Improved Gradient based Adversarial Attacks for Quantized Networks

Kartik Gupta[1,2] and Thalaiyasingam Ajanthan[1]

[1]Australian National University [2]Data61, CSIRO

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 security. 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 CIFAR-10 dataset with VGG-16 and ResNet-18 networks demonstrate that our temperature scaled attacks obtain near-perfect success rate on quantized networks while outperforming original attacks on adversarially trained models and floating-point networks.

1. 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 [5], activation only quantized methods are shown to be vulnerable [10]. Although it has been hinted that there might be some sort of 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 inferior performance of gradient based attacks on binary networks. We answer this question in this paper and introduce improved PGD attacks.

In this work, 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 vanishing gradient issue due to poor forward-backward signal propagation caused by trained binary weights, and our idea is to improve signal propagation of the network without affecting the prediction of the classifier. To this end, we first discuss the conditions to ensure informative gradients and resort to a temperature scaling approach [8] to show that, even with a single positive scalar the vanishing gradients issue in BNNs can be alleviated achieving *near perfect success rate* in all tested cases. Specifically, we introduce a technique to choose the temperature scale based on the singular values of the input-output Jacobian. The justification for this is that if the singular values of input-output Jacobian are concentrated around 1 (defined as dynamical isometry [12], used to solve vanishing/exploding gradients issue) then the network is said to have good signal propagation and we intend to make the mean of singular values to be 1.

We evaluated our improved gradient based adversarial attacks on CIFAR-10 datasets with VGG-16 and ResNet-18 networks quantized using multiple recent techniques [1, 2, 3, 9]. In all tested quantized models, our temperature scaled attacks obtained near perfect success rate outperforming gradient based attacks (FGSM [7], PGD [11]) in their original form. Furthermore, this temperature scaling improved gradient based attacks even on adversarially trained models (both high-precision and quantized), showing the significance of signal propagation for adversarial attacks.

2. Robustness Evaluation of BNNs

We start by evaluating the adversarial accuracy of BNNs trained using various techniques, namely BC [4], PQ [3], PMF [1], MD-$\tanh$-S [2], BNN-WAQ [9] using the Projected Gradient Descent (PGD) attack [11] with L_∞ bound where the attack details are summarized below:

Algorithm	Clean	White Box	Black Box
REF	94.46	0.0	-
PMF [1]	93.24	33.02	2.2
PQ [3]	91.49	22.49	23.82
BNN-WAQ [9]	87.67	8.57	33.87
BC [4]	91.63	4.40	12.84
MD-tanh-S [2]	93.18	26.98	2.92

Table 1: *Recognition accuracy (clean vs. adversarial) on the test set of CIFAR-10 using ResNet-18 for* BNN*s with different methods for quantization.* BNN*s consistently outperform robustness accuracy of floating point networks (*REF*). All* BNN*s are parameter quantized except* BNN-WAQ *for which both weights and activations are quantized.*

- **PGD attack details**: perturbation bound of 8 pixels (assuming each pixel in the image is in $[0, 255]$) with respect to L_∞ norm, step size $\eta = 2$ and the total number of iterations $T = 20$. In all attacks, a randomized step is taken to initialize the perturbations. The attack details are the same in all evaluated setting unless stated otherwise.

We perform experiments on CIFAR-10 dataset using ResNet-18 architecture and report the clean and adversarial accuracy (white box and black box) results in Table 1. It can be clearly and consistently observed that BNNs have high white box adversarial accuracy whereas low or comparable black box adversarial accuracy. 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_∞ bound. This demonstrates that BNNs are prone to gradient masking (vanishing gradients) and exhibit a fake sense of security.

3. Signal Propagation of Neural Networks

In this section, let us first describe how poor signal propagation can cause vanishing or exploding gradients. We then discuss the idea of introducing a single scalar to improve the existing gradient descent 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 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:

$$\ell(\mathbf{a}^K, \mathbf{y}) = -\mathbf{y}^T \log(\mathbf{p}) \;, \qquad \mathbf{p} = \mathrm{softmax}(\mathbf{a}^K) \;, \quad (1)$$

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 (FGSM, PGD), gradient of the loss ℓ is used with respect to the input $\mathbf{x}^0$, which can also be formulated using chain rule as,

$$\frac{\partial \ell(\mathbf{a}^K, \mathbf{y})}{\partial \mathbf{x}^0} = \frac{\partial \ell(\mathbf{a}^K, \mathbf{y})}{\partial \mathbf{a}^K} \frac{\partial \mathbf{a}^K}{\partial \mathbf{x}^0} = \psi(\mathbf{a}^K, \mathbf{y})\, \mathbf{J} \;, \quad (2)$$

where ψ 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 $\times$ $\mathbf{u}$-size.

Notice there are two components that influence the gradients, 1) the Jacobian $\mathbf{J}$ and 2) the error signal ψ. 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 $\partial\ell/\partial\mathbf{x}^0$.

It is known that a network is said to satisfy dynamical isometry [12, 13] if the singular values of $\mathbf{J}$ are concentrated near 1, *i.e.*, for a given $\epsilon > 0$, the singular value σ_j satisfies $1 - \sigma_j \leq \epsilon$ for all j. Thus, just like dynamical isometry speeds up the training for the floating point networks by improving the signal propagation, a similar technique can be useful for gradient based attacks as well.

In fact, almost all initialization techniques (*e.g.*, [6]) approximately ensures that the Jacobian $\mathbf{J}$ is well-conditioned for better trainability. For continuous networks trained on the clean samples, approximate isometry is often preserved even at the end of the training but this is not the case for adversarially trained models or binary networks. In fact, for BNNs, the weights are constrained to be $\{-1, 1\}$ and hence the weight distribution at the end of training is completely different from the random initialization. We illustrate the signal propagation properties of various networks in Table 2.

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 the already trained model.

3.1. 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:

$$\mathbf{p}(\beta) = \mathrm{softmax}(\bar{\mathbf{a}}^K) \;, \qquad \bar{\mathbf{a}}^K = \beta\, \mathbf{a}^K \;. \qquad (3)$$

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

$$\frac{\partial \ell(\beta)}{\partial \mathbf{x}^0} = \frac{\partial \ell(\beta)}{\partial \bar{\mathbf{a}}^K} \frac{\partial \bar{\mathbf{a}}^K}{\partial \mathbf{a}^K} \frac{\partial \mathbf{a}^K}{\partial \mathbf{x}^0} = \psi(\beta)\beta\, \mathbf{J} \;. \qquad (4)$$

Note that β affects the input-output Jacobian linearly while it nonlinearly affects the error signal ψ. A heuristic choice

Methods	REF	Adv. Train	BC [4]	PQ [3]	PMF [1]	MD-tanh-S [2]	BNN-WAQ [9]
JSV (Mean)	8.09e+00	5.15e−01	1.61e+01	2.34e+01	4.46e+01	3.53e+01	1.11e+00
JSV (Std.)	6.27e+00	4.10e−01	1.88e+01	2.35e+01	1.11e+02	3.53e+01	1.97e+00
$\|\psi\|_2$ (Mean)	9.08e−03	2.33e−01	1.18e−02	6.75e−03	8.50e−03	6.20e−03	9.46e−03

Table 2: *Mean and standard deviation of Jacobian Singular Values (*JSV*) and mean $\|\psi\|_2$ for different methods on CIFAR-10 with* ResNet-18 *computed with 500 correctly classified samples. Note the norm of the error signal ψ is very small in all cases except for Adv. Train, indicating that all models are over confident (probabilities close to one-hot) except for Adv. Train which is in fact under confident for correctly classified samples. Furthermore, one can clearly see that* BNN*s (except* BNN-WAQ*) have much higher* JSV *mean and we believe this leads to gradient vanishing,* i.e., *increased scale for logits $\mathbf{a}^K$ and in turn reduced (if not zero) error signal ψ.*

of β (*e.g.* $\beta = 10$) might enable $\psi(\beta) > 0$ but can degrade conditioning of the Jacobian. To this end, we hope to obtain a β that ensures that 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.

4. Network Jacobian Scaling (NJS)

We now discuss a straightforward, two-step approach to attain the aforementioned properties on β. Firstly, to ensure $\beta\mathbf{J}$ is well-conditioned, we simply choose β to be the inverse of the mean of singular values $\mathbf{J}$. This guarantees that the mean of singular values of $\beta\mathbf{J}$ is 1. Formally, let us choose M samples from the test set, we can derive β as follows:

$$\beta = \frac{M\,d}{\sum_{i=1}^{M}\sum_{j=1}^{d}\mu_j(\mathbf{J}_i)}\ , \tag{5}$$

where $\mu_j(\mathbf{J}_i)$ denotes j^{th} singular value of the Jacobian $\mathbf{J}_i$ corresponding to the i^{th} sample.

After this Jacobian based scaling, there can be a situation where the 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 ρ away from 1. We now state it as a proposition to derive β given a lowerbound on $1 - p_k(\beta)$.

Proposition 1. Let $\mathbf{a}^K \in \mathrm{I\!R}^{\mathrm{d}}$ 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. Assuming $a_1^K - a_d^K = \gamma$, we derive a condition on β such that $1 - \mathrm{softmax}(\beta a_1^K) > \rho$.

$$\begin{aligned}
&\exp(\beta a_1^K)/\sum_{\lambda=1}^{d}\exp(\beta a_\lambda^K) < 1-\rho\ , &(6)\\
&1/\big(1+\sum_{\lambda=2}^{d}\exp(\beta(a_\lambda^K - a_1^K))\big) < 1-\rho\ ,\\
&1/\big(1+\sum_{\lambda=2}^{d}\exp(-\beta\gamma)\big) < 1-\rho\ , \quad a_1^K - a_\lambda^K \leq \gamma, \forall\lambda > 1\\
&\beta < -\log(\rho/(d-1)(1-\rho))/\gamma\ .
\end{aligned}$$

□

Algorithm 1 PGD++ with NJS with L_∞, T iterations, radius ϵ, step size η, network $f_{\mathbf{w}^*}$, input $\mathbf{x}^0$, label k, one-hot $\mathbf{y} \in \{0,1\}^d$, gradient threshold ρ.

```
Require: $T, \epsilon, \eta, \rho, \mathbf{x}^0, \mathbf{y}, k$
Ensure: $\|\mathbf{x}^{T+1} - \mathbf{x}^0\|_\infty \leq \epsilon$
  $\beta_1 = (M\,d)/\big(\sum_{i=1}^{M}\sum_{j=1}^{d}\mu_j(\mathbf{J}_i)\big)$          ▷ N/W Jacobian.
  $\mathbf{x}^1 = P_\infty^\epsilon(\mathbf{x}^0 + \mathrm{Uniform}(-1,1))$          ▷ Rand. Init.
  for $t \leftarrow 1, \ldots T$ do
    $\beta_2 = 1.0$
    $\mathbf{p}' = \mathrm{softmax}(\beta_1(f_{\mathbf{w}^*}(\mathbf{x}^t)))$
    if $1 - p'_k \leq \rho$ then          ▷ $\rho = 0.01$
      $\beta_2 = -\log(\rho/(d-1)(1-\rho))/\gamma$ ▷ Proposition 1
    $\ell = -\mathbf{y}^T\log(\mathrm{softmax}(\beta_2\beta_1(f_{\mathbf{w}^*}(\mathbf{x}^t))))$
    $\mathbf{x}^{t+1} = P_\infty^\epsilon(\mathbf{x}^t + \eta\,\mathrm{sign}(\nabla_{\mathbf{x}}\ell(\mathbf{x}^t)))$   ▷ Update Step
```

This β can be used together with the one computed in Eq. (5). We provide the pseudocode for our proposed PGD++ attack with NJS scaling in Algorithm 1. Similar approach can also be applied for FGSM++.

5. Experiments

In this section, we evaluate floating point networks (REF), parameter quantized networks (BC, PQ, PMF, MD-tanh-S) [4, 3, 1, 2] and weight and activation quantized network (BNN-WAQ) [9]. We evaluate our PGD++ attack corresponding Network Jacobian Scaling (NJS) on CIFAR-10 dataset with VGG-16 and ResNet-18 architectures. The values obtained for β using NJS are mostly greater than 1 and vary sample-wise. Briefly, our results indicate that our proposed attack yield attack success rate much higher than original PGD attacks on both floating point networks and binarized networks. Our proposed PGD++ attack also reduces PGD adversarial accuracy of adversarially trained floating point and adversarially trained binarized neural networks. We use the state of the art models trained for binary quantization from respective methods. For NJS, we set the value of $\rho = 0.01$.

5.1. L_∞ bounded Attacks

Attack details are: **PGD**: perturbation bound of $\epsilon = 8$ pixels, step size $\eta = 2$ and number of iterations $T = 20$, **FGSM**: step size $\eta = 8$. We compared the original PGD with improved PGD++ attack and the original FGSM with

Methods	ResNet-18		VGG-16		ResNet-18		VGG-16	
	FGSM	FGSM++	FGSM	FGSM++	PGD	PGD++	PGD	PGD++
REF	7.62	**5.55**	11.01	**10.04**	**0.00**	**0.00**	0.04	**0.00**
BC	11.15	**3.77**	27.38	**4.96**	4.40	**0.00**	9.28	**0.37**
PQ	52.97	**4.50**	27.46	**5.38**	22.49	**0.01**	23.41	**0.00**
PMF	48.65	**3.22**	54.87	**5.19**	33.02	**0.00**	54.11	**0.00**
MD-tanh-S	40.49	**3.46**	57.55	**4.00**	26.98	**0.00**	47.32	**0.00**
BNN-WAQ	40.84	**19.46**	79.92	**15.96**	8.57	**0.03**	78.01	**0.01**

Table 3: *Adversarial accuracy on the test set of CIFAR-10 dataset for binary neural networks using different methods for quantization comparing original* FGSM *attack with* FGSM++ *attack and original* PGD *attack with* PGD++ *attack.*

Methods	FGSM	FGSM++		PGD	PGD++	
		$\beta = 0.1$	NJS		$\beta = 0.1$	NJS
REF	62.38	69.52	**61.43**	48.73	61.27	**47.17**
BC [4]	53.91	62.46	**52.90**	41.29	54.24	**39.35**
GD-tanh [2]	56.13	65.06	**55.54**	42.77	56.78	**42.14**
MD-tanh-S [2]	55.10	63.42	**54.74**	41.34	54.22	**40.76**

Table 4: *Adversarial accuracy on the test set of CIFAR-10 with ResNet-18 for adversarially trained floating and binary neural networks using different methods for quantization comparing original L_∞ bounded* FGSM *attack with* FGSM++ *and original L_∞ bounded* PGD *attack with* PGD++.

improved FGSM++, on CIFAR-10 dataset using ResNet-18 and VGG-16 networks and the adversarial accuracies are reported in Table 3. Our PGD++ attack consistently outperforms original PGD on all binarized networks. Even being a gradient based attack, our proposed PGD++ attack can in fact reach adversarial accuracy close to 0 on CIFAR-10 dataset, demystifying the fake sense of robustness binarized networks tend to possess due to the poor signal propagation issue.

Similarly, even in the one step attack *i.e.* FGSM, our modified attack performs well. 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_∞ attacks only use the sign of the gradients so improved performance indicates, our temperature scaling indeed makes some zero elements in the gradient nonzero.

5.2. Adversarially Trained Models

To further demonstrate the efficacy, we first adversarially trained the parameter quantized networks and floating point networks in a similar manner as in [11], using L_∞ bounded PGD with $T = 7$ iterations, $\eta = 2$ and $\epsilon = 8$. We then evaluate the adversarial accuracies using L_∞ bounded PGD and PGD++ attack with $T = 20$, $\eta = 2$, $\epsilon = 8$ on CIFAR-10 dataset using ResNet-18 and the results are reported in Table 4. The adversarial accuracy results on adversarially trained binary and floating point networks further strengthens the usefulness of our proposed PGD++ attack using NJS. To illustrate the effectiveness of NJS over a heuristic choice of β, we show adversarial accuracy for FGSM++ with $\beta = 0.1$. The inferior adversarial accuracy with heuristic β reflects importance of an optimal way of estimating β.

6. Discussion

In this work, we have shown that BNNs suffer from gradient vanishing issues due to poor signal propagation. To tackle this, we introduced our PGD++ adversarial attack that possesses near-complete success rate on BNNs and also outperforms standard L_∞ PGD attacks on floating-point networks and adversarially trained models. In future, it would be interesting to compare exact robustness bounds on floating point networks and BNNs.

References

[1] T Ajanthan, PK Dokania, R Hartley, and PHS Torr. Proximal mean-field for neural network quantization. *ICCV*, 2019.

[2] T Ajanthan, K Gupta, PHS Torr, R Hartley, and PK Dokania. Mirror descent view for neural network quantization. *arXiv*, 2019.

[3] Y Bai, YX Wang, and E Liberty. Proxquant: Quantized neural networks via proximal operators. *ICLR*, 2019.

[4] M Courbariaux, Y Bengio, and JP David. Binaryconnect: Training deep neural networks with binary weights during propagations. *NeurIPS*, 2015.

[5] A Galloway, GW Taylor, and M Moussa. Attacking binarized neural networks. In *ICLR*, 2018.

[6] X Glorot and Y Bengio. Understanding the difficulty of training deep feedforward neural networks. In *AISTATS*, 2010.

[7] IJ Goodfellow, J Shlens, and C Szegedy. Explaining and harnessing adversarial examples. *arXiv*, 2014.

[8] C Guo, G Pleiss, Y Sun, and KQ Weinberger. On calibration of modern neural networks. In *ICML*, 2017.

[9] I Hubara, M Courbariaux, D Soudry, R El-Yaniv, and Y Bengio. Binarized neural networks. *NeurIPS*, 2016.

[10] J Lin, C Gan, and S Han. Defensive quantization: When efficiency meets robustness. In *ICLR*, 2019.

[11] A Madry, A Makelov, L Schmidt, D Tsipras, and A Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv*, 2017.

[12] J Pennington, S Schoenholz, and S Ganguli. Resurrecting the sigmoid in deep learning through dynamical isometry: theory and practice. In *NeuIPS*, 2017.

[13] AM Saxe, JL McClelland, and S Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *arXiv*, 2013.

This work was supported by the Australian Research Council Centre of Excellence for Robotic Vision (project number CE140100016).