Adversarial Machine Learning with Double Oracle

Kai Wang
Department of Computer Science, University of Southern California, USA
wang319@usc.edu

Abstract

We aim to improve the general adversarial machine learning solution by introducing the double oracle idea from game theory, which is commonly used to solve a sequential zero-sum game, where the adversarial machine learning problem can be formulated as a zero-sum minimax problem between learner and attacker.

1 Introduction

Adversarial machine learning has been proven to be useful in many domains including spam detection, image recognition, and self-driving car. With adversarial training technique, we can make the model more robust against noisy input features. We can also reduce overfitting to the training set by introducing variant adversarial examples. How to correctly and efficiently setup the adversarial machine learning has become a big interest to people.

There are two main directions in the adversarial machine learning: i) as an attacker, how to conduct an attack. ii) as a defender, how to defend against the attacker given the attacker knows how to attack. The first direction is generally done by simply using Fast Gradient Sign Method (FGSM) introduced by [Goodfellow *et al.*, 2014]. FGSM can construct the adversarial examples very efficiently, therefore solving the scalability issue of general adversarial training. A variant to the FGSM method is Projected Gradient Descent proposed by [Madry *et al.*, 2017]. They propose an intuitive way to run gradient descent with constraints, which also leads to an efficient way to further train a robust model against such attack. In [Athalye *et al.*, 2018], they categorize the common attacks into several categories. But all fall into the "obfuscated gradient" based method. They further propose an attack method to conquer "obfuscated gradient" based defense model, which is trained against an obfuscated gradient based attacks.

On the defense side, the problem can be formulated as a minimax optimization problem or so called the robust optimization problem. People usually solve it by using common minimax technique [Goodfellow *et al.*, 2014; Madry *et al.*, 2017]. This technique can iteratively solve the saddle point of the objective surface, leading to a solution of the minimax problem. The adversarial training model is proven and tested to be much more robust against adversarial attacks.

From the attacker's perspective, depending on the accessibility of the model, it could be divided into i) white-box attack, ii) black-box attack. Here we mainly focus on the white-box attack, which assumes the attacker knows all the hyper-parameters of the model. Therefore the attacker can also access to arbitrary gradient of the entire model, which enables him to run the gradient based attack.

On the other hand, from the game theory perspective, this problem can be formulated as a sequential game, where the defender decides the model first, then the attacker comes and conducts the attack, which is generally called Stackelberg game. There are many efficient methods been proposed to solve the Stackelberg equilibrium very efficiently. Among them, double oracle is proposed by [McMahan *et al.*, 2003] and is commonly used to solve large-scale problems in security game [Jain *et al.*, 2011].

In this paper, we aim to solve the Stackelberg equilibrium of the minimax adversarial problem by using double oracle method. In the previous literature, a single predictor is usually proposed to solve the adversarial problem. However, from the game theory perspective, it is quite rare that a single strategy can perform very well against a attacker playing against to the defender. For example, in the rock–paper–scissors game, given the assumption that the defender moves first, it is hard to defend against the attacker's strategy with a single defender pure strategy. Instead, a mixed strategy including randomization could outperform a single pure strategy quite a lot in many cases. Therefore, we aim to compute a mixed strategy over the defender's strategy space to further improve the robustness against a malicious attacker.

2 Problem Statement

Given the training instances $D_{\text{train}} = \{(x_i, y_i)\}$ and testing instance D_{test}, where x_i is the feature and y_i is the label. The defender is using a specific model $m(x, \theta)$, where x is the feature fed into the model and $\theta \in \Theta$ is the hyper-parameter of the model. Given a prediction $m(x, \theta) \in R^n$ and a loss function $L : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$, the loss is $L(m(x, \theta), y)$. The expected loss over the entire training set can be written as $E_{(x,y)\sim D_{\text{train}}} L(m(x, \theta), y)$.

Once the defender's model θ and the instances D are given, the attacker can decide an attack $\delta_{(x,y)} \in \Delta$ for each instance (x, y) subject to some constraints Δ. So the attacker is maxi-

Algorithm 1: Entire Double Oracle

```
Input: training instance $D_{\text{train}} = \{(x, y)\}$ feature $x$ and label $y$; number of defender/attacker strategies $k$
Parameters: defender strategies $\{\theta^1, ..., \theta^k\}$ with probabilities $\{p_1, ..., p_k\}$; attacker strategies $\{\delta^1, ..., \delta^k\}$ with probabilities $\{q_1, ..., q_k\}$.
Initialization: $\theta^i = \theta\ \forall i$ where $\theta$ is the pre-trained non-adversarial model; $\delta^i = 0\ \forall i$.
for iteration = $1, 2, ...$ do
    Compute the payoff of each defender/attacker strategy given the current adversarial strategy.
    Attacker: Choose $\delta^a$ the worst attacker strategy. Compute the optimal attack $\delta^*$ against the defender mixed strategy, which uses model $\theta^i$ with probability $p_i$. Replace $\delta^a$ by $\delta^*$.
    Defender: Choose $\theta$ the worst defender strategy. Improve model $\theta$ against the adversarial example $\{x + \delta_x, y | \delta \sim \text{attacker mixed strategy}\}_{(x,y)\in D_{\text{train}}}$ to update the current model $\theta$.
    Core linear program: compute the payoff matrix $M$ of every pair of defender/attacker strategy. Solve a Stackelberg game with matrix $M$ and obtain the defender optimal mixed strategy $\{p_1^*, ..., p_k^*\}$ and attacker optimal mixed strategy $\{q_1^*, ..., q_k^*\}$.
    Update the probability $p$ and $q$ by $p^*, q^*$.
```

mizing the expected defender loss:

$$\max\nolimits_{\delta_{(x,y)}\in\Delta\ \forall(x,y)} E_{(x,y)\in D} L(m(x+\delta_{(x,y)},\theta),y) \tag{1}$$

which can be generally solved by running gradient descent based method like PGD [Madry *et al.*, 2017].

Given the attacker is maximizing the expected loss, the defender wants to prevent it, which leads to a minimax problem:

$$\min\nolimits_{\theta\in\Theta} \max\nolimits_{\delta_{(x,y)}\in\Delta\ \forall(x,y)} E_{(x,y)\in D} L(m(x+\delta_{(x,y)},\theta),y)$$

In [Madry *et al.*, 2017], they propose an iterative algorithm to solve the above minimax problem by computing the saddle point of the expected loss function. Here, instead of using a single defender model θ here, we aim to randomize over multiple defender models $\hat{\Theta} = \{\theta^1, \theta^2, ..., \theta^k\}$ with probability $\{p_1, p_2, ..., p_k\}$, where k is the number of models. Under such condition, the defender's problem becomes:

$$\min_{\hat{\Theta}} \max_{\delta_{(x,y)}\in\Delta\ \forall(x,y)} E_{\theta^i\sim\hat{\Theta}} E_{(x,y)\in D} L(m(x+\delta_{(x,y)},\theta^i),y) \tag{2}$$

3 Methodology and Algorithms

Our algorithm iteratively chooses the worst defender/attacker strategy to improve. Each player trains its strategy based on the current adversarial mixed strategy, where we call an oracle. The double oracle method comes from the idea of iteratively updating players' strategy, which is guaranteed to converge to equilibrium eventually when the strategy space is finite and the oracle is optimal. In our case, we use gradient-based oracles (hill climbing algorithm) and both strategy spaces are not finite. Therefore we do not possess the theoretical guarantee. This is the common issue of non-convexity of neural network. But empirically, we expect it to outperform the existing single gradient-based adversarial training since we are using randomized model.

3.1 Randomization and Ensemble

If we randomize over several models $\{\theta_1, ..., \theta_k\}$ with probability distribution $\{p_1, ..., p_k\}$, given an attack $\delta \in \Delta^{|D_{\text{test}}|}$ then the expected testing loss would be

$$E_{\theta\sim\{\theta_1,...,\theta_k\}} E_{(x,y)\sim D_{\text{test}}} L(m(x+\delta_x,\theta),y) \tag{3}$$

If we are using ensemble, then we have to combine the predicted probability before making prediction, resulting to the following testing loss:

$$E_{(x,y)\sim D_{\text{test}}} L(E_{\theta\sim\{\theta_1,...,\theta_k\}}[m(x+\delta_x,\theta)],y) \tag{4}$$

Since the loss function is convex, given an attack δ, the testing loss by using ensemble method is always smaller than the testing loss by using randomization. By introducing an optimal attack, the attacker aims to maximize the expected testing loss in equations (3) and (4) respectively. But for every attack δ, (4) is always smaller than (3). Therefore, even after taking a maximization over δ, the optimal testing loss of ensemble is still always smaller than the testing loss of randomization. This implies that we can derandomize the original mixed strategy to a deterministic ensemble method without lossing solution quality.

4 Consclusion

With the help of game theory, we can construct the randomized strategy (model) to defend the attacker. By the convexity and the Jenson inequality, we can further derandomize the randomized strategy to a deterministic strategy without lossing solution quality. This helps enhance the robustness of the machine learning model. The idea can also be adopted to many other domains to gain robustness for free.

References

[Athalye *et al.*, 2018] Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. *ICML*, 2018.

[Goodfellow *et al.*, 2014] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *CoRR*, 2014.

[Jain *et al.*, 2011] Manish Jain, Dmytro Korzhyk, Ondřej Vaněk, Vincent Conitzer, Michal Pěchouček, and Milind Tambe. A double oracle algorithm for zero-sum security games on graphs. AAMAS, 2011.

[Madry *et al.*, 2017] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *ICLR*, 2017.

[McMahan *et al.*, 2003] H Brendan McMahan, Geoffrey J Gordon, and Avrim Blum. Planning in the presence of cost functions controlled by an adversary. ICML, 2003.