One-pass stochastic gradient descent in overparametrized two-layer neural networks

Jiaming Xu **Hanjing Zhu**
The Fuqua School of Business, Duke University

Abstract

There has been a recent surge of interest in understanding the convergence of gradient descent (GD) and stochastic gradient descent (SGD) in overparameterized neural networks. Most previous work assumes that the training data is provided a priori in a batch, while less attention has been paid to the important setting where the training data arrives in a stream. In this paper, we study the streaming data setup and show that with overparamterization and random initialization, the prediction error of two-layer neural networks under one-pass SGD converges in expectation. The convergence rate depends on the eigen-decomposition of the integral operator associated with the so-called neural tangent kernel (NTK). A key step of our analysis is to show a random kernel function converges to the NTK with high probability using the VC dimension and McDiarmid inequality.

1 Introduction

Deep Learning is proven to be successful in many real-life applications, while the underpinning of its success remains elusive. Recently, researchers are interested in understanding the success of neural networks from the optimization perspective. A neural network with the ReLU activation leads to a non-convex and non-smooth objective function, which is usually hard to optimize by gradient descent methods. However, surprisingly, in many cases, gradient descent (GD) or stochastic gradient descent (SGD) on neural networks with the ReLU activation is observed to perform well not only in training but also in generalization [Krizhevsky et al., 2012]. To demystify this surprising phenomenon, an extensive amount of research has been done recently. For instance, the mean-field theory is used in [Chen et al., 2020,Mei et al., 2018,Mei et al., 2019] to analyze the SGD of infinite-width two-layer neural networks. Optimal transport theory is employed in [Chizat and Bach, 2018] to study the gradient flow of neural networks and show that the training error converges to the global optimum under some mild conditions. In addition, [Hu et al., 2019] connects the SGD of neural networks in training to the diffusion process.

A different line of work focuses on understanding the gradient descent of neural networks through kernels, in particular the neural tangent kernel (NTK). It is first introduced by [Jacot et al., 2018], which shows that gradient descent on infinite width neural networks can be viewed as learning through the NTK. Subsequent work [Allen-Zhu et al., 2019a,Du et al., 2019b,Su and Yang, 2019, Arora et al., 2019, Du et al., 2019a, Zou et al., 2020] connects GD and SGD with the NTK, and show that with overparameterization and random initialization, the training error converges to 0. Similar convergence results are also established in other types of neural networks beyond the feed-forward neural networks [Allen-Zhu and Li, 2020, Allen-Zhu and Li, 2019b, Allen-Zhu and Li, 2019a, Allen-Zhu et al., 2019b, Du et al., 2018, Li et al., 2019], such as convolutional neural network (CNN) and residual neural network (ResNet).

Despite the remarkable progress, most previous work focuses on the batch setting where the training data is provided a priori in a batch. Less attention has been paid to the important streaming setting, where the data arrives continuously in a stream. The streaming data arises in a variety of fields such as finance, news organization, and information technology [O'callaghan et al., 2002, Allen-Zhu and Li, 2019b, Ikonomovska et al., 2007]. Such streaming data is usually inspected once and archived afterwards immediately without being examined again. Apart from vast sources of naturally generated streaming data, there are ubiquitous situations where the streaming data is preferred even though batches of samples can be obtained. For in-

Proceedings of the 24th International Conference on Artificial Intelligence and Statistics (AISTATS) 2021, San Diego, California, USA. PMLR: Volume 130. Copyright 2021 by the author(s).

stance, [O'callaghan et al., 2002] points out that in medical or marketing data mining, the volume of data is so large that only one pass over data is allowed due to computation constraints. Moreover, [Feigenbaum et al., 2001, Muthukrishnan, 2005] argues that the streaming data is useful in privacy-preserving data mining, where the data is kept confidentially by users and analyzed via a single pass.

In this paper, we study the streaming data setup where *i.i.d.* data points (X_t, y_t) ($X_t \in \mathbb{R}^d$ is feature, and $y_t \in \mathbb{R}$ is the corresponding label) arrive in a stream. We consider the two-layer neural network with ReLU activation and run the stochastic gradient descent on the streaming data in a single pass to train the neural network under the quadratic loss. Our goal is to study the convergence of the average prediction error. We do not consider the use of sliding window [Tashman, 2000] which views a trunk of consecutive data points as a single input to the neural network [1]. The contributions of this paper are summarized as follows:

- We show that with random initialization and an appropriate step size $\eta_t \leq \frac{\theta}{t+1}$ for $\theta < \frac{1}{4}$, if the number of neurons $m \geq \text{poly}(T, d, 1/\delta)$, then with probability at least $1 - \delta - 2\exp(-2m^{1/3})$, the average prediction error at iteration T is upper bounded by $\prod_{t=1}^{T}(1 - \eta_t\lambda_\ell) \left\|\Delta_0\right\|_2 + \mathcal{R}(\Delta_0, \ell) + O(\theta\sigma_0)$ for every $\ell \geq 1$, where λ_ℓ is the ℓ-th eigenvalue of the integral operator $\mathsf{\Phi}$ associated with the NTK Φ, Δ_0 is the prediction error at the initialization, $\mathcal{R}(\Delta_0, \ell)$ is the L_2 norm of the projection of Δ_0 onto the space spanned by the eigenfunctions corresponding to the eigenvalues $\{\lambda_i\}_{\ell+1}^{\infty}$, and σ_0^2 is the average squared prediction error at initialization. In particular, for an arbitrarily small but fixed constant $\epsilon > 0$, by choosing θ and δ to be sufficiently small, while T and m to be sufficiently large, the average prediction error is at most ϵ.
- On a technical front, our analysis departs significantly from the existing literature. Specifically, in the batch setting, the existing literature such as [Du et al., 2019b] and [Su and Yang, 2019] only need to deal with the kernel matrices and thus simple point-wise concentration plus union bound is enough to obtain the convergence of random kernel matrices with high probability. However, in the streaming data setup, such techniques are not directly applicable to prove the convergence of kernel functions. As such, we employ the VC dimension technique and McDiarmid's inequality to show that a random kernel function converges to the NTK with high probability.

Notation Let $(\mathcal{X}, \mu)$ denote a measurable space with measure μ. Let $L^2(\mathcal{X}, \mu)$ denote the space of functions $f : \mathcal{X} \to \mathbb{R}$ that are integrable, i.e., $\|f\|_2 \triangleq \sqrt{\int f^2(x) d\mu(x)} < \infty$. When $\mathcal{X}$ is the unit sphere $\mathbb{S}^{d-1}$ in $\mathbb{R}^d$, we abbreviate $L^2(\mathbb{S}^{d-1}, \mu)$ as $L^2(\mu)$ for simplicity. Define the L-infinite norm $\|f\|_\infty \triangleq \sup_{x \in \mathcal{X}} |f(x)|$. Given $f, g \in L^2(\mathcal{X}, \mu)$, define their inner products as $\langle f, g \rangle \triangleq \int f(x) g(x) d\mu(x)$ with $\langle f, f \rangle = \|f\|_2^2$. Given a kernel function $K \in L^2(\mathcal{X} \times \mathcal{X}, \mu \otimes \mu)$, define the associated integral operator $\mathsf{K} : L^2(\mathcal{X}, \mu) \to L^2(\mathcal{X}, \mu)$ as $\mathsf{K} f(x) = \int K(x, y) f(y) d\mu(y)$. The operator norm of K is defined as $\|\mathsf{K}\|_2 \triangleq \sup_{\|f\|_2 \leq 1} \|\mathsf{K} f\|_2$. Denote the composition of operators $\mathsf{K}_1, \mathsf{K}_2, \cdots, \mathsf{K}_m$ as $\prod_{i=1}^{m} \mathsf{K}_i$ with $\prod_{i=n+1}^{n} \mathsf{K}_i$ treated as the identity operator.

2 Related Work

To facilitate the discussion and better differentiate the algorithms, we use batch-SGD to denote the gradient descent algorithm where a sub-sample is drawn without replacement from the given batch to compute the gradient at each iteration, i.e., for the given batch $\{(x_i, y_i)\}_{i=1}^{n}$ and a loss function $l(\cdot, \cdot)$, $W(t+1) = W(t) - \frac{\eta_t}{|\mathcal{B}_t|} \sum_{i \in \mathcal{B}_t} \nabla_W l(f(x_i; W(t)), y_i)$ where $W(t)$ is the weight matrix at iteration t, $f(x; W(t))$ is the neural network with parameter $W(t)$ and $\mathcal{B}_t$ is a random subset of the batch $\{(x_i, y_i)\}_{i=1}^{n}$. The data in $\mathcal{B}_t$ may be reused in the later iterations. In the special case where the entire batch is used to compute the gradient at each iteration, i.e., $\mathcal{B}_t = \{(x_i, y_i)\}_{i=1}^{n}$ for any t, we refer the batch-SGD to GD. In contrast, our study focuses on the one-pass SGD, abbreviated as SGD, which draws a *single* fresh sample from the true data distribution to compute the gradient at each iteration. In particular, $W(t+1) = W(t) - \eta_t \nabla_W l(f(x_t; W(t)), y_t)$ where (x_t, y_t) is a freshly drawn sample at the t-th iteration from some unknown distribution μ. The drawn sample (x_t, y_t) is then archived and not used any more. Most existing literature focuses on the batch setting and uses GD/batch-SGD to train the neural networks.

Training error with batch learning. In [Du et al., 2019b], the training error of overparametrized neural networks is shown to converge at linear rate $\left[1 - \frac{\eta}{2}\lambda_{\min}(H)\right]^t$, where t is the number of iterations, η is the step size, and $H \in \mathbb{R}^{n \times n}$ is the gram matrix of the neural network with $H_{ij} = \Phi(x_i, x_j) = x_i^\top x_j \mathbb{E}_{w \sim N(0,I)} \left[\mathbf{1}_{\{\langle w, x_i \rangle \geq 0\}} \mathbf{1}_{\{\langle w, x_j \rangle \geq 0\}}\right]$. Furthermore, [Du et al., 2019a] extends the result to multi-layer neu-

[1] In the streaming data setting where the data is not allowed to be stored, the sliding window is not applicable. Even when it is allowed, due to the *i.i.d.* data assumption, it can be equivalently viewed as one-pass SGD with mini-batches to which our analysis still applies [Dehghani et al., 2019].

ral networks with analytic activation functions, by utilizing the gram matrix H of the last hidden layer. Despite these positive results, [Su and Yang, 2019] proves that as the sample size n grows, $\lambda_{\min}(H)$ decreases to 0 and hence the convergence rate can be very close to 0. Furthermore, [Su and Yang, 2019] proves that the training error is upper bounded by $\left[1-\frac{3\eta}{4}\lambda_r\right]^t + 2\sqrt{2}\mathcal{R}(f^*,r)+\Theta(\frac{1}{\sqrt{n}})$, where λ_r is the r-th largest eigenvalue of the integral operator Φ associated with the NTK Φ, $\mathcal{R}(f^*,r)$ is the L_2 norm of the projection of f^* onto the eigenspaces of kernel Φ associated with $\{\lambda_i\}_{i=r+1}^{\infty}$. Despite that the result in [Su and Yang, 2019] and our result share some similarity in terms of the eigen-decomposition of the NTK, our study differs from [Su and Yang, 2019] in two important aspects. First, the algorithm used in [Su and Yang, 2019] is GD while ours is one-pass SGD. A significant challenge for us is to control the accumulation of the noise due to the stochasticity of the gradients. Moreover, the focus of [Su and Yang, 2019] is on training error, while we focus on the average prediction error and has to deal with the convergence of random kernel functions. As for the batch-SGD, both [Allen-Zhu et al., 2019a] and [Zou and Gu, 2019] show the training error of over-parametrized deep neural networks converges to 0. However, in both works, after proper scaling of the number of neurons m, the step size needed is still of order $\Theta(\frac{1}{\log m})$. For over-parametrized neural networks, this leads to an extremely small step size that is not commonly used in practise [Bengio, 2012]. In contrast, in our study, the step size does not decay with the number of neurons m.

Generalization error with batch learning. Following [Du et al., 2019b], [Arora et al., 2019] derives an upper bound of the generalization error of over-parametrized two-layer neural networks under GD as $\sqrt{\frac{2y^\top H^{-1}y}{n}} + O(\sqrt{\frac{\log(n/\lambda_{\min}(H))}{n}})$, where $y\in\mathbb{R}^n$ is the label of the sample. As mentioned above, $\lambda_{\min}(H)$ decreases to 0 and hence the generalization error blows up to infinity as n grows. In [Ma et al., 2019], the authors consider the minimum-norm estimator $(\widehat{a},\widehat{W}) = \operatorname{argmin}\left\{\frac{1}{m}\sum_{i=1}^m |a_i|\|w_i\|_1 : R_n(a,W)=0\right\}$ for two-layer ReLU activated neural networks $f(x;a,W)=\frac{1}{m}a^\top\sigma(Wx)$, where $R_n=\frac{1}{2n}\sum_{i=1}^n \left(f(x_i;a,W)-y_i\right)^2$ is the empirical loss over the batch $\{(x_i,y_i)\}_{i=1}^n$. They show that a generalization error of order $O\left(\sqrt{\frac{\log(2d)}{n}}\right)$ can be achieved, provided that the number of neurons $m\ge\frac{2n^2\log(4n^2)}{\lambda^2_{\min}(H)}$. However, how to efficiently compute such minimum-norm estimator is unknown, while the estimator from one-pass SGD is easy-to-compute and also widely used in practice.

Generalization error with streaming data Similar to our work, [Cao and Gu, 2019] also considers the one-pass SGD in the streaming setting. The authors apply the online-to-batch conversion proposed in [Cesa-Bianchi et al., 2004] to bound the generalization error $\frac{1}{T}\sum_{s=1}^T \mathbb{E}_{(X,y)}\left[\mathbf{1}_{\{yf(X;W(s))<0\}}\right]$ from above by the empirical loss $\frac{1}{T}\sum_{s=1}^T\mathcal{L}\left(y_s f(x_s;W(s)\right)$ with the hinge loss function $\mathcal{L}(z)=\log(1+\exp(-z))$. Note the online-to-batch conversion follows from an application of martingale concentration equalities. It does not resolve the problem of bounding the generalization error as one still needs to bound the cumulative loss. Indeed the authors bound the cumulative loss following a similar analysis of [Du et al., 2019b] and obtain an upper bound of the generalization error as $O\left(\sqrt{\frac{y^\top H^{-1}y}{T}}\right)+O\left(\sqrt{1/T}\right)$. However, as T increases, $\lambda_{\min}(H)$ decreases to 0 and hence the upper bound which depends on H^{-1} may blow up. On the contrary, our study proves that the average prediction error can indeed be very small. In addition, after proper scaling of the number of neurons m, the step size considered in [Cao and Gu, 2019] is $O\left(\frac{1}{\sqrt{mT}}\right)$, which is extremely small in the overparameterized neural networks.

3 Main Result

3.1 Problem Setup

Given $f^*\in L^2(\mu)$, we assume the data (X,y) is given by $y=f^*(X)+e$, where $X\in\mathbb{R}^d$ is generated according to distribution μ on the unit sphere $\mathbb{S}^{d-1}$, and e is the noise independent of X with mean 0 and variance τ^2. We consider the following two-layer neural network:

$$f(x;W)=\frac{1}{\sqrt{m}}\sum_{i=1}^m a_i\sigma(\langle W_i,x\rangle)$$

where $a_i\in\{\pm1\}$, $\sigma(x)=\max\{0,x\}$ is the ReLU activation function, and $W\in\mathbb{R}^{m\times d}$ is the weight matrix with the i-th row denoted as W_i.

The neural network is trained by running the stochastic gradient descent (SGD) on the streaming data in one pass. In particular, we assume the outer weights a_i's are *i.i.d.* Rademacher random variables and fixed throughout the training process. The weight matrix $W(0)\in\mathbb{R}^{m\times d}$ is initialized as the Gaussian random matrix with *i.i.d.* standard normal entries. Then we update the weight matrix at the t-th iteration as

$$W(t+1)=W(t)-\eta_{t+1}\nabla_W l(W(t),X_t,y_t), \qquad (1)$$

where η_{t+1} is the step size, $l(W,x,y) = \frac{1}{2}\left(y-f\left(x;W\right)\right)^2$ is the quadratic loss function, and (X_t, y_t) is the fresh data independently and identically distributed as (X, y).

3.2 Main Theorem

Denote the prediction error $\Delta_t(x) = f^*(x) - f(x; W(t))$. Let $\sigma_t^2 = \mathbb{E}\left[\|\Delta_t\|_2^2\right] + \tau^2$. Our main result characterizes the convergence of the average prediction error in terms of the spectrum of certain integrator operator. Define the (neural tangent) kernel function

$$\Phi(x, x') = x^T x' \mathbb{E}_{w \sim N(0, I_d)}\left[\mathbf{1}_{\{w^T x \geq 0\}} \mathbf{1}_{\{w^T x' \geq 0\}}\right]$$

and the integral operator $\boldsymbol{\Phi}$ associated with Φ as

$$\boldsymbol{\Phi} g(x) = \int \Phi(x, x') g(x') \mu(dx'), \quad \forall g \in L^2(\mu).$$

Denote the eigenvalues of $\boldsymbol{\Phi}$ as $\{\lambda_i\}_{i=1}^{\infty}$ with $\lambda_1 \geq \lambda_2 \cdots$ and the corresponding eigenfunctions ϕ_i. For any function $g \in L^2(\mu)$, denote $\mathcal{R}(g, \ell)$ as the L_2 norm of the projection of function g onto the space spanned by the eigenfunctions $\{\phi_i\}_{i=\ell+1}^{\infty}$, i.e.,

$$\mathcal{R}(g, \ell) = \sum_{i=\ell+1}^{\infty} \langle g, \phi_i \rangle^2.$$

Theorem 1. *Suppose the step size* $\eta_t \leq \frac{\theta}{t+1}$ *with* $\theta < \frac{1}{4}$. *For any* $T < \infty$, *if*

$$m \geq c\left(d^2 + \max\left\{\left(\frac{(T+1)^{2\theta}}{\theta}\right)^9, \left(\frac{\theta \log(T)}{\delta}\right)^9\right\}\right)$$

for some universal constant $c > 0$ *and some* $\delta > 0$, *then with probability at least* $1 - 2\exp(-2m^{1/3})) - \delta$, $\forall 0 \leq t \leq T$,

$$\begin{aligned} &\mathbb{E}\left[\|\Delta_t\|_2 | W(0)\right] \\ &\leq \inf_{\ell}\left\{\prod_{k=0}^{t-1}(1 - \eta_k \lambda_\ell)\|\Delta_0\|_2 + \mathcal{R}(\Delta_0, \ell)\right\} + 2c_1, \quad (2) \end{aligned}$$

where $c_1 = \sigma_0 \sqrt{\frac{e^{4\theta}\theta^2(2-4\theta)}{1-4\theta}}$.

Remark 3.1. *Under a symmetric initialization motivated by [Su and Yang, 2019] and also used in [Chizat et al., 2019],* $\Delta_0 = f^*$ *and hence* $\|\Delta_0\|_2 = \|f^*\|_2$. *Specifically, first let* $W(0) = \begin{pmatrix} W \\ W \end{pmatrix}$, *where* $W \in \mathbb{R}^{\frac{m}{2} \times d}$ *is random matrix with i.i.d. standard normal* $N(0,1)$ *entries. Then let the outer weights* $a = (b, -b)^T \in \mathbb{R}^m$, *where* $b \in \{\pm 1\}^{m/2}$ *has i.i.d. Rademacher entries.*

Furthermore, under this initialization, following [Su and Yang, 2019], if f^* *is a degree* ℓ^* *polynomial,* $\ell^* \geq 0$ *and* μ *is the uniform distribution on* $\mathbb{S}^{d-1}$, *we know* $\mathcal{R}(f^*, \ell^*) = 0$. *Thus, with high probability,*

$$\mathbb{E}\left[\|\Delta_t\|_2\right] \leq \prod_{k=0}^{t-1}(1 - \eta_k \lambda_{\ell^*})\|\Delta_0\|_2 + 2c_1, \forall 0 \leq t \leq T.$$

Assume $\|f^*\|_2 = 1$ *and* $\tau = 0$, *then for any* $\varepsilon \in (0, 0.5)$, *if* $\theta = \frac{\varepsilon}{4}$, $T = \left(\frac{\varepsilon}{3\|f^*\|_2}\right)^{-1/(\theta\lambda_{\ell^*})}$, $m \geq c\left(d^2 + \max\left\{\left(\frac{(T+1)^{2\theta}-1}{\theta}\right)^9, \frac{\theta^9 \log^9(T)}{\delta^9}\right\}\right)$ *for* $c \geq 14^5(2C_2 + C_3)^3$ *and a small* δ, *we have* $\mathbb{E}\left[\|\Delta_T\|_2\right] \leq \varepsilon$ *with high probability.*

For more general f^*, *there is no guarantee that* $\mathcal{R}(\Delta_0, \ell) = 0$ *for some* $\ell < \infty$. *We provide a way to compute the eigenvalues* λ_ℓ *and the projection* $\mathcal{R}(f^*, \ell)$ *in the Supplementary Material.*

Remark 3.2. *Note that the lower bound of* m *grows in* T. *In order to control* m, *we adopt the early stopping assumption* $T < \infty$ *which is commonly used in practice as shown in [Su and Yang, 2019].*

Remark 3.3. *In terms of generalization error, following a similar analysis of [Arora et al., 2019, Section D.3], we know that if the loss function* $l : \mathbb{R} \times \mathbb{R} \to [0, 1]$ *is* 1*-Lipschitz in the first argument with* $l(z, z) = 0$ *and* $\tau = 0$, *then* $\mathbb{E}\left[\mathcal{L}(W(t)) | W(0)\right] \leq \mathbb{E}\left[\|\Delta_t\|_2 | W(0)\right]$ *where* $\mathcal{L}(W(t)) = \mathbb{E}_X\left[l\left(f\left(X; W(t)\right), y\right)\right]$. *In other words, our result in averaged prediction error can be viewed as an upper bound of the expectation of generalization error.*

Our result sheds light on the trade-off between the convergence rate and the accumulation of approximation errors. The trade-off is two-fold. One is between $\prod_{k=0}^{t}(1 - \frac{\theta\lambda_\ell}{k})$ and $\mathcal{R}(f^*, \ell)$ through ℓ. Denote the principle space $\mathcal{P}_\ell$ as the space spanned by the first ℓ eigen-functions of $\boldsymbol{\Phi}$ and the space spanned by $\ell + 1, \ell + 2, \cdots$ eigen-functions of $\boldsymbol{\Phi}$ as the remainder space $\mathcal{P}_\ell^\perp$. Intuitively, on one hand, larger ℓ implies larger principle spaces which yields smaller $\mathcal{R}(f^*, \ell)$. On the other hand, larger ℓ also implies smaller λ_ℓ. Thus the contraction factor $\prod_{k=0}^{t}(1 - \frac{\theta\lambda_\ell}{k})$ is smaller, indicating slower convergence. The other trade-off is between the contraction factor $\prod_{k=0}^{t}(1 - \frac{\theta\lambda_\ell}{k})$ and the accumulation of approximation error and noise c_1 through θ. To make sure c_1 is small, we need small θ, thus yielding a small contraction factor. In return, we need more iterations to converge.

4 Proof Sketch of Theorem 1

Throughout this section, we condition on the initialization $W(0)$ and outer weights a. The expectation $\mathbb{E}[\cdot]$ is taken over the randomness of the samples drawn at iterations, unless specified otherwise. The complete proof of Theorem 1 is deferred to the supplementary material.

4.1 Proof Overview

We prove (2) via induction over iteration t. The base case $t = 0$ trivially holds as $\|\Delta_0\|_2 \le \|\Delta_0\|_2 + 2c_1$. Assume (2) holds for any $s \le t \le T$, we first show $\mathbb{E}\left[\|W(s+1) - W(0)\|_{\mathrm{F}}\right]$ is small for any $s \le t$.

Lemma 4.1. *For any $t \ge 0$,*

$$\mathbb{E}\left[\|W(t+1) - W(0)\|_{\mathrm{F}}\right] \le \sum_{s=0}^{t} \eta_s \left(\mathbb{E}\left[\|\Delta_s\|_2\right] + \tau\right).$$

The proof of Lemma 4.1 is based on bounding the SGD update given in (1). Plugging the induction hypothesis into Lemma 4.1 and noting $\mathbb{E}\left[\|\Delta_s\|_2\right] \le \|\Delta_0\|_2 + 2c_1$, we get

$$\begin{aligned}&\mathbb{E}\left[\|W(s+1) - W(0)\|_{\mathrm{F}}\right] \\ &\quad \le \left(\|\Delta_0\|_2 + \tau + 2c_1\right) \theta \left(\log(s) + 1\right). \end{aligned} \tag{3}$$

The induction is then completed by the following proposition.

Proposition 4.2. *Suppose the conditions in Theorem 1 hold. If (3) holds for any $s \le t \le T-1$, then (2) holds for $t+1$ with probability at least $1 - 2\exp(-m^{1/3}) - \delta$ over initialization.*

4.2 Proof Sketch of Proposition 4.2

Firstly, we adopt a similar analysis in [Su and Yang, 2019] to obtain the following recursive form of Δ_t:

$$\Delta_{t+1} = \mathsf{Q}_t \circ \Delta_t - v_t + \epsilon_t, \tag{4}$$

where $\mathsf{Q}_t = \mathsf{I} - \eta_t \mathsf{H}_t$ and H_t is the integral operator associated with kernel function H_t,

$$H_t(x, \widetilde{x}) \triangleq \frac{1}{m} \langle x, \widetilde{x} \rangle \sum_{i=1}^{m} \mathbf{1}_{\{\langle W_i(t), x\rangle \ge 0\}} \mathbf{1}_{\{\langle W_i(t), \widetilde{x}\rangle \ge 0\}}.$$

The term v_t measures the noise brought by SGD and is given by

$$\begin{aligned} v_t(x, X_t) \triangleq\ & \eta_t H_t(x, X_t) \left[\Delta_t(X_t) + e_t\right] \\ & - \eta_t \mathbb{E}_{X_t}\left[H_t(x, X_t)\Delta_t(X_t)\right]. \end{aligned}$$

The term ϵ_t captures the perturbation caused by the non-linearity of the ReLU activation function and can be bounded as

$$\begin{aligned} &\left|\epsilon_t(x, X_t)\right| \\ &\le \eta_t \max\left\{\|M_t\|_\infty, \|L_t\|_\infty\right\} \left|\Delta_t(X_t) + e_t\right|, \end{aligned} \tag{5}$$

where

$$L_t(x, x') = \frac{1}{m}\langle x, x'\rangle \sum_{i: a_i = 1} \mathbf{1}_{\{\langle W_i(t), x'\rangle \ge 0\}} \delta_t(x),$$

$$M_t(x, x') = \frac{1}{m}\langle x, x'\rangle \sum_{i: a_i = -1} \mathbf{1}_{\{\langle W_i(t), x'\rangle \ge 0\}} \delta_t(x),$$

$$\delta_t(x) = \mathbf{1}_{\{\langle W_i(t+1), x\rangle \ge 0\}} - \mathbf{1}_{\{\langle W_i(t), x\rangle \ge 0\}}.$$

Both L_t and M_t measure the number of sign changes of neurons between t and $t+1$. Define the number of sign changes of neurons between t and 0 as

$$S_t(x) = \left|\left\{i \in [m] : \operatorname{sgn}(\langle W_i(t), x\rangle) \ne \operatorname{sgn}(\langle W_i(0), x\rangle)\right\}\right|.$$

In view of the definition of H_t, if $S_t(x)$ is small for any x and t, then we expect H_t to be close to H_0. Furthermore, at the initialization, note that $\mathbb{E}_{W(0)}[H_0] = \Phi$ and thus we expect H_0 concentrates on the NTK function Φ. By the triangle inequality, we get that H_t is close to Φ and hence Q_t is close to $\mathsf{K}_t \triangleq \mathsf{I} - \eta_t \mathsf{\Phi}$. To capture this idea, we unroll the recursion (4) and decompose Q_t into $\mathsf{K}_t + \mathsf{D}_t$ to obtain

$$\begin{aligned} \Delta_{t+1} =& \prod_{s=0}^{t} \mathsf{K}_s \circ \Delta_0 + \sum_{r=0}^{t} \left(\prod_{i=r+1}^{t} \mathsf{Q}_i \mathsf{D}_r \prod_{j=0}^{r-1} \mathsf{K}_j \circ \Delta_0 \right) \\ & + \sum_{r=0}^{t} \left(\prod_{s=r+1}^{t} \mathsf{Q}_s \circ (\epsilon_r - v_r) \right), \end{aligned} \tag{6}$$

where $\mathsf{D}_t = \mathsf{Q}_t - \mathsf{K}_t$.

Taking the L_2 norm and conditional expectation on both hand sides of (6), and using the triangle inequality and the fact that $\|\mathsf{Q}_t\|_2 \le 1$ and $\|\mathsf{K}_t\|_2 \le 1$, we get

$$\begin{aligned} \mathbb{E}\left[\|\Delta_{t+1}\|_2\right] \le & \left\| \prod_{s=0}^{t} \mathsf{K}_s \circ \Delta_0 \right\|_2 + \sum_{r=0}^{t} \mathbb{E}\left[\|\mathsf{D}_r\|_2\right] \|\Delta_0\|_2 \\ & + \mathbb{E}\left[\left\| \sum_{r=0}^{t} \prod_{s=r+1}^{t} \mathsf{Q}_s \circ v_r \right\|_2 \right] + \sum_{r=0}^{t} \mathbb{E}\left[\|\epsilon_r\|_2\right]. \end{aligned} \tag{7}$$

Next we analyze each term in (7).

The next result provides an upper bound of the first term of (7) in terms of the eigen-decomposition of $\mathsf{\Phi}$.

Lemma 4.3. *Suppose $\eta_s \lambda_1 < 1$ for any $s \le t$, then,*

$$\left\| \prod_{s=0}^{t} \mathsf{K}_s \circ \Delta_0 \right\|_2 \le \inf_{\ell} \left\{ \prod_{s=0}^{t} (1 - \eta_s \lambda_\ell) \|\Delta_0\|_2 + \mathcal{R}(\Delta_0, \ell) \right\}.$$

To bound the second term of (7), note that $\|\mathsf{D}_s\|_2 = \|\mathsf{Q}_s - \mathsf{K}_s\|_2 \le \eta_s \|H_s - \Phi\|_\infty$. Lemma 4.4 provides an upper bound of $\|H_s - \Phi\|_\infty$ under event $\Omega_1 \cap \Omega_2$, where

$$\begin{aligned} \Omega_1 = \Bigg\{ \sup_{x, R} \Bigg| & \frac{1}{m} \sum_{i=1}^{m} \mathbf{1}_{\{|\langle W_i(0), x\rangle| \le R\}} \\ & - \mathbb{E}\left[\mathbf{1}_{\{|\langle w, x\rangle| \le R\}}\right] \Bigg| \le \frac{1}{m^{1/3}} + C_2 \sqrt{\frac{d}{m}} \Bigg\}, \end{aligned}$$

and

$$\Omega_2 = \Bigg\{ \sup_{x,\widetilde{x}} \bigg| \frac{1}{m}\sum_{i=1}^{m} \mathbf{1}_{\{\langle W_i(0),x\rangle \ge 0\}}\mathbf{1}_{\{\langle W_i(0),\widetilde{x}\rangle \ge 0\}} - \mathbb{E}\left[\mathbf{1}_{\{\langle w,x\rangle \ge 0\}}\mathbf{1}_{\{\langle w,\widetilde{x}\rangle \ge 0\}})\right]\bigg| \le \frac{1}{m^{1/3}} + C_3\sqrt{\frac{d}{m}} \Bigg\}$$

for some universal constant C_2, C_3 and $w \sim N(0, I_d)$.

Both events are defined with respect to the initial randomness $W(0)$, and require the sample mean of some function of $W_i(0)$ to be close to the expectation. Since $W_i(0)$'s are *i.i.d.* Gaussian, using uniform concentration inequalities, we will show in Section 4.3 that both Ω_1 and Ω_2 occur with high probability when m is large.

Lemma 4.4. *Under Ω_2, for any $t \ge 0$,*

$$\|H_t - \Phi\|_\infty \le \frac{2}{m}\|S_t\|_\infty + C_3\sqrt{\frac{d}{m}} + \frac{1}{m^{1/3}}.$$

Lemma 4.4 provides an upper bound to $\|H_t - \Phi\|_\infty$ in terms of the number of sign changes $\|S_t\|_\infty$. Lemma 4.4 directly follows from the triangle inequality $\|H_t - \Phi\|_\infty \le \|H_t - H_0\|_\infty + \|H_0 - \Phi\|_\infty$ and the definition of Ω_2.

The following result further shows that when $\|W(t) - W(0)\|_{\mathrm{F}}$ is small and m is large, under Ω_1, $\|S_t\|_\infty$ is small.

Lemma 4.5. *Under Ω_1,*

$$\|S_t\|_\infty \le m^{\frac{2}{3}} + C_2\sqrt{md} + \frac{2^{\frac{4}{3}} m^{\frac{2}{3}} \|W(t) - W(0)\|_{\mathrm{F}}^{\frac{2}{3}}}{\pi^{1/3}}.$$

For the third term of (7), we utilize $\|\mathsf{Q}_t\|_2 \le 1$ and the fact that v_t is a martingale difference sequence to bound the accumulation of the noise, that is

$$\mathbb{E}\left[\left\|\sum_{r=0}^{t}\prod_{s=r+1}^{t}\mathsf{Q}_s \circ v_r\right\|_2\right] \le \sqrt{\sum_{r=0}^{t}\|v_r\|_2^2}.$$

Then we show $\mathbb{E}\left[\|v_t\|_2^2\right] \le \eta_t^2\sigma_t^2$ to obtain Lemma 4.6.

Lemma 4.6. *Suppose $0 \le \eta_s \le 2$ for any $s \ge 0$, then,*

$$\mathbb{E}\left[\left\|\sum_{s=0}^{t}\prod_{i=s+1}^{t}\mathsf{Q}_i \circ v_s\right\|_2\right] \le \sqrt{\sum_{s=0}^{t}\eta_s^2\sigma_s^2}.$$

We see the third term depends on σ_t. The next lemma shows σ_t does not grow fast in t.

Lemma 4.7. *For any $t \ge 0$,*

$$\sigma_{t+1}^2 \le \prod_{s=0}^{t}(1+2\eta_s)^2\sigma_0^2.$$

Plugging Lemma 4.7 into Lemma 4.6, we obtain

$$\mathbb{E}\left[\left\|\sum_{s=0}^{t}\prod_{i=s+1}^{t}\mathsf{Q}_i \circ v_s\right\|_2\right] \le c_1. \tag{8}$$

For the fourth term of (7), taking the L_2 norm and conditional expectation of (5), we have

$$\mathbb{E}\left[\|\epsilon_r\|_2\right] \le \eta_r\sqrt{\mathbb{E}\left[\|L_r\|_\infty^2 + \|M_r\|_\infty^2\right]}\sigma_r. \tag{9}$$

It remains to bound $\mathbb{E}\left[\|L_r\|_\infty^2\right]$. Note

$$\mathbb{E}\left[\|L_r\|_\infty^2\right] = \mathbb{E}\left[\|L_r\|_\infty^2\mathbf{1}_{\{\mathcal{A}\}}\right] + \mathbb{E}\left[\|L_r\|_\infty^2\mathbf{1}_{\{\mathcal{A}^c\}}\right], \tag{10}$$

where

$$\mathcal{A} = \Bigg\{\max\Big(\|W(r+1) - W(0)\|_{\mathrm{F}}, \|W(r) - W(0)\|_{\mathrm{F}}\Big) \le m^{1/3}\Bigg\}.$$

Through the following Lemma 4.8 and Lemma 4.5, we can upper bound the first component of (10) as

$$\mathbb{E}\left[\|L_r\|_\infty^2\mathbf{1}_{\{\mathcal{A}\}}\right] \le \left[\frac{2}{m^{1/3}} + 2C_2\sqrt{\frac{d}{m}} + \frac{2^{10/3}}{\pi^{1/3}m^{1/9}}\right]^2. \tag{11}$$

Lemma 4.8.

$$\max\left\{\|L_t\|_\infty, \|M_t\|_\infty\right\} \le \frac{1}{m}\|S_t\|_\infty + \frac{1}{m}\|S_{t+1}\|_\infty.$$

Intuitively, if the weight matrix is close to the initialization at iteration t and $t+1$, we expect the number of sign changes S_t and S_{t+1} to be small for any x. Small $\|S_t\|_\infty$ and $\|S_{t+1}\|_\infty$ then lead to small $\|L_t\|_\infty$ and $\|M_t\|_\infty$.

For the second component of (10), note

$$\begin{aligned}\mathbb{E}\left[\|L_r\|_\infty^2\mathbf{1}_{\{\mathcal{A}^c\}}\right] &\overset{(a)}{\le} \mathbb{E}\left[\mathbf{1}_{\{\mathcal{A}^c\}}\right] \\ &\le \mathbb{P}\left[\|W(r+1) - W(0)\|_{\mathrm{F}} > m^{1/3}\right] \\ &\quad + \mathbb{P}\left[\|W(r) - W(0)\|_{\mathrm{F}} > m^{1/3}\right],\end{aligned} \tag{12}$$

where (a) holds by the fact $\|L_r\|_\infty \le 1$.

By (3) and Markov's inequality, we get for $s \in \{r, r+1\}$,

$$\begin{aligned}&\mathbb{P}\left[\|W(s) - W(0)\|_{\mathrm{F}} > m^{1/3}\right] \\ &\le \frac{(\|\Delta_0\|_2 + \tau + 2c_1)\,\theta\,(\log(T) + 1)}{m^{1/3}}.\end{aligned} \tag{13}$$

Plugging (13) into (12) and combining with (11), we have

$$\mathbb{E}\left[\left\|L_r\right\|_\infty^2\right] \le \left[\frac{2}{m^{1/3}} + 2C_2\sqrt{\frac{d}{m}} + \frac{2^{10/3}}{\pi^{1/3}m^{1/9}}\right]^2 + \frac{2\left(\left\|\Delta_0\right\|_2 + \tau + 2c_1\right)\theta\left(\log(T)+1\right)}{m^{1/3}}. \tag{14}$$

Denote $\Omega_3 = \left\{\left\|\Delta_0\right\|_2 \le \frac{\sqrt{\|f^*\|_2+1}}{\delta}\right\}$ for $0 < \delta < 1$. Under Ω_3, we can further bound RHS of (14) in terms of δ.

Therefore, with m sufficiently large, $\mathbb{E}\left[\left\|L_r\right\|_\infty^2\right]$ is small. The result for $\mathbb{E}\left[\left\|M_t\right\|_\infty^2\right]$ can be obtained analogously.

Applying Lemma 4.7 again with (14) and (9), we obtain

$$\sum_{r=0}^{t}\mathbb{E}\left[\left\|\epsilon_r\right\|_2\right] \le \frac{\sqrt{14}e^{2\theta}\left[(t+2)^{2\theta}-1\right]\sigma_0}{m^{1/9}} \tag{15}$$

for a sufficiently large m.

Combining Lemma 4.3, Lemma 4.4, (8) and (15), we get under $\Omega_1 \cap \Omega_2 \cap \Omega_3$, provided that $m \ge c\left(d^2 + \max\left\{\left(\frac{(T+1)^{2\theta}-1}{\theta}\right)^9, \frac{\theta^9\log^9(T)}{\delta^9}\right\}\right)$ for some universal constant c, we have

$$\begin{aligned}&\mathbb{E}\left[\left\|\Delta_{t+1}\right\|_2\right]\\ &\le \inf_r\left\{\prod_{s=0}^{t}\left(1-\eta_s\lambda_r\right)\left\|\Delta_0\right\|_2 + \mathcal{R}(\Delta_0, r)\right\}\\ &+ 2c_1.\end{aligned}$$

It remains to show $\cap_{i=1}^3\Omega_i$ occurs with probability at least $1-\delta-2\exp(-m^{1/3})$.

Lemma 4.9. *For any* $0<\delta<1$,

$$\mathbb{P}\left[\Omega_3\right] \ge 1-\delta.$$

The proof of Lemma 4.9 follows by $\mathbb{E}_{a,W(0)}\left[\left\|\Delta_0\right\|_2^2\right] \le \left\|f^*\right\|_2^2 + 1$ and Markov inequality.

We complete the proof by showing both Ω_1 and Ω_2 occur with probability at least $1-\exp(-m^{1/3})$.

Lemma 4.10.

$$\begin{aligned}\mathbb{P}\left[\Omega_1\right] &\ge 1-\exp(-2m^{1/3}),\\ \mathbb{P}\left[\Omega_2\right] &\ge 1-\exp(-2m^{1/3}).\end{aligned}$$

Remark 4.1. *In Lemma 4.10, we use the VC-dimension and McDiarmid's inequality to obtain the uniform control of* $\left\|H_0-\Phi\right\|_\infty$*. This significantly deviates from the existing literature such as [Du et al., 2019b, Du et al., 2019a, Su and Yang, 2019, Allen-Zhu et al., 2019a, Zou et al., 2020, Arora et al., 2019] that studies the batch setting and obtains the uniform control via pointwise control and union bound. More specifically, in the batch setting with* n *data points* $\{(x_i,y_i)\}_{i=1}^n$*, similar to* Ω_2 *we can define event* $\Omega_2' = \cup_{i,j}\Omega_{i,j}$*, where*

$$\Omega_{i,j} = \left\{W(0) : \left|\frac{1}{m}\left(\sum_{k=1}^{m}\mathbf{1}_{\{\langle W_k(0),x_i\rangle\ge 0\}}\mathbf{1}_{\{\langle W_k(0),x_j\rangle\ge 0\}} - \mathbb{E}_w\left[\mathbf{1}_{\{\langle w,x_i\rangle\ge 0\}}\mathbf{1}_{\{\langle w,x_j\rangle\ge 0\}}\right]\right)\right| < \frac{C_4}{m^{1/3}}\right\}.$$

for some constant C_4.

Then we can show Ω_2' *occurs with high probability by bounding the probability of each individual* $\Omega_{i,j}$ *and applying a union bound. However, such techniques are not directly applicable in the streaming data setting to obtain the desired uniform control on the kernel functions.*

4.3 Proof of Lemma 4.10

Here, we provide the proof of Lemma 4.10 to highlight our new proof strategy. In particular, we show the conclusion for Ω_2; the conclusion for Ω_1 follows analogously. Denote

$$\phi(\mathbf{w}) = \sup_{x,\widetilde{x}}\left|\frac{1}{m}\sum_{i=1}^{m}\mathbf{1}_{\{\langle w_i,x\rangle\ge 0\}}\mathbf{1}_{\{\langle w_i,\widetilde{x}\rangle\ge 0\}} - \mathbb{E}_{w\sim N(0,I_d)}\left[\mathbf{1}_{\{\langle w,x\rangle\ge 0\}}\mathbf{1}_{\{\langle w,\widetilde{x}\rangle\ge 0\}}\right]\right|.$$

where $\mathbf{w} = (w_1, w_2, \cdots, w_m)$.

By triangle inequality, we have

$$\left|\phi(\mathbf{w}) - \phi(w_1,\cdots,w_{i-1},w_i',w_{i+1},\cdots,w_m)\right| \le \frac{1}{m}.$$

Let $W_1,\ldots,W_m$ denote m *i.i.d.* $\mathcal{N}(0,\mathbf{I}_d)$. Thus, by McDiarmid's inequality, we get

$$\begin{aligned}&\mathbb{P}\left[\phi(W_1,\cdots,W_m) \ge m^{-1/3} + \mathbb{E}\left[\phi(W_1,\cdots,W_m)\right]\right]\\ &\le \exp\left(-2m^{1/3}\right).\end{aligned}$$

The proof is then completed by invoking the following claim

$$\mathbb{E}\left[\phi(W_1,\cdots,W_m)\right] \le C_3\sqrt{\frac{d}{m}}.$$

To prove the claim, by [Vershynin, 2019, Theorem 8.3.23], it suffices to show that the VC dimension of $\mathcal{F}_1$

is upper bounded by $C'd$ for some universal constant C', where

$$\mathcal{F}_1 = \left\{ f_{x,x'} : f_{x,x'}(w) = \mathbf{1}_{\{\langle w,x\rangle \geq 0\}} \mathbf{1}_{\{\langle w,x'\rangle \geq 0\}} \right\}.$$

To see this, first we can show that $\text{VC}(\mathcal{G}) = d$, where $\mathcal{G} = \left\{ g_x : g_x(w) = \mathbf{1}_{\{\langle w,x\rangle \geq 0\}} \right\}$.

For any boolean function g, define $D_g = \left\{ w : w \in \mathbb{R}^d, g(w) = 1 \right\}$ and $\mathcal{C}_{\mathcal{F}} \triangleq \{D_g, g \in \mathcal{F}\}$. We are now going to show $\mathcal{C}_{\mathcal{F}_1} = \mathcal{C}_{\mathcal{G}} \sqcap \mathcal{C}_{\mathcal{G}}$ where $\mathcal{C}_1 \sqcap \mathcal{C}_2 \triangleq \{C_1 \cap C_2 : C_j \in \mathcal{C}_j, j = 1, 2\}$. To see this, note for any $f \in \mathcal{F}_1$, we can find g_1 and g_2 in $\mathcal{G}$ such that $D_f = D_{g_1} \cap D_{g_2}$. In particular, if $f(w) = \mathbf{1}_{\{\langle w,x_1\rangle \geq 0\}} \mathbf{1}_{\{\langle w,x_2\rangle \geq 0\}}$, we can take $g_1(w) = \mathbf{1}_{\{\langle w,x_1\rangle \geq 0\}}$ and $g_2(w) = \mathbf{1}_{\{\langle w,x_2\rangle \geq 0\}}$. Similarly, for any $g_1, g_2 \in \mathcal{G}$, $D_{g_1} \cap D_{g_2} = D_f$ for some $f \in \mathcal{F}_1$. Then we get $\text{VC}(\mathcal{F}_1) \leq C'\text{VC}(\mathcal{G}) = C'd$ for some universal constant C' by invoking [Van Der Vaart and Wellner, 2009, Theorem 1.1]. Detailed proof of the claim is deferred to the supplementary material.

5 Numerical Study

In this section, we present two numerical studies to support our theoretical analysis. More numerical experiments can be found in the supplementary material.

We consider the following different choices of f^*:

- Linear: $f^*(x) = \langle b, x\rangle$ with $b \sim N(0, I_d)$.
- Quadratic: $f^*(x) = x^\top Ax + \langle b, x\rangle$, where both $A \in \mathbb{R}^{d\times d}$ and $b \in \mathbb{R}^d$ have *i.i.d.* $N(0, 1)$ entries.
- Teacher neural network: $f^*(x) = \sum_{i=1}^{3} b_i \psi(\langle v_i, x\rangle)$, where $\psi(z) = \frac{1}{1+e^{-z}}$ is the sigmoid function, b_i's are *i.i.d.* Rademacher random variables, and $v_i \sim N(0, I_d)$.
- Random Label: $f^*(x)$ are *i.i.d.* Bernoulli random variables with parameter $\frac{1}{2}$ across all x.

We run the stochastic gradient descent algorithm (1) on the streaming data with constant step size $\eta = 0.2$. We assume the symmetric initialization introduced in Remark 3.1 to ensure the initial prediction error $\Delta_0 = f^*$. At each iteration, we randomly draw data X uniformly from $\mathbb{S}^{d-1}$ and e from $N(0, \tau^2)$ to obtain (X, y) where $y = f^*(X) + e$. The average prediction error is estimated using freshly drawn 400 data points, and the resulting error is further averaged over 20 independent runs.

Figure 1 shows the dynamic (solid lines) of the average prediction error normalized by the error at initialization $\sqrt{\|f^*\|_2^2 + \tau^2}$ for different f^* with $d = 5$, $m = 1000$, and $\tau = 0.1$. The dashed lines represent the optimal (normalized) average prediction error, which is $\frac{\tau}{\sqrt{\|f^*\|_2^2+\tau^2}}$ for linear, quadratic and teacher neural network and is $\frac{\sqrt{1/4+\tau^2}}{\sqrt{\|f^*\|_2^2+\tau^2}}$ for the random label case. Figure 1 shows that SGD is able to learn all the four f^* cases efficiently: the normalized average prediction error converges to the best achievable value. Besides, we see a difference in the convergence rate among different f^*: The convergence is the fastest in the linear case and the slowest in the random label case. This is consistent with our theory as a larger principle space (larger r) is needed for the random label function to have relatively small $\mathcal{R}(f^*, r)$, resulting in a smaller eigenvalue λ_r for the convergence rate.

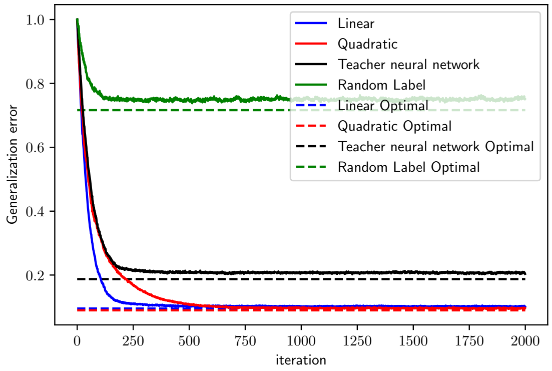
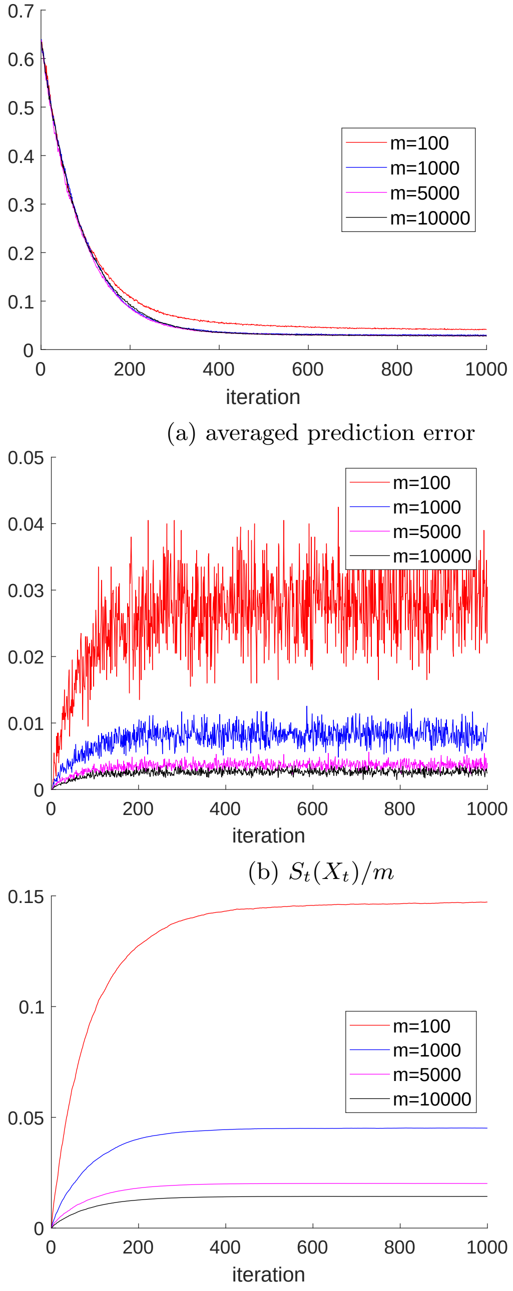


Figure 1: Averaged prediction error under SGD for different target function f^*

Figure 2 considers the setting with a varying number of hidden neurons m, when f^* is teacher neural network and $d = 5$. Figure 2a shows the dynamic of averaged generalization error. The convergence becomes faster when m increases from 100 to 1000, but there is not much difference when m is increased further. This is consistent with our theory, because when m is large enough, the random kernel H_t is already well approximated by the Neural Tangent Kernel Φ. Indeed we observe a small proportion of sign changes from figure 2b when m is above 1000, which leads to a small approximation error ϵ_t in view of Lemma 4.8 and Lemma 4.5. Figure 2c shows the relative deviation of the weight matrix at iteration from the initialization. Following Lemma 4.1, we see $\|W(t) - W(0)\|_{\text{F}} = O(t)$ while $\|W(0)\|_{\text{F}} = O(\sqrt{md})$. As a result, we see $\frac{\|W(t)-W(0)\|_{\text{F}}}{\|W(0)\|_{\text{F}}}$ decreases as m increases for fixed t and $\frac{\|W(t)-W(0)\|_{\text{F}}}{\|W(0)\|_{\text{F}}}$ increases as t grows for fixed m.



(a) averaged prediction error

(b) $S_t(X_t)/m$

(c) $\frac{\|W(t)-W(0)\|_{\mathrm{F}}}{\|W(0)\|_{\mathrm{F}}}$

Figure 2: Comparison of different number of neurons with teacher neural network f^*

6 Conclusion

In this paper, we provide an upper bound to the average prediction error of two-layer neural networks under the one-pass SGD in the streaming data setup, utilizing the eigen-decomposition of the neural tangent kernel Φ. Our analysis relies on proving the uniform convergence of the kernel functions via the VC dimension and McDiarmid inequality. This technique may be also useful for analyzing multi-layer feed-forward neural networks and other types of neural networks.

Acknowledgement

This research is supported in part by the NSF Grants IIS-1838124, CCF-1850743 and CCF-1856424. The authors want to thank all anonymous reviewers for the valuable feedback.

References

[Allen-Zhu and Li, 2019a] Allen-Zhu, Z. and Li, Y. (2019a). Can sgd learn recurrent neural networks with provable generalization? In *Advances in Neural Information Processing Systems*, pages 10331–10341. 1

[Allen-Zhu and Li, 2019b] Allen-Zhu, Z. and Li, Y. (2019b). What can resnet learn efficiently, going beyond kernels? In *Advances in Neural Information Processing Systems*, pages 9017–9028. 1

[Allen-Zhu and Li, 2020] Allen-Zhu, Z. and Li, Y. (2020). Backward feature correction: How deep learning performs deep learning. *arXiv preprint arXiv:2001.04413*. 1

[Allen-Zhu et al., 2019a] Allen-Zhu, Z., Li, Y., and Song, Z. (2019a). A convergence theory for deep learning via over-parameterization. In *International Conference on Machine Learning*, pages 242–252. 1, 3, 7

[Allen-Zhu et al., 2019b] Allen-Zhu, Z., Li, Y., and Song, Z. (2019b). On the convergence rate of training recurrent neural networks. In *Advances in neural information processing systems*, pages 6676–6688. 1

[Arora et al., 2019] Arora, S., Du, S., Hu, W., Li, Z., and Wang, R. (2019). Fine-grained analysis of optimization and generalization for overparameterized two-layer neural networks. In *International Conference on Machine Learning*, pages 322–332. PMLR. 1, 3, 4, 7

[Bengio, 2012] Bengio, Y. (2012). Practical recommendations for gradient-based training of deep architectures. In *Neural networks: Tricks of the trade*, pages 437–478. Springer. 3

[Cao and Gu, 2019] Cao, Y. and Gu, Q. (2019). Generalization bounds of stochastic gradient descent for wide and deep neural networks. In *Advances in Neural Information Processing Systems*, pages 10836–10846. 3

[Cesa-Bianchi et al., 2004] Cesa-Bianchi, N., Conconi, A., and Gentile, C. (2004). On the generalization ability of on-line learning algorithms. *IEEE*

Transactions on Information Theory, 50(9):2050–2057. 3

[Chen et al., 2020] Chen, Z., Cao, Y., Gu, Q., and Zhang, T. (2020). Mean-field analysis of two-layer neural networks: Non-asymptotic rates and generalization bounds. *arXiv preprint arXiv:2002.04026*. 1

[Chizat and Bach, 2018] Chizat, L. and Bach, F. (2018). On the global convergence of gradient descent for over-parameterized models using optimal transport. In *Advances in neural information processing systems*, pages 3036–3046. 1

[Chizat et al., 2019] Chizat, L., Oyallon, E., and Bach, F. (2019). On lazy training in differentiable programming. In *Advances in Neural Information Processing Systems*, pages 2937–2947. 4

[Dehghani et al., 2019] Dehghani, A., Sarbishei, O., Glatard, T., and Shihab, E. (2019). A quantitative comparison of overlapping and non-overlapping sliding windows for human activity recognition using inertial sensors. *Sensors*, 19(22):5026. 2

[Du et al., 2019a] Du, S., Lee, J., Li, H., Wang, L., and Zhai, X. (2019a). Gradient descent finds global minima of deep neural networks. In *International Conference on Machine Learning*, pages 1675–1685. 1, 2, 7

[Du et al., 2018] Du, S. S., Wang, Y., Zhai, X., Balakrishnan, S., Salakhutdinov, R. R., and Singh, A. (2018). How many samples are needed to estimate a convolutional neural network? In *Advances in Neural Information Processing Systems*, pages 373–383. 1

[Du et al., 2019b] Du, S. S., Zhai, X., Poczos, B., and Singh, A. (2019b). Gradient descent provably optimizes over-parameterized neural networks. *ICLR 2019*. 1, 2, 3, 7

[Feigenbaum et al., 2001] Feigenbaum, J., Ishai, Y., Malkin, T., Nissim, K., Strauss, M. J., and Wright, R. N. (2001). Secure multiparty computation of approximations. In *International Colloquium on Automata, Languages, and Programming*, pages 927–938. Springer. 2

[Hu et al., 2019] Hu, W., Li, C. J., Li, L., and Liu, J.-G. (2019). On the diffusion approximation of nonconvex stochastic gradient descent. *Annals of Mathematical Sciences and Applications*, 4(1). 1

[Ikonomovska et al., 2007] Ikonomovska, E., Loskovska, S., and Gjorgjevik, D. (2007). A survey of stream data mining. In *Proceedings of 8th National Conference with International participation, ETAI*, pages 19–21. 1

[Jacot et al., 2018] Jacot, A., Gabriel, F., and Hongler, C. (2018). Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in neural information processing systems*, pages 8571–8580. 1

[Krizhevsky et al., 2012] Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105. 1

[Li et al., 2019] Li, Z., Wang, R., Yu, D., Du, S. S., Hu, W., Salakhutdinov, R., and Arora, S. (2019). Enhanced convolutional neural tangent kernels. *arXiv preprint arXiv:1911.00809*. 1

[Ma et al., 2019] Ma, C., Wu, L., et al. (2019). On the generalization properties of minimum-norm solutions for over-parameterized neural network models. *arXiv preprint arXiv:1912.06987*. 3

[Mei et al., 2019] Mei, S., Misiakiewicz, T., and Montanari, A. (2019). Mean-field theory of two-layers neural networks: dimension-free bounds and kernel limit. In *Conference on Learning Theory*, pages 2388–2464. PMLR. 1

[Mei et al., 2018] Mei, S., Montanari, A., and Nguyen, P.-M. (2018). A mean field view of the landscape of two-layer neural networks. *Proceedings of the National Academy of Sciences*, 115(33):E7665–E7671. 1

[Muthukrishnan, 2005] Muthukrishnan, S. (2005). *Data streams: Algorithms and applications*. Now Publishers Inc. 2

[O'callaghan et al., 2002] O'callaghan, L., Mishra, N., Meyerson, A., Guha, S., and Motwani, R. (2002). Streaming-data algorithms for high-quality clustering. In *Proceedings 18th International Conference on Data Engineering*, pages 685–694. IEEE. 1, 2

[Su and Yang, 2019] Su, L. and Yang, P. (2019). On learning over-parameterized neural networks: A functional approximation perspective. In *Advances in Neural Information Processing Systems*, pages 2641–2650. 1, 2, 3, 4, 5, 7

[Tashman, 2000] Tashman, L. J. (2000). Out-of-sample tests of forecasting accuracy: an analysis and review. *International journal of forecasting*, 16(4):437–450. 2

[Van Der Vaart and Wellner, 2009] Van Der Vaart, A. and Wellner, J. A. (2009). A note on bounds for vc dimensions. *Institute of Mathematical Statistics collections*, 5:103. 8

[Vershynin, 2019] Vershynin, R. (2019). *High-dimensional probability.* Cambridge, UK: Cambridge University Press. 7

[Zou et al., 2020] Zou, D., Cao, Y., Zhou, D., and Gu, Q. (2020). Gradient descent optimizes over-parameterized deep relu networks. *Machine Learning*, 109(3):467–492. 1, 7

[Zou and Gu, 2019] Zou, D. and Gu, Q. (2019). An improved analysis of training over-parameterized deep neural networks. In *Advances in Neural Information Processing Systems*, pages 2055–2064. 3