OPT2020: 12th Annual Workshop on Optimization for Machine Learning 1–12

A Dynamical View on Optimization Algorithms of Overparameterized Neural Networks

Zhiqi Bu ZBU@UPENN.EDU
Shiyun Xu SHIYUNXU@UPENN.EDU
Kan Chen KANCHEN@UPENN.EDU
University of Pennsylvania, USA

Abstract

When equipped with efficient optimization algorithms, the over-parameterized neural networks have demonstrated high level of performance even though the loss function is non-convex and non-smooth. While many works have been focusing on understanding the loss dynamics by training neural networks with the gradient descent (GD), in this work, we consider a broad class of optimization algorithms that are commonly used in practice. For example, we show from a dynamical system perspective that the Heavy Ball (HB) method can converge to global minimum on mean squared error (MSE) at a linear rate (similar to GD); however, the Nesterov accelerated gradient descent (NAG) only converges to global minimum sublinearly.

Our results rely on the connection between neural tangent kernel (NTK) and finite over-parameterized neural networks with ReLU activation, which leads to analyzing the limiting ordinary differential equations (ODE) for optimization algorithms. We show that, optimizing the non-convex loss over the weights corresponds to optimizing some strongly convex loss over the prediction error. As a consequence, we can leverage the classical convex optimization theory to understand the convergence behavior of neural networks. We believe our approach can also be extended to other loss functions and network architectures.

1. Introduction

Neural Tangent Kernel (NTK) Jacot et al. (2018) has taken a huge step in understanding the behaviors of over-parameterized (or wide) and deep neural networks. Leveraging NTK, researchers have focused on the training process of the neural networks and studied how different aspects affect the convergence behavior Lee et al. (2019). In particular, many works have investigated the parameter initialization Du et al., input distribution Du et al. (2018); Allen-Zhu et al. (2019), activation functions Du and Lee (2018), neural network architectures (e.g. two-layer networks Du et al. (2018), multi-layer fully-connected neural networks (FCNN) Du et al.; Allen-Zhu et al. (2019), CNN Arora et al. (2019); Allen-Zhu et al. (2019); Zou and Gu (2019); Zou et al. (2020), ResNet Allen-Zhu et al. (2019)), loss functions Allen-Zhu et al. (2019). On the other hand, one would argue that the efficient optimization of a neural network is as important as designing the architecture of the network. Nevertheless, to our knowledge, this is the first paper to study the convergence behavior of neural networks under the NTK regime, beyond GD and SGD. Especially, we answer the following questions in the affirmative:

- Can other optimizers provably find global minimum on the MSE for neural networks?
- Can we leverage convex optimization theory to explain the convergence behavior (e.g. acceleration) of optimizers on neural networks?

2. Preliminaries

In this section, we introduce the NTK approach to analyze the convergence behavior of any neural networks from a dynamical system perspective. Particularly, we warm ourselves up with some known results of training a two-layer neural network in Du et al. (2018), under the mean squared error (MSE) loss and with the gradient descent (GD).

To start with, we do not specify the neural network architecture (e.g. layers, activation, depth, width, initialization and so on). Given a training set $\{\boldsymbol{x}_i, y_i\}_{i=1}^n$ where $\boldsymbol{x}_i \in \mathbb{R}^p$, we denote $(\boldsymbol{w}_r, \boldsymbol{a})$ with $\boldsymbol{w}_r \in \mathbb{R}^p$ as the weight vectors in the last hidden layer connecting the r-th neuron, $\boldsymbol{W}$ as the union $\{\boldsymbol{w}_r\}$ and $\boldsymbol{a}$ as the set of weights in all the other layers. $f(\boldsymbol{W}, \boldsymbol{a}, \boldsymbol{x}_i)$ is the neural network output. We aim to minimize the MSE loss:

$$L(\boldsymbol{W}, \boldsymbol{a}) = \frac{1}{2}\sum_{i=1}^{n} \left(f(\boldsymbol{W}, \boldsymbol{a}, \boldsymbol{x}_i) - y_i\right)^2$$

Taking the same route as in Du et al. (2018), we focus on optimizing $\boldsymbol{W}$ with $\boldsymbol{a}$ fixed at initialization[1]. Applying the simplest gradient descent with a step size $\eta > 0$, we have

$$\boldsymbol{w}_r(k+1) = \boldsymbol{w}_r(k) - \eta \frac{\partial L(\boldsymbol{W}(k), \boldsymbol{a})}{\partial \boldsymbol{w}_r(k)} \tag{2.1}$$

Since GD is a discretization of its corresponding ordinary differential equation (ODE), known as the gradient flow, we analyze such ODE directly as an equivalent form of GD with an infinitesimal step size. We remark that gradient flows are dynamical systems that are much amenable to analyze and help us understand different optimization algorithms. To be specific, GD corresponds to the following gradient flow,

$$\frac{d\boldsymbol{w}_r(t)}{dt} = -\frac{\partial L(\boldsymbol{W}(t), \boldsymbol{a})}{\partial \boldsymbol{w}_r(t)} \tag{2.2}$$

Simple chain rule gives the following dynamics,

$$\frac{d\boldsymbol{w}_r(t)}{dt} = -\frac{\partial L(\boldsymbol{W}(t), \boldsymbol{a})}{\partial \boldsymbol{f}(t)} \frac{\partial f(t)}{\partial \boldsymbol{w}_r(t)} = -(\boldsymbol{f} - \boldsymbol{y}) \frac{\partial f(t)}{\partial \boldsymbol{w}_r(t)} \tag{2.3}$$

$$\frac{d\boldsymbol{f}(t)}{dt} = \sum_r \frac{\partial \boldsymbol{f}(t)}{\partial \boldsymbol{w}_r(t)} \frac{d\boldsymbol{w}_r(t)}{dt} = -\sum_r \frac{\partial \boldsymbol{f}(t)}{\partial \boldsymbol{w}_r(t)} \left(\frac{\partial \boldsymbol{f}(t)}{\partial \boldsymbol{w}_r(t)}\right)^\top (\boldsymbol{f} - \boldsymbol{y}) \tag{2.4}$$

$$\dot{\Delta}(t) = -\boldsymbol{H}(t)\Delta(t) \tag{2.5}$$

which we call the *weight dynamics*, the *prediction dynamics* and the *error dynamics*, respectively. Here we denote the error of the prediction $\Delta = \boldsymbol{f} - \boldsymbol{y} \in \mathbb{R}^n$ and the $\mathbb{R}^{n\times n}$ NTK matrix as

$$\boldsymbol{H}(t) := \sum_r \frac{\partial \boldsymbol{f}(t)}{\partial \boldsymbol{w}_r(t)} \left(\frac{\partial \boldsymbol{f}(t)}{\partial \boldsymbol{w}_r(t)}\right)^\top \tag{2.6}$$

which is the sum of outer products. The key observation of training the over-parameterized neural networks is that $\boldsymbol{W}(t)$ stays very close to its initialization $\boldsymbol{W}(0)$, even though the loss may change largely. This phenomenon is well-known as 'lazy training'. As a consequence, the neural network f is almost linear in $\boldsymbol{W}$ and the kernel $\boldsymbol{H}(t)$ behaves almost time-independently: $\lim_{m\to\infty} \boldsymbol{H}(0) \approx \boldsymbol{H}(0) \approx \boldsymbol{H}(t)$ (Du et al., 2018, Remark 3.1).

1. in later section we extend our analysis to training all layers simultaneously. We remark that training only the first layer is sufficient to find the global minimum of loss.

Interestingly, suppose we define a pseudo-loss $\hat{L}(t) := \frac{1}{2}\Delta^\top \boldsymbol{H}\Delta$ and notice that $L = \Delta^\top\Delta/2$, then optimizing the *non-convex* loss L over $\boldsymbol{w}_r$ leads to an error dynamics (2.5), as if we were actually optimizing a *strongly-convex* loss $\hat{L}$ over Δ with the same dynamical system as (2.2):

$$\frac{d\Delta(t)}{dt} = -\frac{\partial \hat{L}(\boldsymbol{W}(t), \boldsymbol{a})}{\partial \Delta(t)}$$

The matrix ODE (2.2) with a constant and positive $\boldsymbol{H}$ has a solution converging to 0 at linear rate, as the classical theory on optimizing a strongly convex loss indicates. Therefore it is equivalent to show that $\boldsymbol{H}$ is positive with the smallest eigenvalue bounded away from 0 at all time. We formalize this claim by quoting the results for the two-layer neural network of the following form,

$$f(\boldsymbol{W}, \boldsymbol{a}, \boldsymbol{x}) = \frac{1}{\sqrt{m}} \sum_{r=1}^{m} a_r \sigma\left(\boldsymbol{w}_r^\top \boldsymbol{x}\right)$$

with $\sigma(\cdot)$ being the ReLU activation function. Now we quote an important fact to justify our theorems.

Fact 2.1 (Assumption 3.1 and Theorem 3.1 in Du et al. (2018)) *If for any $i \neq j$, $\mathbf{Z}_i \nparallel \mathbf{Z}_j$, then the least eigenvalue $\lambda_0 := \lambda_{\min}(\mathbf{H}^\infty) > 0$, where matrix $\mathbf{H}^\infty \in \mathbb{R}^{n\times n}$ with*

$$(\mathbf{H})_{ij}^\infty = \mathbb{E}_{\mathbf{w}\sim\mathcal{N}(\mathbf{0},\mathbf{I})}\left[\mathbf{Z}_i^\top \mathbf{Z}_j \mathbb{I}\left\{\mathbf{w}^\top \mathbf{Z}_i \geq 0, \mathbf{w}^\top \mathbf{Z}_j \geq 0\right\}\right]$$

Note that Du et al. (2018) establishes that, for sufficiently wide hidden layer and under some data distributional assumptions, the GD optimizer converges to zero training loss exponentially fast with high probability.

Theorem 1 *Suppose $\forall i, \|\boldsymbol{x}_i\|_2 = 1$ and $|y_i| < C$ for some constant C, and only the hidden layer weights $\{\boldsymbol{w}_r\}$ are optimized. If we set the width $m = \Omega(n^6/\delta^3$ and we i.i.d. initialize $\boldsymbol{w}_r \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), a_r \sim \mathrm{unif}\{-1, 1\}$ for $r \in [m]$, then with high probability at least $1-\delta$ over the initialization, we have*

$$\lambda_{\min}(\boldsymbol{H}(t)) > \frac{1}{2}\lambda_{\min}(\lim_m \boldsymbol{H}(0)) := \frac{1}{2}\lambda_{\min}(\boldsymbol{H}^\infty) := \frac{1}{2}\lambda_0$$

with $\boldsymbol{H}^\infty$ defined in Fact 2.1, and

$$L(t) \leq \exp\left(-\lambda_0 t\right) L(0). \tag{2.7}$$

We note that the NTK matrix is the Gram matrix induced by the ReLU activation:

$$\boldsymbol{H}_{ij}(t) = \sum_{r=1}^{m} \langle \frac{\partial f_i(t)}{\partial \boldsymbol{w}_r}, \frac{\partial f_j(t)}{\partial \boldsymbol{w}_r} \rangle = \frac{1}{m} \boldsymbol{x}_i^\top \boldsymbol{x}_j \sum_{r=1}^{m} \mathbb{I}(\boldsymbol{w}_r^\top \boldsymbol{x}_i \geq 0, \boldsymbol{w}_r^\top \boldsymbol{x}_j \geq 0) \tag{2.8}$$

where $\langle \cdot, \cdot \rangle$ is the inner product with $\langle \boldsymbol{u}, \boldsymbol{v} \rangle = \boldsymbol{u}^\top \boldsymbol{v}$ and its limiting form at the initialization, i.e, $\lim_m \boldsymbol{H}(0)$, has a closed form as follows.

Fact 2.2 (Assumption 3.1 and Theorem 3.1 in Du et al. (2018)) *Define matrix $\mathbf{H}^\infty \in \mathbb{R}^{n\times n}$ with*

$$(\mathbf{H}^\infty)_{ij} = \mathbb{E}_{\mathbf{w_r}\sim N(\mathbf{0},\mathbf{I})}\left[\mathbf{x}_i^\top \mathbf{x}_j \mathbb{I}\left\{\mathbf{w_r}^\top \mathbf{x}_i \geq 0, \mathbf{w_r}^\top \mathbf{x}_j \geq 0\right\}\right] = \left(\frac{1}{2} - \frac{\arccos(\boldsymbol{x}_i^\top \boldsymbol{x}_j)}{2\pi}\right)(\boldsymbol{x}_i^\top \boldsymbol{x}_j)$$

and define $\lambda_0 := \lambda_{\min}(\mathbf{H}^\infty)$. Suppose for any $i \neq j$, $\mathbf{x}_i \nparallel \mathbf{x}_j$, then $\lambda_0 > 0$.

We pause to remark that the framework of Theorem 1 has been extended to training both layers simultaneously Du et al. (2018), analyzing multiple-layer FCNN, CNN, ResNet, training with SGD and other losses including the cross entropy. We now complement this line of research by moving on to exploring the convergence of different optimization algorithms. In this work, we only analyze the continuous flow and we believe our approach can be easily extend to discrete results.

3. Heavy Ball with Friction System

Many paper have adopted the Heavy Ball method in the NTK regime, e.g. this have been empirically observed in Lee et al. (2019). The Heavy Ball method Polyak (1964), by definition, gives

$$\boldsymbol{w}_r(t+1) = \boldsymbol{w}_r(t) - \eta \frac{\partial L(\boldsymbol{w}_r(t), \boldsymbol{a})}{\partial \boldsymbol{w}_r(t)} + \beta(\boldsymbol{w}_r(t) - \boldsymbol{w}_r(t-1)) \tag{3.1}$$

where η is the step size and the momentum term $\beta \in [0, 1]$. The corresponding gradient flow is known as the Heavy Ball with Friction (HBF) system. This is a non-linear dissipative dynamical system, originally proposed by Polyak (1964) and heavily studied in Attouch et al. (2000); Gadat et al. (2018); Cabot et al. (2009); Attouch and Alvarez (2000); Alvarez et al. (2002); Bhaya and Kaszkurewicz (2004); Wilson et al. (2016); Loizou and Richtárik (2020); Liu et al. (2020): with $b > 0$

$$\ddot{\boldsymbol{w}}_r(t) + b\dot{\boldsymbol{w}}_r(t) + \frac{\partial L(\boldsymbol{w}_r(t), \boldsymbol{a})}{\partial \boldsymbol{w}_r(t)} = 0. \tag{3.2}$$

In particular, we study the case as in (Wilson et al., 2016, Equation (7)) and Siegel (2019), when $b = \sqrt{2\lambda_0}$, i.e. twice the strongly convex coefficient:

$$\ddot{\boldsymbol{w}}_r(t) + \sqrt{2\lambda_0}\dot{\boldsymbol{w}}_r(t) + \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{w}_r}(\boldsymbol{f} - \boldsymbol{y}) = 0 \tag{3.3}$$

Our choice of parameter b leads to a linear convergence of training loss to the global minimum without requiring Lipschitz gradients of $\hat{L}$. For other choices of parameters under the Lipschitz condition of $\hat{L}$, HB can enjoy the linear converge locally Polyak (1964); Lessard et al. (2016) and globally Nesterov; Ghadimi et al. (2015); Van Scoy et al. (2017); Siegel (2019); Aujol et al. (2020). To solve a second order ODE requires initial conditions on $\boldsymbol{w}_r$ and $\dot{\boldsymbol{w}}_r$, which we assume as $\dot{\boldsymbol{w}}_r(0) = 0$ without loss of generality. Now we state the our main theorem under MSE loss.

Theorem 2 *Suppose we set the width of the hidden layer* $m = \Omega\left(\frac{n^5}{\delta^2 \lambda_0^{2.5} b^2}\right)$ *and* $b = \sqrt{2\lambda_0}$[2]. *If we i.i.d. initialize* $\boldsymbol{w}_r \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}), a_r \sim \text{unif}\{-1, 1\}$ *for* $r \in [m]$, *then with high probability at least* $1 - \delta$ *over the initialization, we have*

$$L(t) \leq \exp\left(-\sqrt{\lambda_0/2} \cdot t\right) \Delta(0)^\top \boldsymbol{H}(0) \Delta(0) = \exp\left(-\sqrt{\lambda_0/2} \cdot t\right) \hat{L}(0) \tag{3.4}$$

The full proof is in the appendix and we highlight that the analysis is based on the observation $\frac{\partial^2 f_i}{\partial \boldsymbol{w}_r \partial \boldsymbol{w}_l} \overset{\text{a.s.}}{=} 0$. This leads to the error dynamics

$$\ddot{\Delta}(t) + \sqrt{2\lambda_0}\dot{\Delta}(t) + \boldsymbol{H}(t)\Delta(t) = 0 \tag{3.5}$$

which, by the strong convexity of $\hat{L}$, gives the linear convergence via an analysis of Lyapunov function similar to Siegel (2019). We remark that indeed $\sqrt{\lambda_0/2} > \lambda_0$, suggesting a boost in the convergence rate of HB when compared to GD. To see this, we claim that $\lambda_0 < 1/2$ as $\text{tr}(\boldsymbol{H}^\infty) = n/2 = \sum_i \lambda_i > n\lambda_0$. On the other hand, we note that the speedup may come at the cost of non-monotone loss dynamics, as HB is well-known to have oscillating trajectory of losses.

2. discuss this choice is fastest

4. Nesterov Acceleration and General HBF

In this section, we analyze the dynamics of the generalized Nesterov Accelerated Gradient (NAG) descent as follows,

$$\begin{aligned} \boldsymbol{R}(t+1) &= \boldsymbol{w}_r(t) - \eta \frac{\partial L(\boldsymbol{w}_r(t), \boldsymbol{a})}{\partial \boldsymbol{w}_r(t)} \\ \boldsymbol{w}_r(t+1) &= \boldsymbol{R}(t+1) + \frac{t-1}{t+d-1}(\boldsymbol{R}(t+1) - \boldsymbol{R}(t)) \end{aligned} \tag{4.1}$$

where η is step size. Su et al. (2014) gives the corresponding gradient flow as

$$\ddot{\boldsymbol{w}}_r(t) + \frac{d}{t}\dot{\boldsymbol{w}}_r(t) + \frac{\partial L(\boldsymbol{w}_r(t), \boldsymbol{a})}{\partial \boldsymbol{w}_r(t)} = 0 \tag{4.2}$$

with initial conditions $\dot{\boldsymbol{w}}_r(0) = 0$. It follows that the error dynamics

$$\ddot{\Delta}(t) + \frac{d}{t}\dot{\Delta}(t) + \boldsymbol{H}(t)\Delta(t) = 0 = \ddot{\Delta}(t) + \frac{d}{t}\dot{\Delta}(t) + \frac{\partial \hat{L}}{\partial \Delta(t)} \tag{4.3}$$

with the same NTK matrix $\boldsymbol{H}(t)$ as defined in (2.8). Unlike HB and GD, we can only show that NAG converges to global minimum sublinearly, by a similar analysis to Su et al. (2014).

Theorem 3 *Suppose we set the width of the hidden layer* $m = \Omega\left(\frac{n^5}{\delta^2 \lambda_0^{2.5}}\right)$, $4 < \alpha \leq \frac{2d}{3}$ *and* $d \geq 6$. *If we i.i.d. initialize* $\boldsymbol{w}_r \sim \mathcal{N}(\boldsymbol{0}, \mathbf{I}), a_r \sim \mathrm{unif}\{-1, 1\}$ *for* $r \in [m]$, *then with high probability at least* $1 - \delta$ *over the initialization, we have*

$$L(t) \leq A(\alpha, d) t^{-\frac{2d}{3}} \Delta(0)^\top \boldsymbol{H}(0) \Delta(0) = A(\alpha, d) t^{-\frac{2d}{3}} \hat{L}(0) \tag{4.4}$$

where $A(\alpha, d)$ *only depends on* α *and* d.

We pause here to discuss the choice of d in (4.1). In Su et al. (2014), the 'magic constant' d has been extensively studied. When $d \geq 3$, the convergence rate is shown to be $O(t^{-\frac{2d}{3}})$. When $d < 3$, there exist counter-examples that fail the desired $O(1/t^2)$ convergence rate. We remark that NAG may have an improved convergence rate upto $O(1/t^d)$ for $d \geq 3$ (see Su et al. (2014)).

5. Discussion

In this paper we study two most commonly used first-order momentum-based optimizatin algorithms on over-parameterized two-layer FCNN. We show that both optimizers can provably find the global minimum on MSE for over-parameterized neural networks. Our key results are based on the following observation: for *piecewise linear activations*, known as the maxout activations Goodfellow et al. (2013), which include ReLU as a special case, each weight dynamics corresponds to a analogous error dynamics with strongly convex loss. We believe this approach can be easily extended to other optimizers, network architectures and general losses.

Noticeably, since the main focus of this work is on the optimization algorithms, we do not generalize to multi-layer training in deep neural networks. We remark that training deep neural networks under the NTK regime has been well-studied for GD in Allen-Zhu et al. (2019); Du et al. (2018) using the fact that the NTK for deeper model is also positive definite.

References

Zeyuan Allen-Zhu, Yuanzhi Li, and Zhao Song. A convergence theory for deep learning via over-parameterization. In *International Conference on Machine Learning*, pages 242–252. PMLR, 2019.

F Alvarez, H Attouch, J Bolte, and P Redont. A second-order gradient-like dissipative dynamical system with hessian-driven damping.-application to optimization and mechanics. *Journal de mathematiques pures et appliquees*, 8(81):747–779, 2002.

Sanjeev Arora, Simon S Du, Wei Hu, Zhiyuan Li, Russ R Salakhutdinov, and Ruosong Wang. On exact computation with an infinitely wide neural net. In *Advances in Neural Information Processing Systems*, pages 8141–8150, 2019.

Hedy Attouch and Felipe Alvarez. The heavy ball with friction dynamical system for convex constrained minimization problems. In *Optimization*, pages 25–35. Springer, 2000.

Hedy Attouch, Xavier Goudou, and Patrick Redont. The heavy ball with friction method, i. the continuous dynamical system: global exploration of the local minima of a real-valued function by asymptotic analysis of a dissipative dynamical system. *Communications in Contemporary Mathematics*, 2(01):1–34, 2000.

Jean-François Aujol, Charles Dossal, and Aude Rondepierre. Convergence rates of the heavy-ball method for quasi-strongly convex optimization. 2020.

Amit Bhaya and Eugenius Kaszkurewicz. Steepest descent with momentum for quadratic functions is a version of the conjugate gradient method. *Neural Networks*, 17(1):65–71, 2004.

Alexandre Cabot, Hans Engler, and Sébastien Gadat. On the long time behavior of second order differential equations with asymptotically small dissipation. *Transactions of the American Mathematical Society*, 361(11):5983–6017, 2009.

Simon S Du and Jason D Lee. On the power of over-parametrization in neural networks with quadratic activation. *arXiv preprint arXiv:1803.01206*, 2018.

Simon S Du, Jason D Lee, Haochuan Li, Liwei Wang, and Xiyu Zhai. Gradient descent finds global minima of deep neural networks.

Simon S Du, Xiyu Zhai, Barnabas Poczos, and Aarti Singh. Gradient descent provably optimizes over-parameterized neural networks. *arXiv preprint arXiv:1810.02054*, 2018.

Sébastien Gadat, Fabien Panloup, Sofiane Saadane, et al. Stochastic heavy ball. *Electronic Journal of Statistics*, 12(1):461–529, 2018.

Euhanna Ghadimi, Hamid Reza Feyzmahdavian, and Mikael Johansson. Global convergence of the heavy-ball method for convex optimization. In *2015 European control conference (ECC)*, pages 310–315. IEEE, 2015.

Ian Goodfellow, David Warde-Farley, Mehdi Mirza, Aaron Courville, and Yoshua Bengio. Maxout networks. In *International conference on machine learning*, pages 1319–1327. PMLR, 2013.

Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in neural information processing systems*, pages 8571–8580, 2018.

Jaehoon Lee, Lechao Xiao, Samuel Schoenholz, Yasaman Bahri, Roman Novak, Jascha Sohl-Dickstein, and Jeffrey Pennington. Wide neural networks of any depth evolve as linear models under gradient descent. In *Advances in neural information processing systems*, pages 8572–8583, 2019.

Laurent Lessard, Benjamin Recht, and Andrew Packard. Analysis and design of optimization algorithms via integral quadratic constraints. *SIAM Journal on Optimization*, 26(1):57–95, 2016.

Yanli Liu, Yuan Gao, and Wotao Yin. An improved analysis of stochastic gradient descent with momentum. *Advances in Neural Information Processing Systems*, 33, 2020.

Nicolas Loizou and Peter Richtárik. Momentum and stochastic momentum for stochastic gradient, newton, proximal point and subspace descent methods. *Computational Optimization and Applications*, pages 1–58, 2020.

Yu Nesterov. A method of solving a convex programming problem with convergence rate o (1/k2). In *Sov. Math. Dokl*, volume 27.

Boris T Polyak. Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics*, 4(5):1–17, 1964.

Jonathan W Siegel. Accelerated first-order methods: Differential equations and lyapunov functions. *arXiv preprint arXiv:1903.05671*, 2019.

Weijie Su, Stephen Boyd, and Emmanuel Candes. A differential equation for modeling nesterov's accelerated gradient method: Theory and insights. In *Advances in neural information processing systems*, pages 2510–2518, 2014.

Bryan Van Scoy, Randy A Freeman, and Kevin M Lynch. The fastest known globally convergent first-order method for minimizing strongly convex functions. *IEEE Control Systems Letters*, 2 (1):49–54, 2017.

Ashia C Wilson, Benjamin Recht, and Michael I Jordan. A lyapunov analysis of momentum methods in optimization. *arXiv preprint arXiv:1611.02635*, 2016.

Difan Zou and Quanquan Gu. An improved analysis of training over-parameterized deep neural networks. In *Advances in Neural Information Processing Systems*, pages 2055–2064, 2019.

Difan Zou, Yuan Cao, Dongruo Zhou, and Quanquan Gu. Gradient descent optimizes over-parameterized deep relu networks. *Machine Learning*, 109(3):467–492, 2020.

Appendix A. Proof of Theorem 2

First, we use the chain rule to characterize the prediction dynamics of f,

$$\dot{f}(t) = \sum_{r\in[m]} \frac{\partial f}{\partial w_r}\dot{w}_r; \qquad \ddot{f}(t) = \sum_{r,l\in[m]} \frac{\partial^2 f}{\partial w_r \partial w_l}\dot{w}_r\dot{w}_l + \sum_{r\in[m]} \frac{\partial f}{\partial w_r}\ddot{w}_r \overset{\text{a.s.}}{=} \sum_{r\in[m]} \frac{\partial f}{\partial w_r}\ddot{w}_r \tag{A.1}$$

where the last equality follows from a key observation that, with $\delta(\cdot)$ denoting the Dirac Delta function,

$$\begin{aligned} \frac{\partial f_i}{\partial \boldsymbol{w}_r} &= \frac{1}{\sqrt{m}} a_r \boldsymbol{x}_i^\top \mathbb{I}(\boldsymbol{w}_r^\top \boldsymbol{x}_i > 0) \\ \frac{\partial^2 f_i}{\partial \boldsymbol{w}_r \partial \boldsymbol{w}_l} &= 0, \quad \text{for } l \neq r \\ \frac{\partial^2 f_i}{\partial \boldsymbol{w}_r^2} &= \frac{1}{\sqrt{m}} a_r \boldsymbol{x}_i^\top \boldsymbol{x}_i \delta(\boldsymbol{w}_r^\top \boldsymbol{x}_i) \overset{\text{a.s.}}{=} 0 \end{aligned} \tag{A.2}$$

Multiplying $\frac{\partial f}{\partial \boldsymbol{w}_r}$ to (3.2) and sum over r, we obtain the prediction dynamics as

$$\ddot{f}(t) + \sqrt{2\lambda_0}\dot{f}(t) + \boldsymbol{H}(t)(f - \boldsymbol{y}) = 0 \tag{A.3}$$

and consequently, the dynamics of the error is

$$\ddot{\Delta}(t) + \sqrt{2\lambda_0}\dot{\Delta}(t) + \boldsymbol{H}(t)\Delta(t) = 0 \tag{A.4}$$

or in an analogous form to (3.3),

$$\ddot{\Delta}(t) + \sqrt{2\lambda_0}\dot{\Delta}(t) + \frac{\partial \hat{L}}{\partial \Delta(t)} = 0 \tag{A.5}$$

To establish the linear convergence of MSE, i.e, $\Delta(t)^\top \Delta(t)$, we need to guarantee $\boldsymbol{H}(t)$ is positive definite with $\lambda_{\min}(\boldsymbol{H}(t)) \geq \lambda_0/2$. In other words, the pseudo loss $\hat{L}$ is $\frac{\lambda_0}{2}$-strongly convex. We start with $t = 0$, by showing that for wide enough neural networks, $\boldsymbol{H}(0)$ has positive smallest eigenvalue with high probability.

Lemma 4 (Lemma 3.1 in Du et al. (2018)) *If $m = \Omega\left(\frac{n^2}{\lambda_0^2}\log\left(\frac{n^2}{\delta}\right)\right)$, then we have $\|\boldsymbol{H}(0) - \boldsymbol{H}^\infty\|_2 \leq \frac{\lambda_0}{4}$ and $\lambda_{\min}(\boldsymbol{H}(0)) \geq \frac{3}{4}\lambda_0$ with probability of at least $1 - \delta$.*

Next, we introduce a lemma that shows for any t, if $\boldsymbol{w}_r(t)$ is close to $\boldsymbol{w}_r(0)$, then $\boldsymbol{H}(t)$ is close to $\boldsymbol{H}(0)$. Then together with Lemma 4, $\lambda_{\min}(\boldsymbol{H}(t))$ always has a positive smallest eigenvalue.

Lemma 5 (Lemma 3.2 in Du et al. (2018)) *If $\boldsymbol{w}_r$ are i.i.d. generated from $\mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$ for $r \in [m]$, and $\|\boldsymbol{w}_r(0) - \boldsymbol{w}_r\|_2 \leq \frac{c\delta\lambda_0}{n^2} =: R$ for some small positive constant c, then the following holds with probability at least $1 - \delta$ we have $\|\boldsymbol{H}(t) - \boldsymbol{H}(0)\|_2 < \frac{\lambda_0}{4}$ and $\lambda_{\min}(\boldsymbol{H}(t)) > \frac{\lambda_0}{2}$.*

The next lemma gives two important facts given that $\lambda_{\min}(\boldsymbol{H}(s))$ for previous time $s \leq t$: the loss decay exponentially and weights stay close to their initialization at the current time t. We emphasize that Lemma 6 is specific to the choice of optimization algorithms and hence the proof is much different than its analog in (Du et al., 2018, Lemma 3.3) for GD.

Lemma 6 *Assume* $0 \le s \le t$ *and* $\lambda_{\min}(\boldsymbol{H}(s)) \ge \frac{\lambda_0}{2}$. *Then we have* $L(t) \le \exp\left(-\sqrt{\lambda_0/2}t\right)\frac{2\hat{L}(0)}{\lambda_0}$ *and* $\|\boldsymbol{w}_r(t) - \boldsymbol{w}_r(0)\|_2 \le \frac{2}{b}\left(\frac{\sqrt{nC}}{\sqrt{mB}} + e^{-Bt/2}\sqrt{L(0)}\right) =: R'(t)$.

Proof Borrowing the idea of Wilson et al. (2016); Siegel (2019), we define the Lyapunov function or Lyapunov energy as

$$V(t) := \hat{L} + \frac{1}{2}\left\|\sqrt{\frac{\lambda_0}{2}}\Delta(t) + \dot{\Delta}(t)\right\|^2 = \frac{1}{2}\Delta(t)^\top \boldsymbol{H}(t)\Delta(t) + \frac{1}{2}\left\|\sqrt{\frac{\lambda_0}{2}}\Delta(t) + \dot{\Delta}(t)\right\|^2. \tag{A.6}$$

The Lyapunov function represents the total energy of the system and always decreases along the trajectory of the training dynamics since, as we will later show, $\dot{V}(t) < 0$. Here we simplify the notation by denoting the dependence on t in the subscript and use $\alpha := b/2 = \sqrt{\lambda_0/2}$. We bound $\dot{V}(t)$ by the chain rule,

$$\dot{V}(t) = \dot{\Delta}_t^\top \boldsymbol{H}(t)\Delta_t + \frac{1}{2}\Delta_t^\top \dot{\boldsymbol{H}}(t)\Delta_t + \left\langle \alpha\dot{\Delta}_t + \ddot{\Delta}_t, \alpha\Delta_t + \dot{\Delta}_t\right\rangle \tag{A.7}$$

Notice that by (2.8), we have $\dot{\boldsymbol{H}}(t) \overset{\text{a.s.}}{=} 0$. Substituting the error dynamics (A.4) for $\ddot{\Delta}_t$, we have

$$\begin{aligned}\dot{V}(t) &= \left\langle \boldsymbol{H}\Delta_t, \dot{\Delta}_t\right\rangle + \left\langle -\alpha\dot{\Delta}_t - \boldsymbol{H}\Delta_t, \alpha\Delta_t + \dot{\Delta}_t\right\rangle \\ &= -\alpha\left\langle \boldsymbol{H}\Delta_t, \Delta_t\right\rangle - \alpha^2\left\langle \dot{\Delta}_t, \Delta_t\right\rangle - \alpha\left\langle \dot{\Delta}_t, \dot{\Delta}_t\right\rangle\end{aligned}$$

Using $\lambda_{\min}(\boldsymbol{H}) \ge \alpha^2$, we get

$$\left\langle \boldsymbol{H}\Delta_t, \Delta_t\right\rangle \ge \frac{1}{2}\left\langle \boldsymbol{H}\Delta_t, \Delta_t\right\rangle + \frac{\alpha^2}{2}\langle \Delta_t, \Delta_t\rangle = \hat{L}(t) + \frac{\alpha^2}{2}\langle \Delta_t, \Delta_t\rangle$$

and hence we have

$$\begin{aligned}\dot{V}(t) &= -\alpha\hat{L}(t) - \frac{\alpha^3}{2}\langle \Delta_t, \Delta_t\rangle - \alpha^2\left\langle \dot{\Delta}_t, \Delta_t\right\rangle - \alpha\left\langle \dot{\Delta}_t, \dot{\Delta}_t\right\rangle \\ &< -\alpha\left(\hat{L}(t) + \frac{1}{2}\left\|\alpha\Delta_t + \dot{\Delta}_t\right\|^2\right) = -\alpha V(t)\end{aligned}$$

where in the last inequality we throw away $-\frac{\alpha}{2}\left\langle \dot{\Delta}_t, \dot{\Delta}_t\right\rangle$. Clearly $\dot{V}(t) < 0$ for all t. For this first order scalar ODE, we apply the Gronwall's inequality to derive

$$V(t) < e^{-\alpha t}V(0)$$

and we obtain

$$\hat{L}(t) \le V(t) < e^{-\alpha t}V(0) = e^{-\alpha t}\left(\frac{1}{2}\Delta(0)^\top \boldsymbol{H}(0)\Delta(0) + \frac{\alpha^2}{4}\|\Delta(0)\|^2\right).$$

Again using $\lambda_{\min}(\boldsymbol{H}(0)) \ge \alpha$, we have

$$\hat{L}(t) \le \frac{1}{2}\exp\left(-\alpha\cdot t\right)\Delta(0)^\top \boldsymbol{H}(0)\Delta(0) = \exp\left(-\alpha\cdot t\right)\hat{L}(0) \tag{A.8}$$

and

$$L(t) \leq \frac{1}{\alpha^2} \exp\left(-\alpha t\right) \hat{L}(0).$$

In words, $f(t) \to \mathbf{y}$ exponentially fast, with a convergence factor $\alpha = \sqrt{\lambda_0/2}$.

Now we move on to show that $\boldsymbol{w}_r(t)$ stays close to $\boldsymbol{w}_r(0)$. Multiplying $e^{bt} = e^{2\alpha t}$ to the weight dynamics (3.3), we have

$$\frac{d}{dt}\left(e^{2\alpha t}\dot{\boldsymbol{w}}_r\right) = -\frac{1}{\sqrt{m}} e^{2\alpha t} a_r \sum_i (f_i - y_i)\boldsymbol{x}_i \mathbb{I}(\boldsymbol{w}_r^\top \boldsymbol{x}_i \geq 0)$$

which gives a close-form solution

$$\dot{\boldsymbol{w}}_r = -e^{-2\alpha t} \int_0^t \frac{1}{\sqrt{m}} e^{2\alpha s} a_r \sum_i (f_i - y_i)\boldsymbol{x}_i \mathbb{I}(\boldsymbol{w}_r^\top \boldsymbol{x}_i \geq 0) ds$$

whose norm satisfies

$$\begin{aligned}\|\dot{\boldsymbol{w}}_r\| &\leq e^{-2\alpha t}\frac{1}{\sqrt{m}} \int_0^t e^{2\alpha s} \sum_i |f_i(s) - y_i| ds \leq e^{-2\alpha t}\sqrt{\frac{n}{m}} \int_0^t e^{2\alpha s} \|f(s) - \boldsymbol{y}\|_2 ds \\ &= \sqrt{\frac{n}{m}} \int_0^t e^{2\alpha(s-t)} \sqrt{L(s)} ds \leq \sqrt{\frac{n\hat{L}(0)}{m\alpha^2}} \int_0^t e^{\frac{3}{2}\alpha s - 2\alpha t} ds \leq \sqrt{\frac{4n\hat{L}(0)}{9m\alpha^4}} e^{-\frac{\alpha}{2}t}\end{aligned}$$

Finally by Cauchy Schwarz, we bound the weight distance from initialization,

$$\|\boldsymbol{w}_r(t) - \boldsymbol{w}_r(0)\|_2 \leq \int_0^t \|\dot{\boldsymbol{w}}_r(s)\|_2 ds < \sqrt{\frac{n\hat{L}(0)}{9m\alpha^2}}$$

■

We quote the next lemma to show that $R'(t) < R$ indicates that for all $t > 0$, the conditions in Lemma 5 and 6 hold.

Lemma 7 (Lemma 3.4 in Du et al. (2018)) *If $R'(t) < R$ for all $t \geq 0$, then we have $\lambda_{\min}(\boldsymbol{H}(t)) \geq \frac{1}{2}\lambda_0$, for all $r \in [m]$, $\|\boldsymbol{w}_r(t) - \boldsymbol{w}_r(0)\|_2 \leq R'$ and $L(t) \leq \exp\left(-B(b, \lambda_0, \epsilon)t\right) C(\epsilon) L(0)$.*

Finally we study the width requirement for $R' < R$ to hold true, i.e. we need $\sqrt{\frac{n\hat{L}(0)}{9m\alpha^2}} < O(\frac{\delta\lambda_0}{n^2})$ which is equivalent to $m = \Omega(\frac{n^6}{\delta^3\lambda_0^4})$, the same order as GD in Du et al. (2018).

Appendix B. Proof of Theorem 3

Given Lemma 4 and Lemma 5, we will prove Lemma 8 as following:

Lemma 8 *Assume $0 \leq s \leq t$ and $\lambda_{\min}(\boldsymbol{H}(s)) \leq \frac{\lambda_0}{2}$, then we have $L(t) \leq \frac{4C(\alpha, r)}{\lambda_0 t^\alpha (\lambda_0/2)^{\frac{\alpha-2}{2}}} \hat{L}(0)$ for $2 \leq \alpha \leq \frac{2}{3}d$ and $C(\alpha, d)$ only depends on α and d. Furthermore, we have $\|\boldsymbol{w}_r(t) - \boldsymbol{w}_r(0)\|_2 \leq \frac{2}{(\alpha-4)t^\alpha}\sqrt{\frac{16nC(\alpha,d)\hat{L}(0)}{m(\epsilon)\lambda_0(\lambda_0/2)^{\frac{\alpha-2}{2}}(2d-\alpha-2)^2}} := R'(t)$.*

Proof The condition $\lambda_{\min}(\mathbf{H}_s) \geq \frac{\lambda_0}{2}$ gives that $\hat{L}$ is L smooth and μ strongly convex with respect to Δ. We define Lyapunov function

$$V(t;\alpha,d) := t^{\alpha}\hat{L}(t) + \frac{(2d-\alpha)^2 t^{\alpha-2}}{8}\left\|\Delta_t + \frac{2t}{2d-\alpha}\dot{\Delta}_t\right\|^2 \tag{B.1}$$

pseudo loss

$$\hat{L}(t) = \frac{1}{2}\Delta_t^\top H(t)\Delta_t \tag{B.2}$$

and loss function

$$L(t) = \frac{1}{2}\Delta_t^\top \Delta_t. \tag{B.3}$$

For $d \geq 3, 2 \leq \alpha \leq \frac{2d}{3}$, apply Theorem 8 from Su et al. (2014),

$$L(t) \leq \frac{4C(\alpha,d)}{\lambda_0 t^{\alpha}(\lambda_0/2)^{\frac{\alpha-2}{2}}}\hat{L}(0)$$

where $C(\alpha,d)$ only depends on α and r. Since the weight dynamics is

$$\ddot{\boldsymbol{w}}_r(t) + \frac{d}{t}\dot{\boldsymbol{w}}_r(t) + \frac{\partial f}{\partial \boldsymbol{w}_r}(f-\boldsymbol{y}) = 0 \tag{B.4}$$

multiply both side by t^d, then we obtain

$$\frac{d}{dt}(t^d\dot{\boldsymbol{w}}_r(t)) = -t^d\frac{\partial f}{\partial \boldsymbol{w}_r}(f-\boldsymbol{y}) = -\frac{1}{\sqrt{m}}t^d a_r\sum_i (f_i - y_i)\boldsymbol{x}_i\mathbb{I}(\boldsymbol{w}_r^\top \boldsymbol{x}_i \geq 0)$$

which gives a close-form solution

$$\dot{\boldsymbol{w}}_r = -\frac{1}{t^d}\int_0^t \frac{1}{\sqrt{m}}s^d a_r\sum_i (f_i - y_i)\boldsymbol{x}_i\mathbb{I}(\boldsymbol{w}_r^\top \boldsymbol{x}_i \geq 0)ds$$

whose norm satisfies

$$\begin{aligned}
\|\dot{\boldsymbol{w}}_r\| &\leq \frac{1}{t^d\sqrt{m}}\int_0^t s^d\sum_i |f_i(s) - y_i|ds \leq \frac{1}{t^d}\sqrt{\frac{n}{m}}\int_0^t s^d\|f(s)-\boldsymbol{y}\|_2 ds \\
&= \frac{1}{t^d}\sqrt{\frac{n}{m}}\int_0^t s^d\sqrt{L(s)}ds \leq \frac{1}{t^d}\sqrt{\frac{4nC(\alpha,d)\hat{L}(0)}{m\lambda_0(\lambda_0/2)^{\frac{\alpha-2}{2}}}}\int_0^t s^{d-\alpha/2}ds \\
&= t^{1-\alpha/2}\sqrt{\frac{16nC(\alpha,d)\hat{L}(0)}{m\lambda_0(\lambda_0/2)^{\frac{\alpha-2}{2}}(2d-\alpha-2)^2}}
\end{aligned}$$

we bound the weight distance from initialization, when $\alpha > 4$

$$\|\boldsymbol{w}_r(t) - \boldsymbol{w}_r(0)\|_2 \leq \int_0^t \|\dot{\boldsymbol{w}}_r(s)\|_2 ds \leq \frac{2\epsilon^{2-\alpha/2}}{\alpha-4}\sqrt{\frac{16nC(\alpha,d)\hat{L}(0)}{m\lambda_0(\lambda_0/2)^{\frac{\alpha-2}{2}}(2d-\alpha-2)^2}} + o(\epsilon).$$

Let $\epsilon \to 0$, then there exists $m(\epsilon')$ such that

$$\frac{2}{(\alpha-4)t^{\alpha}}\sqrt{\frac{16nC(\alpha,d)\hat{L}(0)}{m(\epsilon)\lambda_0(\lambda_0/2)^{\frac{\alpha-2}{2}}(2d-\alpha-2)^2}} := R'(t) < R$$

■

Similarly, we use Lemma 7 again to show that if $R'(t) < R$ for all $t > 0$, then the conditions in Lemma 4 and Lemma 5 hold.

Lemma 9 (Du et al. (2018)) *If $R'(t) < R$ for all $t \geq 0$, then we have $\lambda_{\min}(\boldsymbol{H}(t)) \geq \frac{1}{2}\lambda_0$, for all $r \in [m]$, $\|w_r(t) - w_r(0)\|_2 \leq R'(t)$ and $L(t) \leq B(\alpha,d)t^{-\frac{2}{3}d}L(0)$ where $4 < \alpha \leq \frac{2}{3}d, d > 6$ and $B(\alpha,d)$ only depends on α and d.*

Finally we study the width requirement for $R'(t) < R$ to hold true, i.e. we need

$$\frac{2}{(\alpha-4)}\sqrt{\frac{16nC(\alpha,d)\hat{L}(0)}{m(\epsilon)\lambda_0(\lambda_0/2)^{\frac{\alpha-2}{2}}(2d-\alpha-2)^2}} < O(\frac{\delta\lambda_0}{n^2})$$

which is equivalent to $m = \Omega\left(\frac{n^5}{\delta^2\lambda_0^{2.5}}\right)$.