Proceedings of Machine Learning Research vol 144:1–12, 2021

The Dynamics of Gradient Descent for Overparametrized Neural Networks

Siddhartha Satpathi SSATPTH2@ILLINOIS.EDU

R Srikant RSRIKANT@ILLINOIS.EDU

Department of Electrical and Computer Engineering, University of Illinois at Urbana Champaign

Abstract

We consider the dynamics of gradient descent (GD) in overparameterized single hidden layer neural networks with a squared loss function. Recently, it has been shown that, under some conditions, the parameter values obtained using GD achieve zero training error and generalize well if the initial conditions are chosen appropriately. Here, through a Lyapunov analysis, we show that the dynamics of neural network weights under GD converge to a point which is close to the minimum norm solution subject to the condition that there is no training error when using the linear approximation to the neural network. To illustrate the application of this result, we show that the GD converges to a prediction function that generalizes well, thereby providing an alternative proof of the generalization results in Arora et al. (2019).

Keywords: Overparametrized Neural Network, Neural Tangent Kernel, Gradient Descent

1. Introduction

Neural Networks have shown promise in many supervised learning tasks like image classification Krizhevsky et al. (2012) and semi-supervised learning like reinforcement learning Mnih et al. (2015). A key reason for the success of neural networks is that they can approximate any continuous function with arbitrary accuracy Hornik (1993); Cybenko (1989). Further choosing the neural network parameters by optimizing a loss function over this complex non-convex function class using simple first order methods, like gradient descent, leads to good generalization for unseen data points. We address the optimization and generalization aspect of neural network training in this paper. Recently it has been shown that when the network is overparametrized, i.e, the number of neurons in a layer is much larger than the number of training points, one can achieve zero training loss by running gradient descent on the squared loss function Du et al. (2019b). This line of research is motivated by the fact when one runs gradient descent on an overparametrized network initialized appropriately, the network behaves as though it is linear function of its weights Jacot et al. (2018); Lee et al. (2019). Although one can achieve zero loss in an overparametrized network, there may be more than one set of network weights which achieve zero loss. In this paper, we study the dynamics of gradient descent for a single hidden layer, overparameterized neural network, and show that gradient descent converges to a certain minimum norm solution. We present an application of such a characterization of the limit behavior of the network weights by providing an alternative proof of a generalization result in Arora et al. (2019).

We are motivated by the results in Du et al. (2019b) where it was shown that if the network width is polynomial in the number of data points, gradient descent converges to the global optimum of the squared loss function. The main idea behind their proof is to show that an overparametrized neural

network behaves similarly to a linear function (linear in the weights). The linear approximation can be described as follows: the input is mapped to a high-dimensional feature vector and the linear approximation is an inner product of the network weights and this high-dimensional feature vector. In the limit as the number of neurons goes to infinity, this mapping of the input to a high-dimensional space can also be viewed in terms of a kernel corresponding to a reproducing kernel Hilbert space (RKHS). Such a kernel is known as the Neural Tangent Kernel (NTK) and was introduced in Jacot et al. (2018). In this paper, we prove that the weights of an overparametrized neural network converge close to the point where the weights of the linearized neural network would have converged had we used the linearized network in the optimization process. To the best of our knowledge, the results in earlier papers do not address such convergence properties.

1.1. Related work

- **Convergence of Gradient Descent for squared loss:** For the squared loss function and ReLU activation function, the convergence of gradient descent was proved in Du et al. (2019b). This is further extended to deep networks in Du et al. (2019a); Allen-Zhu et al. (2018), but the dependence of width grows exponentially with the depth of the network; see Lee et al. (2019); Chizat et al. (2019) for analysis in case of general differentiable activation functions. The dependence of the width of the network in terms of the data points has been further improved in Oymak and Soltanolkotabi (2020) by careful choice of Lyapunov functions.

- **Convergence of Gradient Descent for logistic loss:** Overparametrized neural network with logistic loss function were shown to be in the linear regime for a finite time in Ji and Telgarsky (2019). Under assumptions that the data distribution is linearly separable by the neural tangent kernel, it is shown that gradient descent reaches good test accuracy in this finite time. Unlike the squared loss, optimizing the logistic loss only requires a poly-logarithmic dependence on the number of data points. This is further extended to deep Networks in Chen et al. (2019).

- **Implicit Regularization:** This line of work is closest to our work (see Neyshabur (2017) for an overview). It is known that for least squares linear regression, the iterates of gradient descent converge to the minimum norm solution subject to zero loss. This is also true for certain nonlinear models as shown in Oymak and Soltanolkotabi (2019). We further add to this literature by showing that the iterates of the neural network weights under gradient descent stay close to the minimum norm solution of the linearized model and this distance decreases with the increase in the width of the network. Since gradient descent chooses a particular characterization of weights from all possible solutions achieving zero training error, this phenomenon is often referred to as the 'implicit bias' or 'implicit regularization.' For logistic loss, the weights diverge to infinity, but implicit regularization still occurs. Specifically, with linear classification, the direction along which the weights diverge to infinity matches the direction of the hard margin SVM solution when the data is separable Ji and Telgarsky (2018); Soudry et al. (2018). This result has been extended to some classes of neural networks recently in Chizat and Bach (2020).

- **Generalization results for squared loss:** Shallow Neural Networks are shown to generalize well for a large class of functions in Arora et al. (2019). We provide an alternative proof of their result for the case of the squared loss function. In Ghorbani et al. (2019), th authors consider the generalization properties of Kernel Ridge Regression for NTK under different

activation functions, but their results are not directly applicable to finite-width networks as is the case in this work.

2. Problem Statement and Contribution

Consider a neural network which takes x as its input and produces an output $f(x, w, b)$ where w, b are certain weight parameters to be chosen. We suppose that $f : \mathbb{R}^d \times \mathbb{R}^{md} \times \mathbb{R}^m \to \mathbb{R}$ is of the form

$$f(x, w, b) := \sum_{k=1}^{m} \frac{a_k}{\sqrt{m}} \sigma(w_k^T x + b_k)$$

where $\sigma(.) = \max(0, .)$ is the Rectified Linear Unit (ReLU) activation function and $x \in \mathbb{R}^d$. This describes a neural network with one hidden layer where the input weights are denoted by w's, input biases are b's and output layer weights are a's. We absorb the biases b's as an extra dimension in the weight vector w's. Likewise, let $\tilde{x} = \{x, 1\}$. So we can compactly write the neural network function as $f : \mathbb{R}^d \times \mathbb{R}^{m(d+1)} \to \mathbb{R}$.

$$f(x, w) := \sum_{k=1}^{m} \frac{a_k}{\sqrt{m}} \sigma(w_k^T \tilde{x})$$

We are given n data points $\{x_i, y_i\}_{i=1}^n$ drawn i.i.d. from a distribution $\mathcal{X} \times \mathcal{Y}$ and we would like to minimize the mean-square error

$$L(w) = \sum_{i=1}^{n} (y_i - f(x_i, w))^2$$

over all w. One of the ways to perform the above minimization is to use the Gradient Descent (GD) algorithm. GD is an iterative algorithm where in each step w is updated in the direction of the negative gradient of $L(w)$. Given appropriate initialization of $w(0)$, and step size η for $k = 1, 2, \ldots$

$$w(k+1) = w(k) - \eta \frac{\partial}{\partial w} L(w)\Big|_{w=w(k)}$$

GD is known to converge to a global optimum if $L(w)$ were a convex function of w for small enough values of η. In our case, even if $L(w)$ is non convex, Du et al. (2019b) show that GD initialized appropriately converges to a global optimum, *i.e.*$L(w(k))$ goes to 0 as $k \to \infty$.

We would like to characterize the performance of $w(k)$ given a new data point sampled from the same distribution $\mathcal{X} \times \mathcal{Y}$. In particular, we are interested in the generalization error for $w(k)$ which is defined as

$$E_{x \times y \sim \mathcal{X} \times \mathcal{Y}} (y - f(x, w(k)))^2.$$

We assume that the samples of x are chosen from $\mathcal{X}$. Given x, y is conditionally chosen as $y = f^*(x) + \zeta$, where ζ is drawn from a mean zero distribution independent of $\mathcal{X}$ and f^* is a continuous function to be specified later.

For the noise less case, i.e. $\zeta = 0$, the generalization error has been characterized in Arora et al. (2019) for certain choices of f^*. For the noisy case, to the best of our knowledge the generalization

error has not been characterized. But, Ghorbani et al. (2019) consider the generalization error for the linearized version of the neural network. We first explain this linearization.

Consider the first order Taylor approximation of $f(x, w)$ around $w(0)$. Define

$$f_L(x, w) := f(x, w(0)) + \nabla^\top f(x, w(0))(w - w(0)) = \frac{1}{\sqrt{m}} \sum_{k=0}^{m} a_k \mathbb{1}_{w_k^\top(0)\tilde{x} \geq 0} w_k^\top \tilde{x}.$$

It was shown in Du et al. (2019b) that w stays close to $w(0)$ during the training iterations of gradient descent when then neural network is overparametrized, *i.e.*, $m > poly(n)$. As a result, $f(x_i, w)$ is close to $f_L(x_i, w)$ for $i \in [n]$ during the GD iterations. This makes us consider the following problem. Instead of minimizing $L(w)$, consider the loss function $L_{lin}(w) = \sum_i (y_i - f_L(x_i, w))^2$. Minimizing $L_{lin}(w)$ using GD is expected to achieve zero loss since $L_{lin}(w)$ is a convex function. Since the network is overparametrized, there are inifinitely many values of w achieving zero loss. However, it is known that GD finds the solution which has the lowest ℓ_2 norm. So, minimizing $L_{lin}(w)$ using gradient descent would lead to a solution

$$w_L^* := \arg\min_w \|w - w(0)\| \;\; s.t. \;\; f_L(x_i, w) = y_i, i = \{1, 2, \ldots, n\}. \tag{1}$$

For appropriately chosen $w(0)$, the prediction function $f_L(x, w_L^*)$ can be shown to be close to $\sum_{i=1}^n c_i K^{(m)}(x, x_i)$, for some constants c_i and the kernel function

$$K^{(m)}(x, x_i) = \frac{1}{m} \sum_{k=1}^{m} x^\top x_i \mathbb{1}_{w_k^\top(0)x \geq 0} \mathbb{1}_{w_k^\top(0)x_i \geq 0}.$$

In the infinite width limit, *i.e.*, $m \to \infty$, the kernel $K^{(m)}$ converges to a kernel K which is independent of the initialization $w(0)$ Jacot et al. (2018). If $K(x, y) = \phi^\top(x)\phi(y)$, then by Moore-Aronszajn theorem, the RKHS $\mathcal{H}$ induced by K is given by this: A function $g \in \mathcal{H}$ can be defined by $g(.) = \phi^\top(.)w_g$ and inner product between functions f and g in $\mathcal{H}$ is given by $\langle f, g \rangle_{\mathcal{H}} = w_g^\top w_f$. Further, by Representer theorem, the solution to kernel regression in $\mathcal{H}$ would be given by

$$\begin{aligned} f_{KR}(.) &= \min_{g \in \mathcal{H}} \|g\|_{\mathcal{H}} \text{ s.t. } y_i = g(x_i) \\ &= \sum_{i}^{n} c_i K(., x_i) \quad \text{s.t.} \quad \sum_{i=1}^{n} c_i K(x_j, x_i) = y_j \text{ for } j = \{1, 2, \ldots, n\}. \end{aligned} \tag{2}$$

Hence, in the infinite width limit, the prediction function $f_L(x, w_L^*)$ is close to the solution of kernel regression on the RKHS induced by K. Now we discuss our contributions below.

- The results in Ghorbani et al. (2019) analyse the generalization properties of f_{KR} in the presence of noise. They would apply to our paper in the following manner.

$$E_{x \sim \mathcal{X}, \zeta}(y - f(x, w(k)))^2 \leq 2E_{x \sim \mathcal{X}, \zeta}(y - f_{KR}(x))^2 + 2E_{x \sim \mathcal{X}}(f(x, w(k)) - f_{KR}(x))^2. \tag{3}$$

  If we show that the second term in the RHS of (3) is small, then the generalization results in Ghorbani et al. (2019) applies to our setting. Our goal is to show that the second term is

small and in order to do so we require to show a relation between $w(k)$ and w_L^*. Informally, we show that

$$\|w(k) - w_L^*\| = O\left(\frac{1}{m^{0.125}}\right) \text{ for } m \geq poly(n) \text{ w.h.p for large enough } k. \tag{4}$$

It was observed that in Du et al. (2019b); Arora et al. (2019) that the weights stay in a ball around initialization, i.e., $\|w(k) - w(0)\| = O(poly(n))$. Equation (4) is a more finer characterization of $w(k)$ than Du et al. (2019b).

- In the noiseless case, Arora et al. (2019) provided generalization bounds for the prediction function $f(x, w(k))$. We provide an alternate derivation motivated by the results in Bartlett et al. (2020) in the noiseless case instead of the Rademacher complexity approach in Du et al. (2019a). The results in Bartlett et al. (2020) are derived from the point of view of linear regression. Thus it naturally applies to providing bounds to kernel regression in a RKHS. In the noise-free setting, ($\zeta = 0$) we derive a bound on the first term in (3), i.e, $E_{x\sim\mathcal{X}}(y - f_{KR}(x))^2$.

The structure of the paper is as follows. We establish the result (4) for a continuous-time version of gradient descent in section 3. This develops an intuitive understanding of the proof. In section 4 this is extended to the discrete-time GD. In section 5, we use the results developed on the convergence of weights under GD in section 3-4, to show that $E_{x\sim\mathcal{X}}(f(x, w(k)) - f_{KR}(x))^2$ is small. In addition, $E_{x\sim\mathcal{X}}(y - f(x, w(k)))^2$ is shown to be small for the noise-free case.

2.1. Notation

Let $w(0)$ denote the point at which gradient descent is initialized. The observation vector is $Y = [y_1 \;\; y_2 \ldots y_n]^T$ and the neural network function for the inputs $\{x_i\}_{i=1}^n$ is $f = [f(x_1, w) \ldots f(x_n, w)]^T$. At initialization $f_0 = [f(x_1, w(0)) \ldots f(x_n, w(0))]^T$. Let the ReLU activation function be $\sigma(x) = \max(x, 0)$. The gradient of $f(x, w)$ w.r.t w is $\nabla f(x, w) = [\frac{a_1}{\sqrt{m}}\sigma'(w_1^T\tilde{x})\tilde{x}^T \ldots \frac{a_m}{\sqrt{m}}\sigma'(w_m^T\tilde{x})\tilde{x}^T]^T$. Denote the $m(d+1)\times n$ matrix $\nabla f = [\nabla f(x_1, w) \ldots \nabla f(x_n, w)]$ and the gradient matrix at initialization $\nabla f_0 = [\nabla f(x_1, w(0)) \ldots \nabla f(x_n, w(0))]$. Also $H = \mathbb{E}_{w(0)}\left(\nabla^T f_0 \nabla f_0\right)$. Let the projection matrix onto the column space of ∇f_0 be $P_0 = \nabla f_0(\nabla^T f_0 \nabla f_0)^{-1}\nabla^T f_0$ and the projection matrix onto the null space be $P_0^\perp := I - P_0$. When we write $\|.\|$ it means the 2 norm if . is a vector or the operator norm if . is a matrix.

2.2. Assumptions

- The data points are bounded, i.e., $|y_i| \leq C_y, \|x_i\| = \sqrt{d}, \forall i = 1, \ldots, n$.
- No two data points x_i, x_j are parallel to each other, $x_i \nparallel x_j$.

The assumptions stated above are mild and the same as in the literature Du et al. (2019b). The second assumption essentially ensures that the matrix $H = \mathbb{E}_{w(0)}[\nabla^T f_0 \nabla f_0]$ is positive definite (Du et al., 2019b, Theorem 3.1). We reprove this fact in the Lemma below. Note that H does not depend on the width of the network.

Lemma 1 *Define θ_{min} by $\cos\theta_{\min} := \frac{\max_{i\neq j} x_i^T x_j + 1}{d+1}$. Under the above assumptions, the smallest eigenvalue of $H = \mathbb{E}_{w(0)}[\nabla^T f_0 \nabla f_0]$ is strictly positive, $H \succeq cI, c > 0$ and*

$$\Omega\left(\frac{(d+1)\theta_{\min}}{\sqrt{\log(2n)+1}}\right) = c \leq \lambda_{\min}(H) = O((d+1)\theta_{\min})$$

when $\theta_{\min} < 1$. More complicated expressions for the lower and upper bounds which do not require the condition $\theta_{\min} < 1$ can be found in the proof of this lemma in the appendix.

3. Continuous-Time Gradient Descent Algorithm

First we describe the continuous time gradient descent algorithm below.

- Initialize $w_k(0) \sim \mathcal{N}(0, \kappa^2 I_d)$. a_k's are initialized as 1 with probability $1/2$ and -1 with probability $1/2$.
- Run gradient descent in continuous time, $\dot{w} = \frac{\partial L(w)}{\partial w} := -\nabla f(f - Y)$

3.1. Training loss and bound on $\|w_k - w_k(0)\|$

The training loss goes to zero exponentially fast. This is shown in Theorem 3.2, Du et al. (2019b).

Lemma 2 (Du et al. (2019b)) *The continuous-time gradient descent algorithm achieves zero loss with probability greater than $1-\delta$ (where the randomness is due to the initialization) when $m = \Omega\left(\frac{n^6 d^4 C_y^2}{c^4 \delta^3}\right)$ and $\kappa = 1$. Further, the rate of convergence can be characterized as*

$$\|f(t) - Y\| \leq \exp(-ct/4)\|f_0 - Y\|. \tag{5}$$

Moreover, the weights w_k's remain in a small ball around the initialization $w_k(0)$ in the following sense:

$$\|w_k(t) - w_k(0)\| = O\left(\frac{\sqrt{dn}}{c\sqrt{m}}\right)\|f_0 - Y\|. \tag{6}$$

3.2. Bound on $\|w - w_L^*\|$

We can show that w_L^* from (1) is equal to

$$w_L^* = P_0^{\perp} w(0) + \nabla f_0 (\nabla^T f_0 \nabla f_0)^{-1} Y$$

since $\nabla^T f_0 \nabla f_0$ would be a positive definite matrix. We prove that that the weights $w(t)$ converge to a point close to w_L^* and the distance decreases when the number of neurons m increases.

Theorem 3 *If the number of neurons $m = \Omega\left(\frac{n^6 d^4 C_y^2}{c^4 \delta^3}\right)$, then under assumptions from section 2.2, with probability greater than $1-\delta$ over initialization and $\kappa = 1$*

$$\|w(t) - w_L^*\| \leq \exp(-c/2t)\|w(0) - w_L^*\| + O\left(\frac{(dn)^{2.5} C_y^{1.5}}{c^{2.5}\delta^{1.5} m^{0.25}}\right).$$

Proof Idea: Consider the dynamics of w,

$$\dot{w} = \frac{\partial L(w)}{\partial w} := -\nabla f(f - Y).$$

Since $w(t)$ is close to $w(0)$ from Lemma 2, we can show that $\nabla f \approx \nabla f_0$ throughout the dynamics of w. Hence $\dot{w}$ approximately lies in the column space of ∇f_0. So $P_0^{\perp}\dot{w}$ can be expected to be small. To capture this intuition we choose the Lyapunov functions

$$V_{\perp} := \|P_0^{\perp}(w - w_L^*)\|^2 \text{ and } V_{\|} := \|P_0(w - w_L^*)\|^2. \tag{7}$$

The proof of this theorem is deferred to the Appendix **??**.

4. Discrete-Time Gradient Descent algorithm

In this section we present the discrete-time gradient descent algorithm, and show results similar to the continuous-time version.

- Initialize $w_k(0) \sim \mathcal{N}(0, \kappa^2 I_d)$. a_k's are chosen to be 1 with probability $1/2$ and -1 with probability $1/2$. Choose a step size $\eta > 0$.
- Run gradient descent, i.e., for $k = 0, 1, \ldots$, update the weights as

$$w_{k+1} = w_k - \eta \nabla f(f - Y).$$

4.1. Bound on $\|w - w_L^*\|$ for Gradient Descent

The analysis of the gradient descent in discrete time is similar in spirit to that in continuous time. Hence we relegate the proof of Theorem 4 to the arXiv report. Unlike in continuous time, Theorem 4 requires more neurons for the result to hold, $O\left(\frac{1}{m^{1/8}}\right)$ as opposed to $O\left(\frac{1}{m^{1/4}}\right)$.

Theorem 4 *If the number of neurons* $m = \Omega\left(\frac{(dn)^{10}C_y^6}{c^{10}\delta^6}\right), \kappa = 1$ *and* $\eta = O\left(\frac{c}{(dn)^2}\right)$*, then under assumptions from section 2.2, with probability greater than* $1 - \delta$ *over initialization*

$$\|w(k) - w_L^*\| \leq \left(1 - \frac{c\eta}{2}\right)^{k/2} \|w(0) - w_L^*\| + O\left(\frac{(dnC_y)^{1.5}}{c^{1.5}\delta^{1.5}m^{0.125}}\right)$$

for $k = 0, 1, \ldots,$. *Also,* $w(k)$ *converges to some point* w^* *as* $k \to \infty$.

5. Generalization

In this section we provide generalization results for the output of gradient descent. The analysis depends on the results from Theorem 4 from optimization. Suppose the data points $\{x_i\}_{i=1}^n$ are sampled from a distribution $\mathcal{X}$. In Lemma 5 below we show that the prediction function $f(x, w(k))$ at the end of iteration k of gradient descent is close to the minimum norm interpolator for Kernel Regression from (2). We can show that the solution to (2) is given by $f_{KR}(x)$ given below.

$$f_{KR}(x) = h^T(x)H^{-1}Y,\ h(x) := [K(x, x_i), i = 1 \text{ to } n], \tag{8}$$

$$H_{ij} = K(x_i, x_j) = \tilde{x}_i^T \tilde{x}_j \frac{\pi - \arccos\left(\frac{\tilde{x}_i^T \tilde{x}_j}{d+1}\right)}{2\pi}. \tag{9}$$

Lemma 5 *Suppose at the end of iteration k, the prediction function is $f(x, w(k))$ for a new data point sampled i.i.d from the distribution $\mathcal{X}$. Then*

$$\mathbb{E}_x(f(x, w(k)) - f_{KR}(x))^2 \leq O\left(\frac{1}{\sqrt{n}}\right) + \left(1 - \frac{c\eta}{2}\right)^k (Y^T H^{-1} Y + O(1)) \text{ w.p. } 1 - \delta$$

when $\kappa^2 = O\left(\frac{c\delta}{d^2 n^{1.5}}\right)$, $\eta = O\left(\frac{c}{(dn)^2}\right), m \geq poly(d, n, C_y, 1/c, 1/\delta)$.

Outline of the Proof: We will prove this lemma using various concentration inequalities. The first step is to show that $f(x, w(k))$ is close to its linear approximation $f_L(x, w(k)) = \nabla^\top f(x, w(0))w(k)$ around $w(0)$. The second step is to show that $f_L(x, w(k))$ is close to the linear prediction function at w_L^*, $f_L(x, w_L^*)$. The final step is to show that $f_L(x, w_L^*)$ is close to $f_{KR}(x)$ which close to the limit of $f_L(x, w_L^*)$ as $m \to \infty$.

The characterization in Lemma 5 is essential in showing the generalization result in Theorem 7 below. The kernel $K(x_1, x_2)$ can be expressed as the inner product of infinite-dimensional feature vectors $\phi(x_1)$ and $\phi(x_2)$. Such a feature vector $\phi(.)$ exists because the Kernel is positive definite and symmetric Mohri et al. (2018). Indeed its easy to characterize the feature vector for $K(.,.)$. It is computed in Lemma 6 below.

Lemma 6 *Define a feature vector $\phi(x)$ of data point x as*

$$\phi(x) := [\sqrt{d_0}, \sqrt{d_1}x, \sqrt{d_2}x^{\otimes 2}, \sqrt{d_3}x^{\otimes 3}, \ldots]$$

where $x^{\otimes k}$ denotes the k time Kronecker product of vector x with itself and

$$d_0 = 0.25 + \sum_{p \geq 1} \frac{c_{2p}}{(d+1)^{2p}},\quad d_1 = 0.25 + \sum_{p \geq 1} \frac{2pc_{2p}}{(d+1)^{2p}}$$

$$d_k = \sum_{p \geq \lceil k/2 \rceil} \frac{c_{2p}}{(d+1)^{2p}} \binom{2p}{k} \text{ for } k \geq 2,\quad c_{2p} := \frac{(2p-3)!!(d+1)}{2\pi(2p-2)!!(2p-1)}.$$

Then $K(x, y) = \phi(x)^T \phi(y) = \sum_{k \geq 0} d_k (x^\top y)^k = \tilde{x}_i^T \tilde{x}_j \frac{\pi - \arccos\left(\frac{\tilde{x}_i^T \tilde{x}_j}{d+1}\right)}{2\pi}$ for $\tilde{x} = \{x, 1\}$.

In Theorem 7 below, we show that all functions $y = \phi^T(x)\bar{w}, \|\bar{w}\| < \infty$ which are linear in the feature vector $\phi(x)$ are learnable by a finite width shallow ReLU network. Corollary 8 provides some example functions in this class. This function class can be shown to be learnable from the generalization result in Arora et al. (2019). Their results are presented without using a bias term in each neuron, in which case, the infinite-width NTK RKHS is not a universal approximator. However, it is straightforward to extend their results to allow the addition of a bias term in each neuron, in which case, it is known that the infinite-width NTK RKHS is a universal approximator Ji et al. (2019). In the latter, more general case, their results match the results in this paper. Our contribution is to provide an alternative proof of the result in Arora et al. (2019).

Theorem 7 *Suppose $y = \phi^\top(x)\bar{w}$ and x is sampled from distribution $\mathcal{X}$. Then there exists a constant $C > 0$ such that*

$$E_x(y - f(x, w(k)))^2 \leq C|\bar{w}|^2\sqrt{\frac{d\log(1/\delta)}{n}} \text{ w.p. greater than } 1-\delta \text{ over initialization}$$

when $\kappa^2 = O\left(\frac{c\delta}{d^2n^{1.5}}\right)$, $\eta = O\left(\frac{c}{(dn)^2}\right), m \geq poly(d, n, C_y, 1/c, 1/\delta)$ and the number of gradient descent iterations $k = \Omega\left(\frac{\log\left(n(\|\bar{w}\|^2+1)\right)}{\eta c}\right)$.

Proof Sketch: The argument follows from the proof of the noiseless case in Bartlett et al. (2020). Denote matrix $\Phi = [\phi(x_1), \phi(x_2), \ldots \phi(x_n)]$. Together with Lemma 5 we can show that $\mathbb{E}_x(y - f(x, w))^2 \approx \mathbb{E}_x(y - f_{KR}(x))^2 = \bar{w}^T P_\phi^\perp \mathbb{E}_x\phi(x)\phi^T(x)P_\phi^\perp\bar{w}$ where $P_\phi^\perp$ denotes the projection matrix to the null space of matrix Φ. Observe that the sample covariance matrix $n^{-1}\sum_{i=1}^n \phi(x_i)\phi^T(x_i)$ is orthogonal to the column space of Φ. This reduces the problem of bounding $\mathbb{E}_x(y - f_{KR}(x))^2$ to one of bounding the error between the population and sample covariance of $\phi(x)$. The result then follows from use of McDiarmid's inequality.

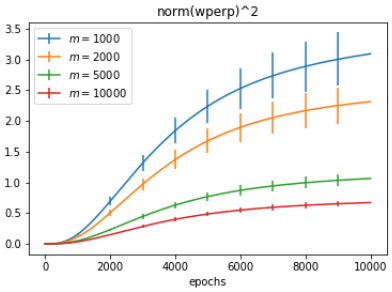
Corollary 8 *If $y = (x^\top\beta)^p = \phi^\top(x)\bar{w}$ for $p = 0, 1, 2, \ldots$ and $\|\beta\| \leq 1$, then*

$$\bar{w} = [0, \ldots, 0, \frac{1}{\sqrt{d_p}}\beta^{\otimes p}, 0, \ldots].$$

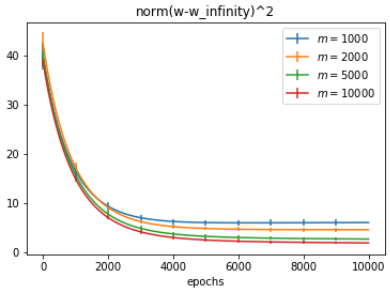
Therefore, by Theorem 7, $\mathbb{E}_x(y - f(x, w(k)))^2 = O\left((d+1)^p(p+1)^{1.5}\sqrt{\frac{\log(1/\delta)}{n}}\right)$ w.p. more than $1-\delta$ for $p \geq 2$ and $\mathbb{E}_x(y - f(x, w(k)))^2 = O\left(\sqrt{\frac{\log(1/\delta)}{n}}\right)$ w.p. more than $1-\delta$ for $p = 0, 1$ when $\kappa^2 = O\left(\frac{c\delta}{d^2n^{1.5}}\right)$, $\eta = O\left(\frac{c}{(dn)^2}\right), m \geq poly(n, d^{p/2}, 1/c, 1/\delta)$ and the number of gradient descent iterations $k = \Omega\left(\frac{\log(n)+p\log(d+1)}{\eta c}\right)$. Using the Stone–Weierstrass Theorem we can show that the set of polynomial functions are dense in the space of continuous functions in the compact input space and hence, the result here can be extended to include all continuous functions with appropriate approximation error.

6. Experiments

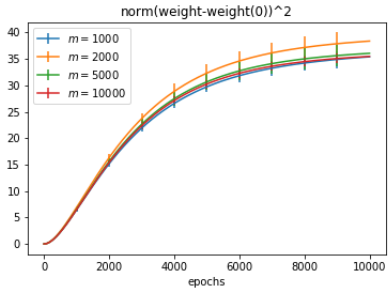
We show an example with a synthetic dataset below. We generate 100 data points from a uniform distribution $[-1, 1]^5$ in $\mathbb{R}^5$ and normalize it to have unit norm. The output is $y = (x^T\beta)^2$, where x is the input and β is chosen from a uniform distribution $[-1, 1]^5$ in $\mathbb{R}^5$. The output points are normalized by subtracting the empirical mean and then dividing by the empirical standard deviation. We fit the data points using a shallow neural network of widths $m = 1000, 2000, 5000, 10000$ and mean squared loss. We perform full gradient descent with a learning rate of 0.01 and do not train the last layer (i.e., the a_i) to be consistent with the theory presented in this paper. The experiment is repeated 5 times with different initialization and the standard deviation is shown in Figure 1 $(b), (c)$. In Figure 1 (b), we plot the different $\|w(t) - w_L^*\|^2$ for different widths across the iterations of gradient descent. Figure 1 (a) shows that $\|P_0^\perp(w(t) - w_L^*)\|^2$ is upper bounded and the upper bound decreases with increase in width of the network. Figure 1 (c) plots distance of the iterates from initialization, $\|w(t) - w(0)\|^2$. We can see that the there is no clear trend on the bound for $\|w(t) - w(0)\|^2$ with increase in m whereas $\|w(t) - w_L^*\|^2$ decreases when m increases.



$(a)\ \|P_0^{\perp}(w - w_L^*)\|^2$



$(b)\ \|w - w_L^*\|^2$



$(c)\ \|w - w(0)\|^2$

Figure 1: Gradient descent iterations for different widths of the network

7. Conclusions

It has been well known for a while that, contrary to traditional wisdom, overparameterized neural networks perform well in training and generalization performance. One intuition behind this is that gradient descent for linear regression performs implicit regularization, i.e., minimizes the network parameter vector among all parameter vectors which provides zero training loss. Motivated by this intuition, recent works have shown that gradient descent, with good initialization, carried out on loss functions of neural networks have good generalization properties. We examine this phenomenon more closely and show that gradient descent with appropriate initialization converges to a point very close to a minimum solution of a linear regression approximation to the neural network training problem. We further use this result to provide generalization guarantees on the prediction function output by gradient descent in the noise free case.

Acknowledgments

Research supported by Navy Grant N00014-19-1-2566, NSF Grants CPS ECCS 1739189, CCF 1934986, NSF/USDA Grant AG 2018-67007-28379 and ARO Grant W911NF-19-1-0379.

References

Zeyuan Allen-Zhu, Yuanzhi Li, and Yingyu Liang. Learning and generalization in overparameterized neural networks, going beyond two layers. *arXiv preprint arXiv:1811.04918*, 2018.

Sanjeev Arora, Simon S Du, Wei Hu, Zhiyuan Li, and Ruosong Wang. Fine-grained analysis of optimization and generalization for overparameterized two-layer neural networks. *arXiv preprint arXiv:1901.08584*, 2019.

Peter L Bartlett, Philip M Long, Gábor Lugosi, and Alexander Tsigler. Benign overfitting in linear regression. *Proceedings of the National Academy of Sciences*, 2020.

Zixiang Chen, Yuan Cao, Difan Zou, and Quanquan Gu. How much over-parameterization is sufficient to learn deep relu networks? *arXiv preprint arXiv:1911.12360*, 2019.

Lenaic Chizat and Francis Bach. Implicit bias of gradient descent for wide two-layer neural networks trained with the logistic loss. *arXiv preprint arXiv:2002.04486*, 2020.

Lenaic Chizat, Edouard Oyallon, and Francis Bach. On lazy training in differentiable programming. In *Advances in Neural Information Processing Systems*, pages 2933–2943, 2019.

George Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314, 1989.

Simon Du, Jason Lee, Haochuan Li, Liwei Wang, and Xiyu Zhai. Gradient descent finds global minima of deep neural networks. In *International Conference on Machine Learning*, pages 1675–1685, 2019a.

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

Behrooz Ghorbani, Song Mei, Theodor Misiakiewicz, and Andrea Montanari. Linearized two-layers neural networks in high dimension. *arXiv preprint arXiv:1904.12191*, 2019.

Kurt Hornik. Some new results on neural network approximation. *Neural networks*, 6(8):1069–1072, 1993.

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.

Ziwei Ji and Matus Telgarsky. Risk and parameter convergence of logistic regression. *arXiv preprint arXiv:1803.07300*, 2018.

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

Ziwei Ji, Matus Telgarsky, and Ruicheng Xian. Neural tangent kernels, transportation mappings, and universal approximation. *arXiv preprint arXiv:1910.06956*, 2019.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012. URL http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf.

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.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of machine learning*. MIT press, 2018.

Behnam Neyshabur. Implicit regularization in deep learning. *arXiv preprint arXiv:1709.01953*, 2017.

Samet Oymak and Mahdi Soltanolkotabi. Overparameterized nonlinear learning: Gradient descent takes the shortest path? In *International Conference on Machine Learning*, pages 4951–4960, 2019.

Samet Oymak and Mahdi Soltanolkotabi. Towards moderate overparameterization: global convergence guarantees for training shallow neural networks. *IEEE Journal on Selected Areas in Information Theory*, 2020.

Daniel Soudry, Elad Hoffer, Mor Shpigel Nacson, Suriya Gunasekar, and Nathan Srebro. The implicit bias of gradient descent on separable data. *The Journal of Machine Learning Research*, 19(1):2822–2878, 2018.