Robust Learning for Data Poisoning Attacks

Yunjuan Wang [1] **Poorya Mianjy** [1] **Raman Arora** [1]

Abstract

We investigate the robustness of stochastic approximation approaches against data poisoning attacks. We focus on two-layer neural networks with ReLU activations and show that under a specific notion of separability in the RKHS induced by the infinite-width network, training (finite-width) networks with stochastic gradient descent is robust against data poisoning attacks. Interestingly, we find that in addition to a lower bound on the width of the network, which is standard in the literature, we also require a distribution-dependent upper bound on the width for robust generalization. We provide extensive empirical evaluations that support and validate our theoretical results.

1. Introduction

Machine learning models based on neural networks power the state-of-the-art systems for various real-world applications, including self-driving autonmous vehicles (Grigorescu et al., 2020), speech recognition (Afouras et al., 2018), reinforcement learning (Li, 2017), etc. Neural networks trained using stochastic gradient descent (SGD) perform well both in terms of optimization (training) and generalization (prediction). However, with great power comes great responsibility, and as several recent studies indicate, systems based on neural networks admit vulnerabilities in the form of adversarial attacks. Especially in over-parametrized settings (wherein the number of parameters is much larger than training sample size), which is typical in most applications, neural networks remain extraordinarily fragile and amenable to depart from their expected behavior due to strategically induced perturbations in data. One such limitation is due to arbitrary adversarial corruption of data at the time of training, commonly referred to as *data poisoning*. Such attacks present a challenging problem, especially in settings where an adversary can affect any part of the training data. Therefore, in this paper, we are interested in quantifying the maximal adversarial noise that is tolerable by SGD when training wide ReLU networks.

One of the earliest works to consider provably tolerant algorithms to a quantifiable error in training examples was that of Valiant (1985), motivated by a need to understand the limitations of the PAC learning framework. This was followed by a series of works that considered computationally unbounded adversaries and posed the question of bounding the error rate tolerable by a learning algorithm in a worst case model of errors (Kearns & Li, 1993; Guruswami & Raghavendra, 2009). These hardness results were later complemented by positive results (Klivans et al., 2009; Awasthi et al., 2014; Diakonikolas et al., 2019a), which give learning algorithms that enjoy information theoretically optimal noise tolerance. Much of this prior work focuses on learning halfspaces (i.e., linear separators) in Valiant's PAC learning model (Valiant, 1984). Instead, we consider Vapnik's general learning, and are interested in convex learning problems and over-parametrized neural networks with ReLU activations. While our theoretical understanding of deep learning has increased vastly in the last few years with several results characterizing the ability of gradient descent to achieve small training loss in over-parameterized regime, our understanding of robustness of such methods to attacks such as data poisoning remains limited.

Arguably, a simplest model of data poisoning is one in which the input features are perturbed, additively, by norm-bounded vectors. A more challenging scenario is where both input features and labels can be corrupted – this is essentially the noise model considered by Valiant (1985); Kearns & Li (1993); Awasthi et al. (2014). A related model studied by Cesa-Bianchi et al. (2011) is one where the learner observes only a noisy version of the data, in a streaming setting, with noise distribution changing arbitrarily after each round. A yet another poisoning attack, studied extensively in the literature, is where the adversary can plant a fraction of the training data; for example, consider movie ratings contributed by malicious users in matrix completion. Recent works have studied numerous other practical data poisoning methods including backdoor attacks, data injection, clean label attacks, and flip-label attacks (we discuss these further in related work).

[1]Department of Computer Science, Johns Hopkins University, Baltimore, MD, USA. Correspondence to: Raman Arora <arora@cs.jhu.edu>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021.

While several defenses have been proposed, each tailored to a specific data poisoning attack, there is no unified, robust learning framework against such attacks. Furthermore, the proposed defenses often depart significantly from the practice of modern machine learning, which increasingly relies on stochastic approximation algorithms such as stochastic gradient descent (SGD), stochastic mirror descent, and variants. Therefore, it is natural to ask whether stochastic approximation algorithms, such as SGD, impart robustness to learning against adversarial perturbations of training data.

In this paper, we investigate the robustness of SGD against various data poisoning attacks for convex learning problems as well as training two-layer over-parameterized neural networks with ReLU activations. Surprisingly, our results show that SGD achieves optimal convergence rates on the excess risk, despite data poisoning, with only a mild deterioration in overall performance, even as the overall noise budget of the adversarial attack grows with the sample size, albeit sublinearly. Our main contributions in this paper are as follows.

- In Section 2, we first consider the clean label attack, where the adversary can additively perturb the input features but not the target labels. In this setting, we show that stochastic gradient descent robustly learns a classifier as long as the overall perturbation is sublinear in the sample size. We extend our results to a more general class of data poisoning attacks and study them in a unified framework of *oracle poisoning*.
- In Section 3, we extend our results to two-layer over-parameterized neural networks with ReLU activations. We discuss clean label attack and label flip attack separately, and establish guarantees for SGD in three regimes under a data-dependent margin assumption. Our bounds hold in the regime where neural networks are moderately wide but not too wide, supporting the conjecture that extreme over-parametrization may render learning susceptible to data poisoning. This is in stark contrast to existing results in deep learning theory that argue for wider networks for better generalization.
- We validate our theoretical results with empirical evaluations on real datasets in Section 5. We confirm that the clean-test accuracy exhibits an inverted U-curve when the training data is poisoned in all of the noisy regimes we consider. In the process, we also discover a new loss function that yields stronger poisoning attacks, which might be of independent interest in itself.

1.1. Problem Setup

We focus on the task of binary classification in presence of data poisoning attacks. We denote the input and the label spaces, respectively, by $\mathcal{X} \subseteq \mathbb{R}^d$ and $\mathcal{Y} = \{-1, +1\}$. We assume that the data (x, y) are drawn from an unknown joint distribution $\mathcal{D}$ on $\mathcal{X} \times \mathcal{Y}$. In a general (clean-data) learning framework, the learner is provided with n i.i.d. samples $\mathcal{S} = \{(\mathrm{x}_i, y_i)\}_{i=1}^n \sim \mathcal{D}^n$, and the goal is to learn a function $f_{\mathrm{w}} : \mathcal{X} \to \mathcal{Y}$, parameterized by w in some parameter space $\mathcal{W}$, with a small generalization error, i.e., small 0-1 loss with resepct to the population, $L(\mathrm{w}) := \mathbb{P}_{(\mathrm{x},y)\sim\mathcal{D}}(y f_{\mathrm{w}}(\mathrm{x}) \leq 0)$.

We model the data poisoning attacks as a malicious adversary who sits between the distribution and the learner. The adversary receives an i.i.d. sample $\mathcal{S} := \{(\mathrm{x}_i, y_i)\}_{i=1}^n \mathcal{D}^n$ of size n, generates the poisoned sample $\tilde{\mathcal{S}} := \{(\tilde{\mathrm{x}}_i, \tilde{y}_i)\}_{i=1}^n$, and passes it over to the learner. For example, in clean label attack, the adversary perturbs the input as $\tilde{\mathrm{x}}_i = \mathrm{x}_i + \delta_i$, where each perturbation δ_i belongs to a perturbation space Δ, and leaves the labels intact, i.e. $\tilde{y}_i = y_i$. Note that in this model, no distributional assumptions are made on the adversarial perturbations. Another example is the label flip attacks, whereby the adversary does not poison the input, i.e. $\tilde{\mathrm{x}}_i = \mathrm{x}_i$, but it flips the sign of the labels with probability β. More precisely, $\tilde{y}_i = -y_i$ with probability β and $\tilde{y}_i = y_i$ otherwise. We focus on the setting where the adversary has access to the clean data $\mathcal{S}$ and is computationally unbounded. In other words, adversary chooses to attack the optimal model (e.g., the empirical risk minimizer), given the sample. However, the adversary has no knowledge of the random bits used by the learner, e.g., when training using stochastic gradient descent.

A common approach to the clean-data learning problem is solving the stochastic optimization problem

$$\min_{\mathrm{w}\in\mathcal{W}} F(\mathrm{w}) := \mathbb{E}_{\mathcal{D}}[\ell(y f(\mathrm{x}; \mathrm{w}))],$$

where $\ell : \mathbb{R} \to \mathbb{R}_{\geq 0}$ is a convex surrogate loss for the 0-1 loss. In practice, this is usually done using first-order optimization techniques such as stochastic gradient descent (SGD) and its variants. The statistical and computational learning theoretic aspects of such methods has been extensively studied in the literature; however, their robustness to data poisoning attacks is yet not well-understood. Therefore, the central question we ask is the following: *"can SGD robustly and efficiently learn certain hypothesis classes?"*

In full generality, of course, the answer to the above question is negative – no learning is possible if we don't impose any restrictions on the perturbations, i.e., the set Δ. Therefore, in this paper, we identify conditions on the perturbations under which SGD can efficiently and robustly learn important hypothesis classes such as linear models as well as two-layer neural networks. In particular, our analysis crucially depends on the following measures of perturbations: 1) the *per-sample corruption budget* $B := \max_i \|\delta_i\|$; 2) the *overall corruption budget* $S := \sum_{i=1}^n \|\delta_i\|$; or 3) the *probability of label flip* β.

We denote scalars, vectors and matrices, respectively, with lowercase italics, lowercase bold and uppercase bold Roman letters, e.g. u, u and U. The ℓ_2 norm is denoted by $\|\cdot\|$. Throughout, we use the standard O-notation ($\mathcal{O}$ and Ω). Further, we use $\lesssim$ and $\mathcal{O}$ interchangeably. We use $\tilde{\mathcal{O}}$ to hide poly-logarithmic dependence on the parameters.

1.2. Related Work

In this section, we survey related prior work on data poisoning attacks and defense strategies, and on convergence analysis of gradient descent based methods for training wide networks.

Data poisoning attacks and defenses. A data poisoning attack, or causative attack, aims at manipulating training samples or model architecture, which leads to misclassification of subsequent input data associated with a specific label (a targeted attack) or manipulate predictions of data from all classes (an indiscriminate attack). A popular data poisoning attack is *backdoor attack*, where the adversary injects strategically manipulated samples (referred to as a *a backdoor pattern*, with a target label into the training data. At prediction time, samples that do not contain the trigger pattern can be categorized correctly, but samples that carry the trigger are likely misclassified as belonging to the target label class (Gu et al., 2017; Liu et al., 2017; Chen et al., 2017). One of the shortcomings of the standard backdoor attack is that the poisoned samples are clearly mislabeled, which can arouse suspicion if subjected to human inspection. This lead to what are known as *clean label attacks* research (Koh & Liang, 2017; Shafahi et al., 2018; Zhu et al., 2019), which focus on adding human imperceptible perturbations to input features without flipping labels of the corrupted inputs. Another attack category is that of *label-flip attacks*, where the adversary can change labels of a constant fraction of the training sample (Biggio et al., 2011; Xiao et al., 2012; Zhao et al., 2017).

Several defense mechanisms have been proposed to counter the data poisoning attacks described above. For the label-flip attacks, (Awasthi et al., 2014) focus on malicious noise model and construct an algorithm to find the optimal halfspace that achieves ϵ error while tolerating $\Omega(\epsilon)$ noise rate for isotropic log-concave distributions. Recently, (Diakonikolas et al., 2019a) proposes a poly $(d, 1/\epsilon)$ time algorithm to solve the same problem under Massart noise. For backdoor attacks, (Liu et al., 2018; Tran et al., 2018) propose strategies to identify the trigger pattern and target the poisoned samples. Several other works have followed up on this idea of data sensitization (outlier removal) (Barreno et al., 2010; Suciu et al., 2018; Jagielski et al., 2018; Diakonikolas et al., 2019b; Wang et al., 2019). For certified defense, (Steinhardt et al., 2017) analyze oracle defense and data-dependent defenses by constructing an approximate upper bound on the loss. Recently (Rosenfeld et al., 2020) apply randomized smoothing to build certifiable robust linear classifier against label-flip attack.

Convergence analysis of gradient descent for wide networks. Our analysis builds on recent advances in theoretical deep learning literature, which focuses on analyzing the trajectory of first-order optimization methods in the limit that the network width goes to infinity (Li & Liang, 2018; Du et al., 2019b;a; Allen-Zhu et al., 2018; Zou et al., 2018; Cao & Gu, 2019). The main insight from this body of work is that when training a sufficiently over-parameterized network using gradient descent, if the initialization is large and the learning rate is small, the weights of the network remain close to the initialization; therefore, the dynamics of the network predictions is approximately linear in the feature space induced by the gradient of the network at the initialization (Li & Liang, 2018; Chizat et al., 2018; Du et al., 2019b; Lee et al., 2019). We are particularly inspired by a recent work of (Ji & Telgarsky, 2019), which studies the setting where the data distribution is separable in this feature space, an assumption that was first introduced and studied in (Nitanda & Suzuki, 2019). While our assumptions and proof techniques are similar to this line of work, we are distinct in that – to the best of our knowledge – none of these prior works study the robustness of SGD to adversarial perturbations. Furthermore, while the existing results suggest that generalization error decreases as the width of the network increases, curiously, we find that robust generalization error exhibits a U-curve as a function of the network width. Our guarantees, accordingly, involve a lower bound and an upper bound on the size of over-parametrization of the network.

2. Warm-up: Convex Learning Problems

In convex learning problems, the parameter space $\mathcal{W}$ is a convex set, and the loss function $\ell(\cdot)$ is convex in w. This framework includes a simple yet powerful class of machine learning problems such as support vector machines and kernel methods. Here, we seek to understand the robustness of SGD based on corrupted (likely biased) gradient estimates $\nabla\ell(\tilde{y}f(\tilde{\mathrm{x}};\mathrm{w}))$ computed on poisoned data $(\tilde{\mathrm{x}}, \tilde{y})$. We begin with a simple observation that under standard regularity conditions, a bounded perturbation in the input/label domain translates to a bounded perturbation in the gradient domain; for example, in the clean label attacks, when $f(\mathrm{x};\mathrm{w}) = \langle \mathrm{w}, \mathrm{x}\rangle$ is a linear function, the following holds.

Proposition 2.1. Assume $\|\mathrm{w}\| \le D$ for all $\mathrm{w} \in \mathcal{W} \subseteq \mathbb{R}^d$, $\|\mathrm{x}\| \le R$ for all $\mathrm{x} \in \mathcal{X} \subseteq \mathbb{R}^d$, and the loss function $\ell(\cdot)$ is L-Lipschitz and α-smooth. Then, for any linear function $f(\mathrm{x};\mathrm{w}) = \langle \mathrm{w}, \mathrm{x}\rangle$, $\mathrm{w} \in \mathcal{W}$, the following holds for any $(\mathrm{x}, y) \in \mathcal{X} \times \mathcal{Y}$, and $\delta \in \mathbb{R}^d$.

$$\|\nabla\ell(yf(\mathrm{x}+\delta;\mathrm{w})) - \nabla\ell(yf(\mathrm{x};\mathrm{w}))\| \le (\alpha DR + L)\|\delta\|.$$

In fact, other poisoning attacks such as label flip attack can also be viewed in terms of poisoning of the first order information about the stochastic objective. In other words, various data poisoning attacks can be studied in a unified framework of *oracle poisoning* which we define formally, next.

Definition ((G, B)-PSFO)**.** Given a function $F : \mathcal{W} \to \mathbb{R}$, a *poisoned stochastic first-order oracle* for F takes $\mathrm{w} \in \mathcal{W}$ as input and returns a random vector $\tilde{\mathrm{g}}(\mathrm{w}) = \hat{\mathrm{g}}(\mathrm{w}) + \zeta$, where $\mathbb{E}[\hat{\mathrm{g}}(\mathrm{w})] \in \partial F(\mathrm{w})$, $\mathbb{E}\|\hat{\mathrm{g}}(\mathrm{w})\|^2 \leq G^2$, and ζ is an arbitrary perturbation that satisfies $\|\zeta\| \leq B$.

Given a step size $\eta > 0$ and an initial parameter $\mathrm{w}_0 \in \mathcal{W}$, SGD makes T queries to the PSFO, receives poisoned stochastic first-order information $\tilde{\mathrm{g}}_t := \tilde{\mathrm{g}}(\mathrm{w}_t) = \hat{\mathrm{g}}(\mathrm{w}_t) + \zeta_t$, and generates a sequence of parameters $\mathrm{w}_1, \ldots, \mathrm{w}_T$, where $\mathrm{w}_{t+1} = \Pi_{\mathcal{W}}(\mathrm{w}_t - \eta\tilde{\mathrm{g}}_t)$ for $t \in \{1, \ldots, T\}$, and $\Pi_{\mathcal{W}}$ projects onto the convex set $\mathcal{W}$. With this introduction, we prove the following robustness guarantee for SGD.

Theorem 2.2 (Robustness of SGD)**.** Let $F : \mathcal{W} \to \mathbb{R}$ be a convex function. Assume that all $\mathrm{w} \in \mathcal{W}$ satisfy $\|\mathrm{w}\| \leq D$. Let $\bar{\mathrm{w}} := \frac{1}{T}\sum_{t=1}^{T} \mathrm{w}_t$ be the average of the SGD iterates after T calls to a (G, B')-PSFO for F, with step sizes $\eta = \frac{D}{\sqrt{T}(G+B')}$, starting from arbitrary initialization $\mathrm{w}_0 \in \mathcal{W}$. Then it holds that

$$\mathbb{E}[F(\bar{\mathrm{w}})] - F(\mathrm{w}_*) \leq \frac{5D(G + B')}{2\sqrt{T}} + \frac{2D\sum_{t=1}^{T}\|\zeta_t\|}{T}.$$

The proof of Theorem 2.2 can be found in Appendix B.1. Theorem 2.2 implies that SGD can robustly learn convex learning problems as long as the cumulative perturbation norm due to the PSFO is sublinear in the number of oracle calls. In particular, when $\sum_{t=1}^{T}\|\zeta_t\| = \mathcal{O}(\sqrt{T})$, the poisoning attack cannot impose any significant statistical overhead on learning problem.

Furthermore, the upper bound presented in Theorem 2.2 is tight in an information-theoretic sense.

Theorem 2.3 (Optimality of SGD)**.** There exists a function $F : [-1, 1] \to \mathbb{R}$, and a $(1, 1)$-PSFO for F, such that any optimization algorithm making T calls to the oracle incurs an excess error of

$$\mathbb{E}[F(\bar{\mathrm{w}})] - F(\mathrm{w}_*) \geq \Omega\left(\frac{1}{\sqrt{T}} + \frac{\sum_{t=1}^{T}\|\zeta_t\|}{T}\right).$$

We note that *inexact* first-order oracles has been studied in several previous papers (Schmidt et al., 2011; Honorio, 2012; Devolder et al., 2014; Hu et al., 2016; Dvurechensky, 2017; Hu et al., 2020; Ajalloeian & Stich, 2020). Most of these works, however, make strong distributional assumptions on the perturbations, which are impractical in real adversarial settings. In a closely related line of work, (Hu et al., 2016; 2020; Amir et al., 2020; Ajalloeian & Stich, 2020) focus on biased SGD, and give convergence guarantees for several classes of important machine learning problems. However, we are not aware of any previous work studying robustness of SGD in neural networks, which is the subject of the next section.

3. Neural Networks

Next, we focus on two-layer neural networks with ReLU activation function and characterize sufficient conditions under which SGD can efficiently and robustly learn the network. A two-layer ReLU net, parameterized using a pair of weight matrices (a, W), computes the following function:

$$f(\mathrm{x}; \mathrm{a}, \mathrm{W}) := \frac{1}{\sqrt{m}}\sum_{s=1}^{m} a_s\sigma(\mathrm{w}_s^\top \mathrm{x}).$$

Here, m corresponds to the number of hidden nodes, i.e., the network width, $\mathrm{W} = [\mathrm{w}_1, \ldots, \mathrm{w}_m]$, $\mathrm{a} = [a_1, \ldots, a_m]$, and $\sigma(z) := \max\{0, z\}$ is the ReLU. We initialize the top layer weights, $a_s \sim \mathrm{unif}(\{-1, +1\})$, and keep them fixed through the training. The bottom layer weights are initialized as $\mathrm{w}_{s,0} \sim \mathcal{N}(0, \mathrm{I}_d)$ and are updated using SGD on the logistic loss $\ell(z) := \log(1 + e^{-z})$. We denote the weight matrix at the t^{th} iterate of SGD as W_t and the incoming weight vector into the s^{th} hidden node at iteration t as $\mathrm{w}_{s,t}$. Since a is fixed during the training, for the simplicity of presentation, we denote the network output on the i^{th} clean and perturbed sample, respectively, as $f_i(\mathrm{W}) := f(\mathrm{x}_i; \mathrm{a}, \mathrm{W})$ and $\tilde{f}_i(\mathrm{W}) := f(\tilde{\mathrm{x}}_i; \mathrm{a}, \mathrm{W})$. Therefore, at time t, the network weights are updated according to $\mathrm{W}_{t+1} = \mathrm{W}_t - \eta_t\nabla\ell(\tilde{y}_t\tilde{f}_t(\mathrm{W}_t))$.

In this section, we assume that the data is normalized so that $\|\mathrm{x}\| = 1$. This assumption is standard in the literature of over-parameterized neural networks (Du et al., 2019b; Allen-Zhu et al., 2018; Cao & Gu, 2019; Ji & Telgarsky, 2019); however, the results can be extended to the setting where the norm of the data is both upper- and lower-bounded by some constants. Moreover, following Ji & Telgarsky (2019), we assume that the distribution is separable by a positive margin in the reproducing kernel Hilbert space (RKHS) induced by the gradient of the infinite-width network at initialization.

Assumption 1 ((Ji & Telgarsky, 2019))**.** Let $\mathrm{z} \sim \mathcal{N}(0, \mathrm{I}_d)$ be a d-dimensional standard Gaussian random vector. There exists a *margin parameter* $\gamma > 0$, and a linear separator $\bar{v} : \mathbb{R}^d \to \mathbb{R}^d$ satisfying (A) $\mathbb{E}_{\mathrm{z}}[\|\bar{v}(\mathrm{z})\|^2] < \infty$; (B) $\|\bar{v}(\mathrm{z})\|_2 \leq 1$ for all $\mathrm{z} \in \mathbb{R}^d$; and (C) $y\mathbb{E}_{\mathrm{z}}[\langle\bar{v}(\mathrm{z}), \mathrm{x}\mathbb{1}[\mathrm{z}^\top\mathrm{x} \geq 0]\rangle] \geq \gamma$ for almost all $(\mathrm{x}, y) \sim \mathcal{D}$.

We note that the assumption above pertaining the linearly separability of data after mapping it into a high-dimensional non-linear feature space is mild and reasonable – this very

idea has been the cornerstone of kernel methods using the radial basis function (RBF) kernel, for example, and for learning with neural networks.

Next, we specify three data poisoning regimes under which SGD can efficiently and robustly learn two-layer ReLU networks under Assumption 1. Recall that the misclassification error due to $f(\cdot; \mathrm{a}, \mathrm{W})$ is denoted by $L(\mathrm{W}) := \mathbb{P}_{\mathcal{D}}(yf(\mathrm{x}; \mathrm{a}, \mathrm{W}) \leq 0)$ – note that a is fixed after the initialization and hence is dropped from the arguments of L.

3.1. Regime A (clean label attacks): large per-sample perturbation, small overall perturbation

Our first result concerns the setting where each individual sample can be arbitrarily poisoned as long as the overall perturbation budget is small compared to the sample size.

Theorem 3.1 (Regime A)**.** Under Assumption 1, for any $\delta \in (0,1)$, with probability at least $1-\delta$ over random initialization and the training samples, the iterates of SGD with constant step size $\eta = \frac{1}{(1+B)^2\sqrt{n}}$ satisfy

$$\frac{1}{n}\sum_{i<n} L(\mathrm{W}_i) \lesssim \frac{\ln^2(\sqrt{n}/4) + \ln(24n/\delta)}{\sqrt{n}\gamma^2},$$

provided that $B \leq \tilde{\mathcal{O}}(\gamma/\sqrt{d})$ and

$$\frac{\ln(\frac{n}{\delta}) + \ln^2(n)}{\gamma^8} \lesssim m \lesssim \frac{n\ln^4(n) + n\ln^2(\frac{n}{\delta})}{\gamma^4 S^2}.$$

We note that both the generalization error rate as well as the lower- and upper-bounds on the width depend on B, the per-sample perturbation budget; we refer the reader to the detailed expressions in Theorem B.8 in the appendix. For the width lower- and upper-bounds in Theorem 3.1 to be consistent, i.e. allowing a non-empty range for the width, the overall perturbation budget S needs to be $\lesssim \gamma^2\sqrt{n}$ (thus, *small cumulative perturbation*). This requirement is indeed the same as what we observed in convex learning problems, i.e. $S = \mathcal{O}(\sqrt{n})$, given by Theorem 2.2 in Section 2. Notably, the per-sample perturbation budget can be large since it is independent of the width, and the sample size.

3.2. Regime B (clean label attacks): small per-sample perturbation, large overall perturbation

Our next result shows that SGD can still succeed even if the overall budget grows linearly with the sample size, provided that the per-sample perturbations are small.

Theorem 3.2 (Regime B)**.** Under Assumption 1, for any $\delta \in (0,1)$, with probability at least $1-\delta$ over random initialization and the training samples, the iterates of SGD with constant step size $\eta = (1+B)^{-2}$ satisfy

$$\frac{1}{n}\sum_{i<n} L(\mathrm{W}_i) \leq \frac{\ln^2(\sqrt{n}/4) + \ln(24n/\delta)}{n\gamma^2}$$

for $m = \Omega\left(\frac{1}{\gamma^8}\left(\ln(n/\delta) + \ln^2(n)\right)\right)$, provided

$$B \lesssim \min\{\frac{1}{\sqrt{md} + \sqrt{m\ln(\frac{m}{\delta})}}, \frac{\gamma}{\gamma + \sqrt{d} + \sqrt{\ln(\frac{mn}{\delta})}}\}.$$

In this regime, we only allow a small per-sample perturbation $\lesssim 1/\sqrt{md}$; however, the cumulative perturbation can grow linearly with the sample size, i.e. $S = \Theta(n)$.

3.3. Regime C (label flip attacks)

Next, we show that SGD can withstand label flip attacks in small amounts.

Theorem 3.3 (Regime C)**.** Under Assumption 1, for any $\delta \in (0,1)$, with probability at least $1-\delta$ over random initialization and the training samples, the iterates of SGD with constant step size $\eta = 1/\sqrt{n}$ satisfy

$$\frac{1}{n}\sum_{i<n} L(\mathrm{W}_i) \lesssim \frac{\ln^2(\sqrt{n}/4) + \ln(16n/\delta)}{\sqrt{n}\gamma^2},$$

provided that $\beta \lesssim \frac{\ln(n/\delta)+\ln^2(n)}{(\sqrt{\ln(n/\delta)}+\ln(\frac{\gamma^2\sqrt{n}}{\ln(n/\delta)+\ln^2(n)}))\gamma\sqrt{n}}$, and $m = \Omega\left(\frac{1}{\gamma^8}\left(\ln(n/\delta) + \ln^2(n)\right)\right)$.

We conclude this section with a couple of remarks.

First, note that the generalization bounds obtained in Regimes A and C, given in Theorems 3.1 and 3.3, are essentially of the same rate of $\mathcal{O}(1/\sqrt{n})$. While the nature of the clean label attacks and label flip attacks corresponding to Regimes A and C are very different, the effective overall perturbation budget in both regimes are almost of the same order of $\tilde{\mathcal{O}}(\sqrt{n})$. We emphasize that there is a tension between the generalization error rate and the perturbation budget, and that different trade-offs can be obtained where faster or slower error rates correspond to smaller or larger perturbation budgets, respectively. On the contrary, Theorem 3.2 in regime B allows a larger overall perturbation budget of order $\mathcal{O}(n)$, and offers faster generalization error rate of $\tilde{\mathcal{O}}(1/n)$. We note, however, that the per-sample perturbation budget in this regime is significantly smaller than regimes A, especially for high-dimensional inputs. Therefore, the results above cover substantially different practical settings and are not directly comparable.

Second, note that in Theorem 3.3, we require $\beta \leq O(1/m)$ (ignoring other terms) which bounds m from above in terms of other parameters. Similarly, there is an implicit upper bound on m in terms of B in Theorem 3.2. In other words, in all three regimes that we consider, the generalization bounds hold if the width is bounded from both above and below.

4. Proof sketch

Our analysis is motivated by recent advances in the literature of over-parameterized neural networks. In particular, a nascent view of the modern over-parameterized models suggests that infinitely wide neural networks behave like linear functions in the reproducing kernel Hilbert space induced by the gradient of the network at the initialization, i.e. the *feature map* $\phi : \mathrm{x} \mapsto \nabla f(\mathrm{x}; \mathrm{w}_0)$ (Jacot et al., 2018; Lee et al., 2019; Du et al., 2019a). Therefore, the dynamics of SGD are approximately linear and are governed by the *neural tangent kernel* (NTK): $k(\mathrm{x}, \mathrm{x}') := \langle \nabla f(\mathrm{x}; \mathrm{w}_0), \nabla f(\mathrm{x}'; \mathrm{w}_0) \rangle$.

It is easy to see that the feature map $\phi_{\mathrm{x}} : \mathrm{z} \mapsto \mathrm{x}\mathbb{1}[\mathrm{z}^\top \mathrm{x} \geq 0]$ is closely related to the gradient of network at initialization through $\frac{\partial f(\mathrm{x}; \mathrm{W}_0, \mathrm{a})}{\partial \mathrm{w}_{s,0}} := \frac{1}{\sqrt{m}} a_s \phi_{\mathrm{x}}(\mathrm{w}_{s,0})$. Define $\bar{\mathrm{U}} = [\bar{\mathrm{u}}_1, \cdots, \bar{\mathrm{u}}_m]$ where $\bar{\mathrm{u}}_s := \frac{1}{\sqrt{m}} a_s \bar{\mathrm{v}}(\mathrm{w}_{s,0})$, and observe that:

$$y \langle \bar{\mathrm{U}}, \nabla f(\mathrm{x}; \mathrm{W}, \mathrm{a}) \rangle = y \cdot \frac{1}{m} \sum_{s=1}^{m} \langle \bar{\mathrm{v}}(\mathrm{w}_{s,0}), \mathrm{x}\mathbb{1}[\mathrm{x}^\top \mathrm{w}_{s,0} \geq 0] \rangle$$

which is a finite-width estimation of the *margin* quantity in part (C) of Assumption 1.

We denote the *instantaneous loss* on the clean sample and the poisoned sample as $R_i(\mathrm{W}) := \ell(y_i \langle \nabla f_i(\mathrm{W}_i), \mathrm{W} \rangle)$ and $\tilde{R}_i(\mathrm{W}) := \ell(\tilde{y}_i \langle \nabla \tilde{f}_i(\mathrm{W}_i), \mathrm{W} \rangle)$, respectively. Therefore, in the t^{th} iterate of SGD, the network weights are updated according to $\mathrm{W}_{t+1} = \mathrm{W}_t - \eta_t \nabla \tilde{R}_t(\mathrm{W}_t)$.

4.1. Proof sketch of Theorem 3.1 and Theorem 3.2

1. Let $Q_i(\mathrm{W}) := -\ell'(y_i \langle \nabla f_i(\mathrm{W}_i), \mathrm{W} \rangle)$ be the derivative of the instantaneous loss $R_i(\mathrm{W})$. An interesting property of $Q_i(\mathrm{W})$ is that it upperbounds the zero-one loss, and is upperbounded by $R_i(\mathrm{W})$. This property has been used in several previous works (Cao & Gu, 2020; Ji & Telgarsky, 2019) to upperbound the average misclassification error as $\frac{1}{n} \sum_{i<n} L(\mathrm{W}_i) < \frac{1}{n} \sum_{i<n} Q(\mathrm{W}_i)$. Using a martingale concentration argument we then show that $\frac{1}{n} \sum_{i<n} Q_i(\mathrm{W}_i)$ is close to $\frac{1}{n} \sum_{i<n} Q(\mathrm{W}_i)$, where $Q(\mathrm{W}_i)$ is the expectation of $Q_i(\mathrm{W}_i)$ with respect to data distribution. Finally, since the instantaneous loss upperbounds its derivative, we arrive at $\frac{1}{n} \sum_{i<n} L(\mathrm{W}_i) < \frac{8}{n} \sum_{i<n} R_i(\mathrm{W}_i) + \epsilon$.

2. To bound $\frac{1}{n} \sum_{i<n} R_i(\mathrm{W}_i)$, we argue that under the perturbation budgets considered in our theorems, $R_i(\mathrm{W}_i)$ is close to $\tilde{R}_i(\mathrm{W}_i)$. In regime A, we appeal to convexity of the loss function and Lipschitzness of the network to bound the difference $R_i(\mathrm{W}_i) - \tilde{R}_i(\mathrm{W}_i)$ as $\mathcal{O}(\sqrt{md}\|\delta_i\|)$, which gives sufficient conditions on the perturbation budget in Regime A. For regime B, we use the convexity of the loss and the fact that $Q_i(\mathrm{W}) \leq R_i(\mathrm{W})$ to show that $(1 - \mathcal{O}(\sqrt{mdB}))R_i(\mathrm{W}_i) \leq \tilde{R}_i(\mathrm{W}_i)$. Therefore, as long as $\mathcal{O}(\sqrt{mdB})$ is not small, we can bound $\frac{1}{n} \sum_{i<n} R_i(\mathrm{W}_i)$ in terms of $\frac{1}{n} \sum_{i<n} \tilde{R}_i(\mathrm{W}_i)$.

3. We then follow (Ji & Telgarsky, 2019) to bound $\frac{1}{n} \sum_{i<n} \tilde{R}_i(\mathrm{W}_i)$. The separability assumption 1 is crucial for this step.

4.2. Proof sketch of Theorem 3.3

1. We first observe that the zero-one loss of (x, y) is the same as the zero-one loss of the expectation of $(\mathrm{x}, \tilde{y})$ with respect to the randomness of label flips, i.e. $\frac{1}{n} \sum_{i<n} L(\mathrm{W}_i) = \mathbb{P}(\mathbb{E}\tilde{y} f(\mathrm{x}; \mathrm{W}_i) \leq 0)$, and is upperbounded by $-2\mathbb{E}_{(\mathrm{x},y)\in\mathcal{D}} \ell'(\mathbb{E}\tilde{y} f(\mathrm{x}; \mathrm{W}))$. Using a martingale concentration argument, we arrive at $\frac{1}{n} \sum_{i<n} L(\mathrm{W}_i) \leq -\frac{8}{n} \sum_{i<n} \ell'(\mathbb{E}\tilde{y}_i f_i(\mathrm{W}_i)) + \epsilon$, which can be further bounded by $\frac{8}{n} \sum_{i<n} \ell(\mathbb{E}\tilde{y}_i f_i(\mathrm{W}_i)) + \epsilon$ because $-\ell'(\cdot) \leq \ell(\cdot)$. Since ℓ is convex, using Jensen's inequality, we further bound the generalization error as $\frac{1}{n} \sum_{i<n} L(\mathrm{W}_i) < \frac{8}{n} \sum_{i<n} \mathbb{E}\ell(\tilde{y}_i f_i(\mathrm{W}_i)) + \epsilon$.

2. We leverage an interesting property of the logistic loss, the fact that $\ell(-z) - \ell(z) = z$, to reduce the expected instantaneous loss above to $\mathbb{E}\ell(\tilde{y}_i f_i(\mathrm{W}_i)) = \ell(y_i \langle \nabla \tilde{f}_i(\mathrm{W}_i), \overline{\mathrm{W}} \rangle) + \beta y_i \langle \nabla \tilde{f}_i(\mathrm{W}_i), \overline{\mathrm{W}} \rangle$. While the first term can be bounded using the proof techniques in (Ji & Telgarsky, 2019), the second term requires β to be *sufficiently small*, which gives the required upperbound on the probability of label flips in the statement of the theorem.

5. Experimental Results

The goal of this section is to provide experimental support for our theoretical findings in Section 2 and Section 3. Code is available on Github [1]. First, we describe the experimental setup.

Datasets. We utilize the MNIST and the CIFAR10 datasets for the empirical evaluation. MNIST is a dataset of 28×28 greyscale handwritten digits, containing $70K$ samples in 10 classes, with $60K$ training images and $10K$ test images. CIFAR10 is a dataset of 32×32 color images, containing $60K$ samples in 10 classes, with $50K$ training images and $10K$ test images.

Model specification. We utilize four different models: a linear model trained on MNIST, an AlexNet model trained on CIFAR10, and two convolutional neural networks, with width ranging from 10 to 100, 000, trained on MNIST and CIFAR. For the MNIST dataset, we use a model with two

[1] https://github.com/bettyttytty/robust_learning_for_data_poisoning_attack

convolutional layers followed by max-pooling layers, as well as two fully connected layers, with ReLU activation. The first and the second convolutional layers have [input channel, output channel, kernel size] equal to $[1, 10, 5]$ and $[10, 20, 5]$, respectively. The first and the second fully connected layers have [input, output] dimensions equal to $[320, width]$ and $[width, 10]$, respectively. For CIFAR10 dataset, we use a network with two convolutional layers with [input channel, output channel, kernel size] equal to $[3, 6, 5]$ and $[6, 16, 5]$, and three fully connected layers with [input, output] dimensions equal to $[400, 120]$, $[120, width]$, $[width, 10]$. The architecture of AlexNet is the same as (Luo, 2018). We initialize the networks using Pytorch initialization and train them using cross-entropy loss. We track the test accuracy of the networks as a function of the width to verify our theorems.

Attack strategy. When generating poisoning attacks using projected gradient ascent, we discovered that a simple modification of the cross-entropy loss generates stronger poisoning attacks, both qualitatively and quantitatively. This new loss function, which we call *negated loss*, is obtained by flipping the sign on both the model prediction and the cross-entropy loss. For example, in the binary classification case, the negated loss is given by $\ell_{--}(z) := -\ell(-z) = -\log(1+\exp(z))$, where z is the model prediction. Among other properties, this loss is concave in z, and lower bounds the zero-one loss, which makes it a useful "surrogate" loss for finding adversarial perturbations.

To generate the poisoned data in regimes A and B (i.e., clean label attacks), we first use mini-batch SGD with batch size 128 to learn the model parameters (w^*) on the clean data. The poisoned data is then generated by taking a stochastic gradient ascent step on the negated loss to maximize the negated loss function $\ell(\cdot; \mathrm{w}^*)$, followed by a projection onto the constraints – the $\ell_{2,1}$ for regime A- or the $\ell_{2,\infty}$-norm ball for regime B. In particular, for regime A, we specify the overall noise budget $S = C\sqrt{n}$, and project the overall perturbation iteratively onto the $\ell_{2,1}$-norm ball. Here n is the number of training samples and C is the corruption rate.

We generalize the label flip attack in regime C to the multiclass classification setting by switching the label, with probability β, of any given training data point from the true class i to $(i+1) \bmod 10$.

We now present our main empirical findings.

Negated loss vs. the original loss. Figure 1 compares the data poisoning attacks generated by PGA on the proposed negated loss against the original cross entropy loss under regime A. The top row compares the (clean) test accuracy under two loss functions as a function of corruption rate (C). The left panel corresponds to a linear model trained on the MNIST dataset and the right panel corresponds to AlexNet trained on CIFAR10. We observe similar plots for ResNet18 – please refer to Appendix A for more details.

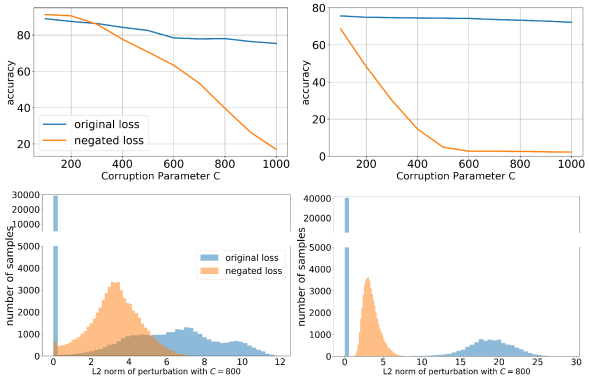

Figure 1. Data poisoning attacks generated by PGA on the proposed negated loss (orange) against the original cross entropy loss (blue). **Left**: a linear model trained on MNIST, **right**: AlexNet trained on CIFAR10. The top row shows the (clean) test accuracy of the model trained on poisoned data. The bottom row shows the histogram of L_2 norm of perturbation vectors generated using the two loss functions.

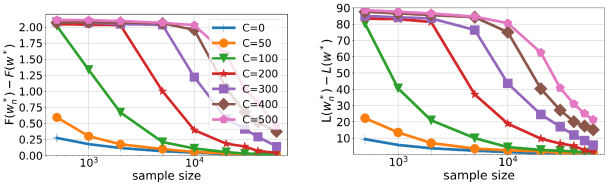

Figure 2. The excess loss $F(\mathrm{w}_n^*) - F(\mathrm{w}^*)$ **(left)**; and the excess error $L(\mathrm{w}_n^*) - L(\mathrm{w}^*)$ **(right)**, as a function of sample size n with different corruption parameter $C \in \{50, 100, 200, 300, 400, 500\}$ under regime A. Here, w_n^* denotes the optimal parameters on the given sample of size n.

We note that the cross entropy is only an upper-bound on the zero-one loss, and hence, a proper surrogate for minimizing the classification error. However, we seek to maximize the zero-one loss when generating the poisoning attacks. Therefore, given a fixed perturbation budget, there is no advantage in perturbing a sample beyond the point that it is misclassified, which can very much happen when maximizing the original cross entropy loss. On the contrary, the negated loss strictly lower-bounds the zero-one loss.

The bottom row of Figure 1 compares the distribution of the L_2 norm of per-sample perturbation generated by the two loss functions; the left and right panels correspond to a linear model and AlexNet, respectively. We see that PGA on the original loss tends to allow an excessive amount of per-sample perturbations at the cost of leaving a large portion of the samples virtually untouched. For example, for the MNIST dataset, a perturbation of size 5 is more than enough to make label prediction hard, e.g., by planting a solid "1" in any image, whereas the poisoning attack using the original

loss "wastes" a lot of the budget on making a few samples "more wrong".

Convex learning problems. We first train a linear model on the clean MNIST dataset and denote it with w^*. We then train several linear models under poisoning attacks for various sample sizes in range $n \in [500, 60000]$, which we denote by w_n^*. Figure 2 shows the excess loss $F(\mathrm{w}_n^*) - F(\mathrm{w}^*)$ as well as the excess error $L(\mathrm{w}_n^*) - L(\mathrm{w}^*)$ as a function of sample size n, for different corruption rates $C \in \{50, 100, 200, 300, 400, 500\}$.

It is not surprising that both the excess loss as well as the excess error are smaller for larger sample sizes or smaller corruption rates, as predicted by Theorem 2.2. More interestingly, the plots suggest a phase transition between the convergence behavior of the curves at $C \approx 250 \approx \sqrt{n}$, which corresponds to the maximum corruption rate under which Theorem 2.2 still yields a non-trivial (decaying with sample size) generalization error bound.

Wide neural networks. Recall that our theoretical results in Section 3 guarantee a small generalization error for networks trained with poisoned data only when the network width falls in a certain range specified in the theorems. While it is not clear whether these bounds are necessary, we observe that the clean test accuracy of models trained on poisoned data exhibits an inverted U curve. In other words, the generalization accuracy decreases if the models are not wide enough or if they are too wide. In Figure 3, we see that for clean data training corresponding to the green curves, the accuracy improves monotonically with the network width. However, in presence of data poisoning attacks, in both left (MNIST) and right (CIFAR10) panels, we observe that the test accuracy is non-monotonic in terms of the network width. In each of the regimes A, B, and C, we see that the accuracy improves as we initially increase the network width. It then hits a plateau and eventually starts to fall as we further increase the width. This observation challenges the nascent view in the deep learning literature that larger models generalize better (Neyshabur et al., 2014; Zhang et al., 2016), at least under adversarial perturbations.

6. Discussion and Future Work

In this paper we study the robustness of SGD to data poisoning attacks in two-layer neural networks. In particular, under a separability assumption in the feature space induced by the gradient of the infinite-width network at initialization, we characterize several practical data poisoning scenarios where SGD efficiently learns the network, provided that the *network width is sufficiently but not excessively large*. In sharp contrast with clean-data training where the generalization error decreases as the width of the network increases (Zhang et al., 2016; Neyshabur et al., 2014), curiously, our empirical findings indicate that robust generalization error exhibits a U-curve as a function of the network width.

There are several natural directions for future work. First, although we observe in practice that ultra-wide neural networks are more vulnerable to data poisoning attacks, our theoretical results do not directly imply that too large of a network width can actually hurt the generalization performance under data poisoning attacks. Therefore, a natural question that remains open is to prove that SGD fails at robustly learning ultra-wide neural networks in presence of adversarial perturbations such as those considered in this work. We would like to highlight that in a very recent work, Bubeck et al. (2020) conjecture that over-parameterization may be necessary for robustness; while our results do not contradict theirs, it certainly calls for further investigation into the role of over-parameterization in imparting or degrading robustness.

Second, our theory heavily depends on the separability assumption and cannot be trivially extended to deeper architectures; yet, our empirical findings go beyond two-layer networks, and hold for natural datasets where the separability assumption is no longer true. It remains to be seen if we can relax the margin assumption and generalize our results to richer network architectures.

Third, our paper focuses on the role of the width; however, it is not immediately clear from our results if the U-curve phenomena is specific to the network width, or if it can more broadly happen for ultra-large networks. It would be interesting to explore the role of other architectural parameters, such as the network depth, in robust learning.

Acknowledgements

This research was supported, in part, by DARPA GARD award HR00112020004, NSF BIGDATA award IIS-1546482 and NSF CAREER award IIS-1943251.

References

Afouras, T., Chung, J. S., Senior, A., Vinyals, O., and Zisserman, A. Deep audio-visual speech recognition. *IEEE transactions on pattern analysis and machine intelligence*, 2018.

Ajalloeian, A. and Stich, S. U. Analysis of sgd with biased gradient estimators. *arXiv preprint arXiv:2008.00051*, 2020.

Allen-Zhu, Z., Li, Y., and Song, Z. A convergence theory for deep learning via over-parameterization. *arXiv preprint arXiv:1811.03962*, 2018.

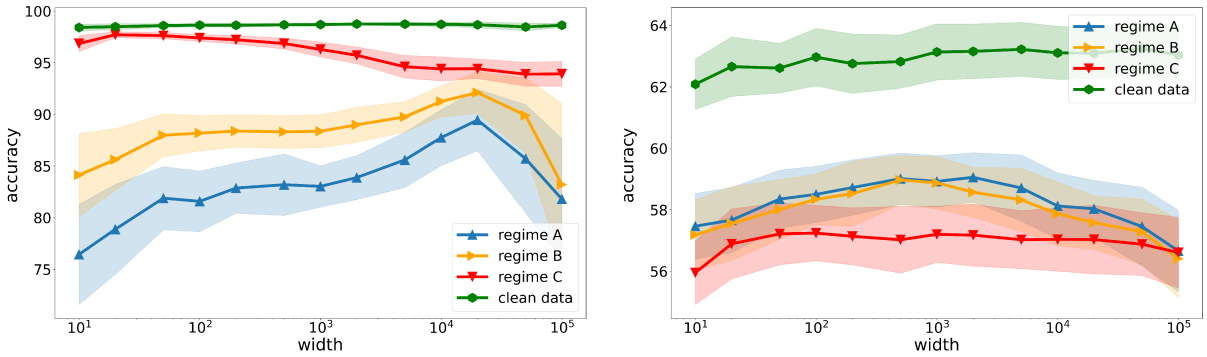

Figure 3. Clean test accuracy as a function of the network width for convolutional neural networks (see model specification) trained under data poisoning attacks on MNIST **(left)** and CIFAR10 **(right)** for regime A ($C = 700$ for MNIST, $C = 200$ for CIFAR10), regime B ($B = 2.5$ for MNIST and $B = 1$ for CIFAR10) and regime C ($\beta = 0.3$ for MNIST and $\beta = 0.2$ for CIFAR10. For regime A and B, the poisoned MNIST data is generated using convolutional neural network with width=100 (see model specification), the poisoned CIFAR10 data is generated using AlexNet. We use vanilla SGD with batch size 128 (no momentum, no weight decay, no data augmentation). Each curve is averaged over 50 runs and shaded regions show standard deviation. We perform 5-fold cross validation to pick model parameters including learning rate.

Amir, I., Attias, I., Koren, T., Livni, R., and Mansour, Y. Prediction with corrupted expert advice. *arXiv preprint arXiv:2002.10286*, 2020.

Awasthi, P., Balcan, M. F., and Long, P. M. The power of localization for efficiently learning linear separators with noise. In *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*, pp. 449–458, 2014.

Barreno, M., Nelson, B., Joseph, A. D., and Tygar, J. D. The security of machine learning. *Machine Learning*, 81 (2):121–148, 2010.

Biggio, B., Nelson, B., and Laskov, P. Support vector machines under adversarial label noise. In *Asian conference on machine learning*, pp. 97–112, 2011.

Bubeck, S., Li, Y., and Nagaraj, D. A law of robustness for two-layers neural networks. *arXiv preprint arXiv:2009.14444*, 2020.

Cao, Y. and Gu, Q. Generalization bounds of stochastic gradient descent for wide and deep neural networks. In *Advances in Neural Information Processing Systems*, pp. 10835–10845, 2019.

Cao, Y. and Gu, Q. Generalization error bounds of gradient descent for learning over-parameterized deep relu networks. In *AAAI*, pp. 3349–3356, 2020.

Cesa-Bianchi, N., Shalev-Shwartz, S., and Shamir, O. Online learning of noisy data. *IEEE Transactions on Information Theory*, 57(12):7907–7931, 2011.

Chen, X., Liu, C., Li, B., Lu, K., and Song, D. Targeted backdoor attacks on deep learning systems using data poisoning. *arXiv preprint arXiv:1712.05526*, 2017.

Chizat, L., Oyallon, E., and Bach, F. On lazy training in differentiable programming. arxiv e-prints, page. *arXiv preprint arXiv:1812.07956*, 2018.

Devolder, O., Glineur, F., and Nesterov, Y. First-order methods of smooth convex optimization with inexact oracle. *Mathematical Programming*, 146(1-2):37–75, 2014.

Diakonikolas, I., Gouleakis, T., and Tzamos, C. Distribution-independent pac learning of halfspaces with massart noise. *arXiv preprint arXiv:1906.10075*, 2019a.

Diakonikolas, I., Kamath, G., Kane, D., Li, J., Steinhardt, J., and Stewart, A. Sever: A robust meta-algorithm for stochastic optimization. In *International Conference on Machine Learning*, pp. 1596–1606, 2019b.

Du, S., Lee, J., Li, H., Wang, L., and Zhai, X. Gradient descent finds global minima of deep neural networks. In *International Conference on Machine Learning*, pp. 1675–1685, 2019a.

Du, S. S., Zhai, X., Poczos, B., and Singh, A. Gradient descent provably optimizes over-parameterized neural networks. In *International Conference on Learning Representations*, 2019b. URL https://openreview.net/forum?id=S1eK3i09YQ.

Dvurechensky, P. Gradient method with inexact oracle for composite non-convex optimization. *arXiv preprint arXiv:1703.09180*, 2017.

Grigorescu, S., Trasnea, B., Cocias, T., and Macesanu, G. A survey of deep learning techniques for autonomous driving. *Journal of Field Robotics*, 37(3):362–386, 2020.

Gu, T., Dolan-Gavitt, B., and Garg, S. Badnets: Identifying vulnerabilities in the machine learning model supply chain. *arXiv preprint arXiv:1708.06733*, 2017.

Guruswami, V. and Raghavendra, P. Hardness of learning halfspaces with noise. *SIAM Journal on Computing*, 39 (2):742–765, 2009.

Honorio, J. Convergence rates of biased stochastic optimization for learning sparse ising models. *arXiv preprint arXiv:1206.4627*, 2012.

Hu, B., Seiler, P., and Lessard, L. Analysis of biased stochastic gradient descent using sequential semidefinite programs. *Mathematical Programming*, pp. 1–26, 2020.

Hu, X., LA, P., György, A., and Szepesvári, C. convex optimization with biased noisy gradient oracles. arxiv. 2016.

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.

Jagielski, M., Oprea, A., Biggio, B., Liu, C., Nita-Rotaru, C., and Li, B. Manipulating machine learning: Poisoning attacks and countermeasures for regression learning. In *2018 IEEE Symposium on Security and Privacy (SP)*, pp. 19–35. IEEE, 2018.

Ji, Z. and Telgarsky, M. Polylogarithmic width suffices for gradient descent to achieve arbitrarily small test error with shallow relu networks. *arXiv preprint arXiv:1909.12292*, 2019.

Kearns, M. and Li, M. Learning in the presence of malicious errors. *SIAM Journal on Computing*, 22(4):807–837, 1993.

Klivans, A. R., Long, P. M., and Servedio, R. A. Learning halfspaces with malicious noise. *Journal of Machine Learning Research*, 10(12), 2009.

Koh, P. W. and Liang, P. Understanding black-box predictions via influence functions. *arXiv preprint arXiv:1703.04730*, 2017.

Lee, J., Xiao, L., Schoenholz, S., Bahri, Y., Novak, R., Sohl-Dickstein, J., and Pennington, J. Wide neural networks of any depth evolve as linear models under gradient descent. In *Advances in neural information processing systems*, pp. 8570–8581, 2019.

Li, Y. Deep reinforcement learning: An overview. *arXiv preprint arXiv:1701.07274*, 2017.

Li, Y. and Liang, Y. Learning overparameterized neural networks via stochastic gradient descent on structured data. In *Advances in Neural Information Processing Systems*, pp. 8157–8166, 2018.

Liu, K., Dolan-Gavitt, B., and Garg, S. Fine-pruning: Defending against backdooring attacks on deep neural networks. In *International Symposium on Research in Attacks, Intrusions, and Defenses*, pp. 273–294. Springer, 2018.

Liu, Y., Ma, S., Aafer, Y., Lee, W.-C., Zhai, J., Wang, W., and Zhang, X. Trojaning attack on neural networks. 2017.

Luo, Z. pytorch-cifar10. https://github.com/icpm/pytorch-cifar10, 2018.

Neyshabur, B., Tomioka, R., and Srebro, N. In search of the real inductive bias: On the role of implicit regularization in deep learning. *arXiv preprint arXiv:1412.6614*, 2014.

Nitanda, A. and Suzuki, T. Refined generalization analysis of gradient descent for over-parameterized two-layer neural networks with smooth activations on classification problems. *arXiv preprint arXiv:1905.09870*, 2019.

Rosenfeld, E., Winston, E., Ravikumar, P., and Kolter, J. Z. Certified robustness to label-flipping attacks via randomized smoothing. *arXiv preprint arXiv:2002.03018*, 2020.

Schmidt, M., Roux, N. L., and Bach, F. R. Convergence rates of inexact proximal-gradient methods for convex optimization. In *Advances in neural information processing systems*, pp. 1458–1466, 2011.

Shafahi, A., Huang, W. R., Najibi, M., Suciu, O., Studer, C., Dumitras, T., and Goldstein, T. Poison frogs! targeted clean-label poisoning attacks on neural networks. In *Advances in Neural Information Processing Systems*, pp. 6103–6113, 2018.

Steinhardt, J., Koh, P. W. W., and Liang, P. S. Certified defenses for data poisoning attacks. In *Advances in neural information processing systems*, pp. 3517–3529, 2017.

Suciu, O., Marginean, R., Kaya, Y., Daume III, H., and Dumitras, T. When does machine learning {FAIL}? generalized transferability for evasion and poisoning attacks. In *27th {USENIX} Security Symposium ({USENIX} Security 18)*, pp. 1299–1316, 2018.

Tran, B., Li, J., and Madry, A. Spectral signatures in backdoor attacks. In *Advances in Neural Information Processing Systems*, pp. 8000–8010, 2018.

Valiant, L. G. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.

Valiant, L. G. Learning disjunction of conjunctions. In *IJCAI*, pp. 560–566. Citeseer, 1985.

Wang, Y., Jha, S., and Chaudhuri, K. An investigation of data poisoning defenses for online learning. *arXiv preprint arXiv:1905.12121*, 2019.

Xiao, H., Xiao, H., and Eckert, C. Adversarial label flips attack on support vector machines. In *ECAI*, pp. 870–875, 2012.

Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530*, 2016.

Zhao, M., An, B., Gao, W., and Zhang, T. Efficient label contamination attacks against black-box learning models. In *IJCAI*, pp. 3945–3951, 2017.

Zhu, C., Huang, W. R., Shafahi, A., Li, H., Taylor, G., Studer, C., and Goldstein, T. Transferable clean-label poisoning attacks on deep neural nets. *arXiv preprint arXiv:1905.05897*, 2019.

Zou, D., Cao, Y., Zhou, D., and Gu, Q. Stochastic gradient descent optimizes over-parameterized deep relu networks. *arXiv preprint arXiv:1811.08888*, 2018.