# Improved Gradient-Based Adversarial Attacks for Quantized Networks

**Kartik Gupta [1,3], Thalaiyasingam Ajanthan [†1,2]**
[1]Australian National University
[2]Amazon Science
[3]DATA61-CSIRO
kartik.gupta@anu.edu.au, thalaiyasingam.ajanthan@anu.edu.au

## Abstract

Neural network quantization has become increasingly popular due to efficient memory consumption and faster computation resulting from bitwise operations on the quantized networks. Even though they exhibit excellent generalization capabilities, their robustness properties are not well-understood. In this work, we systematically study the robustness of quantized networks against gradient based adversarial attacks and demonstrate that these quantized models suffer from gradient vanishing issues and show a fake sense of robustness. By attributing gradient vanishing to poor forward-backward signal propagation in the trained network, we introduce a simple temperature scaling approach to mitigate this issue while preserving the decision boundary. Despite being a simple modification to existing gradient based adversarial attacks, experiments on multiple image classification datasets with multiple network architectures demonstrate that our temperature scaled attacks obtain near-perfect success rate on quantized networks while outperforming original attacks on adversarially trained models as well as floating-point networks.

## Introduction

Neural Network (NN) quantization has become increasingly popular due to reduced memory and time complexity enabling real-time applications and inference on resource-limited devices. Such quantized networks often exhibit excellent generalization capabilities despite having low capacity due to reduced precision for parameters and activations. However, their robustness properties are not well-understood. In particular, while parameter quantized networks are claimed to have better robustness against gradient based adversarial attacks (Galloway, Taylor, and Moussa 2018), activation only quantized methods are shown to be vulnerable (Lin, Gan, and Han 2019).

In this work[1], we consider the extreme case of Binary Neural Networks (BNNs) and systematically study the robustness properties of parameter quantized models, as well as both parameter and activation quantized models against gradient based adversarial attacks. Our analysis reveals that these quantized models suffer from gradient masking issues (Athalye, Carlini, and Wagner 2018) and in turn show fake robustness. We attribute this vanishing gradients issue to poor forward-backward signal propagation caused by trained binary weights, and our idea is to improve signal propagation of the network without changing the prediction.

There is a body of work on improving signal propagation in a neural network (*e.g.*, (Glorot and Bengio 2010; Pennington, Schoenholz, and Ganguli 2017; Lu, Gould, and Ajanthan 2020)), however, we are facing a unique challenge of *improving signal propagation while preserving the decision boundary*, since our ultimate objective is to generate adversarial attacks. To this end, we first discuss the conditions to ensure informative gradients and then resort to a temperature scaling approach (Guo et al. 2017) (which scales the logits before applying softmax cross-entropy) to show that, even with a single positive scalar the vanishing gradients issue in BNNs can be alleviated achieving *near perfect success rate*.

Specifically, we introduce two techniques to choose the temperature scale: 1) based on the singular values of the input-output Jacobian, 2) by maximizing the norm of the Hessian of the loss with respect to the input. The justification for the first case is that if the singular values of input-output Jacobian are concentrated around 1 (defined as dynamical isometry (Pennington, Schoenholz, and Ganguli 2017)) then the network is said to have good signal propagation. On the other hand, the intuition for maximizing the Hessian norm is that if the Hessian norm is large, then the gradient of the loss with respect to the input is sensitive to an infinitesimal change in the input. This is a sufficient condition for the network to have good signal propagation as well as informative gradients under the assumption that the network does not have any randomized or non-differentiable components.

In summary, this paper makes the following contributions:

- We first show via various empirical checks that BNNs possess fake robustness against gradient based adversarial attacks such as FGSM (Goodfellow, Shlens, and Szegedy 2014) and PGD (Madry et al. 2017).
- By accounting poor signal propagation for the failure of gradient based adversarial attacks, we present temperature scaling based solution to improve the existing attacks without changing the prediction of the network.
- In order to estimate appropriate scalar for temperature scaling in gradient based adversarial attacks, we present

[1] Open-source implementation available at https://github.com/kartikgupta-at-anu/attack-bnn. [†]Work done prior to joining Amazon

two variants namely Network Jacobian Scaling (NJS) and Hessian Norm Scaling (HNS) motivated from point of view of improving the signal propagation.

- With experimental evaluations using several network architectures on CIFAR-10/100 datasets, we show that our proposed techniques to modify existing gradient based adversarial attacks achieve near perfect success rate on BNNs with weight quantized (BNN-WQ) and weight and activation quantized (BNN-WAQ). Furthermore, our variants improves attack success even on adversarially trained models as well as floating point networks, showing the significance of signal propagation for adversarial attacks.

# Preliminaries

We first provide some background on the neural network quantization and adversarial attacks.

## Neural Network Quantization

Neural Network (NN) quantization is defined as training networks with parameters constrained to a minimal, discrete set of quantization levels. This primarily relies on the hypothesis that since NNs are usually overparametrized, it is possible to obtain a quantized network with performance comparable to the floating point network. Given a dataset $\mathcal{D} = \{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^n$, NN quantization can be written as:

$$\min_{\mathbf{w} \in \mathcal{Q}^m} L(\mathbf{w}; \mathcal{D}) := \frac{1}{n} \sum_{i=1}^{n} \ell(\mathbf{w}; (\mathbf{x}_i, \mathbf{y}_i)) \,. \tag{1}$$

Here, $\ell(\cdot)$ denotes the input-output mapping composed with a standard loss function (*e.g.*, cross-entropy loss), $\mathbf{w}$ is the $m$ dimensional parameter vector, and $\mathcal{Q}$ is a predefined discrete set representing quantization levels (*e.g.*, $\mathcal{Q} = \{-1, 1\}$ in the binary case).

Most of the NN quantization approaches (Ajanthan et al. 2019, 2021; Bai, Wang, and Liberty 2019; Hubara et al. 2017) convert the above problem into an unconstrained problem by introducing auxiliary variables and optimize via (stochastic) gradient descent. To this end, the algorithms differ in the choice of quantization set (*e.g.*, keep it discrete (Courbariaux, Bengio, and David 2015), relax it to the convex hull (Bai, Wang, and Liberty 2019) or convert the problem into a lifted probability space (Ajanthan et al. 2019)), the projection used, and how differentiation through projection is performed. In the case when the constraint set is relaxed, a gradually increasing annealing hyperparameter is used to enforce a quantized solution (Ajanthan et al. 2019, 2021; Bai, Wang, and Liberty 2019). We refer the interested reader to respective papers for more detail. In this paper, we use BNN-WQ obtained using MD-$\tanh$-S (Ajanthan et al. 2021) and BNN-WAQ obtained using obtained using Straight Through Estimation (Hubara et al. 2017). Briefly MD-$\tanh$-S represents network binarization method based on mirror descent optimization where the mirror map is derived using $\tanh$ projection function.

## Adversarial Attacks

Adversarial examples consist of imperceptible perturbations to the data that alter the model's prediction with high confidence. Existing attacks can be categorized into white-box and black-box attacks where the difference lies in the knowledge of the adversaries. White-box attacks allow the adversaries access to the target model's architecture and parameters, whereas black-box attacks can only query the model. Since white-box gradient based attacks are popular, we summarize them below.

| Method | ResNet-18 | | | VGG-16 | | |
|---|---|---|---|---|---|---|
| | Clean | Adv.(1) | Adv.(20) | Clean | Adv.(1) | Adv.(20) |
| REF | 94.46 | 0.00 | 0.00 | 93.31 | 0.04 | 0.00 |
| BNN-WQ | 93.18 | 26.98 | 17.91 | 91.53 | 47.32 | 38.49 |
| BNN-WAQ | 87.67 | 8.57 | 1.94 | 89.69 | 78.01 | 59.26 |

Table 1: *Clean and adversarial accuracy (PGD attack with $L_\infty$ bound) on the test set of CIFAR-10 using ResNet-18 and VGG-16. In brackets, we mention number of random restarts used to perform the attack. Note,* BNN*s yield higher adversarial accuracy than floating point networks consistently.*

First-order gradient based attacks can be compactly written as Projected Gradient Descent (PGD) on the negative of the loss function (Madry et al. 2017). Formally, let $\mathbf{x}^0 \in \mathbb{R}^{\mathrm{N}}$ be the input image, then at iteration $t$, the PGD update can be written as:

$$\mathbf{x}^{t+1} = P\left(\mathbf{x}^t + \eta\, \mathbf{g}_{\mathbf{x}}^t\right) \,, \tag{2}$$

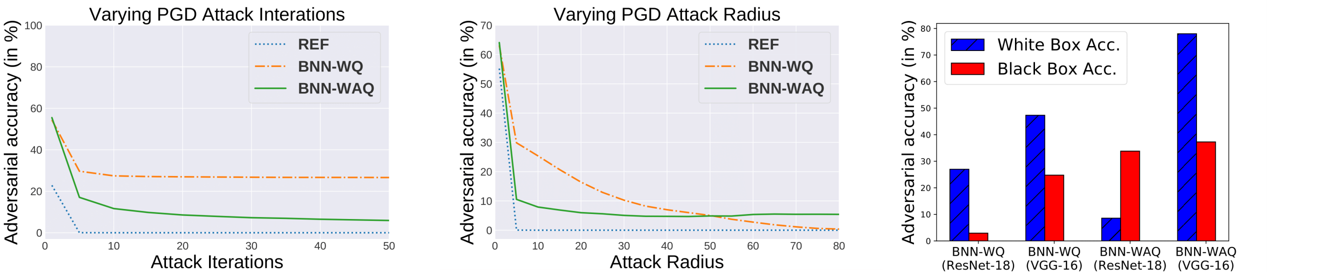
where $P : \mathbb{R}^{\mathrm{N}} \to \mathcal{X}$ is a projection, $\mathcal{X} \subset \mathbb{R}^{\mathrm{N}}$ is the constraint set that bounds the perturbations, $\eta > 0$ is the step size, and $\mathbf{g}_{\mathbf{x}}^t$ is a form of gradient of the loss with respect to the input $\mathbf{x}$ evaluated at $\mathbf{x}^t$. With this general form, the popular gradient based adversarial attacks can be specified:

- **Fast Gradient Sign Method (FGSM)**: This is a one step attack introduced in (Goodfellow, Shlens, and Szegedy 2014). Here, $P$ is the identity mapping, $\eta$ is the maximum allowed perturbation magnitude, and $\mathbf{g}_{\mathbf{x}}^t = \mathrm{sign}\left(\nabla_{\mathbf{x}} \ell(\mathbf{w}^*; (\mathbf{x}^t, \mathbf{y}))\right)$, where $\ell$ denotes the loss function, $\mathbf{w}^*$ is the trained weights and $\mathbf{y}$ is the ground truth label corresponding to the image $\mathbf{x}^0$.
- **PGD with $L_\infty$ bound**: Arguably the most popular adversarial attack introduced in (Madry et al. 2017) and sometimes referred to as Iterative Fast Gradient Sign Method (IFGSM). Here, $P$ is the $L_\infty$ norm based projection, $\eta$ is a chosen step size, and $\mathbf{g}_{\mathbf{x}}^t = \mathrm{sign}\left(\nabla_{\mathbf{x}} \ell(\mathbf{w}^*; (\mathbf{x}^t, \mathbf{y}))\right)$, the sign of gradient same as FGSM.
- **PGD with $L_2$ bound**: This is also introduced in (Madry et al. 2017) which performs the standard PGD in the Euclidean space. Here, $P$ is the $L_2$ norm based projection, $\eta$ is a chosen step size, and $\mathbf{g}_{\mathbf{x}}^t = \nabla_{\mathbf{x}} \ell(\mathbf{w}^*; (\mathbf{x}^t, \mathbf{y}))$ is simply the gradient of the loss with respect to the input.

These attacks have been further strengthened by a random initial step (Tramèr et al. 2017). In this paper, we perform this single random initialization for all experiments with FGSM/PGD attack unless otherwise mentioned.

# Robustness Evaluation of BNNs

We start by evaluating the adversarial accuracy (i.e. accuracy on the perturbed data) of BNNs using the PGD attack with perturbation bound of 8 pixels (assuming each pixel in



(a) Attack iterations does not improve attack. (b) Attack radius does not improve attack. (c) Black-box attacks perform better.

Figure 1: *Gradient masking checks in ResNet-18 on CIFAR-10 for* PGD *attack with $L_\infty$ bound. While (a), (c) show signs of gradient masking, (b) does not. We attribute this discrepancy to the random initial step before* PGD.

the image is in $[0, 255]$) with respect to $L_\infty$ norm, step size $\eta = 2$ and the total number of iterations $T = 20$. The attack details are the same in all evaluated settings unless stated otherwise. We perform experiments on CIFAR-10 dataset using ResNet-18 and VGG-16 architectures and report the clean accuracy and PGD adversarial accuracy with 1 and 20 random restarts in Table 1. It can be clearly and consistently observed that binary networks have high adversarial accuracy compared to the floating point counterparts. Even with 20 random restarts, BNNs clearly outperform floating point networks in terms of adversarial accuracy. Since this result is surprising, we investigate this phenomenon further to understand whether BNNs are actually robust to adversarial perturbations or they show a fake sense of security due to obfuscated gradients (Athalye, Carlini, and Wagner 2018).

**Identifying Obfuscated Gradients.** Recently, it has been shown that several defense mechanisms intentionally or unintentionally break gradient descent and cause obfuscated gradients and thus exhibit a false sense of security (Athalye, Carlini, and Wagner 2018). Several gradient based adversarial attacks tend to fail to produce adversarial perturbations in scenarios where the gradients are uninformative, referred to as gradient masking. Gradient masking can occur due to shattered gradients, stochastic gradients or exploding and vanishing gradients. We try to identify gradient masking in binary networks based on the empirical checks provided in (Athalye, Carlini, and Wagner 2018). If any of these checks fail, it indicates gradient masking issue in BNNs.

To illustrate this, we analyse the effects of varying different hyperparameters of PGD attack on BNNs trained on CIFAR-10 using ResNet-18 architecture. Even though varying PGD perturbation bound does not show any signs of gradient masking, varying attack iterations and black-box vs white-box results (on ResNet-18 and VGG-16) clearly indicate gradient masking issues as depicted in Fig. 1. The black-box attack outperforming white-box attack for BNNs certainly indicates gradient masking issues since the black-box attack do not use the gradient information from model being attacked. Here, our black-box model to a BNN is the analogous floating point network trained on the same dataset and the attack is the same PGD with $L_\infty$ bound.

*These checks demonstrate that* BNNs *are prone to gradient masking and exhibit fake robustness.* Note, shattered gradients occur due to non-differentiable components in the defense mechanism and stochastic gradients are caused by randomized gradients. Since BNNs are trainable from scratch and does not have randomized gradients, we narrow down gradient masking issue to vanishing or exploding gradients. Since, vanishing or exploding gradients occur due to poor signal propagation, by introducing a single scalar, we discuss two approaches to mitigate this issue, which lead to almost $100\%$ success rate for gradient based attacks on BNNs.

## Signal Propagation of Neural Networks

We first describe how poor signal propagation in neural networks can cause vanishing or exploding gradients. Then we discuss the idea of introducing a single scalar to improve the existing gradient based attacks without affecting the prediction (*i.e.*, decision boundary) of the trained models.

We consider a neural network $f_\mathbf{w}$ for an input $\mathbf{x}^0$, having post-activations $\mathbf{a}^l$, for $l \in \{1 \ldots K\}$ up to $K$ layers and logits $\mathbf{a}^K = f_\mathbf{w}(\mathbf{x}^0)$. Now, since softmax cross-entropy is usually used as the loss function, we can write:

$$\ell(\mathbf{a}^K, \mathbf{y}) = -\mathbf{y}^T \log(\mathbf{p}) \;, \qquad \mathbf{p} = \mathrm{softmax}(\mathbf{a}^K) \;, \quad (3)$$

where $\mathbf{y} \in \mathbb{R}^\mathrm{d}$ is the one-hot encoded target label and $\log$ is applied elementwise.

For various gradient based adversarial attacks discussed earlier, gradient of the loss $\ell$ is used with respect to the input $\mathbf{x}^0$, which can also be formulated using chain rule as,

$$\frac{\partial \ell(\mathbf{a}^K, \mathbf{y})}{\partial \mathbf{x}^0} = \frac{\partial \ell(\mathbf{a}^K, \mathbf{y})}{\partial \mathbf{a}^K} \frac{\partial \mathbf{a}^K}{\partial \mathbf{x}^0} = \psi(\mathbf{a}^K, \mathbf{y})\, \mathbf{J} \;, \quad (4)$$

where $\psi$ denotes the error signal and $\mathbf{J} \in \mathbb{R}^{d \times N}$ is the input-output Jacobian. Here we use the convention that $\partial \mathbf{v} / \partial \mathbf{u}$ is of the form $\mathbf{v}$-size × $\mathbf{u}$-size.

Notice there are two components that influence the gradients, 1) the Jacobian $\mathbf{J}$ and 2) the error signal $\psi$. Gradient based attacks would fail if either the Jacobian is poorly conditioned or the error signal has saturating gradients, both of these will lead to vanishing gradients in $\partial \ell / \partial \mathbf{x}^0$.

The effects of Jacobian on the signal propagation is studied in dynamical isometry and mean-field theory literature (Pennington, Schoenholz, and Ganguli 2017; Saxe, McClelland, and Ganguli 2013) and it is known that a network is said to satisfy dynamical isometry if the singular values of $\mathbf{J}$ are concentrated near 1. Under this condition, error signals $\psi$ backpropagate isometrically through the network, approximately preserving its norm and all angles between error vectors. Thus, as dynamical isometry improves the trainability of the floating point networks, a similar technique can be useful for gradient based attacks as well.

In fact, almost all initialization techniques (*e.g.*, (Glorot and Bengio 2010)) approximately ensures that the Jacobian $\mathbf{J}$ is well-conditioned for better trainability and it is hypothesized that approximate isometry is preserved even at the end of the training. But, for BNNs, the weights are constrained to be $\{-1, 1\}$ and hence the weight distribution at end of training is completely different from the random initialization. Furthermore, it is not clear that fully-quantized networks can achieve well-conditioned Jacobian, which guided some research activity in utilizing layerwise scalars (either predefined or learned) to improve BNN training (McDonnell 2018; Rastegari et al. 2016). We would like to point out that the focus of this paper is to improve gradient based attacks on already trained BNNs. To this end learning a new scalar to improve signal propagation at each layer is not useful as it can alter the decision boundary of the network and thus cannot be used in practice on already trained model.

**Temperature Scaling for better Signal Propagation.** In this paper, we propose to use a single scalar per network to improve the signal propagation of the network using temperature scaling. In fact, one could replace softmax with a monotonic function such that the prediction is not altered, however, we will show in our experiments that a single scalar with softmax has enough flexibility to improve signal propagation and yields almost 100% success rate with PGD attacks. Essentially, we can use a scalar, $\beta > 0$ without changing the decision boundary of the network by preserving the relative order of the logits. Precisely, we consider the following:

$$\mathbf{p}(\beta) = \mathrm{softmax}(\bar{\mathbf{a}}^K)\ , \qquad \bar{\mathbf{a}}^K = \beta\, \mathbf{a}^K\ . \tag{5}$$

Here, we write the softmax output probabilities $\mathbf{p}$ as a function of $\beta$ to emphasize that they are softmax output of temperature scaled logits. Now since in this context, the only variable is the temperature scale $\beta$, we denote the loss and the error signal as functions of only $\beta$. With this simplified notation the gradient of the temperature scaled loss with respect to the inputs can be written as:

$$\frac{\partial \ell(\beta)}{\partial \mathbf{x}^0} = \frac{\partial \ell(\beta)}{\partial \bar{\mathbf{a}}^K}\frac{\partial \bar{\mathbf{a}}^K}{\partial \mathbf{a}^K}\frac{\partial \mathbf{a}^K}{\partial \mathbf{x}^0} = \psi(\beta)\beta\,\mathbf{J}\ . \tag{6}$$

Note that $\beta$ affects the input-output Jacobian linearly while it nonlinearly affects the error signal $\psi$. To this end, we hope to obtain a $\beta$ that ensures the error signal is useful (*i.e.*, not all zero) as well as the Jacobian is well-conditioned to allow the error signal to propagate to the input.

We acknowledge that while one can find a $\beta > 0$ to obtain softmax output ranging from a uniform distribution ($\beta = 0$) to one-hot vectors ($\beta \to \infty$), $\beta$ only scales the Jacobian. Therefore, if the Jacobian $\mathbf{J}$ has zero singular values, our approach has no effect in those dimensions. However, since most of the modern networks consist of ReLU nonlinearities (generally positive homogeneous functions), the effect of a single scalar would be equivalent (ignoring the biases) to having layerwise scalars such as in (McDonnell 2018). Thus, we believe a single scalar is sufficient for our purpose.

## Improved Gradients for Adversarial Attacks

Now we discuss strategies to choose a scalar $\beta$ such that the gradients with respect to input are informative. Let us first analyze the effect of $\beta$ on the error signal. To this end,

$$\psi(\beta) = \frac{\partial \ell(\beta)}{\partial \mathbf{p}(\beta)}\frac{\partial \mathbf{p}(\beta)}{\partial \bar{\mathbf{a}}^K} = -(\mathbf{y} - \mathbf{p}(\beta))^T\ . \tag{7}$$

where $\mathbf{y}$ is the one-hot encoded target label, and $\mathbf{p}(\beta)$ is the softmax output of scaled logits.

For adversarial attacks, we only consider the correctly classified images (*i.e.*, $\mathrm{argmax}_j\, y_j = \mathrm{argmax}_j\, p_j(\beta)$) as there is no need to generate adversarial examples corresponding to misclassified samples. From the above formula, it is clear that when $\mathbf{p}(\beta)$ is one-hot encoding then the error signal is $\mathbf{0}$. This is one of the reason for vanishing gradient issue in BNNs. Even if this does not happen for a given image, one can increase $\beta \to \infty$ to make this error signal $\mathbf{0}$. Similarly, when $\mathbf{p}(\beta)$ is the uniform distribution, the norm of the error signal is at the maximum. This can be obtained by setting $\beta = 0$. However, this would also make $\partial\ell(\beta)/\partial\mathbf{x}^0 = \mathbf{0}$ as the singular values of the input-output Jacobian would all be 0.

This analysis indicates that the optimal $\beta$ cannot be obtained by simply maximizing the norm of the error signal and we need to balance both the Jacobian as well as the error signal. To summarize, the scalar $\beta$ should be chosen such that the following properties are satisfied:

1. $\|\psi(\beta)\|_2 > \rho$ for some $\rho > 0$.
2. The Jacobian $\beta\,\mathbf{J}$ is well-conditioned, *i.e.*, the singular values of $\beta\,\mathbf{J}$ is concentrated around 1.

**Network Jacobian Scaling (NJS).** We now discuss a straightforward, two-step approach to attain the aforementioned properties. Firstly, to ensure $\beta\mathbf{J}$ is well-conditioned, we simply choose $\beta$ to be the inverse of the mean of singular values of $\mathbf{J}$. This guarantees that the mean of singular values of $\beta\mathbf{J}$ is 1.

After this scaling, it is possible that the resulting error signal is very small. To ensure that $\|\psi(\beta)\|_2 > \rho > 0$, we ensure that the softmax output $p_k(\beta)$ corresponding to the ground truth class $k$ is at least $\rho$ away from 1. We now state it as a proposition to derive $\beta$ given a lowerbound on $1 - p_k(\beta)$.

**Proposition 1.** Let $\mathbf{a}^K \in \mathbb{R}^d$ with $d > 1$ and $a_1^K \geq a_2^K \geq \ldots \geq a_d^K$ and $a_1^K - a_d^K = \gamma$. For a given $0 < \rho < (d-1)/d$, there exists a $\beta > 0$ such that $1 - \mathrm{softmax}(\beta a_1^K) > \rho$, then $\beta < -\log(\rho/(d-1)(1-\rho))/\gamma$.

*Proof.* This is derived via a simple algebraic manipulation of softmax. Please refer to Appendix. □

This $\beta$ can be used together with the one computed using inverse of mean Jacobian Singular Values (JSV). We provide the pseudocode for our proposed PGD++ (NJS) attack in Section A of Appendix. Similar approach can also be applied for FGSM++. Notice that, this approach is simple and it adds negligible overhead to the standard PGD attacks. However, it has a hand-designed hyperparameter $\rho$. To mitigate this, next we discuss a hyperparameter-free approach to obtain $\beta$.

**Hessian Norm Scaling (HNS).** We now discuss another approach to obtain informative gradients. Our idea is to maximize the Frobenius norm of the Hessian of the loss with respect to the input, where the intuition is that if the Hessian norm is large, then the gradient $\partial\ell/\partial\mathbf{x}^0$ is sensitive to an infinitesimal change in $\mathbf{x}^0$. This means, the infinitesimal perturbation in the input is propagated in the forward pass to the last layer and propagated back to the input layer without attenuation (*i.e.*, the returned signal is not zero), assuming there are no randomized or non-differentiable components in the network. This clearly indicates that the network has good signal propagation as well as the error signals are not all zero. This objective can now be written as:

$$\begin{aligned} \beta^* &= \operatorname*{argmax}_{\beta>0} \left\| \frac{\partial^2 \ell(\beta)}{\partial (\mathbf{x}^0)^2} \right\|_F \\ &= \operatorname*{argmax}_{\beta>0} \left\| \beta \left[ \psi(\beta)\frac{\partial \mathbf{J}}{\partial \mathbf{x}^0} + \beta \left( \frac{\partial \mathbf{p}(\beta)}{\partial \bar{\mathbf{a}}^K}\mathbf{J}\right)^T \mathbf{J} \right] \right\|_F . \end{aligned} \quad (8)$$

The derivation is provided in Appendix. Note, since $\mathbf{J}$ does not depend on $\beta$, $\mathbf{J}$ and $\partial\mathbf{J}/\partial\mathbf{x}^0$ are computed only once, $\beta$ is optimized using grid search as it involves only a single scalar. In fact, it is easy to see from the above equation that, when the Hessian is maximized, $\beta$ cannot be zero. Similarly, $\psi(\beta)$ cannot be zero because if it is zero, then the prediction $\mathbf{p}(\beta)$ is one-hot encoding (Eq. (7)), consequently $\partial\mathbf{p}(\beta)/\partial\bar{\mathbf{a}}^K = \mathbf{0}$ and this cannot be a maximum for the Hessian norm. Hence, this ensures that $\|\psi(\beta^*)\|_2 > \rho$ for some $\rho > 0$ and $\beta^*$ is bounded according to Proposition 1. Therefore, the maximum is obtained for a finite value of $\beta$. Even though, it is not clear how exactly this approach would affect the singular values of the input-output Jacobian ($\beta\,\mathbf{J}$), we know that they are finite and not zero.

Furthermore, there are some recent works (Moosavi-Dezfooli et al. 2019; Qin et al. 2019) show that adversarial training makes the loss surface locally linear around the vicinity of training samples and enforcing local linearity constraint on loss curvature can achieve better robust to adversarial attacks. On the contrary, our idea of maximizing the Hessian, *i.e.*, increasing the nonlinearity of $\ell$, could make the network more prone to adversarial attacks and we intend to exploit that. The psuedocode for PGD++ attack with HNS is summarized in Section A of Appendix.

## Experiments

We evaluate robustness accuracies of BNNs with weight quantized (BNN-WQ), weight and activation quantized (BNN-WAQ), floating point networks (REF). We evaluate our two PGD++ variants corresponding to HNS and NJS on CIFAR-10 and CIFAR-100 datasets with multiple network architectures. In order to demonstrate the effectiveness of our proposed variants on adversarially robust models, we also performed comparisons against stronger attacks such as DeepFool (Moosavi-Dezfooli, Fawzi, and Frossard 2016) and Brendel & Bethge Attack (BBA) (Brendel et al. 2019) on adversarially trained REF and BNN-WQ. We further provide experimental comparisons against more recent gradient based/free attacks (Auto-PGD (Croce and Hein 2020), Square Attack (Andriushchenko et al. 2020)) proposed to alleviate the issue of gradient obfuscation. More analysis on signal propagation issue in BNNs and our variants success in improving it is provided in Section C.5 of Appendix.

We use state of the art models trained for binary quantization (where all layers are quantized) for our experimental evaluations. We provide adversarial attack parameters used for FGSM/PGD in Table **??** of Appendix and for other attacks, we use default parameters used in Foolbox (Rauber, Brendel, and Bethge 2017). We also provide some other experimental comparisons such as comparisons against combinatorial attack proposed in (Khalil, Gupta, and Dilkina 2019) in the Appendix. For our HNS variant, we sweep $\beta$ from a range such that the hessian norm is maximized for each image, as explained in Appendix. For our NJS variant, we set the value of $\rho = 0.01$. In fact, our attacks are not very sensitive to $\rho$ and we provide the ablation study in the Appendix.

### Results

Our comparisons against the original PGD ($L_2/L_\infty$) and FGSM attack for different BNN-WQ are reported in Table 2. Our PGD++ variants consistently outperform original PGD on all networks on both datasets. Even being a gradient based attack, our proposed PGD++ ($L_2/L_\infty$) variants can in fact reach adversarial accuracy close to 0 on CIFAR-10 dataset, demystifying the fake robustness binarized networks tend to exhibit due to poor signal propagation.

Similarly, for one step FGSM attack, our modified versions outperform original FGSM attacks by a significant margin consistently for both datasets on various network architectures. We would like to point out such an improvement in the above two attacks is considerably interesting, knowing the fact that FGSM, PGD with $L_\infty$ attacks only use the sign of the gradients so improved performance indicates, our temperature scaling indeed makes some zero elements in the gradient nonzero. We would like to point out here that one can use several random restarts to increase the success rate of original form of FGSM/PGD attack further but to keep comparisons fair we use single random restart for both original and modified attacks. Nevertheless, as it has been observed in Table 1 even with 20 random restarts PGD adversarial accuracies for BNNs cannot reach zero, whereas our proposed PGD++ variants consistently achieve perfect success rate.

The adversarial accuracies of REF and BNN-WAQ trained on CIFAR-10 using ResNet-18/50, VGG-16 and DenseNet-121 for our variants against original counterparts are reported in Table 3. Overall, for both REF and BNN-WAQ, our variants outperform the original counterparts consistently. Particularly interesting, PGD++ variants improve the attack success rate on REF networks. This effectively expands the applicability of our PGD++ variants and encourages to consider signal propagation of any trained network to improve gradient based attacks. PGD++ with $L_\infty$ variants achieve near-perfect success rate on all BNN-WAQs, again validating the hypotheses of fake robustness of BNNs.

| | Network | Adversarial Accuracy (%) | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | FGSM | FGSM++ | | PGD ($L_\infty$) | PGD++ ($L_\infty$) | | PGD ($L_2$) | PGD++ ($L_2$) | |
| | | | NJS | HNS | | NJS | HNS | | NJS | HNS |
| CIFAR-10 | ResNet-18 | 40.49 | 3.46 | **2.51** | 26.98 | **0.00** | **0.00** | 55.68 | **0.05** | **0.05** |
| | VGG-16 | 57.55 | 4.00 | **3.43** | 47.32 | **0.00** | **0.00** | 56.66 | **0.35** | 1.32 |
| | ResNet-50 | 57.62 | 6.44 | **5.35** | 43.14 | **0.00** | **0.00** | 59.11 | 0.11 | **0.08** |
| | DenseNet-121 | 26.80 | 4.67 | **4.24** | 9.11 | **0.00** | **0.00** | 45.78 | **0.03** | 0.06 |
| | MobileNet-V2 | 33.50 | 6.42 | **5.42** | 26.86 | **0.00** | **0.00** | 34.40 | 0.12 | **0.09** |
| CIFAR-100 | ResNet-18 | 25.22 | 14.08 | **1.80** | 8.23 | 2.45 | **0.00** | 25.20 | 6.79 | **0.26** |
| | VGG-16 | 19.82 | 7.98 | **1.76** | 17.44 | 0.88 | **0.16** | 16.25 | 3.17 | **0.63** |
| | ResNet-50 | 37.76 | 16.33 | **14.17** | 25.71 | **2.33** | 2.73 | 30.77 | 7.90 | **7.41** |
| | DenseNet-121 | 28.32 | 12.21 | **10.86** | 8.87 | 1.15 | **1.09** | 24.65 | 4.54 | **4.16** |
| | MobileNet-V2 | 12.09 | 10.18 | **8.79** | 1.44 | **0.57** | 0.66 | 6.12 | 3.39 | **3.01** |

Table 2: *Adversarial accuracy on the test set for* BNN-WQ*. Both our* NJS *and* HNS *variants consistently outperform original $L_\infty$ bounded* FGSM *and* PGD *attack, and $L_2$ bounded* PGD *attack.*

| | Network | Adversarial Accuracy (%) | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | FGSM | FGSM++ | | PGD ($L_\infty$) | PGD++ ($L_\infty$) | | PGD ($L_2$) | PGD++ ($L_2$) | |
| | | | NJS | HNS | | NJS | HNS | | NJS | HNS |
| REF | ResNet-18 | 7.62 | 5.55 | **5.35** | **0.00** | **0.00** | **0.00** | 1.12 | 0.09 | **0.05** |
| | VGG-16 | 11.01 | 10.04 | **9.66** | 0.04 | **0.00** | **0.00** | 2.23 | **0.78** | 1.10 |
| | ResNet-50 | 21.64 | 6.08 | **5.70** | 0.69 | **0.00** | **0.00** | 0.37 | **0.07** | 0.09 |
| | DenseNet-121 | 11.40 | 7.58 | **7.30** | **0.00** | **0.00** | **0.00** | 0.65 | 0.08 | **0.06** |
| BNN-WAQ | ResNet-18 | 40.84 | 19.46 | **19.09** | 8.57 | **0.03** | 0.04 | 25.24 | **2.33** | 2.59 |
| | VGG-16 | 79.92 | 15.96 | **15.39** | 78.01 | **0.01** | 0.02 | 85.62 | **0.49** | 0.62 |
| | ResNet-50 | 33.16 | **25.89** | 27.05 | 0.49 | **0.23** | 0.45 | 19.41 | **6.68** | 8.77 |
| | DenseNet-121 | 37.20 | **23.89** | 24.69 | 0.81 | **0.10** | 0.18 | 48.37 | **3.72** | 6.17 |

Table 3: *Adversarial accuracy on the test set of CIFAR-10 for* REF *and* BNN-WAQ*. Both our* NJS *and* HNS *variants consistently outperform original* FGSM *and* PGD *($L_\infty/L_2$ bounded) attacks.*

To further demonstrate the efficacy of proposed attack variants, we first adversarially trained the BNN-WQs (quantized using BC (Courbariaux, Bengio, and David 2015), GD-tanh/MD-tanh-S (Ajanthan et al. 2021)) and floating point networks in a similar manner as in (Madry et al. 2017), using $L_\infty$ bounded PGD with $T = 7$ iterations, $\eta = 2$ and $\epsilon = 8$. We report the adversarial accuracies of $L_\infty$ bounded attacks and our variants on CIFAR-10 using ResNet-18 in Table 4. These results further strengthens the usefulness of our proposed PGD++ variants. Moreover, with a heuristic choice of $\beta = 0.1$ to scale down the logits before performing gradient based attacks performs even worse. Finally, even against stronger attacks (DeepFool, BBA) under the same $L_\infty$ perturbation bound, our variants outperform consistently on these adversarially trained models in Table 4. We would like to point out that our variants have negligible computational overhead over the original gradient based attacks, whereas stronger attacks are much slower in practice requiring 100-1000 iterations with an adversarial starting point (instead of random initial perturbation).

To illustrate the effectiveness of our proposed variants in improving signal propagation, we compare against gradient based attacks performed using recently proposed Difference of Logits Ratio (DLR) loss (Croce and Hein 2020) that aims to avoid the issue of saturating error signals. Also, we provide comparisons against recently introduced Auto-PGD (APGD) attack performed using DLR loss and a gradient free attack namely, Square Attack (Andriushchenko et al. 2020). We show these experimental comparisons performed on ResNet-18 models trained on CIFAR-10 dataset in Table 5. The attack parameters are same as used for the other experiments. It can be observed that our proposed variants perform better than both PGD or APGD with DLR loss and Square Attack, consistently achieving 0% adversarial accuracy. Infact, much computationally expensive Square attack is unable to achieve 0% adversarial accuracy in any of the cases under the enforced $L_\infty$ bound. The margin of difference is significant in case of FGSM attack and adversarial trained models. Infact, it is important to note that gradient based attacks with DLR loss and Square Attack perform worse on adversarially trained models than the original gradient based attacks.

**ImageNet.** For other large scale datasets such as ImageNet, BNNs are hard to train with full binarization of parameters and result in poor performance. Thus, most existing works (Yang et al. 2019) on BNNs keep the first and the last layers floating point and introduce several layer-wise scalars to achieve good results on ImageNet. In such experimental setups, according to our experiments, trained

| Network | Adversarial Accuracy (%) | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | FGSM | FGSM | FGSM++ | | PGD | PGD | Deep | BBA | PGD++ | |
| | | $\beta = 0.1$ | NJS | HNS | | $\beta = 0.1$ | Fool | | NJS | HNS |
| REF | 62.38 | 69.52 | 61.43 | **61.40** | 48.73 | 61.27 | 51.01 | 48.43 | **47.17** | 48.54 |
| BC | 53.91 | 62.46 | 52.90 | **52.27** | 41.29 | 54.24 | 42.65 | 40.14 | 39.35 | **39.34** |
| GD-tanh | 56.13 | 65.06 | 55.54 | **54.81** | 42.77 | 56.78 | 44.78 | 42.94 | **42.14** | 42.30 |
| MD-tanh-S | 55.10 | 63.42 | 54.74 | **53.82** | 41.34 | 54.22 | 43.46 | 40.69 | 40.76 | **40.67** |

Table 4: *Adversarial accuracy on CIFAR-10 with ResNet-18 for adversarially trained* REF *and* BNN-WQ *using different quantization methods (*BC, GD-tanh, MD-tanh-S*). Our improved attacks are compared against* FGSM, PGD *($L_\infty$), a heuristic choice of $\beta = 0.1$, DeepFool and* BBA. *Albeit on adversarially trained networks, our methods outperform all the comparable methods.*

| Network | Adversarial Accuracy (%) | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | FGSM | FGSM | FGSM++ | | PGD | PGD | APGD | Square | PGD++ | |
| | | (DLR) | NJS | HNS | | (DLR) | | Attack | NJS | HNS |
| REF | 7.62 | 19.48 | 5.55 | **5.35** | **0.00** | **0.00** | **0.00** | 0.55 | **0.00** | **0.00** |
| BNN-WQ | 40.49 | 19.72 | 3.46 | **2.51** | 26.98 | **0.00** | **0.00** | 0.41 | **0.00** | **0.00** |
| BNN-WAQ | 40.84 | 41.78 | 19.46 | **19.09** | 8.57 | 4.57 | 6.32 | 21.45 | **0.03** | 0.04 |
| REF$^*$ | 62.38 | 66.39 | 61.43 | **61.40** | 48.73 | 49.73 | 49.00 | 54.05 | **47.17** | 48.54 |
| BNN-WQ$^*$ | 55.10 | 59.14 | 54.74 | **53.82** | 41.34 | 41.42 | 40.85 | 46.67 | 40.76 | **40.67** |

Table 5: *Adversarial accuracy for* REF, BNN-WQ, *and* BNN-WAQ *trained on CIFAR-10 using ResNet-18. Here $^*$ denotes adversarially trained models. Both our* NJS *and* HNS *variants consistently outperform $L_\infty$ bounded* FGSM, PGD *and Auto-*PGD *(*APGD*) (Croce and Hein 2020) attack performed using Difference of Logits Ratio (*DLR*) loss and a gradient free attack namely, Square Attack (Andriushchenko et al. 2020) under $L_\infty$ bound (8/255). Notice,* FGSM, PGD *and* APGD *attack with* DLR *loss and Square Attack perform even worse than their original form on adversarially trained models in most cases.*

BNNs do not exhibit gradient masking issues or poor signal propagation and thus are easier to attack using original FGSM/PGD attacks with complete success rate. In such experiments, our modified versions perform equally well compared to the original forms of these attacks.

## Related Work

Adversarial examples are first observed in (Szegedy et al. 2014) and subsequently efficient gradient based attacks such as FGSM (Goodfellow, Shlens, and Szegedy 2014) and PGD (Madry et al. 2017) are introduced. There exist recent stronger attacks such as (Moosavi-Dezfooli, Fawzi, and Frossard 2016; Carlini and Wagner 2017; Yao et al. 2019; Finlay, Pooladian, and Oberman 2019; Brendel et al. 2019), however, compared to PGD, they are much slower to be used for adversarial training in practice. For a comprehensive survey related to adversarial attacks, we refer the reader to (Chakraborty et al. 2018).

Some recent works focus on the adversarial robustness of BNNs (Bernhard, Moellic, and Dutertre 2019; Sen, Ravindran, and Raghunathan 2020; Galloway, Taylor, and Moussa 2018; Khalil, Gupta, and Dilkina 2019; Lin, Gan, and Han 2019), however, a strong consensus on the robustness properties of quantized networks is lacking. In particular, while (Galloway, Taylor, and Moussa 2018) claims parameter quantized networks are robust to gradient based attacks based on empirical evidence, (Lin, Gan, and Han 2019) shows activation quantized networks are vulnerable to such attacks and proposes a defense strategy assuming the parameters are floating-point. Differently, (Khalil, Gupta, and Dilkina 2019) proposes a combinatorial attack hinting that activation quantized networks would have obfuscated gradients issue. Though as shown in the paper, the combinatorial attack is not scalable and thus experiments were shown on only few layered MLPs trained on MNIST. (Sen, Ravindran, and Raghunathan 2020) shows ensemble of mixed precision networks to be more robust than original floating point networks; however (Tramer et al. 2020) later shows the presented defense method can be attacked with minor modification in the loss function. In short, although it has been hinted that there maybe gradient masking in BNNs (especially in activation quantized networks), a thorough understanding is lacking on whether BNNs are robust, if not what is the reason for the failure of most commonly used gradient based attacks on binary networks. We answer this question in this paper and introduce improved gradient based attacks.

## Conclusion

In this work, we have shown that both BNN-WQ and BNN-WAQ tend to show a fake sense of robustness on gradient based attacks due to poor signal propagation. To tackle this issue, we introduced our two variants of PGD++ attack, namely NJS and HNS. Our proposed PGD++ variants not only possess near-complete success rate on binarized networks but also outperform standard $L_\infty$ and $L_2$ bounded PGD attacks on floating point networks. We finally show improvement in attack success on adversarially trained REF and BNN-WQ against stronger attacks (DeepFool and BBA).

## Acknowledgements

This work was supported by the Australian Research Council Centre of Excellence for Robotic Vision (project number CE140100016). We acknowledge the DATA61, CSIRO for their support and thank Puneet Dokania for useful discussions.

## References

Ajanthan, T.; Dokania, P. K.; Hartley, R.; and Torr, P. H. 2019. Proximal mean-field for neural network quantization. *ICCV*.

Ajanthan, T.; Gupta, K.; Torr, P.; Hartley, R.; and Dokania, P. 2021. Mirror descent view for neural network quantization. In *International Conference on Artificial Intelligence and Statistics*, 2809–2817. PMLR.

Andriushchenko, M.; Croce, F.; Flammarion, N.; and Hein, M. 2020. Square attack: a query-efficient black-box adversarial attack via random search. In *European Conference on Computer Vision*, 484–501. Springer.

Athalye, A.; Carlini, N.; and Wagner, D. 2018. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. *arXiv preprint arXiv:1802.00420*.

Bai, Y.; Wang, Y.-X.; and Liberty, E. 2019. Proxquant: Quantized neural networks via proximal operators. *ICLR*.

Bernhard, R.; Moellic, P.-A.; and Dutertre, J.-M. 2019. Impact of Low-bitwidth Quantization on the Adversarial Robustness for Embedded Neural Networks. In *2019 International Conference on Cyberworlds (CW)*, 308–315. IEEE.

Brendel, W.; Rauber, J.; Kümmerer, M.; Ustyuzhaninov, I.; and Bethge, M. 2019. Accurate, reliable and fast robustness evaluation. In *Advances in Neural Information Processing Systems*, 12861–12871.

Carlini, N.; and Wagner, D. 2017. Towards evaluating the robustness of neural networks. In *IEEE Symposium on Security and Privacy*.

Chakraborty, A.; Alam, M.; Dey, V.; Chattopadhyay, A.; and Mukhopadhyay, D. 2018. Adversarial attacks and defences: A survey. *arXiv preprint arXiv:1810.00069*.

Courbariaux, M.; Bengio, Y.; and David, J.-P. 2015. Binaryconnect: Training deep neural networks with binary weights during propagations. *NeurIPS*.

Croce, F.; and Hein, M. 2020. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. *ICML*.

Finlay, C.; Pooladian, A.-A.; and Oberman, A. 2019. The logbarrier adversarial attack: making effective use of decision boundary information. In *Proceedings of the IEEE International Conference on Computer Vision*, 4862–4870.

Galloway, A.; Taylor, G. W.; and Moussa, M. 2018. Attacking Binarized Neural Networks. In *International Conference on Learning Representations*.

Glorot, X.; and Bengio, Y. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, 249–256.

Goodfellow, I. J.; Shlens, J.; and Szegedy, C. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.

Guo, C.; Pleiss, G.; Sun, Y.; and Weinberger, K. Q. 2017. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, 1321–1330. JMLR. org.

Hubara, I.; Courbariaux, M.; Soudry, D.; El-Yaniv, R.; and Bengio, Y. 2017. Quantized neural networks: Training neural networks with low precision weights and activations. *JMLR*.

Khalil, E. B.; Gupta, A.; and Dilkina, B. 2019. Combinatorial Attacks on Binarized Neural Networks. In *International Conference on Learning Representations*.

Lin, J.; Gan, C.; and Han, S. 2019. Defensive Quantization: When Efficiency Meets Robustness. In *International Conference on Learning Representations*.

Lu, Y.; Gould, S.; and Ajanthan, T. 2020. Bidirectional Self-Normalizing Neural Networks. *arXiv preprint arXiv:2006.12169*.

Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; and Vladu, A. 2017. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*.

McDonnell, M. D. 2018. Training wide residual networks for deployment using a single bit for each weight. *ICLR*.

Moosavi-Dezfooli, S.-M.; Fawzi, A.; and Frossard, P. 2016. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2574–2582.

Moosavi-Dezfooli, S.-M.; Fawzi, A.; Uesato, J.; and Frossard, P. 2019. Robustness via curvature regularization, and vice versa. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 9078–9086.

Pennington, J.; Schoenholz, S.; and Ganguli, S. 2017. Resurrecting the sigmoid in deep learning through dynamical isometry: theory and practice. In *Advances in neural information processing systems*, 4785–4795.

Qin, C.; Martens, J.; Gowal, S.; Krishnan, D.; Dvijotham, K.; Fawzi, A.; De, S.; Stanforth, R.; and Kohli, P. 2019. Adversarial robustness through local linearization. In *Advances in Neural Information Processing Systems*, 13824–13833.

Rastegari, M.; Ordonez, V.; Redmon, J.; and Farhadi, A. 2016. Xnor-net: Imagenet classification using binary convolutional neural networks. *ECCV*.

Rauber, J.; Brendel, W.; and Bethge, M. 2017. Foolbox: A python toolbox to benchmark the robustness of machine learning models. *arXiv preprint arXiv:1707.04131*.

Saxe, A. M.; McClelland, J. L.; and Ganguli, S. 2013. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *arXiv preprint arXiv:1312.6120*.

Sen, S.; Ravindran, B.; and Raghunathan, A. 2020. EMPIR: Ensembles of Mixed Precision Deep Networks for Increased Robustness Against Adversarial Attacks. In *International Conference on Learning Representations*.

Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.; and Fergus, R. 2014. Intriguing properties of neural networks. In *International Conference on Learning Representations*.

Tramer, F.; Carlini, N.; Brendel, W.; and Madry, A. 2020. On adaptive attacks to adversarial example defenses. *arXiv preprint arXiv:2002.08347*.

Tramèr, F.; Kurakin, A.; Papernot, N.; Goodfellow, I.; Boneh, D.; and McDaniel, P. 2017. Ensemble adversarial training: Attacks and defenses. *arXiv preprint arXiv:1705.07204*.

Yang, J.; Shen, X.; Xing, J.; Tian, X.; Li, H.; Deng, B.; Huang, J.; and Hua, X.-s. 2019. Quantization networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 7308–7316.

Yao, Z.; Gholami, A.; Xu, P.; Keutzer, K.; and Mahoney, M. W. 2019. Trust region based adversarial attack on neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 11350–11359.