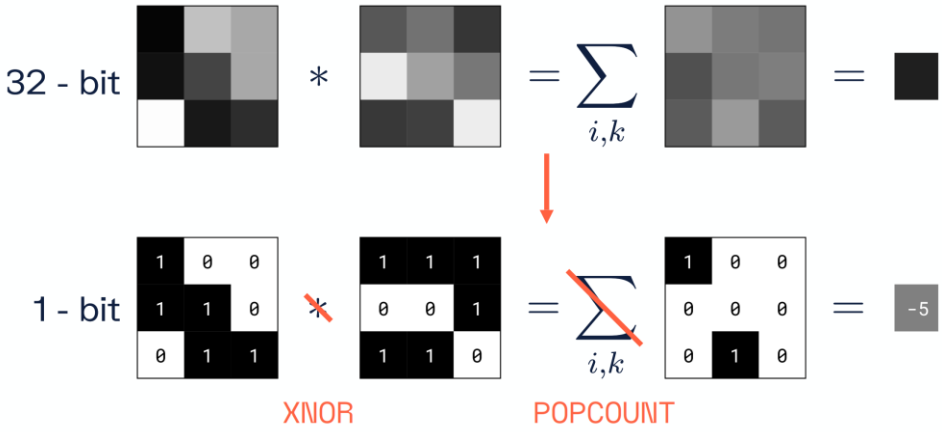

Figure 1. *Performing the binarized neural network calculation, quoted from [12].*

Inputs and parameters are binarized. They are approximated according to a rule to minimize information loss. With the weight parameter W≈α*sign(W), where α is the positive scaling factor, the input is evaluated similarly to I≈sign(I)*K, where K is the scale factor matrix. Then the convolution calculation of a layer in the network is shown in formula (1).

$$\boldsymbol{I} * \boldsymbol{W} \approx \left(\boldsymbol{sign}(\boldsymbol{I}) \circledast \boldsymbol{sign}(\boldsymbol{W})\right) \odot \boldsymbol{K}\boldsymbol{\alpha} \tag{1}$$

In binary representation, the convolutional operation is replaced by XNOR and POPCOUNT. This approximation method of the XNOR-Net network [1] has been extensively tested on data sets such as MNIST, CIFAR-10, and ImageNet resulting in reasonably good accuracy. In addition, other BNN networks such as BinaryNet [2] adopt slightly different approximation methods. The content of this paper focuses on study and evaluation using XNOR-Net.

2.2. Adversarial attack

The adversarial attack uses of adversarial examples to fool the neural networks. Adversarial patterns are introduced by adding a small amount of noise to the clean sample that is intended to trick the deep learning network into giving incorrect results. Adversarial attacks often target image and audio data [4]. After noise insertion, the data are not changed too much so that humans can still confidently recognize them, but the neural network almost wrongly recognizes them.

In figure 2, the image was classified as “panda” with 57.7% accuracy, after adding a small amount of adversarial to the last image was classified as “gibbon” with 99.3% accuracy. We can see that the last image does not change much compared to the original image, but the deep learning network has been tricked into making completely wrong decisions. To generate adversarial samples, researchers have come up with different methods to compute adversarial samples.

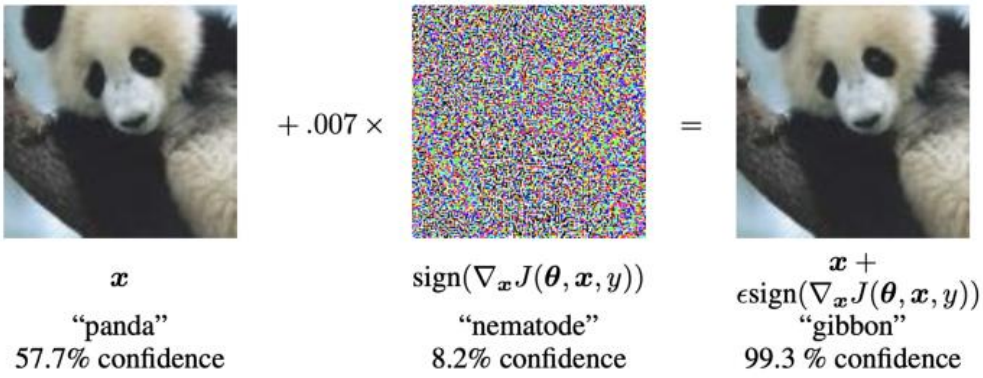

Figure 2. *Example of adversarial pattern, quoted from [5].*

Several algorithms for generating adversarial examples are given such as limited-memory Broyden–Fletcher–Goldfarb–Shanno (L-BFGS) [4], Fast Gradient Sign Method (FGSM) [5], the Basic Iterative Method (BIM)/Projected Gradient Descent (PGD) [6].

Goodfellow et al. [5] give FGSM adversarial examples, which are generated by the following formula (2).

$$\boldsymbol{x^{adv} = x + \varepsilon \times sign\left(\nabla_x J(\theta, x, y)\right)} \tag{2}$$

Where ε is the noise factor, J is the loss function, x is the input, y is the output, and x^{adv} is the input that has been jammed. The FGSM will calculate the derivative of the loss function as an additional quantity to the input causing error in the classification process.

Next, Kurakin et al. [6] introduced two algorithms that generate adversarial examples that improve FGSM. BIM executes FGSM in small steps and cuts them over a range of magnitudes ε in T cycles. At the N^{th} cycle the input value is calculated as follows:

$$\boldsymbol{x_0^{adv} = x, x_{N+1}^{adv} = Clip_{X,\varepsilon}\{X_N^{adv} + \alpha \times sign\left(\nabla_X J\left(\theta, X_N^{adv}, y\right)\right)\}} \tag{3}$$

With the constraint, αT=ε is the magnitude of the noise. PGD is a generalized version of BIM that does not force αT= ε. This approach is very computationally intensive but gives a powerful attack algorithm, therefore, it is often used to evaluate the strength of deep learning networks. PGD is shown in formula (4).

$$\boldsymbol{x_{N+1}^{adv} = Proj\{X_N^{adv} + \alpha \times sign\left(\nabla_X J\left(\theta, X_N^{adv}, y\right)\right)\}} \tag{4}$$

3. ENHANCING ROBUSTNESS OF BINARIZED NEURAL NETWORK

To improve the robustness of deep learning networks, adversarial training is typically adopted. This method strengthens the deep learning network against adversarial attacks by training the deep learning network with augmented adversarial examples. Goodfellow et al. [5] have empowered deep learning networks by training with both clean and adversarial samples.

$$\boldsymbol{J}(\boldsymbol{\theta}, \boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{\alpha J}(\boldsymbol{\theta}, \boldsymbol{x}, \boldsymbol{y}) + (\mathbf{1} - \boldsymbol{\alpha})\boldsymbol{j}\left(\boldsymbol{\theta}, \boldsymbol{x} + \boldsymbol{\varepsilon} \times \boldsymbol{sign}\left(\boldsymbol{\nabla}_{\boldsymbol{x}}\boldsymbol{J}(\boldsymbol{\theta}, \boldsymbol{x}, \boldsymbol{y})\right)\right) \quad (5)$$

Where the first component is the loss function of the clean sample, the second component is the loss function of the adversarial sample. To balance between the clean and noisy samples, the coefficient α is used (commonly α=0.5 is set). This method improves the strength of the deep learning network with FGSM patterns has a litter effect against stronger iterative attacks such as BIM and PGD.

Madry et al. [7] used PGD adversarial examples to train deep learning networks, and the evaluation results against FGSM, PGD, and CW attacks are acceptably good. However, this method requires a very long training time. For example, training with the Cifar-10 dataset takes about 80 hours [7].

Therefore, many studies find methods to improve adversarial training performance for deep learning networks. In which, there are two typical methods: "free" Adversarial Training [8] and Fast Adversarial Training (FAT) [9]. Both of these methods improve the training time of the deep learning network against PGD adversarial attacks. They achieve the same results (in terms of the network accuracy) as the adversarial training of PGD [7] but take 10 hours and 12 minutes on the Cifar-10 dataset [9]. Both approaches are based on adversarial training of FGSM with some improvements to achieve the desired results. Because of using adversarial training FGSM, the training time is significantly reduced.

BNN networks, like other deep learning networks are affected by adversarial attacks. When attacking the BNN network on the MNIST dataset, as we will detail later, the accuracy decreased from 99.19% to 31.34% with the FGSM attack, down to 0.18% with the PGD attack. This reduction makes the BNN network almost entirely fooled by the PGD attack.

To strengthen the BNN network, in this study, we adopt the FAT method in [9] in view that the method could deliver decent robustness against PGD attack with a much shorter training time. FAT is based on FGSM adversarial pattern training combined with random initialization for enhancing efficiency. Furthermore, based on the original FAT in [9], we propose a modification in resetting the threshold of the noise scale ε with a tendency for large random values to have a higher rate (EFAT -Modified FAT). Our proposed method exhibits better accuracy without adding extra training time. The pseudo-code of FAT and EFAT algorithms are combinedly presented below.

Algorithm 1: Train BNN network using FAT/EFAT method

For each epoch we perform the following steps:

for i=1..M do (M: the number of batches of the dataset)

$\theta \approx \beta * sign(\theta)$	*// θ : network parameters, β : scale factor*
$\varepsilon = 0.3$	*// Initialize for FAT*
$\varepsilon = Uniform/Triangular/Gauss\ (0.3)$	*// Initialize for EFAT*
$\delta = Uniform(-\varepsilon, \varepsilon)$	*// Initialize δ*

$\delta = \delta + \alpha \times sign\left(\nabla_{\delta} l(f_{\theta}(x_i + \delta), y_i)\right)$ *// Calculating according to FGSM*

$\delta = max(min(\delta, \varepsilon), -\epsilon)$

$x_i^{adv} = x_i + \delta$ *// Calculating the adversarial example*

$\theta = \theta - \nabla_{\theta} l\left(f_{\theta}\left(x_i^{adv}\right), y_i\right)$ *// Update parameters for network*

In algorithm 1, ε is fixed (for FAT) or randomized according to some pre-defined distribution (EFAT) before initializing δ by taking it randomly in the interval [-ε, ε]. Then, δ is calculated according to the FGSM plus the previous random initialization. The factor δ is intentionally tied within the allowed range (-ϵ, +ϵ) to avoid excessive noise added to the clean samples. Furthermore, we use adversarial samples to train the network by calculating feedforward and backpropagation before updating the network parameters with an optimizer such as SGD or ADAM. Parameters such as ε and α are variable parameters and will affect the noise δ generated. These adversarial training algorithms will be used for training and evaluation of the XNOR-Net network in the following Section.

4. EXPERIMENTS

4.1. Adversarial Attacks on the BNN

As mentioned earlier, we use the XNOR-Net network for evaluation. This is the first binary neural network that successfully trains with datasets such as MNIST, Cifar-10, and ImageNet [3]. The practical BNN model is the LeNet-5 network. This is a network model with one input layer, two convolutional layers, and two fully connected layers with a total number of parameters are 431220. The test dataset is MNIST with 60000 handwritten alphanumeric images from 1 to 10, of which 50,000 samples are used for training and 10,000 are used for testing. In the framework of the article, the author focuses on MNIST data to evaluate the defense ability of the original XNOR-Net and the trained network with FAT and EFAT under adversarial attack using FGSD and PGD samples.

Before executing training with FAT, we evaluated the strength of the XNOR-Net network with two attack schemes: FGSM (ε=0.3) and PGD (ε=0.3, α=0.375, iter=40). These are two attacks with commonly given parameters to evaluate. In addition, parameters can be increased or decreased to obtain different attack samples.

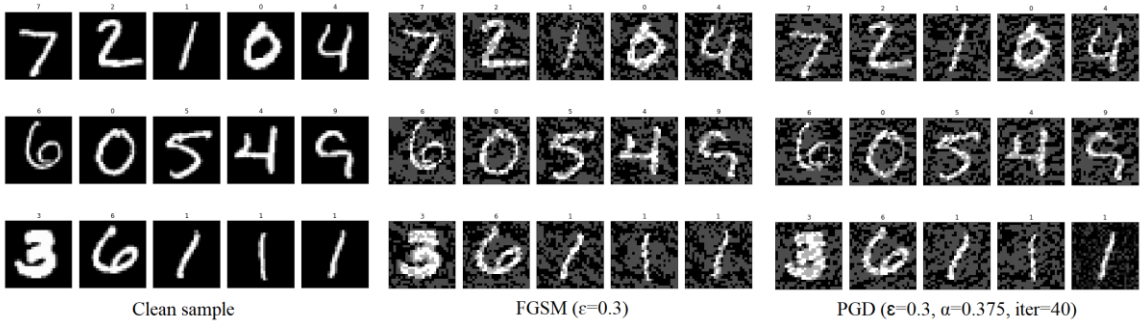

Figure 3. *Input data samples, (a) clean sample, (b) FGSM, and (c) PGD.*

After adding noise (figures 3b and 3c), we can see that the samples are easily distinguishable to the naked eye (i.e., could not fool humans) even with strong adversarial attacks like PGD. But with neural networks in general and BNN networks have been fooled and give results that are much worse than the original. Note that when the noise level with $\varepsilon > 0.3$, the output image quality is often very low, making it difficult for the human eye to observe, so in this study, we limit $\varepsilon \leq 0.3$.

It can be seen that with adversarial samples, the XNOR-Net network FGSM has significantly reduced its accuracy to only 31.34%; For stronger adversarial samples, PGD, XNOR-Net gives almost completely wrong results, although it is still easy to distinguish with the naked eye (figure 3c). To increase the robustness of XNOR-Net, we used the FAT/EFAT method to train the XNOR-Net network.

4.2. Improving Accuracy of BNN by the FAT

We conducted training XNOR-Net using the FAT method with MNIST dataset according to algorithm 1 and re-evaluated the accuracy of the trained network. The major results are reported in Table 1. In this experiment, we trained by the original FAT algorithm in [9] with fixed $\varepsilon = 0.3$, $\alpha = 0$ for 50 epochs, the batch size is 50 for training and 100 for testing. The training and inference models have been developed using Pytorch.

From the results, it can be seen that the XNOR-Net network exhibits much better defense ability against both FGSM and PGD attacks. These results are well in line with the results reported for applying FAT for CNNs network. The major difference here is that FAT uses a random-initialized FGSM adversarial sample, as opposed to fixed initial in FGSM training, that greatly reduces the training time without degrading the network accuracy against PGD attacks. In particular, the training time in our setup (Intel Core i5/ 16GB RAM/ GPU NVIDIA RTX5000) takes about 5.8 minutes. According to Table 1, the accuracy of the XNOR-Net, trained by FAT network has been significantly improved with the FGSM attack ($\varepsilon = 0.3$), i.e. increases from 31.34% to 96.42%; with the PGD attack, the original XNOR-net model gives almost wrong classification results, but with FAT training, the model accuracy restores to 80.07%, i.e., is significantly improved.

Table 1. *Evaluation of XNOR-Net trained by FAT/EFAT under PGD and FGSM attacks.*

	Clean sample	**FGSM (ε=0.3)**	**PGD (ε=0.3, α=0.375, iter=40)**
XNOR-Net	99.19%	31.34%	0.18%
XNOR-Net with FAT (**ε=0.3**) [9]	97.86%	96.42%	80.07%
XNOR-Net with EFAT (*Uniform Dist.)*	**98.92%**	**96.96%**	**85.08%**
XNOR-Net with EFAT (*Triangular Dist.)*	98.47%	94.01%	85.03%
XNOR-Net with EFAT (*Gauss Dist.)*	98.49%	94.97%	80.17%

Furthermore, we also investigate FGSM and PGD attacks by changing the noise level ε from 0 (clean samples) to a maximum level of 0.35 with step 0.05, the results are plotted in fig. 4.

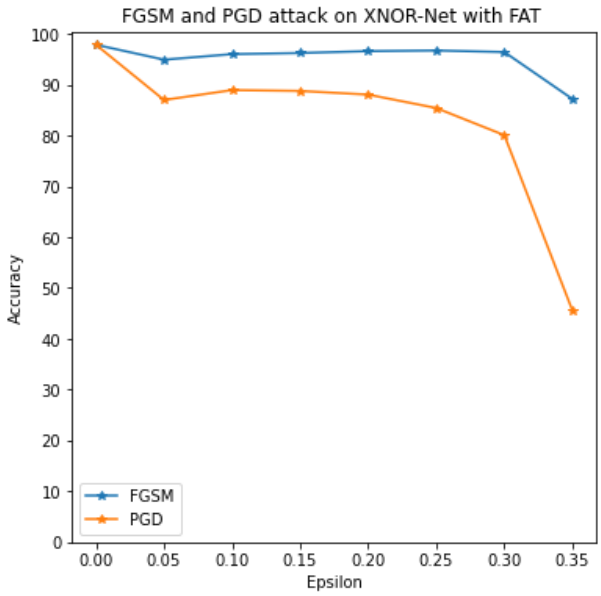

Figure 4. *FGSM and PGD adversarial attack on XNOR-Net.*

From the figure, it can be seen that the FGSM attack has little impact on the trained network, and the accuracy is almost level off, about above 96% for $\varepsilon \leq 0.3$. The accuracy degrades sharply to ~87% with $\varepsilon = 0.35$, but it is an excessively high level of noise (i.e., even humans find it difficult to recognize). Similarly, the FAT-trained network shows good resistance against PGD attacks. However, the PGD attack degrades the accuracy of the model by almost 12% for $\varepsilon \leq 0.3$ and falls to about 45% with $\varepsilon = 0.35$. This observation may indicate the true limitation of the XNOR-net in practice, or we still need a better training method to restore the level of accuracy to the maximum.

4.3. Improving the robustness of BNN by EFAT method

In the following, we conduct a modified training scheme from FAT, namely Enhanced-FAT or EFAT, to evaluate the training process for XNOR-Net. As shown in Algorithm 1, in the proposed EFAT, ε is randomly populated by several different methods: Uniform, Triangular and Gaussian distribution so that the randomness is concentrated at ε=0.3 and in the near vicinity. The reason we chose many points ε near 0.3 is to cause the opposite samples with large noise coefficients often because of a higher degree of confusion and make it difficult for humans. The distribution chart of ε values for the cases is shown in Fig 5.

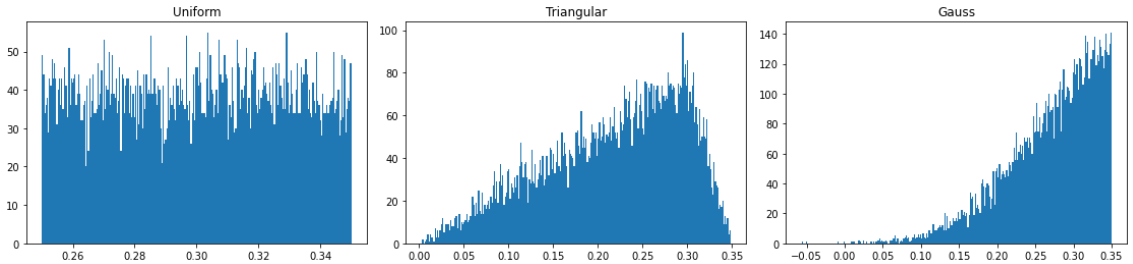

Figure 5. *Histogram of 3 distributions for ε (a) Uniform, (b) Triangular and (c) Gauss.*

The results of training randomly selected by 3 different randomization methods are presented in Table 1.

The results in Table 1 show that whether the XNOR-Net network trained with the FAT method has a fixed (FAT) or variable (EFAT) of ε factor, it increases its strength against adversarial attacks significantly compared to when not trained with FAT. For FGSM, then enhancement of both EFAT and FAT possibly reach the saturated level therefore, we do not see much difference, except training with Gaussian and Triangle distributions is not as good as the Uniform one (lower by 2-3%). However, the EFAT with all randomization methods shows an improvement of ~5% in the classification accuracy when testing with PGD samples. This could be empirically explained by the fact that the EFAT model has been trained with more level of randomness than the conventional FAT. Comparing the different randomization methods, it can be seen that training with uniform and Triangle distributed ε gives slightly better results than that of Gaussian distributed ε. If taking the training effect against both FGSM and PGD samples, the XNOR-Net trained with EFAT using Uniform randomization gives the best results than training with fixed ε and other randomization methods.

5. CONCLUSION

In this paper, we have used the EFAT training method to strengthen the BNN network against adversarial attacks. The experimental results prove that BNN networks, as well as normal CNNs could be easily fooled with adversarial examples. We showed that adversarial training greatly improves the robustness of XNOR-net BNNs against both FGSM and PGD attacks. First, by adopting the FAT training method in [9] for BNN, it could effectively improve the XNOR-net BNN accuracy from 31.34% to 96.42% for FGSM and from 0.18% to 80.07% samples. Furthermore, we proposed EFAT, which is FAT with a random value of the noise factor ε according to Uniform, Triangle, and Gaussian distributions. Regardless of the distribution form, the results demonstrate that the EFAT is superior to the conventional FAT method by about 5% for PGD attacks. The results also indicate that the EFAT with Uniform exhibits the best results against both FGSM and PGD adversarial attacks. In the future, we plan to extend this study for a more robust training method and/or to expand the study we implemented on larger networks and data sets.

Acknowledgement: *This research is funded by Vietnam National Foundation for Science and Technology Development (NAFOSTED) under grant number 102.01-2018.310.*

REFERENCES

[1]. Mohammad Rastegari, Vicente Ordonez, Joseph Redmon, and Ali Farhadi. "*XNOR-Net: ImageNet Classification Using Binary Convolutional Neural Networks.*" ECCV (2016).

[2]. Matthieu Courbariaux, Itay Hubara, Daniel Soudry, Ran El-Yaniv and Yoshua Bengio. "*Binarized Neural Networks: Training Deep Neural Networks with Weights and Activations Constrained to +1 or -1.*" arXiv: Learning (2016).

[3]. Mohammad Rastegari, Vicente Ordonez, Joseph Redmon, and Ali Farhadi. "*Enabling AI at the edge with XNOR-networks*". Communications of the ACM (2020), **Vol.63**, pp. 83-90.

[4]. Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. "*Intriguing properties of neural networks.*" arXiv preprint arXiv:1312.6199 (2013).

[5]. Alexey Kurakin, Ian Goodfellow, and Samy Bengio. "*Adversarial examples in the physical world.*" (2016).
[6]. Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. "*Explaining and harnessing adversarial examples.*" arXiv preprint arXiv:1412.6572 (2014).
[7]. Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. "*Towards deep learning models resistant to adversarial attacks.*" arXiv preprint arXiv:1706.06083 (2017).
[8]. Ali Shafahi, Mahyar Najibi, Amin Ghiasi, Zheng Xu, John Dickerson, Christoph Studer, Larry S. Davis, Gavin Taylor, and Tom Goldstein. "*Adversarial training for free!.*" arXiv preprint arXiv:1904.12843 (2019).
[9]. Eric Wong, Leslie Rice, and J. Zico Kolter. "*Fast is better than free: Revisiting adversarial training.*" arXiv preprint arXiv:2001.03994 (2020).
[10]. Angus Galloway, Graham W. Taylor, and Medhat Moussa. "*Attacking binarized neural networks.*" arXiv preprint arXiv:1711.00449 (2017).
[11]. Manoj Rohit Vemparala, Alexander Frickenstein, Nael Fasfous, Lukas Frickenstein, Qi Zhao, Sabine Kuhn, Daniel Ehrhardt et al. "*BreakingBED--Breaking Binary and Efficient Deep Neural Networks by Adversarial Attacks.*" arXiv preprint arXiv:2103.08031 (2021).
[12]. Lukas Geiger. "*Binarized Neural Networks on microcontrollers.*" tinyML Talks (2021).

TÓM TẮT

CẢI THIỆN SỨC MẠNH CHO MẠNG NƠ-RON NHỊ PHÂN SỬ DỤNG PHƯƠNG PHÁP EFAT

Trong những năm gần đây với sự bùng nổ các nghiên cứu về trí tuệ nhân tạo, mô hình học sâu dựa trên mạng nơ-ron tích chập (CNN) là một trong những kiến trúc hứa hẹn cho ứng dụng thực tế nhờ vào độ chính xác cao. Tuy nhiên, mạng CNN đặc trưng bởi các lớp tích chập thường có số lượng tham số và yêu cầu tính toán lớn dẫn đến tiêu tốn năng lượng khi huấn luyện cũng như thực thi dự đoán. Để khắc phục nhược điểm đó, mô hình mạng nơ-ron nhị phân (BNN) được đề xuất. BNN biểu diễn nhị phân cho giá trị đầu vào và tham số của các lớp trong mạng giúp giảm yêu cầu về bộ nhớ và đơn giản hóa các phép tính toán trong khi duy trì được độ chính xác chấp nhận được. Do đó, BNN rất phù hợp để triển khai ứng dụng Edge-AI trên các thiết bị hạn chế về tài nguyên và năng lượng như thiết bị nhúng hoặc thiết bị di động. CNN và BNN đều thực thi các lớp biến đổi tuyến tính, vì vậy chúng đều dễ dàng bị đánh lừa bởi các mẫu đối nghịch. Trong công trình này, chúng tôi xem xét tác động của tấn công đối nghịch (Adversarial attack) với BNN và đề xuất giải pháp nâng cao độ chính xác của BNN trước các loại tấn công này. Cụ thể, chúng tôi sử dụng phương pháp huấn luyện đối nghịch nhanh cải tiến (Enhanced Fast Aversarial Training - EFAT) để huấn luyện cho mạng. Sử dụng phương pháp này giúp mạng BNN chống lại được các tấn công mạnh với thời gian huấn luyện ngắn. Kết quả thực nghiệm với tấn công Fast Gradient Sign Method (FGSM) và Projected Gradient Descent (PGD) trên mạng BNN với tập dữ liệu MNIST tăng độ chính xác từ 31.34% và 0.18% lên 96.96% và 85.08%.

Từ khoá: Mạng nơ-ron nhị phân; Tấn công đối nghịch; Phòng thủ đối nghịch; Huấn luyện đối nghịch nhanh.