The Role of ‘Sign’ and ‘Direction’ of Gradient on the Performance of CNN

Akshay Agarwal[1], Richa Singh[2], Mayank Vatsa[2]
[1]IIIT-Delhi, India; [2]IIT Jodhpur, India
[1]akshaya@iiitd.ac.in; [2]{richa, mvatsa}@iitj.ac.in

Abstract

State-of-the-art deep learning models have achieved superlative performance across multiple computer vision applications such as object recognition, face recognition, and digits/character classification. Most of these models highly rely on the gradient information flows through the network for learning. By utilizing this gradient information, a simple gradient $\mathbf{sign}$ *method based attack is developed to fool the deep learning models. However, the primary concern with this attack is the perceptibility of noise for large degradation in classification accuracy. This research address the question of whether an imperceptible gradient noise can be generated to fool the deep neural networks? For this, the role of* $\mathbf{sign}$ *function in the gradient attack is analyzed. The analysis shows that without-*$\mathbf{sign}$ *function, i.e. gradient magnitude, not only leads to a successful attack mechanism but the noise is also imperceptible to the human observer. Extensive quantitative experiments performed using two convolutional neural networks validate the above observation. For instance, AlexNet architecture yields* 63.54% *accuracy on the CIFAR-10 database which reduces to* 0.0% *and* 26.39% *when* $\mathbf{sign}$ *(i.e., perceptible) and without-*$\mathbf{sign}$ *(i.e., imperceptible) of the gradient is utilized, respectively.*

Further, the role of the direction of the gradient for image manipulation is studied. When an image is manipulated in the positive direction of the gradient, an adversarial image is generated. On the other hand, if the opposite direction of the gradient is utilized for image manipulation, it is observed that the classification error rate of the CNN model is reduced. On AlexNet, the error rate of 36.46% *reduces to* 4.29% *when images of CIFAR-10 are manipulated in the negative direction of the gradient. To explore other enthusiastic results on multiple object databases, including CIFAR-100, fashion-MNIST, and SVHN, please refer to the full paper.*

1. Introduction

Machine learning algorithms, especially deep convolutional neural networks (CNNs), are widely used for various computer vision applications such as biometric identification, object recognition, game theory, and robotics. However, state-of-the-art CNN models are vulnerable towards intelligently crafted minute noise. Goodfellow et al. [8] have shown that manipulating the image pixels through the gradient information can lead to misclassifications. Specifically, $\mathbf{sign}$ of the gradient of an image was added in the image itself for generating an adversarial image. This attack is popularly known as a fast gradient $\mathbf{sign}$ method (FGSM). The major drawback of the FGSM attack is the perceptibility of noise with the increase in the strength parameter of noise. Based on the definition of the FGSM attack, this research explores the importance of the $\mathbf{sign}$ of gradient and direction of manipulation. The key contributions of this research are:

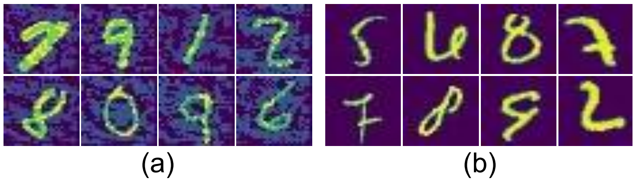


Figure 1: Adversarial examples pertaining to perceptible and imperceptible noise generate on MNIST [16] images. The examples are generated using the (a) $\mathbf{Sign}$ and (b) without $\mathbf{Sign}$ of the gradient computed using **ResNet-18** [11].

1. analyzing the importance of the $\mathbf{sign}$ function in the FGSM attack. To alleviate the problem of perceptibility of noise, a variant of the FGSM attack termed as fast gradient magnitude attack (FGM) is proposed,

2. inspired by the learning of deep models using gradient information, a new defense strategy is proposed to enhance the recognition performance of CNNs. The impact of the direction of the gradient is analyzed to modify the input image for correct classification,

3. extensive experimental evaluation concerning the above two points are performed using AlexNet [14] and ResNet [11] models.

Figure 1 shows the role of $\mathbf{sign}$ and without $\mathbf{sign}$ of the gradient in adversarial examples generation. The noise in the $\mathbf{signed}$ version is clearly perceptible while the adversarial examples generated using magnitude gradient contains imperceptible noise.

2. Related Work

In this section, a brief literature towards improving the object recognition performance in the deep learning era is presented, followed by a brief summary of algorithms for generating adversarial examples.

Starting with LeNet [15], convolutional neural networks (CNNs) have achieved state-of-the-art performances in various computer vision tasks such as object recognition and human identification. A major possible reason for this performance is the availability of large scale databases, including ImageNet [4] and computing resources such as GPUs and cloud computing. In 2012, the huge drop in the error rate on ImageNet had marked the generation of deeper CNN models with better object recognition performance. Some popular deeper networks which are proposed after the introduction of AlexNet [14] in 2012 are VGG-16 [23], GoogLeNet [26], ResNet [11], and DenseNet [12]. These popular architectures consist of 16, 19, 18-201, and 121-161 layers and billions of trainable parameters to optimize. For complex databases, the trend is to increase the number of layers [17] in CNNs for better performance.

It is our intuition that a large amount of data used in the training of these deeper CNNs and optimization of these parameters has led to the biasness towards texture and image parts. Therefore, shuffling of image parts or modification in image texture leads to adversarial attacks [5]. Szegedy et al. [27] have shown generation of adversarial examples (which were earlier getting correctly classified) by solving the box constraint optimization. Later Goodfellow et al. [8] utilized part of the learning algorithm of most of the CNNs to modify the image pixel for possible misclassification. The advancements in generation of adversarial examples has led to new algorithms such as l_2 norm [2] and elastic-net norm [3] minimization based, decision boundary based [19], universal perturbation [18], and hand-crafted attacks [9, 10]. These adversarial algorithms can generate examples that are visually similar to clean/natural examples but can mislead the CNN models. However, without any adversarial noise present/embedded in the natural images, due to redundant, textural or shape information, some clean images also get misclassified (these clean misclassified samples are referred to as negative examples). The details of existing adversarial domain algorithms can also be found in [1, 6, 7, 21, 24, 29].

This research covers both adversarial and negative examples through two studies: (i) the role of $\mathbf{sign}$ function on adversarial examples generation and (ii) the role of direction of gradient to boost the performance of CNN models.

3. Adversarial Examples Generation

Szegedy et al. [27] for the first time demonstrated the vulnerability of deep learning models using the adversarial examples generated by minimizing the following optimization function:

$$min||r||_2 \;\; s.t. \;\; f(x+r) = l; \;\; I_c + r \in [0,1]^m \tag{1}$$

where, r, I_c, l, and $f(\cdot)$ denote the adversarial noise, clean image, target class label, and machine learning classifier, respectively. Further, Goodfellow et al. [8] have generated adversarial examples by utilizing the $\mathbf{sign}$ of the gradient of an image. Let $f : R^m \rightarrow 1....k$ be the deep CNN classifier that predicts the confidence score corresponding to each image; of being in one of the k classes. To generate the optimized scores corresponding to an image, the loss function of the CNN classifier is optimized through gradient descent. The gradient of the loss function concerning the parameters (θ) of the CNN model denoted by $\bigtriangledown J(\theta, I_c, l)$ is computed and the following optimization function is solved to generate the adversarial images:

$$\begin{aligned} f(x + \epsilon \cdot \mathbf{sign}(\bigtriangledown J(\theta, I_c, l))) = l \\ s.t. \; x + \epsilon \cdot \mathbf{sign}(\bigtriangledown J(\theta, I_c, l)) \in [0,1]^m \end{aligned} \tag{2}$$

where, ϵ and $sign(\cdot)$ denote the strength parameter of noise and $\mathbf{sign}$ function, respectively. The gradient $\mathbf{sign}$ method aims to increase the loss of the classifier, and ϵ helps in controlling the l_∞ norm of the noise. The $\mathbf{sign}$ function in the original formulation (i.e., [8]) ensures that the magnitude of the loss is maximized. Therefore, in this research, first, the role of $\mathbf{sign}$ function in adversarial example generation is analyzed on multiple databases and CNN models.

Adversarial examples are generated with and without the $\mathbf{sign}$ function using the following two optimizations, respectively:

$$FGSM = I_c + \epsilon \cdot \rho_1 \; where, \; \rho_1 = \mathbf{sign}(\bigtriangledown J(\theta, I_c, l)) \tag{3}$$

$$FGM = I_c + \epsilon \cdot \rho_2 \; where, \; \rho_2 = \bigtriangledown J(\theta, I_c, l) \tag{4}$$

where, I_c, $FGSM$, and FGM represent the clean image, adversarial image through the $\mathbf{signed}$ gradient, and adversarial example through gradient only, respectively.

As shown in Figure 2, the gradient information of an image mainly consists of the edge information. For example, by looking at the first and second images (from right) in the first row, we can recognize the digits (4 and 7) used to compute the gradient images. On the other hand, the $\mathbf{sign}$ of the gradient depicts no such information. $\mathbf{Sign}$ information is just a random noise added to the original image

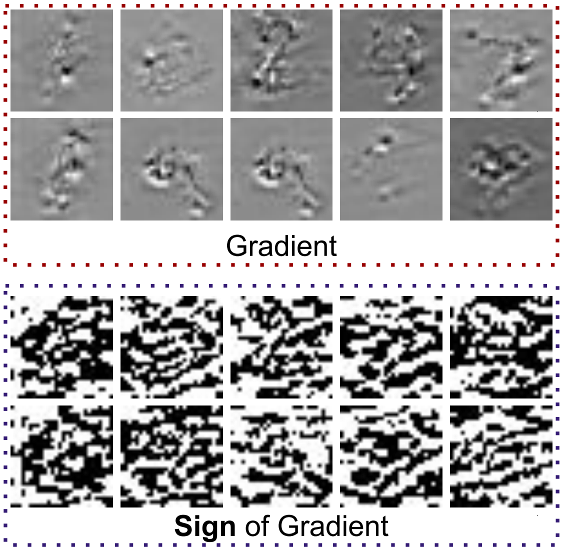


Figure 2: Illustration of the gradient and **signed** version of gradient on the MNIST images using **ResNet-18**.

for misclassification. Another interesting observation after the addition of a **sign** of gradient is the perceptibility of the noise when combined back with higher strength value. We have observed that when only gradient information is added in the image, the adversarial attack is successful without being perceptible to human observers. The observation can be found in Figure 3 on the MNIST database[1]. It is clear from Figure 3 that the noise (with $\epsilon = 2.0$ or higher) is visible for a successful attack in case of the **sign** of gradient. Whereas, when gradient magnitude (FGM) is added, even with the same strength parameter, the attack is successful without being perceptible.

4. The Impact of Sign of Gradient on Classification Loss

In this section, the effect of adversarial attack is evaluated with and without the **sign** function. We first summarize the databases and CNN models used for evaluation followed by the results and analysis. The decrease in classification performance corresponding to each database is reported using AlexNet, followed by the experimental results using ResNet-18.

4.1. Databases and CNN Models

Databases: To evaluate the classification performance of CNN models in presence of the proposed research, multiple databases namely (i) MNIST [16], (ii) CIFAR-10 [13], (iii) CIFAR-100 [13], (iv) SVHN [20], and (v) Fashion MNIST (F-MNIST) [28] are used.

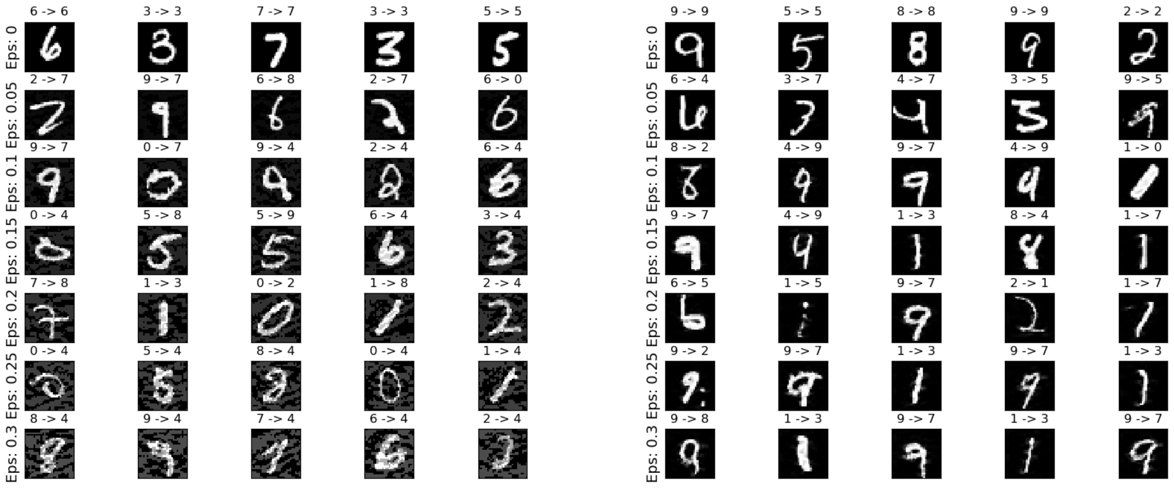
CNNs: Experiments are performed with two state-of-the-art CNN models namely AlexNet [14] and ResNet-18 [11]. On each database, the corresponding model is trained on the pre-defined training set only (no extra training set is utilized, even if provided, such as provided with SVHN).

4.1.1 Adversarial Examples with AlexNet

Table 1 shows the results on clean and adversarial examples generated using AlexNet on each database. Apart from the visualization of gradient and **sign** of gradient, accuracies of deep networks on adversarial examples are also reported using the original formulation and its variant (shown in equations 3 and 4) of gradient attack. When clean test images of MNIST and F-MNIST are used, the model yields 98.23% and 87.63% accuracy, respectively. In the case of adversarial examples generated using the **sign** of gradient, a large drop in accuracy is observed due to intense noise. On the F-MNIST database, when **signed** gradient (FGSM) adversarial examples are generated with 0.20 strength value, 86.71% drop in recognition accuracy (i.e. from 87.63% → 0.92%) is observed in comparison to 12.73% (i.e. from 87.63% → 74.90%) drop when only the gradient magnitude (FGM) is used for generating adversarial examples. The drop in recognition performance increases by increasing the value of the strength parameter for the FGM attack. For example, on F-MNIST, ϵ value of 0.9 leads to 24.22% recognition accuracy. However, even with a large strength value of noise, the adversarial noise goes unnoticed (Figure 3 (b)). In this research, we have observed that for the drop in recognition performance, lower strength value is required for the FGSM while higher strength value is required for the FGM attack. Another observation is the perceptibility of noise, which remains imperceptible when added using FGM as compared to FGSM. A possible reason for this can be obtained from the visualization of gradient and **sign** of gradient (Figure 2). Figure 4 shows the adversarial examples generated using **sign** and magnitude of gradient. To the best of our knowledge, this is the first work with such findings related to adversarial examples corresponding to the direction and **sign** of gradient.

On the CIFAR-10 database, AlexNet yields 63.54% accuracy on the test set. However, the performance drops down to 26.39% with imperceptible adversarial noise. The vulnerability of the model is also noticed with a **signed** gradient attack, where the recognition performance drops down to 0.0% on the CIFAR-10 database when the strength is high and the noise is visible. The adversarial examples generated on CIFAR-100 using ρ_1 (**sign**) and ρ_2 (magnitude) show similar vulnerability. When the adversarial examples on SVHN are generated using ρ_1 and ρ_2 with

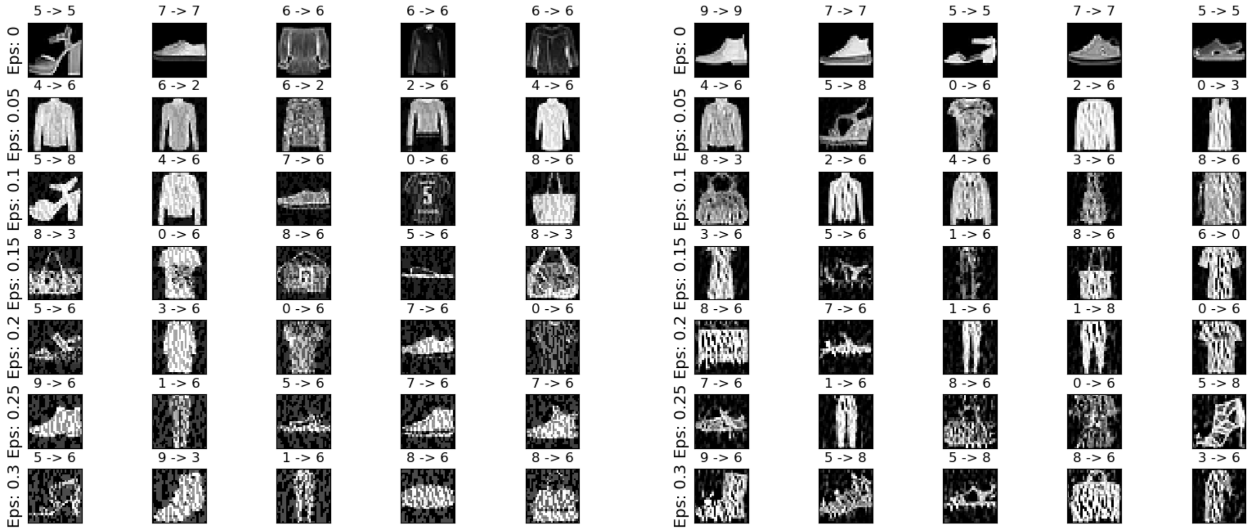
[1] The MNIST database contains gray-scale black-white images, which makes the display of random noise easy in comparison to complex color images of other databases such as CIFAR and SVHN. However, similar phenomena are observed on these color databases.



(a) Illustrating the effect of the existing FGSM attack.

(b) Illustrating the effect of proposed FGM attack.

Figure 3: Adversarial examples generated using **ResNet-18** [11] by gradient magnitude and **sign** of gradient computed over each image of MNIST database. In $a \rightarrow b$, a represents the initial predicted true label of an image and b represents the misclassified label after attack. Eps: 0 represents when no attack is performed. Images are randomly selected i.e., they are not cherry picked.



(a) Illustrating the effect of the existing FGSM attack.

(b) Illustrating the effect of proposed FGM attack.

Figure 4: Adversarial examples generated using **ResNet-18** [11] by gradient magnitude and **sign** of gradient computed over each image of F-MNIST database. In $a \rightarrow b$, a represents the initial predicted true label of an image and b represents the misclassified label after attack. Eps: 0 represents when no attack is performed. Images are randomly selected i.e., they are not cherry picked.

$\epsilon = 0.3$, the accuracy of the CNN model drops by more than 73% and 28% respectively. The decline in recognition performance is lower in the case of ρ_2, but the imperceptibility of the adversarial noise is the most significant advantage.

4.1.2 Adversarial Examples with ResNet-18

The AlexNet model (which contains 5 convolutional layers each followed by max-pooling and ReLU non-linearity) has not performed well on complex databases such as CIFAR and SVHN. Therefore, another state-of-the-art model, namely ResNet-18 [11], is also used for adversarial example generation using these databases. ResNet-18 is trained for 10 epochs with an initial learning rate set to 0.001 and batch

Table 1: Adversarial examples on CIFAR-10, CIFAR-100, MNIST, and F-MNIST database via **sign** and without-**sign** variation of gradient using **AlexNet**.

Accuracy (%)	Epsilon (ϵ)	CIFAR-10		CIFAR-100		MNIST		F-MNIST	
		FGSM	FGM	FGSM	FGM	FGSM	FGM	FGSM	FGM
Natural	0.0	**63.54**		**25.23**		**98.23**		**87.63**	
After Attack	0.05	00.20	42.70	00.77	20.36	84.58	96.95	36.33	81.67
	0.10	00.00	36.20	00.90	17.71	49.34	96.45	11.28	78.65
	0.15	00.00	32.32	00.10	14.93	19.32	96.12	02.93	76.68
	0.20	00.00	29.69	00.10	13.38	10.97	95.76	00.92	74.90
	0.25	00.00	27.76	00.10	12.09	08.50	95.52	00.59	73.29
	0.30	**00.00**	**26.39**	**00.00**	**11.02**	**07.30**	**95.18**	**00.48**	**72.13**

Table 2: Adversarial examples on CIFAR-10, CIFAR-100, MNIST, and F-MNIST database via **sign** and without-**sign** variation of gradient using **ResNet-18**.

Accuracy (%)	Epsilon (ϵ)	CIFAR-10		CIFAR-100		MNIST		F-MNIST	
		FGSM	FGM	FGSM	FGM	FGSM	FGM	FGSM	FGM
Natural	0.0	**83.43**		**53.37**		**99.35**		**93.52**	
After Attack	0.05	**01.94**	04.00	00.36	01.17	95.98	98.10	04.30	17.48
	0.10	02.43	02.70	00.29	00.56	72.15	95.94	**00.93**	03.63
	0.15	02.95	**02.66**	**00.20**	00.39	32.01	91.96	01.28	**02.47**
	0.20	02.92	02.76	00.26	00.32	13.92	86.31	01.85	02.78
	0.25	02.93	03.00	00.31	**00.28**	07.86	80.07	02.24	03.18
	0.30	03.15	03.01	00.31	00.34	**05.99**	**73.65**	02.44	03.65

size to 32. The ResNet-18 model shows improvement in recognition performance of 19.89%, 28.14%, and 18.14% on the CIFAR-10, CIFAR-100, and SVHN databases in comparison to AlexNet, respectively.

Attack on ResNet-18: Similar to AlexNet, when adversarial examples generated using the gradient of the model are presented as input, the performance of the model decreases. Results on each database are reported in Table 2. It is interesting to note that in comparison to Alexnet, the magnitude variant (FGM) also shows a huge drop in the performance of the ResNet-18 model. On the Alexnet model, when a gradient magnitude attack (FGM) is used the accuracy on CIFAR-10 drops by 31.22%, whereas, the performance on ResNet-18 drops by 80.77% with the same strength value of 0.15. Similar trends are observed on CIFAR-100 and SVHN databases, where the deep ResNet model shows higher vulnerability towards both **signed** (FGSM) and gradient magnitude (FGM) attack. Hence, the proposed quantitative analysis, which is different from the original formulation of the FGSM attack, shows that CNN models are also vulnerable towards gradient magnitude information. However, the advantage of an gradient magnitude attack (i.e., FGM) is the imperceptibility of the noise added to the images.

In the experiments performed across multiple databases, CNNs, and adversarial attacks, we have observed that apart from the visibility of the noise with increasing strength, the recognition performance on some databases improves with specific noise strength parameters. For example, on the CIFAR-10 database with the ResNet-18 model, the adversarial accuracy increases after the strength of 0.15.

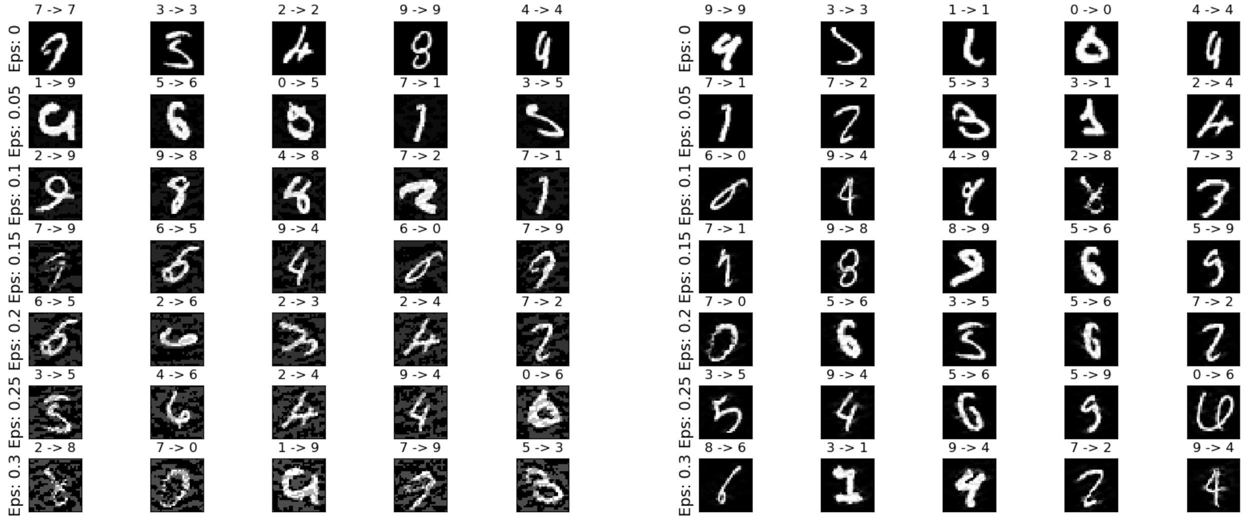
5. Cooperative Examples Generation

It has been argued recently that adversarial examples lie in the data manifolds and represent the low probability region where no training data was present [22, 25]. We assert that a similar low probability argument might be valid for the misclassified examples[2]. Therefore, there might be a way to traverse in this misclassified data manifold that can significantly reduce the sensitivity of the network to increase its performance. We hypothesize that the system might not resist linear perturbation in reverse direction (similar to CNN parameter learning). Hence, we can improve the confidence in the negative examples[3].

Let $C : R^m \rightarrow 1...k$ is the CNN classifier with parameters Θ, which maps the input to the probabilities values corresponding to each of the k categories. The classifier is associated with the loss function $\jmath$ to its parameters, input, and output probabilities. By linearizing the loss function, the perturbation can be found using $\kappa = \bigtriangledown_I \jmath(\Theta, I, y)$. Reiteratively, the weight update of neural works:

[2] Clean images of the classes network have seen, but might be from different viewpoints and in environmental conditions. Therefore gets misclassified.

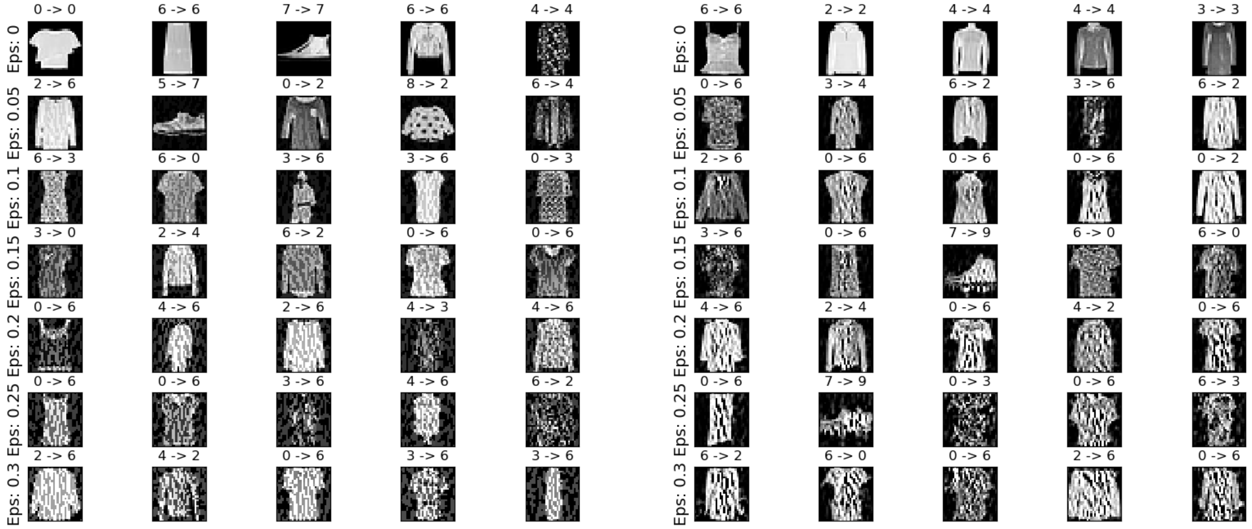
[3] Positive and negative examples are the clean images that are correctly classified and misclassified by CNN, respectively.



(a) Illustrating the effect of FGSM defense.

(b) Illustrating the effect of FGM defense.

Figure 5: Cooperative examples generated using **ResNet-18** via gradient magnitude and **sign** of gradient computed over each image of the MNIST database. In $a \to b$, a represents the initial wrong predicted label of a negative image and b represents the predicted true label after defense. Eps: 0 represents when no defense is applied. The images are randomly selected not cherry picked.



(a) Illustrating the effect of FGSM defense.

(b) Illustrating the effect of FGM defense.

Figure 6: Cooperative examples generated using **ResNet-18** via gradient magnitude and **sign** of gradient computed over each image of the F-MNIST database. In $a \to b$, a represents the initial wrong predicted label of a negative image and b represents the predicted true label after defense. Eps: 0 represents when no defense is applied. The images are randomly selected not cherry picked.

$$\Theta = \Theta - \eta \frac{\partial(\Theta, x, y)}{\partial \Theta} \tag{5}$$

In the above equation for the fixed image x and true label y, parameters Θ are optimized in the negative direction of the gradient. The motivation of going in a negative direction (i.e., subtract the gradient) is to **minimize the loss** through small modification in Θ. In the case of adversarial examples, in place of optimizing the parameter Θ, input image x is optimized itself by going in the direction of the gradient (i.e., add the gradient). The adversarial formulation where Θ and y are fixed is defined below:

$$x = x + \eta \frac{\partial(\Theta, x, y)}{\partial x} \tag{6}$$

Here the motivation is to **maximize the loss** by making small changes in input. Based on the difference between the above two equations, i.e., arithmetic operation ($+$ or $-$), we question whether this gradient optimization can be applied to optimize the input so that the loss is minimized and images started getting classified correctly.

The cooperative[4] examples are generated by solving the following optimization on negative examples:

$$D - FGSM = I - \epsilon \cdot \mathbf{sign}(\kappa), \quad s.t., \quad I' \in [0,1]^m \quad (7)$$

The above defense process is referred to as defended-FGSM (D-FGSM). The cooperative examples can be defined as the misclassified images (initially) that are correctly classified after the proposed optimization.

A variant of the above optimization can be derived using the following equation:

$$D - FGM = I - \epsilon \cdot \kappa, \quad s.t., \quad I' \in [0,1]^m \qquad (8)$$

where I is the negative example and I' is the cooperative example. ϵ represents the strength of linear perturbation on negative examples. $\bigtriangledown$ is the gradient corresponding to parameters (θ), input (I), and output (y) respectively.

Based on the intuition of the above-defined optimization function, experiments discussed in the following section are performed to reduce the classification error of the CNN models.

6. The Role of Direction of Gradient in Classification Improvement

The impact on the classification performance of AlexNet and ResNet-18 by manipulating images using the opposite gradient direction is discussed below.

6.1. AlexNet

The breakthrough CNN model in computer vision proposed by Krizhevsky et al. [14] incurs some natural error on object recognition databases. The model incurs 1.77%, 12.37%, 36.46%, 74.77%, and 23.53% natural error on MNIST, F-MNIST, CIFAR-10, CIFAR-100, and SVHN database respectively. On the MNIST database, the error reduces to 0.25% after the implementation of the proposed cooperative optimization (equations 7 and 8). To achieve a similar reduction, higher strength (i.e., 1.0 or 1.5) is applied for D-FGM in comparison to lower strength (i.e., 0.5) for D-FGSM. On the F-MNIST database, the proposed formulation reduces the error rate to 1.75% (Table 3), which is 10.62% less than the natural error caused by the model. However, it is observed that the error starts increasing with the strength parameter. The reason might be the increase in perceptible noise.

For experiments are performed with colored and complex databases such as CIFAR-10 and CIFAR-100, even the lower value of the strength parameter shows a higher reduction in error rates. In the case of grayscale and relatively easy databases such as MNIST and F-MNIST, we have observed that a minimum strength value of 0.05 is required for maximum reduction. However, lower costs, such as 0.02 and 0.04, show the maximum drawdown on the CIFAR-10 and CIFAR-100 database, respectively. It may be due to the richness of texture and edge information in these databases, which may get corrupted with a higher level of noise. On CIFAR-10 and CIFAR-100 databases, a massive reduction in error rate is observed with a value of 32.17% and 35.18%, respectively. Similarly, on the SVHN database, the proposed formulation can reduce the negative examples by 16.84% with a 0.02 strength value. The results of defense, as discussed above, are listed in Table 3 on MNIST, F-MNIST, CIFAR-10, and CIFAR-100 databases.

6.2. ResNet-18

The defense optimization shown in Equations 7 and 8 are also able to reduce the error rate of the ResNet-18 model. The initial model yields 16.57% and 46.63% error on the CIFAR-10 and CIFAR-100 databases, respectively. The error on CIFAR-10 reduces to 0.55% ($\epsilon = 0.02$) and 0.75% ($\epsilon = 0.01$) when lower strength cooperative noise is applied on the negative examples. On applying the optimization to the negative examples of CIFAR-100, the model shows a significant reduction of 42.43% in the error rate with a $\mathbf{signed}$ gradient. When ResNet-18 with negative gradient direction is used on the SVHN database, the negative examples error rate reaches 0.38%, whereas, the lowest error rate achieved by the AlexNet model is 6.69%. Table 4 shows the reductions in error rate on MNIST and CIFAR database with the ResNet-18 model.

Figure 5 shows the transfer of negative examples into cooperative examples using the optimization defined in equations 7 and 8. For example, in case of $\mathbf{signed}$ defense (i.e., Figure 5 (a)), the images of 9, 6, and 5 are initially misclassifed into 1, 5, and 3 respectively. On applying cooperative noise to these images, the image gets correctly classified into their respective true classes. Similarly, in case of without $\mathbf{sign}$ (i.e., Figure 5 (b)), without even being perceptible in the last row of higher strength, the images of 6, 4, and 1 that were previously misclassified as 8, 3, and 9, respectively, get classified correctly. Figure 6 demonstrates similar results on the F-MNIST database.

[4] the examples which were initially misclassified by the CNN but later correctly classified due to the proposed optimization

Table 3: Cooperative examples on CIFAR-10, CIFAR-100, MNIST, and Fashion-MNIST databases via **sign** and without-**sign** variation of gradient using **AlexNet**. The performance is reported in terms of error, hence, the lower the value, the better it is for a successful defense.

Error (%)	Epsilon (ϵ)	CIFAR-10		CIFAR-100		MNIST		F-MNIST	
		Defense		Defense		Defense		Defense	
		D-FGSM	D-FGM	D-FGSM	D-FGM	D-FGSM	D-FGM	D-FGSM	D-FGM
Natural	0.0	**36.46**		**74.77**		**1.77**		**12.37**	
After Defense	0.05	**5.17**	7.51	**40.57**	67.59	**0.25**	0.44	1.96	5.34
	0.10	6.87	4.67	50.68	60.88	0.27	**0.25**	**1.75**	3.36
	0.15	8.18	**4.56**	57.58	55.81	0.32	**0.25**	2.12	2.50
	0.20	9.21	4.74	61.84	52.24	0.40	0.29	2.54	2.29
	0.25	10.33	4.98	64.73	49.38	0.50	0.31	3.00	**2.07**
	0.30	11.42	5.30	67.02	**47.42**	0.58	0.34	3.57	2.14

For **signed** (i.e., S) defense, on CIFAR-10 and CIFAR-100 lowest error values i.e., 4.29% and 39.49% are achieved by lower epsilon values 0.02 and 0.04, respectively.

Table 4: Cooperative examples on CIFAR-10, CIFAR-100, MNIST, and F-MNIST databases via **sign** and without-**sign** gradient using **ResNet-18**. The performance is reported in terms of error, hence, the lower the value, the better it is for a successful defense.

Error (%)	Epsilon (ϵ)	CIFAR-10		CIFAR-100		MNIST		F-MNIST	
		Defense		Defense		Defense		Defense	
		D-FGSM	D-FGM	D-FGSM	D-FGM	D-FGSM	D-FGM	D-FGSM	D-FGM
Natural	0.0	**16.57**		**46.63**		**0.65**		**6.48**	
After Defense	0.01	0.57	**0.75**	7.42	8.94	0.49	0.53	1.00	1.06
	0.02	**0.55**	1.21	**4.20**	**8.07**	0.37	0.49	0.43	0.85
	0.03	0.98	1.90	4.90	10.57	0.24	0.39	**0.36**	**0.77**
	0.04	1.98	2.75	7.09	13.95	0.14	0.34	**0.36**	0.81
	0.05	3.25	3.56	10.47	17.17	0.11	0.28	0.62	0.94
	0.10	8.20	6.75	27.61	28.69	**0.05**	0.14	1.29	1.61
	0.15	10.27	8.78	37.04	34.61	0.11	0.07	2.13	2.17
	0.20	11.07	10.26	41.82	38.26	0.21	**0.05**	2.90	2.76
	0.25	11.57	11.11	44.09	40.38	0.30	**0.05**	3.41	3.21
	0.30	12.14	11.83	45.25	41.84	0.38	**0.05**	3.74	3.51

7. Conclusion

This research analyzes the role of **sign** function in adversarial example generation and the role of the direction of gradient manipulation towards classification performance. The experiments on different databases showcase that with and without **sign** function, the classification performance of CNN models decreases. However, no free lunch theorem holds on the **sign** formulation, i.e., "higher reduction has not been achieved with the perceptibility of the noise". On the other hand, the performance reduction without the **sign** is approximately the same on complex databases (i.e., CIFAR and SVHN) when a deep CNN model is used, i.e., ResNet-18, but the noise remains imperceptible even with high strength value. In the second study, we have observed that if the images are manipulated in the opposite direction of the gradient, the classification error rates reduce drastically. In the future, the adversarial examples related to **sign** and direction of gradient can be explored to increase the robustness of CNNs.

8. Acknowledgement

A. Agarwal was partially supported by the Visvesvaraya PhD Fellowship. M. Vatsa is partially supported through the Swarnajayanti Fellowship by the Government of India.

References

[1] A. Agarwal, R. Singh, M. Vatsa, and N. Ratha. Are image-agnostic universal adversarial perturbations for face recognition difficult to detect? *IEEE International Conference on Biometrics: Theory, Applications, and Systems*, pages 1–7, 2018.

[2] N. Carlini and D. Wagner. Towards evaluating the robustness of neural networks. In *IEEE Symposium on Security and Privacy*, pages 39–57, 2017.

[3] P. Chen, Y. Sharma, H. Zhang, J. Yi, and C. Hsieh. EAD: elastic-net attacks to deep neural networks via adversarial examples. In *AAAI Conference on Artificial Intelligence*, 2018.

[4] J. Deng, W. Dong, R. Socher, L. Li, and and. Imagenet: A large-scale hierarchical image database. *IEEE Computer Vision and Pattern Recognition, 2009*, pages 710–719, 2009.

[5] L. A Gatys, A. S Ecker, and M. Bethge. Texture and art with deep neural networks. *Current opinion in neurobiology*, 46:178–186, 2017.

[6] A. Goel, A. Agarwal, M. Vatsa, R. Singh, and N. Ratha. DeepRing: Protecting deep neural network with blockchain. *IEEE Computer Vision and Pattern Recognition Workshop*, 2019.

[7] A. Goel, A. Agarwal, M. Vatsa, R. Singh, and N. Ratha. Securing CNN model and biometric template using blockchain. *IEEE International Conference on Biometrics: Theory, Applications, and Systems*, pages 1–6, 2019.

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

[9] G. Goswami, A. Agarwal, N. Ratha, R. Singh, and M. Vatsa. Detecting and mitigating adversarial perturbations for robust face recognition. *International Journal of Computer Vision*, 127(6-7):719–742, 2019.

[10] G. Goswami, N. Ratha, A. Agarwal, R. Singh, and M. Vatsa. Unravelling robustness of deep learning based face recognition against adversarial attacks. In *AAAI Conference on Artificial Intelligence*, pages 6829–6836, 2018.

[11] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[12] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q Weinberger. Densely connected convolutional networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4700–4708, 2017.

[13] A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

[14] A. Krizhevsky, I. Sutskever, and G. E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 1097–1105, 2012.

[15] Y. LeCun, B. Boser, J. S Denker, D. Henderson, R. E Howard, W. Hubbard, and L. D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.

[16] Y. LeCun, C. Cortes, and CJ Burges. Mnist handwritten digit database. *AT&T Labs [Online]. Available: http://yann. lecun. com/exdb/mnist*, 2:18, 2010.

[17] M. Lin, Q. Chen, and S. Yan. Network in network. *arXiv preprint arXiv:1312.4400*, 2013.

[18] S. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard. Universal adversarial perturbations. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1765–1773, 2017.

[19] S. Moosavi-Dezfooli, A. Fawzi, and P. Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2574–2582, 2016.

[20] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y Ng. Reading digits in natural images with unsupervised feature learning. *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2011.

[21] K. Ren, T. Zheng, Z. Qin, and X. Liu. Adversarial attacks and defenses in deep learning. *Engineering*, pages 1–15, 2020.

[22] P. Samangouei, M. Kabkab, and R. Chellappa. Defense-gan: Protecting classifiers against adversarial attacks using generative models. *International Conference on Learning Representations*, 2018.

[23] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[24] R. Singh, A. Agarwal, M. Singh, S. Nagpal, and M. Vatsa. On the robustness of face recognition algorithms against attacks and bias. *AAAI Conference on Artificial Intelligence Senior Member Track*, 2020.

[25] Y. Song, T. Kim, S. Nowozin, S. Ermon, and N. Kushman. Pixeldefend: Leveraging generative models to understand and defend against adversarial examples. *International Conference on Learning Representations*, 2018.

[26] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015.

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

[28] H Xiao, K. Rasul, and R. Vollgraf. Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

[29] X. Yuan, P. He, Q. Zhu, and X. Li. Adversarial examples: Attacks and defenses for deep learning. *IEEE Transactions on Neural Networks and Learning Systems*, 30(9):2805–2824, 2019.