POLARIZING FRONT ENDS FOR ROBUST CNNS

Can Bakiskan * *Soorya Gopalakrishnan* *Metehan Cekic* *Upamanyu Madhow* *Ramtin Pedarsani*

University of California, Santa Barbara, Department of Electrical and Computer Engineering

Index Terms— adversarial machine learning, quantization, front-end defense

ABSTRACT

The vulnerability of deep neural networks to small, adversarially designed perturbations can be attributed to their "excessive linearity." In this paper, we propose a bottom-up strategy for attenuating adversarial perturbations using a nonlinear front end which polarizes and quantizes the data. We observe that ideal polarization can be utilized to completely eliminate perturbations, develop algorithms to learn approximately polarizing bases for data, and investigate the effectiveness of the proposed strategy on the MNIST and Fashion MNIST datasets.

1. INTRODUCTION

Given the immense impact of deep learning on a diversity of fields, its vulnerability to tiny *adversarial* perturbations [1, 2] is of great concern. For image datasets, for example, such perturbations are almost imperceptible for humans, but they can render state-of-the-art models useless, causing misclassification with high confidence. State of the art adversarial attacks are variants of gradient ascent, utilizing the local linearity of deep networks. State of the art defenses are based on adversarial training, using training examples obtained using adversarial attacks, but yield little insight into, or guarantees of, the achieved robustness.

In this paper, we investigate a systematic, bottom-up approach to robustness, studying a defense based on a nonlinear front end for attenuating adversarial perturbations before they reach the deep network. We focus on ℓ_∞-bounded perturbations. Our approach consists of *polarizing* the input data into well-separated clusters by projecting onto an appropriately selected basis (implemented using convolutional filters), and then quantizing the output using thresholds that scale with the ℓ_1 norm of the basis functions. For ideal polarization, we prove that perturbations are completely eliminated. We introduce a regularization technique to learn polarizing bases from data, and demonstrate the efficacy of the proposed defense for the MNIST and Fashion MNIST datasets.

*Corresponding author: canbakiskan@ucsb.edu

2. BACKGROUND

Suppose we have a classifier that takes in inputs $\boldsymbol{x} \in \mathbb{R}^N$, and outputs predictions (confidence scores for M classes) $\boldsymbol{y} \in [0,1]^M$. Our goal is to defend against malicious inputs of the form $\boldsymbol{x}+\boldsymbol{e}$, where $\boldsymbol{e} \in \mathbb{R}^N$ is a small perturbation that aims to cause misclassification. Formally, we can describe such adversarial attacks as a maximization problem:

$$\max_{\boldsymbol{e}\in\mathcal{S}} L(\boldsymbol{\theta}, \boldsymbol{x}+\boldsymbol{e}, \boldsymbol{y}_{\text{true}}), \tag{1}$$

where L is a loss function, $\boldsymbol{\theta}$ denotes network weights and biases and $\boldsymbol{y}_{\text{true}}$ is the vector of true labels. The adversary aims to find the perturbation that maximizes L, subject to the condition that $\boldsymbol{e}$ is chosen from a set $\mathcal{S}$ (typically ℓ_p bounded). In this paper, we focus on ℓ_∞ bounded attacks: $\|\boldsymbol{e}\|_\infty \leq \epsilon$ for an "attack budget" $\epsilon > 0$. Furthermore, we assume a "white box" attack, in which the adversary has full knowledge of the network structure and weights.

Attacks: State of the art ℓ_∞ bounded attacks (used in our evaluations) are all based on gradient ascent on the cost function in (1). The *Fast Gradient Sign Method (FGSM)* [3], computes the perturbation by

$$\boldsymbol{e} = \epsilon \cdot \text{sign}(\nabla_{\boldsymbol{x}} L(\boldsymbol{\theta}, \boldsymbol{x}, \boldsymbol{y})) \tag{2}$$

An iterative version of FGSM known as the *Basic Iterative Method (BIM)* [4] finds the perturbation as

$$\boldsymbol{e}_{i+1} = \text{Clip}_\epsilon\Big(\boldsymbol{e}_i + \alpha \cdot \text{sign}(\nabla_{\boldsymbol{x}} L(\boldsymbol{\theta}, \boldsymbol{x}+\boldsymbol{e}_i, \boldsymbol{y}))\Big) \tag{3}$$

where α is the step size for each iteration, and ϵ is the overall ℓ_∞ attack budget. It was noted in [5] that BIM is a formulation of Projected Gradient Descent (PGD), a well-known method in convex optimization. The PGD attack suggested in [5] employs BIM with multiple random starting points sampled from a uniform distribution in the ϵ box around the data point. We term this scheme *PGD with Restarts.*

Defenses: Defenders seek to minimize (1), so that learning in an adversarial setting may be viewed as a minimax game. A number of defense mechanisms have been proposed, only to be defeated by stronger adversaries [6, 7]. The current state of the art defense employs retraining with adversarial examples [5]. However, there is no design intuition as to why and

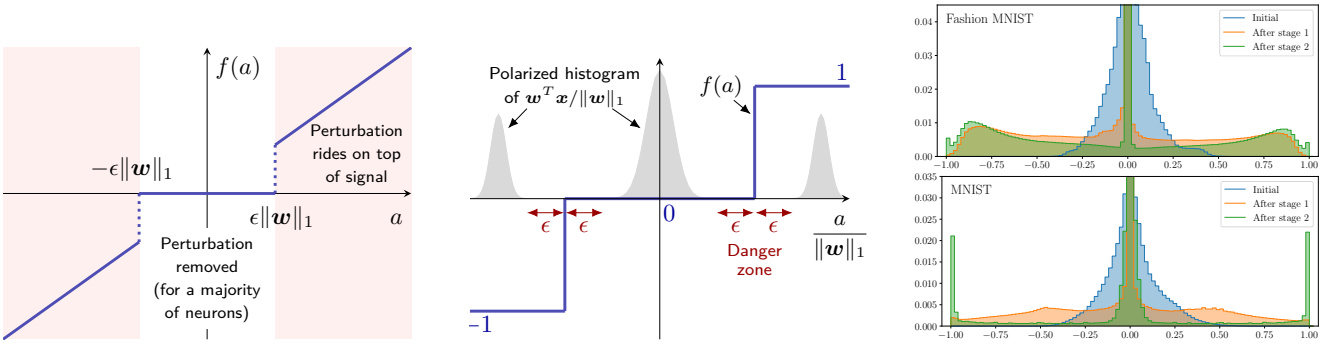

Fig. 1. (a) Activation sparsity (Eq. 4) alone is not sufficient to achieve robustness: perturbations can ride on top of strongly activated neurons (shaded region). (b) Polarization of neural activity can fully eliminate perturbations. For the shown hypothetical histogram (gray) of $\boldsymbol{w}^T\boldsymbol{x}/\|\boldsymbol{w}\|_1$, a ternary activation (blue) is effective. (c) Probability distribution of normalized front-end filter outputs $a_k/\|\boldsymbol{w}_k\|_1$.

how perturbations are being controlled as they flow up the network. It is also computationally intensive, slowing down training by an order of magnitude. A more efficient and interpretable line of defenses employ data preprocessing prior to the network. For example, sparsity-based preprocessing was shown to be effective for linear classifiers [8] and neural networks [9, 10]. More recently, [11] proposed preprocessing by randomly erasing pixels of the image, followed by reconstruction using well-known matrix estimation methods. When combined with adversarial training, [11] achieves state-of-the-art performance on MNIST, CIFAR-10, SVHN, and Tiny-ImageNet datasets.

A number of quantization-based defense methods have been proposed in literature, within the neural network [12, 13] and as a front end [14–17]. The key difference in our proposed strategy is that we employ polarization prior to quantization, which enables theoretical guarantees on robustness (Section 3).

Gradient Masking: The use of non-differentiable functions or functions with a saturation region can cause state of the art gradient-based attacks to falter. However, defenses that rely on such "gradient masking" are not robust: they are easily circumvented by the attacker, as shown in [7], by replacing the non-differentiable function by a differentiable approximation. We test our defense using the gradient approximation methods of [7], replacing non-differentiable functions with identity in the gradient calculations. We have also performed experiments with other differentiable approximations to our quantization function, but found identity approximation to be the most effective for the attack.

3. POLARIZATION-BASED DEFENSE

We investigate a defense based on a front end which preprocesses the inputs via a linear transformation followed by a nonlinear activation f. Following convention, the linear operation of a particular filter is termed a *neuron*. Consider a typical front-end neuron with weights $\boldsymbol{w}$, and scalar output a. For perturbed input $\boldsymbol{x}+\boldsymbol{e}$ with ℓ_∞ bound $\|\boldsymbol{e}\|_\infty < \epsilon$, a contains two components: desired signal $\boldsymbol{w}^T\boldsymbol{x}$, and an output perturbation $\boldsymbol{w}^T\boldsymbol{e}$ that is constrained in magnitude: $\left|\boldsymbol{w}^T\boldsymbol{e}\right| \leq \|\boldsymbol{e}\|_\infty\|\boldsymbol{w}\|_1 \leq \epsilon\|\boldsymbol{w}\|_1$ due to Hölder's inequality. For the defense to be successful, the nonlinearity f must be chosen such that $f(a = \boldsymbol{w}^T(\boldsymbol{x}+\boldsymbol{e})) \approx f(\boldsymbol{w}^T\boldsymbol{x})$.

One design approach is to promote sparse activations by increasing the threshold for neurons to fire, which makes it difficult for a small perturbation to induce firing:

$$f(a) = \begin{cases} 0, & |a| \leq \epsilon\|\boldsymbol{w}\|_1 \\ a, & \text{otherwise.} \end{cases} \tag{4}$$

While this method helps (see [8–10] for a similar approach), Fig. 1a shows why it cannot be completely successful. When a neuron resides near the middle of the unshaded region, no perturbation can change the signal output ($f(a) = 0$). However, neurons with a strong desired signal component (large $|\boldsymbol{w}^T\boldsymbol{x}|$) can serve as hosts for the perturbation, allowing it to propagate through the defense. Hence activation sparsity can only be a part of the solution.

What if we could somehow *polarize* neural activity to obtain well-separated clusters of neurons? Consider for instance the three clusters of activations shown in Fig. 1b. In such a scenario, we can completely eliminate perturbations by using a quantized nonlinearity (in this case, ternary quantization). Note that it is important for neurons to avoid the "danger zones" of width 2ϵ shown in the figure: this ensures that perturbations cannot switch data from one quantization level to the next. These observations are formalized in the following proposition.

Proposition 1. *Suppose the front end polarizes activations into a multimodal distribution with L clusters, with minimum*

inter-cluster separation $d > 2\epsilon\|\boldsymbol{w}\|_1$. Let $c_1 < c_2 < \ldots c_{L-1}$ denote the midpoints between adjacent clusters. Then the following L-level quantizer (with thresholds at c_i) completely eliminates perturbations with ℓ_∞ norm smaller than ϵ:

$$f(a) = \frac{1}{2}\sum_{i=1}^{L-1} \text{sign}(a - c_i). \tag{5}$$

Proof. Since we use a quantizing nonlinearity, perturbations can cause distortion only if the output switches quantization levels. We know that for a perturbation $\boldsymbol{e}$ with ℓ_∞ budget ϵ, the maximum output distortion is $\epsilon\|\boldsymbol{w}\|_1$. Therefore, if clusters are separated by a distance of $2\epsilon\|\boldsymbol{w}\|_1$, perturbations cannot propagate through the defense. □

This result motivates a second design approach, where we seek a neural basis in which outputs are well-polarized for clean inputs, with clusters of $\boldsymbol{w}^T\boldsymbol{x}/\|\boldsymbol{w}\|_1$ separated by at least 2ϵ as shown in Fig. 1b. We can then choose a piecewise constant nonlinearity (Eq. 5) to eliminate the effect of perturbations. Equipped with these design principles, we now detail training procedures to learn polarizing bases from data.

3.1. Implementing a Polarizing Front End

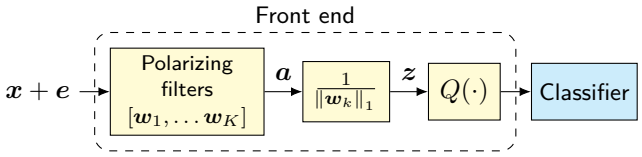

Fig. 2. Block diagram of front end defense, showing a polarizing filter followed by ℓ_1 normalization and quantization.

We employ a front end (shown in Fig. 2) which uses convolutional filters to learn polarized and quantized latent representations of data. For front end neuron $\boldsymbol{w}_k$, let $z_k = a_k/\|\boldsymbol{w}_k\|_1$ denote the normalized activation. We seek a multimodal distribution for $\boldsymbol{z}$, with clusters separated by at least 2ϵ. We achieve this by training with *bump regularizers* $B_1(\cdot)$ and $B_2(\cdot)$ which promote polarization of data. We train in three stages by minimizing the modified loss function:

$$\mathcal{L}(\boldsymbol{y}, \boldsymbol{y}_{\text{true}}, \boldsymbol{z}) = \mathcal{L}_{CE}(\boldsymbol{y}, \boldsymbol{y}_{\text{true}}) + \frac{\lambda}{K}\sum_{k=1}^{K} B(z_k).$$

where $\mathcal{L}_{CE}$ is the cross-entropy loss determined by the true label and outputs of the classifier, K is the number of neurons, $\boldsymbol{z}$ is the vector of activations of all neurons $[z_1, \ldots z_K]$, B is the regularizer and λ is a scaling coefficient. These stages can be described as follows:

1. We start by training the polarizer without using any quantization. The front end filters are initialized from a uniform distibution described in [18]. Due to the random initialization, normalized activations are typically clustered around zero initially (shown in Fig. 1c). Next we incorporate a bump regularizer $B(\cdot) = B_1(\cdot)$ to drive the normalized activations away from the origin, pushing $\boldsymbol{z}$ towards the endpoints -1 and 1.

$$B_1(z_k) = e^{-z_k^2/2\sigma_1^2}.$$

2. After achieving a sufficiently even level of distribution throughout the interval $[-1, 1]$, we switch to the second bump regularizer $B(\cdot) = B_2(\cdot)$, aimed at pushing the normalized activations away from the quantization thresholds $\pm c$ and polarizing $\boldsymbol{z}$ into three clusters centered at -1, 0 and 1.

$$B_2(z_k) = e^{-(z_k-c)^2/2\sigma_2^2} + e^{-(z_k+c)^2/2\sigma_2^2}.$$

3. Now we introduce the quantization function $f_2(\cdot)$ described in Eq. 6. We also freeze and stop training the filters in the front end, and remove the regularizer. We train the classifier to let the weights adapt to the quantization.

$$f_2(z_k) = 0.5\,\text{sgn}(z_k - c) + 0.5\,\text{sgn}(z_k + c). \tag{6}$$

For testing, we continue using the quantized activation in (6) to eliminate perturbations. Details regarding the choice of parameters such as λ, c, σ_1 and σ_2 are given in Section 4.

We find that these three stages suffice for Fashion MNIST and MNIST, but depending on the dataset, one could potentially repeat Stage 2 with an increasing number of clusters until the desired level of polarization is achieved. Fig. 1c demonstrates the effects bump regularizers have on the distribution of normalized activations.

Fig. 3 shows the filters obtained after each stage for MNIST and Fashion MNIST. Interestingly, we find that the

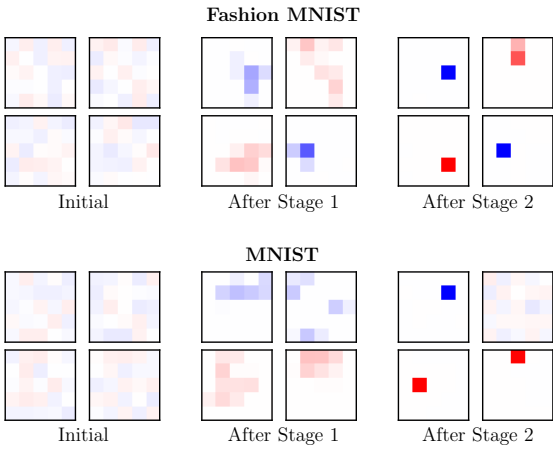

Fig. 3. Typical progression of front-end filters over stages.

learnt filters appear similar to pixel bases. This is consistent with the observations in [5] about first-layer filters learnt by adversarial training on MNIST.

4. EXPERIMENTS AND RESULTS

4.1. Training Details

For a fair comparison we use the small convolutional neural network from [5], consisting of two convolutional layers and two fully connected layers. Convolutional layers have 32 and 64 number of filters that are 5x5 in size. Each convolutional layer is followed by 2x2 maxpooling operation. Every layer except the last uses ReLU activation function. The outputs of the last layer are fed into a softmax function to generate classification probabilities. In every run, the model is trained for 20 epochs in each stage for a total of 60 epochs. Gradient descent is achieved using the Adam optimizer [19] with learning rate 10^{-3} and default hyperparameters in PyTorch library.

During training with bump regularizers, stage 1 and stage 2 bump widths are picked to be $\sigma_1 = 0.35$ and $\sigma_2 = 0.15$ respectively. To make the adaptation of weights smoother, we increase the bump coefficient λ linearly from 0 to 1 in each stage, as the stages progress. The quantization threshold is chosen to be $c = 0.3$ for Fashion MNIST and $c = 0.5$ for MNIST. When adversarially training using the methods of Madry et al. [5] we use 10 restarts and 20 steps in each restart.

Attack Setup: We evaluate our defense against the white box attacks described in Section 2: FGSM, BIM and PGD with Restarts. We use attack budget $\epsilon = 0.3$ for MNIST and $\epsilon = 0.1$ for Fashion MNIST. In iterative methods, we use step size $\alpha = \epsilon/10$. In BIM, we use 20 steps. In PGD we choose the best performing attack from 20 random restarts, with 100 steps in each restart.

4.2. Results and Discussion

Fig. 4 shows the effect of attack budget on accuracy, showing that our front end increases adversarial accuracy across a wide range of ϵ. Table 1 details our results against different attacks, with a comparison with methods from literature.

Our defense significantly improves robustness against a variety of attacks, but falls short of the accuracies obtained by adversarial training. This is because perfect polarization is not possible in practice, leading to some leakage of adversarial perturbations through the front end. However, the polarization approach is amenable to interpretation, and provides an avenue for further efforts in systematic bottom-up design. In contrast, empirical experiments are the only means of verifying the efficacy of state of the art adversarial training.

Table 1. Experimental results for different attacks.

	Fashion MNIST ($\epsilon = 0.1$)				MNIST ($\epsilon = 0.3$)			
	Clean	FGSM	BIM	PGD	Clean	FGSM	BIM	PGD
No defense	91.6	19.7	1.49	0.11	99.4	21.9	0.47	0.00
Adv. Training	83.8	77.9	75.6	74.1	97.5	93.1	90.0	86.7
Ours	87.3	69.4	54.9	44.5	99.1	93.2	86.5	70.8

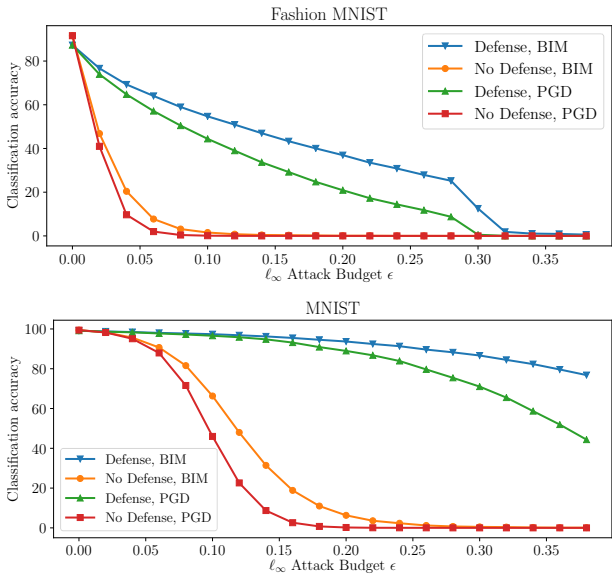

Fig. 4. Classification accuracy versus ℓ_∞ attack budget.

5. CONCLUSIONS

In this paper, we have shown that polarization is a promising tool for defense against adversarial attacks: when data is perfectly polarized, quantization can provably eliminate perturbations. Our training procedures for learning polarizing bases indicate that pixel bases are effective for polarizing datasets like MNIST and Fashion MNIST, which is consistent with the first-layer filters learnt in adversarially trained models for these datasets [5]. While we consider a supervised learning framework here, we have also obtained promising results with unsupervised learning of polarizing bases, but omit discussion due to space constraints. Open problems for future work include combining polarizing front ends with nonlinearities within the network in order to provably attenuate attacks as they flow through the network, and obtaining polarizing bases for more complex datasets such as CIFAR and ImageNet.

6. ACKNOWLEDGEMENTS

This work was supported in part by the Army Research Office under grant W911NF-19-1-0053, and by the National Science Foundation under grant CIF-1909320.

7. REFERENCES

[1] B. Biggio and F. Roli, "Wild patterns: Ten years after the rise of adversarial machine learning," *Pattern Recognition*, vol. 84, pp. 317–331, December 2018.

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

[3] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *International Conference on Learning Representations*, 2015.

[4] A. Kurakin, I. J. Goodfellow, and S. Bengio, "Adversarial machine learning at scale," in *International Conference on Learning Representations*, 2017.

[5] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," in *International Conference on Learning Representations*, 2018.

[6] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *IEEE Symposium on Security and Privacy*, 2017, pp. 39–57.

[7] A. Athalye, N. Carlini, and D. Wagner, "Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples," in *International Conference on Machine Learning*, 2018.

[8] Z. Marzi, S. Gopalakrishnan, U. Madhow, and R. Pedarsani, "Sparsity-based defense against adversarial attacks on linear classifiers," in *IEEE International Symposium on Information Theory (ISIT)*, 2018.

[9] S. Gopalakrishnan, Z. Marzi, U. Madhow, and R. Pedarsani, "Combating adversarial attacks using sparse representations," in *International Conference on Learning Representations (ICLR) Workshop Track*, 2018.

[10] S. Gopalakrishnan, Z. Marzi, U. Madhow, and R. Pedarsani, "Robust adversarial learning via sparsifying front ends," *arXiv preprint arXiv:1810.10625*, 2018.

[11] Y. Yang, G. Zhang, D. Katabi, and Z. Xu, "ME-Net: Towards effective adversarial robustness with matrix estimation," in *International Conference on Machine Learning*, 2019.

[12] J. Lin, C. Gan, and S. Han, "Defensive quantization: When efficiency meets robustness," in *International Conference on Learning Representations*, 2019.

[13] A.S. Rakin, J. Yi, B. Gong, and D. Fan, "Defend deep neural networks against adversarial examples via fixed and dynamic quantized activation functions," *arXiv preprint arXiv:1807.06714*, 2018.

[14] H. Ali, H. Hammad Tariq, M. A. Hanif, F. Khalid, S. Rehman, R. Ahmed, and M. Shafique, "QuSecNets: Quantization-based defense mechanism for securing deep neural network against adversarial attacks," in *IEEE International Symposium on On-Line Testing and Robust System Design*, 2019.

[15] P. Panda, I. Chakraborty, and K. Roy, "Discretization based solutions for secure machine learning against adversarial attacks," *IEEE Access*, vol. 7, 2019.

[16] J. Chen, X. Wu, V. Rastogi, Liang Y., and S. Jha, "Towards understanding limitations of pixel discretization against adversarial attacks," in *IEEE European Symposium on Security and Privacy*, 2019.

[17] Y. Zhang and P. Liang, "Defending against whitebox adversarial attacks via randomized discretization," in *International Conference on Artificial Intelligence and Statistics*, 2019.

[18] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1026–1034.

[19] D.P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *International Conference on Learning Representations*, 2015.