Adversarial Attack on Machine Learning Models

V. Sahaya Sakila, Sandeep M, Praveen Hari Krishna N

Abstract—Machine Learning (ML) models are applied in a variety of tasks such as network intrusion detection or malware classification. Yet, these models are vulnerable to a class of malicious inputs known as adversarial examples. These are slightly perturbed inputs that are classified incorrectly by the ML model. The mitigation of these adversarial inputs remains an open problem. As a step towards understanding adversarial examples, we show that they are not drawn from the same distribution than the original data, and can thus be detected using statistical tests. Using this knowledge, we introduce a complimentary approach to identify specific inputs that are adversarial. Specifically, we augment our ML model with an additional output, in which the model is trained to classify all adversarial inputs.

Keywords—Adversarial attacks, Generative Adversarial Network, Robust Classification.

I. INTRODUCTION

Deep neural networks(DNN) are highly sensitive to very small well-designed perturbations of the inputs, known as adversarial perturbations, which lead to misclassifications of these perturbed inputs. Think about pictures for instance of the information. A crude picture, which is effectively grouped by the neural system, can be adjusted in a little (and regularly intangible to our vision) way, with the goal that the subsequent distorted picture is named an alternate class [1]. A related issue is that Deep Neural Network characterize certain pictures as having a place with some class, in spite of the fact that they are unrecognizable to people as models of any class [2]. These perturbated or tricking pictures relate to patches of the picture space that have a little estimation of the target work utilized in preparing and are situated far from any of the preparation information. These perceptions has tested the honesty of Deep Neural Network order and has prompted a conclusion that their expectations are deceitful, however comparable issues are shared by numerous other AI strategies, for example, bolster vector machines, calculated relapse, K-closest neighbors and others further examination have appeared both antagonistic pictures and silly pictures can be exchanged between a wide range of models having unmistakable designs, distinctive hyper parameters, and even prepared on various preparing sets [1, 2]. in addition, a portion of the ill-disposed and perturbated pictures can be exchanged between different arrangement of models in AI [3]. this clears path for a potential ill-disposed assault, when a programmer can prepare his very own model and make a lot of pictures that are misclassified by it, and after that send this arrangement of contributions against another unfortunate casualty model, which will likewise misclassify them. every now and again they don't require any interior information of the injured individual model. signifying a security issue, transferability recommends that different computational models adapt fundamentally the same as portrayals of the information. it likewise demonstrates that so as to address these issues, one may need to configuration preparing calculations that become familiar with an altogether different portrayal of the information contrasted with the current techniques. a perfect answer for these issues ought to be a calculation that relegates little estimations of the target work just to those territories of the picture space that are unmistakable by people as pictures of the relating class. it additionally needs a considerable and conspicuous by people twisting of the underlying accurately ordered picture towards an alternate target class before the name changes. notwithstanding a significant measure of work on these issues, no calculation have been distinguished yet which satisfies these necessities and in the meantime is aggressive to best in class calculations as far as arrangement exactness. it is likewise suggested that the thick cooperative memory () models with larger amount communications in the vitality work learn portrayals of the information, which unequivocally rely upon the intensity of the collaboration vertex. this system extricates traits structure the present information for little estimations of this power, however as the intensity of the cooperation vertex is expanded there is a slow move to a structure based portrayal, the two outrageous routines of example acknowledgment is known in subjective brain science. phenominally, there is a wide scope of forces of the vitality work, for which the portrayal of the information is now in the model routine, yet the exactness of order is as yet focused to the calculations dependent on DNN with results this demonstrates the dam models may carry on in all respects distinctively in correlation with the standard techniques utilized in profound learning as for antagonistic distortions. in this paper we report three fundamental outcomes. in the first place, utilizing an inclination not too bad in the pixel space, a lot of "perturbated" pictures is developed which compares to the minima of the target work utilized in preparing. this is done on the mnist dataset of manually written characters utilizing distinctive estimations of the intensity of the cooperation vertex, which is meant by n. for little estimations of the power n these pictures really

Revised Manuscript Received on April 12, 2019.

V. SahayaSakila, Assistant Professor Dept of Computer Science and Engg. SRM Institute of Science and Technology Ramapuram, Chennai, Tamil Nadu, India.(shylia1992lipna@gmail.com)

Sandeep M, Dept of Computer Science and Engg. SRM Institute of Science and Technology Ramapuram, Chennai, Tamil Nadu, India. (sandeepmurugadas@gmail.com)

Praveen Hari Krishna N, Dept. Of Computer Science and Engg.SRM Institute of Science and Technology Ramapuram, Chennai, Tamil Nadu, India.(praveenharikrishna@gmail.com)

look like clamor, and don't have any huge substance for human vision. by and by, as the intensity of the communication vertex is expanded the pictures step by step become not so much dotted but rather more altogether significant. at the point when n ≈ 20...30, these pictures are never again garbage by any means. they speak to conceivable pictures of transcribed characters that could have been created by a human. also, beginning from clean pictures from the dataset, a lot of ill-disposed pictures are built so that each picture is set precisely on the choice limit between two name classes. for little powers n these pictures seems to be similar to the underlying clean picture with a tad of spotted commotion included, however are misclassified by the neural system. in any case, as the intensity of the association vertex is expanded these ill-disposed pictures become less and less like the underlying picture. with extremely substantial forces these ill-disposed pictures look either like a transformed picture of two digits (the underlying clean picture and another digit from the class that the disfigurement targets), or the underlying digit superimposed on an "apparition" picture from the objective class.

II. BACKGROUNDS

A.Adversarial Training

Introducing a rational explanation for the adversarial example, and a fast technique to generate adversarial perturbation. It is observed that adversarial examples exist because models are too linear. The suggested a fast gradient sign method such that.

$$\eta = E\ \text{sign}(\nabla x J(\theta, x, y)) \quad (1)$$

where η is an adversarial perturbation, θ denotes parameters of the network, x is the input with the label y. Note that J(θ,x,y) denotes the cost function to train the classifier network. DNN trained by standard supervised methods are unprotected to adversarial examples. Adversarial training helps DNN to be robust to adversarial perturbation. The adversarial training is to introduce an adversarial objective function using fast gradient sign method. Let J'(θ,x,y) be the loss function of adversarial training, which optimizes network against an adversary.

$$J\tilde{}(\theta, x, y) = \alpha J(\theta, x, y) + (1 - \alpha) J(\theta, x + E\ \text{sign}(\nabla x J(\theta, x, y)), y) \quad (2)$$

Eq. (2) consists of two cost functions. The first cost function is the original cross-entropy loss function for a deep neural network. The second is the loss function with adversarial perturbations added to each input x. α is a hyperparameter that adjusts the ratio between the two cost functions. Analyzing the principle of adversarial training and found a strong connection between robust optimization, regularization. This initiates a minimization-maximization approach for adversarial training, which in turn makes neural networks stable around training points. They also generalized the fast gradient sign method and presented another adversarial training method with L2 norm constraint.

$$\eta = \epsilon \frac{\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y)}{\|\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y)\|_2} \quad (3)$$

$$\tilde{J}(\boldsymbol{\theta}, \boldsymbol{x}, y) = \alpha J(\boldsymbol{\theta}, \boldsymbol{x}, y) + (1 - \alpha) J\left(\boldsymbol{\theta}, \boldsymbol{x} + \epsilon \frac{\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y)}{\|\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y)\|_2}, y\right) \quad (4)$$

B.Generative Adversarial Network

GAN is a new framework for generative models. The conventional means of training this model is to maximize the likelihood function, which computes various features such as marginal probabilities and partition functions, which are computationally intractable in most cases. It allows us to train a model without the intractable computation. A GAN framework allows two networks to compete with each other. These two networks are: a generative model, which maps a sample z (noise distribution) to the data distribution, and a discriminative model, which discriminates between training data and a sample from a generative model. The goal of a generative model is to maximize the posibility that the discriminative model will produce a mistake. Thus, generative and discriminative models play the following two-player minimax game with a value function V (G,D).

$$\min_G \max_D V(D,G) = E_{x \sim p_{data}(x)}[\log D(x)] + E_{z \sim p_z(z)}[\log(1 - D(G(z)))] \quad (5)$$

The competition in this minimax game allows both models to improve their ability until the discriminator cannot distinguish a generated sample from a data sample.

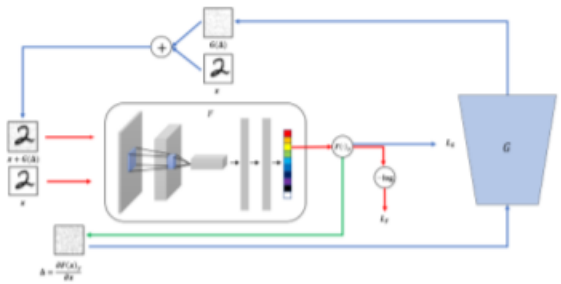

Figure 1: Adversarial training with Generative Adversarial Trainer:(1) Generative Adversarial Trainer G is trained to generate an adversarial perturbation that can fool the classifier network using the gradient of each image. (2) Classifier Network F is trained to classify correctly both original and adversarial examples generated by G.

III. PROPOSED METHOD

Here, we propose a novel adversarial training framework. However, we first introduce a generative adversarial trainer (GAT), which plays a major role in adversarial training. The objective of the GAT is to generate adversarial perturbations that can easily fool the classifier network using a gradient of images. A classifier is trained to classify correctly the original and the adversarial images generated by the GAT. The entire procedure is shown in Fig. 1. In Section 3.1, we describe our notation. Section 3.2 describes the structure of the GAT and Section 3.3 explains the adversarial training mechanism used with the GAT.

A.Notations

We denote a labeled training set by {(x(i),y(i))}N i=1, where x (i) ∈ R H×W×C represents input images with height H, width W, and channel C, and y(i) ∈{1,••• ,K}is a label for an input x (i). Two neural networks in the proposed method. One is a standard K-class classifier network F(x;θf) which is defined by:

$$F : RH\times W\times C \rightarrow RK,\ F(x) = [F(x)1, F(x)2, \bullet\bullet\bullet, F(x)K]\ T \quad (6)$$

where F(x) represents the class probability vector computed using the softmax function. The other neural network is a GAT G(Δ;θg) which is defined by:

$$G : RH\times W\times C \rightarrow RH\times W\times C \quad (7)$$

Note that G(Δ) represents the perturbation of the input image x, where Δ = ∂F(x)y/ ∂x denotes the gradient of input images with respect to the class probability of the label. We use a cross entropy loss function for the classifier F(x;θf), which is denoted by:

$$J(\theta f, x, y) = -\log F(x;\theta f)y \quad (8)$$

B.Generative Adversarial Trainer

The main idea of the GAT is to use a neural network to find the perturbation generator function specific to the classifier rather than just the sign or normalized functions used in the fast gradient method. The objective of the GAT is to find best perturbation image using the gradient of each image. To achieve this goal, the loss function is defined as follows:

$$LG(\Delta, y) = F(x + G(\Delta))y + cg \bullet \|G(\Delta)\|2\ 2 \quad (9)$$

The loss function of GAT consists of the two cost functions. One is the loss function, which is used to find perturbation images that lower the classifier's class probability. The other cost function restricts the power of the perturbation to prevent it from being too large. In (9), cg is a hyperparameter which adjusts the ratio between two cost functions. If cg is too low, it will find only a trivial solution with very high perturbation power. If cg is too high, only a zero perturbation image will be generated. Hence, finding the appropriate cg through a hyperparameter search is crucial.

C.Adversarial Arainer with GAT

As an analogy, our adversarial training framework is compared to the spring training of a baseball team. A trainer analyzes the vulnerable points of a player and, based on this analysis, trains the player's weakest parts in addition to providing general training. This process is repeated over and over again. The goal at the end of the spring camp, is to make the player overcome most of his weaknesses and become a better player. GAT also plays a similar role to a trainer for a baseball team. During each and every training step, GAT learns to generate the best adversarial perturbation for each input. Simultaneously, a classifier network is trained to classify correctly both original and adversarial examples generated by GAT. The loss function of GAT is given as (9), whereas that of the classifier network is based on the adversarial objective function.

$$LF = \alpha \bullet J(\theta f, x, y) + (1-\alpha) \bullet J(\theta f, x + G(\Delta), y) \quad (10)$$

For the sake of simplicity, we used α = 0.5 in all experiments. Similar to GAN, completely optimizing GAT in the inner loop of training is computationally expensive and would result in overfitting if we do not have a large number of datasets. Instead, we alternately optimize generator network k steps and the classifier network 1 step.

IV. RESULT &DISCUSSIONS

We proposed a adversarial training method by combining adversarial training [6] and GAN [4]. Experimental results suggest that our proposed method is not only robust against adversarial examples, but also effective in improving the generalization accuracy of the classifier. We believe that there are two main reasons to describe the better performance than the conventional fast gradient method. First, the classifier has different robustness for every training data. In some images, the classifier can be fooled easily when having only low perturbation power. Nevertheless, in other images, it cannot be easily fooled even with very high perturbation power. Because the conventional gradient method normalizes the size of every gradient, it generates adversarial images of the same perturbation power for all training images, which means that it is difficult for networks to converge. Generating adaptive adversarial examples based on the degree of robustness of every image can help efficiently train the network. Second, the classifier network is a non-linear function. If the classifier is a perfect linear function, finding better adversarial images than those found by the fast gradient method is impossible. However, because the classifier is a non-linear function, GAT can detect non-linear patterns in the network and use a gradient to produce better perturbations than with the fast gradient method. Our proposed GAT effectively solves both problems because it does not normalize the gradient vector and evolves adaptively as the classifier is trained. Therefore, training a classifier that is robust to various adversarial examples is possible, and accordingly, effectively regularizes the model. However, further study is required because the proposed method takes 3 to 4 times longer training (depending on the capacity of the generator network) than the conventional fast gradient method. In addition, hyperparameters of each network should be carefully tuned because of the properties of GAN training.

V. CONCLUSSION

In this paper, we proposed a method to make a classifier robust to adversarial examples using a Generative Adversarial Network framework. The generator network generates a perturbation by finding the weaknesses of the classifier, and the classifier re-learns the image generated by the GAN back to the original label. As the two networks learn alternately, the classifier becomes more robust to the adversary image, and eventually the generator network will not be able to find a proper image that could fool the classifier. Our adversarial training method is practical as it does not need expensive optimization process in the inner loop to find optimal adversarial images. The classifier with our adversarial training method is highly robust to the adversarial examples. Further, it was found that the proposed method was effective in regularizing neural networks. To the best of our knowledge, this is the first method to apply a GAN framework to adversarial training (or supervised learning). Hence, much work remains to improve the method. What is the optimal capacity of a generator? Does additional information other than a gradient exist that can help the generator to find a better adversarial image? When our method is applied to larger networks such as Inception [19], can similar results to those in this study be achieved? Further research is required to address these issues.

REFERENCES

1. N. Carlini and D. Wagner. Towards evaluating the robustness of neural networks. In Security and Privacy (SP), 2017 IEEE Symposium on, pages 39–57. IEEE, 2017.
2. I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572, 2014.
3. Nguyen, A., Yosinski, J. and Clune, J., 2015, June. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 427-436). IEEE.
4. Papernot, N., McDaniel, P. and Goodfellow, I., 2016. Transferability in Machine Learning: from Phenomena to Black-Box Attacks using Adversarial Samples. arXiv preprint arXiv:1605.07277.
5. Papernot, N., McDaniel, P., Goodfellow, I., Jha, S., BerkayCelik, Z. and Swami, A., 2016. Practical Black-Box Attacks against Deep Learning Systems using Adversarial Examples. arXiv preprint arXiv:1602.02697.
6. Kurakin, A., Goodfellow, I. and Bengio, S., 2016. Adversarial examples in the physical world. arXiv preprint arXiv:1607.02533.
7. Miyato, T., Maeda, S.I., Koyama, M., Nakae, K. and Ishii, S., 2015. Distributional smoothing with virtual adversarial training. stat, 1050, p.25.