Towards Understanding the Dynamics of the First-Order Adversaries

Zhun Deng [1] **Hangfeng He** [2] **Jiaoyang Huang** [3] **Weijie J. Su** [2]

Abstract

An acknowledged weakness of neural networks is their vulnerability to adversarial perturbations to the inputs. To improve the robustness of these models, one of the most popular defense mechanisms is to alternatively maximize the loss over the constrained perturbations (or called adversaries) on the inputs using projected gradient ascent and minimize over weights. In this paper, we analyze the dynamics of the maximization step towards understanding the experimentally observed effectiveness of this defense mechanism. Specifically, we investigate the non-concave landscape of the adversaries for a two-layer neural network with a quadratic loss. Our main result proves that projected gradient ascent finds a local maximum of this non-concave problem in a polynomial number of iterations with high probability. To our knowledge, this is the first work that provides a convergence analysis of the first-order adversaries. Moreover, our analysis demonstrates that, in the initial phase of adversarial training, the scale of the inputs matters in the sense that a smaller input scale leads to faster convergence of adversarial training and a "more regular" landscape. Finally, we show that these theoretical findings are in excellent agreement with a series of experiments.

1. Introduction

Neural networks have achieved remarkable success in many fields such as image recognition (He et al., 2016) and natural language processing (Devlin et al., 2018). However, it has been recognized that neural networks are not robust against adversarial examples – prediction labels can be easily manipulated by human imperceptible perturbations (Goodfellow et al., 2014; Szegedy et al., 2013). In response, many defense mechanisms have been proposed against adversarial attacks such as input de-noising (Guo et al., 2017), randomized smoothing (Ilyas et al., 2019), gradient regularization (Papernot et al., 2017), and adversarial training (Madry et al., 2017). Among these, one of the most popular techniques is adversarial training, which proposes to add adversarial examples into the training set as a way of improving the robustness.

As oppose to earlier work that only adds adversarial examples several times during the training phase, more recently, Madry et al. (2017) propose to formulate adversarial training through the lens of robust optimization, showing substantial improvement. More precisely, robust optimization for a loss function L in its simples setting takes the form

$$\min_{\boldsymbol{\theta}\in\Theta} \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}} \max_{\|\boldsymbol{\delta}\|_p\leqslant\varepsilon} L(\boldsymbol{\theta}, \boldsymbol{x}+\boldsymbol{\delta}, y),$$

where $\boldsymbol{\theta} \in \Theta$ is the parameter and $(\boldsymbol{x}, y)$ are the input and label following some unknown joint distribution $\mathcal{D}$. The inner maximization problem is to find an adversarial example, where $\boldsymbol{\delta}$ is an adversarial perturbation with l_p norm constraint for some integer $1 \leqslant p \leqslant \infty$.

For neural networks, the inner maximization problem is typically non-concave and the most commonly used method in implementation is through the first-order method – projected gradient ascent. However, as pointed out by (Wang et al., 2019), the degree to which it solves the inner maximization problem has not been thoroughly understood. While there are several papers providing great theoretical insights to the convergence of adversarial training, they either formulate the inner maximization problem as maximizing the first order taylor expansion of the loss (Wang et al., 2019), or treat the inner maximization problem abstractly as a general function of data and study the convergence in the neural tangent kernel regime (Gao et al., 2019). In our paper, we make the first step to analyze the dynamics of projected gradient ascent of neural networks.

The first question is about the effectiveness of projected gradient ascent. To prove the effectiveness, we need to consider the time cost of using projected gradient ascent in the inner maximization problem. In (Madry et al., 2017), one claim is that using projected gradient ascent can find the adversaries rapidly. That claim is very important since adversarial training usually takes much longer than usual training due to the inner maximization problem. Specifically, if we use

[1] Harvard University [2] University of Pennsylvania [3] Institute of Advanced Study. Correspondence to: Zhun Deng <zhundeng@g.harvard.edu>.

Proceedings of the 37th International Conference on Machine Learning, Online, PMLR 119, 2020. Copyright 2020 by the author(s).

gradient method in the alternative optimization problem for both inner maximization and outer minimization, and denote the number of epochs taken to find adversaries with given weights by n_1, number of updates of weights by n_2, then the epochs taken by the adversarial training is $n_1 n_2$. To make the time cost of adversarial training bearable, the fact that n_1 is not large plays a key role here.

Another issue about effectiveness is whether the projected gradient ascent can truly find a local maximum and not be stuck at a saddle point. In (Madry et al., 2017), Madry et al. claims the loss (as a function of model parameters) typically has many local maximums with very similar values. So, if the projected gradient ascent truly finds a local maximum, the effectiveness of the adversarial training is trustworthy.

We summarize our first question below.

Questions 1. *Does projected gradient ascent truly find a local maximum rapidly?*

The second question we try to explore is whether the scale of inputs matters. In the adversarial training, ε's scale is usually in proportional to the scale of input $\boldsymbol{x}$:

$$\varepsilon = r\mathbb{E}\|\boldsymbol{x}\|_p.$$

For adversarial attacks on images, the ratio r is supposed to be small, so as to reflect the fact that the attacks are visually imperceptible.

For fixed $r > 0$, ε and the input scale $\mathbb{E}\|\boldsymbol{x}\|_p$ are closely related – a smaller input scale implies a smaller ε. In the implementation of image recognition using neural networks, people usually rescale the image pixels to $[0, 1]$ or $[-1, 1]$. While that seems not affecting regular optimization, it may affect adversarial training. So, we have the following question.

Questions 2. *When we fix the ratio r, do smaller input scales (implying smaller ε) help optimization of adversarial training?*

If the answer to Question 2 is positive, it will be helpful in the future applications to rescale the inputs to a smaller scale.

Both questions above have not been studied yet due to the highly non-concave landscape of adversaries.

1.1. Our contributions

Our analysis provides answers to Question 1 and 2 for the initial phase of the adversarial training, i.e. the weights are drawn from Xavier initialization (Glorot & Bengio, 2010). Even for this simple case, nothing has been discussed theoretically so far.

In Section 3 and 4, we provide the answer to Question 1 by showing projected gradient ascent indeed can find a local maximum rapidly by providing a convergence theorem.

Theorem 1.1 (Informal). *Projected gradient ascent can obtain an approximate local maximum, which is close to a true local maximum on the sphere in polynomial number of iterations when the learning rate is small enough. If we further allow learning rate shrinking with time, projected gradient ascent can converge to a local maximum.*

In Section 5, we answer Question 2 by showing a smaller scale helps in the perspectives of landscapes and convergence of trajectories. From our theory, we show a smaller input scale helps the trajectory converge faster if we had a bad initialization. Besides, a smaller input scale makes the local maximums concentrate better, which can partially explain why the loss value of local maximums share similar values (Madry et al., 2017). Lastly, we verify the previous claims by extensive numerical experiments.

Our work mainly focuses on the initial phase of adversarial learning, which may be a good start towards understanding the first-order adversaries.

1.2. Related work

Adversarial attack and defense Besides projected gradient ascent, some other have also been proposed to generate adversarial examples, such as FGSM (Goodfellow et al., 2014), l-BFGS (Szegedy et al., 2013) and C& W attack (Carlini & Wagner, 2017). Also, some attacks are proposed to attack black-box models, in order to defend against those attacks, many defense mechanisms have been proposed (Brendel et al., 2017; Chen et al., 2017). However, many of these defense models have been evaded by new attacks (Athalye et al., 2018) except (Madry et al., 2017). Besides, a line of work focus on providing certified robustness and robustness verification (Weng et al., 2018; Wong et al., 2018; Zhang et al., 2018), which also provide useful insights theoretically.

Adversarial training The first work to propose adversarial training is (Goodfellow et al., 2014), in which the authors advocate adding adversarial examples during training to improve the robustness. In (Madry et al., 2017), the authors use projected gradient ascent to find adversaries and reach a state of art performance. However, as we mentioned before, running projected gradient method is very slow, and some work (Shafahi et al., 2019) intend to solve this problem. Besides, the introducing of adversarial training also motivates a line of theoretical work, such as (Agarwal et al., 2018; Liu & Hsieh, 2019; Yin et al., 2018). However, none of them address the inner maximization problem using projected gradient ascent.

Non-convex optimization Non-convex optimization is notoriously hard to analyze. However, some work provide valuable guide. In (Ge et al., 2015), the authors analyze the dynamics of noisy gradient descent in the non-convex setting. Some following work including (Du et al., 2017) show gradient descent can take very long to escape saddle point but noisy gradient descent does not, and (Jin et al., 2017) shows noisy gradient descent can converge to a second order stationary point vert fast. In our setting, we do not need extra noise, but can still yield a good convergence result.

2. Preliminaries

Notations Throughout the paper, we use $[n]$ to denote $\{1, 2, \cdots, n\}$ and use $\|\cdot\|_p$ to denote l_p norm. In particular, for l_2 norm, we use $\|\cdot\|$ or $\|\cdot\|_2$ exchangeably. For any function $L : \mathbb{R}^d \mapsto \mathbb{R}$, ∇L and $\nabla^2 L$ denote the gradient vector and Hessian matrix respectively. We use $\mathbb{B}$ to denote ball and $\mathbb{S}$ to denote sphere. We denote $\angle[\boldsymbol{u}, \boldsymbol{v}] = \boldsymbol{u}^T \boldsymbol{v}/(\|\boldsymbol{u}\|\|\boldsymbol{v}\|)$, which is the cosine value of the angle between the two vectors $\boldsymbol{v}$ and $\boldsymbol{u}$. For a function $h(\boldsymbol{x})$, we sometimes use shorthand $\partial h(\boldsymbol{x})$ for gradient $\partial h(\boldsymbol{x})/\partial \boldsymbol{x}$.

Setup Recall adversarial learning aims to solve the robust optimization of loss function L:

$$\min_{\boldsymbol{\theta} \in \Theta} \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}} \max_{\|\boldsymbol{\delta}\|_p \leqslant \varepsilon} L(\boldsymbol{\theta}, \boldsymbol{x} + \boldsymbol{\delta}, y),$$

where $\boldsymbol{\theta} \in \Theta$ is the parameter, $(\boldsymbol{x}, y) \in \mathbb{R}^d \times \mathbb{R}$ is d-dimensional input and scalar output, which follows a joint distribution $\mathcal{D}$. The corresponding empirical version for samples $\{\boldsymbol{x}_i, y_i\}_{i=1}^n$ is

$$\min_{\boldsymbol{\theta} \in \Theta} \frac{1}{n} \sum_{i=1}^{n} \max_{\forall i \in [n], \|\boldsymbol{\delta}_i\|_p \leqslant \varepsilon} L(\boldsymbol{\theta}, \boldsymbol{x}_i + \boldsymbol{\delta}_i, y_i). \tag{1}$$

For fixed $\boldsymbol{\theta}$, solving the optimization problem (1) can be optimized as n different optimization problems separately: for each $\boldsymbol{x}_i$, we need to obtain a corresponding $\boldsymbol{\delta}_i$. In this paper, we focus on studying the convergence rate of finding adversaries, i.e. maximizing $\boldsymbol{\delta} \in \mathbb{R}^d$ when the constraint is the l_2-norm and the loss is the quadratic loss of shallow neural network:

$$\max_{\boldsymbol{\delta}} L(\boldsymbol{\theta}, \boldsymbol{x}+\boldsymbol{\delta}, y) = (y - f(\boldsymbol{a}, \boldsymbol{W}, \boldsymbol{\delta}+\boldsymbol{x}))^2, \text{ s.t. } \|\boldsymbol{\delta}\|_2^2 \leqslant \varepsilon^2. \tag{2}$$

Here, f is a two-layer neural network:

$$f(\boldsymbol{a}, \boldsymbol{W}, \boldsymbol{\delta} + \boldsymbol{x}) = \sum_{r=1}^{m} a_r \sigma(\boldsymbol{w}_r^T (\boldsymbol{x} + \boldsymbol{\delta})).$$

In the above equation, $\boldsymbol{a} = (a_1, a_2, \cdots, a_m)^T$ is an m-dimensional vector, $\boldsymbol{W} = (\boldsymbol{w}_1, \cdots, \boldsymbol{w}_m)$ is an $m \times d$-matrix and $\boldsymbol{\theta} = (\boldsymbol{a}^T, \text{Vec}(\boldsymbol{W})^T)^T$, where $\text{Vec}(\cdot)$ is the vectorization operator. We use σ to denote the softplus activation function such that $\sigma(x) = \log(1 + e^x)$.

We study the projected gradient ascent:

$$\boldsymbol{\delta}_{t+1} = \mathcal{P}_{\mathbb{B}(\mathbf{0},\varepsilon)} \Big[\boldsymbol{\delta}_t + \eta \frac{\partial L(\boldsymbol{\delta}_t)}{\partial \boldsymbol{\delta}_t}\Big], \quad t \geqslant 0,$$

where $\mathbb{B}(\mathbf{0}, \varepsilon)$ is a ball centered at $\mathbf{0}$ with radius ε in Euclidean distance, and $\mathcal{P}$ is the projection operator. $\boldsymbol{\delta}_0$ is uniformly sampled in the ball $\mathbb{B}(\mathbf{0}, \varepsilon)$.

In this paper, we always consider the problem under the following settings unless we state explicitly otherwise.

1. $\boldsymbol{w}_r$'s are i.i.d drawn from d-dimensional Gaussian $\mathcal{N}(0, \kappa^2 I)$, where $0 < \kappa \leqslant 1$ controls the magnitude of initialization.

2. a_r's are i.i.d drawn from Bernoulli distribution, which take $\pm\gamma$ with $1/2$ probability.

3. There exist $\mathcal{L}, \mathcal{U} > 0$ such that $\mathcal{L} < |y - f(\boldsymbol{a}, \boldsymbol{W}, \boldsymbol{\delta} + \boldsymbol{x})| < \mathcal{U}$ for all $\boldsymbol{\delta} \in \mathbb{B}(\mathbf{0}, \varepsilon)$.

4. $\boldsymbol{\delta}_0$ is initialized by drawing from a uniform distribution over $\mathbb{B}^\circ(\mathbf{0}, \varepsilon)$, where $\mathbb{B}^\circ$ stands for the interior of the ball $\mathbb{B}$.

In this paper, we will take the parameters according to Xavier initialization, which means $\kappa = d^{-1/2}$ and $\gamma = m^{-1/2}$.

Remark 1. *We study the case when the weights are drawn from commonly used distributions for initialization. Our analysis can be viewed as studying the dynamics of finding adversaries in the initial phase of training.*

3. Main Results

We present our main results on the convergence of projected gradient descent (PGD) in this section. Since the objective of optimization is $\boldsymbol{\delta}$, we use $L(\boldsymbol{\delta})$ for loss and we denote the constraint as $c(\boldsymbol{\delta}) = \|\boldsymbol{\delta}\|^2 - \varepsilon^2$. For convenience, we consider the minimization version:

$$L(\boldsymbol{\delta}) = -(y - f(\boldsymbol{a}, \boldsymbol{W}, \boldsymbol{\delta} + \boldsymbol{x}))^2.$$

The original problem in (2) is equivalent to:

$$\min_{\boldsymbol{\delta}} L(\boldsymbol{\delta}), \quad \text{s.t. } c(\boldsymbol{\delta}) \leqslant 0.$$

Then, the iterative optimization algorithm used becomes the projected gradient descent (PGD)

$$\boldsymbol{\delta}_{t+1} = \mathcal{P}_{\mathbb{B}(\mathbf{0},\varepsilon)} \Big[\boldsymbol{\delta}_t - \eta \frac{\partial L(\boldsymbol{\delta}_t)}{\partial \boldsymbol{\delta}_t}\Big], \quad t \geqslant 0.$$

Here, we provide the formal statement of our main results.

Theorem 3.1 (**Main Theorem**). *Suppose $m = \Omega(d^{5/2})$, there exists $\varepsilon_{\max}(m) = \Theta\big((\log m)^{-2}\big)$ and $\eta_{\max}(m, \varepsilon) = \min\{\Theta\big((\log m)^{-2}\big), \varepsilon^2\}$, if $\varepsilon < \varepsilon_{\max}(m)$, for any $\eta < \eta_{\max}(m, \varepsilon)$, with high probability, in $O(\eta^{-2})$ iterations, projected gradient descent will output a point $\boldsymbol{\delta}_t$ on the sphere which is $O(\eta^{1/2})$ close to some local minimum $\boldsymbol{\delta}^*$.*

Remark 2. *Our width requirement is much smaller compared to the results with respect to neural tangent kernels (Du et al., 2018; Jacot et al., 2018). The latter one requires $m = O(poly(n))$, where n is the samples size. Notice the scale of ε only requires to be upper bounded by $O(poly((\log m)^{-1})$, under that requirement, the activation function will be activated along the update of $\boldsymbol{\delta}$ with constant probability when $\|\boldsymbol{x}\|$ is small.*

Corollary 3.1 (**Shrinking learning rate**). *Under the assumptions of Theorem 3.1, for $\tilde{t}$ satisfying $\boldsymbol{\delta}_{\tilde{t}} \in \varepsilon\mathbb{S}^{d-1}$ and the tangent component of $\partial f(\boldsymbol{\delta}_{\tilde{t}})$ (for every point on the sphere, the tangent component of a vector is its projection to the tangent plane at that point) being smaller than $\eta^{1/2}$, let $D_s := \|\boldsymbol{\delta}_{\tilde{t}+s} - \boldsymbol{\delta}^*\|^2$, if we shrink the learning rate after $\tilde{t}$, in a way that*

$$\boldsymbol{\delta}_{\tilde{t}+s+1} = \mathcal{P}_{\mathbb{B}(\mathbf{0},\varepsilon)}\Big[\boldsymbol{\delta}_{\tilde{t}+s} - \eta_s \partial L(\boldsymbol{\delta}_{\tilde{t}+s})\Big], \quad t \geqslant 0,\ s \geqslant 0,$$

for $\eta_0 < \eta$, as long as $\eta_s \to 0$ as $s \to \infty$ and $\Pi_{i=0}^{k}(1 - \gamma\eta_i/2) \to 0$ as $k \to \infty$, we will have $D_s \to 0$. Furthermore, if

$$\eta_s \Pi_{i=0}^{k}(1 - \frac{\beta\eta_{s+i}}{2}) \leqslant \eta_{s+k+1} \tag{3}$$

for all $s, k \in \mathbb{N}$, where β is a constant depending on $(d, m, \varepsilon, \eta)$ and can be calculated explicitly, then for all $s \in \mathbb{N}$,

$$D_s \leqslant O(\eta_s).$$

Remark 3. *One concrete example satisfying Eq. (3) is the following one: if $\eta_s = 2/(\beta s + \beta z)$ for large enough integer z,*

$$D_s \leqslant O(\frac{1}{z+s}).$$

3.1. Our interpretation

Our results state that for a wide enough one hidden layer neural network, if the attack size ε is small, then we can choose small enough learning rate, such that the trajectory of PGD can quickly reach a point that is very close to one of the minimizers. Besides, the minimizer is located on the sphere with high probability. The theory can partially explain the observation in (Madry et al., 2017): it does not take too many iterations to find an adversary, which is the key to guarantee the time cost of robust optimization modest. Also, our theory is consistent with the observation that the PGD will end up on the sphere for most samples in the implementation of adversarial training.

4. Proof Sketch

In this section, we briefly sketch our proof. We show with high probability, the gradient is non-vanishing in the ball. Meanwhile, on the sphere, there is no saddle points. Besides, the trajectory will not get stuck near local maximums and can converge to a local minimum in polynomial number of iterations.

Lemma 4.1 (**Dynamics in the ball**). *For $m = \Omega(d^{5/2})$, there exists $\varepsilon_{\max}(m) = \Theta\big((\log m)^{-1/2}\big)$ and $\eta_{\max}(m, \varepsilon) = \min\{\Theta\big((\log m)^{-1}\big), \varepsilon^2\}$, if $\varepsilon < \varepsilon_{\max}(m)$, $\eta < \eta_{\max}(m, \varepsilon)$, with high probability, whenever $\boldsymbol{\delta}_{t+1} \in \mathbb{B}^{\circ}(\mathbf{0}, \varepsilon)$*

$$L(\boldsymbol{\delta}_{t+1}) - L(\boldsymbol{\delta}_t) \leqslant -\Omega(\eta).$$

The above lemma shows the trajectory is very unlikely to terminate in the ball since the $(t + 1)$-th step can make progress if $\boldsymbol{\delta}_{t+1} \in \mathbb{B}^{\circ}(\mathbf{0}, \varepsilon)$.

Next, we focus on studying the dynamics on the sphere. For constrained optimization, we can locally transform it into an unconstrained problem by introducing Lagrangian multipliers:

$$L(\boldsymbol{\delta}, \lambda) = L(\boldsymbol{\delta}) - \lambda c(\boldsymbol{\delta}).$$

Under some regularity conditions, we can obtain the Lagrangian multiplier $\lambda^*(\cdot)$:

$$\lambda^*(\boldsymbol{\delta}) = \operatorname{argmin}_{\lambda} \|\partial L(\boldsymbol{\delta}) - \lambda \partial c(\boldsymbol{\delta})\|.$$

There are two key quantities. The first quantity can be viewed as an approximate gradient when we have constraints, which we will denote as Γ:

$$\Gamma(\boldsymbol{\delta}) = \partial L(\boldsymbol{\delta}, \lambda)|_{(\boldsymbol{\delta}, \lambda^*(\boldsymbol{\delta}))} = \frac{\partial L(\boldsymbol{\delta})}{\partial \boldsymbol{\delta}} - \lambda^*(\boldsymbol{\delta}) \frac{\partial c(\boldsymbol{\delta})}{\partial \boldsymbol{\delta}}.$$

Another important quantity can be viewed as the approximate Hessian of constraint optimization:

$$\Xi(\boldsymbol{\delta}) = \partial^2 L(\boldsymbol{\delta}, \lambda)|_{(\boldsymbol{\delta}, \lambda^*(\boldsymbol{\delta}))} = \frac{\partial^2 L(\boldsymbol{\delta})}{\partial \boldsymbol{\delta}^2} - \lambda^*(\boldsymbol{\delta}) \frac{\partial^2 c(\boldsymbol{\delta})}{\partial \boldsymbol{\delta}^2}.$$

For $\boldsymbol{\delta}, \boldsymbol{\delta}' \in \varepsilon\mathbb{S}^{d-1}$, if $\partial^2 L(\boldsymbol{\delta}, \lambda^*)$ is ρ-Lipschitz, i.e. $\|\partial^2 L(\boldsymbol{\delta}_a, \lambda^*) - \partial^2 L(\boldsymbol{\delta}_b, \lambda^*)\| \leqslant \rho \|\boldsymbol{\delta}_a - \boldsymbol{\delta}_b\|$ for all $\boldsymbol{\delta}_a, \boldsymbol{\delta}_b \in \mathbb{B}(\mathbf{0}, \varepsilon)$, we can obtain

$$\begin{aligned} L(\boldsymbol{\delta}, \lambda^*) \leqslant{} & L(\boldsymbol{\delta}', \lambda^*) + \partial L(\boldsymbol{\delta}', \lambda^*)^T(\boldsymbol{\delta} - \boldsymbol{\delta}') \\ & + \frac{1}{2}(\boldsymbol{\delta} - \boldsymbol{\delta}')^T \partial^2 L(\boldsymbol{\delta}', \lambda^*)(\boldsymbol{\delta} - \boldsymbol{\delta}') + \frac{\rho}{6}\|\boldsymbol{\delta} - \boldsymbol{\delta}'\|^3. \end{aligned}$$

Since $\boldsymbol{\delta}, \boldsymbol{\delta}'$ are on the sphere, we know $L(\boldsymbol{\delta}, \lambda^*) = L(\boldsymbol{\delta})$ and $L(\boldsymbol{\delta}', \lambda^*) = L(\boldsymbol{\delta}')$, we have

$$\begin{aligned} L(\boldsymbol{\delta}) \leqslant{} & L(\boldsymbol{\delta}') + \Gamma(\boldsymbol{\delta}')^T(\boldsymbol{\delta} - \boldsymbol{\delta}') + \frac{1}{2}(\boldsymbol{\delta} - \boldsymbol{\delta}')^T \Xi(\boldsymbol{\delta}')(\boldsymbol{\delta} - \boldsymbol{\delta}') \\ & + \frac{\rho}{6}\|\boldsymbol{\delta} - \boldsymbol{\delta}'\|^3. \end{aligned} \tag{4}$$

Further, we denote $\mathcal{T}(\boldsymbol{\delta})$ as the tangent space at $\boldsymbol{\delta}$ on the sphere, and $\mathcal{P}_{\mathcal{T}(\boldsymbol{\delta})}$ is the operator for projection to the tangent space $\mathcal{T}(\boldsymbol{\delta})$. The projected gradient descent can be approximated in the manner stated in the following lemma.

Lemma 4.2 (**Approximation of PGD**). *For any $\hat{v} \in \mathbb{S}^{d-1}$, let $\tilde{\boldsymbol{\delta}}_1 = \boldsymbol{\delta}_0 + \eta\hat{v}$ and $\tilde{\boldsymbol{\delta}}_2 = \boldsymbol{\delta}_0 + \eta\mathcal{P}_{\mathcal{T}_0} \cdot \hat{v}$*

$$\|\mathcal{P}_{\mathbb{B}(\mathbf{0},\varepsilon)}(\tilde{\boldsymbol{\delta}}_1) - \tilde{\boldsymbol{\delta}}_2\| \leqslant \frac{4\eta^2}{\varepsilon}.$$

It is worth noting that $\Gamma(\boldsymbol{\delta})$ is actually the tangent component of $\partial f(\boldsymbol{\delta})$

$$\Gamma(\boldsymbol{\delta}) = \mathcal{P}_{\mathcal{T}(\boldsymbol{\delta})} \cdot \partial f(\boldsymbol{\delta}).$$

As a result, for $\boldsymbol{\delta}_t \in \varepsilon\mathbb{S}^{d-1}$

$$\|\boldsymbol{\delta}_{t+1} - (\boldsymbol{\delta}_t - \eta\Gamma(\boldsymbol{\delta}_t))\| \leqslant \frac{4\eta^2}{\varepsilon}. \tag{5}$$

We can use the above Eq. (4) and (5) to calculate the progress at each step. Thus, in order to analyze the progress, we only need to carefully analyze Γ and Ξ. In the following paragraph, we discuss Γ and Ξ case by case.

For each point on the sphere, we loosely define "near" and "away from" local optimums by looking into the angle between the gradient and the spherical normal vector. If the gradient is parallel to the spherical normal vector at a point on the sphere, then the point is a fixed point for projected gradient descent. It is either a local optimum or a saddle point. We will show such points are not saddle points under some regularity conditions. Since $c(\boldsymbol{\delta}) = \|\boldsymbol{\delta}\|^2 - \varepsilon^2$, the unit spherical normal vector is $\boldsymbol{\delta}/\|\boldsymbol{\delta}\|$ at each point on the sphere and the cosine value of the angle we are looking at is $\angle[\partial f(\boldsymbol{\delta}), \boldsymbol{\delta}]$. If $\angle[\partial f(\boldsymbol{\delta}), \boldsymbol{\delta}]$ is close to ±1, then such $\boldsymbol{\delta}$ is close to a critical point.

Lemma 4.3 (**Away from critical points on the sphere**). *For $m = \Omega(d^{5/2})$, there exists a threshold $\varepsilon_{\max}(m) = \Theta((\log m)^{-1})$, if $\varepsilon < \varepsilon_{\max}$, with high probability, for any $\boldsymbol{\delta} \in \varepsilon\mathbb{S}^{d-1}$ and any $0 \leqslant \beta \leqslant 1$ such that*

$$\angle[\partial f(\boldsymbol{\delta}), \boldsymbol{\delta}] \leqslant \beta,$$

we have

$$\|\Gamma(\boldsymbol{\delta})\| \geqslant \sqrt{1-\beta^2}\|\partial f(\boldsymbol{\delta})\| \geqslant \mathcal{L}B_l\sqrt{1-\beta^2},$$

where B_l is of order $\Theta(1)$.

Recall $\mathcal{L}$ is the lower bound such that $|y - f(\boldsymbol{a}, \boldsymbol{W}, \boldsymbol{\delta} + \boldsymbol{x})| > \mathcal{L}$ for all $\boldsymbol{\delta} \in \mathbb{B}(\mathbf{0}, \varepsilon)$. The above lemma shows if the trajectory is away from critical points, each step can decrease the loss value by $-\Omega(\eta)$ since $\boldsymbol{\delta}_{t+1} \approx \boldsymbol{\delta}_t - \eta\Gamma(\boldsymbol{\delta}_t)$ and $L(\boldsymbol{\delta}_{t+1}) \leqslant L(\boldsymbol{\delta}_t) + \Gamma(\boldsymbol{\delta}_t)^T(\boldsymbol{\delta}_{t+1} - \boldsymbol{\delta}_t) + O(\|\boldsymbol{\delta}_{t+1} - \boldsymbol{\delta}_t\|^2)$.

The hard case is when the trajectory is near a critical point on the sphere. We will first show that the critical points on the sphere are not saddle points under some regularity conditions.

Lemma 4.4 (**Near critical points on the sphere**). *For $m = \Omega(d^{5/2})$, there exists a threshold $\varepsilon_{\max}(m) = \Theta((\log m)^{-1})$, if $\varepsilon < \varepsilon_{\max}$, with high probability, there exists universal constants $\phi, \gamma > 0$, for any $\boldsymbol{\delta} \in \varepsilon\mathbb{S}^{d-1}$, such that*

$$\angle[\partial f(\boldsymbol{\delta}), \boldsymbol{\delta}] \geqslant \phi,$$

then for all $\|\boldsymbol{v}\| = 1$,

$$sgn\left((y-u)\boldsymbol{\delta}^T\partial f(\boldsymbol{\delta})\right) \cdot \boldsymbol{v}^T\Xi\boldsymbol{v} \geqslant \gamma.$$

Lemma 4.4 implies Ξ is either positive definite or negative definite near a critical point, thus, none of the critical points on the sphere are saddle points.

Since near a local minimum, the trajectory can converge to that local minimum by traditional analysis technique, the only thing left to deal with is when the trajectory is near local maximums. The following lemma states the trajectory will not be stuck near any local maximum with high probability.

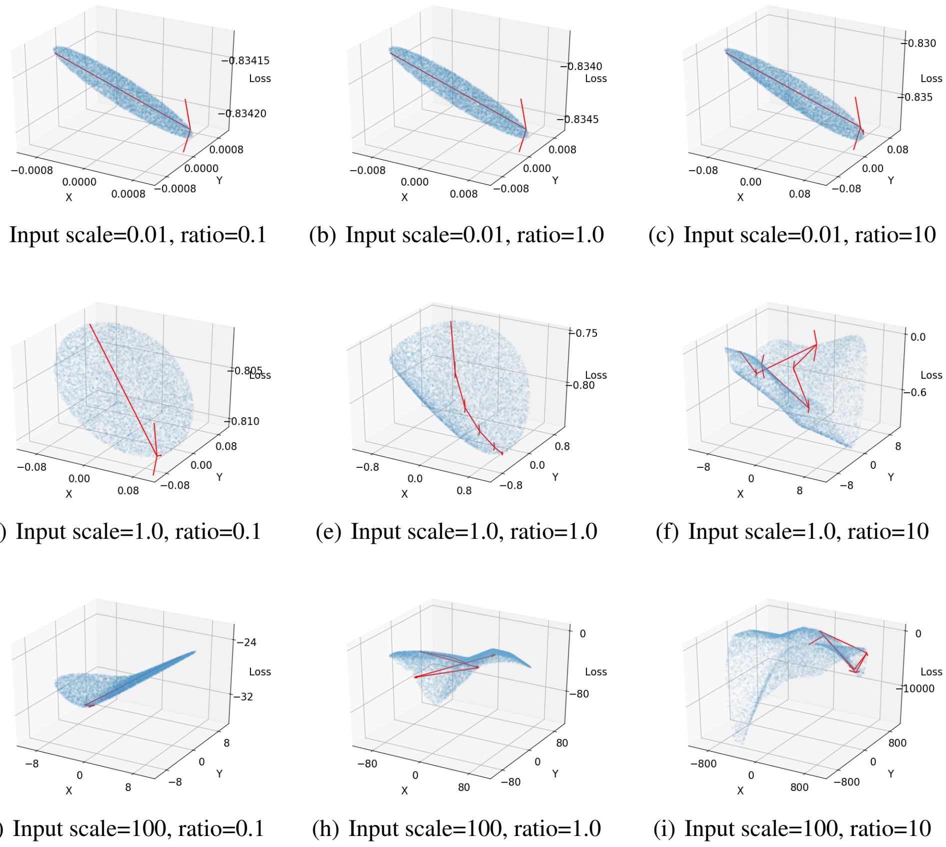
We denote the set $\Delta_\eta^- = \{\boldsymbol{\delta} : \angle[\partial f(\boldsymbol{\delta}), \boldsymbol{\delta}] \leqslant -1 + \sqrt{\eta}/(\mathcal{L}B_l), \boldsymbol{\delta} \in \varepsilon\mathbb{S}^{d-1}\}$ and $\Delta_\eta^+ = \{\boldsymbol{\delta} : \angle[\partial f(\boldsymbol{\delta}), \boldsymbol{\delta}] \geqslant 1 - \sqrt{\eta}/(\mathcal{L}B_l), \boldsymbol{\delta} \in \varepsilon\mathbb{S}^{d-1}\}$. Notice that $\angle[\partial f(\boldsymbol{\delta}), \boldsymbol{\delta}] = \pm1$ when the spherical normal vector is parallel to the gradient at $\boldsymbol{\delta}$. Thus, for small η, the two sets are the collections of points near local maximums and local minimums respectively.

Lemma 4.5 (**Trajectory and local optimums**). *For learning rate η such that $\eta < \min\{1, \mathcal{L}B_l\}$, if*

$$\arccos\left(\angle[\partial f(\boldsymbol{\delta}), \partial f(\boldsymbol{\delta}')]\right) + \arccos\left(\sqrt{\frac{(\mathcal{L}B_l)^2 - \eta}{(\mathcal{L}B_l)^2}}\right) \leqslant \frac{\pi}{4} \tag{6}$$

for all $\boldsymbol{\delta}, \boldsymbol{\delta}' \in \varepsilon\mathbb{S}^{d-1}$, the trajectory initialized by drawing from a uniform distribution over $\mathbb{B}^\circ(\mathbf{0}, \varepsilon)$ will never reach Δ_η^-. Meanwhile, if there exists t^* such that $\boldsymbol{\delta}_{t^*} \in \Delta_\eta^+$, then for all $t \geqslant t^*$, $\boldsymbol{\delta}_t \in \Delta_\eta^+$.

From the discussions above, it is easy to see Lemma 4.5 holds for small enough ε and η. The above lemma states the trajectory will not be stuck near local maximums and $\|\Gamma(\boldsymbol{\delta})\| \geqslant \sqrt{\eta}$ if $\boldsymbol{\delta} \notin \Delta_\eta^+$ for $\boldsymbol{\delta} \in \varepsilon\mathbb{S}^{d-1}$. That can ensure $L(\boldsymbol{\delta}_{t+1}) - L(\boldsymbol{\delta}_t) \leqslant -\Omega(\eta^2)$ for $\boldsymbol{\delta}_t, \boldsymbol{\delta}_{t+1} \in \varepsilon\mathbb{S}^{d-1}$. As a result, the trajectory can constantly make progress until the trajectory reaches Δ_η^+. Then, traditional techniques for convex optimization can be applied and gives us the final convergence result.



(a) Input scale=0.01, ratio=0.1
(b) Input scale=0.01, ratio=1.0
(c) Input scale=0.01, ratio=10
(d) Input scale=1.0, ratio=0.1
(e) Input scale=1.0, ratio=1.0
(f) Input scale=1.0, ratio=10
(g) Input scale=100, ratio=0.1
(h) Input scale=100, ratio=1.0
(i) Input scale=100, ratio=10

Figure 1. Landscapes and trajectories on simulated data. We compare the landscapes and trajectories with three different input scales and three different perturbation ratios. If the input scale is small enough (i.e. 0.01 here), the landscape has only one local minimum and PGD can easily escape the local maximal with few steps even with large perturbation ratio such as 10 . On the other hand, if the input scale is not small enough, we will have a less regular landscape with a lot of local minimums and it takes a lot of steps to escape from a local maximum. For large input scales, we have to reduce the perturbation ratio, so as to make the landscape become more regular and make escaping from the local maximums faster. Our simulations are based on two-dimensional inputs and two-layer neural networks. More details can be found in the supplementary materials.

5. Implications and Extensions

So far, we have derived the theory about finding adversaries in the initial phase of adversarial training. Through our theoretical analysis, we also identify several interesting phenomena concerning the scale of input $\boldsymbol{x}$. In this section, we briefly discuss the implications of our theory on experiments and show how to extend our arguments to general losses.

5.1. Scale, Landscape and Convergence

In this subsection, we state the high level conclusions and the details of the theoretical results are left in the supplementary materials.

As we stated in the introduction, ε's scale is usually formulated in proportional to the scale of input $\boldsymbol{x}$. In the empirical optimization (1)

$$\min_{\boldsymbol{\theta}\in\Theta} \frac{1}{n}\sum_{i=1}^{n} \max_{\forall i\in[n],\|\boldsymbol{\delta}_i\|_p\leqslant\varepsilon} L(\boldsymbol{\theta}, \boldsymbol{x}_i+\boldsymbol{\delta}_i, y_i),$$

ε takes the form

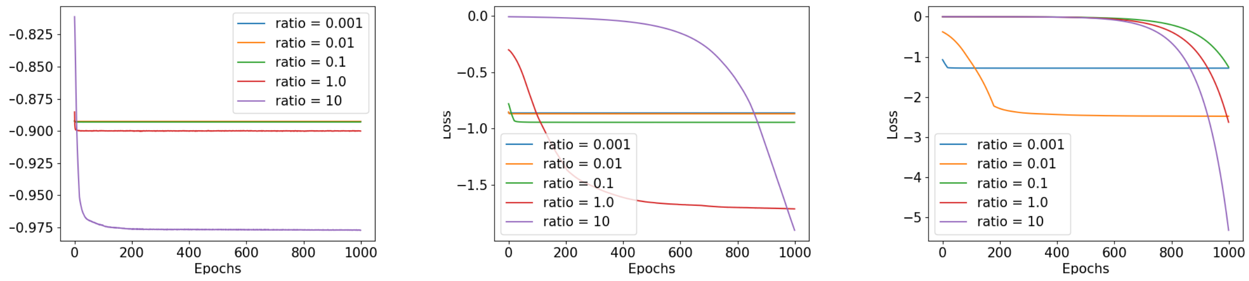
$$\varepsilon = r\sum_{i=1}^{n}\frac{\|\boldsymbol{x}_i\|_p}{n}$$

for small $r>0$, where r stands for a small constant ratio.

In this section, we shed some light on Question 2, which we restate here.

When we fix the ratio r, do smaller input scales (implying smaller ε) help optimization of adversarial training?

Our answer to that question is positive — at least the input scale matters in the initial phase of adversarial training.



(a) Input scale=0.1 (b) Input scale=10 (c) Input scale=1000

Figure 2. Trajectories from local maxima to local minima on real-world data. We show the adversarial losses of each point on the trajectories from local maxima with three different input scales and five different perturbation ratios. For a fixed perturbation ratio, a smaller input scale means that escaping from local maxima is easier. If the input scale is small enough (i.e. 0.1 here), PGD can easily escape the local maxima even with a large perturbation ratio such as 10 as shown in Fig. 2(a). If the input scale is not small enough, escaping from a local maximum will be easier with a smaller perturbation ratio as shown in Fig. 2(b). If the input scale is too large, escaping from a local maximum will be difficult even with a small perturbation ratio 0.001 as shown in Fig. 2(c). These results are consistent with those on the simulated data in Fig. 1. The experiments are based on a real-world dataset MNIST and a practical multi-layer CNN. More details can be found in the supplementary materials.

We experimentally and theoretically answer that question from the perspectives of landscapes and convergence of trajectories.

5.1.1. Smaller input scales imply more regular landscapes

In our proofs, the concentration results for all quantities such as $\sup_{\boldsymbol{\delta}\in\mathbb{B}(\mathbf{0},\varepsilon)}\|\partial f(\boldsymbol{a},\boldsymbol{W},\boldsymbol{\delta}+\boldsymbol{x})/\partial\boldsymbol{\delta}\|$ and $\min_{\boldsymbol{\delta},\boldsymbol{\delta}'\in\mathbb{B}(\mathbf{0},\varepsilon)}\angle[\partial f(\boldsymbol{\delta}),\partial f(\boldsymbol{\delta}')]$ depend only on the scale of ε since in the initial phase, $\boldsymbol{a}$ and $\boldsymbol{W}$ are drawn from initialization distributions which are independent to the inputs. That fact implies with a fixed ratio r, a smaller input scale will result in a smaller ε, so as to make all the concentration results hold with a higher probability. Even if the ratio r is large, which means the adversarial attack is more aggressive, the concentration results can hold regardlessly.

Moreover,

$$\min_{\boldsymbol{\delta},\boldsymbol{\delta}'}\angle[\partial f(\boldsymbol{\delta}),\partial f(\boldsymbol{\delta}')]\to 1,$$

as $\varepsilon\to 0$, which means the angle between $\partial f(\boldsymbol{\delta})$ and $\partial f(\boldsymbol{\delta}')$ will be very small if ε is small. Besides, for $\boldsymbol{\delta}\in\varepsilon\mathbb{S}^{d-1}$, $\boldsymbol{\delta}$ is a local optimum if and only if $\boldsymbol{\delta}$ is parallel to $\partial f(\boldsymbol{\delta})$. Combining the above facts, it is natural to expect the local minimums will be closer to each other when a smaller ε is chosen. Actually, there is a threshold $\tau_\varepsilon>0$, when ε is smaller than τ_ε, there is only one minimum on the sphere.

Theorem 5.1 (Informal). *Under the settings of Theorem 3.1, there exists a threshold $\tau_\varepsilon>0$, such that for $\varepsilon<\tau_\varepsilon$, there is only one local minimum on the sphere with high probability.*

Theorem 5.1 implies in the initial phase of adversarial training, a smaller input scale of $\|\boldsymbol{x}\|$ actually can ensure there exists only one single local minimum on the sphere which is also the global minimum. Combined with previous results, the projected gradient descent is able to reach global minimum with high probability.

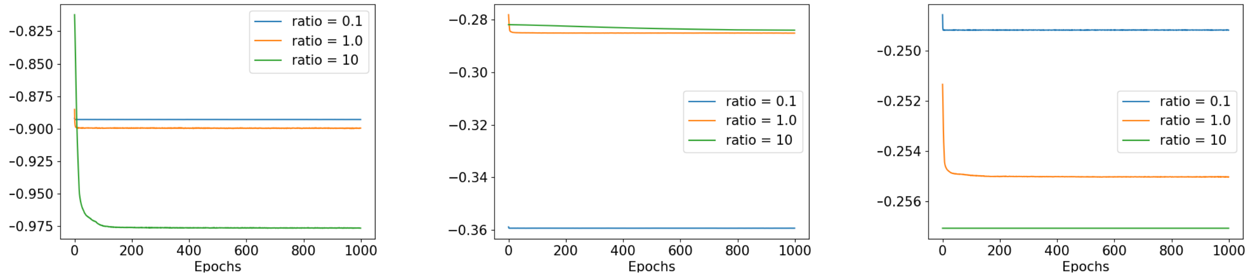
In Figure 1, we can see for a fixed r, smaller input scale make the landscape more regular, for instance, the upper left one has only one local minimum. For a large input scale, the landscape will become very complex (see subfigure (i)) unless we use very small perturbation ratio r (see subfigure (g)) .

5.1.2. Smaller input scales help convergence

Another interesting discovery is inspired by Lemma 4.5 in the previous section. If ε is not small enough, Eq. (6) in Lemma 4.5 cannot stand. Thus, when the initial adversary $\boldsymbol{\delta}_0\in\mathbb{B}^\circ(\mathbf{0},\varepsilon)$ is close to one of the local maximums on the sphere, it is possible that the trajectory of projected gradient descent can reach the region Δ_η^- on the sphere, who contains points close to local maximums. Too close to a local maximum will result in a very small progress in the loss decay at each step, which will take much longer to reach a local minimum. As an illustration, we can see from Figure 2 and 3, by judging from the decay rate of loss function, we can see a smaller input scale leads to faster loss value decay in the initial phase of adversarial training.

5.2. General losses

Previously, we have derived the theory with respect to quadratic loss. In this subsection, we extend the theory



(a) Epoch=0

(b) Epoch=10

(c) Epoch=100

Figure 3. The dynamics of trajectories from local maxima to local minima during the adversarial training process. We use the same setting as that in Figure 2 and fix the input scale as 0.1. After a few epochs of adversarial training, escaping from local maxima is still easy for the small input scale as shown in Fig. 3(b). However, escaping from local maxima will be significantly harder after a lot of training epochs as shown in Fig. 3(c). This influence is more significant on large input scales compared to small ones.

to general losses of $(\boldsymbol{x}, y) \in \mathbb{R}^d \times \mathbb{R}$ in the following form:

$$L(y, f(\boldsymbol{\theta}, \boldsymbol{x} + \boldsymbol{\delta})),$$

where we still take f as a two-layer neural network discussed previously:

$$f(\boldsymbol{a}, \boldsymbol{W}, \boldsymbol{\delta} + \boldsymbol{x}) = \sum_{r=1}^{m} a_r \sigma(\boldsymbol{w}_r^T(\boldsymbol{x} + \boldsymbol{\delta})).$$

Taking derivative with respect to $\boldsymbol{\delta}$:

$$\frac{\partial L}{\partial \boldsymbol{\delta}} = \frac{\partial L}{\partial f} \cdot \frac{\partial f}{\partial \boldsymbol{\delta}}, \quad \frac{\partial^2 L}{\partial \boldsymbol{\delta}^2} = \frac{\partial L}{\partial f} \cdot \frac{\partial^2 f}{\partial \boldsymbol{\delta}^2} + \frac{\partial^2 L}{\partial f^2} \cdot \frac{\partial f}{\partial \boldsymbol{\delta}} \left(\frac{\partial f}{\partial \boldsymbol{\delta}} \right)^T.$$

Actually the only difference of deriving theory for general losses compared to quadratic losses lies in the different form of $\partial L/\partial \boldsymbol{\delta}$. As long as $\partial L/\partial \boldsymbol{\delta}$ satisfies $\mathcal{L} < |\partial L/\partial f| < \mathcal{U}$ for all $\boldsymbol{\delta} \in \mathbb{B}(\boldsymbol{0}, \varepsilon)$ for some $\mathcal{L}, \mathcal{U} > 0$, and $|\partial^2 L/\partial f^2|$ is upper bounded by some constant $\mathcal{B} > 0$, all our previous conclusions stand without changing the scale of ε and η. Instead of going into too many details, we leave the details to readers who are interested in checking. In the later paragraph, we focus on discussing whether the above assumptions are reasonable.

Generally, the loss chosen in the optimization has the following property: $L(y, f) = 0$ if and only if $y = f$. The final goal of optimization is to make $L(y, f)$ small and in the initial phase, since we initialize the parameters randomly, we would expect $f(\boldsymbol{\theta}, \boldsymbol{x} + \boldsymbol{\delta})$ to be "far from" the label y, in other words, $|L(y, f)|$ is lower bounded by some positive constant $\mathcal{L}$. Then, by continuity of the loss function, if ε is small, the change of $|L(y, f)|$ would be expected to be small. As a result, it is reasonable to assume $\partial L/\partial \boldsymbol{\delta}$ satisfies $\mathcal{L} < |\partial L/\partial f| < \mathcal{U}$ for all $\boldsymbol{\delta} \in \mathbb{B}(\boldsymbol{0}, \varepsilon)$ for some $\mathcal{L}, \mathcal{U} > 0$. Also, with smoothness assumptions on L over f, and smoothness assumptions on f over input $\boldsymbol{x}$, since the change of ε is over a compact set, $|\partial^2 L/\partial f^2|$ should be upper bounded.

We wrap up this subsection with another concrete example besides quadratic loss – cross entropy loss:

$$L(y, f) = -y \log \big(\frac{\exp(f)}{1 + \exp(f)} \big) - (1 - y) \log \big(\frac{1}{1 + \exp(f)} \big).$$

Then,

$$\frac{\partial L}{\partial f} = \frac{\exp(f)}{1 + \exp(f)} - y, \quad \frac{\partial^2 L}{\partial f^2} = \frac{\exp(f)}{(1 + \exp(f))^2}.$$

As discussed above, in the initial phase, we usually have the estimated probability $\exp(f)/(1 + \exp(f))$ is not equal to the true probability (here the true probability y is either 0 or 1). And with small $\varepsilon > 0$, we would expect $\partial L/\partial \boldsymbol{\delta}$ satisfies $\mathcal{L} < |\partial L/\partial f| < \mathcal{U}$ for all $\boldsymbol{\delta} \in \mathbb{B}(\boldsymbol{0}, \varepsilon)$ for some $\mathcal{L}, \mathcal{U} > 0$. Meanwhile, apparently $0 \leqslant \partial^2 L/\partial f^2 \leqslant 1$.

6. Conclusions and Future Work

In this paper, we theoretically characterize the dynamics of finding adversaries in two-layer fully connected neural networks in the initial phase of training. We also talk about the experimental implications the theory brings. The main take-away is that in the initial phase of adversarial training, projected gradient method is trustworthy and a smaller input scale can help the adversarial training perform better.

In the future, we hope to extend our theory to higher layer neural networks and to the full dynamics involving weight updates. When considering the full dynamics, as the adversarial training process goes on, the weights become more and more dissimilar to gaussian vectors. Usually, as the adversarial training goes on, $L(y, f)$ will goes to 0, so we can expect the convergence rate on finding adversaries will be slower since $\frac{\partial L}{\partial \boldsymbol{\delta}} = \frac{\partial L}{\partial f} \cdot \frac{\partial f}{\partial \boldsymbol{\delta}}$ and $\frac{\partial L}{\partial \boldsymbol{\delta}}$ should be close to 0.

The landscape of adversaries in the later phase of training will become very complicated due to the intervention of δ and θ. More importantly, using first order optimization method is possible to result in a cyclic dynamic. It is also interesting to explore how to get rid of the cyclic dynamic problem in the future.

Acknowledgements

This work is in part supported by NSF award 1763665.

References

Agarwal, N., Gonen, A., and Hazan, E. Learning in non-convex games with an optimization oracle. *arXiv preprint arXiv:1810.07362*, 2018.

Athalye, A., Carlini, N., and Wagner, D. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. *arXiv preprint arXiv:1802.00420*, 2018.

Brendel, W., Rauber, J., and Bethge, M. Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. *arXiv preprint arXiv:1712.04248*, 2017.

Carlini, N. and Wagner, D. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, pp. 39–57. IEEE, 2017.

Chen, P.-Y., Zhang, H., Sharma, Y., Yi, J., and Hsieh, C.-J. Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, pp. 15–26, 2017.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Du, S. S., Jin, C., Lee, J. D., Jordan, M. I., Singh, A., and Poczos, B. Gradient descent can take exponential time to escape saddle points. In *Advances in neural information processing systems*, pp. 1067–1077, 2017.

Du, S. S., Zhai, X., Poczos, B., and Singh, A. Gradient descent provably optimizes over-parameterized neural networks. *arXiv preprint arXiv:1810.02054*, 2018.

Gao, R., Cai, T., Li, H., Wang, L., Hsieh, C.-J., and Lee, J. D. Convergence of adversarial training in overparametrized networks. *arXiv preprint arXiv:1906.07916*, 2019.

Ge, R., Huang, F., Jin, C., and Yuan, Y. Escaping from saddle pointsonline stochastic gradient for tensor decomposition. In *Conference on Learning Theory*, pp. 797–842, 2015.

Glorot, X. and Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 249–256, 2010.

Goodfellow, I. J., Shlens, J., and Szegedy, C. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

Guo, C., Rana, M., Cisse, M., and Van Der Maaten, L. Countering adversarial images using input transformations. *arXiv preprint arXiv:1711.00117*, 2017.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Ilyas, A., Santurkar, S., Tsipras, D., Engstrom, L., Tran, B., and Madry, A. Adversarial examples are not bugs, they are features. In *Advances in Neural Information Processing Systems*, pp. 125–136, 2019.

Jacot, A., Gabriel, F., and Hongler, C. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in neural information processing systems*, pp. 8571–8580, 2018.

Jin, C., Ge, R., Netrapalli, P., Kakade, S. M., and Jordan, M. I. How to escape saddle points efficiently. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 1724–1732. JMLR. org, 2017.

Liu, X. and Hsieh, C.-J. Rob-gan: Generator, discriminator, and adversarial attacker. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 11234–11243, 2019.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

Papernot, N., McDaniel, P., Goodfellow, I., Jha, S., Celik, Z. B., and Swami, A. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia conference on computer and communications security*, pp. 506–519, 2017.

Shafahi, A., Najibi, M., Ghiasi, M. A., Xu, Z., Dickerson, J., Studer, C., Davis, L. S., Taylor, G., and Goldstein, T. Adversarial training for free! In *Advances in Neural Information Processing Systems*, pp. 3353–3364, 2019.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

Wang, Y., Ma, X., Bailey, J., Yi, J., Zhou, B., and Gu, Q. On the convergence and robustness of adversarial training. In *International Conference on Machine Learning*, pp. 6586–6595, 2019.

Weng, T.-W., Zhang, H., Chen, H., Song, Z., Hsieh, C.-J., Boning, D., Dhillon, I. S., and Daniel, L. Towards fast computation of certified robustness for relu networks. *arXiv preprint arXiv:1804.09699*, 2018.

Wong, E., Schmidt, F., Metzen, J. H., and Kolter, J. Z. Scaling provable adversarial defenses. In *Advances in Neural Information Processing Systems*, pp. 8400–8409, 2018.

Yin, D., Ramchandran, K., and Bartlett, P. Rademacher complexity for adversarially robust generalization. *arXiv preprint arXiv:1810.11914*, 2018.

Zhang, H., Weng, T.-W., Chen, P.-Y., Hsieh, C.-J., and Daniel, L. Efficient neural network robustness certification with general activation functions. In *Advances in neural information processing systems*, pp. 4939–4948, 2018.