Robust Sparse Regularization: Defending Adversarial Attacks Via Regularized Sparse Network

Adnan Siraj Rakin
Arizona State University
asrakin@asu.edu

Zhezhi He
Arizona State University
zhezhihe@asu.edu

Li Yang
Arizona State University
lyang166@asu.edu

Yanzhi Wang
Northeastern University
yanz.wang@northeastern.edu

Liqiang Wang
University of Central Florida
lwang@cs.ucf.edu

Deliang Fan
Arizona State University
dfan@asu.edu

ABSTRACT

Deep Neural Network (DNN) trained by the gradient descent method is known to be vulnerable to maliciously perturbed adversarial input, aka. adversarial attack. As one of the countermeasures against adversarial attacks, increasing the model capacity for DNN robustness enhancement was discussed and reported as an effective approach by many recent works. In this work, we show that shrinking the model size through proper weight pruning can even be helpful to improve the DNN robustness under adversarial attack. For obtaining a simultaneously robust and compact DNN model, we propose a multi-objective training method called *Robust Sparse Regularization* (RSR), through the fusion of various regularization techniques, including channel-wise noise injection, lasso weight penalty, and adversarial training. We conduct extensive experiments to show the effectiveness of RSR against popular white-box (i.e., PGD and FGSM) and black-box attacks. Thanks to RSR, **85%** weight connections of ResNet-18 can be pruned while still achieving **0.68%** and **8.72%** improvement in clean- and perturbed-data accuracy respectively on CIFAR-10 dataset, in comparison to its PGD adversarial training baseline.

CCS CONCEPTS

• **Computing methodologies** → **Object recognition**; *Adversarial learning*.

KEYWORDS

Robust, Sparse, Adversarial Defense

ACM Reference Format:
Adnan Siraj Rakin, Zhezhi He, Li Yang, Yanzhi Wang, Liqiang Wang, and Deliang Fan. 2020. Robust Sparse Regularization: Defending Adversarial Attacks Via Regularized Sparse Network. In *Great Lakes Symposium on VLSI 2020 (GLSVLSI '20), September 7–9, 2020, Virtual Event, China.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3386263.3407651

GLSVLSI '20, September 7–9, 2020, Virtual Event, China

ACM ISBN 978-1-4503-7944-1/20/09…$15.00
https://doi.org/10.1145/3386263.3407651

1 INTRODUCTION

Deep Neural Networks (DNNs) have led to tremendous success in various applications, such as image classification [16], speech recognition [15], medical applications [17] and etc. Wide deployment of DNNs has raised several major security concerns [1, 3, 7]. For example, in the context of image classification, an adversarial example is a carefully modified image that is visually imperceptible to human eyes, but fools the DNN successfully [7]. Recently, there have been a cohort of works toward developing new adversarial attack techniques, which have exposed the underlying vulnerability of DNN [2, 22]. In order to counter adversarial attacks, several works have proposed different techniques, such as training the network with adversarial samples [7, 22], regularization [14, 20] and other various methods [20].

In a separate yet related track, investigation towards generating efficient and compact networks have also been accelerated. Many prior works have been conducted regarding compression techniques including quantization [5, 23, 24, 32] and weight pruning [10]. It is shown that many DNNs can function properly (no accuracy loss) even after significantly (>90%) network pruning [9]. Such sparse DNN achieves significant speed-up and compression rate which opens the door for DNN in memory and resource constraint applications [11]. Previously, several works have tried to generate both sparse and robust networks [8, 31] by combining the network pruning (i.e. compactness) technique to defend adversarial examples. However, their efforts either suffer from poor test data accuracy or do not improve the robustness significantly.

Overview of RSR. In this work, we propose a multi-objective optimization mechanism that could lead these two different yet related tracks, namely pursuit of network robustness and compression, to merge. To achieve this objective, we propose a novel Robust Sparse Regularization (RSR) method which integrates several regularization techniques to achieve such dual optimization. First, we propose to train a DNN with channel-wise noise injection (CNI) embedded with adversarial training to improve network robustness. Such technique injects a channel-wise Gaussian noise which is trainable during adversarial training. CNI improves test accuracy for both clean and perturbed data. Second, in order to simultaneously achieve network compactness and robustness, we propose a new ensemble loss function including an L-1 weight penalty term (i.e. lasso). Lasso regularization during adversarial training performs weight selection by constraining some weight values to a very small values as shown in figure 1. When training is

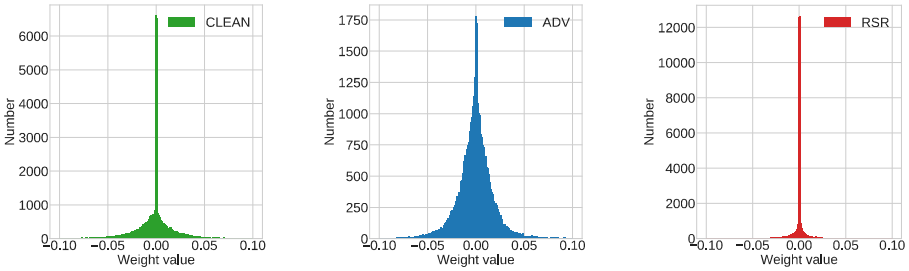

Figure 1: Weight distribution of ResNet-18 for the second convolution layer. We show three cases sequentially 1) clean data training 2) adversarial training(PGD) and 3) RSR training. Observation 1: Adversarial trained network has less sparsity than clean training thus making network pruning difficult. Observation 2: RSR training can achieve the desired sparseness with adversarial training.

done, we could prune the small weight values based on a threshold to achieve a sparse network. Our extensive experiments show that RSR training is an effective network pruning scheme to achieve improved robustness without sacrificing any clean data accuracy across different architectures.

2 RELATED WORKS

Several works [7, 22] have proposed to jointly train the network with adversarial and clean samples, called adversarial training, to achieve network robustness. Later, development of backward pass differential attack (BPDA) [2] has exposed the underlying vulnerability of many other defense methods relying on gradient obfuscation [6, 18]. Then, training the network with adversarial examples has become one of the most popular defense approach to defend adversarial examples. Meanwhile, there is a cohort of work investigating the effect of regularization techniques such as quantization [20, 25] , noise injection [14, 19, 21] and pruning [6, 8, 30, 31] to improve the robustness. Several previous works have investigated the effects of network pruning on robustness [8, 30, 31]. Recently, [31] proposed concurrent weight pruning and adversarial training to generate robust and sparse network. However, their ADMM based pruning method's performance on smaller network (i.e, lesser width) suffers from poor test accuracy for both clean and adversarial data. Further, [8] showed that pruned network will defend adversarial examples provided that the network is not over-sparsified.

3 APPROACH

In this section, we first introduce the proposed *Robust Sparse Regularization* (RSR) technique, which is incorporated into a multi-objective optimization process that simultaneously improves network robustness and compactness. Our proposed RSR mainly consists of two components: 1) a trainable Channel-wise Noise Injection (CNI) and 2) lasso weight penalty (L-1 norm) for model pruning, which will be introduced in this section.

3.1 Adversarial Training

Training the neural network with adversarial examples is a popular defense method [7, 22]. Since our method integrates with such adversarial training, we briefly introduce it first. The goal of adversarial training can be formalized as: if we have a set of inputs- $\boldsymbol{x}$ and target labels- t, adversarial training tries to obtain the optimal solution of network parameters $\boldsymbol{\theta}$ (i.e, weights, biases) for the following min-max optimization problem:

$$\arg\min_{\boldsymbol{\theta}} \left\{ \arg\max_{\boldsymbol{x}' \in P_\epsilon(\boldsymbol{x})} \mathcal{L}\big(g(\hat{\boldsymbol{x}}; \boldsymbol{\theta}), t\big) \right\} \tag{1}$$

where the min-max optimization is composed of inner maximization and outer minimization problem. For inner maximization we acquire the perturbed data $\hat{\boldsymbol{x}}$ as shown in the description of PGD attack [22]. While the outer minimization is optimized through gradient descent method during network training.

3.2 Channel-wise Noise Injection

The first regularization technique used in RSR is to inject learnable channel-wise noise on weights during the DNN adversarial training process. Considering a convolution layer in DNN with 4-D weight tensor $\boldsymbol{W} \in \mathbb{R}^{q \times p \times kh \times kw}$, where q, p, kh, kw denotes number of output channel, input channel, kernel height and kernel width respectively, the *Channel-wise Noise Injection* (CNI) can be mathematically described as:

$$\tilde{\boldsymbol{W}} = f_{\text{CNI}}(\boldsymbol{W}) = \boldsymbol{W} + \boldsymbol{\alpha} \times \boldsymbol{\eta}; \quad \boldsymbol{\eta} \sim \mathcal{N}(0, \sigma^2) \tag{2}$$

where $\boldsymbol{\alpha} \in \mathbb{R}^{q,1,1,1}$ is trainable noise scaling coefficient. $\boldsymbol{\eta} \in \mathbb{R}^{q \times p \times kh \times kw}$ is the noise tensor where its elements are independently sampled from a Gaussian distributed source with zero mean and variance as σ^2. Note that, σ^2 is the variance of $\boldsymbol{W}$ that is statistically calculated in the run-time. Preliminary work [14] shows that parametric noise injection is an improved variant of adversarial training, where such trainable noise injection method could effectively regularize DNN during the adversarial training. We follow similar optimization and update rule for α, but extending it into channel-wise version, where weights for each output channel shares the same noise scaling coefficient instead of whole layer.

We train the network with both clean and adversarial samples to achieve a good balance between adversarial data and clean test data accuracy. Optimization problem of equation 1 can be solved by minimizing the ensemble loss $\mathcal{L}_{\text{ens}}$ in equation 3. The ensemble loss is basically the weighted sum of losses for clean and adversarial data with channel-wise trainable noise injected on weights of DNN

model:

$$\mathcal{L}_{\text{ens}} = a \cdot \mathcal{L}(g(\boldsymbol{x}; f_{\text{CNI}}(\boldsymbol{\theta})), \boldsymbol{t}) + (1-a) \cdot \mathcal{L}(g(\hat{\boldsymbol{x}}; f_{\text{CNI}}(\boldsymbol{\theta})), \boldsymbol{t}) \quad (3)$$

where a is the coefficient to balance the ensemble loss terms which is chosen as 0.5 by default. Optimizing the loss function $\mathcal{L}$ improves network robustness. The optimizer tries to solve for both model parameters $\boldsymbol{\theta}$ and $\boldsymbol{\alpha}$ to find an equilibrium between clean and perturbed data.

3.3 Lasso Weight Penalty

For incorporating the network pruning into the adversarial training, we propose to train the neural network with lasso weight penalty. *Lasso* is known as least absolute shrinkage and selection operator [27]. It was introduced as a L-1 regularizer that penalizes the features with higher values. Lasso is an ideal choice for weight pruning as it shrinks the lesser important weights to zero [13, 28, 29]. We include the lasso weight penalty term into $\mathcal{L}_{\text{ens}}$ and reformat equation 3 as:

$$\mathcal{L}_{\text{ens}} = a \cdot \mathcal{L}(g(\boldsymbol{x}; f_{\text{CNI}}(\boldsymbol{\theta})), t) + (1-a) \cdot \mathcal{L}(g(\hat{\boldsymbol{x}}; f_{\text{CNI}}(\boldsymbol{\theta})), t) \quad (4)$$
$$+\lambda \cdot \textstyle\sum_{l=1}^{L} ||\boldsymbol{W}||_1$$

where $\boldsymbol{W}_l$ denotes the weight tensor of l-th layer, and L is the total number of parametric layers (i.e., convolution and fully-connected layer). $||\cdot||$ is the absolute sum of all the elements of a tensor. The effect of lasso weight penalty is determined by the coefficient λ, where λ in larger value would generate a sparse model containing a significant amount of weight with near zero values. We tune λ experimentally and describe the procedure for selecting optimized value of λ in section 4.2.

3.4 Weight Pruning

The proposed ensemble loss $\mathcal{L}_{\text{ens}}$ serves the purpose of multi-objective loss function. We expect a network after training with the ensemble loss to be more resilient to adversarial samples. At the same time, due to the presence of lasso weight penalty, we expect a significant portion of the weight tensor to converge to near zero values. We then perform weight pruning after training with the proposed ensemble loss function, by setting the weights below a certain threshold (γ) to zero. Note that, after pruning, we remove the noise injection term for zero-value weights. As a result, during inference, we only add noise to the non-zero elements of the weight tensor. For the weight tensor in a fully connected layer, lets assume $\boldsymbol{W} = (a_{i,j})_{i,j=1}^{n,m}$ ($\boldsymbol{W} \in \mathbb{R}^{m \times n}$). For convolution layer, $\boldsymbol{W} = (a_{i,j,k,l})_{i,j,k,l=1}^{q,p,kh,kw}$ ($\boldsymbol{W} \in \mathbb{R}^{q \times p \times kh \times kw}$). Then, the pruning operation can be described as:

$$\text{FC layer} \rightarrow a_{i,j} = 0 \text{ if } |a_{i,j}| < \gamma \quad (5)$$

$$\text{Conv. layer} \rightarrow a_{i,j,k,l} = 0 \text{ if } |a_{i,j,k,l}| < \gamma \quad (6)$$

Here γ is the threshold, which is the least absolute non-zero value in the weight tensor after pruning. Again, we can tune the value of γ for different networks to achieve different sparsity ratio. Hence, by tuning the value of γ, we can effectively show the maximum amount of parameters that can be pruned without causing robustness degradation.

4 EXPERIMENTS

4.1 Experiment setup

Datasets and network architectures. In this work, we only consider CIFAR-10 dataset for image classification task as most of the baseline works report their robustness in terms of under-attack accuracy on this dataset. CIFAR-10 is composed of 50K training samples and 10K test samples. Our data augmentation method is same as described in [12]. Attacker can directly add noise to the natural images as our data normalization layer is placed in front of the DNN as a non-trainable layer. We adopt three classical networks, ResNet-20, ResNet-18 [12] and VGG-16 [26], to perform comparative analysis. We also show the analysis on the effect of network width by varying the width of ResNet-18 network. We report the mean accuracy with 5 trials due to the presence of randomness in both CNI and PGD [22]. We tune the hyper parameter λ to be $1e^{-5}$ for both ResNet-18 and VGG-16 and $5e^{-5}$ for ResNet-20.

Competing methods for adversarial defense. In this work, PGD adversarial training [22] is selected as the primary baseline method. Additionally, our work includes channel-wise noise injection, so we also compare the method with parametric noise injection (PNI) [14]. Additionally, we also compare our work with several network compression and pruning methods [20, 31]. Finally, comparison with several state-of-the art regularization techniques serving as a adversarial defense [19, 21] is also presented.

4.2 Results

White-Box Attack. Our simulation results on two popular white-box attack PGD [22] and FGSM [7] are presented in table 1. During adversarial training, as stated in section 3.1, we use PGD algorithm to generate the adversarial samples. First, for the regular models, we do not perform any weight pruning. RSR helps to achieve significant robustness enhancement and even improves the clean data accuracy compared to baseline PGD training [22]. We observe that, with increasing the model capacity, network robustness increases as well. The observation of robustness enhancement with increasing network capacity is consistent with previous works [14, 22]. For our proposed RSR, the pattern remains the same. Our best accuracy was obtained using VGG-16 network. We could improve the clean test data accuracy by 0.95% and perturbed data accuracy by 9.48% Under strong PGD attack for VGG-16.

Our proposed RSR can prune 60%, 85% and 50% of the network's weight for ResNet-20, ResNet-18 and VGG-16, respectively, without any clean test accuracy loss. To show the comparative effect of network robustness and sparsity, we prune each of the four training cases (PGD/ CNI/ Lasso/ RSR) by equal amount. The level of sparsity can always be tuned by choosing different values of γ. As expected, both PGD and CNI performance suffers significantly after pruning. On the contrary, RSR outperforms baseline PGD (even without pruning) training method. We observe 8.72% and 6.83% improvement on test accuracy under PGD and FGSM attack respectively for ResNet-18 architecture. Again the most significant improvement observed in the VGG-16 network which has the largest capacity. Such observation confirms that increasing the number of parameters increases the effect of weight penalty and noise injection to enhance the network robustness. Another question to be asked is

Table 1: Summary of CIFAR-10 Results: We report clean and perturbed-data(under PGD and FGSM attack) accuracy (%) on CIFAR-10 test data. To visualize the effect of lasso and CNI, we also report the independent test accuracy for both channel-wise noise injection (CNI) and lasso loss. We report the percentage of weight being pruned (exactly equal to zero) as the sparsity(%). Capacity denotes the number of trainable parameters in the network.

	ResNet-20				ResNet-18				VGG-16			
Capacity	269,722				11,173,962				138,357,544			
Scheme	Clean	PGD	FGSM	Sparsity	Clean	PGD	FGSM	Sparsity	Clean	PGD	FGSM	Sparsity
Before Pruning												
PGD	83.58	39.44	46.87	0	86.11	44.31	53.52	0	82.88	37.57	46.94	0
CNI	84.67	46.11	54.40	0	86.82	47.85	56.04	0	83.13	44.23	51.56	0
Lasso	83.56	38.69	45.78	0	85.92	46.94	55.2	0	83.26	41.93	50.33	0
RSR	**84.96**	**47.95**	**56.72**	0	**86.95**	**52.94**	**60.89**	0	**83.83**	**47.05**	**54.05**	0
After Pruning												
PGD	51.58	12.49	16.11	60.47	70.31	31.00	35.8	85.43	78.40	32.14	42.21	50.62
CNI	55.93	23.91	29.11	60.74	50.97	22.54	25.31	85.27	75.79	40.39	46.37	50.62
Lasso	83.64	38.46	45.44	60.14	85.92	46.8	55.2	85.38	83.24	42.01	50.32	50.15
RSR	**84.32**	**47.44**	**55.74**	**60.85**	**86.79**	**53.03**	**60.35**	**85.36**	**83.02**	**47.70**	**54.16**	**50.93**

Table 2: Ablation study with varying width. We report clean and perturbed-data(under PGD and FGSM attack) accuracy on CIFAR-10. ResNet-18(1×) is chosen as the baseline. Network width $w = 0.5\times$ and $w = 0.25\times$ denotes that the width of the network's both input channel and output channel is scaled by $0.5\times$ and $0.25\times$ respectively.

	Clean Test (%)		Adversarial Attack(PGD) %		Sparsity: (%)		(%) of parameter remain in the network compared to ResNet-18(1×)
Channel Width	Adv. Trained	RSR	Adv. Trained	RSR	Adv. Trained	RSR	RSR
$w = 0.25\times$	82.68	83.18	39.01	45.38	0	38.33	→(100-38.33)×0.125=7.7
$w = 0.5\times$	84.99	84.85	43.33	50.7	0	63.17	→(100-63.17)×0.25=9.21
$w = 1\times$	86.82	86.79	47.85	53.03	0	85.36	→(100-85.36)=14.37

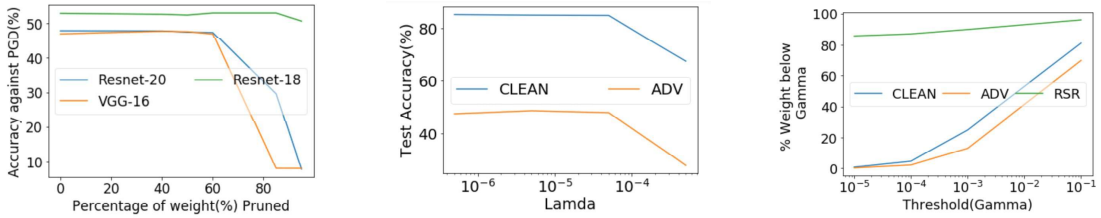

Figure 2: a) The relationship between test accuracy (%) under PGD attack Vs Percentage of weight pruned(exactly equal to zero). It shows each network can be pruned up to certain level of sparsity. Pruning beyond that level would make the model over-sparsified [8] and the network no longer remains robust. b) The plot shows both clean and perturbed data (PGD) accuracy (%) for ResNet-20(RSR) VS Lambda(λ). λ is the regularization parameter for the lasso loss. c) X- axis contains different Gamma(γ) values and Y-axis shows the percentage of weight below a certain threshold Gamma(γ). γ is least absolute value after pruning in a network. Clean, Adv and RSR denotes clean test data training method, adversarial training method and Our proposed RSR method respectively. This plot is only for convolution layer of ResNet-18 architecture.

what if we want to prune the network beyond the reported sparsity. For example, if we want to prune ResNet-18 beyond 85% , does the network still remain robust? We try to answer this question in the next paragraph where we explain the effect of network width with sparsity.

Effect of Network Width. [31] demonstrated that decreasing a network width may have negative impact on robustness. To verify if our method also follows the same trend, we show an ablation study on ResNet-18 with decreasing network width in table 2. Our observation confirms that RSR method still remains more robust than the baseline PGD method for each case (i.e., $w = 0.25\times/0.5\times/1\times$). On the other side, we achieve less sparsity on network with smaller network width. In $w = 0.25\times$ case, we could only achieve 38.33% sparsity without sacrificing any clean or perturbed data accuracy. This observation is quite intuitive as ResNet-18(0.25×) network

Table 3: We compare our method with three major categories of defense: a) Adversarial training defenses: Projected Gradient Descent (PGD) training [22], Parametric Noise Injection (PNI) [14] b) Compression or pruning techniques: Defensive Quantization (DQ) [20], Second Rethinking of Network Pruning (SR) [31] and c) Regularization techniques: Differential Privacy (DP) [19] and Robust Self Ensemble(RSE) [21].

	Adversarial Training		Compression		Regularization		This work
	PGD	PNI	DQ	SR	DP	RSE	RSR
Model	ResNet-18	ResNet-20(4×)	Wide ResNet	ResNet-18	Wide ResNet	ResNext	ResNet-18
Clean(%)	86.11	**87.7**	87.0	81.83	87.0	87.5	86.79
PGD(%)	44.31	49.1	51.8	48.00	25.0	40.0	**53.03**
Sparsity (%)	0	0	(6×) Compression	-	0	0	**85.36**

already has 0.125× less parameters than that of ResNet-18 (1×). Thus, even after performing less amount of weight pruning, the percentage of parameter (7.7%) in the network would still be smaller compared to ResNet-18(1×) (14.37%). Finally, such observation also answers the question asked previously: A particular architecture (e.g., ResNet-18) can be pruned up to a certain amount of sparsity levels based on the network width. The maximum number of parameters which can be pruned without any sacrifice in robustness may vary across different architectures. Figure 2 (a) shows ResNet-20, ResNet-18 and VGG-16 test accuracy under PGD attack starts to drop at different sparsity levels (% weight equal to zero). If any model is sparsified beyond this point, it falls under the definition of over-sparsified model [8] and the network no longer remains robust.

Robustness improvement coming from Lasso training? or CNI training? or Both? We have provided comprehensive experimental analysis on our proposed RSR method to show its performance enhancement on three fronts: clean data accuracy, robustness (i.e. under attack accuracy) and sparsity. Table 1, confirms that lasso loss primarily contributes to the sparse model generation through weight shrinkage. However, in order to identify the chief contributor towards robustness improvement, an ablation study is shown in table 1, where we also report effect of training the network only with lasso loss and CNI, respectively. The regularization effect of lasso is less significant for ResNet-20 and CNI plays the dominant role in network robustness improvement. However, both lasso and Channel-wise noise injection contributes towards the improvement of robustness for redundant networks (i.e, VGG-16). Both lasso and CNI can improve the network robustness by close to 4 % and 7 %, respectively, on VGG-16. Nonetheless, we choose lasso because it helps shrink weight values to a very small value, thus performing a robust model selection during adversarial training. Apart from that, lasso regularization also supplements CNI towards defending adversarial examples .

Choice of Lambda (λ). In figure 2 (b), we show a plot of test accuracy on both clean and perturbed data versus Lambda(λ) for ResNet-20. Clearly, both the test accuracy starts to drop if we increase λ beyond $5e^{-5}$. So for ResNet-20 we choose $5e^{-5}$ as the standard value of λ to achieve the maximum sparsity without any degradation in test accuracy. Similarly, the value of λ for other architectures (i.e, ResNet-18, VGG-16) is optimized experimentally.

Black-Box Attack. We report the black-box attack accuracy for ResNet-20 architecture in table 4. We test our defense method against un-targeted ZOO attack [4]. We randomly select 200 test samples to calculate the attack success rate. Our proposed method defends ZOO attack better as it decreases the attack success rate by 12 % compared to baseline PGD method.

Table 4: Black-Box attack summary. ZOO attack success rate (in 2nd column) is the percentage of test sample being successfully classified to a wrong class by the attack. We report two sets of transfer attack accuracy: one with VGG16 as the source (3rd column) and the other with ResNet-18 as the source (4th column). For both PGD and RSR ResNet-20 is the target model.

Method	ZOO Success rate (%)	Source(VGG-16) Accuracy(%)	Source(ResNet-18) Accuracy(%)
PGD	68.50	66.13	67.44
RSR	**56.00**	66.04	67.27

To perform transferable attack on RSR and PGD, we use VGG-16 and ResNet-18 network as the source model. For both cases, our RSR performs on par with the PGD method. Additionally, our proposed RSR reports higher test accuracy against black-box attack compared to white-box PGD method. Better resistance against black-box attack is considered as a sign of a defense that does not effectively uses obfuscated or masked gradient [2] as a defense tool.

Comparison to state-of-the art techniques. In table 3, we summarize the performance of our defense in comparison to some other state-of-the-art defense techniques. Our proposed RSR method outperforms these comparative defenses and achieves significant robustness improvement.

Note that, we compare with the unbroken defenses that are not reported to show signs of obfuscated gradients [2] yet. Again there are some previous works on network pruning and robustness [6] which might suffer from gradient obfuscation [2]. [8] first theoretically shows the effect of pruning on non-linear DNN to demonstrate the vulnerability of over-sparsified model to adversarial attacks. However, we are the first to formulate an improved adversarial defense with sparse regularization. Our proposed RSR generates sparse and compact neural network that can achieve state-of-the-art under-attack accuracy and much improved robustness.

5 ANALYSIS

RSR is performing regularization. Robust Sparse Regularization is performing regularization on the network to enhance both robustness and compactness. It does not show any obvious signs of

gradient masking proposed in [2]. First, RSR performs better against single step attack (i.e, FGSM) compared to multiple step attack (i.e., PGD). Also we report higher test accuracy against black-box attack than white-box. Finally, increasing the attack strength linearly decreases the effectiveness of our defense. Such observations confirm primarily our robustness enhancement is not achieved through any gradient obfuscation or masking [2]. Instead, our improvement in robustness primarily comes from regularized training method.

Optimal Gamma provides the improvement on three fronts. We can fine-tune the model after training to prune the weights of a network below a certain threshold (γ). During training apart from enhancing robustness, RSR mainly shrinks the weights of the network. The demonstration of weight shrinkage is presented in figure 2 (c). ResNet-18 network training with RSR contains 85% weights with near zero value (less than $1e^{-5}$). So pruning weights with such small values will have minimal effect on clean test accuracy and robustness. Thus, the value of γ can be tuned to an optimal point for each network to achieve improvement on three fronts: clean data accuracy, robustness(i.e., under attack accuracy) and sparsity.

6 CONCLUSION

We successfully co-optimize the objective of network robustness and compactness through our proposed RSR training method. As a result, we show that heavily sparse network can resist adversarial examples to generate both robust and compact neural network at the same time. Our proposed method performs dual optimization during training to resist state-of-art white-box and black-box attacks using a more compact network.

ACKNOWLEDGEMENT

This work is supported in part by the National Science Foundation under Grant No.1931871.

REFERENCES

[1] Naveed Akhtar and Ajmal Mian. 2018. Threat of Adversarial Attacks on Deep Learning in Computer Vision: A Survey. *IEEE Access* 6 (2018), 14410–14430.
[2] Anish Athalye, Nicholas Carlini, and David Wagner. 2018. Obfuscated Gradients Give a False Sense of Security: Circumventing Defenses to Adversarial Examples. In *Proceedings of the 35th International Conference on Machine Learning (Proceedings of Machine Learning Research)*, Jennifer Dy and Andreas Krause (Eds.), Vol. 80. PMLR, Stockholmsmässan, Stockholm Sweden, 274–283.
[3] Hongge Chen, Huan Zhang, Pin-Yu Chen, Jinfeng Yi, and Cho-Jui Hsieh. 2017. Show-and-Fool: Crafting Adversarial Examples for Neural Image Captioning. *arXiv preprint arXiv:1712.02051* (2017).
[4] Pin-Yu Chen, Huan Zhang, Yash Sharma, Jinfeng Yi, and Cho-Jui Hsieh. 2017. Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*. ACM, 15–26.
[5] Matthieu Courbariaux, Yoshua Bengio, and Jean-Pierre David. 2015. Binaryconnect: Training deep neural networks with binary weights during propagations. In *Advances in neural information processing systems*. 3123–3131.
[6] Guneet S. Dhillon, Kamyar Azizzadenesheli, Jeremy D. Bernstein, Jean Kossaifi, Aran Khanna, Zachary C. Lipton, and Animashree Anandkumar. 2018. Stochastic activation pruning for robust adversarial defense. In *International Conference on Learning Representations*. https://openreview.net/forum?id=H1uR4GZRZ
[7] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572* (2014).
[8] Yiwen Guo, Chao Zhang, Changshui Zhang, and Yurong Chen. 2018. Sparse dnns with improved adversarial robustness. In *Advances in neural information processing systems*. 242–251.
[9] Song Han, Huizi Mao, and William J Dally. 2015. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv:1510.00149* (2015).
[10] Song Han, Jeff Pool, John Tran, and William Dally. 2015. Learning both weights and connections for efficient neural network. In *Advances in neural information processing systems*. 1135–1143.
[11] Seungyeop Han, Haichen Shen, Matthai Philipose, Sharad Agarwal, Alec Wolman, and Arvind Krishnamurthy. 2016. Mcdnn: An approximation-based execution framework for deep stream processing under resource constraints. In *Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services*. ACM, 123–136.
[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.
[13] Yihui He, Xiangyu Zhang, and Jian Sun. 2017. Channel pruning for accelerating very deep neural networks. In *Proceedings of the IEEE International Conference on Computer Vision*. 1389–1397.
[14] Zhezhi He, Adnan Siraj Rakin, and Deliang Fan. 2019. Parametric Noise Injection: Trainable Randomness to Improve Deep Neural Network Robustness against Adversarial Attack. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.
[15] Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, et al. 2012. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine* 29, 6 (2012), 82–97.
[16] Geoffrey Hinton, Nitsh Srivastava, and Kevin Swersky. 2012. Neural networks for machine learning. *Coursera, video lectures* 264 (2012).
[17] Chen-Ying Hung, Wei-Chen Chen, Po-Tsun Lai, Ching-Heng Lin, and Chi-Chun Lee. 2017. Comparing deep neural network and other machine learning algorithms for stroke prediction in a large-scale population-based electronic medical claims database. In *2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE, 3110–3113.
[18] Colin Raffel Ian Goodfellow Jacob Buckman, Aurko Roy. 2018. Thermometer Encoding: One Hot Way To Resist Adversarial Examples. *International Conference on Learning Representations* (2018). https://openreview.net/forum?id=S18Su--CW accepted as poster.
[19] M Lecuyer, V Atlidakis, R Geambasu, D Hsu, and S Jana. 2018. Certified Robustness to Adversarial Examples with Differential Privacy. *ArXiv e-prints* (2018).
[20] Ji Lin, Chuang Gan, and Song Han. 2019. Defensive Quantization: When Efficiency Meets Robustness. In *International Conference on Learning Representations*. https://openreview.net/forum?id=ryetZ20ctX
[21] Xuanqing Liu, Minhao Cheng, Huan Zhang, and Cho-Jui Hsieh. 2017. Towards Robust Neural Networks via Random Self-ensemble. *arXiv preprint arXiv:1712.00673* (2017).
[22] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2018. Towards Deep Learning Models Resistant to Adversarial Attacks. In *International Conference on Learning Representations*. https://openreview.net/forum?id=rJzIBfZAb
[23] Adnan Siraj Rakin, Shaahin Angizi, Zhezhi He, and Deliang Fan. 2018. Pim-tgan: A processing-in-memory accelerator for ternary generative adversarial networks. In *2018 IEEE 36th International Conference on Computer Design (ICCD)*. IEEE, 266–273.
[24] Adnan Siraj Rakin, Zhezhi He, and Deliang Fan. 2019. Bit-Flip Attack: Crushing Neural Network With Progressive Bit Search. In *The IEEE International Conference on Computer Vision (ICCV)*.
[25] Adnan Siraj Rakin, Jinfeng Yi, Boqing Gong, and Deliang Fan. 2018. Defend deep neural networks against adversarial examples via fixed anddynamic quantized activation functions. *arXiv preprint arXiv:1807.06714* (2018).
[26] Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014).
[27] Robert Tibshirani. 1996. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)* 58, 1 (1996), 267–288.
[28] Huan Wang, Qiming Zhang, Yuehai Wang, and Haoji Hu. 2018. Structured Pruning for Efficient ConvNets via Incremental Regularization. *arXiv preprint arXiv:1811.08390* (2018).
[29] Wei Wen, Chunpeng Wu, Yandan Wang, Yiran Chen, and Hai Li. 2016. Learning structured sparsity in deep neural networks. In *Advances in neural information processing systems*. 2074–2082.
[30] Shaokai Ye, Siyue Wang, Xiao Wang, Bo Yuan, Wujie Wen, and Xue Lin. 2018. Defending DNN Adversarial Attacks with Pruning and Logits Augmentation. https://openreview.net/forum?id=S1qI2FJDM
[31] Shaokai Ye, Kaidi Xu, Sijia Liu, Hao Cheng, Jan-Henrik Lambrechts, Huan Zhang, Aojun Zhou, Kaisheng Ma, Yanzhi Wang, and Xue Lin. 2019. Second Rethinking of Network Pruning in the Adversarial Setting. *arXiv preprint arXiv:1903.12561* (2019).
[32] Shuchang Zhou, Yuxin Wu, Zekun Ni, Xinyu Zhou, He Wen, and Yuheng Zou. 2016. DoReFa-Net: Training low bitwidth convolutional neural networks with low bitwidth gradients. *arXiv preprint arXiv:1606.06160* (2016).