Defend Deep Neural Networks Against Adversarial Examples via Fixed andDynamic Quantized Activation Functions

4 authors, including:

Adnan Siraj Rakin
Arizona State University
25 PUBLICATIONS   98 CITATIONS

Deliang Fan
University of Central Florida
107 PUBLICATIONS   961 CITATIONS

Defend Deep Neural Networks Against Adversarial Examples via Fixed and Dynamic Quantized Activation Functions

Adnan Siraj Rakin[1], Jinfeng Yi[2], Boqing Gong[3] and Deliang Fan[*1]

[1]Department of Computer Engineering, University of Central Florida
[2]JD AI Research, Beijing, China
[3]Tencent AI Lab, Bellevue, WA, USA

Abstract

Recent studies have shown that deep neural networks (DNNs) are vulnerable to adversarial attacks. To this end, many defense approaches that attempt to improve the robustness of DNNs have been proposed. In a separate and yet related area, recent works have explored to quantize neural network weights and activation functions into low bit-width to compress model size and reduce computational complexity. In this work, we find that these two different tracks, namely the pursuit of network compactness and robustness, can be merged into one and give rise to networks of both advantages. To the best of our knowledge, this is the first work that uses quantization of activation functions to defend against adversarial examples. We also propose to train robust neural networks by using adaptive quantization techniques for the activation functions. Our proposed Dynamic Quantized Activation (DQA) is verified through a wide range of experiments with the MNIST and CIFAR-10 datasets under different white-box attack methods, including FGSM, PGD, and C&W attacks. Furthermore, Zeroth Order Optimization and substitute model based black-box attacks are also considered in this work. The experimental results clearly show that the robustness of DNNs could be greatly improved using the proposed DQA.

1 Introduction

Deep Neural Networks (DNNs) have achieved great success in various tasks, including but not limited to image classification [1], speech recognition [2], machine translation [3], and autonomous driving [4]. Despite the remarkable progress, recent studies [5, 6, 7] have shown that DNNs are vulnerable to adversarial examples. In image classification, an adversarial example is a carefully crafted image that is visually imperceptible to the original image but can cause DNN model to misclassify as shown in figure 1. In addition to image classification, attacks to other DNN-related tasks have also been actively investigated, such as visual QA [8], image captioning [9], semantic segmentation [10], machine translation [11], speech recognition [12], and medical prediction [13].

There is a cohort of works on generating adversarial attacks and developing corresponding defense methods. The adversarial attacks can be grouped into two major categories: (i) *white-box attack* [5, 7], where the adversary has full access to the network architecture and parameters, and (ii) *black-box attack* [14, 15, 16],

---

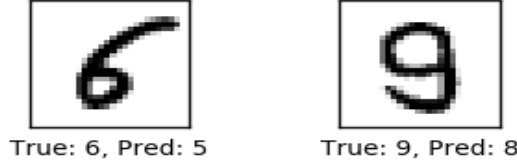[*]Corresponding Author: dfan@ucf.edu

Figure 1: Adversarial attack miss-classifying MNIST digits

where the adversary can access the input and output of a DNN but not its internal configurations. Many attack algorithms have been proposed to generate adversarial examples [5, 17, 18, 19, 20, 21, 22, 7]. Among them, the fast gradient sign method (FGSM) [6] is one of the pioneering and most popular white-box attack algorithms. It uses the sign of gradients with respect to the input to generate adversarial examples and is by far one of the most efficient attack algorithms. The projected gradient descent (PGD) is among the most powerful white-box attacks till date [23]. Besides, Carlini & Wagner (C&W) [7] attack is another powerful attack that can achieve nearly 100% attack success rate. In this work, we employ all these attack algorithms to evaluate the robustness of our proposed defense. We also conduct popular black box attacks to even further establish our claim of improved robustness [16, 24]. To achieve the same objective previously many defense methods have been proposed to defend against adversarial examples [25, 26, 27, 28]. Some of them work well against early attack methods (e.g., FGSM [6] and Jacobian saliency map attack (JSMA) [19]), but fail to deliver strong robustness against more powerful attacks [23, 7, 21].

Meanwhile, from another perspective, DNNs with increasing depth, larger model size, and more complex building blocks have been proposed to achieve better performance and perform more sophisticated tasks. For instance, Inception-ResNet [29] and DenseNet [30] have 96 and 161 layers, and 56.1 million and 29 million parameters, respectively. The large sizes prohibit them from deploying in resource intensive systems such as mobile phones. Fortunately, DNN models usually contain significant redundancy, and the state-of-the-art accuracy can also be achieved after model compression [31]. As one of the most popular model compression techniques, weight and activation quantization is widely explored to significantly shrink the model size and reduce the computational cost [32, 33, 34]. BinaryConnect [32] introduces gradient clipping and is the first binary CNN algorithm that yields close to the state-of-the-art accuracy on CIFAR-10. After that, both BWN in [33] and DoreFa-Net [34] show better or close validation accuracy on ImageNet dataset. We note that although weight quantization is more widely-used in reducing model size, quantization of activation functions can also make network more compact since the input bit-width of next layer is determined by the quantization level of activation functions. In this work, we mainly consider activation quantization as an effective defense methodology against adversarial attacks.

In this work, we show that the tracks of pursuing network compactness and robustness are likely to merge, although they seem to be independent. Specifically, we propose an activation quantization technique called Dynamic Quantized Activation (DQA), which treats thresholds of activation quantization function as another tuning parameter set to improve the robustness of DNN in adversarial training. In this method, such threshold parameters are adapted in adversarial training process and play an important role in improving network robustness against adversarial attacks. In this work, we first test fixed activation quantization method under a wide range of attacks. It shows empirically that, by quantizing the output at each activation layer, the effect of adversarial noises can be greatly diminished. To the best of our knowledge, this is the very first work that proposes to use quantized activation function to defend against adversarial examples. Then, we show that the proposed DQA can further supplement this claim by including more learnable parameters

2

during adversarial training process. By doing so, the compactness and robustness of neural networks are simultaneously achieved.

2 Related Works

Recent development of some of the strongest [23, 7, 21, 16] attacks have exposed the underlying vulnerability of DNN more than ever. As a result, a series of work have been conducted towards the development of robust DNN [35, 15, 23, 36, 37, 38, 39, 40, 41, 42, 43]. Among them, training the model with adversarial examples gives the initial breakthrough towards universal robustness of DNN [23]. Most of later works have followed this path to supplement their defense with adversarial training. Other defenses have focused on transforming the input to DNN [44, 39]. However, input transformation seems to suffer from obfuscated gradient issue [21]. Recent development of Generative Adversarial Network (GAN) have showed promising signs for GAN networks to work as a potential defense model [37, 35]. One of the most recent defense that is similar to our work is L2 non-expansive neural networks [36]. The principle of this defense is to suppress the noise inside DNN and stop it from becoming large as it goes through the network. Our idea behind introducing quantization is similar. However, due to the hard constraints of L2 non-expansive defense, it suffers from poor clean data accuracy. On the other hand, previous works have shown quantization of activation does not hamper the clean data accuracy much [34] compared to L2 non-expansive defense.

3 Fixed and Dynamic Activation Quantization

In this section, we first describe the fixed quantization technique and its role in working as a defense against adversarial examples. Then, we present the working principle of our proposed Dynamic Quantized Activation as an effective methodology to improve the robustness of DNN against adversarial examples.

3.1 Fixed Quantization

Fixed Method: We start from a fixed activation quantization method that shares some similarities with Dorefa-Net [34]. The inputs to quantization function are first passed through a tanh function that maps the input x to range (-1, 1):

$$
\begin{aligned}
y = tanh(x) &= \frac{sinh(x)}{cosh(x)} \\
&= \frac{e^x - e^{-x}}{e^x + e^{-x}}
\end{aligned}
\tag{1}
$$

We then shift the output y to range of $(0, \ 1)$ using the following function:

$$
y' = f(y) = \frac{y}{2\max(|y|)} + \frac{1}{2}.
\tag{2}
$$

Where, $max(|y|)$ denotes the maximum values in the y matrix without sign. Then an n-bit quantization is achieved by:

$$
y_n = f(y') = 2 \times [\frac{1}{2^n - 1} \times round[(2^n - 1) \times (y')]] - 1,
\tag{3}
$$

In this way, the output tensors will only contain 2^n discrete levels in the range of -1 to +1. For example, when $n = 2$, the outputs will be quantized to 4 discrete levels -1, -0.33, 0.33,and 1.

A Simple Explanation   Although the qunatization method is very simple, they work surprisingly well in improving the robustness of neural networks as will be discussed in the experiment section. Here, we first provide a simple explanation of this phenomenon.

In deep neural networks, a small change in the inputs will usually cause significant changes in the outputs. Existing works [36, 6] have shown that this is one of the main reasons why adversarial examples exist. To see this, let's consider a simple neuron

$$y = f(\mathbf{x}^\top \mathbf{w}), \tag{4}$$

where $\mathbf{x} \in \mathcal{R}^d$ is the input signal, $\mathbf{w} = [w_1, \cdots, w_d]^\top$ is the neuron weights, and $f(\circ)$ denotes the widely-used rectified linear unit (ReLU) activation function

$$ReLU(z) = \begin{cases} z, & \text{if } z \geq 0 \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

When adding a small perturbation $\epsilon \in \mathcal{R}^d$ to the input vector $\mathbf{x}$, the neuron output becomes

$$\tilde{y} = f[(\mathbf{x} + \epsilon)^\top \mathbf{w}] = f[\mathbf{x}^\top \mathbf{w} + \epsilon^\top \mathbf{w}] \tag{6}$$

When each element of $\mathbf{x}^\top \mathbf{w} + \epsilon^\top \mathbf{w}$ is larger than 0, this makes the neuron output to grow by as large as $\epsilon^\top \mathbf{w}$. If the average magnitude of the weight vector element is a and $\|\epsilon\|_\infty = r$, then the neuron output will grow by $r \times d \times a$, which is non negligible when the input dimensionality is large. Since a deep neural network usually contains a large number of neurons and layers, a small change in the inputs usually causes a very significant change in the outputs.

Indeed, this observation also motivates some recent defense techniques. For example, [36] proposes an ℓ_2-nonexpansive neural network that enforces a unit amount of change in the inputs can cause at most a unit amount of change in the outputs. However, this non expansive constraint is so strong that may hurt the expressibility and learnability of neural network [45]. As a result, its performance on clean data is largely sacrificed. In contrast, our method addresses the exploding output problem by using quantized activation functions, a technique widely-used in model compression. By doing so, the proposed method can simultaneously achieve two goals: (i) improving the robustness of neural networks; (ii) making neural networks more compact.

3.2   Dynamic Activation Quantization

Adversarial Training: Given a set of training images and lables $(\mathbf{x}, y)$ sampled from the distribution L and a loss function J, the adversarial training method aims to learn parameters γ to minimize the risk $R_{(\mathbf{x}, y)} \sim_L [J(\gamma, \mathbf{x}, y)]$. Typically, γ in a DNN consists of weights (W), biases (b) and other parameters like batch normalization layer parameters (z).

The first step in adversarial training is to choose an attack model to generate adversary examples. In this work, we employ the PGD attack since it can generate universal adversary examples among the first order approaches [23]. Using this model, for each image $\mathbf{x}$, the adversary example $\mathbf{x} + \epsilon$ within the bounded ℓ_∞ norm is generated. Similar to [17, 23, 46], given the generated adversarial examples, we aim to minimize the following empirical risk to improve the model robustness

$$\min R_{(\mathbf{x}, y)} \sim_L [\max J(\gamma, \mathbf{x} + \epsilon, y)]. \tag{7}$$

The optimal solution of the above min-max problem is achieved by tuning parameters such as weights and biases. According to our previous analysis, quantized activation functions may also improve the network

robustness. It motivates us to integrate the quantized activation functions into adversarial training, and treat the thresholds between different quantization levels as another set of tunable parameters to improve network robustness.

Dynamic Quantized Activation: In our previous method, the thresholds between different quantization levels are fixed and uniformly distributed. Here we propose a dynamic activation quantization method in which the thresholds for different discrete levels are tunable parameters in adversarial training. For n-bit quantized activation function, it has 2^n output discrete levels and $2^n - 1$ tunable thresholds. Let $m = 2^n$, then the quantization will have m-1 threshold values t_1, t_2, .., $t_{m/2}$, ..., t_{m-2} and t_{m-1}. Then, we make m level quantization function as:

$$
\begin{aligned}
f(x) \;=\; & 0.5 \times [sgn(x - t_{m-1}) + \\
& \sum_{i=m-1}^{m/2+1} t_i(sgn(t_i - x) + sgn(x - t_{i-1})) \\
& + \sum_{i=m/2}^{2} t_{i-1}(sgn(t_i - x) + sgn(x - t_{i-1})) - \\
& sgn(t_1 - x)],
\end{aligned}
\tag{8}
$$

where sgn denotes the sign function. For example, when $n = 1$ and 2, the 1-bit and 2-bit dynamic quantizations are:

$$
f(x) = 0.5 \times [sgn(x - t_1) - sgn(t_1 - x)]
$$

and

$$
f(x) = \begin{cases}
-1, & \text{if } x < t_1 \\
t_1, & \text{if } t_1 < x < t_2 \\
t_3, & \text{if } t_2 < x < t_3 \\
+1, & \text{if } x > t_3,
\end{cases}
\tag{9}
$$

respectively. Therefore, equation (8) adds a new set of learnable parameters $T := [t_1, t_2, ..., t_{m-1}]$. Since they can be learned independently w.r.t the existing DNN parameters γ, the objective function (5) then becomes:

$$
\min R_{(\mathbf{x}, y)} \sim_L [\max \; J([\gamma, T], \mathbf{x} + \epsilon, y)],
\tag{10}
$$

which is more flexible and desirable than the previous approach that can only tune DNN parameters γ.

To illustrate the effect of dynamic activation quantization, as shown in figure 2, we plot the data distribution after the first activation layer of a sample network for one random input patch in CIFAR10 dataset. For both 1-bit and 2-bit activation quantization, we show the distribution first using fixed quantized activation. Then we replace the fixed activation with dynamically trained activation to indicate the difference of data distribution. It is clear that fixed quantized levels are uniformly distributed between -1 to +1. On the other hand, in dynamic activation quantization method, discrete levels are adjusted and the frequency at each quantized levels are not even as well. Such adjusted distribution reflects the quantization function's involvement in the adversarial training and its tuning to the network performance. Its efficacy in improving network robustness will be verified in later experimental section.
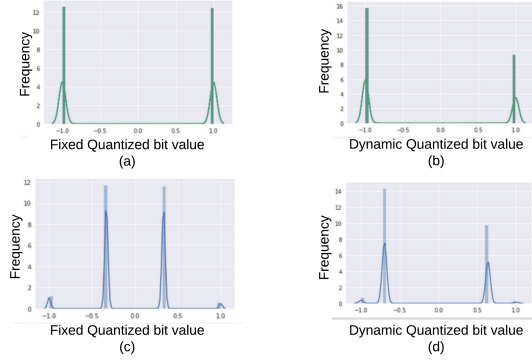
Figure 2: Activation layer distributiion: a)1-bit fixed threshold b)1-bit dynamic threshold c)2-bit fixed threshold and d) 2-bit dynamic threshold

4 Experimental Setup

Datasets and networks. We first test LeNet [47] on the MNIST dataset [48]. MNIST is a set of handwritten digit images with 60,000 training examples and 10,000 testing examples. Besides, we test ResNet-18 [49] on the CIFAR-10 dataset [50]. CIFAR-10 contains 60,000 RGB images of size 32x32. Following the standard practice, we use 50,000 examples for training and the remaining 10,000 for testing. The images are drawn evenly from 10 classes. Since we do not have a validation set for our experiments we run the simulations five times for each case to report the average of them.

Attack methods. To evaluate the robustness of our method, we employ multiple powerful white-box attack methods, including the projected gradient descent (PGD) attack [23], the fast gradient sign method (FGSM) [6], and the Carlini & Wagner (C&W) attack [7]. We also conduct state-of-the-art black box attacks in a seperate section as well [24, 16] to test our defenses robustness. The parameters of all the attack algorithms are tuned to give the best attack performance.

Defense techniques and baselines. We conduct experiments for our defense including adversarial training. Because the very essence of DQA working as a defense requires the model to be trained with adversarial training. In adversarial training, the adversarial examples are generated by the PGD attack algorithm. Before mixing adversarial examples in the training, we train the model with clean data for some initial epoch of training.

We compare our defense result with the original PGD trained defense model [23] that uses ReLu as the full-precision activation function. Later, we also show how our defense compares with other state-of-the-art defenses in defending the popular attacks.

5 White Box Attack

We test our idea of introducing quantized activation as a defense against adversarial attack using Lenet and Resnet-18 architecture for MNIST and CIFAR10, respectively. Adversarial training is incorporated in both

fixed and dynamic activation quantization method for a fair comparison.

Activation Bits	Clean		PGD		FGSM	
	Fixed	Dynamic	Fixed	Dynamic	Fixed	Dynamic
1-bit	98.52 %	98.53%	98.47%	98.53%	98.48%	97.44%
2-bit	98.65%	98.80%	98.46%	98.75%	98.59%	98.75%
3-bit	99.05%	99.01%	98.37%	98.51%	98.7%	97.15%
Full-Precision	99.24%		94.04%		97.1%	

Table 1. Summary of DQA's preformance as a defense for MNIST

5.1 Results on MNIST Dataset

We run two different sets of simulation for fixed and dynamic activation quantiztion, where each consists of different bit-widths of quantization levels. First, we obtain the results using ReLu activation function reported as 'full-precision' activation in table 1. Later we report the clean data and under-attack accuracy for 1-, 2-, 3-bit fixed and dynamic activation quantization. The summary of simulation results under both PGD and FGSM attack for MNIST dataset is shown in Table 1. PGD based adversarial training is used as a base defense method that achieves about 94% accuracy under PGD attack and 97.1% accuracy under FGSM attack. It can be seen that, after incorporating activation quantization, the accuracies under both attacks are easily increased to close to 98%. For example, 1-bit fixed activation quantization could improve the accuracy from 94.04% to 98.47% under PGD attack. Meanwhile, slight improvements are observed when dynamic activation functions are used for all the cases. Both methods could improve ~4% accuracy compared to baseline (i.e. full-precision) under PGD attack. But dynamic quantization activation function results in a stronger defense for PGD than that for FGSM attack for 1-bit and 3-bit activation. However, by using 2-bit activation we could get the best defense accuracy of 98.75 % for FGSM attack.

5.2 Results on CIFAR10

We also test our defense method using Resnet-18 architecture for cifar10 dataset. As a baseline case (i.e. full-precision), we similarly employ ReLu activaiton function. Then, both fixed and dynamic activation quantization methods are simulated and tabulated in table 2.

Activation Bits	Clean		PGD		FGSM	
	Fixed	Dynamic	Fixed	Dynamic	Fixed	Dynamic
1-bit	81.43 %	83.16%	72.4%	77.62%	72.9%	76.09%
2-bit	82.33%	85.76%	70.51%	79.83%	71.8%	76.37%
3-bit	84.14%	84.17%	63.23%	75.01%	73.18%	74.51%
Full-Precision	87.59%		48.70%		58.16%	

Table 2. Summary of DQA's preformance as a defense for CIFAR10

First, similar as many other defense methods [23, 36], activation quantization method will slightly degrade clean data accuracy. For example, the clean data accuracy degrades from 87.59% to 81.43% when fixed binary activation function is used. But, it is interesting to also observe that such accuracy degradation effect

could be mitigated when we introduce dynamic activation quantization as shown in table 2. For example, it improves binary activation accuracy from 81.43% to 83.16%. Moreover, DQA could get as high as 85.76 % clean data accuracy using 2-bit dynamic quantized activation function.

Second, for fixed activation quantization function, the defense against adversarial examples increases with decreasing quantization bit-width. For example, under strong PGD attack, CIFAR 10 accuracy increases up to 72.4% for 1-bit activation function, from 48.7% for full precision activation function. The defense decreases significantly as we increase the activation quantization bit width. For example, a 3-bit activation quantization leads accuracy to drop down to 63.23% under PGD attack. However, it is still 14.53% better than full precision activation function. A similar trend was also observed for FGSM attack as well.

Third, for dynamic activation quantization function, we observe a twofold improvement. First, dynamic activation recovers the clean data accuracy to some extent. Second, as explained earlier, adding dynamic quantization thresholds to the loss function works in favor of model robustness. Our experimental results show accuracy improvement for all the low bit width activation function under PGD and FGSM attacks, compared to fixed bit quantization. Employing DQA method, we could push the accuracy up to 79.83% using 2-bit dynamic activation function across all layers. Again, for CIFAR10, we observe FGSM attack becomes stronger compared to PGD which is similar for MNIST as well. However, around 4% higher accuracy is observed with dynamic activation quantization function for both 1-bit and 2-bit cases when attacked using FGSM.
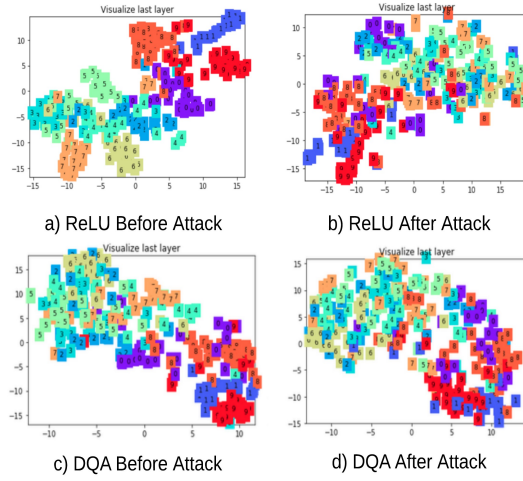

Figure 3: tSNE visualization before and after attack using both Relu and DQA activation

In order to better illustrate how DQA could help improve DNN robustness, we employ t-SNE to visualize input data of last fully connected layer for CIFAR10 dataset as shown in figure 3. t-SNE is a popular data visualization technique to get more insight to high dimensional data by reducing the dimensionality [51]. Typical shape and distance of a t-SNE plot does not carry much significance. Rather, for classification tasks, the data that belongs to the same class should create a cluster of points close to each other. A network performing well on the classification task should create a cluster without much scattering. We plot the same data using both ReLU and DQA function before attacking in figure 3(a)&(c) and after attacking in figure 3(b)&(d). For both data in figure 3(a)&(c), it is clear that the data points belong to the same group stay in a

close cluster before attack. However, under adversarial attack, these data become much more scattered in traditional ReLU activation function compared to our proposed dynamic quantized activation function.

C & W Attack   We also conduct experiments of our proposed defense methods under C & W attack [7]. It is an effective optimization-based attack model. The amount of perturbation to an input image is generated through L_2 norm by solving an optimization problem based on an objective function. The purpose of such objective function is to find adversary that would miss-classify the image. A higher value of L_2 norm indicates a more robust network or potential failure of the attack. We measure the averaged L_2 norm required to successfully attack each cases for Cifar 10 dataset under C & W attack in this subsection.

Table 3. C & W attack summary:

Activation	Fixed quantization Without PGD training (L2 norm)	DQA With PGD training Attack Succ. Rate
32-bit	0.12	0
3-bit	0.17	0
2-bit	0.26	0
1-bit	0.55	0

Table 3 shows the L_2 norm required for C & W attack to get 100 percent success in different cases in the first column. Again, for C & W attack, we observe the same pattern as we observed before for both PGD and FGSM attacks. Average L_2 norm required for 100 percent successfull attack keeps decreasing with increasing activation quantization bit width. We also run the experiments of dynamic activation quantization under C & W attack using adversarial training. But our proposed DQA successfully defends C & W attack as shown in the second column. The C & W attack fails to miss-classify the input with reasonable L_2 norm within a certain amount of time when supplemented by PGD training. As explained in [23], defenses trained with l_∞ bounded universal attack tends to perform well against l_2 bounded attacks, resulting in high l_2 norm which are eventually detectable even by human eyes.

5.3   Comparison to other state-of-the-art defense methodologies

In this subsection, we compare our proposed defense performance with several recent popular defense methodologies. One of the most successful defense is madry's defense [23] that uses PGD adversarial training as the main form of defense. Such adversarial training method has become the standard for most recent defense techniques to improve DNN robustness. For example, one such defense is thermometer encoding that performs encoding of input data as a defense [44]. However, their proposed method has gradient scattering issue [21]. In another work called defense GAN, it uses GAN discriminator and generator to counter the adversarial examples [35]. Cowboy is another GAN based defense that uses the scores from a GAN's discriminator to separate adversary from clean examples [37]. Note that, all of the defenses here include PGD based adversarial training except the Cowboy defense. Here, we report accuracy under PGD and FGSM attack mdoels for different defenses in table 4, where the values of ϵ are 0.3 and 0.031, respectively, for MNIST and CIFAR10 datasets. Meanwhile, all of the defense accuracy corresponding to PGD attack employ attack steps of 40 and 7 for MNIST and CIFAR10, respectively.

Table 4 summarizes the comparison results of our proposed DQA defense methods over other recent popular defense methods. Since our methodology involves quantization of activation function, we expect a

Table 4. Our defense performance compared to the state of art defenses

Defense	Arch.	Clean %	(PGD) %	FGSM %
MNIST				
Madry [23]	Vanilla	98.80	93.2	95.6
T. E. [44]	Vanilla	99.03	94.02	95.82
Defense GAN [35]	GAN	99.20	___	98.43
Cowboy[37]	GAN	97	78	81
DQA (Ours)	Lenet (2-bit)	98.80	98.75	98.75
CIFAR10				
Madry[23]	Wide Resnet	87.3	50.00	56.1
T.E.[44]	Wide Resnet	89.88	79.16	80.96
Cowboy[37]	GAN	78	53	53
DQA (Ours)	Resnet 18 (2bit)	85.76	79.83	76.37

lower clean data accuracy especially when supplemented by adversarial training. However, our clean data accuracy of 99.01 % and 85.76 % for MNIST and CIFAR10 respectively are still within 0.5-2 % of recent defenses. Our major success lies in achieving greatly improved accuracy in defending PGD attack. For both MNIST and CIFAR10, our reported accuracies are higher than that of all previous defenses reported here. As for FGSM attack, our method gives higher accuracy for MNIST than all of the reported attacks. In case of CIFAR, we could achieve better accuracy than PGD defense [23] and cowboy [37] as well. Another defense L2NNN [36] that uses almost similar theory as our defense of suppressing the input noise from increasing can actually get 91.7 % and 22 % accuracy for MNIST and CIFAR10 dataset, respectively, against l2 bounded adversarial attack without adversarial training. However, their clean data accuracy suffers even badly than ours reaching only 72 % for CIFAR10 accuracy. The attempt to suppress the noise early in the network would result in hampering the clean data accuracy. Thus DQA's achievement of more than 85 % accuracy for CIFAR with improved defense becomes even more significant.

6 Black-Box Attack

In this section, we also test our defense model against a variety of black box attacks. In black box setup, we assume that the attacker has no knowledge of the target DNN. Even in a black box setup the attacker does not always need to train a substitute model, rather they can directly estimate the gradient of the target DNN model based on input and output scores [16]. In another form of black box attack, the attacker can always train a substitute model to attack the target model which we call the source [24].

ZOO Black Box Attack   We also conduct zeroth order optimization(ZOO) attack on our defence [16]. ZOO is a form of black box attack that does not require training a substitute model, rather it can approximate

the gradient of the target DNN just based on input image and output scores. We use Zeroth Order Stochastic Coordinate Descent with Coordinate-wise ADAM. Similar as the original paper, we test our defense on random 200 samples for untargetted attack to observe the effectiveness of the attack on proposed defense. We summarize the attack's success rate in table 5 for CIFAR 10 dataset.

Table 5. CIFAR 10 accuracy for ZOO Attack

Bit -Width	DNN	Clean Data acc.	Attack Success rate (%)
32-bit	Res-18	87.59	81.75
3-bit	Res-18	84.17	0
2-bit	Res-18	85.76	0
1-bit	Res-18	83.16	0

It can be seen that our proposed dynamic quantization successfully defends against ZOO attack. Similar as the C & W attack, ZOO- attack fails to approximate the gradient due to the intermediate quantization function. As a result it fails to attack the defence with any kind of success.

Black-Box Attack Using a Substitute Model: In this subsection, we conduct black box attack, we train a substitute model to perform the exact same classification task as the target. The approach is similar to the popular approach for black box attack [24]. In our experiments, we separately train VGG16 and Alexnet models with full precision activation function to use it as source models.

Table 6. Black Box attack on Cifar 10 dataset using substitute model

Bit-width	Our Defense Accuracy No attack (%)	Substitute Accuracy No attack (%)	Substitute Accuracy Under Attack (%)	Defense Accuracy Under Attack(%)
VGG16 as Substitute Model				
3-bit	84.93	84.32	56.88	60.85
2-bit	85.06	85.06	56.62	60.46
1-bit	83.16	83.09	44.94	49.18
Alexnet as Substitute Model				
3-bit	84.93	84.05	46.62	82.59
2-bit	85.06	85.03	52.42	82.11
1-bit	83.16	83.05	36.06	79.74

The summary of the results obtained using a substitute model is reported in table 6. We report our model's defense accuracy, substitute model's accuracy, substitute model's accuracy under attack and our defense model's accuracy under attack sequentially in the table. We use full-precision activation at the source in order to precisely approximate the decision boundaries of the target model. It is observed that our defense's accuracy degrades to some extent when attacked using a VGG16 substitute model. However, it is much better than the substitue DNN model under the same attack. Using a 32bit VGG-16 model at the source could generate attack that would give as low as 49.18 % accuracy for binary activation. An obvious reason would

11

be using a VGG16 model would generate very strong adversarial examples due to the model's increased capacity and full- precision activation. However, for Alexnet substitute model we could get much higher accuracy than the white box counter part. Our defense could achieve 82.59, 82.11 and 79.74 % accuracy for 3,2,1 bits respectively. The reported accuracies are much higher than the white box attack and also higher than the accuracy achieved attacking the substitute model.Our proposed DQA defense preforms reasonably well against both substitute attack model and ZOO model to further establish our hypothesis. As suppressing the noise using lower bit-width activation and adding more parameters in the adversarial training process certainly makes the DNN more robust against both white box and black box attack.

7   Summary

In this work, we first propose to employ quantized activation function in DNN to defend adversarial examples. Further, we utilize the benefit of using adversarial training and quantized activation function even more by introducing dynamic quantization. Dynamic quantization pushes the state-of-the-art defense accuracy of both MNIST and CIFAR 10 further under strong attacks. Our hypothetical analysis is proven empirically through a variety of experiments showing both fixed and dynamic quantization can provide strong resistance to adversarial attack. Thus network compression and model robustness can be achieved at the same time which may lead to further compact robust neural networks research.

References

[1] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[2] Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, 29(6):82–97, 2012.

[3] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[4] Chenyi Chen, Ari Seff, Alain Kornhauser, and Jianxiong Xiao. Deepdriving: Learning affordance for direct perception in autonomous driving. In *Computer Vision (ICCV), 2015 IEEE International Conference on*, pages 2722–2730. IEEE, 2015.

[5] 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.

[6] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[7] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *Security and Privacy (SP), 2017 IEEE Symposium on*, pages 39–57. IEEE, 2017.

[8] Xiaojun Xu, Xinyun Chen, Chang Liu, Anna Rohrbach, Trevor Darell, and Dawn Song. Can you fool ai with adversarial examples on a visual turing test? *arXiv preprint arXiv:1709.08693*, 2017.

[9] Hongge Chen, Huan Zhang, Pin-Yu Chen, Jinfeng Yi, and Cho-Jui Hsieh. Show-and-fool: Crafting adversarial examples for neural image captioning. *arXiv preprint arXiv:1712.02051*, 2017.

[10] Jan Hendrik Metzen, Mummadi Chaithanya Kumar, Thomas Brox, and Volker Fischer. Universal adversarial perturbations against semantic image segmentation. *stat*, 1050:19, 2017.

[11] Minhao Cheng, Jinfeng Yi, Huan Zhang, Pin-Yu Chen, and Cho-Jui Hsieh. Seq2sick: Evaluating the robustness of sequence-to-sequence models with adversarial examples. *arXiv preprint arXiv:1803.01128*, 2018.

[12] Nicholas Carlini and David Wagner. Audio adversarial examples: Targeted attacks on speech-to-text. *arXiv preprint arXiv:1801.01944*, 2018.

[13] Mengying Sun, Fengyi Tang, Jinfeng Yi, Fei Wang, and Jiayu Zhou. Identify susceptible locations in medical records via adversarial attacks on deep predictive models. *arXiv preprint arXiv:1802.04822*, 2018.

[14] Yanpei Liu, Xinyun Chen, Chang Liu, and Dawn Song. Delving into transferable adversarial examples and black-box attacks. *arXiv preprint arXiv:1611.02770*, 2016.

[15] Nicolas Papernot, Patrick McDaniel, and Ian Goodfellow. Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. *arXiv preprint arXiv:1605.07277*, 2016.

[16] Pin-Yu Chen, Huan Zhang, Yash Sharma, Jinfeng Yi, and Cho-Jui Hsieh. Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, pages 15–26. ACM, 2017.

[17] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533*, 2016.

[18] Jernej Kos and Dawn Song. Delving into adversarial attacks on deep policies. *arXiv preprint arXiv:1705.06452*, 2017.

[19] Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z Berkay Celik, and Ananthram Swami. The limitations of deep learning in adversarial settings. In *Security and Privacy (EuroS&P), 2016 IEEE European Symposium on*, pages 372–387. IEEE, 2016.

[20] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2574–2582, 2016.

[21] Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. *arXiv preprint arXiv:1802.00420*, 2018.

[22] Pin-Yu Chen, Yash Sharma, Huan Zhang, Jinfeng Yi, and Cho-Jui Hsieh. Ead: Elastic-net attacks to deep neural networks via adversarial examples. *arXiv preprint arXiv:1709.04114*, 2017.

[23] 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.

[24] Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*, pages 506–519. ACM, 2017.

[25] Shixiang Gu and Luca Rigazio. Towards deep neural network architectures robust to adversarial examples. *arXiv preprint arXiv:1412.5068*, 2014.

[26] Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. Distillation as a defense to adversarial perturbations against deep neural networks. In *Security and Privacy (SP), 2016 IEEE Symposium on*, pages 582–597. IEEE, 2016.

[27] Nicolas Papernot, Patrick McDaniel, Arunesh Sinha, and Michael Wellman. Towards the science of security and privacy in machine learning. *arXiv preprint arXiv:1611.03814*, 2016.

[28] Weilin Xu, David Evans, and Yanjun Qi. Feature squeezing: Detecting adversarial examples in deep neural networks. *arXiv preprint arXiv:1704.01155*, 2017.

[29] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *AAAI*, volume 4, page 12, 2017.

[30] Gao Huang, Zhuang Liu, Kilian Q Weinberger, and Laurens van der Maaten. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, volume 1, page 3, 2017.

[31] Song Han, Huizi Mao, and William J Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv:1510.00149*, 2015.

[32] Matthieu Courbariaux, Yoshua Bengio, and Jean-Pierre David. Binaryconnect: Training deep neural networks with binary weights during propagations. In *Advances in neural information processing systems*, pages 3123–3131, 2015.

[33] Mohammad Rastegari, Vicente Ordonez, Joseph Redmon, and Ali Farhadi. Xnor-net: Imagenet classification using binary convolutional neural networks. In *European Conference on Computer Vision*, pages 525–542. Springer, 2016.

[34] Shuchang Zhou, Yuxin Wu, Zekun Ni, Xinyu Zhou, He Wen, and Yuheng Zou. Dorefa-net: Training low bitwidth convolutional neural networks with low bitwidth gradients. *arXiv preprint arXiv:1606.06160*, 2016.

[35] Pouya Samangouei, Maya Kabkab, and Rama Chellappa. Defense-gan: Protecting classifiers against adversarial attacks using generative models. *arXiv preprint arXiv:1805.06605*, 2018.

[36] Haifeng Qian and Mark N Wegman. L2-nonexpansive neural networks. *arXiv preprint arXiv:1802.07896*, 2018.

[37] Gokula Krishnan Santhanam and Paulina Grnarova. Defending against adversarial attacks by leveraging an entire gan. *arXiv preprint arXiv:1805.10652*, 2018.

[38] Cihang Xie, Jianyu Wang, Zhishuai Zhang, Zhou Ren, and Alan Yuille. Mitigating adversarial effects through randomization. *arXiv preprint arXiv:1711.01991*, 2017.

[39] Chuan Guo, Mayank Rana, Moustapha Cissé, and Laurens van der Maaten. Countering adversarial images using input transformations. *arXiv preprint arXiv:1711.00117*, 2017.

[40] Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Dan Boneh, and Patrick McDaniel. Ensemble adversarial training: Attacks and defenses. *arXiv preprint arXiv:1705.07204*, 2017.

[41] Aditi Raghunathan, Jacob Steinhardt, and Percy Liang. Certified defenses against adversarial examples. *arXiv preprint arXiv:1801.09344*, 2018.

[42] Guneet S Dhillon, Kamyar Azizzadenesheli, Zachary C Lipton, Jeremy Bernstein, Jean Kossaifi, Aran Khanna, and Anima Anandkumar. Stochastic activation pruning for robust adversarial defense. *arXiv preprint arXiv:1803.01442*, 2018.

[43] Aaditya Prakash, Nick Moran, Solomon Garber, Antonella DiLillo, and James Storer. Deflecting adversarial attacks with pixel deflection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8571–8580, 2018.

[44] Colin Raffel Ian Goodfellow Jacob Buckman, Aurko Roy. Thermometer encoding: One hot way to resist adversarial examples. *International Conference on Learning Representations*, 2018. accepted as poster.

[45] Henry W Lin, Max Tegmark, and David Rolnick. Why does deep and cheap learning work so well? *Journal of Statistical Physics*, 168(6):1223–1247, 2017.

[46] Abraham Wald. Statistical decision functions which minimize the maximum risk. *Annals of Mathematics*, pages 265–280, 1945.

[47] Yann LeCun et al. Lenet-5, convolutional neural networks. *URL: http://yann. lecun. com/exdb/lenet*, page 20, 2015.

[48] Yann LeCun, Yoshua Bengio, et al. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10):1995, 1995.

[49] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[50] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Cifar-10 (canadian institute for advanced research). *URL http://www. cs. toronto. edu/kriz/cifar. html*, 2010.

[51] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.