Regularizer to Mitigate Gradient Masking Effect during Single-Step Adversarial Training

Vivek B S, Arya Baburaj and R. Venkatesh Babu
Video Analytics Lab, Department of Computational and Data Sciences
Indian Institute of Science, Bangalore, India

Abstract

Neural networks are susceptible to adversarial samples: samples with imperceptible noise, crafted to manipulate network's prediction. In order to learn robust models, a training procedure, called Adversarial Training has been introduced. During adversarial training, models are trained with mini-batch containing adversarial samples. In order to scale adversarial training for large datasets and networks, fast and simple methods (e.g., FGSM:Fast Gradient Sign Method) of generating adversarial samples are used while training. It has been shown that models trained using single-step adversarial training methods (i.e., adversarial samples generated using non-iterative methods such as FGSM) are not robust, instead they learn to generate weaker adversaries by masking the gradients. In this work, we propose a regularization term in the training loss, to mitigate the effect of gradient masking during single-step adversarial training. The proposed regularization term causes training loss to increase when the distance between logits (i.e., pre-softmax output of a classifier) for FGSM and R-FGSM (small random noise is added to the clean sample before computing its FGSM sample) adversaries of a clean sample becomes large. The proposed single-step adversarial training is faster than computationally expensive state-of-the-art PGD adversarial training method, and also achieves on par results.

1. Introduction

Deep Neural Networks (DNN) achieve impressive performance across various computer vision tasks including critical applications such as autonomous driving, and medical diagnosis. On the negative side, these networks are susceptible to adversarial samples [23, 4, 16]: samples with imperceptible noise, crafted to manipulate network's prediction. Further, Szegedy *et al*. [23] showed that these adversarial samples transfer across multiple models i.e., adversarial samples crafted on one model are capable of misleading even other models with different architecture. This transferable nature of adversarial samples increases the susceptibility of models deployed in the real world, i.e., vulnerable to black-box attacks [10, 17] (no knowledge of the deployed model is available). Generation of these adversarial samples is performed by making use of simple [4, 24] to complex optimization techniques [2, 13, 15, 20, 19, 14].

In order to learn robust models, Goodfellow *et al*. [4] proposed *Adversarial Training* method. During adversarial training, mini-batches are augmented with adversarial samples. These adversarial samples are generated using fast and simple methods such as *Fast Gradient Sign Method* (FGSM) [4] and its variants, so as to scale adversarial training to large networks and datasets. Kurakin *et al*. [8] observed that models trained using single-step adversarial training methods (i.e., adversarial samples are generated using non-iterative methods such as FGSM) are susceptible to multi-step attacks (e.g., iterative methods such as I-FGSM). Further, Tramer *et al*. [24] explained this pseudo robustness of models trained using single-step adversarial training method is due to gradient masking effect i.e., linear approximation of model's loss function becomes unreliable to generate adversaries of higher perturbation strength. To summarize, models trained using single-step adversarial training, (i) exhibit Gradient Masking effect [24], and Label Leaking effect [8] (for some datasets), and (ii) vulnerable to transfer attacks i.e., black-box attacks.

Madry *et al*. [12] demonstrated that, adversarial training can yield robust models, if adversarial samples included while training closely maximizes the model's loss, and further showed that this can be achieved by generating adversaries using Projected Gradient Descent (PGD) [12]. Since, PGD method is an iterative method, it causes training time to increase substantially. Whereas, in this work we show that it is possible to learn robust models using single-step adversarial training by penalizing the gradient masking effect. To achieve this, we introduce a regularization term in the training loss that penalizes gradient masking effect, and this in turn helps in the inclusion of stronger adversaries that maximizes the training loss. The proposed regularization term causes training loss to increase when the Euclidean distance between logits for FGSM and R-FGSM [24] (ran-

dom noise is added before computing FGSM sample) adversaries of a clean sample is large. Following are the major contributions of this work:

- We propose a regularization term in the training loss that penalizes gradient masking effect during adversarial training. Unlike, models trained using existing single-step adversarial training methods, models trained using proposed method are robust to both single-step and multi-step attacks.

- The proposed single-step adversarial training with regularizer is much faster than SOTA PGD adversarial training [12], and achieves on par results.

Note, that adversarial training with R-FGSM or with both R-FGSM and FGSM samples does not improve the model's robustness against adversarial attacks. Results for these experiments are shown in section 5

The paper is organized as follows: section 2 discusses existing works that are relevant, section 3 introduces the notation followed in the subsequent sections of the paper, section 4 presents the proposed adversarial training method, section 5 hosts the experiments and results, and section 6 concludes the paper.

2. Related works

For defense against adversarial attack multiple methods [18, 4, 1, 11, 5, 3, 26, 22, 21, 25] have been proposed. In this direction, adversarial training method by [4] has shown promising results. Kurakin *et al*. [8] observed that models trained using single-step adversarial training method were susceptible to multi-step adversarial attack. Further, Tramer *et al*. [24] observed these models to be highly susceptible to transfer attacks, and explained this pseudo robustness of the model trained using single-step adversarial training method is due to gradient masking effect i.e., linear approximation of model's loss function is unreliable to generate adversarial sample. Madry *et al*. [12] demonstrated that adversarial training can yield robust models, if perturbation crafted during training maximizes the model's loss and this is achieved by generating adversaries using *Projected Gradient Descent* (PGD) which is an iterative method.

Whereas, in this work we show that it is possible to learn robust models using single-step adversarial training by penalizing the gradient masking effect. To achieve this, we introduce a regularization term in the training loss that penalizes gradient masking effect, and this in turn helps in the inclusion of stronger adversaries that maximizes the training loss. The proposed regularization term causes training loss to increase when the Euclidean distance between logits for FGSM and R-FGSM [24] (random noise is added before computing FGSM sample) adversaries of a clean sample is large.

3. Notations and Terminology

In this section we define the notations followed through out this paper:

- x : clean image from the dataset.
- y_{true} : ground truth label corresponding to the image x.
- f : neural network that maps input image x to the class score.
- θ : parameters of the neural network.
- J : loss function used to train neural network e.g., cross-entropy loss.
- $\nabla_x J$: gradient of loss with respect to input image x
- m : size of training mini-batch.
- ϵ : strength of perturbation/crafted noise added to the clean image.
- x_{fgsm} : potential adversarial sample corresponding to the image x, generated using FGSM.
- x_{rfgsm} : potential adversarial sample corresponding to the image x, generated using R-FGSM.

3.1. Adversarial Sample Generation Methods

In this section we explain methods for generating adversarial samples. All these attacks use L_∞ norm constraint on generated perturbation for perceptual constraints.

Fast Gradient Sign Method (FGSM): Proposed by [4], generates adversarial samples based on the first order approximation of the loss function and via performing simple gradient ascent:

$$x^* = x + \epsilon.sign\big(\nabla_x J(f(x;\theta), y_{true})\big) \tag{1}$$

Random + Fast Gradient Sign Method (R-FGSM): Proposed by [24]. This method adds small random noise before generating adversarial sample using FGSM method.

$$x' = x + \alpha.sign\big(\mathcal{N}(0^d, I^d)\big) \tag{2}$$

$$x^* = x' + (\epsilon - \alpha).sign\big(\nabla_{x'} J(f(x';\theta), y_{true})\big) \tag{3}$$

Iterative Fast Gradient Sign Method (I-FGSM): In this method FGSM is applied in iterative fashion with small step size. In our experiments we use $\alpha = \epsilon/steps$.

$$x^0 = x \tag{4}$$

$$x^{N+1} = x^N + \alpha.sign\big(\nabla_{x^N} J(f(x^N;\theta), y_{true})\big) \tag{5}$$

Projected Gradient Descent (PGD): Proposed by [12], here the perturbation is initialized with a random point within the allowed L_p-norm ball and then I-FGSM is applied with re-projection.

Algorithm 1: Adversarial training of network N with proposed regularization term.

Input:

m = Size of the training mini-batch

$MaxItertion$ = Maximum training iterations

Hyper-parameters: λ

Initialization

Randomly initialize network N

$iteration = 0$

while $iteration \leq MaxItertion$ **do**

  Read minibatch $B = \{x^1, ..., x^m\}$ from training set

  Generate FGSM adversarial samples B_1=$\{x^1_{fgsm}, ..., x^m_{fgsm}\}$ from corresponding clean samples $\{x^1, ..., x^m\}$ using the current state of the network N

  Generate R-FGSM adversarial samples B_2=$\{x^1_{rfgsm}, ..., x^m_{rfgsm}\}$ from corresponding clean samples $\{x^1, ..., x^m\}$ using the current state of the network N

  Make new mini-batch B^*=$\{B_1, B_2\}$

  Forward pass with mini-batch B^*

  Compute $\mathcal{L}oss$ (Eq. 8)

  Backward pass and update model's parameters

  $iteration = iteration + 1$

end

4. Proposed Approach

In this section, first we will explain the criteria for learning robust models [12], followed by the effect of gradient masking during single-step adversarial training, and finally the proposed single-step adversarial training with regularization term.

4.1. Criteria for learning robust models

Madry *et al*. [12] demonstrated that, adversarial training can yield robust models, if adversarial samples included while training maximizes the model's loss. This objective can be formulated as a mini-max optimization problem Eq.(6).

$$\min_{\theta} \rho(\theta), \tag{6}$$

$$where\ \rho(\theta) = E_{(x,y)\in D}\left[\max_{\delta \in S} J\big(f(x+\delta;\theta), y_{true}\big)\right] \tag{7}$$

At each iteration, we need to find a perturbation δ with L_∞ norm constraint ϵ, that maximizes the model's loss and further we need to update the model's parameter (θ) which minimizes this loss. Madry *et al*. solves this inner maximization problem by generating adversarial samples using PGD method (iterative method) with pre-fixed iterations/steps. Iterative methods ensure that generated perturbation will always increase the model's loss, since at each step of the generation process perturbation with small ϵ is added to the image. This is not true when perturbation with high ϵ is added to the image in a single step.

4.2. Effect of gradient masking during single-step adversarial training

In section 5.1, we empirically show that the extent of maximization of loss that is achieved by FGSM (non-iterative method) adversaries during the initial stages of single-step adversarial training, is similar to that achieved by PGD (iterative method) adversaries i.e., the difference between loss on FGSM adversaries and on PGD adversaries is small (see bottom-left plot of Fig. 1 for ϵ=0.3). As training progress, due to gradient masking the ability of FGSM samples to maximize the loss diminishes i.e., the difference between loss on FGSM adversaries and on PGD adversaries becomes large (see bottom-right plot of Fig. 1 for ϵ=0.3). Further, we observe that this difference between loss on FGSM adversaries and on PGD adversaries becomes large, when the Euclidean distance between logits of FGSM sample and R-FGSM samples becomes large (see top plot of Fig. 1). During single-step adversarial training, when model starts to mask the gradient, its decision surface exhibits sharp curvature near the data points [24]. This sharp curvature obfuscates the linear approximation of loss function. Further, adding a small noise to the image causes $\nabla_x J$ (gradient of loss w.r.t image 'x') to change significantly. Then, if the model is exhibiting gradient masking effect, its pre-softmax representation (i.e., logits) for adversarial sample generated using FGSM and R-FGSM would be different (measured in terms of Euclidean distance).

Whereas, during PGD adversarial training we observe that for the entire training duration this Euclidean distance between logits of FGSM samples and R-FGSM samples is small (see top plot of Fig. 2), and also the difference between average loss on FGSM adversaries and on PGD adversaries is small (see bottom plots of Fig. 2 for ϵ=0.3).

Table 1: Architecture of networks used for FGSM and Ensemble Adversarial Training on MNIST dataset.

LeNet+	A	B	C	D
Conv(32,5,5) + Relu	Conv(64,5,5) + Relu	Dropout(0.2)	Conv(128,3,3) + Tanh	{ FC(300) +Relu Dropout(0.5) } × 4
MaxPool(2,2)	Conv(64,5,5) + Relu	Conv(64,8,8) + Relu	MaxPool(2,2)	
Conv(64,5,5) + Relu	Dropout(0.25)	Conv(128,6,6) + Relu	Conv(64,3,3) + Tanh	FC + Softmax
MaxPool(2,2)	FC(128) + Relu	Conv(128,5,5) + Relu	MaxPool(2,2)	
FC(1024) + Relu	Dropout(0.5)	Dropout(0.5)	FC(128) + Relu	
FC + Softmax	FC + Softmax	FC + Softmax	FC + Softmax	

Table 2: Setup used for Ensemble Adversarial Training. For MNIST networks refer table 1.

	Network to be trained	Pre-trained Models
CIFAR-10	ResNet-34(Ensemble A)	ResNet-34, ResNet-18
	ResNet-34(Ensemble B)	ResNet-34, VGG-16
	ResNet-34(Ensemble C)	ResNet-18, VGG-16
MNIST	A(Ensemble A)	A,B,C
	B(Ensemble B)	B, C ,D
	C(Ensemble C)	C, D, A
	D(Ensemble D)	D, A ,B

4.3. Proposed single-step adversarial training with regularization term

In section 4.2, we showed that when the model starts to mask the gradient then the Euclidean distance between logits of FGSM and R-FGSM adversaries of a clean sample becomes large. Based on this observation, we introduce a regularization term in the training loss Eq.(8) in order to mitigate the effect of gradient masking during single-step adversarial training. In Eq.(8), the first term corresponds to classification loss e.g., Cross-Entropy loss, and the second term represents the proposed regularization. During training, if the model starts to mask the gradient then the Euclidean distance between logits of FGSM and R-FGSM adversaries of clean sample increases, this in turn causes the training loss Eq.(8) to increase. This behavior of the proposed regularization prevents the model from masking the gradient. Unlike, existing single-step adversarial training methods, models trained using the proposed method are robust to both single-step and multi-step attacks. Note, that adversarial training with R-FGSM or with both R-FGSM and FGSM samples does not improve the model's robustness against adversarial attacks. Results for these experiments are shown in section 5.

$$\begin{aligned}\mathcal{L}oss =& \frac{1}{m}\sum_{i=1}^{m} J(f(x^i_{fgsm};\theta), y^i_{true}) \\ &+ \lambda \frac{1}{m}\sum_{j=1}^{m} \left\| logits^j_{fgsm} - logits^j_{rfgsm} \right\|_2^2 \end{aligned} \tag{8}$$

5. Experiments

In our experiments we show results on MNIST [9], and CIFAR-10 [7] datasets. We use LeNet+ shown in table 1 for MNIST dataset. For CIFAR-10 dataset, ResNet-34 [6] is used. These networks are trained using SGD with momentum, and for learning rate scheduling step-policy is used. For all the datasets, images are pre-processed to be in [0,1] range. For CIFAR-10 dataset, random crop and horizontal flip are performed for data-augmentation. We follow [12], for the attack perturbation strength (ϵ) and attack parameters. For all attacks, we use L_∞ norm for perceptual constraints.

In order to show the effectiveness of the proposed regularization term, we show results for two ablation experiments. (i) FGSM + R-FGSM adv.: train with mini-batch containing both FGSM and R-FGSM samples. (ii) Proposed adv. with λ=0: training without proposed regularization term.

5.1. Effect of gradient masking during single-step adversarial training

We train LeNet+ shown in table 1 on MNIST dataset using FGSM adversarial training method. During training, we compute the average Euclidean distance between logits of FGSM and R-FGSM adversaries with ϵ=0.3. Figure 1 shows the obtained plot of average Euclidean distance between logits of FGSM and R-FGSM adversaries versus training iteration. From the plot, it can be observed that after 20 iterations ($\times$ 50), the L_2 distance increases to a large value (i.e., L_2 distance is in the range of 300 to 450) rapidly. Whereas, for the model trained using PGD method, the L_2 distance is relatively low (i.e., L_2 distance is in the range of 0 to 1.2) for the entire training duration, shown in Fig. 2.

In order to validate that the increase in the average L_2 distance is due to gradient masking effect, we obtain the plot of average loss of the model on validation set versus perturbation strength (ϵ) of PGD, R-FGSM and FGSM attacks respectively. Bottom-left of Fig. 1 represents this plot obtained at iteration 20 ($\times$ 50) i.e., before L_2 distance increases, and the bottom-right plot of Fig. 1 is obtained at iteration 100 ($\times$ 50) i.e., after the increase in the L_2 dis-

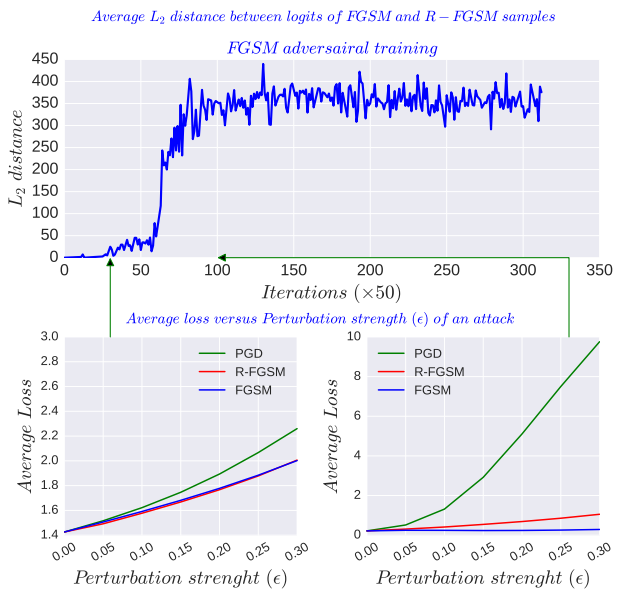

Figure 1: Top: Plot of average L_2 distance between logits of FGSM and R-FGSM adversaries of clean samples, obtained for the model trained on MNIST dataset using FGSM adversarial training method. Observe the increases in the L_2 distance after 20 iterations ($\times$ 50). Bottom: Plot of average loss of the model on val. set versus ϵ of PGD, R-FGSM and FGSM attacks. Bottom-left: Plot obtained at iteration 20 ($\times$ 50), Bottom-right: Plot obtained at iteration 100 ($\times$ 50). Observe the gradient masking effect in the bottom-right plot i.e., for ϵ=0.3 difference between the average loss on PGD and on FGSM samples is large.

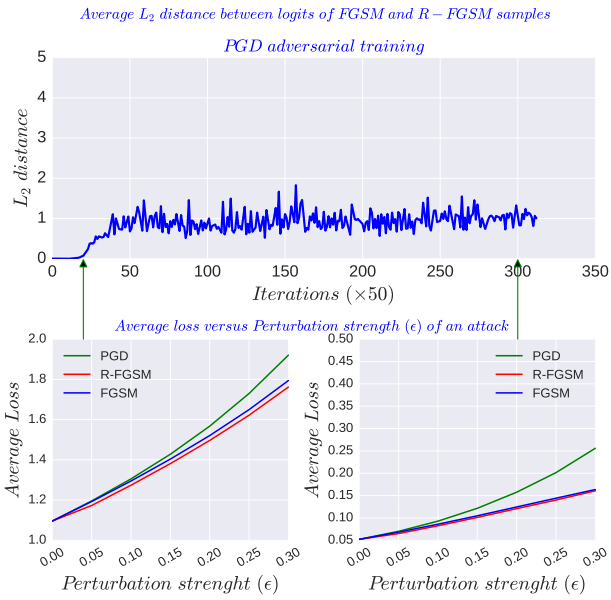

Figure 2: Top: Plot of average L_2 distance between logits of FGSM and R-FGSM adversaries of clean samples, obtained for the model trained on MNIST dataset using PGD adversarial training method. Observe that for the entire training duration average L_2 distance is relatively small. Bottom: Plot of average loss of the model on validation set versus ϵ of PGD, R-FGSM and FGSM attacks. Bottom-left: Plot obtained at iteration 20 ($\times$ 50), Bottom-right: Plot obtained at iteration 300 ($\times$ 50).

tance. From bottom-left plot of Fig. 1, it can observed that for ϵ=0.3 the loss on PGD and on FGSM adversaries are in the same range i.e., difference between average loss on PGD and on FGSM adversaries is small, and this indicates that there is no gradient masking effect. The extent of maximization of loss (Eq. 7) achieved by FGSM (non-iterative method) adversaries during the initial stages of single-step adversarial training, is similar to that achieved by PGD (iterative method) adversaries. Further, from bottom-right plot of Fig. 1, it can be observed that for ϵ=0.3, the difference between average loss on PGD adversaries and on FGSM adversaries is very large indicating gradient masking effect. Whereas, for the model trained using PGD method, this difference between the average losses is small for the entire training duration as shown in bottom plots of Fig. 2.

5.2. Proposed single-step adversarial training with regularization term

MNIST: We train LeNet+ shown in table 1 on MNIST dataset, using the proposed adversarial training method (Alogrithm 1). We use ϵ=0.3 [12] for generating FGSM and R-FGSM adversarial samples. We set hyper-parameter, λ=5. For PGD adversarial training [12], we use adversaries generated by PGD method with $\epsilon = 0.3$, $steps = 40$ and $step\_size = 0.01$. For Ensemble Adversarial Training (EAT) [24], we use the setup shown in table 2. Table 3 compares the performance of models trained using different training methods against white-box attacks. Rows represent training methods and its performance against different attack methods. From table 3 it can be observed that models trained using FGSM and EAT methods fail to defend against multi-step attacks (I-FGSM and PGD), whereas models trained using the proposed method and PGD method are able to defend against multi-step attacks. Table 5 shows the performance of models trained using proposed method and PGD method, against FGSM black-box attack with $\epsilon = 0.3$.

CIFAR-10: We train ResNet-34 on CIFAR-10 dataset, using the proposed adversarial training method (Alogrithm 1). We use ϵ=$8/255$ [12] for generating FGSM and R-FGSM adversarial samples. We set hyper-parameter, λ=5. For PGD adversarial training [12], we use adversaries generated by PGD method with $\epsilon = 8/255$, $steps = 7$ and $step\_size = 2/255$. For Ensemble Adversarial Train-

Table 3: White-Box attack: Classification accuracy (%) of models trained on MNIST dataset using different training methods. For all attacks ϵ=0.3 is used and for PGD attack $step\_size$ is set to 0.01. Note that models trained using PGD and the proposed adversarial training methods are robust to both single-step attack (FGSM) and multi-step attacks (I-FGSM and PGD).

Training Method		Clean	Attack Method			
			FGSM	I-FGSM steps=40	PGD steps=40	PGD steps=100
Normal		99.24	11.65	0.31	0.01	0.00
FGSM Adv.		99.34	89.04	1.19	0.17	0.01
R-FGSM Adv.		93.92	57.06	41.85	29.31	28.76
EAT	A	99.35	83.48	18.75	10.13	3.41
	B	99.31	80.16	48.13	37.85	22.86
	C	99.20	82.48	4.00	1.29	0.08
	D	97.66	56.85	0.87	0.29	0.08
R-FGSM + FGSM Adv.		95.39	61.60	44.62	34.15	33.41
Ablation-Proposed λ=0		97.62	92.52	4.92	1.89	0.17
PGD Adv.		98.41	95.56	92.64	92.08	91.13
Proposed		98.74	95.1	89.91	89.48	87.74

Table 4: White-Box attack: Classification accuracy (%) of models trained on CIFAR-10 dataset using different training methods. For all attacks ϵ=8/255 is used and for PGD attack $step\_size$ is set to 2/255. Note that models trained using PGD and the proposed adversarial training methods are robust to both single-step attack (FGSM) and multi-step attacks (I-FGSM and PGD).

Training Method		Clean	Attack Method			
			FGSM	I-FGSM steps=7	PGD steps=7	PGD steps=20
Normal		91.52	14.00	0.00	0.00	0.00
FGSM Adv.		92.42	98.58	0.09	0.05	0.00
R-FGSM Adv.		79.39	98.64	0.35	0.22	0.02
EAT	A	90.80	82.14	10.56	4.69	0.67
	B	90.43	60.59	32.76	28.9	18.31
	C	90.28	66.49	36.49	29.41	20.24
R-FGSM + FGSM Adv.		80.88	97.92	2.18	1.16	0.22
Ablation-Proposed λ=0		77.00	98.96	0.21	0.06	0.01
PGD Adv.		79.44	53.25	50.53	50.08	47.51
Proposed		80.45	53.14	49.83	49.13	46.07

Table 5: Accuracy (%) of models trained on MNIST dataset for black-box attack. Source models are used to generate FGSM adversarial samples with ϵ=0.3 and tested on target model. Subscript denotes the training method. Here M represents LeNet+

Source Model	Target Model	
	M_{PGD}	$M_{Proposed}$
Model-A	93.70	93.53
Model-B	93.57	93.25

Table 6: Accuracy (%) of models trained on CIFAR-10 dataset for black-box attack. Source models are used to generate FGSM adversarial samples with ϵ=8/255 and tested on target model. Subscript denotes the training method. Here M represents ResNet-34.

Source Model	Target Model	
	M_{PGD}	$M_{Proposed}$
VGG-11	76.09	76.81
VGG-19	77.24	77.92

ing (EAT), we use the setup shown in table 2. Table 4 compares the performance of models trained using different training methods against white-box attacks. For the model trained using FGSM method *label leaking* effect is observed i.e., accuracy of the model on FGSM adversarial set is greater than that on clean set. Further, it can be observed that models trained using FGSM and EAT methods fail to defend against multi-step attacks (I-FGSM and PGD). Although, models trained using the proposed method and PGD method are not fully robust against white-box attacks, they are able to defend against it to an extent. Table 6 shows the performance of models trained using the proposed method and PGD method, against FGSM black-box attack with $\epsilon = 8/255$. Figure 4 shows the plot of average loss on test set vs. perturbation strength (ϵ) of PGD and FGSM attack, obtained for models trained using the proposed method.

Accuracy versus Perturbation strength of PGD attack: In order to verify that gain in the robustness of the model trained using the proposed method is not due to gradient masking effect, we obtain the plot of test-set accuracy of models trained using the proposed method, against PGD adversaries of different perturbation strength (ϵ). Figure 3 shows the obtained plot. It can be observed that for higher perturbation strength model's performance degrades. The purpose of obtaining this plot is to verify that the model's robustness is not due to gradient masking effect. If gradient masking effect is present, model's accuracy does not drop even for the attack with higher perturbation strength (normally for higher perturbation strength, image gets distorted and performance of model should degrade).

5.3. Time Complexity

In this subsection we show the time complexity of different training methods, which are measured in terms of training time per epoch. Table 7 compares the training time per epoch (seconds) for models trained on MNIST and CIFAR-10 datasets respectively, using different training methods. We ran these timing experiments on a machine with NVIDIA Titan Xp GPU, with no other jobs running on it.

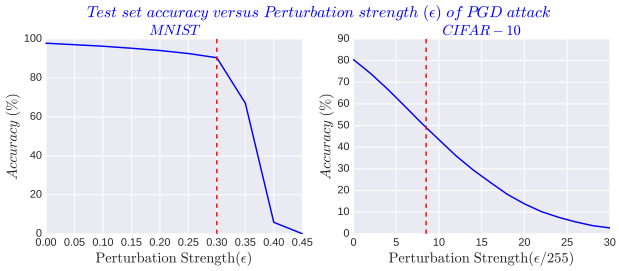

Figure 3: Performance of models trained using the proposed method, against PGD adversaries of different perturbation strength (ϵ). For PGD attack, we set $steps = 40$ for MNIST dataset, and $steps = 7$ for CIFAR-10 dataset. Dashed lines indicates the ϵ used while adversarial training.

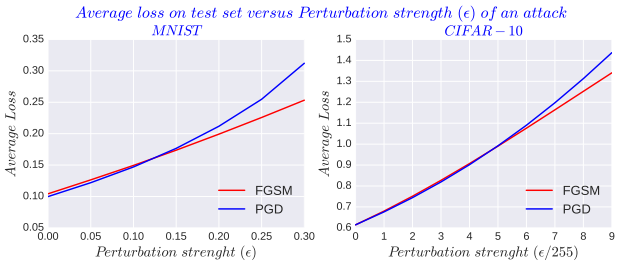

Figure 4: Average loss on test set versus perturbation strength (ϵ) of an attack. Obtained for models trained on MNIST and CIFAR-10 dataset using the proposed method.

Table 7: Comparison of training time per epoch of models trained on MNIST and CIFAR-10 datasets respectively, obtained for different training methods. For PGD adversarial training, steps=40 is used for MNIST dataset and steps=7 is used for CIFAR-10 dataset. † For EAT, training time of pre-trained source models are not considered.

Method	Training time per epoch (sec.)	
	MNIST	CIFAR-10
Normal Training	~ 2.7	~ 31
FGSM Adv. Training	~ 4.1	~ 53
EAT†	~ 5.5	~ 59
PGD Adv. Training	~ 53.0	~ 238
Proposed	~ 8.8	~ 108

6. Conclusion

In this work, we have demonstrated that models trained using single-step adversarial training method can be made robust against adversarial attacks, if gradient masking effect is penalized. We achieved this by introducing regularization term in the training loss, which causes training loss to increase when Euclidean distance between logits of FGSM and R-FGSM adversaries of a clean sample is high. Unlike models trained using existing single-step adversarial training methods, models trained using the proposed adversarial training method are robust to both single-step and multi-step attacks in white-box and black-box settings. Proposed method is faster than state-of-the-art PGD adversarial training method and achieves on par results.

References

[1] Jacob Buckman, Aurko Roy, Colin Raffel, and Ian Goodfellow. Thermometer Encoding: One Hot Way To Resist Adversarial Examples. In *International Conference on Learning Representations (ICLR)*, 2018. 2

[2] Nicholas Carlini and David A. Wagner. Towards Evaluating the Robustness of Neural Networks. In *2017 IEEE Symposium on Security and Privacy, SP 2017, San Jose, CA, USA, May 22-26, 2017*, pages 39–57, 2017. 1

[3] Guneet S. Dhillon, Kamyar Azizzadenesheli, Jeremy D. Bernstein, Jean Kossaifi, Aran Khanna, Zachary C. Lipton, and Animashree Anandkumar. Stochastic Activation Pruning for Robust Adversarial Defense. In *International Conference on Learning Representations (ICLR)*, 2018. 2

[4] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and Harnessing Adversarial Examples. In *International Conference on Learning Representations (ICLR)*, 2015. 1, 2

[5] Chuan Guo, Mayank Rana, Moustapha Cisse, and Laurens van der Maaten. Countering Adversarial Images using Input Transformations. In *International Conference on Learning Representations (ICLR)*, 2018. 2

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 4

[7] Alex Krizhevsky. Learning Multiple Layers of Features from Tiny Images. Technical report, 2009. 4

[8] Alexey Kurakin, Ian J. Goodfellow, and Samy Bengio. Adversarial Machine Learning at Scale. In *International Conference on Learning Representations (ICLR)*, 2017. 1, 2

[9] Yann LeCun. The MNIST database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*. 4

[10] Yanpei Liu, Xinyun Chen, Chang Liu, and Dawn Song. Delving into Transferable Adversarial Examples and Black-box Attacks. In *International Conference on Learning Representations (ICLR)*, 2017. 1

[11] Xingjun Ma, Bo Li, Yisen Wang, Sarah M. Erfani, Sudanthi Wijewickrema, Grant Schoenebeck, Michael E. Houle, Dawn Song, and James Bailey. Characterizing Adversarial Subspaces Using Local Intrinsic Dimensionality. In *International Conference on Learning Representations (ICLR)*, 2018. 2

[12] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Tsipras Dimitris, and Adrian Vladu. Towards Deep Learning Models Resistant to Adversarial Attacks. In *International Conference on Learning Representations (ICLR)*, 2018. 1, 2, 3, 4, 5

[13] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. DeepFool: A Simple and Accurate Method to Fool Deep Neural Networks. In *The IEEE Computer Vision and Pattern Recognition (CVPR)*, 2016. 1

[14] Konda Reddy Mopuri, Aditya Ganeshan, and R. Venkatesh Babu. Generalizable Data-free Objective for Crafting Universal Adversarial perturbations. *In IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 2019. 1

[15] Konda Reddy Mopuri, Utsav Garg, and R. Venkatesh Babu. Fast Feature Fool: A Data Independent Approach to Universal Adversarial Perturbations. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2017. 1

[16] Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z Berkay Celik, and Ananthram Swami. The Limitations of Deep Learning in Adversarial Settings. In *Security and Privacy (EuroS&P), 2016 IEEE European Symposium on*, pages 372–387. IEEE, 2016. 1

[17] Nicolas Papernot, Patrick D. McDaniel, Ian J. Goodfellow, Somesh Jha, Z. Berkay Celik, and Ananthram Swami. Practical Black-Box Attacks against Deep Learning Systems using Adversarial Examples. In *Asia Conference on Computer and Communications Security (ASIACCS)*, 2017. 1

[18] Nicolas Papernot, Patrick D. McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. Distillation as a Defense to Adversarial Perturbations Against Deep Neural Networks. In *IEEE Symposium on Security and Privacy, SP 2016, San Jose, CA, USA, May 22-26, 2016*, pages 582–597, 2016. 2

[19] Konda Reddy Mopuri, Phani Krishna Uppala, and R. Venkatesh Babu. Ask, Acquire, and Attack: Data-free UAP Generation using Class Impressions. In *The European Conference on Computer Vision (ECCV)*, September 2018. 1

[20] Konda Reddy Mopuri, Utkarsh Ojha, Utsav Garg, and R. Venkatesh Babu. NAG: Network for Adversary Generation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 1

[21] Pouya Samangouei, Maya Kabkab, and Rama Chellappa. Defense-GAN: Protecting Classifiers Against Adversarial Attacks Using Generative Models. In *International Conference on Learning Representations (ICLR)*, 2018. 2

[22] Yang Song, Taesup Kim, Sebastian Nowozin, Stefano Ermon, and Nate Kushman. PixelDefend: Leveraging Generative Models to Understand and Defend against Adversarial Examples. In *International Conference on Learning Representations (ICLR)*, 2018. 2

[23] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations (ICLR)*, 2013. 1

[24] Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Dan Boneh, and Patrick McDaniel. Ensemble Adversarial Training: Attacks and Defenses. In *International Conference on Learning Representations (ICLR)*, 2018. 1, 2, 3, 5

[25] B. S. Vivek, Konda Reddy Mopuri, and R. Venkatesh Babu. Gray-box Adversarial Training. In *The European Conference on Computer Vision (ECCV)*, September 2018. 2

[26] Cihang Xie, Jianyu Wang, Zhishuai Zhang, Zhou Ren, and Alan Yuille. Mitigating Adversarial Effects Through Randomization. In *International Conference on Learning Representations (ICLR)*, 2018. 2