PLOMS Journal of Artificial Intelligence (PLOMS AI)

PLOMS AI

Review Article

Adversarial Training and Machine Learning

Fahdah Mahdi AL-Tbenawey[1,*] and Araa Aref Al-Hamazani[1]

College of Computer Science and Engineering, University of Hail, Hail, Saudi Arabia; s20190422@UOH.EDU.SA

Correspondence: s20190422@UOH.EDU.SA

Received: 01-05-2021; Accepted: 01-06-2021; Published: 04-06-2021

Abstract: Artificial intelligence can be described as the study of machine systems with the ability to reason and perform cognitive functions in a manner almost similar to human intelligence. Artificial Intelligence has grown in prominence over the past few decades. Today, artificially intelligent algorithms control complex banking and financial systems, self-driving cars, and even news feeds. Machine Learning as a subfield has been at the forefront of AI adoption in several industries and sub-fields of AI. Today, ML is used in several applications such as facial recognition, malware detection, robotics, and self-driving cars. Like every computer-based system, however, ML poses its own set of challenges in cybersecurity. This is made harder by the fact that it is increasingly being adopted at a much faster rate than other technological systems. This has great risk not only for businesses and clients who use AI systems but also for the adoption of AI. This paper explored the cyber risks and the potential impact of AI. It detailed the external and internal organizational risks associated with the adoption of AI. In particular, it was concerned with Adversarial Machine Learning as a cybersecurity risk and its potential implications. A review of the literature found several organizations had experienced Adversarial Machine Learning as a threat. A number of these attacks were evasion attacks that manipulated data sets and were therefore hard to detect. This paper used stochastic adversarial training methods to show Adversarial Training can make ANNs adversarial robust. This paper recommends the use of Adversarial Training as a way of combatting Adversarial ML attacks.

Keywords: Artificial Intelligence; Machine Learning; Adversarial Machine Learning; Adversarial Training

1. Introduction

Artificial Intelligence has four basic sub-fields: neural networks, machine learning, natural language processing, and computer vision systems [1]. Machine Learning is a branch of Artificial Intelligence that utilizes data to allow the machine to make predictions about the data without explicit programming. In machine learning, machines are trained to use partially labeled data and data structures in a process called supervised learning. Machines are then allowed to make predictions about the dataset [2]. When the machine makes a mistake, it is fine-tuned to make better predictions. This process is called reinforcement learning. It allows machines to learn from their own mistakes and successes. Machine learning relies on the presence of neural networks to function. Neural networks are structures with various configurations that allow machines to perform supervised, unsupervised, and reinforcement learning. Artificial neural networks (ANNs) mimic the structure of neurons in the human brain and allows machines to create outputs in response to stimuli [3]. This allows the machine to learn much as a human child would learn. As such, Machine learning is good with predicting patterns in large datasets. Machine learning algorithms need well captured, properly labeled data sets to properly function. Since machines require more data than human beings, the supply and quality of data present a challenge in mitigating cybersecurity risks [4]. ML systems might require large datasets which are hard to acquire to properly function but the organization might not have enough resources. Additionally, the datasets have to contain quality data and be free of human error. Human beings are typically not good at spotting errors in large datasets.

This makes ML systems vulnerable to Adversarial ML attacks. Adversarial ML attacks refer to are malicious inputs designed to trick machine learning models [4]. There are two kinds of adversarial attacks, machine-learning models can either be: (a) presented with inaccurate or misrepresentative data during training, or (b) introduced to maliciously designed data that is used to deceive already trained models into making errors. In both cases, a machine learning system M with a sample C can be classified by an ML system as true, that is, M(C) = ytrue. However, it is possible for a sample D which is indistinguishable from C to be classified correctly as M(D)= ytrue. Additionally, adversarial attacks are transferrable i.e., an adversarial ML attack that can be used on ML model M1 can also be used to attack M2. This makes it easy to perform a misclassification attack without necessarily understanding the underlying architecture [5]. In Adversarial attacks, attackers have three objectives. The first one is to access ML systems while evading detection without necessarily compromising normal system operation- these kinds of attacks are classified as security violations. Additionally, an attacker might cause privacy violations by obtaining private information about a system, its users, or data by reverse-engineering the learning algorithm [6]. An attacker might also aim to have a sample misclassified as a specific class. These attacks might be limited by knowledge restrictions about the system from the attacker [7]. In a white-box attack, the attacker knows everything about the neural network including all data on which this network was trained. In black-box methods, the attacker might only be able to send information and get simple results about a class. Adversarial attacks can be classified as either evasion attacks, poison attacks, or privacy attacks [8]. In evasion attacks, inputs that are wrongly classified by ML models are used during the training of the ML model. For example, changing pixels on an image such that the model cannot recognize it. In poison attacks, the attacker injects the system with noise data to purposely exploit it. This can involve things like label modification, data modification, data injection, and logic corruption. In privacy attacks, the attacker attempts to explore the system such as the neural network or dataset. Here, the attacker needs to have some knowledge about the system since many AI algorithms are proprietary. This paper will explore how adversarial training can be used to prevent different kinds of adversarial ML attacks.

Adversarial attacks are dangerous as many ML systems are safety-critical [9]. For example, the safety of self driving cars or ML programs used in surgery could be the difference between life and death. In the past few years, companies that have invested heavily in machine learning have faced multiple adversarial attacks. These include Google, Amazon, Microsoft, and Tesla. For example, in 2018, internet trolls found a way to manipulate Microsoft's "Tay" to have it make racist statements [10]. Despite this, many cybersecurity experts do not know how to prevent adversarial ML attacks. In the past few years, researchers have been exploring different methods such as adversarial training and defensive distillation. This work has however been done only on a small dataset. This paper will focus on adversarial training in the CIFAR-10 dataset. Adversarial Training is a branch of ML in which a neural network is trained on adversarial examples. It is one of the few defenses against adversarial attacks that withstands strong attacks. Consider a neural network that has been trained on an input distribution X with the corresponding label set L. Given an input example x with a corresponding label l, it can be shown that an adversarial example x′ can be obtained from x by adding a very small perturbation to the original input such that x′ is classified differently as compared to x. A typical non targeted adversarial ML attack takes the form of:

$$\max \delta\ l(x + \delta, y, \theta), \text{ subject to } ||\delta||p \leq e1 \tag{1}$$

Where δ is the adversarial perturbation, l is the classification proxy loss, x is the data image, θ the parameters of a fixed classifier, $||\delta||p$ is some l_p-norm distance metric, and e1 is the adversarial manipulation budget. This means the attacker attempts to maximize the size of the class he/she can manipulate. To solve this, we need to come up with a robust optimization formulation to ensure that the model cannot be attacked even if the adversary has full knowledge of the model. This means we optimize the min-max objective. This can be gotten by solving the outer minimization problem to ensure we are always one step ahead of the attacker. The min-max objective takes the form of the equation:

$$\min \theta\ 1/\ |S|\ \sum x,y\in S\ \max\|\delta\|\leq\epsilon\ell(h\theta(x+\delta),y) \qquad (2)$$

To optimize for θ, we can use stochastic gradient descent. Where θ is optimized in respect to the loss function such that :

$$\theta :=\theta-\alpha 1|B|\sum x,y\in B\ \nabla\theta\max\|\delta\|\leq\epsilon\ell(h\theta(x+\delta),y) \qquad (3)$$

$$Loss = \frac{1}{(m-k)+\lambda k}\left(\sum_{i\in CLEAN} L(X_i|y_i) + \lambda \sum_{i\in ADV} L(X_i^{adv}|y_i)\right) \qquad (4)$$

where $L(X|y)$ is a loss on a single example X with true class y; m is the total number of training examples in the minibatch; k is the number of adversarial examples in the minibatch and λ is a parameter that controls the relative weight of adversarial examples in the loss. The inner gradient $\nabla\theta\max\|\delta\|\leq\epsilon\ell(h\theta(x+\delta),y)$, can be gotten using Danskin's Theorem which states that to compute the (sub)gradient of a function containing a max term, we need to 1) find the maximum, and 2) compute the normal gradient evaluated at this point. This:

$$\theta\max\|\delta\|\leq\epsilon\ell(h\theta(x+\delta),y)=\nabla\theta\ell(h\theta(x+\delta\star(x)),y) \qquad (5)$$

$$\text{where } \delta\star(x)=\arg\ \max\|\delta\|\leq\epsilon\ell(h\theta(x+\delta),y) \qquad (6)$$

Summarily, the objective of stochastic adversarial training is to:

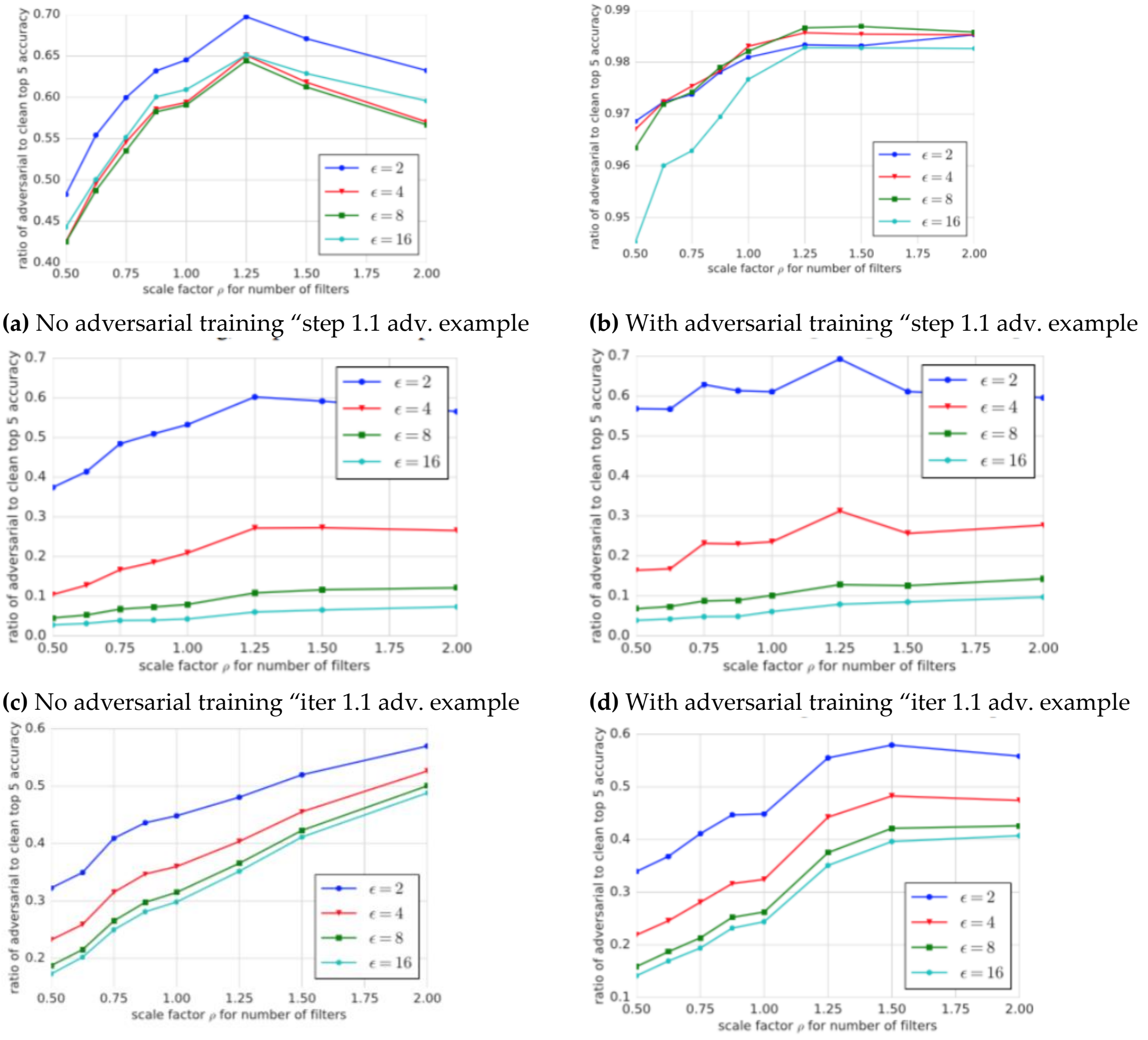
- Initialize gradient vector $g:=02$.
- For each (x,y) in B:a. Find an attack perturbation $\delta\star$ by (approximately) optimizing $\delta\star=\arg$
- $\max\|\delta\|\leq\epsilon\ell(h\theta(x+\delta),y)$b.
- Add gradient at $\delta\star g:=g+\nabla\theta\ell(h\theta(x+\delta\star),y)$
- Update parameters θ such that $\theta:=\theta-\alpha|B|g$

2. Results of Adversarial Training

Many experiments have been done with single-step models. We concluded that the aggressive method, the net type, the one-step method, increases the strength and toughness of all the models of aggression that have been tested. A distance still exists between the precision that is variable on the set of models used in training and evaluation. The antagonistic model is reduced by (<1%) in the clean models in ImageNet experiments. And that this is a clear difference from the patterns of hostility that were previously talked about and reported. As can be seen, training on accuracy has increased in the models tested [8] [9]. That is a clear statement of one of the possible causes of the hostile model operating in the arrangement. For datasets in which few digits are for short examples the primary concern, however, the hostile system reduces the test error rate. With regard to datasets such as ImageNet, it is common for modern models to contain a large error. Our results indicate that hostile training should be used in two scenarios: 1. The model must be worked on and an organizer is required. 2. When there is a situation where security against hostile examples is an unsuspecting source, then hostile training is the method that provides maximum safety, and we be aware that we may lose a little bit of accuracy. By comparing different one-step methods of antagonist training, we observed that the best results in terms of or accuracy in the test group are achieved using "Step # 1". Or "the second step." method. Moreover, the use of these two methods helped the model become robust to the hostile examples generated by the other one-step methods. Hence, for the final experiments, we used the 'first step'. Adversarial method.

a. Results with Different Activation Functions

Evaluation of antagonistic models When working on them we evaluated the robustness of examples. The network was trained with different nonlinear activation functions instead of standard relu activation when used with hostile training on "Step l.l." Hostile pictures. We tried to use the following activation functions in place of relu:



(a) No adversarial training "step 1.1 adv. example

(b) With adversarial training "step 1.1 adv. example

(c) No adversarial training "iter 1.1 adv. example

(d) With adversarial training "iter 1.1 adv. example

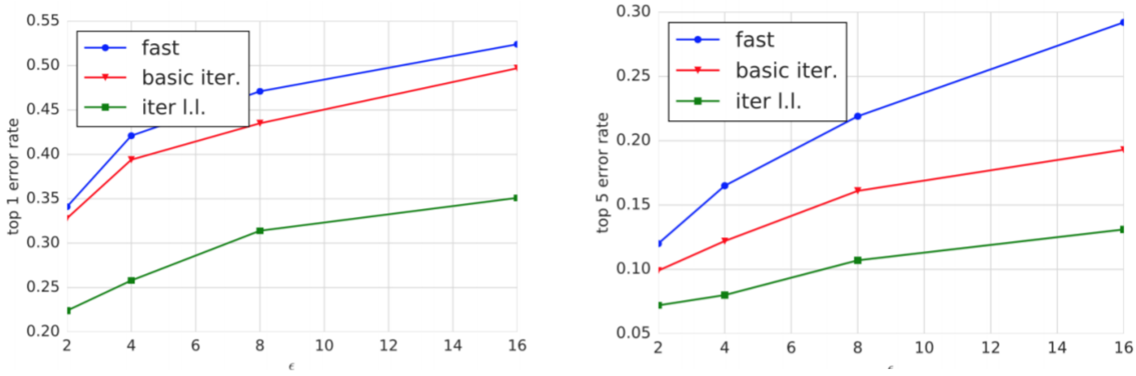
(e) No adversarial training "relu 1.1 adv. example

(f) With adversarial training "relu 1.1 adv. example

Figure 1. *Results with Different Activation Functions*

relu6(x) = min(relu(x) (6)
ReluDecay (x) = relu(x) for β ∈ {0.1, 0.01, 0.001}
β 1+βrelu(x)2

tanh and ReluDecayβ = 0.1 lose about 2% -3% accuracy in the clean examples and about 10% -20% in the "step l.l." Adversarial examples. Relu6, ReluDecayβ = 0.01, and ReluDecayβ = 0.001 showed similar accuracy (in the range of ± 1%) for re-dependence on clean images and a little loss of percent accuracy in "step l.l." Pictures. At the same time, all nonlinear activation functions increased classification accuracy in some form of iterative antagonism. We observe a change in the effect of the size of the hostile disorder size on the error rate using another model. Both source and target models were Inception v3 networks with different random settings.



(a) Top 1 error rate **(b)** Top 5 error rate

Figure 2. *Different Number of Adversarial Examples in The Minibatch*

b. Results with Different Number of Adversarial Examples in The Minibatch

Conclusions regarding the effect of numbers on K antagonism models were extracted in minibatch clean examples and antagonism. An increase in the accuracy of the antagonistic models and a decrease in the accuracy were observed, the presence of more than half of the antagonistic examples in the minibatch (which corresponds to k>16 in our case) and that this does not achieve an improvement in the accuracy of the antagonistic models.

3. Conclusions

We studied in some depth in this paper on how to increase strength and accuracy For hostile examples of large models (v3 receivers) trained on a large data set (ImageNet). Work has been done on models of hostility, training on it, and how to provide models for adversity using one-step methods. While hostile models did not function as expected to help counter iterative techniques, we noted that examples of antagonism resulting from iterative techniques are less likely to be transferred between networks, providing indirect strength against hostile black box attacks.

Author Contributions: The authors contributed equally in this paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

[1]. Biggio, B., & Roli, F. (2018). Wild patterns: Ten years after the rise of adversarial machine learning. Pattern Recognition, 84, 317-331.

[2]. Hoadley, D. S., & Lucas, N. J. (2018). Artificial intelligence and national security.

[3]. Carton, S., Mei, Q., & Resnick, P. (2018). Extractive adversarial networks: High-recall explanations for identifying personal attacks in social media posts. arXiv preprint arXiv:1809.01499.

[4]. Pinot, R., Meunier, L., Araujo, A., Kashima, H., Yger, F., Gouy-Pailler, C., & Atif, J. (2019). Theoretical evidence for adversarial robustness through randomization. arXiv preprint arXiv:1902.01148.

[5]. Kurakin, A., Goodfellow, I., & Bengio, S. (2016). Adversarial machine learning at scale. arXiv preprint arXiv:1611.01236.

[6]. Marsland, S. (2015). Machine learning: an algorithmic perspective. CRC press.

[7]. Yavanoglu, O., & Aydos, M. (2017, December). A review on cyber security datasets for machine learning algorithms. In 2017 IEEE International Conference on Big Data (Big Data) (pp. 2186-2193). IEEE.

[8]. Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. (2014). Generative adversarial networks. arXiv preprint arXiv:1406.2661.

[9]. Miyato, T., Dai, A. M., & Goodfellow, I. (2016). Adversarial training methods for semi-supervised text classification. arXiv preprint arXiv:1605.07725.