Adversarial Robustness of Multi-bit Convolutional Neural Networks

Adversarial Robustness of Multi-Bit Convolutional Neural Networks

Lukas Frickenstein*[1], Shambhavi Balamuthu Sampath*[1], Pierpaolo Mori*[3], Manoj-Rohit Vemparala[1], Nael Fasfous[1], Alexander Frickenstein[1], Christian Unger[1], Claudio Passerone[3], Walter Stechele[2]
* indicates equal contributions

[1] BMW Autonomous Driving
[2] Technical University of Munich
[3] Politecnico di Torino
[1]firstname.lastname@bmw.de, [2]firstname.lastname@tum.de,
[3]firstname.lastname@polito.it

Abstract. Deploying convolutional neural networks (CNNs) on resource-constrained, embedded hardware constitutes challenges in balancing task-related accuracy and resource-efficiency. For safety-critical applications, a third optimization objective is crucial, namely the robustness of CNNs. To address these challenges, this paper investigates the tripartite optimization problem of task-related accuracy, resource-efficiency, and adversarial robustness of CNNs by utilizing multi-bit networks (MBNs). To better navigate the tripartite optimization space, this work thoroughly studies the design space of MBNs by varying the number of weight and activation bases. First, the pro-active defensive model MBN3x1 is identified, by conducting a systematic evaluation of the design space. This model achieves better adversarial accuracy (+10.3 pp) against the first-order attack PGD-20 and has 1.3× lower bit-operations, with a slight degradation of natural accuracy (-2.4 pp) when compared to a 2-bit fixed-point quantized implementation of ResNet-20 on CIFAR-10. Similar observations hold for deeper and wider ResNets trained on different datasets, such as CIFAR-100 and ImageNet. Second, this work shows that the defensive capability of MBNs can be increased by adopting a state-of-the-art adversarial training (AT) method. This results in an improvement of adversarial accuracy (+13.6 pp) for MBN3x3, with a slight degradation in natural accuracy (-2.4 pp) compared to the costly full-precision ResNet-56 on CIFAR-10, which has 7× more bit-operations. To the best of our knowledge, this is the first paper highlighting the improved robustness of differently configured MBNs and providing an analysis on their gradient flows.

Keywords: Adversarial Robustness, Neural Network Quantization, Multi-Bit Convolutional Neural Networks

1 Introduction

Convolutional neural networks (CNNs) have become prevalent in the field of computer-vision, tackling a wide-range of complex problems with unprecedented task-related accuracy [18,35]. Despite their rise in popularity, several drawbacks have limited their adoption in embedded, safety-critical settings [9–11]. Most prominent of these drawbacks are their increasing demand in memory and computational complexity [18], as well as their susceptibility to edge-cases and adversarial attacks [2]. This has led to extensive research into CNN compression [1,16, 17,20,21,26,30,31,37,38] and defensive adversarial training [13,25,33,40]. Among the wide-range of techniques in both research domains, binary neural networks (BNNs) have been shown to effectively tackle both challenges by constraining the parameters of the CNN to $\mathbf{B} \in \{-1, 1\}$. This highly discrete representation of the parameters allows for the execution of the costly arithmetic operations of a CNN by simple boolean XNOR operations on inference hardware [7, 31], and requiring 1-bit of memory to store the CNN's parameters and intermediate activations. The discreteness of BNNs necessitates a complex training scheme involving gradient approximation to allow gradient-descent-based training over the discrete binarization functions [8, 27, 31]. As first-order adversarial-attacks try to exploit vulnerabilities of the latent weight representations, the binarized weights used during inference show higher resilience to the produced adversarial example [12, 29]. Through this gradient approximation at training-time, the severity of gradient-based adversarial attacks is reduced. Although BNNs partly tackle both problems of resource-constrained, embedded inference and adversarial robustness, their learning capacity is hampered due to the low-information representation of their parameters, resulting in lower task-related natural accuracy. To address the low task-related accuracy of BNNs, research into multi-bit networks (MBNs) increased the number of binary representations a single layer can have [26]. A single full-precision weight filter or input feature map can be represented with an arbitrary number of binary tensors, called bases.

Considering all three aspects of parsimonious inference, robustness of BNNs and improved natural task-related accuracy of MBNs, the design choices involved in producing the deployed neural network become a balancing act of maintaining all three desired targets in a tripartite solution space. This work navigates this solution space and builds an understanding of the interactions between the three target optimization criteria, which is tested empirically on a wide-range of possible MBN configurations. To the best of our knowledge, this is the first paper highlighting the improved robustness of differently configured MBNs and providing an analysis on their gradient flows. The contributions of this work can be summarized as follows:

- Performing a thorough investigation of the design space of MBNs by varying the number of weight and activation bases. With the correct choice of bases, a model with +10.3 pp better adversarial accuracy against the *ultimate* first-order attack PGD-20 and 1.3× lower bit-operations can be found when compared to a 2-bit fixed-point quantized implementation of ResNet-20 trained on CIFAR-10, with a slight degradation in natural accuracy (-2.4 pp).

- Showing that the defensive capability of MBNs can be increased by adapting a state-of-the-art adversarial training (AT) method. With different architectural setups, various practical trade-offs can be achieved, such as an improvement of adversarial accuracy (+13.6 pp) for MBN3x3, with a slight degradation in natural accuracy (-2.4 pp) compared to the costly full-precision ResNet-56 on CIFAR-10, while providing a 7× improvement in bit-operations.
- Supporting the empirical evidence by formulating an understanding of the gradient flows of full-precision, fixed-point quantized, and multi-bit networks, relating to different levels of adversarial robustness.

2 Related Work

2.1 Quantized and Binary Neural Networks

Quantization relies on representing a CNN's parameters with a discrete, constrained set of values. This typically requires complex training schemes to enable standard stochastic gradient descent (SGD) to update the model parameters. In [42], Zhou *et al.* limit the magnitude of the latent weights and activations between [0, 1], where the latent datatypes are deterministically quantized such that the straight-through estimator (STE) [4] is required. Choi *et al.* aim to improve the training scheme of quantized neural networks (QNNs) with PACT [6] by learning the optimal clipping level for the activations of each layer at training time. Thus, the representational capability is increased, leading to an increase in task-related accuracy.

Binarization represents the most drastic form of quantization, where the parameters of a CNN are constrained to either $\{+1, -1\}$. A common solution to train BNNs is to maintain full-precision latent parameters θ used during training to update the highly discrete model parameters θ_b with gradient information. For the forward pass, the latent model parameters are deterministically mapped to either $\{+1, -1\}$ through the `sign` function. However, this creates a gradient-vanishing problem during backpropagation, as the derivative of the `sign` function is mostly zero, causing all the gradients of parameters before the `sign` to take the value of zero. To tackle this issue, Bengio *et al.* [4] propose the straight-through-estimator (STE) which passes the identity if the argument is positive. This can be seen as passing the gradients through the piece-wise linear activation function `hard tanh` (`htanh`). This ensures sufficient gradient flow during backpropagation to update all the BNN parameters, bypassing the `sign` operation.

With BinaryNet [7], Courbariaux and Bengio presented the first BNN with binary weights and activations. Building on top of basic BNNs, Rastegari *et al.* introduced XNOR-Net [31], a scheme to train BNNs with latent weights by approximating the convolutions of input feature maps A^{l-1} and weights W^l of the layer l by a combination of XNOR operations and popcounts, multiplied with a trainable scaling factor α, resulting in: $\texttt{Conv}(A^{l-1}, W^l) \approx (\texttt{sign}(A^{l-1}) \oplus \texttt{sign}(W^l)) \cdot \alpha$. This introduced a significant improvement in accuracy, as the added trainable scaling factor could allow more information to be learned by the

BNN. To further mitigate the accuracy degradation of BNNs, Lin *et al.* [26] extended BNNs by approximating the full-precision convolutions in CNNs by using linear combinations of multiple binary bases for both weights M and activations N, resulting in Accurate Binary Convolutional Neural Networks (ABC-Nets). Thus, the convolutions of multi-bit networks (MBN) can be implemented by computing $M \times N$ bit-wise convolutions in parallel.

2.2 Adversarial Robust Compression

Szegedy *et al.* [36] first proved the existence of adversarial attacks in the domain of image classification. Adversarial examples are generated by adding some imperceptible perturbation δ to some given original input images, fooling the network into changing its prediction. Carlini *et al.* [5] proposed three characteristics that specify a defined threat model τ of the adversarial attack. First, the *adversary goal* defines a successful attack. Second, the *capability of adversarial attacks* can be formulated as a set of allowed perturbations $\mathcal{S} : D(X, X_{adv}) \leq \epsilon$, where some distance D between the original and the adversarial image does not extend some perturbation budget ϵ. Third, the degree of accessibility of the adversarial attack to the underlying neural network defines the *adversarial knowledge*. For white-box attacks (*e.g.* PGD), the complete model parameters are exposed to the adversary. In general, Carlini *et al.* [5] describe the problem of finding an adversarial example X_{adv} for a given model $f(\cdot)$ and the label Y as maximizing the loss $\mathcal{L}$ for a given perturbation budget ϵ as shown in Eq. 1.

$$\mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}} \left[\max_{X_{adv}\in\mathcal{S}} \mathcal{L}\left(f(X_{adv}), Y\right) \right] \tag{1}$$

In the work of Goodfellow *et al.* [14], a simple and fast method of generating adversarial examples is introduced, namely the Fast Gradient Sign Method (FGSM). FGSM performs a step into the ascent direction of the gradient of the loss function, scaled by the perturbation budget ϵ. Madry *et al.* [28] use a more powerful adversary to maximize the loss by introducing the multi-step variant of FGSM, which represents Projected Gradient Descent (PGD). Inspired by Tramér *et al.* [39], Madry *et al.* initialize the PGD attack by randomly choosing starting points inside $\mathcal{S}$. This ensures varying explorations of the non-concave and constrained maximization problem by the PGD attack, which is able to converge to local maxima. As stated by Madry *et al.* [28], this renders the PGD attack as the *ultimate* first-order adversary. A more detailed description is provided in Sec. 4.1. In the following sections, we apply the PGD attack to full-precision CNNs, QNNs, BNNs, a wide range of MBNs, as well as adversarially-trained networks. We leverage our understanding of the different gradient flow characteristics in these types of neural networks (presented in Sec. 2.1) to explain their robustness (or lack thereof).

Recent works addressed the simultaneous optimization with regard to network compression and adversarial robustness. In [13], Goldblum *et al.* leverage knowledge distillation to distill adversarial robustness onto a smaller student

from a larger teacher network. In [15], Guo *et al.* investigated to mitigate potential threats of adversarial examples through robust neural architecture search (NAS) techniques. With [24], Kundu *et al.* aim for highly compressed CNNs while maintaining their robustness through robust pruning. Galloway *et al.* [12] evaluated and interpreted the adversarial robustness of BNNs. The reduced memory consumption and faster inference of BNNs is complemented with adversarial robustness, by demonstrating an improved or at least on par robustness against several attacks compared to full-precision models. The introduced discontinuity and approximated gradients of BNNs account for the improved robustness over the full-precision networks. Building on this knowledge, our work aims to exploit the increased representation capabilities of MBNs to boost natural accuracy compared to BNNs, while utilizing the resilience of binary parameters against adversarial attacks. With [25], Lin *et al.* jointly optimize the efficiency and robustness of DNNs by robust quantization.

In Tab. 1, we summarize the exploration and investigations performed in existing literature. In this work, we holistically consider the tripartite optimization space by using multi-bit networks and analyse the effect of MBN design decisions on all three optimization targets.

Table 1: Classification of related work explorations of the tripartite optimization space, considering the natural accuracy, the HW efficiency and the robustness against adversarial attacks.

Exploration	[6, 26, 31, 42]	[28, 40, 41]	[12, 13, 15, 25]	[This Work]
Accuracy	✓	✓	✗	✓
HW-Efficiency	✓	✗	✓	✓
Robustness	✗	✓	✓	✓

3 Methodology

This paper navigates the tripartite optimization problem by studying the design space of MBNs. In this section, the considered design space of MBNs is formulated with varying number of bases for weights and activations (Sec. 3.1). Furthermore, we aim to analyse the gradient flows of multi-bit networks to achieve different levels of adversarial robustness, and compare them to floating-point and fixed-point networks along with addressing the *gradient obfuscation* problem known in literature when assessing the robustness of quantized and binarized CNNs [3](Sec. 3.2). Lastly, the number of bit-operations (`BOPS`) is formulated, allowing an evaluation and comparison of the complexity of neural networks, such as BNNs, QNNs, MBNs and full-precision representations (Sec. 3.3).

3.1 Design Space of Multi-Bit Networks

As noted in Sec. 2.1, the accuracy gap between full-precision networks and BNNs can be mitigated by approximating the full-precision convolutions as a linear combination of multiple binary bases M for weights and N for activations (shown in Eq. 2). Although each convolution of two individual binary bases m and n still has a limited information capacity, the *learned* linear combination of the $M \times N$ binary convolutions is collectively more capable of representing the information of a full-precision convolution more accurately.

$$\texttt{Conv}(W, A) \approx \sum_{m=1}^{M} \sum_{n=1}^{N} \alpha_m \beta_n \texttt{Conv}(\texttt{sign}(W_m), \texttt{sign}(A_n)) \tag{2}$$

The multiple binary activations $A_{b,n} = \texttt{sign}(A_n)$, their corresponding scaling coefficients β_n and the multiple binary weight approximations $W_{b,m} = \texttt{sign}(W_m)$ along with the weight scaling coefficients α_m result in the whole convolution scheme. Each convolution of a weight and activation base-pair can be computed bit-wise independently from other convolutions. More specifically, the multi-bit convolution operation can be parallelized by up to $M \times N$ convolutions, requiring the end-to-end latency of a single, standard binary convolution. It is important to note, that parallelizing bit-wise computations of an equivalent QNN is not possible due to the data dependencies among the bits belonging to the same elements of the convolution operands. We can intuitively expect that increasing $M \times N$ up to a certain extent, increases the accuracy of the MBN, at the cost of more bit-operation computations. However, to understand the effect of increasing (or decreasing) the number of bases (for weights and/or activations) on adversarial robustness (naturally or adversarially trained) is a complex problem, requiring more in-depth analysis and experimental evaluation. Throughout this work we follow the notation of MBN $M \times N$ representing a multi-bit network configurations with M number of weight bases and N number of activation bases.

3.2 Analysing Gradient Flows

In this section, valuable insights into the tripartite optimization problem are extracted, by analysing the differences in the forward pass and the gradient flows in the backward pass of convolutions with different numerical representation methods i.e. fixed-point, multi-bit, and floating-point networks (see Fig. 1).

Multi-Bit: The forward pass of multi-bit networks follows a linear combination of multiple binary bases M and N, described in Eq. 2. In the backward pass, the gradients of the weights g_{W^l} of the layer l and the gradients of the activations $g_{A^{l-1}}$ are computed as a concatenation of scaling coefficients and the identities of the gradients of the binary bases, as shown in Eq. 3.

$$g_{A^{l-1}} = \sum_{n=1}^{N} \beta_n 1_{A \leq 1} \frac{\delta A^l}{\delta A_{b,n}^{l-1}} \quad ; \quad g_{W^l} = \sum_{m=1}^{M} \alpha_m 1_{W \leq 1} \frac{\delta A^l}{\delta W_{b,m}^{l}} \tag{3}$$

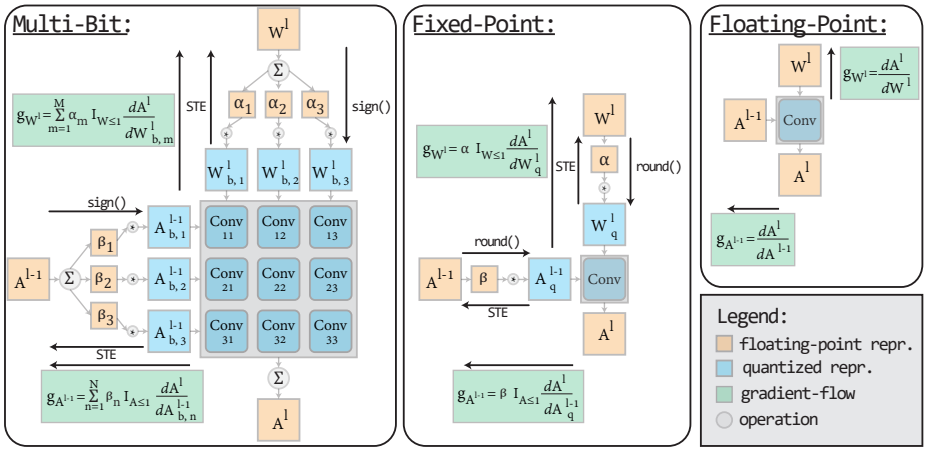

Fig. 1: Comparison of convolutions with different numerical representation methods i.e. multi-bit (left), fixed-point (middle) and floating-point (right).

Here, each binary base for weights $W_{b,m}$ and activations $A_{b,n}$ has separate trainable scaling parameters, α_m's and β_n's respectively, increasing the learning capabilities with increased number of bases. For multi-bit convolutions the individual discrete `sign` function is bypassed with the STE to allow gradients to flow, representing a coarse approximation. This induces an increased degree of gradient approximation (i.e. between latent and actual binary parameters), producing large discrepancies between forward and backward pass.

Fixed-Point: Considering the approximated convolutions of fixed-point CNNs, the `round`() operand is used to map from full-precision latent parameters (A^{l-1} and W^l) to the fixed-point quantized representations (A_q^{l-1} and W_q^l), as shown in Eq. 4.

$$\texttt{Conv}(W^l, A^{l-1}) \approx \alpha\beta\texttt{Conv}(\texttt{round}(W^l, q_W), \texttt{round}(A^{l-1}, q_A)) \tag{4}$$

The scaling and shifting happens before `round`(), such that $W \in [0, 1]$. Note that the number of bits used to represent weights and activations is denoted by q_W and q_A respectively, ranging from integer values q =[1; 8]. The gradients in the backward pass follow as in Eq. 5.

$$g_{A^{l-1}} = \beta 1_{A \leq 1} \frac{\delta A^l}{\delta A_q^{l-1}} \quad ; \quad g_{W^l} = \alpha 1_{W \leq 1} \frac{\delta A^l}{\delta W_q^l} \tag{5}$$

Convolutions of fixed-point networks rely on a single trainable scaling factor α for weights and β for activations, collapsing into one scaling factor. In the backward pass, the `round`() operation is bypassed with the STE for weights and activations. This bypassing represents a closer approximation of the gradients compared to the bypassing of the `sign` as done in multi-bit convolutions, leading to smaller discrepancies between forward and backward passes for fixed-point networks.

The well-established computations of floating-point DNNs do not induce any discrepancies, as they do not require any discrete operations. Analysing the forward and backward pass of the convolutions of different numerical representations, the gradient analysis points to the following implications:

- First, the multi-bit convolution introduces large discrepancies between the forward and backward pass, due to the harsh approximation of the `sign` function with the STE for each binary base. The huge gradient approximation demands stronger gradient-based attacks with larger perturbations to change the output of MBNs. This implies an improved inherent resilience of MBNs compared to floating-point and fixed-point networks.
- Second, as the number of binary bases of MBNs increase, the approximation gap to the floating-point convolution and its respective gradient flow decreases. This implies that an increased number of bases reduces the inherent robustness against gradient-based attacks.
- Lastly, adversarial training methods demand an increased learning ability, due to the additional examples provided by the attacks. This suggests that training MBNs with adversarial examples requires an increased number of binary bases to improve the trade-off between natural accuracy and adversarial robustness.

Obfuscated Gradients: In general, iterative optimization-based attacks (e.g. PGD (see Sec. 2.2)) require gradient information of the underlying model to create strong adversarial attacks. A main concern in literature are false defense mechanisms exploiting the effect of gradient masking, or the special case of obfuscated gradients [3]. The phenomenon of obfuscated gradients can lead to a false sense of security against adversarial attacks by not providing enough gradient information. We are **highlighting** the fact that our investigations on different quantization techniques in this work neither exploit shattered gradients, stochastic gradients nor vanishing/exploding gradients, representing the three types of gradient obfuscation. This work follows the common practice to use the STE [4] to ensure gradient flow over all non-differentiable operations (*e.g.* `round`(), `sign`()). This is ensured for *both* training the neural network *and* the process of generating the adversarial attack for the underlying model. This is also detailed in Fig. 1, showing CNN layers of different numerical representation methods, all allowing the backward pass to bypass the non-differentiable quantization functions with the STE.

3.3 Compute Complexity

Hardware arithmetic computation units can be classified as either being bit-serial or bit-parallel (vectorized) in their processing of the input operands. To fairly compare the computational complexity of low-precision and binary neural networks, they must be evaluated on hardware architectures which can exploit their respective benefits. In this paper, we refer to bit-serial based HW improvements, as they are able to flexibly process *any* bit-width inputs, albeit with an added

latency as the input bit-width grows. Bit-serial computation units break down the operands and perform the computation bit-by-bit until the bit-width of both inputs is exhausted [34]. In principle, two 1-bit (binary) operands need 1 cycle of computation, whereas two 16-bit operands require 256 cycles to complete an arithmetic operation. As such, we use the Bit-OPs (`BOPS`) metric to evaluate and compare the complexity of neural networks, for all BNN, MBN, QNN and full-precision representations.

4 Experiments

Breaking CNNs can be achieved by simply adding large perturbations onto the input. However, finding the minimum necessary input perturbations is more practical to understanding the robustness of CNNs [29]. Choosing ϵ for PGD is based on breaking the full-precision version of ResNet [18], as we aim to compare floating-point, fixed-point and multi-bit networks and their resilience against PGD. If the classification accuracy drops below random guessing, the target model is considered broken. The PGD threat model τ^{IN}_{PGD}={ϵ=2, α=0.5} is used to assess the inherent (IN) robustness of the target models, representing the worst-case threat model for the mentioned numerical representations and the considered perturbation budget $\epsilon = 2$. The process of identifying the worst-case threat model for all considered models is described in Sec. 4.1. For adversarially trained (AT) models [40], τ^{AT}_{PGD}={ϵ=8, α=2} is considered to further stress the target models. The reported PGD accuracies are averaged over five runs, ensuring varying explorations of the set of allowed perturbations $\mathcal{S}$, as described in Sec. 2.2.

Experiments are carried out on CIFAR-10/100 [23] and ImageNet [32]. For CIFAR-10, 50K train and 10K test images (32×32 pixels) are used to train and evaluate the multi-bit configurations of ResNet-20/56. ImageNet consists of $\sim$ 1.28M train and 50K validation images (256×256 pixels), on which ResNet-18 is trained and evaluated as well as various multi-bit configurations of ResNet-18 are trained and evaluated. If not otherwise mentioned, all hyper-parameters defining the training or the attacks are adopted from the reference implementations.

First, the worst-case PGD threat model is identified for different network configurations by varying the step size α (Sec. 4.1). Second, the design space of MBNs is systematically evaluated by (1) naturally training the networks on original image data, (2) evaluating the resource-efficiency of the configurations based on the number of `BOPS` and (3) assessing the inherent adversarial robustness against the ultimate first-order adversarial attack (PGD) (Sec. 4.2). We train and evaluate the multi-bit configurations of ResNet-20 , ResNet-56, and ResNet-18 on CIFAR-10, CIFAR-100, and ImageNet, respectively. Third, combining MBNs with state-of-the-art adversarial training methods (*e.g.* FastAT [40]) can further increase the defensive capability, showing the learning capabilities of larger MBNs (Sec. 4.3). We adversarially train and evaluate multi-bit configurations of ResNet-56 on CIFAR-10/100. To assess the performance with respect to the tripartite optimization space, we report the prediction accuracy on original images

(Top-1), attacked images (PGD-20/50) and the number of bit-operations (BOPS). Note that for floating-point networks, we report the BOPS of the 8-bit version since we expect no natural accuracy degradation [22], however, the respective Top-1 and PGD experiments are still performed on classic 32-bit floating-point CNNs.

4.1 Worst-Case Threat-Model

The iterative multi-step PGD attack, introduced by Madry *et al.* [28], maximizes the problem of finding adversarial examples. PGD performs iterative steps into the ascent direction of the gradient of the corresponding target model's loss function, scaled by the PGD step size α, and projected π onto the legal set $\mathcal{S}$, see Eq. 6.

$$X_{adv}^{i+1} = \pi_{\mathcal{S}} \left(X_{adv}^{i} + \alpha \cdot \nabla \mathcal{L} \left(X_{adv}^{i}, Y, \theta \right) \right) \tag{6}$$

Inspired by Tramér *et al.* [39], Madry *et al.* initialize the PGD attack by randomly choosing starting points with uniform random noise $\mathcal{U}(-\epsilon, \epsilon)$ inside the defined legal set $\mathcal{S}$, with $\mathcal{S} : D(X, X_{adv}) \leq \epsilon$. This ensures random starting points on the highly non-concave maximization problem of finding adversarial examples, where PGD is able to converge to local maxima. The local maxima represent possible worst-case adversarial examples for the target model. Having the non-concave loss surface in mind, starting from random starting points results in subsequent varying exploration of potentially varying local maxima, representing different worst-case scenarios. As stated by Madry *et al.* [28], this renders the PGD attack as the *ultimate* first-order adversary. Therefore, we follow the relevant literature [24,25] which suggests using PGD attack, an iterative optimization-based attack, to evaluate the adversarial robustness of CNNs.

The threat model for PGD τ_{PGD} comprises of the perturbation budget ϵ, the step size α, and the iterations i. As stated by Carlini *et al.* [5], adapting the adversarial threat model of state-of-the-art adversarial attacks is a compulsory step to demonstrate an upper bound of adversarial robustness. Therefore, the following performs an exploration of the PGD attack configuration to identify the worst-case threat model τ_{PGD}^{*} over a variety of numerical representations i.e. floating-point (Fig. 2), fixed-point (Fig. 3) and multi-bit (Fig. 4).

Identifying the worst-case threat model τ_{PGD}^{*} for various compressed CNN variants relies on two metrics. First, the overall accuracy level after converged attack for a specific τ_{PGD}, where the worst-case results in the lowest accuracy after attack. Second, the required number of iterations i of PGD to break the model. Having the metrics in mind, the goal of the PGD threat model exploration is to empirically identify a value for the step size α such that it results in the worst-case PGD attack for the underlying model. In general, the PGD step size α balances out convergence speed and the characteristic of escaping local maxima.

Versions of ResNet-20 and ResNet-56 are naturally trained on CIFAR-10, followed by exposing them to the PGD attack with $\epsilon = 2$, while varying the PGD step size $\alpha = \{0.1, 0.5, 1\}$ to find the most suitable value. The PGD attacks

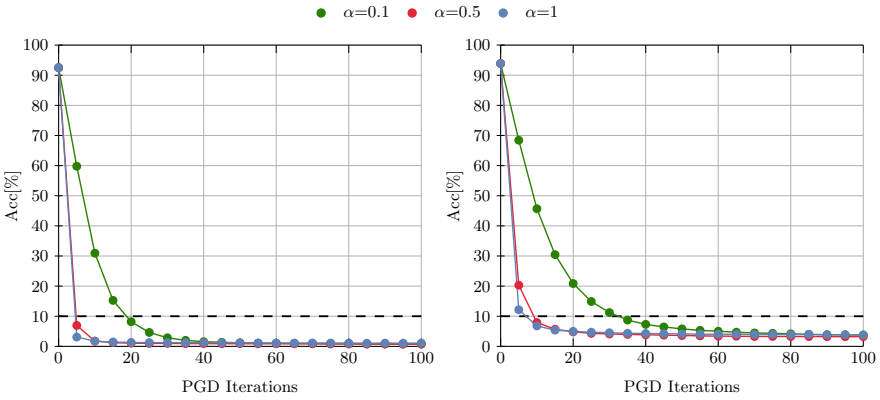

Fig. 2: PGD attack accuracy over PGD iterations for ResNet-20 (left) and ResNet-56 (right) with floating-point representation averaged over five runs on CIFAR-10 with a fixed perturbation budget ϵ and varying step size parameters α.

perform $1,000$ iterations i to ensure convergence of the multi-step optimization-based attack, as proposed by Carlini *et al.* [5]. However, it is worth mentioning that the following results are shown up to $i = 100$, since the attack convergence is observed in that range. Additionally, each variant is exposed to all threat models τ_{PGD} five times to utilize the described uniform random initialization of the attack. The results reported in this experiment are then averaged over the five runs.

Fig. 2 visualizes the threat model exploration for floating-point versions of ResNet-20 (left) and ResNet-56 (right). Similarly, Fig. 3 shows the threat model exploration for fixed-point quantized versions of ResNet-20 (left column) and ResNet-56 (right column). In detail, PACT-2bit [6] (top row) and PACT-4bit [6] (bottom row) are exposed to the defined PGD threat models. Lastly, Fig. 4 contains the threat model exploration for various multi-bit versions of ResNet-20 (left column) and ResNet-56 (right column). MBN1x3 (top row), MBN3x1 (middle row) and MBN3x3 (bottom row) are exposed to the PGD attacks as defined in the experimental setup.

The worst-case threat model was identified for all investigated CNNs and numerical representations as $\tau^*_{PGD} = \{\epsilon = 2, \alpha = 0.5\}$. First, τ^*_{PGD} results in the highest PGD attack effectiveness over all numerical representations (i.e. floating-point, fixed-point, multi-bit) for ResNet-20/56. Second, τ^*_{PGD} requires an adequate number of iterations to break the target models, rendering PGD-20/50 as a valid assessment of inherent robustness of the target models.

4.2 Inherent Robustness of Multi-bit Networks

We compare the inherent robustness of MBNs to full-precision (ResNet [18]), fixed-point (PACT [6]), binary (XNOR-Net [31]), and pruned (AMC [19]) net-

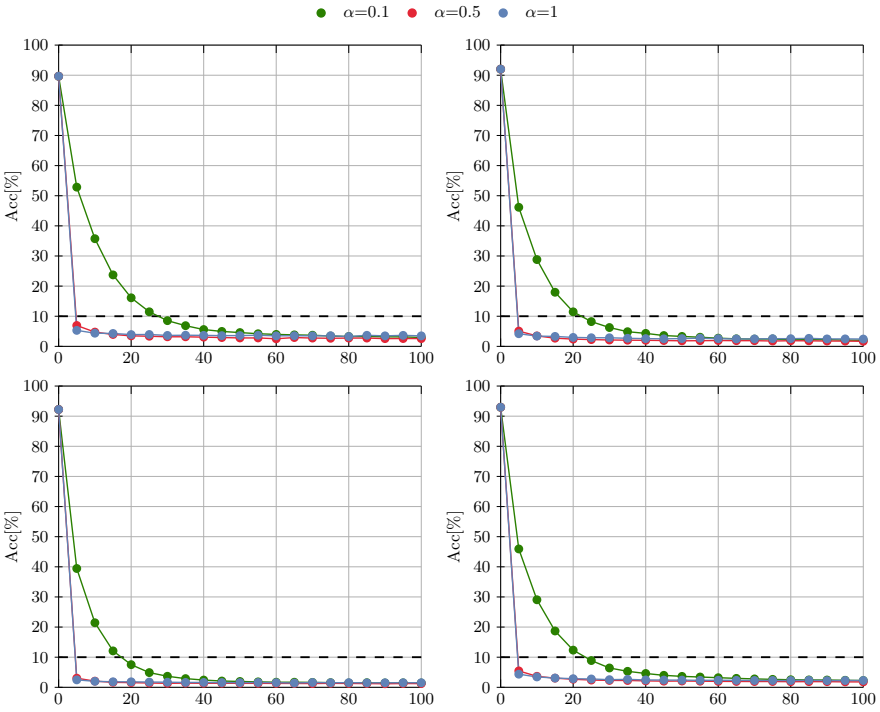

Fig. 3: PGD attack accuracy over PGD iterations for ResNet-20 (left column) and ResNet-56 (right column) variants with fixed-point representations averaged over five runs for PACT-2bit (top row) and PACT-4bit (bottom row) on CIFAR-10. Similarly, the perturbation budget ϵ is fixed while step size parameter α varies for the threat model.

works, showing the increased resilience of multi-bit networks to the gradient-based PGD attack, implied in Sec. 3.2. To better understand the gradient flows for MBNs and the influence of varying the number of bases M for weights and N for activations on the tripartite optimization, various MBN configurations are systematically evaluated. The extensive exploration of the design search space is presented in Tab. 2.

In general, MBNs provide an increased adversarial accuracy against PGD, compared to other numerical representations. Particularly, the pro-active defensive model MBN3x1 achieves better adversarial accuracy (+10.3 pp) against PGD-20 and has 1.3× lower `BOPS` , with a slight degradation of original accuracy (-2.4 pp) when compared to a 2-bit fixed-point quantized implementation of ResNet-20 on CIFAR-10. Similar trends hold for the ResNet-56 and ResNet-18 versions of MBN3x1, trained on CIFAR-100 and ImageNet respectively. This highly supports the theoretical implication, provided in Sec. 3.2, that the induced discrepancies between forward and backward pass of multi-bit networks

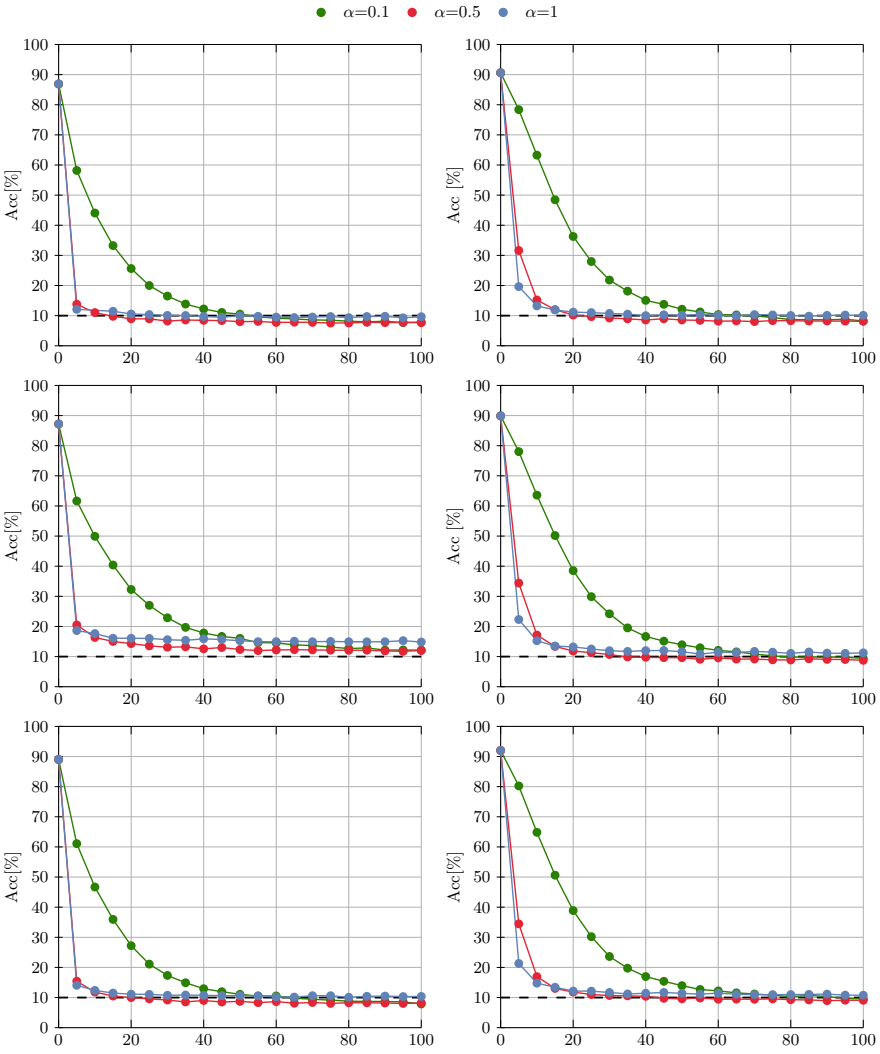

Fig. 4: PGD attack accuracy over PGD iterations for ResNet-20 (left column) and ResNet-56 (right column) variants with multi-bit representation averaged over five runs for MBN1x3 (top row), MBN3x1 (middle row) and MBN3x3 (bottom row) on CIFAR-10.

Table 2: Comparisons of naturally trained (inherent robustness) multi-bit, full-precision, fixed-point and pruned networks of ResNet-20/56/18, on CIFAR-10/100 and ImageNet respectively.

Model/ Dataset	Method	Number of Bases/Bits Weights	Activations	BOPS [10^8]	Top-1 [%]	PGD-20 [%]	PGD-50 [%]
ResNet-20 CIFAR-10	ResNet-20 [18]	8	8	25.95	92.49	1.02	0.81
	AMC W. Pr. [19]	8	8	12.98	88.71	0.82	0.69
	AMC Ch. Pr. [19]	8	8	12.98	89.71	0.77	0.62
	PACT-4 [6]	4	4	6.70	92.21	1.49	1.27
	PACT-2 [6]	2	2	1.89	89.63	3.94	3.33
	XNOR-Net [31]	1	1	0.68	83.98	9.93	8.28
	MBN [26]	1	1	0.68	84.44	11.10	9.59
		3	**1**	**1.49**	**87.19**	**14.28**	**12.54**
		1	3	1.49	87.54	9.09	7.96
		5	1	2.29	87.01	14.13	12.55
		1	5	2.29	89.02	8.68	7.71
		7	1	3.09	86.88	14.29	12.80
		1	7	3.09	89.49	6.49	5.74
		3	3	3.98	89.03	10.00	8.78
		5	5	10.31	90.86	7.60	6.57
ResNet-56 CIFAR-10	ResNet-56 [18]	8	8	80.31	93.89	4.75	3.54
	PACT-4 [6]	4	4	20.29	92.98	2.43	2.01
	PACT-2 [6]	2	2	5.29	92.08	2.46	2.03
	XNOR-Net [31]	1	1	1.53	85.61	17.23	15.47
	MBN [26]	1	1	1.53	87.39	9.04	7.63
		3	1	4.03	89.82	11.07	9.67
		1	3	4.03	90.76	9.83	8.50
		5	1	6.54	89.77	11.08	9.47
		1	5	6.54	91.78	8.83	7.56
		7	1	9.04	89.78	10.55	9.16
		1	7	9.04	92.25	7.48	6.35
		3	3	11.54	91.96	10.99	9.73
		5	5	31.54	92.68	9.52	7.58
ResNet-56 CIFAR-100	ResNet-56 [18]	8	8	80.31	72.62	0.90	0.82
	PACT-4 [6]	4	4	20.29	70.44	0.95	0.88
	PACT-2-4 [6]	2	4	10.29	70.40	2.42	2.19
	PACT-2 [6]	2	2	5.29	67.79	1.56	1.46
	XNOR-Net [31]	1	1	1.53	58.46	6.76	6.25
	MBN [26]	1	1	1.53	60.67	4.08	3.49
		3	**1**	**4.03**	**65.41**	**3.94**	**3.47**
		1	3	4.03	66.21	2.76	2.48
		5	1	6.54	66.23	4.13	3.49
		1	5	6.54	68.10	2.62	2.33
		7	1	9.04	64.75	4.88	4.22
		1	7	9.04	69.23	2.24	1.97
		3	3	11.54	68.40	3.00	2.63
		5	5	31.54	69.74	2.64	2.30
ResNet-18 ImageNet	ResNet-18 [18]	8	8	1259.6	69.01	0.07	0.07
	PACT-2 [6]	2	2	149.8	60.04	0.04	0.03
	MBN [26]	1	1	94.3	43.52	0.39	0.32
		3	**1**	**131.3**	**56.18**	**0.75**	**0.64**
		1	3	131.3	57.12	0.41	0.34
		3	3	242.3	60.49	0.64	0.48

provide an increased resilience against PGD compared to other numerical representations. We notice that limiting the number of bases for the activations $N = 1$, while choosing the number of weight bases $M > 1$, produces more robust MBN configurations compared to MBN configurations with weight bases $M = 1$ and activation bases $N > 1$. This behavior is associated with the structural characteristics of the activation bases, where the number of activation bases determine the gradient flow to the next layer. Since the activation bases serve as "gates" of information flow to the next layer, increasing the number of gates enables the attack to exploit more information about the gradients. Considering the tripartite optimization space, MBN3x1 is a favourable solution, providing a practical trade-off between the three objectives. Similar trends hold for various model complexities of ResNet, across multiple datasets.

4.3 Adversarial Training of Multi-bit Networks

The defensive capability of NNs can be increased by adversarial training (*e.g.* FastAT [40]). We train a variety of full-precision, fixed-point, pruned and multi-bit networks, to show the learning capabilities of MBNs in detecting both original and adversarial images (see Tab. 3).

In general, MBNs achieve better resilience against PGD, compared to floating-point, fixed-point quantized, and pruned versions of ResNet over various model complexities and datasets. The multi-bit configuration MBN3x3 improves the adversarial robustness (+13.6 pp) against PGD-20 and has 7× lower `BOPS`, with a slight degradation of original accuracy (-2.4 pp) when compared to the costly full-precision implementation of ResNet-56 on CIFAR-10. Unlike fixed-point models, increasing the number of binary bases for MBNs further scales the original accuracy and robustness against PGD. This empirically supports the implication, that adversarial training demands large learning capacities, as provided by MBNs with increased number of binary bases for weights and activations.

5 Conclusion

This work aims to find the balance among the three objectives of task-related accuracy, resource-efficiency, and robustness of CNNs by utilizing MBNs. The solution space is navigated, performing an analysis and thorough evaluation of the design space of MBNs by varying the number of weight and activation bases. Their inherent robustness is assessed against the gradient-based PGD attack and their learning capabilities in the context of adversarial training, compared to floating-point, fixed-point, and pruned networks. To the best of our knowledge, this is the first paper highlighting the improved robustness of differently configured MBNs and providing an analysis on their gradient flows. The proactive configuration MBN3x1 improves the robustness by +10.3 pp, providing 1.3× fewer bit-operations, with a slight degradation in natural accuracy by -2.4 pp, compared to a 2-bit implementation of ResNet-20 trained on CIFAR-10.

Table 3: Adversarial robustness comparison of multi-bit to floating-point, fixed-point and pruned networks of ResNet-56, adversarially trained on CIFAR-10 and CIFAR-100.

Model/ Dataset	Method	Number of Bases/Bits Weights	Activations	BOPS [10^8]	Top-1 [%]	PGD-20 [%]
FastAT [40] CIFAR-10	ResNet-56 [18]	8	8	80.31	84.03	38.45
	AMC W. Pr. [19]	8	8	40.16	83.94	41.04
	PACT-4 [6]	4	4	20.29	85.56	40.48
	PACT-2 [6]	2	2	5.29	81.80	45.98
	XNOR-Net [31]	1	1	1.53	75.64	44.15
	MBN [26]	3	1	4.03	77.01	51.98
		1	3	4.03	79.42	51.00
		5	1	6.54	75.27	50.10
		1	5	6.54	80.76	50.02
		3	**3**	**11.54**	**81.65**	**52.02**
		5	5	31.54	82.01	50.71
FastAT [40] CIFAR-100	ResNet-56 [18]	8	8	80.31	56.17	23.64
	PACT-4 [6]	4	4	20.29	58.32	22.14
	PACT-2 [6]	2	2	5.29	54.86	22.53
	XNOR-Net [31]	1	1	1.53	42.52	27.25
	MBN [26]	3	1	4.03	47.23	25.88
		1	3	4.03	50.35	26.09
		5	1	6.54	49.14	27.46
		1	5	6.54	52.70	25.50
		3	**3**	**11.54**	**53.49**	**26.70**
		5	5	31.54	55.25	26.56

References

1. Sungsoo Ahn, Shell Xu Hu, Andreas C. Damianou, Neil D. Lawrence, and Zhenwen Dai. Variational information distillation for knowledge transfer. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9155–9163, 2019.
2. Naveed Akhtar and Ajmal S. Mian. Threat of Adversarial Attacks on Deep Learning in Computer Vision: A Survey. *IEEE Access*, 6:14410–14430, 2018.
3. Anish Athalye, Nicholas Carlini, and David A. Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *ICML*, pages 274–283, 2018.
4. Yoshua Bengio, N. Léonard, and Aaron C. Courville. Estimating or Propagating Gradients Through Stochastic Neurons for Conditional Computation. *ArXiv*, abs/1308.3432, 2013.
5. Nicholas Carlini, Anish Athalye, Nicolas Papernot, Wieland Brendel, Jonas Rauber, Dimitris Tsipras, Ian J. Goodfellow, Aleksander Madry, and Alexey Kurakin. On Evaluating Adversarial Robustness. *CoRR*, abs/1902.06705, 2019.
6. Jungwook Choi, Zhuo Wang, Swagath Venkataramani, Pierce I.-Jen Chuang, Vijayalakshmi Srinivasan, and Kailash Gopalakrishnan. PACT: Parameterized Clip-

ping Activation for Quantized Neural Networks. *arXiv:1805.06085 [cs]*, July 2018. arXiv: 1805.06085.
7. Matthieu Courbariaux and Yoshua Bengio. BinaryNet: Training Deep Neural Networks with Weights and Activations Constrained to +1 or -1.
8. Sajad Darabi, Mouloud Belbahri, Matthieu Courbariaux, and Vahid Partovi Nia. BNN+: Improved Binary Network Training. abs/1812.11800, Jan. 2018.
9. Nael Fasfous, Manoj-Rohit Vemparala, Alexander Frickenstein, Mohamed Badawy, Felix Hundhausen, Julian Höfer, Christian Unger Naveen-Shankar Nagaraja, Hans-Jörg Vögel, Jürgen Becker, Tamim Asfour, and Walter Stechele. Binary-LoRAX: Low-power and Runtime Adaptable XNOR Classifier for Semi-Autonomous Grasping with Prosthetic Hands. In *International Conference on Robotics and Automation*, 2021.
10. Nael Fasfous, Manoj-Rohit Vemparala, Alexander Frickenstein, Lukas Frickenstein, Mohamed Badawy, and Walter Stechele. BinaryCoP: Binary Neural Network-based COVID-19 Face-Mask Wear and Positioning Predictor on Edge Devices. In *IPDPS-RAW*, 2021.
11. Alexander Frickenstein, Manoj Rohit Vemparala, Jakob Mayr, Naveen Shankar Nagaraja, Christian Unger, Federico Tombari, and Walter Stechele. Binary DAD-Net: Binarized Driveable Area Detection Network for Autonomous Driving. *2020 IEEE International Conference on Robotics and Automation*, pages 2295–2301, 2020.
12. Angus Galloway, Graham W. Taylor, and Medhat Moussa. Attacking Binarized Neural Networks. In *International Conference on Learning Representations (ICLR)*, 2018.
13. Micah Goldblum, Liam Fowl, Soheil Feizi, and Tom Goldstein. Adversarially robust distillation. In *Conference of Association for the Advancement of Artificial Intelligence (AAAI)*, volume 34, 2020.
14. Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and Harnessing Adversarial Examples. In *International Conference on Learning Representations (ICLR)*, 2015.
15. Minghao Guo, Yuzhe Yang, Rui Xu, and Ziwei Liu. When nas meets robustness: In search of robust architectures against adversarial attacks, 2019.
16. Yiwen Guo, Anbang Yao, and Yurong Chen. Dynamic Network Surgery for Efficient DNNs. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems (NeurIPS)*, volume 29, pages 1379–1387. Curran Associates, Inc., 2016.
17. Song Han, Jeff Pool, John Tran, and William Dally. Learning both Weights and Connections for Efficient Neural Network. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems (NeurIPS)*, volume 28, pages 1135–1143. Curran Associates, Inc., 2015.
18. Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, Jun. 2016.
19. Yihui He, Ji Lin, Zhijian Liu, Hanrui Wang, Li-Jia Li, and Song Han. AMC: AutoML for Model Compression and Acceleration on Mobile Devices. In *Proceedings of the European Conference on Computer Vision (ECCV)*, volume 7, pages 815–832, Sep. 2018.
20. Geoffrey Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the Knowledge in a Neural Network. In *NIPS Deep Learning and Representation Learning Workshop*, 2015.

21. Qiangui Huang, Shaohua Kevin Zhou, Suya You, and Ulrich Neumann. Learning to Prune Filters in Convolutional Neural Networks. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 709–718, 2018.
22. Benoit Jacob, Skirmantas Kligys, Bo Chen, Menglong Zhu, Matthew Tang, Andrew Howard, Hartwig Adam, and Dmitry Kalenichenko. Quantization and training of neural networks for efficient integer-arithmetic-only inference. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.
23. Alex Krizhevsky. Learning Multiple Layers of Features from Tiny Images. *University of Toronto*, May 2012.
24. Souvik Kundu, Mahdi Nazemi, Peter A. Beerel, and Massoud Pedram. Dnr: A tunable robust pruning framework through dynamic network rewiring of dnns. In *Proceedings of the 26th Asia and South Pacific Design Automation Conference*, ASPDAC '21, page 344–350, New York, NY, USA, 2021. Association for Computing Machinery.
25. Ji Lin, Chuang Gan, and Song Han. Defensive quantization: When efficiency meets robustness. In *International Conference on Learning Representations (ICLR)*, 2019.
26. Xiaofan Lin, Cong Zhao, and Wei Pan. Towards Accurate Binary Convolutional Neural Network. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30, pages 345–353. Curran Associates, Inc., 2017.
27. Zechun Liu, Baoyuan Wu, Wenhan Luo, Xin Yang, Wei Liu, and Kwang-Ting Cheng. Bi-Real Net: Enhancing the Performance of 1-bit CNNs with Improved Representational Capability and Advanced Training Algorithm. In *Proceedings of the European Conference on Computer Vision (ECCV)*, Sep. 2018.
28. Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards Deep Learning Models Resistant to Adversarial Attacks. In *International Conference on Learning Representations (ICLR)*, 2018.
29. Manoj Rohit Vemparala, Alexander Frickenstein, Nael Fasfous, Lukas Frickenstein, Qi Zhao, Sabine Franziska Kuhn, Daniel Ehrhardt, Yuankai Wu, Christian Unger, Naveen Shankar Nagaraja, Walter Stechele. Breakingbed - breaking binary and efficient deep neural networks by adversarial attacks. In *Intelligent Systems Conference*, 2021.
30. Manoj Rohit Vemparala, Nael Fasfous, Lukas Frickenstein, Alexander Frickenstein, Anmol Singh, Driton Salihu, Christian Unger, Naveen-Shankar Nagaraja, Walter Stechele. Hardware-aware mixed-precision neural networks using in-train quantization. In *British Machine Vision Conference*, 2021.
31. Mohammad Rastegari, Vicente Ordonez, Joseph Redmon, and Ali Farhadi. XNOR-Net: ImageNet Classification Using Binary Convolutional Neural Networks. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Proceedings of the European Conference on Computer Vision (ECCV)*, volume 14, pages 525–542, Cham, 2016. Springer International Publishing.
32. Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.
33. Ali Shafahi, Mahyar Najibi, Mohammad Amin Ghiasi, Zheng Xu, John Dickerson, Christoph Studer, Larry S Davis, Gavin Taylor, and Tom Goldstein. Adversarial

training for free! In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems (NeurIPS)*, volume 32, pages 3358–3369. Curran Associates, Inc., 2019.
34. S. Sharify et al. Loom: Exploiting weight and activation precisions to accelerate convolutional neural networks. In *DAC*, 2018.
35. Evan Shelhamer, Jonathan Long, and Trevor Darrell. Fully Convolutional Networks for Semantic Segmentation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3431–3440, 2015.
36. Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In Yoshua Bengio and Yann LeCun, editors, *International Conference on Learning Representations (ICLR)*, 2014.
37. Wei Tang, Gang Hua, and Liang Wang. How to Train a Compact Binary Neural Network with High Accuracy? In *Proceedings of the Conference on Artificial Intelligence (AAAI)*, volume 31, pages 2625–2631. AAAI Press, 2017.
38. Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive Representation Distillation. In *International Conference on Learning Representations (ICLR)*, volume 8, 2020.
39. Florian Tramèr, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. The Space of Transferable Adversarial Examples, 2017.
40. Eric Wong, Leslie Rice, and J. Zico Kolter. Fast is better than free: Revisiting adversarial training. In *International Conference on Learning Representations (ICLR)*, 2020.
41. Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric P. Xing, Laurent El Ghaoui, and Michael I. Jordan. Theoretically principled trade-off between robustness and accuracy. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the International Conference on Machine Learning, (ICML)*, volume 97 of *Proceedings of Machine Learning Research*, pages 7472–7482. PMLR, 2019.
42. 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, 2018.