IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received April 12, 2019, accepted May 21, 2019, date of publication May 27, 2019, date of current version June 10, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2919463

Discretization Based Solutions for Secure Machine Learning Against Adversarial Attacks

PRIYADARSHINI PANDA, (Student Member, IEEE), INDRANIL CHAKRABORTY, AND KAUSHIK ROY, (Fellow, IEEE)
School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907, USA

Corresponding author: Priyadarshini Panda (pandap@purdue.edu)

This work was supported in part by the C-BRIC, Center for Brain-Inspired Computing, a JUMP Center sponsored by the Defense Advanced Research Projects Agency (DARPA) and Semiconductor Research Corporation (SRC), in part by SRC, in part by the National Science Foundation, in part by Intel Corporation, in part by the Vannevar Bush Faculty Fellowship, and in part by the U.K. Ministry of Defense under Grant W911NF-16-3-0001.

ABSTRACT Adversarial examples are perturbed inputs that are designed (from a deep learning network's (DLN) parameter gradients) to mislead the DLN during test time. Intuitively, constraining the dimensionality of inputs or parameters of a network reduces the "space" in which adversarial examples exist. Guided by this intuition, we demonstrate that discretization greatly improves the robustness of the DLNs against adversarial attacks. Specifically, discretizing the input space (or allowed pixel levels from 256 values or 8*bit* to 4 values or 2*bit*) extensively improves the adversarial robustness of the DLNs for a substantial range of perturbations for minimal loss in test accuracy. Furthermore, we find that binary neural networks (BNNs) and related variants are intrinsically more robust than their full precision counterparts in adversarial scenarios. Combining input discretization with the BNNs furthers the robustness, even waiving the need for adversarial training for the certain magnitude of perturbation values. We evaluate the effect of discretization on MNIST, CIFAR10, CIFAR100, and ImageNet datasets. Across all datasets, we observe maximal adversarial resistance with 2*bit* input discretization that incurs an adversarial accuracy loss of just $\sim 1\% - 2\%$ as compared to clean test accuracy against single-step attacks. We also show standalone discretization remains vulnerable to stronger multi-step attack scenarios necessitating the use of adversarial training with discretization as an improved defense strategy.

INDEX TERMS Adversarial robustness, deep learning, discretization techniques, binarized neural networks.

I. Introduction

Deep Learning Networks (DLNs) have exhibited better than human performance in several vision-related tasks [1]. However, they have been recently shown to be vulnerable toward adversarial attacks [2]–[4]: slight changes of input pixel intensities can fool a DLN to misclassify an input with high confidence (Fig. 1). What is more worrying is that such small changes (that craft adversaries) are visually imperceptible to humans, yet, mislead a DLN. This vulnerability severely limits the potential safe-use and deployment of DLNs in real-world scenarios. For instance, an attacker may fool a DLN deployed on a self-driving car to mispredict a STOP sign as a GO signal, and cause fatal accidents.

Subsequently, there have been several theories pertaining to the adversarial susceptibility of DLNs [3]. The most common one suggests that the presence of adversary is an outcome of the excessive linearity of a DLN (a property of high-dimensional dot-products). While one can argue that rectified linear unit (ReLU) activation imposes non-linearity in a model, the linear operations such as Convolution, Pooling exceed the number of non-linear ReLU operations. Further, ReLU is typically a linear functionality in the > 0 regime, and hence, plagues a DLN to be sufficiently linear. Now, this linearity causes a model to extrapolate its behavior for points in the hyper-space (of data and model parameters) that lie outside the training/test data manifold. Adversarial inputs, essentially, are images that are synthesized such that they 'lie far' from the typical data manifold and hence get misclassified.

Fig. 2 (a) demonstrates this data manifold intuition and adversarial input creation with a cartoon. Since DLNs are discriminative models, they partition a very high-dimensional

The associate editor coordinating the review of this manuscript and approving it for publication was Rajeeb Dey.

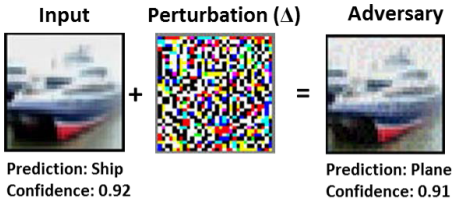


FIGURE 1. *An image of a ship perturbed with adversarial noise yields an adversarial image that fools the classifier. The classifier predicts the original image correctly with a confidence of 92%, while gets fooled by the adversarial image mispredicting it as plane with a high confidence of 91%.*

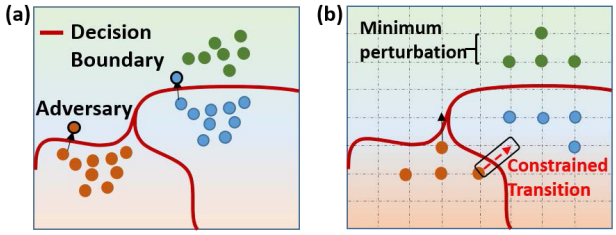


FIGURE 2. *Cartoon of the intuition behind adversary creation and discretization. (a) The data points (shown as 'dots') encompass the data manifold in the high-dimensional subspace. The classifier is trained to separate the data into different categories or hyper-volumes based on which the decision boundary is formed. Note, the decision boundary is a characteristic of the trained parameters (weights) of the model. The decision boundary is, however, extrapolated to vast regions of the high-dimensional subspace that are unpopulated and untrained because of linear model behavior. Adversaries are created by perturbing the data points into these empty regions or hyper-volumes and are thus misclassified (orange mispredicted as green in this case).*
(b) Discretization quantizes the data manifold thereby introducing a minimum perturbation required to shift a data point. As quantization will increase, so will the minimum allowed distortion. Further, discretization constrains the creation of adversaries since not all transitions can cause a data point to shift between hyper-volumes.

input space into different classes by learning appropriate decision boundaries. The class-specific decision boundaries simply divide the space into hyper-volumes. Interestingly, these hyper-volumes encompass the training data examples as well as large areas of unpopulated space that is arbitrary and untrained. This extrapolation of decision boundary beyond the training data space is a result of *'linearity'*, that in turn, gives rise to generalization ability. The fact that a model trained only on training data is able to predict well on unseen test data (termed as, generalization) is a favorable outcome of this extrapolation. Unfortunately, generalization also exposes a model to adversarial attacks. Adversarial data are created by simply adding small perturbations to an input data point, that shifts it from its manifold (or hyper-volume) to a different hyper-volume (that the model has not been trained upon and shows extrapolated behavior), causing misclassification.

From the above intuition, one can deduce that adding regularization features to a DLN's training will improve adversarial robustness. In fact, the most effective form of adversarial defense so far is training a model with adversarial data augmentation (called adversarial training) [5]. It is evident that explicit training on adversarial data will increase the model's capability to generalize and hence, predict correctly on unseen adversarial data. However, the above discussion on excessive linearity and hyper-space dimensionality points to an alternate and unexplored regularization possibility, that is discretizing or constraining the data manifold for achieving adversarial robustness. For instance, discretizing the input data (say from 256-pixel value levels (or 8-bit) to 4 levels (or 2-bit)) reduces the regions into which data can be perturbed. In other words, the minimum perturbation required to shift a particular data point from one hyper-volume to another will increase in a discretized space (Fig. 2 (b)). This in turn will intrinsically improve the resistance of a DLN. Similarly, discretizing the parameter space (as in binarized neural networks (BNNs) [6]) will introduce discontinuities and quantization in the manifold (that is non-linear by nature). This will further decrease the extent of hyper-volume space that is arbitrary/untrained and thus reduce adversarial susceptibility (Fig. 2(b)). It is evident that such discretization methods have an added advantage of computational efficiency. In fact, low-precision neural networks (BNNs and related XNOR-Nets [7]) were introduced with a key motif of obtaining reduced memory and power-consumption for hardware deployment of DLNs.

In this paper, we demonstrate that discretization, besides offering obvious efficiency improvements, has far-reaching implication on a model's adversarial resistance. We particularly emphasize on three different discretization themes and illustrate their suitability toward improving a DLN's adversarial robustness, as follows:

- *Discretization of input space:* We reduce the input dimensionality by quantizing the RGB pixel intensities into a variable range: $2^8 = 256$ to $2^2 = 4$. We show that for minimal loss in accuracy, the adversarial robustness of a model substantially improves ($<1\%$ accuracy difference between clean test and adversarial test data), even, without any adversarial training. Furthermore, we show that combining adversarial training with 2-bit input discretization makes a model substantially more robust (than adversarial training with full 8-bit input precision) for large perturbation ranges.
- *Discretization of parameter space:* We show that models trained with low-precision weights and activations, such as BNNs, are intrinsically more robust to adversarial perturbations than full precision networks. Furthermore, we find that training BNNs with adversarial data augmentation is difficult. However, increasing the capacity of the BNN (with more neurons and weights) minimizes the adversarial training difficulty. For sufficient model capacity, adversarially trained BNNs yield higher adversarial robustness than their full-precision counterpart.
- *Discretization of both input & parameter space:* We demonstrate that combining input discretization with binarized weight /activation training greatly improves a model's robustness. In fact, training a BNN with input discretization (say, 2-bit input) yields similar or better

adversarial accuracy as that of an adversarially trained full-precision model. Thus, the combined discretization scheme can be seen as an efficient alternative to achieving adversarial robustness without the expensive data augmentation procedure (note, only for single-step attacks).

The rest of the paper is organized as follows: Section II discusses the related work as well as delineates our contributions with respect to prior efforts. Section III describes the fundamentals on adversarial attacks, wherein, we describe the different kinds of attacks, how to generate them and explain the adversarial training method which is the state-of-the-art defense approach. In Section IV, we discuss the experimental methodology with each sub-section detailing the advantage of individual discretization themes. In the final sub-section in Section IV, we present a comparative analysis between our results and prior works.

II. RELATED WORK

Based on the intuition demonstrated in Fig. 2, robustness of DLNs can be attributed to two factors: property of the input and model property. Consequently, there have been many recent proposals [8]–[12] that exploit the input-dependent factor and try to remove adversarial perturbations by applying input preprocessing or transformations. Due to the simplicity of this approach, these methods are attractive for practical implementations as they do not incur large computational overhead (as with adversarial training) and do not interfere with the learning process. Our paper complements the results of the prior works while presenting a novel result on the effectiveness of combined parameter and input discretization for adversarial robustness.

One of the first works on input discretization by Xu *et al.* [11] proposes a *depth-color-squeezing* technique wherein they reduce the degrees of freedom available to an adversary by 'removing' unnecessary features. Our pixel discretization scheme is based on their color depth reduction technique. However, the key idea in [11] is to compare the model's prediction on the original input with its prediction on the squeezed input during testing. Xu et al. train the model with regular inputs and during inference use pixel discretization to detect adversarial inputs. That is, if the original and squeezed inputs produce predictions with large difference (greater than an user-defined threshold), the model deems the input to be adversarial and rejects it. Ultimately, the model outputs prediction for only legitimate or non-adversarial inputs. In contrast, the key novel aspect of our work is to train a model with discretized pixel data such that the model looks at a limited range of input values during training that decreases or constrains its' ability to overly generalize in the high-dimensional subspace. Similarly, the thermometer encoding technique and input transformation technique proposed in [8] and [9] are guided by the same intuition of limiting the range of adversarial perturbations by constraining the input. Guo *et al.* [8] trained the network with images transformed in various ways and observed improved adversarial resistance. However, they measured robustness for controlled gray-box attack settings (where, the model parameters are known to the attacker but the input transformations are unknown). Our paper's results on white-box attack is a stronger notion of robustness as we assume all parameters as well as input discretization known to the attacker. Thus, our results while supporting the claims of [8] are more substantial and generalizable. In Buckman *et al.* [9] propose a thermometer encoding technique to map input pixels to a binary vector in order to make more meaningful change during pixel discretization without losing any information from the original image. While, the authors show good adversarial robustness results on small tasks, such as, MNIST [13], they are shown to achieve poor performance on more complex datasets (like, CIFAR10 [14]) [10].

In order to address the limitation of the prior works solely based on input transformation, we investigate the effect of combining model discretization with input discretization thereby leveraging both criteria that contribute to adversarial dimensions. A recent work [15] demonstrated the effectiveness of BNNs against adversarial attacks and observed a similar difficulty in adversarial training with BNNs. However, they did not consider input space discretization and its impact on robustness. While complementing their results, we show that quantizing the input pixels of a BNN during training greatly improves its robustness, even waiving the need for the expensive and time-consuming adversarial training, for certain perturbation ranges. To summarize, the key contributions and novelty that discerns this paper from above mentioned works are:

- In contrast to previous works that use input transformation during test phase only, our analysis shows that DLN models can resist single-step attacks across different black-box, white-box and gray-box scenarios if the quantization measure is introduced during training.
- We show that combined input and parameter discretization during training is a major enabler towards imparting adversarial robustness. To the best of our knowledge, this work is the first to formally evaluate and analyze the impact of input and parameter space discretization for DLNs (across simple and complex datasets including CIFAR10, CIFAR100, ImageNet (or ILSVRC2012) dataset) on robustness.
- We provide an extensive comparison of our approach (combining input and parameter discretization) against prior works using only input discretization [11] or only parameter discretization [15] for multi-step iterative attack scenarios. The results (discussed in Section IV.F) establish combined scheme as a stronger defense in imparting intrinsic robustness to DLNs. However, such combined schemes while doing better than prior works [11], [15] still remain vulnerable in iterative attack scenarios. This is a significant aspect of our work as it exposes the vulnerability/limitation of discretization techniques.

III. BACKGROUND ON ADVERSARIAL ATTACKS

Generating Adversaries: Adversarial examples are created using a trained DLN's parameters and gradients. As shown in Fig. 1, the adversarial perturbation, Δ, is not just some random noise, but carefully designed to bias the network's prediction on a given input towards a wrong class. Goodfellow *et al.* [3] proposed a simple method called Fast Gradient Sign Method (FGSM) to craft adversarial examples by linearizing a trained model's loss function ($\mathcal{L}$, say cross-entropy) with respect to the input (X):

$$X_{adv} = X + \epsilon \times sign(\nabla_X \mathcal{L}(\theta, X, y_{true})) \quad (1)$$

Here, y_{true} is the true class label for the input X, θ denotes the model parameters (weights, biases etc.) and ϵ quantifies the magnitude of distortion. The net perturbation added to the input ($\Delta = \epsilon \times sign(\nabla_X \mathcal{L}(\theta, X, y_{true}))$) is, thus, regulated by ϵ. Distorting the input image in the direction of steepest gradient has the maximal effect on the loss function during prediction. Intuitively, referring to Fig.2, this distortion shifts the data point from the trained region or hyper-volume to an arbitrary region thereby fooling the model.

Types of Attacks: There are two kinds of attacks: Black-Box (BB), White-Box (WB) that are used to study adversarial robustness [16]. WB adversaries are created using the *target* model's parameters, that is, the attacker has full knowledge of a target model's training information. BB attacks refer to the case when the attacker has no knowledge about the target model's parameters. In this case, adversaries are created using a different *source* model's parameters trained on the same classification task as the target model. Since BB attacks are transferred onto the target model, they are weaker than WB attacks. Security against WB attacks is a stronger notion and robustness against WB attacks guarantees robustness against BB for similar perturbation (ϵ) range.

Adversarial Training: Adversarial training simply injects adversarial examples into the training dataset of a model [5]. For each training sample in the dataset, an adversary is created using FGSM [3]. There are several forms of adversarial training. For instance, instead of using the same ϵ for all training examples, [16] and [17] propose to sample a unique ϵ (from a random normal distribution) for each training example. This increases the variation in the adversaries created, thereby, increasing the robustness of a network to larger range of ϵ values. Madry *et al.* [18] use WB adversaries created, using a multi-step variant of FGSM to guarantee a strong defense against both BB and WB attacks. Note, the common theme across all adversarial training methods is data augmentation.

In this work, in most experiments (Section IV.A - IV.D), we focus on adversarial attacks created using FGSM and evaluate the robustness of models against WB adversaries. We evaluate a model's robustness and report adversarial accuracy on the adversarial dataset created using the test data for a given task. We show additional attack results considering BB/gray-box attack with adversaries created using FGSM (in Section IV.E). We also compare our analysis on input and parameter discretized network's robustness with prior works [10], [11], [15] utilizing either input or parameter discretization on multi-step projected gradient descent [18] based iterative WB attack scenarios in Section IV.F.

IV. EXPERIMENTS

We conduct a series of experiments for each discretization theme, primarily, using MNIST [13] (Fully Connected Network, FCN) and CIFAR10 [14] (AlexNet [19] architecture), detailing the advantages and limitation of each approach. We compare the adversarial robustness of each discretization approach with its' full-precision counterpart (with and without adversarial training), using ϵ values reported in recent works [16], [18]. For adversarial training, we employ Random-step FGSM (R-FGSM) proposed in [16] to create a variety of training set adversaries. R-FGSM perturbs the input X with a small random step (sampled from a normal distribution $\mathcal{N}$) before adding the loss gradient to the input: $X_{adv} = X' + \Delta$, where $X' = X + \alpha sign(\mathcal{N}(0, I), \alpha = \epsilon/2$. We use WB adversarial training to confer strong robustness toward all forms of attacks. Note, for MNIST (CIFAR), we use $\epsilon = 0.3$ (8/255) during adversarial training. For evaluating the robustness of parameter space discretization, we use BNNs [6], [19] to evaluate CIFAR10 and MNIST datasets. We also evaluate the robustness of discretization methods on large-scale datasets, CIFAR100 (ResNet20 architecture [20]) and ImageNet [21] (AlexNet architecture) using XNOR networks [7], [22]. Please note, for MNIST we use two different FCN architectures: FCN1-4 hidden layer network with 6144 neurons each ($784 - 6144(\times 4) - 10$), FCN2-4 hidden layer network with 600 neurons each ($784 - 600(\times 4) - 10$). We imported github models [19], [22] for implementing our experiments in PyTorch. We used the same hyperparameters (such as weight decay value, learning rate etc.) as used in [19] and [22] to train our models.

In all experiments, we impose the discretization constraints (input discretization or parameter discretization or both) during training as well as testing. To elucidate this, we describe the training and test methodology for each of the discretization themes below:

- *Input Discretization Training/Testing Method*: Here, we quantize the input both during training and testing. Say, for 2-bit input discretization (all pixel intensities clipped to 4 values {0, 96, 160, 224}), we train a model with 2-bit discretized inputs. During testing, both adversarial and clean test inputs are quantized into 2-bit discretized forms and fed into the model to monitor the prediction accuracy. Note, for input discretization experiments in Section IV.A, we use full-precision models (that is, parameters and activations of the DLN model have 32-bit floating point precision).
- *Parameter Discretization Training/Testing Method*: Here, we quantize the parameter/activation values of a DLN both during training and testing. Say, for BNN, we train a model with extremely low precision weights and activation values clamped to $\{+1, -1\}$. During

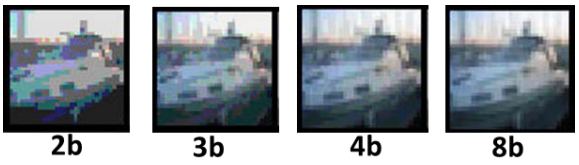


FIGURE 3. *Sample images from CIFAR10 dataset for varying levels of input pixel discretization:* $2-bit, 3-bit, 4-bit, 8-bit$.

testing, both adversarial and clean test inputs are fed into the binarized model to monitor the prediction accuracy. Note, for parameter discretization experiments in Section IV.B, we use full-precision 8-bit inputs (that is, input pixel intensities range from [0, 255]).

- *Input and Parameter Discretization Training/Testing Method*: Here, we quantize the input as well as parameter/activation values of a DLN both during training and testing. Say, for 2-bit input discretization BNN, we train a model with $\{+1, -1\}$ weight/activation values while showing 2-bit quantized inputs. During testing, both adversarial and clean test inputs are again quantized to 2-bit range and fed into the BNN to monitor the prediction accuracy. Note, for input and parameter discretization experiments in Section IV.C, we use BNN or XNOR models with 2-bit/4-bit/8-bit input precision.

It is worth mentioning that our paper is the only other work besides [15] demonstrating the effectiveness of discretized/binarized parameter space on adversarial attacks. While [15] conducted experiments with various forms of attacks (primarily, on MNIST), we restrict ourselves to the WB attack scenario and extrapolate our analysis on larger datasets (CIFAR10, CIFAR100, ImageNet).

A. DISCRETIZATION OF INPUT SPACE

With input space discretization, we convert raw integer pixel intensities ($0 \leq I \leq 255$ or $I \in \mathbb{R}^8$) that are typically 8-bit (or $8b$) values to a low precision range of δ bits ($I_\delta \in \mathbb{R}^{2^\delta}$) as:

$$I_\delta = \lfloor \frac{I}{(256/2^\delta)} \rfloor (\frac{256}{2^\delta}) + \frac{1}{2}(\frac{256}{2^\delta}) \tag{2}$$

where $\lfloor\rfloor$ denotes integer division. Such quantization reduces the number of data points (given a grayscale input image of size $N \times N$) in the manifold from 2^{8N^2} to $2^{\delta N^2}$. This can be broadly interpreted as constraining the data space (that can also be viewed as limiting the range of values across different input dimensions). In accordance with our intuition in Fig. 2, constrained data space can possibly limit the creation of adversaries as well. Fig. 3 illustrates sample CIFAR10 images discretized to varying δ values. The corresponding accuracy (trained on AlexNet for 20 epochs) is shown in Table 1. There is a natural tradeoff between input discretization and overall accuracy of a network. Yet, the test accuracy loss from the full precision $8b$ to $3b$ is $\sim 2.2\%$. This verifies the presence of unnecessary input features that do not substantially contribute to the classification task or accuracy. This result has also been noted by Xu *et al.* [11]. $2b$ discretization decreases the accuracy by a larger margin ($\sim 6\%$). Note, this accuracy loss can be minimized by training the $2b$ inputs for more epochs. However, for iso-comparison, we fix the number of epochs across all experiments for a given dataset.

TABLE 1. CIFAR10 accuracy.

Input-bit	Accuracy
2b	82
3b	86.64
4b	87.1
8b	88.9

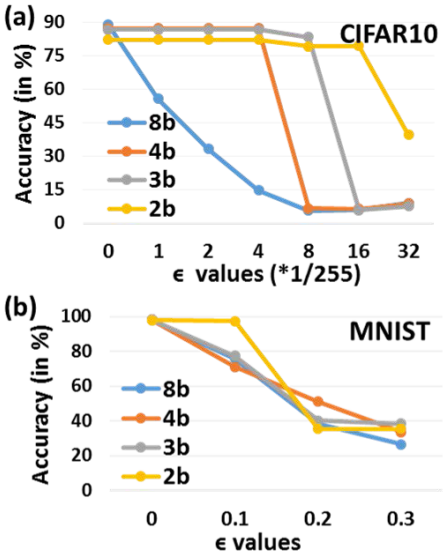


FIGURE 4. *Adversarial accuracy on test data for varying perturbation values on (a) CIFAR10 (b) MNIST for different input discretization.*

A remarkable outcome of this discretization method is the substantial improvement in a model's adversarial accuracy. Fig. 4 illustrates the evolution of adversarial accuracy of the CIFAR10 models (from Table 1) with increasing level of perturbation, ϵ. Here, the discretized adversarial inputs are created using FGSM on discretized clean inputs as follows:

$$X_{adv} = X_\delta + \epsilon \times sign(\nabla_{X_\delta} \mathcal{L}(\theta, X_\delta, y_{true})) \tag{3}$$

$$X_{adv} \rightarrow X_{adv_\delta} \tag{4}$$

For a given discretization constraint δ (say, $2b$), $2b$ input (X_δ) is passed to a model. Then, the corresponding loss $\mathcal{L}$ is linearized and multiplied with perturbation value ϵ which is then added to the $2b$ input to create the adversary X_{adv}. We quantize X_{adv} to create the $2b$ discretized adversary X_{adv_δ}. It is evident that quantization after creation of X_{adv} reduces the effect of perturbation ϵ. Note, however, the same approach is also used in prior works [8], [11] to create adversaries with quantized inputs. The claim is that quantization can be perceived as an input preprocessing step which, serves as a deterrent to the attacker by reducing the intensity of perturbation and, hence, improve adversarial robustness.

In Fig. 4, $\epsilon = 0$ corresponds to clean test accuracy. It is clear that clamping the input precision to lower values

increases the resistance of the model to larger magnitude of distortion. We speculate that constraining the input precision or allowed range of input values reduces the overall hyper-volume space thereby leaving 'less' space for shifting or adversarially perturbing a data point (referring to Fig.2(b) intuition). 8*b* input model shows a decline in accuracy even for a small value of $\epsilon = 1/255$. Thus, we can deduce that higher input precision (which implies larger range of input values for a data point) allows even small perturbations to shift the data point (as seen earlier in Fig. 2(a)). In contrast, increasing discretization increases the minimum ϵ that affects a model's accuracy catastrophically. What is surprising is that for 2*b* input, a model's adversarial accuracy ($\sim 79\%$) for large $\epsilon = (8, 16)/255$ is almost similar to that of clean accuracy ($\sim 82\%$). For larger $\epsilon = 32/255$, the accuracy of all models declines to $< 10\%$, except 2*b*. This is a very interesting result since we have not employed any adversarial training, and still achieve substantial adversarial resistance for a large ϵ range.

Fig. 4 (b) shows the adversarial accuracy results for MNIST (trained on FCN2 for 10 epochs). We observe a similar trend of increasing adversarial resistance with increasing discretization for larger ϵ. Since MNIST is a simple dataset with predominantly black-background, input discretization (8*b*, 4*b*, 3*b*) does not contribute much to adversarial resistance until we go to extremely low 2*b* precision. In fact, 2*b* discretization yields adversarial accuracy similar to the clean test accuracy (for $\epsilon = 0.1$) exhibiting the effectiveness of this technique even for simple datasets. Note, in all experiments, we imposed input discretization constraints both during training and testing as mentioned earlier.

Next, we trained the 2*b* input discretized CIFAR10 and MNIST models with adversarial training to observe the improvement in adversarial accuracy compared to 8*b* input adversarial training (Table 2). Note, the adversaries were discretized to 2*b* precision (using Eqn. 3, 4) to adversarially train the 2*b*-input CIFAR10, MNIST models. Compared to the results in Fig. 4 (a, b), adversarial training substantially improves the robustness of a model with full 8*b* input for larger ϵ values. Input discretization greatly furthers this robustness with $> 10\%$ accuracy gain across different perturbation ranges in both MNIST and CIFAR10. It is worth mentioning that the CIFAR10 accuracy (79%) without adversarial training for $\epsilon = (8, 16)/255$ for 2*b* input is as good as the accuracy (83%) with adversarial training. This shows that input discretization is a good regularization scheme that improves the generalization capability of a network on adversarial data. Note, for $\epsilon = 32/255$ in case of CIFAR10, the accuracy is similar for 8*b*, 2*b* since the adversarial training was conducted with adversaries created using $\epsilon = 8/255$. Including larger perturbation adversaries during adversarial training will yield improved accuracy gain.

B. DISCRETIZATION OF PARAMETER SPACE

Since input discretization gave us such promising results, we were naturally inclined toward analyzing a binarized neural network's (BNN) behavior against adversarial attacks. Here, the weights and activations (or parameters) are discretized to extremely low precision values $\{+1, -1\}$ [6]. The discretization constraints are imposed on a BNN during training, wherein, the parameters are clamped to $\{+1, -1\}$ after every backpropagation step. One can view this discretization as an implicit form of regularization. In fact, it is this extreme form of regularization that makes a BNN difficult to train (clean test accuracy observed with BNNs is, typically, lower than full-precision networks). As suggested in [15], the difficulty in training a BNN translates to difficulty in attacking the BNN as well. Referring to the data manifold intuition (Fig. 2), we can deduce that constraining the parameter space during a model's training will introduce discontinuities and non-smoothness in its decision boundary. Since adversaries are created using gradients of a model (that is a property of the model's decision boundary), generating gradients (and hence adversaries) for non-smooth functions will be difficult. This in turn will make a BNN less susceptible to adversaries. Note, the input image to a BNN is full 8*b* precision.

Fig.5 (a) compares the adversarial accuracy obtained for varying ϵ values for CIFAR10 BNN (AlexNet architecture) against a similar architecture full-precision network (with 32*b* precision for weights and activations). We trained the networks for 40 epochs since BNNs require more training iterations to attain comparable accuracy as that of a

TABLE 2. Accuracy with adversarial training for varying ϵ. Text in red are the ϵ values for MNIST and corresponding accuracy.

Data	Model	Clean	ϵ:8/255 0.1	ϵ:16/255 0.2	ϵ:32/255 0.3
CIFAR10	2b	83.1	82.7	82.7	43.9
	8b	84.3	62.2	53.6	45.5
MNIST	2b	98.5	98.5	84.7	85.4
	8b	98	84.8	74.5	65.9

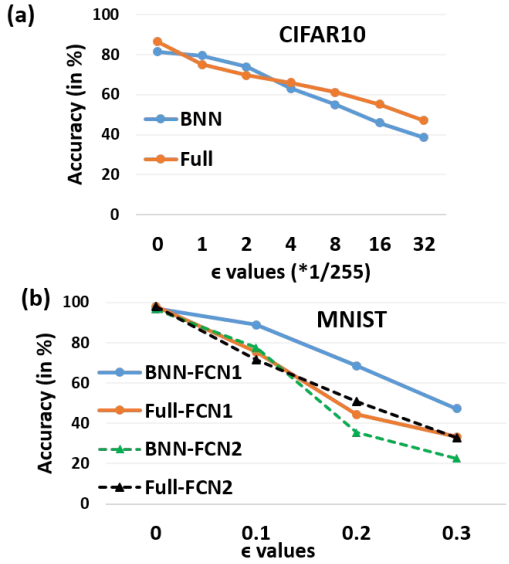


FIGURE 5. *Adversarial accuracy on test data for varying perturbation values on (a) CIFAR10 (b) MNIST for binarized and full-precision (32b weights) models.*

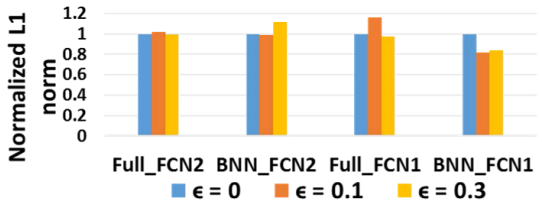


FIGURE 6. *Normalized L1 norm of first hidden layer activations in response to clean ($\epsilon = 0$) and adversarial inputs ($\epsilon = 0.1, 0.3$) for binarized and full-precision MNIST model of different architecture: FCN1, FCN2.*

full-precision network. Here, we do not incorporate input discretization in our analyses. All networks are fed full-precision $8b$ inputs (both during training and testing). In Fig. 5(a), for $\epsilon \leq 2/255$, BNN shows better adversarial resistance (i.e. adversarial accuracy is closer to clean accuracy). However, the BNN's accuracy declines steeply as we move toward larger perturbation ranges. We note a similar trend for MNIST (trained for 10 epochs on FCN2 architecture), wherein the full-precision network yields improved robustness than the BNN for $\epsilon \geq 0.2$. These results contradict our intuition that increased discretization of BNNs should result in lesser adversarial susceptibility.

To understand this, we calculated the L1 norm of the first hidden layer activation of the FCN2 network in response to clean input images. We found that BNNs generally have a larger variance and range of values than full-precision network. Since BNN uses weight values $(+/-1)$ which are typically of greater magnitude than the small weight values of a full precision network, we observe a larger range in the former case. Interestingly, we find that the L1 norm of the BNN activations (in response to adversarial images perturbed with lower ϵ values) approximately lie within the same range as that of the clean input case. In contrast, L1 norm for higher ϵ adversaries have a much higher range. For a full-precision network, the L1 norm range of the different ϵ adversaries and clean data typically intersect with each other owing to the lower weight values (Fig. 6). We believe that the extreme quantization of weight values in BNNs to higher magnitudes causes adversarial susceptibility for larger range perturbations. While the L1 norm analysis is not very substantial from a mathematical standpoint, it hinted us to increase the capacity (more neurons and weights) of the network. The motif here is that increasing the capacity would increase the overall range of activation values that might incorporate larger range perturbations. Exploding the network capacity for MNIST (FCN1 architecture) yielded a sizable improvement in adversarial resistance with BNN as compared to its corresponding full-precision counterpart (Fig. 5(b)). This is a crucial detail of our analysis that: *while BNNs are intrinsically robust to adversaries (for small ϵ), only models with sufficient capacity can withstand against large ϵ values*.

Even with adversarial training, we observed the same trend that binarized networks of insufficient capacity do not yield as good adversarial robustness as that of a full-precision

TABLE 3. **Accuracy with adversarial training for varying ϵ. Text in red are the ϵ values for MNIST and corresponding accuracy.**

Data	Model	Clean	ϵ:8/255 0.1	ϵ:16/255 0.2	ϵ:32/255 0.3
CIFAR10	BNN	79.7	53.1	43.6	35.3
	Full	82.7	72.2	63.6	55.5
MNIST (FCN1)	BNN	96.9	89.1	74.5	65.8
	Full	98	84.8	71	61.7

network (Table 3). For CIFAR10, full-precision network is the clear winner. While for MNIST (with excessive capacity FCN1 architecture), BNN yields improved robustness. A noteworthy observation here is that adversarial training substantially improves the robustness of a full-precision network (see CIFAR10 results in Fig. 5(a), Table 3), while BNNs do not benefit much from them. In fact, we find that BNNs are difficult to train with adversarial training. The learning rate/other hyperparameters need to be tuned carefully to ensure that the BNN model converges to lower error values during adversarial training. Reference [15] also observed a similar trend and explained that binarized weights are not as *'malleable'* as full-precision weights and hence cannot easily adjust to all possible variations of adversarial data augmented to the training dataset. We think that increasing the capacity of the network compensates for the *'non-malleability'* of the constrained parameters to certain extent. As a result, we see improved accuracy for MNIST in Table 3 with FCN1 architecture.

C. DISCRETIZATION OF INPUT AND PARAMETER SPACE

Next, we combined both discretization strategies and analyzed the adversarial robustness of BNNs with varying image-level discretization. We compare the adversarial accuracy of BNNs to that of a full-precision network for iso-input discretization scenarios, as shown in Table 4 for CIFAR10 (AlexNet architecture trained for 40 epochs). In Table 4, *BNN-2b* (*Full-2b*) refers to a binarized (full-precision) model with $2b$ input precision. Note, we use Eqn. 3, 4 to create discretized adversarial inputs for a given input precision correspnding to each experiments. That is, the adversarial accuracy for $BNN - 2b$ or $Full - 2b$ in Table 4 corresponds to testing with $2b$ discretized adversarial data. Full precision models have $32b$ precision weights and activations. While input discretization for a full-precision network suffers a sizeable accuracy loss, BNN's accuracy fluctuation is marginal with a maximum of 1% change. This is expected since BNNs (owing to $+/-1$ binarized parameters) do not have as many dimensions (as a full-precision network with $32b$ weights and activations) to fit the extra information in the $8b$ input data. Thus, BNNs fit $2b$, $8b$ data likewise yielding similar generalization error. As opposed to the results seen earlier with $8b$ inputs, BNNs with lower input precision $(2b, 4b)$ have significantly higher adversarial resistance than their full-precision counterparts even for large ϵ values. *Model capacity does not restrict the adversarial resistance in this case. This is an artefact of the two-step quantization that increases*

TABLE 4. Adversarial accuracy with CIFAR10 for varying ϵ with different combinations of input and parameter discretization.

Model	Clean	ϵ:8/255	ϵ:16/255	ϵ:32/255
BNN-2b	81	80	80	36.7
Full-2b	82	79.1	79.3	39.6
BNN-4b	81.9	58.3	52.3	36.8
Full-4b	81.1	53.8	45.1	37.3
BNN-8b	81.5	55.1	45.9	38.6
Full-8b	86.5	61.1	55.2	47.1

TABLE 5. Adversarial accuracy with MNIST for varying ϵ with different combinations of input and parameter discretization.

Model	Clean	ϵ:0.1	ϵ:0.2	ϵ:0.3
BNN-2b	96.4	96.4	60.7	62.3
Full-2b	97.8	97.4	35.4	35.3
BNN-4b	96.4	88.9	76.7	58.7
Full-4b	98.1	71.1	50.9	33.7
BNN-8b	97.1	89.4	56.1	33.6
Full-8b	98.2	75.9	38.5	26.4

the minimum allowable perturbation to shift a data point. We can also draw an alternate insight from this result: The constrained parameter space of BNNs restricts their overall exploration of the data manifold during training. Referring to Fig. 2 (b), this increases the probability of untrained or arbitrary hyper-volumes (for BNNs) thereby increasing their adversarial susceptibility. Increasing the capacity enables a BNN to explore the manifold better during training. By discretizing the input, we are restricting the overall data manifold space. This allows a model, even, with lower capacity to explore the manifold well thereby decreasing the extent of arbitrary hyper-volumes. Table 5 illustrates the accuracy results for MNIST (FCN1 architecture trained for 20 epochs).

We conducted adversarial training with $2b$ input discretized BNNs to find out if it helps build adversarial robustness. Note, adversarial training was performed with $2b$ precision adversaries (created using Eqn. 3, 4). The results are shown in Table 6. Comparing to the $8b$ input BNN adversarial training results in Table 3, we observe a substantial gain in adversarial accuracy. However, contrasting the BNN results against Table 2 ($2b$ input full-precision networks), we observe similar performance gains. In fact, the accuracy gains for $2b$ input CIFAR10 BNN with and without adversarial training (Table4/Table 6) are nearly the same. Earlier, we saw that the accuracy (for low ϵ values) of a full-precision network working on $2b$ input data without adversarial training is similar to that of an adversarially trained network on $8b$ inputs (Table 2, Fig. 4). Combining the adversarial training results till now, we can deduce the following: *1) For low input-precision ($2b$) regime, adversarial training does not compound the adversarial resistance of a network (irrespective of binarized or full-precision parameters), for lower ϵ values. Adversarial training helps when the input has higher ($8b$) precision. 2) Input discretization, in general, offers very strong adversarial defense for lower ϵ values. Discretizing the input as well as the parameter space furthers adversarial robustness. Adversarial training in a discretized input and parameter space does not benefit much and hence can be waived only for single-step FGSM attacks.* However, in case

TABLE 6. Accuracy with adversarial training for varying ϵ. Text in red are the ϵ values for MNIST and corresponding accuracy.

Data	Model	Clean	ϵ:8/255 0.1	ϵ:16/255 0.2	ϵ:32/255 0.3
CIFAR10	BNN-2b	78.4	78.1	78.1	30.5
MNIST (FCN1)	BNN-2b	95.7	95.7	89.3	88.6

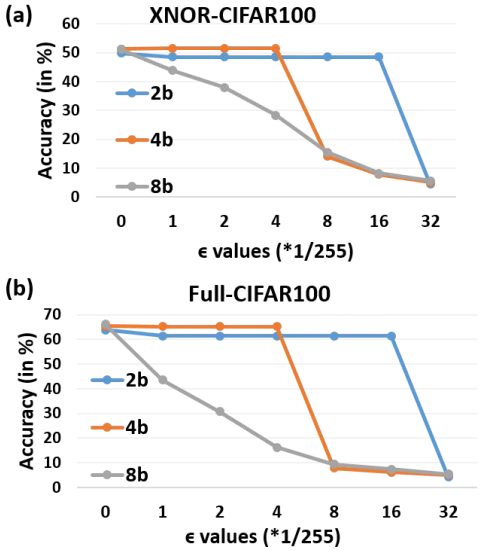


FIGURE 7. *Adversarial accuracy on test data for increasing ϵ values on binarized and full-precision (32b weights) models trained on CIFAR100 with different input discretization:8b, 4b, 2b.*

of stronger multi-step attack scenarios and to gain robustness against larger perturbations (such as $\epsilon = 32/255, 0.3$ in CIFAR10, MNIST), the network needs to be adversarial trained with corresponding large ϵ values.

D. ANALYSIS ON CIFAR100 AND IMAGENET

Scaling up the discretization analysis to larger datasets yielded similar results as observed with CIFAR10, MNIST. Fig. 7 demonstrates the adversarial accuracy evolution for CIFAR100 (trained on ResNet20 architecture for 164 epochs) for binarized XNOR ($1b$ weights and activations) networks and corresponding full-precision ($32b$ weights and activations) models. Note, XNOR networks are similar to BNNs (1-bit weights/activations) with additional scaling factors to achieve higher accuracy on complex datasets. It is evident that input discretization is the most beneficial to obtain adversarial robustness. $2b$ input discretized models in both cases yield adversarial accuracy close to the clean accuracy ($\epsilon = 0$) for a large range of perturbations. The accuracy loss between clean and $\epsilon = 16/255$ adversary for $2b$-input XNOR (1.6%) is slightly better than the $2b$-input full-precision model (2.4%). This can be attributed to the intrinsic robustness offered by discretizing the parameter space of XNOR networks. Furthermore, the fact that $8b$-input XNOR yields higher adversarial accuracy for iso-perturbation values than $8b$-input full-precision model further demonstrates the ability of binarized networks to counter adversarial attacks. A noteworthy

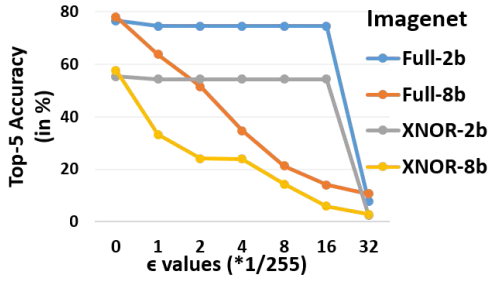


FIGURE 8. *Top-5 Adversarial accuracy on test data for increasing ϵ values on binarized and full-precision (32b weights) models trained on ImageNet with different input discretization:8b, 2b.*

observation here is that the loss in clean accuracy between $2b$ and $8b$ input discretized full-precision network is small ($\sim$ 2%) as compared to the large 6% loss observed earlier with CIFAR10 (Table 1). As we scale up the complexity of the dataset (and the complexity of the DLN architecture), the features in the input/parameter space increase (that can be viewed as increase in the range of input/parameter values). Discretizing the input/parameters for a complex dataset on a larger DLN architecture (such as ResNet20) possibly eliminates certain range of input/parameter values that do not necessarily contribute to the accuracy. In contrast, smaller datasets or smaller DLN architectures have lesser range of input/parameter values and are thus at a risk of suffering a large accuracy drop with discretization.

Fig. 8 shows the accuracy results for ImageNet (trained on AlexNet). We only show the top-5 adversarial accuracy. We see similar trends as CIFAR100. Note, the XNOR models are trained for 50 epochs, while full-precision models are trained for 90 epochs. As a result, we see lower baseline accuracy ($\epsilon = 0$) in the former case. Like CIFAR100, the loss in clean test accuracy between $2b$ and $8b$ input discretization is minimal for each model. Also, the accuracy difference between clean and $\epsilon = 16/255$ adversarial data for $2b - XNOR$ (0.9%) is much lower than $2b - full$ precision models (2.8%). This highlights the intrinsic robustness capability of binarized networks even for large-scale datasets.

E. OTHER ATTACK SCENARIOS

Till now, we have focused on WB attack scenarios. For the sake of completeness, we analyzed the adversarial accuracy in gray-box attack scenarios for CIFAR100 dataset with XNOR/ full-precision models with $2b, 8b$ input discretization (Table 7). Here, the CIFAR100 attack model for an XNOR (or full-precision parameter) target is a separately trained XNOR (or full-precision parameter) model, respectively. Note, the attack model is a ResNet20 architecture with $2b, 8b$ input precision depending upon the input precision of the target. Essentially, we attack an XNOR-2b (or Full-2b) model with another XNOR-2b (Full-2b) model trained separately. The adversaries are created using FGSM. In this regime, even though the attack model is different from the target, the nature of input discretization is known to the attacker

TABLE 7. Gray-Box Adversarial accuracy with CIFAR100 for varying ϵ with different combinations of input and parameter discretization.

Model	Clean	ϵ:4/255	ϵ:8/255	ϵ:16/255	ϵ:32/255
XNOR-2b	49.9	44.6	39.6	30.7	18.7
Full-2b	63.9	59.4	54.2	43.8	24.5
XNOR-8b	51.1	42.9	35.6	25.4	16.3
Full-8b	66.2	56.9	48.3	34.1	19.63

TABLE 8. Black-Box Adversarial accuracy with CIFAR10 for varying ϵ with different combinations of input and parameter discretization.

Model	Clean	ϵ:4/255	ϵ:8/255	ϵ:16/255	ϵ:32/255
BNN-2b	81	79.8	79.8	79.8	56.1
Full-2b	82	81.8	81.8	81.7	54.1
BNN-4b	81.9	80.74	72.5	66.9	59.73
Full-4b	81.1	80.75	71.7	65.4	57.4

(hence, gray-box attack scenario). In Table 7, higher accuracy ($\sim>$ 25%) observed for large range of ϵ values in the $8b$ input regime establishes that gray-box attacks are weaker than WB. A noteworthy observation here is that gray-box attacks on $2b$ input discretized models (both full-precision and XNOR) seem to be stronger for $\epsilon = 4, 8/255$ than WB attacks (refer to Fig. 7). However, WB attacks (Fig. 7) are extensively more devastating in the higher $\epsilon > 8/255$ regime than gray-box. We conjecture that transferred attacks have more variability in the perturbations generated with lower ϵ FGSM. As a result, a model (XNOR or full-precision) may not be able to counter such variability. While this variability remains in the higher ϵ range, the gray-box nature of the attack decreases the overall strength. Further investigation is required to analyze the nature of transferred attacks in a low ϵ regime on input-/parameter-discretized networks. This observation is a novel result as prior works on input transformation schemes (note, no parameter discretization has been explored in earlier works) have mostly focused on white-box or gray-box attacks for $\epsilon > 8/255$ regime and have thus never observed this phenomenon of higher vulnerability in gray-box than WB regime for low ϵ perturbations.

To show the effectiveness of discretization against black-box attacks created with FGSM, we analysed the CIFAR10 dataset with BNN/ full-precision models with $2b, 4b$ input discretization. Here, the CIFAR10 attack model is a separately trained full-precision parameter model with $8b$ input discretization. It is evident from this attack/target model experimental setting that both input and parameter discretization regimes of the target model are unknown to the attacker (hence, BB attack scenario). Table 8 illustrates the results. Here, we observe higher adversarial accuracy (nearly equal to the clean accuracy) for a larger range of perturbations $\epsilon = 4, 8, 16/255$. For $\epsilon = 32/255$, the adversarial accuracy drops significantly. The higher range of accuracy in this case proves that BB attacks are the weakest among gray-box and WB attacks.

F. COMPARISON WITH PRIOR WORKS

A significant contribution of our analysis is that we include input or parameter discretization during training and highlight the relevance of discretized training for adversarial

TABLE 9. WB adversarial accuracy subject to Carlini-Wagner L2 attacks of increasing strength on MNIST dataset with different combinations of input and parameter discretization. The adversarial accuracy shown is for different iterations of CWL2 attacks: 10, 40, 100.

Network	Model	Clean	CWL2 Attack Iterations: 10/ 40/ 100
FCN1	BNN-8b (Galloway [15]) BNN-2b (Ours)	99 96.4	98/38/0.1 96/57/10

TABLE 10. WB adversarial accuracy subject to PGD attacks of $\epsilon = 0.3$ over 100 steps on MNIST dataset with different combinations of input and parameter discretization.

Network	Model	Clean	PGD Attack $\epsilon = 0.3$
4-layer CNN	Full-2b (Xu [11]) BNN-2b (Ours)	98.8 96.4	75.3 82.7

TABLE 11. WB adversarial accuracy subject to PGD attacks of $\epsilon = 8/255$ over 40 steps on CIFAR10 dataset with different combinations of input and parameter discretization.

Network	Model	Clean	PGD Attack $\epsilon = 8/255$
ResNet	Full-3b (Xu [11]) BNN-3b (Ours)	76.1 78.4	32.5 42

robustnees. To further elucidate the effectiveness of discretization during training phase, we compare our results with that of prior works of Xu *et al.* [11] and Galloway *et al.* [15]. Xu *et al.* propose to use color depth bit reduction along with 2×2 median filter smoothing (referred to as *feature squeezing* in their paper) to detect adversarial images. However, Xu *et al.* train a model with full-precision inputs and use feature squeezing during the test phase to identify clean vs. adversarial input. We replicated their experiments by utilizing the open source framework and code made available by the authors [23]. Galloway *et al.* simply analysed the effect of attacking binarized neural networks (without input discretization). We utilized the open source framework [24], to investigate how input-and-parameter discretized models perform in attack scenarios (crafted with the widely used CleverHans library to benchmark adversarial vulnerability of neural networks [25]) as compared to only parameter discretized models.

Table 9 - 12 summarizes our comparative analysis for MNIST, CIFAR10 and ImageNet. Here, we conducted multi-step white box iterative attacks with iso-attack, perturbation parameters and model architectures to have a fair comparison with prior works. The various cases are as follows: a) Carlini-Wagner L2 attack (CWL2 [26]) over 10, 40, 100 attack iterations [15] on FCN1 architecture for MNIST (Table 9), b) Projected Gradient Descent (PGD [18]) attack over 100 attack iterations with $\epsilon = 0.3$ for MNIST on a 4-layer convolutional architecture as [10], [11] (Table 10), c) PGD attack over 40 attack iterations with $\epsilon = 8/255$ for CIFAR10 on a ResNet architecture as [10], [11] (Table 11), d) PGD attack over 40 attack iterations with $\epsilon = 4/255$ for ImageNet on a InceptionResNetV2 architecture as [10], [11] (Table 12).

In each case, we compare our combined input/parameter discretized network proposal (either XNOR or BNN with $2b, 3b$ input precision: $BNN-2b$, $BNN-3b$, $XNOR-3b$) that

TABLE 12. WB adversarial accuracy subject to PGD attacks of $\epsilon = 4/255$ over 40 steps on ImageNet dataset with different combinations of input and parameter discretization.

Network	Model	Clean	PGD Attack $\epsilon = 4/255$
InceptionResNetV2	Full-3b (Xu [11]) XNOR-3b (Ours)	65.6 59	10.1 18.8

are trained with quantized input and weight values to prior works: a) Xu et al. where only $2b, 3b$ input discretization is performed during the testing phase on a full-precision network ($Full-3b$, $Full-2b$), b) Galloway *et al.* [15] where only parameter discretization is performed and only MNIST results for iterative WB attacks are available. As mentioned earlier, the inclusion of input/parameter discretization during training serves as a major benefactor in advancing adversarial robustness of models. Consequently, we observe a substantial improvement over Xu *et al.* in Table 10,11, 12. It is worth mentioning that iterative attacks cause drastic drop in accuracy in CIFAR10 and ImageNet datasets than simpler MNIST data. This can be addressed by using adversarial training defense. Thus, we can deduce that while discretization extends the inherent robustness of a model, adversarial training is pertinent for defense against stronger iterative attacks. The results in Table 9 illustrate the effectiveness of input discretization (during training) as an extra layer of intrinsic defense in addition to the binarized parameter or activation model.

V. DISCUSSION & CONCLUSION

Low-precision models or quantization techniques, so far, have been explored to reduce the resource utilization of DLNs for energy-efficient deployment on edge devices. We have demonstrated that *discretization* also warrants *security* against adversarial attacks, thereby, offering a key benefit of *robustness* in hardware implementation for a certain range of attacks. Our work also exposes the vulnerability of quantization or discretization techniques for multi-step iterative attacks necessitating the use of adversarial training with discretization as an improved defense strategy. In summary, the main findings/recommendations from this work are:

- Input discretization is major benefactor for adversarial robustness (with both binarized and full-precision) models. $2b$ input discretized models (without adversarial training) yield similar adversarial accuracy as adversarially trained $8b$ input models for lower ϵ values. Robustness against higher ϵ and multi-step attack requires adversarial training.
- Binarized (low-precision weights/activations) models are intrinsically more (although, slight) robust than full-precision ($32b$ weights/activations) models. Adversarial training needs to be carefully done on sufficient capacity binarized networks to attain similar adversarial robustness as the full-precision models.
- Combining input and parameter discretization is an efficient way of obtaining adversarial robustness to a

moderate range of perturbation values without conducting the iterative adversarial training.

- Performing adversarial training on models with combined input and parameter discretization improves the adversarial accuracy ($\sim>$ 10%) in comparison to adversarially trained full-precision models with 8*b* inputs.

Our work unravels a simple idea that: *hardware optimization related techniques can potentially resolve or resist software vulnerabilities (specifically, adversarial attacks)*. While we focus on discretization, there is a lot of future scope to explore other efficiency-driven techniques (such as, stochasticity, model pruning etc.) to gauge their implication on adversarial robustness. As seen earlier, standard discretization remains vulnerable to stronger multi-step attack scenarios and hence needs to be defended with adversarial training or data augmentation. We believe that stochasticity or pruning techniques tied with discretization can possibly resist better, even against stronger attacks. This requires further investigation. Finally, through this work, our goal is to project the security advantages of discretization techniques that are primarily used to deploy efficient DLN models in general-purpose (GPU) or accelerator (TPU or FPGA) computing platforms, thereby, paving possible research directions of exploring efficiency-robustness tradeoff in artificial intelligence applications.

REFERENCES

[1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, p. 436, 2015.

[2] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," Jul. 2016, *arXiv:1607.02533*. [Online]. Available: https://arxiv.org/abs/1607.02533

[3] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," Dec. 2014, *arXiv:1412.6572*. [Online]. Available: https://arxiv.org/abs/1412.6572

[4] P. Panda and K. Roy, "Implicit generative modeling of random noise during training for adversarial robustness," Jul. 2018, *arXiv:1807.02188*. [Online]. Available: https://arxiv.org/abs/1807.02188

[5] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," Dec. 2013, *arXiv:1312.6199*. [Online]. Available: https://arxiv.org/abs/1312.6199

[6] I. Hubara, M. Courbariaux, D. Soudry, R. El-Yaniv, and Y. Bengio, "Binarized neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 4107–4115.

[7] M. Rastegari, V. Ordonez, J. Redmon, and A. Farhadi, "XNOR-Net: Imagenet classification using binary convolutional neural networks," in *Proc. Eur. Conf. Comput. Vis.* Amsterdam, The Netherlands: Springer, 2016, pp. 525–542.

[8] C. Guo, M. Rana, M. Cisse, and L. van der Maaten, "Countering adversarial images using input transformations," in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–12.

[9] J. Buckman, A. Roy, C. Raffel, and I. Goodfellow, "Thermometer encoding: One hot way to resist adversarial examples," in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–22.

[10] J. Chen, X. Wu, Y. Liang, and S. Jha, "Towards understanding limitations of pixel discretization against adversarial attacks," May 2018, *arXiv:1805.07816*. [Online]. Available: https://arxiv.org/abs/1805.07816

[11] W. Xu, D. Evans, and Y. Qi, "Feature squeezing: Detecting adversarial examples in deep neural networks," Apr. 2017, *arXiv:1704.01155*. [Online]. Available: https://arxiv.org/abs/1704.01155

[12] J. Lin, C. Gan, and S. Han, "Defensive quantization: When efficiency meets robustness," in *Proc. Int. Conf. Learn. Represent.*, 2019, pp. 1–14.

[13] Y. LeCun. (1998). *The MNIST database of Handwritten Digits*. [Online]. Available: http://yann.lecun.com/exdb/mnist/

[14] A. Krizhevsky, V. Nair, and G. Hinton. (2009). *Cifar-10 and Cifar-100 Datasets*. Accessed: Mar. 1, 2016. [Online]. Available: https://www.cs.toronto.edu/kriz/cifar.html

[15] A. Galloway, G. W. Taylor, and M. Moussa, "Attacking binarized neural networks," in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–14.

[16] F. Tramèr, A. Kurakin, N. Papernot, I. Goodfellow, D. Boneh, and P. McDaniel, "Ensemble adversarial training: Attacks and defenses," in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–20.

[17] A. Kurakin, I. J. Goodfellow, and S. Bengio, "Adversarial machine learning at scale," in *Proc. Int. Conf. Learn. Represent.*, 2016, pp. 1–17.

[18] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–27.

[19] [Online]. Available: https://github.com/itayhubara/BinaryNet.pytorch

[20] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 770–778.

[21] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2009, pp. 248–255.

[22] [Online]. Available: https://github.com/jiecaoyu/XNOR-Net-PyTorch

[23] [Online]. Available: https://evadeML.org/zoo

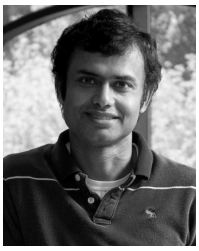
[24] [Online]. Available: https://github.com/AngusG/cleverhans-attacking-bnns

[25] N. Papernot, I. Goodfellow, R. Sheatsley, R. Feinman, and P. McDaniel, "Technical report on the CleverHans v2.1.0 adversarial examples library," Oct. 2016, *arXiv:1610.00768*. [Online]. Available: https://arxiv.org/abs/1610.00768

[26] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2017, pp. 39–57.

PRIYADARSHINI PANDA received the B.E. degree in electrical & electronics engineering and the M.Sc. degree in physics from BITS Pilani, India, in 2013. She was a recipient of Gold Medal in physics for academic excellence for his M.Sc. degree. She is currently pursuing the Ph.D. degree with Purdue University, West Lafayette, IN, USA, under the supervision of Prof. K. Roy. In 2013, she joined Intel, Bengaluru, India, where she worked on RTL design for graphics power management. She was a Research Intern with Intel Labs, OR, USA, in 2017. She will be joining as an Assistant Professor with the Electrical Engineering Department, Yale University, New Haven, CT, USA, from 2019. Her research interests include neuromorphic computing and deep learning, specifically, developing scalable energy-efficient design methodologies for deep learning applications (object recognition), novel supervised/unsupervised learning algorithms for deep spiking/dynamic reservoir networks for spatio-temporal data processing, and adversarial robustness of deep learning and spiking networks.

INDRANIL CHAKRABORTY received the B.E. degree from Jadavpur University, India, in 2013, and the master's degree from the IIT Bombay, Mumbai, in 2016. His master's thesis was on physics-based modelling of PCMO-based devices. He was a recipient of the Best M.Tech. Thesis Award and the Academic Excellence Award during his time at IIT Bombay for his academic performance. He is currently pursuing the Ph.D. degree under the supervision of Prof. K. Roy. His current research interests include in-memory computing platforms based on CMOS, beyond-CMOS technologies and Si Photonics, and technology-aware algorithms under the broad umbrella of neuromorphic computing.



KAUSHIK ROY received the B.Tech. degree in electronics and electrical communications engineering from IIT Kharagpur, Kharagpur, India, and the Ph.D. degree from the Electrical and Computer Engineering Department, University of Illinois at Urbana–Champaign, in 1990. He was with the Semiconductor Process and Design Center, Texas Instruments, Dallas, where he worked on FPGA architecture development and low-power circuit design. He joined the Electrical and Computer Engineering Faculty, Purdue University, West Lafayette, IN, USA, in 1993, where he is currently a Edward G. Tiedemann Jr. Distinguished Professor. He was a Research Visionary Board Member of Motorola Labs, in 2002 and held the M.K. Gandhi Distinguished Visiting Faculty, IIT Bombay. He has published more than 600 papers in refereed journals and conferences, holds 15 patents, supervised 65 Ph.D. dissertations, and has coauthored two books on *Low Power CMOS VLSI Design* (John Wiley & McGraw Hill). His research interests include spintronics, device-circuit co-design for nano-scale Silicon and non-Silicon technologies, low-power electronics for portable computing and wireless communications, and new computing models enabled by emerging technologies.

Dr. Roy received the National Science Foundation Career Development Award, in 1995, the IBM Faculty Partnership Award, the ATT/Lucent Foundation Award, the 2005 SRC Technical Excellence Award, the SRC Inventors Award, the Purdue College of Engineering Research Excellence Award, the Humboldt Research Award, in 2010, the 2010 IEEE Circuits and Systems Society Technical Achievement Award, the Distinguished Alumnus Award from IIT Kharagpur, Kharagpur, a Fulbright-Nehru Distinguished Chair, DoD National Security Science and Engineering Faculty Fellow, from 2014 to 2019, the Semiconductor Research Corporation Aristotle Award, in 2015, and the Best Paper Awards at 1997 International Test Conference, the IEEE 2000 International Symposium on Quality of IC Design, the 2003 IEEE Latin American Test Workshop, the 2003 IEEE Nano, the 2004 IEEE International Conference on Computer Design, the 2006 IEEE/ACM International Symposium on Low Power Electronics & Design, and the 2005 IEEE Circuits and system society Outstanding Young Author Award (Chris Kim), the 2006 IEEE TRANSACTIONS ON VLSI SYSTEMS Best Paper Award, the 2012 ACM/IEEE International Symposium on Low Power Electronics and Design Best Paper Award, and the 2013 IEEE TRANSACTIONS ON VLSI Best Paper Award. He was a Purdue University Faculty Scholar, from 1998 to 2003. He has been in the Editorial Board of the IEEE DESIGN AND TEST, the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS, the IEEE TRANSACTIONS ON VLSI SYSTEMS, and the IEEE TRANSACTIONS ON ELECTRON DEVICES. He was a Guest Editor for Special Issue on Low-Power VLSI in the IEEE DESIGN AND TEST, in 1994, and the IEEE TRANSACTIONS ON VLSI SYSTEMS, in 2000, *IEE Proceedings–Computers and Digital Techniques*, in 2002, and the IEEE JOURNAL ON EMERGING AND SELECTED TOPICS IN CIRCUITS AND SYSTEMS, in 2011.