Max-Planck-Institut
für Mathematik
in den Naturwissenschaften
Leipzig

Implicit bias of gradient descent for mean squared error regression with wide neural networks

by

Hui Jin and Guido Montúfar

Preprint no.: 63 2020

Implicit bias of gradient descent for mean squared error regression with wide neural networks

Hui Jin
Department of Mathematics
University of California, Los Angeles
Los Angeles, CA 90095
huijin@ucla.edu

Guido Montúfar
Department of Mathematics and Department of Statistics
University of California, Los Angeles
Los Angeles, CA 90095;
Max Planck Institute for Mathematics in the Sciences
04103 Leipzig, Germany
montufar@math.ucla.edu

Abstract

We investigate gradient descent training of wide neural networks and the corresponding implicit bias in function space. Focusing on 1D regression, we show that the solution of training a width-n shallow ReLU network is within $n^{-1/2}$ of the function which fits the training data and whose difference from initialization has smallest 2-norm of the second derivative weighted by $1/\zeta$. The curvature penalty function $1/\zeta$ is expressed in terms of the probability distribution that is utilized to initialize the network parameters, and we compute it explicitly for various common initialization procedures. For instance, asymmetric initialization with a uniform distribution yields a constant curvature penalty, and thence the solution function is the natural cubic spline interpolation of the training data. The statement generalizes to the training trajectories, which in turn are captured by trajectories of spatially adaptive smoothing splines with decreasing regularization strength.

Keywords. Implicit bias, overparametrized neural network, cubic spline interpolation, spatially adaptive smoothing spline, effective capacity.

1 Introduction

Deep neural networks have achieved tremendous success in many areas. Understanding why neural networks trained in the overparametrized regime and without explicit regularization generalize well in practice is an important problem (Zhang et al., 2017). It has been observed that some form of capacity control different from network size must be at play (Neyshabur et al., 2014). Specifically the implicit bias of parameter optimization has been identified to play a key role in the generalization performance of neural networks (Neyshabur et al., 2017). By implicit bias we mean that among the many hypotheses that fit the training data, the algorithm selects one which satisfies additional properties that may be beneficial for its performance on new data.

The implicit bias of parameter optimization has been investigated in terms of the properties of the loss function at the points reached by different optimization methodologies (Keskar et al., 2017; Wu et al., 2017; Dinh et al., 2017). In terms of the solutions, Maennel et al. (2018) show that gradient flow for shallow ReLU networks initialized close to zero quantizes features in a way that depends on the training data but not on the network size. Soudry et al. (2018) show that in classification problems

with separable data, gradient descent with linear networks converges to a max-margin solution. Gunasekar et al. (2018b) present a result on implicit bias for deep linear convolutional networks, and Ji and Telgarsky (2019) study non-separable data. Chizat and Bach (2020) show that gradient flow for logistic regression with infinitely wide two-layer networks yields a max-margin classifier in a certain space. Gunasekar et al. (2018a) analyze the implicit bias of different optimization methods (natural gradient, steepest and mirror descent) for linear regression and separable linear classification problems, and obtain characterizations in terms of minimum norm or max-margin solutions.

Jacot et al. (2018) and Lee et al. (2019) showed that the training dynamics of shallow and deep wide neural networks is well approximated by that of the linear Taylor approximation of the models at a suitable initialization. Chizat et al. (2019) observe that a model can converge to zero training loss while hardly varying its parameters, a phenomenon that can be attributed to scaling of the output weights and makes the model behave as its linearization around the initialization. Zhang et al. (2019) consider linearized models for regression problems and show that gradient flow finds the global minimum of the loss function which is closest to initialization in parameter space. This type of analysis connects with trajectory based analysis of neural networks (Saxe et al., 2014). Oymak and Soltanolkotabi (2019) studied the overparametrized neural networks directly and showed that gradient descent finds a global minimizer of the loss function which is close to the initialization.

Towards interpreting parameters in function space, Savarese et al. (2019) and Ongie et al. (2020) studied infinite-width neural networks with parameters having bounded norm, in 1D and multi-dimensional input spaces, respectively. They showed that, under a standard parametrization, the complexity of the functions represented by the network, as measured by the 1-norm of the second derivative, can be controlled by the 2-norm of the parameters. Using these results, one can show that gradient descent with ℓ_2 weight penalty leads to simple functions. Sahs et al. (2020) relates function properties, such as breakpoint and slope distributions, to the distributions of the network parameters.

In this work, we study the implicit bias of gradient descent for regression problems. We focus on wide networks with rectified linear units (ReLU) and describe the bias in function space. We present our main results in Section 2, and develop the main theory in Sections 3 and 4. In the interest of a concise presentation, technical proofs and extended discussions are deferred to appendices.

2 Main results and discussion

We obtain a description of the implicit bias in function space when applying gradient descent to regression problems with wide ReLU neural networks. Our strategy and main results are as follows.

1. In Section 3, for a linearized model Theorem 2 shows that gradient descent with sufficiently small step size finds the minimizer of the training objective which is closest to the initial parameter (similar to a result by Zhang et al., 2019). Then Theorem 3 shows that the training dynamics of the linearization of a wide network is well approximated in parameter and function space by that of a lower dimensional linear model which trains only the output weights.
2. In Section 4, for networks with a single input and a single layer of ReLUs, we relate the implicit bias of gradient descent in parameter space to an alternative optimization problem. In Theorem 5 we show that the solution of this problem has a well defined limit as the width of the network tends to infinity, which allows us to obtain a variational formulation.
3. In Theorem 6 we translate the description of the bias from parameter space to function space. This is expressed as the minimization of the 2-norm of second derivative weighted by a function that depends on the distribution that is utilized to initialize the parameters, subject to fitting the training data. In Theorem 9 we provide explicit descriptions of the weight function for various common initialization procedures.

Finally, we can utilize recent results bounding the difference in function space of the solutions obtained from training a wide network and its linearization (Lee et al., 2019, Theorem H.1). We prove the following result (proof in Appendix D).

Theorem 1 (Implicit bias of gradient descent in wide ReLU networks)**.** *Consider a feedforward network with a single input unit, a hidden layer of n rectified linear units, and a single linear output unit. Assume standard parametrization* (2) *and that for each hidden unit the input weight and bias are initialized from a sub-Gaussian* $(\mathcal{W}, \mathcal{B})$ (3) *with joint density $p_{\mathcal{W},\mathcal{B}}$. Then, for any finite data set*

$\{(x_i, y_i)\}_{i=1}^M$ and sufficiently large n there exist constant u and v so that optimization of the mean square error on the adjusted training data $\{(x_i, y_i - ux_i - v)\}_{i=1}^M$ by full-batch gradient descent with sufficiently small step size converges to a parameter θ^* for which $f(x, \theta^*)$ attains zero training error. Furthermore, letting $\zeta(x) = \int_{\mathbb{R}} |W|^3 p_{\mathcal{W},\mathcal{B}}(W, -Wx)\, \mathrm{d}W$ and $S = \operatorname{supp}(\zeta) \cap [\min_i x_i, \max_i x_i]$, we have $\|f(x, \theta^*) - g^*(x)\|_2 = O(n^{-\frac{1}{2}}), x \in S$ (the 2-norm over S) with high probability over the random initialization θ_0, where g^* solves following variational problem:

$$\begin{aligned} \min_{g \in C^2(S)} \quad & \int_S \frac{1}{\zeta(x)} (g''(x) - f''(x, \theta_0))^2 \, \mathrm{d}x \\ \text{subject to} \quad & g(x_i) = y_i - ux_i - v, \quad i = 1, \ldots, M. \end{aligned} \tag{1}$$

Here, the curvature penalty function $1/\zeta(x)$ emphasizes different parts of the input space depending on the density $p_{\mathcal{W},\mathcal{B}}$ from which weights and biases are initialized.

Interpretation An intuitive interpretation of the theorem is that at those regions of the input space where ζ is smaller, we can expect the difference between the functions after and before training to have a small curvature. We may call $\rho = 1/\zeta$ a curvature penalty function. The bias induced from initialization is expressed explicitly. We note that under suitable asymmetric parameter initialization (see Appendix C.2), it is possible to achieve $f(\cdot, \theta_0) \equiv 0$. Then the regularization is on the curvature of the output function itself. Moreover, if we substitute constraints $g(x_i) = y_i$ by a quadratic term $\frac{1}{\lambda}\frac{1}{M}\sum_{i=1}^M (g(x_i) - y_i)^2$ added to the objective, we obtain the variational problem for a so-called spatially adaptive smoothing spline (see Abramovich and Steinberg, 1996; Pintore et al., 2006). This problem can be solved explicitly and can be shown to approximate early stopping. This allows us to describe the optimization trajectory in function space (see Appendix M). In Theorem 9 we obtain the explicit form of ζ for various common parameter initialization procedures. In particular, when the parameters are initialized independently from a uniform distribution on a finite interval, ζ is constant and the problem is solved by the natural cubic spline interpolation of the data. The adjustment of the training data simply accounts for the fact that second derivatives define a function only up to linear terms. In practice we can use the coefficients a and b of linear regression $y_i = ax_i + b + \epsilon_i$, $i = 1, \ldots, M$, and set the adjusted data as $\{(x_i, \epsilon_i)\}_{i=1}^M$.

We illustrate Theorem 1 numerically in Figure 1 and more extensively in Appendix A. In close agreement with the theory, the solution to the variational problem captures the solution of gradient descent training uniformly with error of order $n^{-1/2}$. To illustrate the effect of the curvature penalty function, Figure 1 also shows the solutions to the variational problem for different values of ζ corresponding to different initialization distributions. We see that at input points where ζ is small / peaks strongly, the solution function tends to have a lower curvature / be able to use a higher curvature in order to fit the data.

Relation to previous and concurrent works Zhang et al. (2019) described the implicit bias of gradient descent in the kernel regime as minimizing a kernel norm from initialization, subject to fitting the training data. Our result can be regarded as making the kernel norm explicit, thus providing an interpretable description of the bias in function space and further illuminating the role of the parameter initialization procedure. We prove the equivalence in Appendix L. Savarese et al. (2019) showed that infinite-width networks with 2-norm weight regularization represent functions with smallest 1-norm of the second derivative, an example of which are linear splines. We further discuss this in Appendix C.4. A recent preprint further develops this direction, for two-layer networks with certain activation functions that interpolate data while minimizing a weight norm (Parhi and Nowak, 2019). In contrast, our result characterizes the solutions of training from a given initialization without explicit regularization, which turn out to minimize a weighted 2-norm of the second derivative and hence correspond to cubic splines. In finishing this work we became aware of a recent preprint (Heiss et al., 2019) which discusses ridge weight penalty, adaptive splines, and early stopping for one-input ReLU networks training only the output layer. These results are closely related to ours, although they do not connect to training the full network.

Consequences and possible generalizations The key innovation from this work is the interpretable description of the implicit bias of gradient descent in function space for regression problems. One of the technical challenges that we encountered was the reformulation of the bias in parameter space into a suitable optimization problem which has a continuous limit and proving convergence

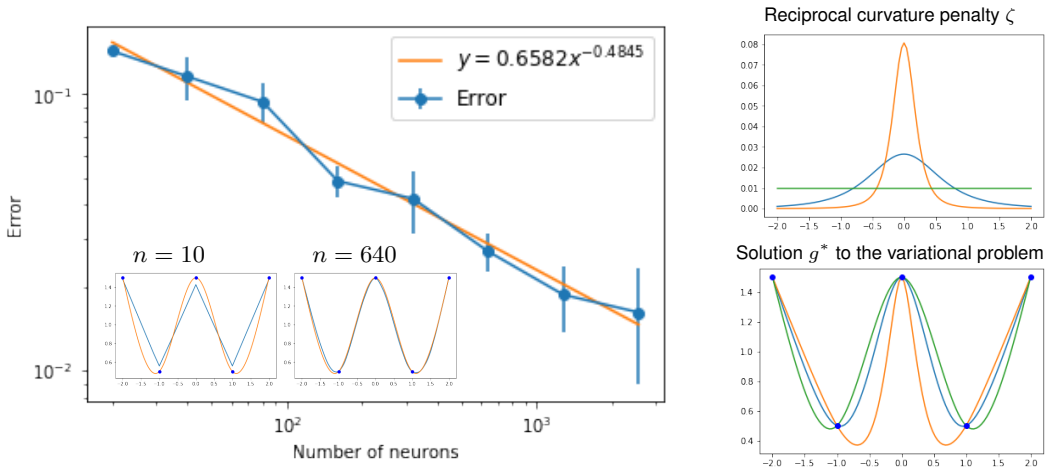

Figure 1: Illustration of Theorem 1. Left: Uniform error between the solution g^* to the variational problem and the functions $f(\cdot, \theta^*)$ obtained by gradient descent training of a neural network (in this case with uniform initialization $W \sim U(-1, 1)$, $B \sim U(-2, 2)$), against the number of neurons. The inset shows examples of the trained networks (blue) alongside with the training data (dots) and the solution to the variational problem (orange). Right: Effect of the curvature penalty function on the shape of the solution function. The bottom shows g^* for various different ζ shown at the top. Again dots are training data. The green curve is for ζ constant on $[-2, 2]$, derived from initialization $\mathcal{W} \sim U(-1, 1)$, $\mathcal{B} \sim U(-2, 2)$; blue is for $\zeta(x) = 1/(1 + x^2)^2$, derived from $\mathcal{W} \sim N(0, 1)$, $\mathcal{B} \sim N(0, 1)$; and orange for $\zeta(x) = 1/(0.1 + x^2)^2$, derived from $\mathcal{W} \sim N(0, 1)$, $\mathcal{B} \sim N(0, 0.1)$. Theorem 9 shows how to compute ζ for the above distributions.

of the corresponding solutions. With the presented bias description we can formulate heuristics for parameter initialization either to ease optimization or also to induce specific smoothness priors on the solutions. In particular, by Proposition 8 any curvature penalty $1/\zeta$ can be implemented by an appropriate choice of the parameter initialization distribution. By our analysis, the effective capacity of the model, understood as the set of possible output functions after training, is adapted to the size M of the training dataset and is well captured by a space of cubic splines relative to the initial function. This is a space with dimension of order M independently of the number of parameters of the network. Several generalizations are interesting to consider in more detail, including multi-dimensional inputs, other network architectures and activations, and other loss functions. We comment on these in Appendix O.

3 Wide networks and parameter space

3.1 Notation and problem setup

Consider a feedforward network fully connected between subsequent layers with n_0 inputs, L hidden layers of widths $n_1, \ldots, n_L$, and k outputs. Given an input $x \equiv x^{(0)} \in \mathbb{R}^{n_0}$, the pre-activation value $h^{(l)}$ and post-activation value $x^{(l)}$ at layer l are given by:

$$h^{(l)} = W^{(l)} x^{(l-1)} + b^{(l)}, \quad x^{(l)} = \phi(h^{(l)}), \tag{2}$$

where ϕ is a point-wise activation function, $W^{(l)} \in \mathbb{R}^{n_l \times n_{l-1}}$ and $b^{(l)} \in \mathbb{R}^{n_l}$ are the weights and biases of layer l. For any given input x, the output of the network is $f(x, \theta) = h^{L+1}(x) \in \mathbb{R}^k$. We write $\theta = \mathrm{vec}(\cup_{l=1}^{L+1} \{W^{(l)}, b^{(l)}\})$ for the vector of all network parameters. These parameters are initialized by independent samples of pre-specified random variables $\mathcal{W}$ and $\mathcal{B}$ in the following way:

$$W_{i,j}^{(l)} \overset{d}{=} \sqrt{\frac{1}{n_{l-1}}} \mathcal{W}, \quad b_j^{(l)} \overset{d}{=} \sqrt{\frac{1}{n_{l-1}}} \mathcal{B}. \tag{3}$$

More generally, we will also allow weight-bias pairs to be sampled from a joint distribution of $(\mathcal{W}, \mathcal{B})$ which we only assume to be sub-Gaussian. In the analysis of Jacot et al. (2018); Lee et al. (2019),

$\mathcal{W}$ and $\mathcal{B}$ are Gaussian $\mathcal{N}(0, \sigma^2)$. In the default initialization of PyTorch, $\mathcal{W}$ and $\mathcal{B}$ have uniform distribution $\mathcal{U}(-\sigma, \sigma)$. The setting (2) is known as the standard parametrization. Some works (Jacot et al., 2018; Lee et al., 2019) utilize the so-called NTK parametrization, where the factor $\sqrt{1/n_{l-1}}$ is carried outside of the trainable parameter. If we fix the learning rate for all parameters, gradient descent leads to different trajectories under these two parametrizations. Our results are presented for the standard parametrization. Details on this in Appendix C.3.

We consider a regression problem for data $\{(x_i, y_i)\}_{i=1}^M$ with inputs $\mathcal{X} = \{x_i\}_{i=1}^M$ and outputs $\mathcal{Y} = \{y_i\}_{i=1}^M$. For a loss function $\ell \colon \mathbb{R}^k \times \mathbb{R}^k \to \mathbb{R}$, the empirical risk of our function is $L(\theta) = \sum_{i=1}^M \ell(f(x_i, \theta), y_i)$. We use full batch gradient descent with a fixed learning rate η to minimize $L(\theta)$. Writing θ_t for the parameter at time t, and θ_0 for the initialization, this defines an iteration

$$\theta_{t+1} = \theta_t - \eta \nabla L(\theta) = \theta_t - \eta \nabla_\theta f(\mathcal{X}, \theta_t)^T \nabla_{f(\mathcal{X}, \theta_t)} L, \tag{4}$$

where $f(\mathcal{X}, \theta_t) = [f(x_1, \theta_t), \ldots, f(x_M, \theta_t)]^T$ is the vector of network outputs for all training inputs, and $\nabla_{f(\mathcal{X}, \theta_t)} L$ is the gradient of the loss with respect to the model outputs. We will use subscript i to index neurons and subscript t to index time.

3.2 Implicit bias in parameter space for a linearized model

In this section we describe how training a linearized network or a wide network by gradient descent leads to solutions that are biased, having parameter values close to the values at initialization. First, we consider the linearized model. This is obtained by the first order Taylor expansion of the network function with respect to the parameter, at the initial parameter value,

$$f^{\text{lin}}(x, \omega) = f(x, \theta_0) + \nabla_\theta f(x, \theta_0)(\omega - \theta_0). \tag{5}$$

We write ω for the parameter of the linearized model, in order to distinguish it from the parameter of the nonlinearized model. The empirical loss of the linearized model is defined by $L^{\text{lin}}(\omega) = \sum_{i=1}^M \ell(f^{\text{lin}}(x_i, \omega), y_i)$. The gradient descent iteration for the linearized model is given by

$$\omega_0 = \theta_0, \quad \omega_{t+1} = \omega_t - \eta \nabla_\theta f(\mathcal{X}, \theta_0)^T \nabla_{f^{\text{lin}}(\mathcal{X}, \omega_t)} L^{\text{lin}}. \tag{6}$$

Next, we consider wide neural networks. Assume that the widths of the hidden layers are identical, i.e. $n_1 = n_2 = \cdots = n_L = n$. According to Lee et al. (2019, Theorem H.1),

$$\sup_t \|f^{\text{lin}}(x, \omega_t) - f(x, \theta_t)\|_2 = O(n^{-\frac{1}{2}})$$

with arbitrarily high probability. So gradient descent training of a wide network or of the linearized model give similar trajectories and solutions in function space. Both fit the training data perfectly, meaning $f^{\text{lin}}(\mathcal{X}, \omega_\infty) = f(\mathcal{X}, \theta_\infty) = \mathcal{Y}$, and are also approximately equal outside the training data.

Let $\hat{\Theta}_n$ be the empirical neural tangent kernel (NTK) of the standard parametrization at time 0, which is the matrix $\hat{\Theta}_n = \frac{1}{n} \nabla_\theta f(\mathcal{X}, \theta_0) \nabla_\theta f(\mathcal{X}, \theta_0)^T$. According to Yang (2019), if the parameters are initialized by a Gaussian, the empirical NTK converges in probability to an analytic NTK then defined by $\Theta := \lim_{n \to \infty} \hat{\Theta}_n$ in probability. Let $\lambda_{\max}(\hat{\Theta}_n)$ be the maximum eigenvalue of $\hat{\Theta}_n$. Now we consider the implicit bias of training the linearized model using (6). Zhang et al. (2019) show that gradient flow converges to the solution with zero empirical loss which is closest to the initial weights. We show a similar result for the case of gradient descent with small enough learning rate.

Theorem 2 (Bias of the linearized model in parameter space)**.** *Consider a convex loss function ℓ with a unique finite minimum and which is K-Lipschitz continuous, i.e. $|\ell(\widehat{y}_1, y) - \ell(\widehat{y}_2, y)| \leq K|\widehat{y}_1 - \widehat{y}_2|$. If $\operatorname{rank}(\nabla_\theta f(\mathcal{X}, \theta_0)) = M$, then the gradient descent iteration* (6) *with learning rate $\eta \leq \frac{1}{Kn\sqrt{M}\lambda_{\max}(\hat{\Theta}_n)}$ converges to the unique solution of following constrained optimization problem:*

$$\min_\omega \|\omega - \theta_0\|_2 \quad \textit{s.t. } f^{\text{lin}}(\mathcal{X}, \omega) = \mathcal{Y}. \tag{7}$$

The proof and further remarks are provided in Appendix E. Note that this statement is valid for the linearization of any set of functions, not only neural networks.

3.3 Training only the output layer approximates training all parameters

In the following we consider networks with a single hidden layer of n ReLUs and a linear output:

$$f(x,\theta) = \sum_{i=1}^{n} W_i^{(2)}[W_i^{(1)}x + b_i^{(1)}]_+ + b^{(2)}. \tag{8}$$

We show that the functions and parameter vectors obtained by training the linearized model are close to those obtained by training only the output layer. Hence, by the arguments of the previous section, training all parameters of a wide network or training only the output layer gives similar functions.

Let $\theta_0 = \mathrm{vec}(\overline{W}^{(1)}, \overline{b}^{(1)}, \overline{W}^{(2)}, \overline{b}^{(2)})$ be the parameter at initialization so that $f^{\mathrm{lin}}(\cdot,\theta_0) = f(\cdot,\theta_0)$. After training the linearized network let the parameter be $\omega_\infty = \mathrm{vec}(\widehat{W}^{(1)}, \widehat{b}^{(1)}, \widehat{W}^{(2)}, \widehat{b}^{(2)})$. Using initialization (3), with probability arbitrarily close to 1, $\overline{W}_i^{(1)}, \overline{b}_i^{(1)} = O(1)$ and $\overline{W}_i^{(2)}, \overline{b}^{(2)} = O(n^{-\frac{1}{2}})$.[1] Therefore, writing H for the Heaviside function, we have

$$\begin{aligned}
\nabla_{W_i^{(1)}, b_i^{(1)}} f(x,\theta_0) &= \Big[\overline{W}_i^{(2)} H(\overline{W}_i^{(1)}x + \overline{b}^{(1)}) \cdot x \ ,\ \overline{W}_i^{(2)} H(\overline{W}_i^{(1)}x + \overline{b}_i^{(1)})\Big] = O(n^{-\frac{1}{2}}),\\
\nabla_{W_i^{(2)}, b^{(2)}} f(x,\theta_0) &= \Big[[\overline{W}_i^{(1)}x + \overline{b}_i^{(1)}]_+ \ ,\ 1\Big] = O(1).
\end{aligned} \tag{9}$$

So when n is large, if we use gradient descent with a constant learning rate for all parameters, then the changes of $W^{(1)}$, $b^{(1)}$, $b^{(2)}$ are negligible compared with the changes of $W^{(2)}$. So approximately we can train just the output weights, $W_i^{(2)}, i = 1,\dots,n$, and fix all other parameters. This corresponds to a smaller linear model. Let $\widetilde{\omega}_t = \mathrm{vec}(\overline{W}_t^{(1)}, \overline{b}_t^{(1)}, \widetilde{W}_t^{(2)}, \overline{b}_t^{(2)})$ be the parameter at time t under the update rule where $\overline{W}^{(1)}, \overline{b}^{(1)}, \overline{b}^{(2)}$ are kept fixed at their initial values, and

$$\widetilde{W}_0^{(2)} = \overline{W}^{(2)}, \quad \widetilde{W}_{t+1}^{(2)} = \widetilde{W}_t^{(2)} - \eta \nabla_{W^{(2)}} L^{\mathrm{lin}}(\widetilde{\omega}_t). \tag{10}$$

Let $\widetilde{\omega}_\infty = \lim_{t\to\infty} \widetilde{\omega}_t$. By the above discussion, we can expect that $f^{\mathrm{lin}}(x, \widetilde{\omega}_\infty)$ will be close to $f^{\mathrm{lin}}(x, \omega_\infty)$. In fact, we prove the following in the case of the MSE loss.

Theorem 3 (Training only output weights vs linearized network)**.** *Consider a finite data set $\{(x_i, y_i)\}_{i=1}^M$. Assume that (1) we use the MSE loss $\ell(\widehat{y}, y) = \frac{1}{2}\|\widehat{y} - y\|_2^2$; (2) $\liminf_n \lambda_{\min}(\hat{\Theta}_n) > 0$. Let ω_t denote the parameters of the linearized model at time t when we train all parameters using* (6)*, and let $\widetilde{\omega}_t$ denote the parameters at time t when we only train weights of the output layer using* (10)*. If we use the same learning rate η in these two training processes and $\eta < \frac{2}{n\lambda_{\max}(\hat{\Theta}_n)}$, then for any $x \in \mathbb{R}$, with probability arbitrarily close to 1 over the random initialization* (3)*,*

$$\sup_t |f^{\mathrm{lin}}(x, \widetilde{\omega}_t) - f^{\mathrm{lin}}(x, \omega_t)| = O(n^{-1}), \text{ as } n \to \infty. \tag{11}$$

Moreover, in terms of the parameter trajectories we have

$$\sup_t \|\overline{W}_t^{(1)} - \widehat{W}_t^{(1)}\|_2 = O(n^{-1}), \quad \sup_t \|\overline{b}_t^{(1)} - \widehat{b}_t^{(1)}\|_2 = O(n^{-1}), \tag{12}$$

$$\sup_t \|\widetilde{W}_t^{(2)} - \widehat{W}_t^{(2)}\|_2 = O(n^{-3/2}), \quad \sup_t \|\overline{b}_t^{(2)} - \widehat{b}_t^{(2)}\| = O(n^{-1}). \tag{13}$$

The proof and further remarks are provided in Appendix F. By combining Theorem 3 and the fact that training a linearized model approximates training a wide network (Lee et al., 2019, Theorem H.1), we obtain the following.

Corollary 4 (Training only output weights vs training all weights)**.** *Consider the settings of Theorem 3, and assume that the joint distribution of $(\mathcal{W}, \mathcal{B})$ is sub-Gaussian. Then $\sup_t \|f^{\mathrm{lin}}(x, \tilde{\omega}_t) - f(x, \theta_t)\|_2 = O(n^{-\frac{1}{2}})$ with arbitrarily high probability over the random initialization* (3)*.*

The proof is provided in Appendix G. In view of Corollary 4, in the next sections we will focus on training only the output weights and understanding the corresponding solution functions.

[1] More precisely, for any $\delta > 0$, $\exists C$, s.t. with prob. $1-\delta$, $|\overline{W}_i^{(2)}|, |\overline{b}^{(2)}| \le Cn^{-1/2}$ and $|\overline{W}_i^{(1)}|, |\overline{b}_i^{(1)}| \le C$.

4 Gradient descent leads to simple functions

In this section we provide a function space characterization of the implicit bias previously described in parameter space. By Theorem 2, gradient descent training of the output weights (10) achieves zero loss, $f^{\text{lin}}(x_j, \widetilde{\omega}_\infty) - f^{\text{lin}}(x_j, \theta_0) = \sum_{i=1}^n (\widetilde{W}_i^{(2)} - \overline{W}_i^{(2)})[\overline{W}_i^{(1)} x_j + \overline{b}_i]_+ = y_j - f(x_j, \theta_0)$, $j = 1, \dots, M$, with minimum $\|\widetilde{W}^{(2)} - \overline{W}^{(2)}\|_2^2$. Hence gradient descent is actually solving

$$\min_{W^{(2)}} \|W^{(2)} - \overline{W}^{(2)}\|_2^2 \quad \text{s.t.} \quad \sum_{i=1}^n (W_i^{(2)} - \overline{W}_i^{(2)})[W_i^{(1)} x_j + b_i]_+ = y_j - f(x_j, \theta_0),\ j = 1, \dots, M. \tag{14}$$

To simplify the presentation, in the following we let $f^{\text{lin}}(x, \theta_0) \equiv 0$ by using the ASI trick (see Appendix C.2). The analysis still goes through without this.

4.1 Infinite width limit

We reformulate problem (14) in a way that allows us to consider the limit of infinitely wide networks, with $n \to \infty$, and obtain a deterministic counterpart, analogous to the convergence of the NTK. Let μ_n denote the empirical distribution of the samples $(W_i^{(1)}, b_i)_{i=1}^n$, so that $\mu_n(A) = \frac{1}{n} \sum_{i=1}^n \mathbb{1}_A \left((W_i^{(1)}, b_i) \right)$. Here $\mathbb{1}_A$ is the indicator function for measurable subsets A in $\mathbb{R}^2$. We further consider a function $\alpha_n \colon \mathbb{R}^2 \to \mathbb{R}$ whose value encodes the difference of the output weight from its initialization for a hidden unit with input weight and bias given by the argument, $\alpha_n(W_i^{(1)}, b_i) = n(W_i^{(2)} - \overline{W}_i^{(2)})$. Then (14) with ASI can be rewritten as

$$\begin{aligned} \min_{\alpha_n \in C(\mathbb{R}^2)} \quad & \int_{\mathbb{R}^2} \alpha_n^2(W^{(1)}, b)\, \mathrm{d}\mu_n(W^{(1)}, b) \\ \text{subject to} \quad & \int_{\mathbb{R}^2} \alpha_n(W^{(1)}, b)[W^{(1)} x_j + b]_+ \, \mathrm{d}\mu_n(W^{(1)}, b) = y_j, \quad j = 1, \dots, M. \end{aligned} \tag{15}$$

Here we minimize over functions α_n in $C(\mathbb{R}^2)$, but since only the values on $(W_i^{(1)}, b_i)_{i=1}^n$ are taken into account, we can take any continuous interpolation of $\alpha_n(W_i^{(1)}, b_i)$, $i = 1, \dots, n$. Now we can consider the infinite width limit. Let μ be the probability measure of $(\mathcal{W}, \mathcal{B})$. We obtain a continuous version of problem (15) by substituting μ for μ_n. By the Glivenko-Cantelli Theorem in two dimensions (Lo et al., 2016), we know that μ_n weakly converges to μ. We prove that in fact the solution of problem (15) converges to the solution of the continuous problem. Let

$$g_n(x, \alpha_n) = \int_{\mathbb{R}^2} \alpha_n(W^{(1)}, b)[W^{(1)} x + b]_+ \, \mathrm{d}\mu_n(W^{(1)}, b)$$

be the function represented by a network with n hidden neurons after training, and let $g(x, \alpha) = \int_{\mathbb{R}^2} \alpha(W^{(1)}, b)[W^{(1)} x + b]_+ \, \mathrm{d}\mu(W^{(1)}, b)$ be the function represented by the infinite-width network. We prove the following theorem. Details in Appendix H.

Theorem 5. *Let $(W_i^{(1)}, b_i)_{i=1}^n$ be i.i.d. samples from a pair $(\mathcal{W}, \mathcal{B})$ of random variables with finite fourth moment. Suppose μ_n is the empirical distribution of $(W_i^{(1)}, b_i)_{i=1}^n$ and $\overline{\alpha}_n(W^{(1)}, b)$ is the solution of* (15)*. Let $\overline{\alpha}(W^{(1)}, b)$ be the solution of the continuous problem with μ in place of μ_n. Then for any bounded $[-L, L]$, $\sup_{x \in [-L, L]} |g_n(x, \overline{\alpha}_n) - g(x, \overline{\alpha})| = O(n^{-1/2})$ with high probability.*

4.2 Function space description of the implicit bias

Next we connect the problem from the previous section to second derivatives by first rewriting it in terms of breakpoints. Consider the breakpoint $c = -b/W^{(1)}$ of a ReLU with weight $W^{(1)}$ and bias b. We define a corresponding random variable $\mathcal{C} = -\mathcal{B}/\mathcal{W}$ and let ν denote the distribution of $(\mathcal{W}, \mathcal{C})$.[2]

[2]Here we assume that $\mathbb{P}(\mathcal{W} = 0) = 0$ so that the random variable $\mathcal{C}$ is well defined. It is not an important restriction, since neurons with weight $W^{(1)} = 0$ give constant functions that can be absorbed in the bias of output layer.

Then with $\gamma(W^{(1)}, c) = \alpha(W^{(1)}, -cW^{(1)})$ the continuous version of (15) is equivalently given as

$$\begin{aligned} \min_{\gamma \in C(\mathbb{R}^2)} \quad & \int_{\mathbb{R}^2} \gamma^2(W^{(1)}, c) \, \mathrm{d}\nu(W^{(1)}, c) \\ \text{subject to} \quad & \int_{\mathbb{R}^2} \gamma(W^{(1)}, c)[W^{(1)}(x_j - c)]_+ \, \mathrm{d}\nu(W^{(1)}, c) = y_j, \quad j = 1, \dots, M. \end{aligned} \tag{16}$$

Let $\nu_{\mathcal{C}}$ denote the distribution of $\mathcal{C} = -\mathcal{B}/\mathcal{W}$, and $\nu_{\mathcal{W}|\mathcal{C}=c}$ the conditional distribution of $\mathcal{W}$ given $\mathcal{C} = c$. Suppose $\nu_{\mathcal{C}}$ has support $\mathrm{supp}(\nu_{\mathcal{C}})$ and a density function $p_{\mathcal{C}}(c)$. Let $g(x, \gamma) = \int_{\mathbb{R}^2} \gamma(W^{(1)}, c)[W^{(1)}(x - c)]_+ \, \mathrm{d}\nu(W^{(1)}, c)$, which again corresponds to the output function of the network. Then, the second derivative g'' with respect to x (see Appendix I) satisfies

$$g''(x, \gamma) = p_{\mathcal{C}}(x) \int_{\mathbb{R}} \gamma(W^{(1)}, x) \big| W^{(1)} \big| \, \mathrm{d}\nu_{\mathcal{W}|\mathcal{C}=x}(W^{(1)}). \tag{17}$$

Thus $\gamma(W^{(1)}, c)$ is closely related to $g''(x, \gamma)$ and we can try to express (16) in terms of $g''(x, \gamma)$. Since $g''(x, \gamma)$ determines $g(x, \gamma)$ only up to linear functions, we consider the following problem:

$$\begin{aligned} \min_{\gamma \in C(\mathbb{R}^2), u \in \mathbb{R}, v \in \mathbb{R}} \quad & \int_{\mathbb{R}^2} \gamma^2(W^{(1)}, c) \, \mathrm{d}\nu(W^{(1)}, c) \\ \text{subject to} \quad & u x_j + v + \int_{\mathbb{R}^2} \gamma(W^{(1)}, c)[W^{(1)}(x_j - c)]_+ \, \mathrm{d}\nu(W^{(1)}, c) = y_j, \quad j = 1, \dots, M. \end{aligned} \tag{18}$$

Here u, v are not included in the cost. They add a linear function to the output of the neural network. If u and v in the solution of (18) are small, then the solution is close to the solution of (16). Ongie et al. (2020) also use this trick to simplify the characterization of neural networks in function space. Next we study the solution of (18) in function space. This is our main technical result.

Theorem 6 (Implicit bias in function space)**.** *Assume $\mathcal{W}$ and $\mathcal{B}$ are random variables with $\mathbb{P}(\mathcal{W} = 0) = 0$, and let $\mathcal{C} = -\mathcal{B}/\mathcal{W}$. Let ν denote the probability distribution of $(\mathcal{W}, \mathcal{C})$. Suppose $(\overline{\gamma}, \overline{u}, \overline{v})$ is the solution of* (18)*, and consider the corresponding output function*

$$g(x, (\overline{\gamma}, \overline{u}, \overline{v})) = \overline{u}x + \overline{v} + \int_{\mathbb{R}^2} \overline{\gamma}(W^{(1)}, c)[W^{(1)}(x - c)]_+ \, \mathrm{d}\nu(W^{(1)}, c). \tag{19}$$

Let $\nu_{\mathcal{C}}$ denote the marginal distribution of $\mathcal{C}$ and assume it has a density function $p_{\mathcal{C}}$. Let $\mathbb{E}(\mathcal{W}^2|\mathcal{C})$ denote the conditional expectation of $\mathcal{W}^2$ given $\mathcal{C}$. Consider the function

$$\zeta(x) = p_{\mathcal{C}}(x)\mathbb{E}(\mathcal{W}^2|\mathcal{C} = x). \tag{20}$$

Assume that training data $x_i \in \mathrm{supp}(\zeta)$, $i = 1, \dots, m$. Consider the set $S = \mathrm{supp}(\zeta) \cap [\min_i x_i, \max_i x_i]$. Then $g(x, (\overline{\gamma}, \overline{u}, \overline{v}))$ satisfies $g''(x, (\overline{\gamma}, \overline{u}, \overline{v})) = 0$ for $x \notin S$ and for $x \in S$ it is the solution of the following problem:

$$\min_{h \in C^2(S)} \int_S \frac{(h''(x))^2}{\zeta(x)} \, \mathrm{d}x \quad \textit{s.t.} \quad h(x_j) = y_j, \quad j = 1, \dots, m. \tag{21}$$

The proof is provided in Appendix I, where we also present the corresponding statement without ASI. Theorem 6 formulates the implicit bias in terms of an optimization problem in function space which depends on the function ζ. We study the explicit form of this function in the next section.

4.3 Explicit form of the curvature penalty function

Proposition 7. *Let $p_{\mathcal{W},\mathcal{B}}$ denote the joint density function of $(\mathcal{W}, \mathcal{B})$ and let $\mathcal{C} = -\mathcal{B}/\mathcal{W}$ so that $p_{\mathcal{C}}$ is the breakpoint density. Then $\zeta(x) = \mathbb{E}(W^2|C = x)p_{\mathcal{C}}(x) = \int_{\mathbb{R}} |W|^3 p_{\mathcal{W},\mathcal{B}}(W, -Wx) \, \mathrm{d}W$.*

The proof is presented in Appendix J. If we allow the initial weight and biases to be sampled from a suitable joint distribution, we can make the curvature penalty $\rho = 1/\zeta$ arbitrary.

Proposition 8 (Constructing any curvature penalty)**.** *Given any function $\varrho \colon \mathbb{R} \to \mathbb{R}_{>0}$, satisfying $Z = \int_{\mathbb{R}} \frac{1}{\varrho} < \infty$, if we set the density of $\mathcal{C}$ as $p_{\mathcal{C}}(x) = \frac{1}{Z}\frac{1}{\varrho(x)}$ and make $\mathcal{W}$ independent of $\mathcal{C}$ with non-vanishing second moment, then $(\mathbb{E}(W^2|C = x)p_{\mathcal{C}}(x))^{-1} = (\mathbb{E}(W^2)p_{\mathcal{C}}(x))^{-1} \propto \varrho(x)$, $x \in \mathbb{R}$.*

Further remarks on sampling and independent variables are provided in Appendix J. To conclude this section we compute the explicit form of ζ for several common initialization procedures.

Theorem 9 (Explicit form of the curvature penalty for common initializations)**.**

(a) Gaussian initialization. Assume that $\mathcal{W}$ and $\mathcal{B}$ are independent, $\mathcal{W} \sim \mathcal{N}(0, \sigma_w^2)$ and $\mathcal{B} \sim \mathcal{N}(0, \sigma_b^2)$. Then ζ is given by $\zeta(x) = \frac{2\sigma_w^3 \sigma_b^3}{\pi(\sigma_b^2 + x^2\sigma_w^2)^2}$.

(b) Binary-uniform initialization. Assume that $\mathcal{W}$ and $\mathcal{B}$ are independent, $\mathcal{W} \in \{-1, 1\}$ and $\mathcal{B} \sim \mathcal{U}(-a_b, a_b)$ with $a_b \geq L$. Then ζ is constant on $[-L, L]$.

(c) Uniform initialization. Assume that $\mathcal{W}$ and $\mathcal{B}$ are independent, $\mathcal{W} \sim \mathcal{U}(-a_w, a_w)$ and $\mathcal{B} \sim \mathcal{U}(-a_b, a_b)$ with $\frac{a_b}{a_w} \geq L$. Then ζ is constant on $[-L, L]$.

The proof is provided in Appendix K. Theorem 9 (b) and (c) show that for certain distributions of $(\mathcal{W}, \mathcal{B})$, ζ is constant. In this case problem (21) is solved by the cubic spline interpolation of the data with natural boundary conditions (Ahlberg et al., 1967). The case of general ζ is solved by space adaptive natural cubic splines, which can be computed numerically by solving a linear system and theoretically in an RKHS formalism. We provide details in Appendix N.

Acknowledgments

This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement n° 757983).

References

Felix Abramovich and David M. Steinberg. Improved inference in nonparametric regression using L_k-smoothing splines. *Journal of Statistical Planning and Inference*, 49(3):327 – 341, 1996. ISSN 0378-3758. doi: https://doi.org/10.1016/0378-3758(95)00021-6. URL `http://www.sciencedirect.com/science/article/pii/0378375895000216`.

J. H. Ahlberg, Edwin N. Nilson, and J. L. Walsh. *The Theory of Splines and Their Applications*. ISSN. Elsevier Science, 1967. ISBN 9780080955452. URL `https://books.google.com/books?id=S7d1pjJHsRgC`.

Zeyuan Allen-Zhu, Yuanzhi Li, and Zhao Song. A convergence theory for deep learning via over-parameterization. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 242–252, Long Beach, California, USA, 09–15 Jun 2019. PMLR. URL `http://proceedings.mlr.press/v97/allen-zhu19a.html`.

Christopher Bishop. Regularization and complexity control in feed-forward networks. In *Proceedings International Conference on Artificial Neural Networks ICANN'95*, volume 1, pages 141–148. EC2 et Cie, January 1995. URL `https://www.microsoft.com/en-us/research/publication/regularization-and-complexity-control-in-feed-forward-networks/`.

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.

Laurent Dinh, Razvan Pascanu, Samy Bengio, and Yoshua Bengio. Sharp minima can generalize for deep nets. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1019–1028, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR. URL `http://proceedings.mlr.press/v70/dinh17b.html`.

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

G German. Smoothing and non-parametric regression. *International Journal of Systems Science*, 2001.

Suriya Gunasekar, Jason Lee, Daniel Soudry, and Nathan Srebro. Characterizing implicit bias in terms of optimization geometry. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 1832–1841, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018a. PMLR. URL http://proceedings.mlr.press/v80/gunasekar18a.html.

Suriya Gunasekar, Jason D Lee, Daniel Soudry, and Nati Srebro. Implicit bias of gradient descent on linear convolutional networks. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 9461–9471. Curran Associates, Inc., 2018b. URL http://papers.nips.cc/paper/8156-implicit-bias-of-gradient-descent-on-linear-convolutional-networks.pdf.

Boris Hanin and David Rolnick. Complexity of linear regions in deep networks. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2596–2604, Long Beach, California, USA, 09–15 Jun 2019. PMLR. URL http://proceedings.mlr.press/v97/hanin19a.html.

Jakob Heiss, Josef Teichmann, and Hanna Wutte. How implicit regularization of neural networks affects the learned function - part i. *arXiv preprint arXiv:1911.02903*, 2019.

Arthur Jacot, Franck Gabriel, and Clement Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 8571–8580. Curran Associates, Inc., 2018.

Ziwei Ji and Matus Telgarsky. The implicit bias of gradient descent on nonseparable data. In Alina Beygelzimer and Daniel Hsu, editors, *Proceedings of the Thirty-Second Conference on Learning Theory*, volume 99 of *Proceedings of Machine Learning Research*, pages 1772–1798, Phoenix, USA, 25–28 Jun 2019. PMLR. URL http://proceedings.mlr.press/v99/ji19a.html.

Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On large-batch training for deep learning: Generalization gap and sharp minima. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/pdf?id=H1oyRlYgg.

Jaehoon Lee, Jascha Sohl-dickstein, Jeffrey Pennington, Roman Novak, Sam Schoenholz, and Yasaman Bahri. Deep neural networks as Gaussian processes. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=B1EA-M-0Z.

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 H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8572–8583. Curran Associates, Inc., 2019.

Gane Samb Lo, Modou Ngom, and Tchilabalo Atozou Kpanzou. Weak convergence (IA). Sequences of random vectors. *arXiv preprint arXiv:1610.05415*, 2016.

Hartmut Maennel, Olivier Bousquet, and Sylvain Gelly. Gradient descent quantizes ReLU network features. *arXiv preprint arXiv:1803.08367*, 2018.

C. Nasim. The solution of an integral equation. *Proceedings of the American Mathematical Society*, 40(1):95–101, 1973. ISSN 00029939, 10886826. URL http://www.jstor.org/stable/2038642.

Radford M Neal. Priors for infinite networks. In *Bayesian Learning for Neural Networks*, pages 29–53. Springer, 1996.

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

Behnam Neyshabur, Ryota Tomioka, Ruslan Salakhutdinov, and Nathan Srebro. Geometry of optimization and implicit regularization in deep learning. *arXiv preprint arXiv:1705.03071*, 2017.

Greg Ongie, Rebecca Willett, Daniel Soudry, and Nathan Srebro. A function space view of bounded norm infinite width ReLU nets: The multivariate case. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=H1lNPxHKDH.

Samet Oymak and Mahdi Soltanolkotabi. Overparameterized nonlinear learning: Gradient descent takes the shortest path? In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 4951–4960, Long Beach, California, USA, 09–15 Jun 2019. PMLR. URL http://proceedings.mlr.press/v97/oymak19a.html.

Rahul Parhi and Robert D. Nowak. Minimum "norm" neural networks are splines. *arXiv preprint arXiv:1910.02333*, 2019.

Alexandre Pintore, Paul Speckman, and Chris C. Holmes. Spatially adaptive smoothing splines. *Biometrika*, 93(1):113–125, 03 2006. ISSN 0006-3444. doi: 10.1093/biomet/93.1.113. URL https://doi.org/10.1093/biomet/93.1.113.

Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred Hamprecht, Yoshua Bengio, and Aaron Courville. On the spectral bias of neural networks. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 5301–5310, Long Beach, California, USA, 09–15 Jun 2019. PMLR. URL http://proceedings.mlr.press/v97/rahaman19a.html.

Justin Sahs, Aneel Damaraju, Ryan Pyle, Onur Tavaslioglu, Josue Ortega Caro, Hao Yang Lu, and Ankit Patel. A functional characterization of randomly initialized gradient descent in deep ReLU networks, 2020. URL https://openreview.net/forum?id=BJl9PRVKDS.

Pedro Savarese, Itay Evron, Daniel Soudry, and Nathan Srebro. How do infinite width bounded norm networks look in function space? In Alina Beygelzimer and Daniel Hsu, editors, *Proceedings of the Thirty-Second Conference on Learning Theory*, volume 99 of *Proceedings of Machine Learning Research*, pages 2667–2690, Phoenix, USA, 25–28 Jun 2019. PMLR. URL http://proceedings.mlr.press/v99/savarese19a.html.

Andrew M. Saxe, James L. McClelland, and Surya Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. In Yoshua Bengio and Yann LeCun, editors, *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014. URL http://arxiv.org/abs/1312.6120.

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.

Roman Vershynin. Four lectures on probabilistic methods for data science. In M.W. Mahoney, J.C. Duchi, and A.C. Gilbert, editors, *The Mathematics of Data*, IAS/Park City Mathematics Series, pages 231–271. American Mathematical Society, 2018. ISBN 9781470435752. URL https://books.google.de/books?id=7HJ6DwAAQBAJ.

Lei Wu, Zhanxing Zhu, and E Weinan. Towards understanding generalization of deep learning: Perspective of loss landscapes. *arXiv preprint arXiv:1706.10239*, 2017.

Greg Yang. Scaling limits of wide neural networks with weight sharing: Gaussian process behavior, gradient independence, and neural tangent kernel derivation. *arXiv preprint arXiv:1902.04760*, 2019.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. In *International Conference on Learning Representations, ICLR 2017*, 2017. URL https://arxiv.org/abs/1611.03530.

Yaoyu Zhang, Zhi-Qin John Xu, Tao Luo, and Zheng Ma. A type of generalization error induced by initialization in deep neural networks. *arXiv preprint arXiv:1905.07777*, 2019.

Appendices

The appendices are organized as follows. In Appendix A we illustrate our theoretical results numerically, and in Appendix B we provide details on the numerical implementation.

In Appendix C we briefly comment on definitions and settings around the parametrization and initialization of neural networks, as well as on the limiting NTK and the linearization of a neural network. In Appendices D E, F, G, H, I, J, K we provide the proofs of the formal results from the main part.

In Appendix L we show the equivalence between our variational characterization of the implicit bias of gradient descent in function space and the description in terms of a kernel norm minimization problem. We provide an interpretable description of the kernel norm.

In Appendix M we discuss the relation between the gradient descent optimization trajectory and a trajectory of spatially adaptive smoothing splines with decreasing smoothness regularization coefficient which converges to the spatially adaptive interpolating spline.

In Appendix N we give the explicit form of the solution to our variational problem, i.e. the spatially adaptive interpolating spline, which corresponds to the output function upon gradient descent training in the infinite width limit.

In Appendix O we comment on some of the possible extensions and generalizations of the analysis. In particular, we give a generalization of Theorem 1 to the case of multi-dimensional inputs, and a formulation for neural networks with activation function different from ReLU.

A Numerical illustration of the theoretical results

Gradient descent training and variational problem To illustrate Theorem 1 across different initialization procedures, in Figures A1 and A2 we show analogous experiments to those in the left panel of Figure 1, but using two types of Gaussian initialization instead of the uniform initialization. As we already observed in the right panel of Figure 1, here the effect of the curvature penalty function is also visible. In portions of the input space where ζ is peaked, the solution function can have a high curvature, and, conversely, in portions of the input space where ζ takes small values, the solution function has a small second derivative and is more linear.

To verify that the results are stable over different data sets, in Figure A3 we show an experiment similar to that of Figure 1, but for a larger data set.

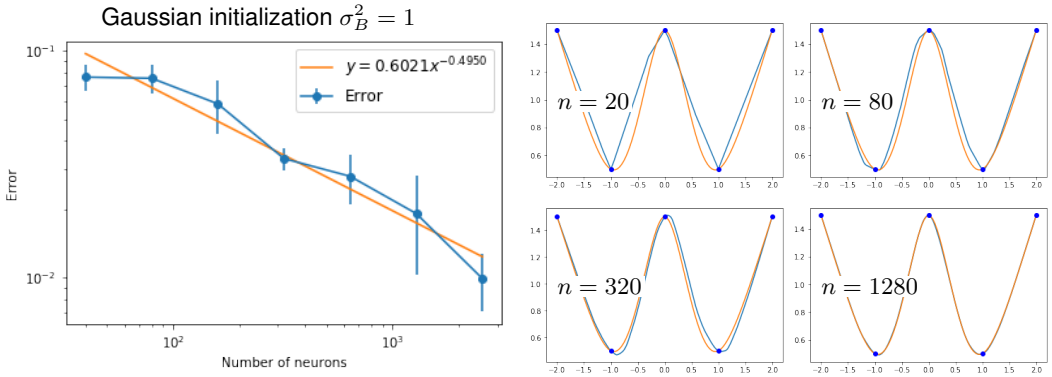

Figure A1: Illustration of Theorem 1. Shown is the error between the output function $f(\cdot, \theta^*)$ of the trained neural network and the solution g^* to the variational problem (21) against the number of neurons, n. Shown is the average over 5 repetitions, with error bars indicating the standard deviation. Here the training data is fixed, and the parameters were initialized with $W \sim N(0, 1)$ and $B \sim N(0, 1)$. The right panel shows the data (dots), trained network functions (blue) with 20, 80, 320, 1280 neurons, and the solution (orange) to the variational problem.

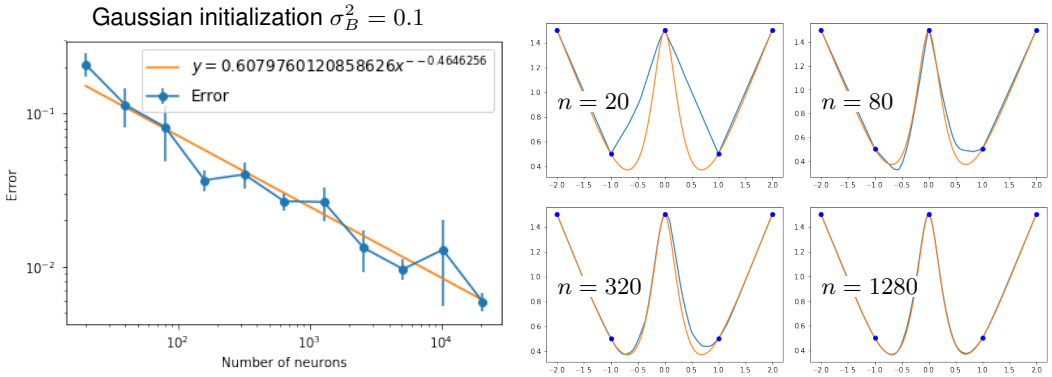

Figure A2: Illustration of Theorem 1. Similar to Figure A1, but with a different initialization $W \sim N(0, 1)$ and $N(0, 0.1)$, which gives rise to a curvature penalty function ζ that is more strongly peaked around $x = 0$ (see Figure 1). We observe in particular that the solutions are more curvy around $x = 0$.

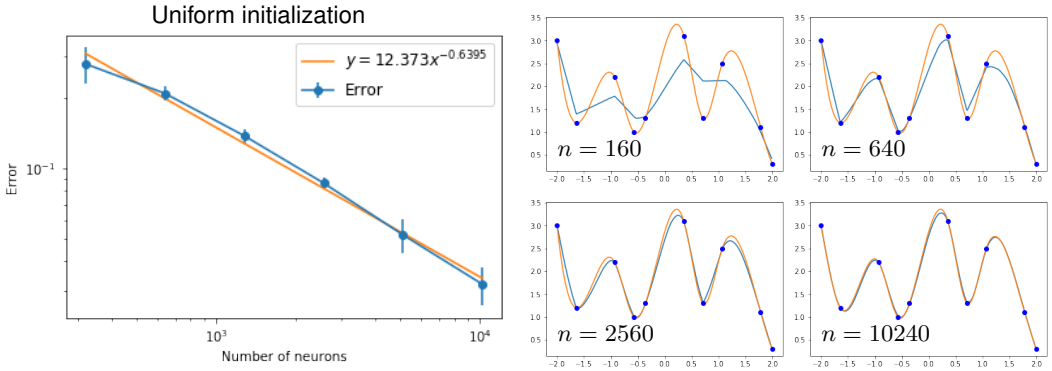

Figure A3: Illustration of Theorem 1. Similar to Figure 1, with uniform initialization, but with a larger dataset and larger networks.

Training all layers versus training only the output layer To illustrate Theorem 3, we conduct the following experiment. We use the same training set as in Figure 1 and use uniform initialization. Starting from the same initial weights, we train the network in two ways. One way is only training the output layer and another way is training all layers of the network. The result is shown in Figure A4. The left panel plots the error between two trained network functions against the number of neurons n. In this experiment the error is of order $n^{-3/2}$, which is even smaller than the upper bound n^{-1} given in Theorem 3. Potentially the bound can be improved. The right panel plots two trained network functions with 20, 80, 320, 1280 neurons.

Effect of linear function on implicit bias In our main result Theorem 1, since the variational problem defines functions only up to addition of linear functions, we need to adjust training data by subtracting a specific linear function $ux + v$. However, in our previous experiments, we don't adjust the training data and the statement of Theorem 1 still approximately holds. The reason might be that the coefficients u and v of the linear function which we need to subtract are relatively small. In order to see the effect of linear function on implicit bias, we conduct the following experiment. Similar to Figure 1, we use uniform initialization. We add a linear function $10x + 10$ to the training data in Figure 1. So the training data we use are $\{(-2, -8.5), (-1, 0.5), (0, 11.5), (1, 20.5), (2, 31.5)\}$. In Figure A5 we show analogous experiments to those in the left panel of Figure 1. In order to clearly show the difference between the trained network function and the solution to the variational problem, we subtract $10x + 10$ from these two functions in the right panel of Figure A5. From the right panel of Figure A5, we see that the difference between plotted two functions is relatively larger than that in Figure 1. From the left panel of Figure A5, we see that the error between these two functions stops to decrease when number of neurons n is larger than 1280. It means that the limit of trained network

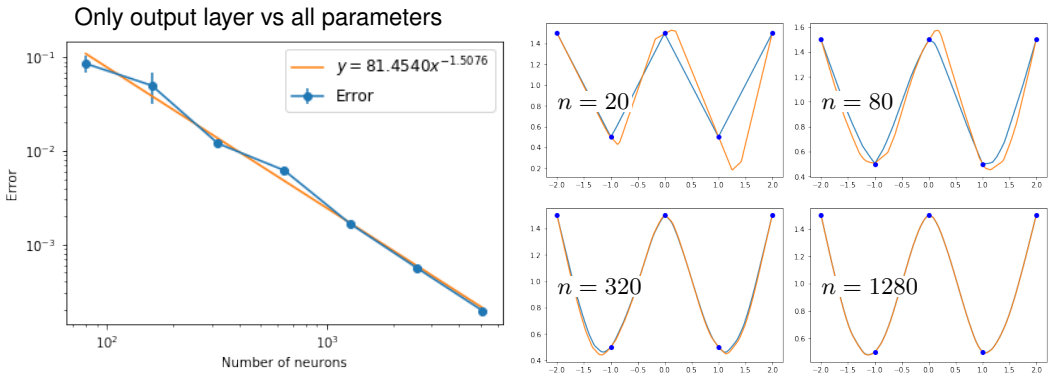

Figure A4: Illustration of Theorem 3. Training only output layer vs training all parameters of the network. We use uniform initialization and the same training set as in Figure 1. The left panel plots the error between two trained network functions against the number of neurons n. For one network, we only train the output layer while for the another one, we train all layers. The right panel shows the data (dots) and two trained network functions with 20, 80, 320, 1280 neurons.

function as $n \to \infty$ is slightly different from the solution to the variational problem. If we choose bigger u and v, we expect that the difference will become larger.

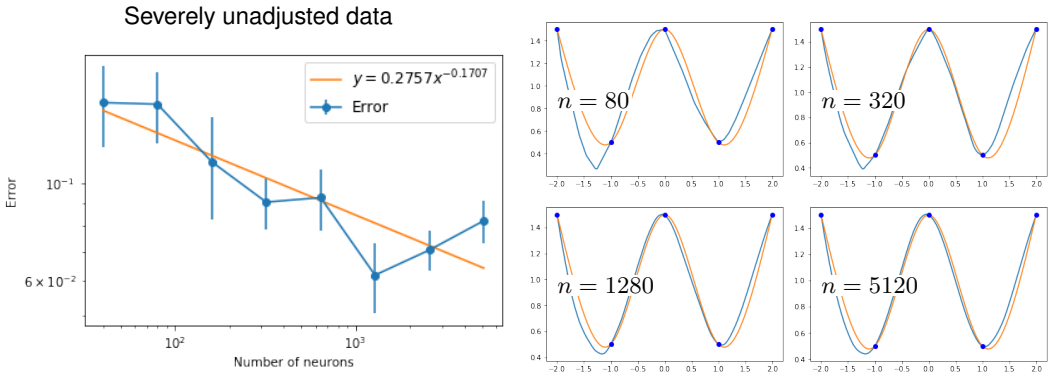

Figure A5: Effect of not adjusting the data. We use uniform initialization and add a linear function $10x + 10$ to the training data of Figure 1. In order to clearly show the difference between trained network function and the solution to the variational problem, we subtract $10x + 10$ from these two functions in the right panel. In the right panel we see that if we ignore u and v in the variational problem (18), the solution is slightly different from (21).

B Details on the numerical implementation

Implementation of gradient descent Training is implemented as full-batch gradient descent. In practice we choose the learning rate as follows. We start with a large learning rate and keep decreasing it by half until we observe that the loss function decreases. After that, we start training with the fixed learning rate we found. We observe that the learning rate we found is inversely proportional to the width n of the neural network. This observation is in accord with Theorem 2 with respect to the upper bound of the learning rate in order to converge.

We note that the implicit bias in parameter space Theorem 2 is independent of the specific step size that is used in the optimization, so long as it is small enough. See Appendix E. The stopping criterion for training of the neural network is that the change in the training loss in consecutive iterations is less than a pre-specified threshold: $|L(\theta_t) - L(\theta_{t-1})| \leq 10^{-8}$.

For the comparison of the functions $f(\cdot, \theta^*)$ and g^*, the 2-norm of error $\|f(\cdot, \theta^*) - g^*\|_2$ is computed by numerical integration with step 0.01 over $[\min(_i x_i), \max_i(x_i)]$.

We use ASI (see Appendix C.2) at initialization. Then the initial output function of the network is $f(\cdot, \theta_0) \equiv 0$. Then according to Theorem 1, the weighted 2-norm of the second derivative of the trained network function is minimized. So in the figures the output function is actually equal to the difference from initialization.

Numerical solution of the variational problem The variational problem for cubic splines can be solved explicitly as described in Appendix N. For a general non-constant curvature penalty $1/\zeta$, we can obtain a numerical solution to problem (21) as follows. First we discretize the interval $[-L, L]$ evenly with points $x_j = -L + 2jL/n$, $j = 0, \dots, n$. For simplicity we suppose that the input training data points are among these grid points, and we denote them by $x_{j_1}, \dots x_{j_m}$. Then we initialize $f(x_j) = 0$ for x_j not in the training data (to be optimized) and $f(x_{j_i}) = y_i$ (fixed values during optimization). We use central differences to approximate the second derivative, $f''(x_j) = \frac{f(x_{j+1}) - 2f(x_j) + f(x_{j-1})}{h^2}$, where $h = |x_{j+1} - x_j|$. Then the objective function in (21) is approximated by $\sum_{j=1}^{n-1} \frac{1}{\zeta(x_j)} \left(\frac{f(x_{j+1}) - 2f(x_j) + f(x_{j-1})}{h^2} \right)^2$. This is quadratic problem in $f(x_j)$, $j \in \{1, \dots, n\} \setminus \{j_1, \dots, j_m\}$. If we equate the gradient to zero, we obtain a linear system. The solution can be written in closed form in terms of the inverse of a design matrix. As with any linear regression problem, in practice we may still prefer to use an iterative approach to obtain a numerical solution. We use a discretization of the interval $[-2, 2]$ into 200 pieces and use conjugate gradient descent for solving the linear system.

C Additional comments

C.1 NTK convergence and positive-definiteness

The convergence of the empirical NTK to a deterministic limiting NTK as the width of the network tends to infinity and the positive-definiteness of this limiting kernel can be ensured whenever the neural network converges to a Gaussian process. The arguments from Jacot et al. (2018) to prove convergence and positive definiteness hold in this case. As they mention, the limiting NTK only depends on the choice of the network activation function, the depth of the network, and the variance of the parameters at initialization. They prove positive definiteness when the input data is supported on a sphere. More generally, positive definiteness can be proved based on the structure of the NTK as a covariance matrix. Let $\|f\|_p^2 = \mathbb{E}_{x \sim p}[f(x)^T f(x)]$, where p denotes the distribution of inputs. The NTK is positive definite when the span of the partial derivatives $\partial_{\theta_i} f(\cdot, \theta)$, $i = 1, \dots, d$, becomes dense in function space with respect to $\| \cdot \|_p$ as the width of the network tends to infinity (Jacot et al., 2018). For a finite data set $x_1, \dots, x_M$, positive definiteness of the corresponding Gram matrix is equivalent to $\partial_{\theta_i} f(x_j, \cdot)$ being linearly independent (Du et al., 2018, Theorem 3.1). This condition for positive definiteness does not depend on the specific distribution of the parameters, but if anything it only depends on the support of the distribution of parameters and on the input data. The precise value of the least eigenvalue may be affected by changes in the distribution however. The convergence of the network function to a Gaussian process in the limit of infinite width and independent parameter initialization is a classic result (Neal, 1996). To verify this Gaussian process assumption it is sufficient that $\sum_i W_i^{(2)} \sigma(W_i^{(1)} x + b_i)$ is a sum of independent random variables with finite variance.

C.2 Anti-Symmetrical Initialization (ASI)

The AntiSymmetrical Initialization (ASI) trick as proposed by Zhang et al. (2019) creates duplicate hidden units with opposite output weights, ensuring that $f(\cdot, \theta_0) \equiv 0$. More precisely, ASI defines $f_{\text{ASI}}(x, \vartheta) = \frac{\sqrt{2}}{2} f(x, \vartheta') - \frac{\sqrt{2}}{2} f(x, \vartheta'')$. Here $\vartheta = (\vartheta', \vartheta'')$ is initialized with $\vartheta_0' = \vartheta_0''$, so that

$$f_{\text{ASI}}(x, \vartheta_0) = \sum_{i=1}^{n} \frac{\sqrt{2}}{2} \overline{V}_i^{(2)} [\overline{V}_i^{(1)} x + \overline{a}_i^{(1)}]_+ + \sum_{i=1}^{n} -\frac{\sqrt{2}}{2} \overline{V}_i^{(2)} [\overline{V}_i^{(1)} x + \overline{a}_i^{(1)}]_+ \equiv 0. \tag{A1}$$

The parameter vector is thus $\vartheta_0 = \text{vec}(\overline{V}^{(1)}, \overline{V}^{(1)}, \overline{a}^{(1)}, \overline{a}^{(1)}, \frac{\sqrt{2}}{2}\overline{V}^{(2)}, -\frac{\sqrt{2}}{2}\overline{V}^{(2)}, \frac{\sqrt{2}}{2}\overline{a}^{(2)}, -\frac{\sqrt{2}}{2}\overline{a}^{(2)})$.

The basic statistics on the size of the parameters remains like (3), even if now there are perfectly correlated pairs of parameters. Hence the analysis and results on limits when the number of hidden units tends to infinity remain valid under ASI. The ASI is not needed for our analysis, which can be

used to compare different types of initialization procedures, but it simplifies some of the presentation. One motivation for using ASI in practical applications is that it provides a simple way to implement a simple output function at initialization. Since the output function at initialization directly influences the bias of the gradient descent solution, this is a particular way to control the bias. Manipulating the bias from initialization is also the motivation presented by Zhang et al. (2019). A related discussion also appears in

C.3 Standard vs NTK parametrization

We have focused on the standard parametrization of the neural network. Jacot et al. (2018) use a non-standard parametrization which is now known as the NTK parametrization. We briefly discuss the difference. A network with NTK parameterization is described as

$$\begin{cases} h^{(l+1)} = \sqrt{\frac{1}{n_l}} w^{(l+1)} x^l + b^{(l+1)} \\ x^{(l+1)} = \phi(h^{(l+1)}) \end{cases} \quad \text{and} \quad \begin{cases} w_{i,j}^{(l)} \sim \mathcal{N}(0,1) \\ b_j^{(l)} \sim \mathcal{N}(0,1) \end{cases} . \tag{A2}$$

In contrast to the standard parametrization, in the NTK parametrization the factor $\sqrt{1/n_l}$ is carried outside of the trainable parameter. In this case, the scaling of the derivatives is $\nabla_{w_i^{(1)}} f(x, \theta_0) = O(n^{-\frac{1}{2}})$ and $\nabla_{w_i^{(2)}} f(x, \theta_0) = O(n^{-\frac{1}{2}})$. In turn, during training the changes of $w_i^{(1)}$ and $w_i^{(2)}$ are comparable in magnitude. This implies that we can not ignore the changes of $w_i^{(1)}$ and approximate the dynamics by that of the linearized model that trains only the output weights as we did in the case of the standard parameterization. In particular, we can not use problem (A62) to describe the result of gradient descent as $n \to \infty$.

C.4 Weight norm minimization

Savarese et al. (2019) studied networks of the form (8) allowing the width to tend to infinity. They showed that the minimum weight norm for approximating a given function g is related to a measure of the smoothness of g by $\lim_{\epsilon \to 0} (\inf_\theta C(\theta) \text{ s.t. } \|f(\cdot, \theta) - g\|_\infty \leq \epsilon) = \max\{\int_{-\infty}^{\infty} |g''(x)| \, \mathrm{d}x, \ |g'(-\infty) + g'(\infty)|\}$, where $C(\theta) = \frac{1}{2} \sum_{i=1}^{n} ((W_i^{(2)})^2 + (W_i^{(1)})^2)$. Here the derivatives are understood in the weak sense. This implies that infinite width shallow networks trained with weight norm regularization (sparing biases) represent functions with smallest 1-norm of the second derivative, an example of which are linear splines. (Note that $C(\theta)$ is not strictly convex in the space of all parameters and also the 1-norm of the second derivative is not strictly convex, hence the solution is not unique).

The result of Savarese et al. (2019) is illuminating in that it connects properties of the parameters and properties of the represented functions. However, the result does not necessarily inform us about the functions represented by the network upon gradient descent training without explicit weight norm regularization. Indeed, if we initialize the parameters by (3) with sub-Gaussian distribution, the neural network can be approximated by the linearized model. Then by Theorem 2, $\|\omega - \theta_0\|_2$ is minimized rather than $\|\omega\|_2$. But in this case $\|\theta_0\|_2$ is bounded away from zero with high probability and the 2-norm of all parameters (or also of the weights only) is not minimized. On the other hand, if we initialize the parameters with $\|\theta_0\|_2$ close to 0, then the neural network might not be well approximated by the linearized model. This has been observed experimentally by Chizat et al. (2019) and we further illustrate it in Appendix C.5.

Even if we assume that the linearization of a network at the origin is valid, in order for the network to approximate certain complex functions, the weights necessarily have to be bounded away from zero. This means that reaching zero training error requires to move far from the basis point, where the difference between linearized and non-linearized model could become significant. In turn, the implicit bias description derived from a linearization at the origin may not accurately reflect the implicit bias of gradient descent in the original non-linearized model.

The paper by Parhi and Nowak (2019) follows the approach of Savarese et al. (2019) and generalizes the result of Savarese et al. (2019) to different types of activation functions σ. Then they show that minimizing the weight "norm" of two-layer neural networks with activation function σ is actually minimizing 1-norm of $\mathrm{L}f$ in place of the second derivative, where f is the output function of the neural network. Here L and σ satisfy $L\sigma = \delta$, i.e. σ is a Green's function of L. Such activation

functions can be used in combination with our analysis. We comment further on such generalizations in Appendix O.

C.5 Basis parameter for linearization of the model

We discuss how the quality of the approximation of a neural network by a linearized model depends on the basis point. For a feedforward ReLU network and a list $\mathcal{X} = (x_i)_{i=1}^m$ of input data points, the mapping $\theta \mapsto f(\mathcal{X}, \theta) = [f(x_1, \theta), \ldots, f(x_m, \theta)]$ is piecewise multilinear. Each of the pieces is smooth and we can assume that it is approximated reasonably well by its Taylor expansion. However, the quality of the approximation can drop when we cross the boundary between smooth pieces. Consider a single-input network with a layer of n ReLUs and a single output unit. At an input x the prediction is $f(x; \theta) = W^{(2)}[W^{(1)}x + b^{(1)}]_+ + b^{(2)}$, where $\theta = (W^{(1)}, b^{(1)}, W^{(2)}, b^{(2)})$. The Jacobian is non-smooth whenever $\theta \in H_{xj} = \{W_{j1}^{(1)} x + b_j^{(1)} = 0\}$ for some $j = 1, \ldots, n$. Hence for m input data points x_i, $i = 1, \ldots, m$, the locus of non-smoothness is given by m central hyperplanes H_{ij}, $i = 1, \ldots, m$ in the parameter space of each hidden unit $j = 1, \ldots, n$. For an individual ReLU, if the parameter θ_0 is drawn from a centrally symmetric probability distribution, the probability p that an ϵ ball around $c\theta_0$ intersects one of the non-linearity hyperplanes H_i, $i = 1, \ldots, m$, behaves roughly as $p = O(mc^{-1})$. Hence we can expect that the prediction function will be better approximated by its linearization $f^{\text{lin}}(x, \theta) = f(x, c\theta_0) + \nabla_\theta f(x, c\theta_0)(\theta - c\theta_0)$ at a point $c\theta_0$ if c is larger. This is well reflected numerically in Figure A6. As we see, for larger initialization the model looks more linear. We observed that this qualitative behavior remains same if we try to adjust the size of the window around the initial value.

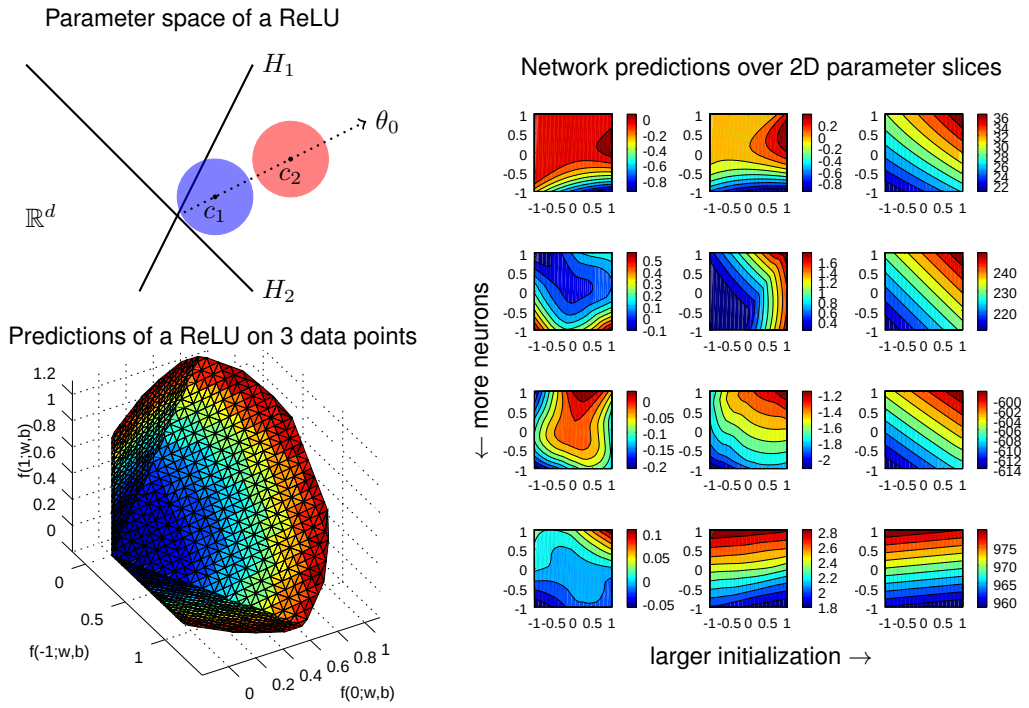

Figure A6: Left: For a single ReLU, the map $\theta \mapsto f(\mathcal{X}, \theta)$ from parameters to prediction vectors over a set $\mathcal{X} = \{x_1, \ldots, x_m\}$ of m input data points is piecewise linear, with pieces separated by m central hyperplanes. Right: Shown is the prediction $f(x, \theta)$ of a shallow ReLU network on a fixed input point x, over a 2D slice of parameters $\theta = c\theta_0 + v_1\xi_1 + v_2\xi_2$ spanned by two random orthogonal unit norm vectors v_1, v_2 and parametrized by $(\xi_1, \xi_2) \in [-1, 1]^2$. From top to bottom, the number of hidden units is $n = 1, 5, 25, 125$ and in each row the initial parameter θ_0 is drawn i.i.d. from a standard Gaussian. In each column we use a different scaling constant $c = 0, 0.5, 10$. As we see, for larger scaling c of the initialization the model looks more linear.

D Proof of Theorem 1

Proof of Theorem 1. The convergence to zero training error for ReLU networks is by now a well known result (Du et al., 2018; Allen-Zhu et al., 2019). We proceed with the implicit bias result.

For simplicity, we give out the proof under ASI (see Appendix C.2). In Section 4.2, we relax the optimization problem (16) to (18). Suppose $(\overline{\gamma}, \overline{u}, \overline{v})$ is the solution of (18). The we can adjust the training samples $\{(x_i, y_i)\}_{i=1}^M$ to $\{(x_i, y_i - \overline{u}x_i - \overline{v})\}_{i=1}^M$. It's easy to see that on the adjusted training samples, $(0, 0, \overline{\gamma})$ is the solution of (18). Then $\overline{\gamma}$ is the solution of (16). Furthermore, the solution of (16) in function space, $g(x, \overline{\gamma})$, equals to the solution of (18) in function space, $g(x, (\overline{\gamma}, 0, 0))$, i.e.

$$g(x, \overline{\gamma}) = g(x, (\overline{\gamma}, 0, 0)). \tag{A3}$$

It we change the variable γ to α as in Section 4.2, we get

$$g(x, \overline{\alpha}) = g(x, \overline{\gamma}), \tag{A4}$$

where $g(x, \overline{\alpha})$ is the solution of the continuous version of problem (15) with μ in place of μ_n. On the set $S = \operatorname{supp}(\zeta) \cap [\min_i x_i, \max_i x_i]$, according to Theorem 5,

$$\sup_{x \in S} |g_n(x, \overline{\alpha}_n) - g(x, \overline{\alpha})| = O(n^{-1/2}), \tag{A5}$$

where $g_n(x, \overline{\alpha}_n)$ is the solution of problem (15) in function space. Since problem (15) is equivalent to problem (14), $g_n(x, \overline{\alpha}_n)$ is also the solution of (14) in function space. According to discussion in Section 4, $f^{\text{lin}}(x_j, \widetilde{\omega}_\infty)$ is the solution of (14). Then

$$g_n(x, \overline{\alpha}_n) = f^{\text{lin}}(x_j, \widetilde{\omega}_\infty). \tag{A6}$$

According to Corollary 4,

$$\|f^{\text{lin}}(x, \tilde{\omega}_\infty) - f(x, \theta^*)\|_2 = O(n^{-\frac{1}{2}}). \tag{A7}$$

Finally, according to Theorem 6 and Proposition 7, $g(x, (\overline{\gamma}, 0, 0))$ restricted on S is the solution of (1), which is $g^*(x)$. Then on the set S,

$$g(x, (\overline{\gamma}, 0, 0)) = g^*(x) \tag{A8}$$

Combining (A3), (A4), (A5), (A6), (A7), (A8), and using the fact that on domain S, $\|f\|_2 \leq \operatorname{vol}(S)\|f\|_\infty$, we prove the theorem. □

E Proof of Theorem 2

Here we give out the proof of Theorem 2. We note that Zhang et al. (2019) prove a similar result for gradient flow. Our proof is for finite step size and different from theirs.

Remark A10 (Remark on Theorem 2, step size). The proof remains valid for a changing step size as long as this satisfies the required inequality.

Proof of Theorem 2. We use gradient descent to minimize $L^{\text{lin}}(\omega) = \sum_{i=1}^M \ell(f^{\text{lin}}(x_i, \omega), y_i)$. First we prove that $\nabla_\omega L^{\text{lin}}(\omega)$ is Lipschitz continuous. Since

$$\begin{aligned}
&\|\nabla_\omega L^{\text{lin}}(\omega_1) - \nabla_\omega L^{\text{lin}}(\omega_2)\|_2 \\
=&\|\nabla_\theta f(\mathcal{X}, \theta_0)^\top \nabla_{f^{\text{lin}}(\mathcal{X}, \omega_1)} L - \nabla_\theta f(\mathcal{X}, \theta_0)^\top \nabla_{f^{\text{lin}}(\mathcal{X}, \omega_2)} L\|_2 \\
\leq&\|\nabla_\theta f(\mathcal{X}, \theta_0)^\top\|_2 \|\nabla_{f^{\text{lin}}(\mathcal{X}, \omega_1)} L - \nabla_{f^{\text{lin}}(\mathcal{X}, \omega_2)} L\|_2 \\
\leq&K\|\nabla_\theta f(\mathcal{X}, \theta_0)^\top\|_2 \|f^{\text{lin}}(\mathcal{X}, \omega_1) - f^{\text{lin}}(\mathcal{X}, \omega_2)\|_1 \quad \text{(K-Lipschitz continuity of } \ell) \\
\leq&K\sqrt{M}\|\nabla_\theta f(\mathcal{X}, \theta_0)^\top\|_2 \|f^{\text{lin}}(\mathcal{X}, \omega_1) - f^{\text{lin}}(\mathcal{X}, \omega_2)\|_2 \\
=&K\sqrt{M}\|\nabla_\theta f(\mathcal{X}, \theta_0)^\top\|_2 \|\nabla_\theta f(\mathcal{X}, \theta_0)(\omega_1 - \omega_2)\|_2 \\
\leq&K\sqrt{M}\|\nabla_\theta f(\mathcal{X}, \theta_0)^\top\|_2 \|\nabla_\theta f(\mathcal{X}, \theta_0)\|_2 \|(\omega_1 - \omega_2)\|_2 \\
\leq&Kn\sqrt{M}\lambda_{\max}(\hat{\Theta}_n)\|\omega_1 - \omega_2\|_2.
\end{aligned} \tag{A9}$$

So $L^{\text{lin}}(\omega)$ is Lipschitz continuous with Lipschitz constant $Kn\sqrt{M}\lambda_{\max}(\hat{\Theta}_n)$. Since L^{lin} is convex over ω, gradient descent with learning rate $\eta = \frac{1}{Kn\sqrt{M}\lambda_{\max}(\hat{\Theta}_n)}$ converges to a global minimium of $L^{\text{lin}}(\omega)$. By assumption that $\text{rank}(\nabla_\theta f(\mathcal{X}, \theta_0)) = M$, the model can perfectly fit all data. Then the minimium of $L^{\text{lin}}(\omega)$ is zero and gradient descent converges to zero loss.

Let $\omega_\infty = \lim_{t\to\infty} \omega_t$. Then $f^{\text{lin}}(\mathcal{X}, \omega_\infty) = \mathcal{Y}$. According to gradient descent iteration,

$$\begin{aligned}\omega_\infty &= \theta_0 - \sum_{t=0}^{\infty} \eta \nabla_\theta f(\mathcal{X}, \theta_0)^T \nabla_{f^{\text{lin}}(\mathcal{X}, \omega_t)} L^{\text{lin}} \\ &= \theta_0 - \eta \nabla_\theta f(\mathcal{X}, \theta_0)^T \sum_{t=0}^{\infty} \nabla_{f^{\text{lin}}(\mathcal{X}, \omega_t)} L^{\text{lin}}\end{aligned} \tag{A10}$$

Since f^{lin} is linear over weights ω and $\|\omega - \theta_0\|_2$ is strongly convex, the constrained optimization problem (7) is a strongly convex optimization problem. The first order optimality condition of the problem is

$$\begin{cases}\omega - \theta_0 + \nabla_\theta f^{\text{lin}}(\mathcal{X}, \theta_0)^T \lambda = 0 \\ f^{\text{lin}}(\mathcal{X}, \omega) = \mathcal{Y}.\end{cases} \tag{A11}$$

Let $\lambda = \sum_{t=0}^{\infty} \nabla_{f^{\text{lin}}(\mathcal{X}, \theta_t)} L$, we can easily check out that ω_∞ satisfies condition (A11). So ω_∞ is the solution of problem (7). □

Remark A11 (Remark to Theorem 2). Making an analogous statement to Theorem 2 to describe the bias in parameter space when training wide networks rather than the linearized model is interesting, but harder, because the gradient direction is no longer constant. Oymak and Soltanolkotabi (2019) obtain bounds on the trajectory length in parameter space, putting the final solution within a factor $4\beta/\alpha$ of $\min_\theta \|\theta_0 - \theta\|$, where β and α are upper and lower bounds on the singular values of the Jacobian over the relevant region. However, currently it is unclear whether the solution upon gradient optimization is indeed the distance minimizer from initialization.

F Proof of Theorem 3

We note that assumption (2) $\liminf_{n\to\infty} \lambda_{\min}(\hat{\Theta}_n) > 0$ is satisfied if the empirical NTK converges and the limit NTK is positive definite. For details see Appendix C.1.

Proof of Theorem 3. According to (6),

$$\omega_{t+1} = \omega_t - \eta \nabla_\theta f(\mathcal{X}, \theta_0)^T \nabla_{f^{\text{lin}}(\mathcal{X}, \omega_t)} L^{\text{lin}}. \tag{A12}$$

Since we use the MSE loss,

$$\omega_{t+1} = \omega_t - \eta \nabla_\theta f(\mathcal{X}, \theta_0)^T (f^{\text{lin}}(\mathcal{X}, \omega_t) - \mathcal{Y}). \tag{A13}$$

Using (5), we get

$$\begin{aligned}f^{\text{lin}}(\mathcal{X}, \omega_{t+1}) &= f^{\text{lin}}(\mathcal{X}, \omega_t) - \eta \nabla_\theta f(\mathcal{X}, \theta_0) \nabla_\theta f(\mathcal{X}, \theta_0)^T (f^{\text{lin}}(\mathcal{X}, \omega_t) - \mathcal{Y}) \\ &= f^{\text{lin}}(\mathcal{X}, \omega_t) - n\eta \hat{\Theta}_n (f^{\text{lin}}(\mathcal{X}, \omega_t) - \mathcal{Y}).\end{aligned} \tag{A14}$$

Then we have

$$f^{\text{lin}}(\mathcal{X}, \omega_{t+1}) - \mathcal{Y} = (I - n\eta\hat{\Theta}_n)(f^{\text{lin}}(\mathcal{X}, \omega_t) - \mathcal{Y}), \tag{A15}$$

and

$$\begin{aligned}f^{\text{lin}}(\mathcal{X}, \omega_t) - \mathcal{Y} &= (I - n\eta\hat{\Theta}_n)^t (f^{\text{lin}}(\mathcal{X}, \theta_0) - \mathcal{Y}) \\ &= (I - n\eta\hat{\Theta}_n)^t (f(\mathcal{X}, \theta_0) - \mathcal{Y}).\end{aligned} \tag{A16}$$

According to the update rule of ω_t, we know that $\omega_t = \nabla_\theta f(\mathcal{X}, \theta_0)^T \xi + \theta_0$, where ξ is a column vector. Then

$$\begin{aligned}f^{\text{lin}}(\mathcal{X}, \omega_t) - \mathcal{Y} &= f^{\text{lin}}(\mathcal{X}, \omega_t) - f(\mathcal{X}, \theta_0) + f(\mathcal{X}, \theta_0) - \mathcal{Y} \\ &= \nabla_\theta f(\mathcal{X}, \theta_0)(\omega_t - \theta_0) + f(\mathcal{X}, \theta_0) - \mathcal{Y} \\ &= \nabla_\theta f(\mathcal{X}, \theta_0) \nabla_\theta f(\mathcal{X}, \theta_0)^T \xi + f(\mathcal{X}, \theta_0) - \mathcal{Y} \\ &= n\hat{\Theta}_n \xi + f(\mathcal{X}, \theta_0) - \mathcal{Y} \\ &= (I - n\eta\hat{\Theta}_n)^t (f(\mathcal{X}, \theta_0) - \mathcal{Y}).\end{aligned} \tag{A17}$$

From above equation we can solve for ξ:

$$\xi = -n^{-1}\hat{\Theta}_n^{-1}[I - (I - n\eta\hat{\Theta}_n)^t](f(\mathcal{X}, \theta_0) - \mathcal{Y}). \tag{A18}$$

Therefore

$$\omega_t = -n^{-1}\nabla_\theta f(\mathcal{X}, \theta_0)^T\hat{\Theta}_n^{-1}[I - (I - n\eta\hat{\Theta}_n)^t](f(\mathcal{X}, \theta_0) - \mathcal{Y}) + \theta_0. \tag{A19}$$

For any $x \in \mathbb{R}$,

$$\begin{aligned} f^{\text{lin}}(x, \omega_t) &= f(x, \theta_0) + \nabla_\theta f(x, \theta_0)(\omega_t - \theta_0) \\ &= f(x, \theta_0) - n^{-1}\nabla_\theta f(x, \theta_0)\nabla_\theta f(\mathcal{X}, \theta_0)^T\hat{\Theta}_n^{-1}[I - (I - n\eta\hat{\Theta}_n)^t](f(\mathcal{X}, \theta_0) - \mathcal{Y}). \end{aligned} \tag{A20}$$

For the training process (10), we can define the corresponding empirical neural tangent kernel in the following way:

$$\tilde{\Theta}_n = \frac{1}{n}\nabla_{W^{(2)}} f(\mathcal{X}, \theta_0)\nabla_{W^{(2)}} f(\mathcal{X}, \theta_0)^T. \tag{A21}$$

Using the same argument, we have

$$\widetilde{W}_t^{(2)} = -n^{-1}\nabla_{W^{(2)}} f(\mathcal{X}, \theta_0)^T\tilde{\Theta}_n^{-1}[I - (I - n\eta\tilde{\Theta}_n)^t](f(\mathcal{X}, \theta_0) - \mathcal{Y}) + \overline{W}_0^{(2)} \tag{A22}$$

and

$$f^{\text{lin}}(x, \widetilde{\omega}_t) = f(x, \theta_0) - n^{-1}\nabla_{W^{(2)}} f(x, \theta_0)\nabla_{W^{(2)}} f(\mathcal{X}, \theta_0)^T\tilde{\Theta}_n^{-1}[I - (I - n\eta\tilde{\Theta}_n)^t](f(\mathcal{X}, \theta_0) - \mathcal{Y}). \tag{A23}$$

Then

$$\begin{aligned} &|f^{\text{lin}}(x, \widetilde{\omega}_t) - f^{\text{lin}}(x, \omega_t)| \\ =&n^{-1}\Big|\nabla_\theta f(x, \theta_0)\nabla_\theta f(\mathcal{X}, \theta_0)^T\hat{\Theta}_n^{-1}[I - (I - n\eta\hat{\Theta}_n)^t](f(\mathcal{X}, \theta_0) - \mathcal{Y}) \\ &-\nabla_{W^{(2)}} f(x, \theta_0)\nabla_{W^{(2)}} f(\mathcal{X}, \theta_0)^T\tilde{\Theta}_n^{-1}[I - (I - n\eta\tilde{\Theta}_n)^t](f(\mathcal{X}, \theta_0) - \mathcal{Y})\Big|. \end{aligned} \tag{A24}$$

The next step is to compute the difference between $\tilde{\Theta}_n$ and $\hat{\Theta}_n$. Let $\Delta\Theta = \hat{\Theta}_n - \tilde{\Theta}_n$, then the ij-th entry of the matrix $\Delta\Theta$ is

$$\begin{aligned} (\Delta\Theta)_{ij} = \frac{1}{n}\Bigg[&\sum_{k=1}^{n}\Big(\nabla_{W_k^{(1)}} f(x_i, \theta_0)\nabla_{W_k^{(1)}} f(x_j, \theta_0) + \nabla_{b_k^{(1)}} f(x_i, \theta_0)\nabla_{b_k^{(1)}} f(x_j, \theta_0)\Big) \\ &+ \nabla_{b^{(2)}} f(x_i, \theta_0)\nabla_{b^{(2)}} f(x_j, \theta_0)\Bigg]. \end{aligned} \tag{A25}$$

According to initialization (3), we can find a $C > 0$ such that $|W_i^{(1)}|, |b_i^{(1)}| \leq C$ and $|W_i^{(2)}|, |b^{(2)}| \leq Cn^{-\frac{1}{2}}$ with probability at least $(1 - \delta/4)$. Then given $x \in \mathbb{R}$,

$$|\nabla_{W_i^{(1)}} f(x, \theta_0)| = |W_i^{(2)} H(W_i^{(1)}x + b) \cdot x| \leq Cn^{-\frac{1}{2}}x = O(n^{-\frac{1}{2}}), \tag{A26}$$

$$|\nabla_{b_i^{(1)}} f(x, \theta_0)| = |W_i^{(2)} H(W_i^{(1)}x + b_i^{(1)})| \leq Cn^{-\frac{1}{2}} = O(n^{-\frac{1}{2}}) \tag{A27}$$

$$|\nabla_{W_i^{(2)}} f(x, \theta_0)| = [W_i^{(1)}x + b_i^{(1)}]_+ \leq C|x| + C = O(1), \tag{A28}$$

$$|\nabla_{b^{(2)}} f(x, \theta_0)| = 1 = O(1). \tag{A29}$$

So

$$\begin{aligned} |(\Delta\Theta)_{ij}| &\leq \frac{1}{n}\left[\sum_{k=1}^{n}\Big(O(n^{-\frac{1}{2}})O(n^{-\frac{1}{2}}) + O(n^{-\frac{1}{2}})O(n^{-\frac{1}{2}})\Big) + O(1)O(1)\right] \\ &= O(n^{-1}). \end{aligned} \tag{A30}$$

Since the size of $\Delta\Theta$ is $M \times M$, which does not change as n goes up. So $\|\Delta\Theta\|_2 = O(n^{-1})$, which means $\|\hat{\Theta}_n - \tilde{\Theta}_n\|_2 = O(n^{-1})$.

Now we measure the difference of each part in (A24). According to assumption (2), $\inf_n \lambda_{\min}(\hat{\Theta}_n) > 0$, then

$$\lambda_{\min}(\hat{\Theta}_n) \geq \frac{1}{\inf_n \lambda_{\min}(\hat{\Theta}_n)} = O(1) \tag{A31}$$

$$\lambda_{\min}(\tilde{\Theta}_n) \geq \frac{1}{\inf_n \lambda_{\min}(\hat{\Theta}_n) - O(n^{-1})} = O(1). \tag{A32}$$

So

$$\begin{aligned} \|\hat{\Theta}_n^{-1} - \tilde{\Theta}_n^{-1}\|_2 &= \|\hat{\Theta}_n^{-1}(\tilde{\Theta}_n - \hat{\Theta}_n)\tilde{\Theta}_n^{-1}\|_2 \\ &\leq \|\hat{\Theta}_n^{-1}\|_2 \|\Delta\Theta\|_2 \|\tilde{\Theta}_n^{-1}\|_2 \\ &= O(n^{-1}). \end{aligned} \tag{A33}$$

The assumption $\eta < \frac{2}{n\lambda_{\max}(\hat{\Theta}_n)}$ implies

$$\|I - n\eta\hat{\Theta}_n\|_2 < 1, \tag{A34}$$

And

$$\begin{aligned} \|I - n\eta\tilde{\Theta}_n\|_2 &\leq \|I - n\eta\hat{\Theta}_n\|_2 + n\eta\|\hat{\Theta}_n - \Theta\|_2 \\ &\leq \max\{n\eta\frac{\lambda_{\max}(\Theta)}{2}, 1 - n\eta\lambda_{\min}(\hat{\Theta}_n)\} + O(n^{-1}). \end{aligned} \tag{A35}$$

As n is large enough, we also have $\|I - n\eta\tilde{\Theta}_n\|_2 < 1$. Then as n is large enough,

$$\begin{aligned} &\|[I - (I - n\eta\hat{\Theta}_n)^t] - [I - (I - n\eta\tilde{\Theta}_n)^t]\|_2 \\ &= \|(I - n\eta\hat{\Theta}_n)^t - (I - n\eta\tilde{\Theta}_n)^t\|_2 \\ &\leq \|[(I - n\eta\hat{\Theta}_n) - (I - n\eta\tilde{\Theta}_n)](I - n\eta\hat{\Theta}_n)^{t-1}\|_2 \\ &\quad + \|(I - n\eta\tilde{\Theta}_n)[(I - n\eta\hat{\Theta}_n) - (I - n\eta\tilde{\Theta}_n)](I - n\eta\hat{\Theta}_n)^{t-2}\|_2 \\ &\quad + \cdots \\ &\quad + \|(I - n\eta\tilde{\Theta}_n)^{t-1}[(I - n\eta\hat{\Theta}_n) - (I - n\eta\tilde{\Theta}_n)]\|_2 \\ &\leq \eta\|\hat{\Theta}_n - \tilde{\Theta}_n\|_2\|I - n\eta\hat{\Theta}_n\|_2^{t-1} \\ &\quad + \eta\|I - n\eta\tilde{\Theta}_n\|_2\|\hat{\Theta}_n - \tilde{\Theta}_n\|_2\|I - n\eta\hat{\Theta}_n\|_2^{t-2} \\ &\quad + \cdots \\ &\quad + \eta\|I - n\eta\tilde{\Theta}_n\|_2^{t-1}\|\hat{\Theta}_n - \tilde{\Theta}_n\|_2 \\ &\leq \eta\|\hat{\Theta}_n - \tilde{\Theta}_n\|_2 \cdot t \cdot (\max\{\|I - n\eta\hat{\Theta}_n\|_2, \|I - n\eta\tilde{\Theta}_n\|_2\})^{t-1}. \end{aligned} \tag{A36}$$

Since $\max\{\|I - n\eta\hat{\Theta}_n\|_2, \|I - n\eta\tilde{\Theta}_n\|_2\} < 1$, $\sup_{t>0} t \cdot (\max\{\|I - n\eta\hat{\Theta}_n\|_2, \|I - n\eta\tilde{\Theta}_n\|_2\})^{t-1}$ is a finite number. So

$$\begin{aligned} \|[I - (I - n\eta\hat{\Theta}_n)^t] - [I - (I - n\eta\tilde{\Theta}_n)^t]\|_2 &\leq O(\eta\|\hat{\Theta}_n - \tilde{\Theta}_n\|_2) \\ &\leq O(n^{-1}). \end{aligned} \tag{A37}$$

Let $\Delta\Theta(x, \mathcal{X}) = n^{-1}(\nabla_\theta f(x, \theta_0)\nabla_\theta f(\mathcal{X}, \theta_0)^T - \nabla_{W^{(2)}} f(x, \theta_0)\nabla_{W^{(2)}} f(\mathcal{X}, \theta_0)^T)$, then the i-th entry of the vector $\Delta\Theta(x, \mathcal{X})$ is

$$\begin{aligned} (\Delta\Theta(x, \mathcal{X}))_i =& \frac{1}{n}\left[\sum_{k=1}^{n}\left(\nabla_{W_k^{(1)}} f(x, \theta_0)\nabla_{W_k^{(1)}} f(x_i, \theta_0) + \nabla_{b_k^{(1)}} f(x, \theta_0)\nabla_{b_k^{(1)}} f(x_i, \theta_0)\right)\right. \\ &\left. + \nabla_{b^{(2)}} f(x, \theta_0)\nabla_{b^{(2)}} f(x_i, \theta_0)\right]. \end{aligned} \tag{A38}$$

According to (A26), (A27), (A28) and (A29), we have

$$\begin{aligned} |(\Delta\Theta(x, \mathcal{X}))_i| &\leq \frac{1}{n}\left[\sum_{k=1}^{n}\left(O(n^{-\frac{1}{2}})O(n^{-\frac{1}{2}}) + O(n^{-\frac{1}{2}})O(n^{-\frac{1}{2}})\right) + O(1)O(1)\right] \\ &= O(n^{-1}). \end{aligned} \tag{A39}$$

Since the size of $\Delta\Theta(x,\mathcal{X})$ is M, which does not change as n goes up. So

$$\|\Delta\Theta(x,\mathcal{X})\|_2 = O(n^{-1}). \tag{A40}$$

Let $\tilde{\Theta}_n(x,\mathcal{X}) = n^{-1}(\nabla_{W^{(2)}} f(x,\theta_0)\nabla_{W^{(2)}} f(\mathcal{X},\theta_0)^T))$, then the i-th entry of the vector $\tilde{\Theta}_n(x,\mathcal{X})$ is

$$\begin{aligned}
|(\tilde{\Theta}_n(x,\mathcal{X}))_i| &\leq \frac{1}{n}\sum_{k=1}^{n} |\nabla_{W_k^{(2)}} f(x,\theta_0)\nabla_{W_k^{(2)}} f(x_i,\theta_0)| \\
&\leq \frac{1}{n}\sum_{k=1}^{n} |O(1)O(1)| \\
&= O(1).
\end{aligned} \tag{A41}$$

Since the size of $\tilde{\Theta}_n(x,\mathcal{X})$ is M, which does not change as n goes up. So

$$\|\Theta(x,\mathcal{X})\|_2 = O(n^{-1}). \tag{A42}$$

Neal (1996), Lee et al. (2018) show that as n goes to infinity, the output function at initialization $f(\cdot,\theta_0)$ tends to a Gaussian process, which means that $f(\mathcal{X},\theta_0) \sim \mathcal{N}(0,\mathcal{K}(\mathcal{X},\mathcal{X}))$. Here $\mathcal{K}(\mathcal{X},\mathcal{X})$ can be computed recursively. So as n is large enough, we can find a $C_1 > 0$ such that $f(x_i,\theta_0) \leq C_1, i = 1,\cdots,M$ with probability at least $(1-\delta/4)$. Then

$$\|f(\mathcal{X},\theta_0) - \mathcal{Y}\|_2 = O(1). \tag{A43}$$

Combine all these together, we have that with probability at least $(1-\delta)$, (A31), (A32), (A33), (A34), (A37), (A40), (A41) and (A43) hold true. Then follow the equation (A24), we get

$$\begin{aligned}
&|f^{\text{lin}}(x,\widetilde{\omega}_t) - f(x,\theta_t)| \\
=&n^{-1}|\nabla_\theta f(x,\theta_0)\nabla_\theta f(\mathcal{X},\theta_0)^T\hat{\Theta}_n^{-1}[I-(I-n\eta\hat{\Theta}_n)^t](f(\mathcal{X},\theta_0)-\mathcal{Y}) \\
&-\nabla_{W^{(2)}} f(x,\theta_0)\nabla_{W^{(2)}} f(\mathcal{X},\theta_0)^T\tilde{\Theta}_n^{-1}[I-(I-n\eta\tilde{\Theta}_n)^t](f(\mathcal{X},\theta_0)-\mathcal{Y})| \\
=&n^{-1}\|\nabla_\theta f(x,\theta_0)\nabla_\theta f(\mathcal{X},\theta_0)^T\hat{\Theta}_n^{-1}[I-(I-n\eta\hat{\Theta}_n)^t] \\
&-\nabla_{W^{(2)}} f(x,\theta_0)\nabla_{W^{(2)}} f(\mathcal{X},\theta_0)^T\tilde{\Theta}_n^{-1}[I-(I-n\eta\tilde{\Theta}_n)^t]\|_2\|f(\mathcal{X},\theta_0)-\mathcal{Y}\|_2 \\
=&n^{-1}\|\nabla_\theta f(x,\theta_0)\nabla_\theta f(\mathcal{X},\theta_0)^T\hat{\Theta}_n^{-1}[I-(I-n\eta\hat{\Theta}_n)^t] \\
&-\nabla_{W^{(2)}} f(x,\theta_0)\nabla_{W^{(2)}} f(\mathcal{X},\theta_0)^T\tilde{\Theta}_n^{-1}[I-(I-n\eta\tilde{\Theta}_n)^t]\|_2\cdot O(1).
\end{aligned} \tag{A44}$$

And

$$\begin{aligned}
&n^{-1}\|\nabla_\theta f(x,\theta_0)\nabla_\theta f(\mathcal{X},\theta_0)^T\hat{\Theta}_n^{-1}[I-(I-n\eta\hat{\Theta}_n)^t] \\
&-\nabla_{W^{(2)}} f(x,\theta_0)\nabla_{W^{(2)}} f(\mathcal{X},\theta_0)^T\tilde{\Theta}_n^{-1}[I-(I-n\eta\tilde{\Theta}_n)^t]\|_2 \\
\leq&n^{-1}\|\nabla_\theta f(x,\theta_0)\nabla_\theta f(\mathcal{X},\theta_0)^T - \nabla_{W^{(2)}} f(x,\theta_0)\nabla_{W^{(2)}} f(\mathcal{X},\theta_0)^T\|\|\hat{\Theta}_n^{-1}\|_2\|I-(I-n\eta\hat{\Theta}_n)^t\|_2 \\
&+n^{-1}\|\nabla_{W^{(2)}} f(x,\theta_0)\nabla_{W^{(2)}} f(\mathcal{X},\theta_0)^T\|_2\|\hat{\Theta}_n^{-1}-\tilde{\Theta}_n^{-1}\|_2\|I-(I-n\eta\hat{\Theta}_n)^t\|_2 \\
&+n^{-1}\|\nabla_{W^{(2)}} f(x,\theta_0)\nabla_{W^{(2)}} f(\mathcal{X},\theta_0)^T\|_2\|\tilde{\Theta}_n^{-1}\|_2\|[I-(I-n\eta\hat{\Theta}_n)^t]-[I-(I-n\eta\tilde{\Theta}_n)^t]\|_2 \\
\leq&O(n^{-1})O(1)O(1)+O(1)O(n^{-1})O(1)+O(1)O(1)O(n^{-1}) \\
=&O(n^{-1}).
\end{aligned} \tag{A45}$$

So we have $|f^{\text{lin}}(x,\widetilde{\omega}_t) - f(x,\theta_t)| = O(n^{-1})$, and $O(n^{-1})$ does not contain any constant factor which is related to t. Then

$$\sup_t |f^{\text{lin}}(x,\widetilde{\omega}_t) - f^{\text{lin}}(x,\omega_t)| = O(n^{-1}), \text{ as } n\to\infty. \tag{A46}$$

For the difference of parameters, we have

$$\widetilde{\omega}_t - \omega_t = \text{vec}(\overline{W}_t^{(1)} - \widehat{W}_t^{(1)}, \overline{b}_t^{(1)} - \widehat{b}_t^{(1)}, \widetilde{W}_t^{(2)} - \widehat{W}_t^{(2)}, \overline{b}_t^{(2)} - \widehat{b}_t^{(2)}). \tag{A47}$$

According to (A19) and (A22),

$$\begin{aligned}
\|\overline{W}_t^{(1)} - \widehat{W}_t^{(1)}\|_2 &= \|n^{-1}\nabla_{W^{(1)}} f(\mathcal{X},\theta_0)^T\hat{\Theta}_n^{-1}[I-(I-n\eta\hat{\Theta}_n)^t](f(\mathcal{X},\theta_0)-\mathcal{Y})\|_2 \\
&\leq \|n^{-1}\nabla_{W^{(1)}} f(\mathcal{X},\theta_0)^T\|_2\|\hat{\Theta}_n^{-1}\|_2\|I-(I-n\eta\hat{\Theta}_n)^t\|_2\|f(\mathcal{X},\theta_0)-\mathcal{Y}\|_2 \\
&\leq n^{-1}\|\nabla_{W^{(1)}} f(\mathcal{X},\theta_0)^T\|_2\cdot O(1).
\end{aligned} \tag{A48}$$

Here $\nabla_{W^{(1)}} f(\mathcal{X}, \theta_0)^T$ is a $n \times M$ matrix, the ij-th entry of the matrix is $\nabla_{W_i^{(1)}} f(x_j, \theta_0)$. According to (A26), we have $\nabla_{W_i^{(1)}} f(x_j, \theta_0) = O(n^{-1/2})$. Then $\|\nabla_{W^{(1)}} f(\mathcal{X}, \theta_0)^T\|_2 = O(1)$. So we have $\|\overline{W}_t^{(1)} - \widehat{W}_t^{(1)}\|_2 = O(n^{-1})$, and $O(n^{-1})$ does not contain any constant factor which is related to t. Then

$$\sup_t \|\overline{W}_t^1 - \widehat{W}_t^1\|_2 = O(n^{-1}), \text{ as } n \to \infty. \tag{A49}$$

Similarly we can prove

$$\sup_t \|\bar{b}_t^1 - \hat{b}_t^1\|_2 = O(n^{-1}), \text{ as } n \to \infty. \tag{A50}$$

$$\sup_t \|\bar{b}_t^2 - \hat{b}_t^2\| = O(n^{-1}), \text{ as } n \to \infty. \tag{A51}$$

For $\widetilde{W}_t^{(2)} - \widehat{W}_t^{(2)}$, we have

$$\begin{aligned}
\|\overline{W}_t^{(2)} - \widehat{W}_t^{(2)}\|_2 &= \|n^{-1}\nabla_{W^{(2)}} f(\mathcal{X}, \theta_0)^T \Big(\hat{\Theta}_n^{-1}[I - (I - n\eta\hat{\Theta}_n)^t] - \\
&\qquad \tilde{\Theta}_n^{-1}[I - (I - n\eta\tilde{\Theta}_n)^t]\Big) (f(\mathcal{X}, \theta_0) - \mathcal{Y})\|_2 \\
&\le \|n^{-1}\nabla_{W^{(2)}} f(\mathcal{X}, \theta_0)^T\|_2 \Big(\|\hat{\Theta}_n^{-1} - \tilde{\Theta}_n^{-1}\|_2 \|I - (I - n\eta\hat{\Theta}_n)^t\|_2 + \\
&\qquad \|\tilde{\Theta}_n^{-1}\|_2 \|[I - (I - n\eta\hat{\Theta})^t] - [I - (I - n\eta\tilde{\Theta})^t]\|_2\Big) \|f(\mathcal{X}, \theta_0) - \mathcal{Y}\|_2 \\
&\le n^{-1}\|\nabla_{W^{(2)}} f(\mathcal{X}, \theta_0)^T\|_2 (O(n^{-1})O(1) + O(1)O(n^{-1})) \cdot O(1) \\
&= O(n^{-2})\|\nabla_{W^{(2)}} f(\mathcal{X}, \theta_0)^T\|_2.
\end{aligned} \tag{A52}$$

Here $\nabla_{W^{(2)}} f(\mathcal{X}, \theta_0)^T$ is a $n \times M$ matrix, the ij-th entry of the matrix is $\nabla_{W_i^{(2)}} f(x_j, \theta_0)$. According to (A28), we have $\nabla_{W_i^{(2)}} f(x_j, \theta_0) = O(1)$. Then $\|\nabla_{W^{(2)}} f(\mathcal{X}, \theta_0)^T\|_2 = O(n^{1/2})$. So we have $\|\widetilde{W}_t^{(2)} - \widehat{W}_t^{(2)}\|_2 = O(n^{-3/2})$, and $O(n^{-3/2})$ does not contain any constant factor which is related to t. Then

$$\sup_t \|\widetilde{W}_t^2 - \widehat{W}_t^2\|_2 = O(n^{-3/2}), \text{ as } n \to \infty. \tag{A53}$$

□

G Proof of Corollary 4

Corollary 4 is obtained by combining Theorem 3 and the fact that training a linearized model approximates training a wide network (Lee et al., 2019, Theorem H.1). Although Lee et al. (2019, Theorem H.1) consider Gaussian initialization, the arguments extend to sub-Gaussian initialization.

Proof of Corollary 4. Using Theorem 3, we have that

$$\sup_t |f^{\text{lin}}(x, \widetilde{\omega}_t) - f^{\text{lin}}(x, \omega_t)| = O(n^{-1}), \text{ as } n \to \infty. \tag{A54}$$

According to Lee et al. (2019, Theorem H.1), in the case of Gaussian initialization, we have

$$\sup_t |f^{\text{lin}}(x, \omega_t) - f(x, \theta)| = O(n^{-\frac{1}{2}}), \text{ as } n \to \infty. \tag{A55}$$

Under our neural network setting, which is a one-input network with a single hidden layer of n ReLUs and a linear output, we can generalize the above result to sub-Gaussian initialization. In the remark of Theorem 3, we illustrate that the empirical NTK converges to analytic NTK for initialization with finite variance distribution. Then for sub-Gaussian initialization the empirical NTK still converges to analytic NTK. Then the only part we need to adapt in the proof of Lee et al. (2019, Theorem H.1) is the following theorem (Lee et al., 2019, Theorem G.3):

Theorem A12. *Let A be an $N \times n$ random matrix whose entries are independent standard normal random variables. Then for every $t \ge 0$, with probability at least $1 - 2\exp(-t^2/2)$ one has*

$$\|A\|_{\text{op}} \le \sqrt{N} + \sqrt{n} + t. \tag{A56}$$

Then Lee et al. (2019) applies the above theorem to weight matrices in the neural network. In our case, the weight matrices $W^{(1)}$ and $W^{(2)}$ are $1 \times n$ matrices, which can be regarded as vectors. So

$$\|W^{(1)}\|_{\text{op}} = \sqrt{\sum_{i=1}^{n}(W_i^{(1)})^2}. \tag{A57}$$

Now we use sub-Gaussian initialization. Then $\mathbb{P}(|W_i^{(1)}| \geq t) \leq 2\exp(-t^2/2\sigma^2)$ for some positive σ. Then $(W_i^{(1)})^2$ is sub-exponential. Using the property of sub-Gaussian exponential, we have $\mathbb{E}\exp(|W_i^{(1)})^2|/\lambda) \leq 2$ for some positive λ. Using Vershynin (2018, Theorem 1.4.1), we have

$$\mathbb{P}\left(|\sum_{i=1}^{n}(W_i^{(1)})^2 - \mathbb{E}\sum_{i=1}^{n}(W_i^{(1)})^2| \geq t\right) \leq 2\exp\left[-c\min\left(\frac{t^2}{n\lambda^2}, \frac{t}{\lambda}\right)\right]. \tag{A58}$$

Let $t = n\lambda$, then we have

$$\mathbb{P}\left(\sum_{i=1}^{n}(W_i^{(1)})^2 \geq \mathbb{E}\sum_{i=1}^{n}(W_i^{(1)})^2 + n\lambda\right) \leq 2\exp(-cn). \tag{A59}$$

Since $2\exp(-cn) \to 0$ as $n \to \infty$, the above equation means that with arbitrarily high probability,

$$\begin{aligned}\sum_{i=1}^{n}(W_i^{(1)})^2 &\leq \mathbb{E}\sum_{i=1}^{n}(W_i^{(1)})^2 + n\lambda \\ &= n\mathbb{E}(W_i^{(1)})^2 + n\lambda \\ &= O(n).\end{aligned} \tag{A60}$$

So $\|W^{(1)}\|_{\text{op}} = O(\sqrt{n})$. For the same reason, $\|W^{(2)}\|_{\text{op}} = O(1)$. Then follow the remaining argument of Lee et al. (2019) we can show that

$$\sup_t |f^{\text{lin}}(x, \omega_t) - f(x, \theta)| = O(n^{-\frac{1}{2}}), \text{ as } n \to \infty. \tag{A61}$$

Combine the above equation with (A54) then we finish the proof. □

H Proof of Theorem 5

We consider the continuous version of problem (15):

$$\begin{aligned}\min_{\alpha \in C(\mathbb{R}^2)} \quad & \int_{\mathbb{R}^2} \alpha^2(W^{(1)}, b)\, \mathrm{d}\mu(W^{(1)}, b) \\ \text{subject to} \quad & \int_{\mathbb{R}^2} \alpha(W^{(1)}, b)[W^{(1)}x_j + b]_+ \, \mathrm{d}\mu(W^{(1)}, b) = y_j, \quad j = 1, \ldots, M.\end{aligned} \tag{A62}$$

Here the only difference between (15) and (A62) is the difference of measures μ_n and μ.

Proof of Theorem 5. The Lagrangian of problem (15) is

$$L(\alpha_n, \lambda^n) = \int_{\mathbb{R}^2} \alpha_n^2(W^{(1)}, b)\, \mathrm{d}\mu_n(W^{(1)}, b) + \sum_{j=1}^{M} \lambda_j^n(g_n(x_j, \alpha_n) - y_j) \tag{A63}$$

The optimal condition is $\nabla_{\alpha_n} L = 0$, which means

$$\nabla_{\alpha_n} L = 2\alpha_n(W^{(1)}, b) + \sum_{j=1}^{M} \lambda_j^n[W^{(1)}x_j + b]_+ = 0 \text{ when } (W^{(1)}, b) = (W_i^{(1)}, b_i),\ i = 1, \cdots, k. \tag{A64}$$

Then

$$\overline{\alpha}_n(W^{(1)}, b) = -\frac{1}{2}\sum_{j=1}^{M} \lambda_j^n[W^{(1)}x_j + b]_+ \text{ when } (W^{(1)}, b) = (W_i^{(1)}, b_i),\ i = 1, \cdots, k. \tag{A65}$$

Since only function values on $(W_i^{(1)}, b_i)_{i=1}^M$ are really taken into account in problem (15), we can let

$$\overline{\alpha}_n(W^{(1)}, b) = -\frac{1}{2}\sum_{j=1}^{M} \lambda_j^n [W^{(1)} x_j + b]_+ \quad \forall (W^{(1)}, b) \in \mathbb{R}^2 \tag{A66}$$

without changing $\int_{\mathbb{R}^2} \overline{\alpha}_n^2(W^{(1)}, b)\, \mathrm{d}\mu_n(W^{(1)}, b)$ and $g_n(x, \overline{\alpha}_n)$.

Here $\lambda_j^n, j = 1, \cdots, M$ are chosen to make $g_n(x_i, \overline{\alpha}_n) = y_i, i = 1, \cdots, M$. It means that

$$-\frac{1}{2}\sum_{j=1}^{M} \lambda_j^n \int_{\mathbb{R}^2} [W^{(1)} x_j + b]_+ [W^{(1)} x_i + b]_+ \, \mathrm{d}\mu_n(W^{(1)}, b) = y_i, i = 1, \ldots, M. \tag{A67}$$

Similarly the Lagrangian of problem (A62) is

$$\widetilde{L}(\alpha, \lambda) = \int_{\mathbb{R}^2} \alpha^2(W^{(1)}, b)\, \mathrm{d}\mu(W^{(1)}, b) + \sum_{j=1}^{M} \lambda_j (g(x_j, \alpha) - y_j). \tag{A68}$$

The optimal condition is $\nabla_\alpha \widetilde{L} = 0$, which means

$$\nabla_\alpha \widetilde{L} = 2\alpha(W^{(1)}, b) + \sum_{j=1}^{M} \lambda_j [W^{(1)} x_j + b]_+ = 0 \quad \forall (W^{(1)}, b) \in \mathbb{R}^2. \tag{A69}$$

Then

$$\overline{\alpha}(W^{(1)}, b) = -\frac{1}{2}\sum_{j=1}^{M} \lambda_j [W^{(1)} x_j + b]_+ \quad \forall (W^{(1)}, b) \in \mathbb{R}^2. \tag{A70}$$

Here $\lambda_j, j = 1, \ldots, M$ are chosen to make $g(x, \alpha) = y_i, i = 1, \ldots, M$. It means that

$$-\frac{1}{2}\sum_{j=1}^{M} \lambda_j \int_{\mathbb{R}^2} [W^{(1)} x_j + b]_+ [W^{(1)} x_i + b]_+ \, \mathrm{d}\mu(W^{(1)}, b) = y_i, i = 1, \ldots, M. \tag{A71}$$

Compare (A67) and (A71). Since the number of samples is finite, x_i is also bounded. Then by the assumption that $\mathcal{W}$ and $\mathcal{B}$ have finite fourth moments, we have that $[W^{(1)} x_j + b]_+ [W^{(1)} x_i + b]_+$ has finite variance. According to central limit theorem, as $n \to \infty$, $\int_{\mathbb{R}^2} [W^{(1)} x_j + b]_+ [W^{(1)} x_i + b]_+ \, \mathrm{d}\mu_n(W^{(1)}, b)$ tends to Gaussian distribution of variance $O(n^{-1})$. Then

$$\begin{aligned} &\Big| \int_{\mathbb{R}^2} [W^{(1)} x_j + b]_+ [W^{(1)} x_i + b]_+ \, \mathrm{d}\mu_n(W^{(1)}, b) - \int_{\mathbb{R}^2} [W^{(1)} x_j + b]_+ [W^{(1)} x_i + b]_+ \, \mathrm{d}\mu(W^{(1)}, b) \Big| \\ &= O(n^{-1/2}) \end{aligned} \tag{A72}$$

$\forall i = 1, \cdots, M, \ \forall j = 1, \ldots, M$ with high probability. Since (A67) and (A71) are systems of linear equations and coefficients of (A67) converge to coefficients of (A71) at the rate of $O(n^{-1/2})$, then

$$|\lambda_j^n - \lambda_j| = O(n^{-1/2}), \quad j = 1, \ldots, M. \tag{A73}$$

Compare (A66) and (A70). Given $(W^{(1)}, b)$, we have

$$|\overline{\alpha}_n(W^{(1)}, b) - \overline{\alpha}(W^{(1)}, b)| = O(n^{-1/2}) \tag{A74}$$

Next we want to prove that $\sup_{x \in [-L, L]} |g_n(x, \overline{\alpha}_n) - g(x, \overline{\alpha})| = O(n^{-1/2})$. Firstly, we prove that $\sup_{x \in [-L, L]} |g_n(x, \overline{\alpha}) - g(x, \overline{\alpha})| = O(n^{-1/2})$. Since

$$\begin{aligned} g_n(x, \overline{\alpha}) &= \int_{\mathbb{R}^2} \overline{\alpha}(W^{(1)}, b) [W^{(1)} x + b]_+ \, \mathrm{d}\mu_n(W^{(1)}, b) \\ g(x, \overline{\alpha}) &= \int_{\mathbb{R}^2} \overline{\alpha}(W^{(1)}, b) [W^{(1)} x + b]_+ \, \mathrm{d}\mu(W^{(1)}, b) \end{aligned} \tag{A75}$$

So

$$\begin{aligned}\mathbb{E}g_n(x,\overline{\alpha}) &= g(x,\overline{\alpha})\\ \operatorname{Var} g_n(x,\overline{\alpha}) &= \frac{1}{n}\int_{\mathbb{R}^2}[\overline{\alpha}(W^{(1)},b)[W^{(1)}x+b]_+ - g(x,\overline{\alpha})]^2\,\mathrm{d}\mu(W^{(1)},b).\end{aligned} \tag{A76}$$

Here the expectation and the variance are with respect to $(W_i^{(1)},b_i)_{i=1}^n$. According to (A70) and the assumption that $\mathcal{W}$ and $\mathcal{B}$ have finite fourth moments, the integral in (A76) is bounded on $[-L,L]$. So $\sup_{x\in[-L,L]}\operatorname{Var} g_n(x,\overline{\alpha}) = O(n^{-1})$. According to central limit theorem, as $n\to\infty$, $g_n(x,\overline{\alpha})$ tends to Gaussian distribution of variance $O(n^{-1})$ for any $x\in[-L,L]$. Then $|g_n(x,\overline{\alpha})-g(x,\overline{\alpha})| = O(n^{-1/2})$ pointwise on $[-L,L]$ with high probability. Then we only need to prove that the sequence of functions $\{g_n(x,\overline{\alpha})\}_{n=1}^\infty$ is uniformly equicontinuous. Actually, $\forall x_1,x_2\in[-L,L]$

$$\begin{aligned}&|g_n(x_1,\overline{\alpha})-g_n(x_2,\overline{\alpha})|\\ \le&\int_{\mathbb{R}^2}\left|\overline{\alpha}(W^{(1)},b)[W^{(1)}x_1+b]_+ - \overline{\alpha}(W^{(1)},b)[W^{(1)}x_2+b]_+\right|\,\mathrm{d}\mu_n(W^{(1)},b)\\ \le&\int_{\mathbb{R}^2}\left|\overline{\alpha}(W^{(1)},b)\right|\left|W_i^{(1)}\right|\,|x_1-x_2|\,\mathrm{d}\mu_n(W^{(1)},b)\\ \le&\int_{\mathbb{R}^2}\left|\overline{\alpha}(W^{(1)},b)\right|\left|W_i^{(1)}\right|\,\mathrm{d}\mu_n(W^{(1)},b)\,|x_1-x_2|\,.\end{aligned} \tag{A77}$$

Because $\int_{\mathbb{R}^2}\left|\overline{\alpha}(W^{(1)},b)\right|\left|W_i^{(1)}\right|\,\mathrm{d}\mu_n(W^{(1)},b)\to\int_{\mathbb{R}^2}\left|\overline{\alpha}(W^{(1)},b)\right|\left|W_i^{(1)}\right|\,\mathrm{d}\mu(W^{(1)},b)$ with probability 1 according to the law of large numbers. So $\int_{\mathbb{R}^2}\left|\overline{\alpha}(W^{(1)},b)\right|\left|W_i^{(1)}\right|\,\mathrm{d}\mu_n(W^{(1)},b)$ is bounded and the bound is independent of n. So $\{g_n(x,\overline{\alpha})\}_{n=1}^\infty$ is uniformly equicontinuous. Then by the argument similar to Arzela-Ascoli theorem, with high probability,

$$\sup_{x\in[-L,L]}|g_n(x,\overline{\alpha})-g(x,\overline{\alpha})| = O(n^{-1/2}). \tag{A78}$$

Finally, we prove that $\sup_{x\in[-L,L]}|g_n(x,\overline{\alpha}_n)-g_n(x,\overline{\alpha})| = O(n^{-1/2})$. Since $\forall x\in[-L,L]$

$$\begin{aligned}&|g_n(x,\overline{\alpha}_n)-g_n(x,\overline{\alpha})|\\ \le&\int_{\mathbb{R}^2}\left|\overline{\alpha}_n(W^{(1)},b)[W^{(1)}x+b]_+ - \overline{\alpha}(W^{(1)},b)[W^{(1)}x+b]_+\right|\,\mathrm{d}\mu_n(W^{(1)},b)\\ \le&\int_{\mathbb{R}^2}\left|\overline{\alpha}_n(W^{(1)},b)-\overline{\alpha}(W^{(1)},b)\right|[W^{(1)}x+b]_+\,\mathrm{d}\mu_n(W^{(1)},b)\\ \le&\int_{\mathbb{R}^2}\left|-\frac{1}{2}\sum_{j=1}^M(\lambda_j^n-\lambda_j)[W^{(1)}x_j+b]_+\right|[W^{(1)}x+b]_+\,\mathrm{d}\mu_n(W^{(1)},b)\\ \le&\frac{1}{2}\sum_{j=1}^M|\lambda_j^n-\lambda_j|\int_{\mathbb{R}^2}[W^{(1)}x_j+b]_+[W^{(1)}x+b]_+\,\mathrm{d}\mu_n(W^{(1)},b)\\ \le&\frac{1}{2}\left(\max_{x\in[-L,L]}\int_{\mathbb{R}^2}[W^{(1)}x_j+b]_+[W^{(1)}x+b]_+\,\mathrm{d}\mu_n(W^{(1)},b)\right)\sum_{j=1}^M|\lambda_j^n-\lambda_j|.\end{aligned} \tag{A79}$$

Because $[-L,L]$ is compact and $\int_{\mathbb{R}^2}[W^{(1)}x_j+b]_+[W^{(1)}x+b]_+\,\mathrm{d}\mu_n(W^{(1)},b)$ converges according to the law of large numbers, $\max_{x\in[-L,L]}\int_{\mathbb{R}^2}[W^{(1)}x_j+b]_+[W^{(1)}x+b]_+\,\mathrm{d}\mu_n(W^{(1)},b)$ is a finite number independent of n. Then according to (A73), $\max_{(W^i,b)\in\operatorname{supp}(\mu)}\left|\overline{\alpha}_n(W^{(1)},b)-\overline{\alpha}(W^{(1)},b)\right|\to 0$. Then

$$\sup_{x\in[-L,L]}|g_n(x,\overline{\alpha}_n)-g_n(x,\overline{\alpha})| = O(n^{-1/2}). \tag{A80}$$

Combined with (A78), we have

$$\sup_{x\in[-L,L]}|g_n(x,\overline{\alpha}_n)-g(x,\overline{\alpha})| = O(n^{-1/2}). \tag{A81}$$

This concludes the proof. □

I Proof of Theorem 6

The detailed calculation of (17) for the second derivative g'' is as follows:

$$\begin{aligned} g''(x,\gamma) &= \int_{\mathbb{R}^2} \gamma(W^{(1)},c)\left|W^{(1)}\right| \delta(x-c)\,\mathrm{d}\nu(W^{(1)},c) \\ &= \int_{\mathrm{supp}(\nu_{\mathcal{C}})} \left(\int_{\mathbb{R}} \gamma(W^{(1)},c)\left|W^{(1)}\right|\,\mathrm{d}\nu_{\mathcal{W}|\mathcal{C}=c}(W^{(1)})\right)\delta(x-c)\,\mathrm{d}\nu_{\mathcal{C}}(c) \\ &= \int_{\mathrm{supp}(\nu_{\mathcal{C}})} \left(\int_{\mathbb{R}} \gamma(W^{(1)},c)\left|W^{(1)}\right|\,\mathrm{d}\nu_{\mathcal{W}|\mathcal{C}=c}(W^{(1)})\right)\delta(x-c)p_{\mathcal{C}}(c)\mathrm{d}c \\ &= p_{\mathcal{C}}(x)\int_{\mathbb{R}} \gamma(W^{(1)},x)\left|W^{(1)}\right|\,\mathrm{d}\nu_{\mathcal{W}|\mathcal{C}=x}(W^{(1)}). \end{aligned} \tag{A82}$$

Proof of Theorem 6. First, if $x \notin \mathrm{supp}(\zeta)$, similar to (17), we have

$$\begin{aligned} g(x,(\overline{\gamma},\overline{u},\overline{v})) &= p_{\mathcal{C}}(x)\int_{\mathbb{R}} \gamma(W^{(1)},x)\left|W^{(1)}\right|\,\mathrm{d}\nu_{\mathcal{W}|\mathcal{C}=x}(W^{(1)}) \\ &= 0. \end{aligned} \tag{A83}$$

Next, we prove that $g(x,(\overline{\gamma},\overline{u},\overline{v}))$ restricted on $\mathrm{supp}(\zeta)$ is the solution of the following problem:

$$\begin{aligned} \min_{h\in C^2(\mathrm{supp}(\zeta))} \quad & \int_{\mathrm{supp}(\zeta)} \frac{(h''(x))^2}{\zeta(x)}\,\mathrm{d}x \\ \text{subject to} \quad & h(x_j)=y_j, \quad j=1,\ldots,m. \end{aligned} \tag{A84}$$

Let $L(f)=\int_{\mathrm{supp}(\zeta)} \frac{(f''(x))^2}{p(x)\mathbb{E}(\mathcal{W}^2|\mathcal{C}=x)}\mathrm{d}x$. Then the functional $L(f)$ is strictly convex on space $\{f\in C^2(\mathbb{R}^2)|f(x_i)=y_i,\ i=1,\ldots,m\}$ when $m\geq 2$. It means that the minimizer of problem (A84) is unique.

Suppose $h(x)$ is the minimizer of problem (A84) and $h(x)$ is different from $g(x,(\overline{\gamma},\overline{u},\overline{v}))$ restricted on $\mathrm{supp}(\zeta)$. Then by uniqueness of the solution,

$$L(h) < L(g(\cdot,(\overline{\gamma},\overline{u},\overline{v}))). \tag{A85}$$

Now our goal is to find out another (γ,u,v) with smaller cost in problem (18). Then $(\overline{\gamma},\overline{u},\overline{v})$ is not the solution of (18), which is a contradiction. We set

$$\gamma(W^{(1)},c)=\frac{h''(c)|W^{(1)}|}{p_{\mathcal{C}}(c)\mathbb{E}(\mathcal{W}^2|\mathcal{C}=c)}, \quad c\in\mathrm{supp}(\zeta). \tag{A86}$$

Then according to (17),

$$\begin{aligned} g''(x,\gamma) &= p(x)\int_{\mathbb{R}} \gamma(W^{(1)},x)\left|W^{(1)}\right|\,\mathrm{d}\nu_{\mathcal{W}|\mathcal{C}=x}(W^{(1)}) \\ &= p(x)\int_{\mathbb{R}} \frac{h''(x)|W^{(1)}|}{p(x)\mathbb{E}(\mathcal{W}^2|\mathcal{C}=x)}\left|W^{(1)}\right|\,\mathrm{d}\nu_{\mathcal{W}|\mathcal{C}=x}(W^{(1)}) \\ &= \frac{h''(x)}{\mathbb{E}(\mathcal{W}^2|\mathcal{C}=x)}\int_{\mathbb{R}} \left|W^{(1)}\right|^2\,\mathrm{d}\nu_{\mathcal{W}|\mathcal{C}=x}(W^{(1)}) \\ &= \frac{h''(x)}{\mathbb{E}(\mathcal{W}^2|\mathcal{C}=x)}\mathbb{E}(\mathcal{W}^2|\mathcal{C}=x) \\ &= h''(x), \quad x\in\mathrm{supp}(\zeta). \end{aligned} \tag{A87}$$

It means that we can find $u,v\in\mathbb{R}$ such that $ux+v+g(x,\gamma)\equiv h(x)$. Then we find out (γ,u,v) such that $g(x,(\gamma,u,v))=ux+v+g(x,\gamma)=h(x)$ on $\mathrm{supp}(\zeta)$. So $g(x_j,(\gamma,u,v))=h(x_j)=y_j$.

It means that (γ, u, v) satisfies the condition in problem (18). Next we compute the cost of (γ, u, v):

$$\begin{aligned}
&\int_{\mathbb{R}^2} \gamma^2(W^{(1)}, c)\, \mathrm{d}\nu(W^{(1)}, c) \\
=&\int_{\mathbb{R}^2} \left(\frac{h''(c)|W^{(1)}|}{p_{\mathcal{C}}(c)\mathbb{E}(\mathcal{W}^2|\mathcal{C}=c)}\right)^2 \mathrm{d}\nu(W^{(1)}, c) \\
=&\int_{\mathrm{supp}(\zeta)} \left(\int_{\mathbb{R}} \left(\frac{h''(c)|W^{(1)}|}{p_{\mathcal{C}}(c)\mathbb{E}(\mathcal{W}^2|\mathcal{C}=c)}\right)^2 \mathrm{d}\nu_{\mathcal{W}|\mathcal{C}=c}(W^{(1)})\right) \mathrm{d}\nu_{\mathcal{C}}(c) \\
=&\int_{\mathrm{supp}(\zeta)} \left(\frac{h''(c)}{p_{\mathcal{C}}(c)\mathbb{E}(\mathcal{W}^2|\mathcal{C}=c)}\right)^2 \left(\int_{\mathbb{R}} |W^{(1)}|^2\, \mathrm{d}\nu_{\mathcal{W}|\mathcal{C}=c}(W^{(1)})\right) p_{\mathcal{C}}(c)\mathrm{d}c \\
=&\int_{\mathrm{supp}(\zeta)} \left(\frac{h''(c)}{p_{\mathcal{C}}(c)\mathbb{E}(\mathcal{W}^2|\mathcal{C}=c)}\right)^2 \left(\int_{\mathbb{R}} |W^{(1)}|^2\, \mathrm{d}\nu_{\mathcal{W}|\mathcal{C}=c}(W^{(1)})\right) p_{\mathcal{C}}(c)\mathrm{d}c \\
=&\int_{\mathrm{supp}(\zeta)} \frac{(h''(c))^2}{p_{\mathcal{C}}(c)\mathbb{E}(\mathcal{W}^2|\mathcal{C}=c)}\mathrm{d}x \\
=&L(h).
\end{aligned} \tag{A88}$$

On the other hand, the cost of $(\overline{\gamma}, \overline{u}, \overline{v})$ is

$$\begin{aligned}
&\int_{\mathbb{R}^2} \overline{\gamma}^2(W^{(1)}, c)\, \mathrm{d}\nu(W^{(1)}, c) \\
=&\int_{\mathrm{supp}(\zeta)} \left(\int_{\mathbb{R}} \overline{\gamma}^2(W^{(1)}, c)\, \mathrm{d}\nu_{\mathcal{W}|\mathcal{C}=c}(W^{(1)})\right) p_{\mathcal{C}}(c)\mathrm{d}c \\
\geq&\int_{\mathrm{supp}(\zeta)} \frac{\left(\int_{\mathbb{R}} \overline{\gamma}(W^{(1)}, c)|W^{(1)}|\, \mathrm{d}\nu_{\mathcal{W}|\mathcal{C}=c}\right)^2}{\int_{\mathbb{R}} |W^{(1)}|^2\, \mathrm{d}\nu_{\mathcal{W}|\mathcal{C}=c}} p_{\mathcal{C}}(c)\mathrm{d}c \quad \text{(Cauchy-Schwarz inequality)} \\
=&\int_{\mathrm{supp}(\zeta)} \frac{\left(g''(c, \overline{\gamma})/p_{\mathcal{C}}(c))\right)^2}{\int_{\mathbb{R}} |W^{(1)}|^2\, \mathrm{d}\nu_{\mathcal{W}|\mathcal{C}=c}} p_{\mathcal{C}}(c)\mathrm{d}c \quad \text{(according to (17))} \\
=&\int_{\mathrm{supp}(\zeta)} \frac{\left(g''(c, \overline{\gamma})\right)^2}{p_{\mathcal{C}}(c)\mathbb{E}(\mathcal{W}^2|\mathcal{C}=c)}\mathrm{d}c \\
=&L(g(\cdot, \overline{\gamma})) \\
=&L(g(\cdot, (\overline{\gamma}, \overline{u}, \overline{v}))) \quad (g(\cdot, (\overline{\gamma}, \overline{u}, \overline{v})) \text{ has the same second derivative as } g(\cdot, \overline{\gamma})).
\end{aligned} \tag{A89}$$

Then

$$\begin{aligned}
\int_{\mathbb{R}^2} \gamma^2(W^{(1)}, c)\, \mathrm{d}\nu(W^{(1)}, c) &= L(h) \quad \text{(according to (A88))} \\
&< L(g(\cdot, (\overline{\gamma}, \overline{u}, \overline{v}))) \quad \text{(according to (A85))} \\
&\leq \int_{\mathbb{R}^2} \overline{\gamma}^2(W^{(1)}, c)\, \mathrm{d}\nu(W^{(1)}, c) \quad \text{(according to (A89))}.
\end{aligned} \tag{A90}$$

It means that the cost of (γ, u, v) is smaller than the cost of $(\overline{\gamma}, \overline{u}, \overline{v})$. So $(\overline{\gamma}, \overline{u}, \overline{v})$ is not the solution of (18), which is a contradiction. So our assumption is wrong. So $h(x) \equiv g(x, (\overline{\gamma}, \overline{u}, \overline{v}))$ on $\mathrm{supp}(\zeta)$, and $g(x, (\overline{\gamma}, \overline{u}, \overline{v}))$ is the solution of problem (A84). In the last step we prove that $g''(x, (\overline{\gamma}, \overline{u}, \overline{v})) = 0$ when $x \notin [\min_i x_i, \max_i x_i]$ and $g(x, (\overline{\gamma}, \overline{u}, \overline{v}))$ restricted on $\mathrm{supp}(\zeta) \cap [\min_i x_i, \max_i x_i]$ is the solution of (21). We only need to prove these statements for $h(x)$, which is the solution of (A84).

Since $|x_i| \in [\min_i x_i, \max_i x_i]$, the function values on $(-\infty, \min_i x_i)$ and $(\max_i x_i, \infty)$ are not related to constraints of problem (21), so $h(x)$ can be replaced by following $\tilde{h}(x)$ which also satisfies the constraints of problem (21):

$$\tilde{h}(x) = \begin{cases} h(x) & x \in [\min_i x_i, \max_i x_i] \\ h'(\min_i x_i)(x - \min_i x_i) + h(\min_i x_i) & x \in (-\infty, \min_i x_i) \\ h'(\max_i x_i)(x - \max_i x_i) + h(\max_i x_i) & x \in (\max_i x_i, \infty). \end{cases} \tag{A91}$$

Then

$$\tilde{h}''(x) = \begin{cases} h''(x) & x \in [\min_i x_i, \max_i x_i] \\ 0 & x \in (-\infty, \min_i x_i) \\ 0 & x \in (\max_i x_i, \infty). \end{cases} \tag{A92}$$

So the cost of $\tilde{h}(x)$ is less than that of $h(x)$. Then the fact $h(x)$ is the minimizer of (A84) tell us that $h(x) \equiv \tilde{h}(x)$. So $h(x)$ should be linear on $(-\infty, \min_i x_i)$ and $(\max_i x_i, \infty)$. Then $h''(x) = 0$ when $x \notin [\min_i x_i, \max_i x_i]$. Let $h(x)|_S$ denote the function $h(x)$ restricted on $S = \operatorname{supp}(\zeta) \cap [\min_i x_i, \max_i x_i]$. Since $h(x)$ is the solution of the problem (A84), we get $h(x)|_S$ is the solution of the problem (21). □

In the case of not using ASI, problem (18) becomes:

$$\begin{aligned} \min_{\gamma \in C(\mathbb{R}^2), u \in \mathbb{R}, v \in \mathbb{R}} \quad & \int_{\mathbb{R}^2} \gamma^2(W^{(1)}, c) \, \mathrm{d}\nu(W^{(1)}, c) \\ \text{subject to} \quad & u x_j + v + \int_{\mathbb{R}^2} \gamma(W^{(1)}, c)[W^{(1)}(x_j - c)]_+ \, \mathrm{d}\nu(W^{(1)}, c) = y_j - f(x_j, \theta_0), \\ & \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad j = 1, \ldots, M. \end{aligned} \tag{A93}$$

Then Theorem 6 without ASI is stated as follows.

Theorem A13 (Theorem 6 without ASI)**.** *Suppose* $(\overline{\gamma}, \overline{u}, \overline{v})$ *is the solution of* (A93)*, and consider the corresponding output function*

$$g(x, (\overline{\gamma}, \overline{u}, \overline{v})) = \overline{u} x + \overline{v} + \int_{\mathbb{R}^2} \overline{\gamma}(W^{(1)}, c)[W^{(1)}(x - c)]_+ \, \mathrm{d}\nu(W^{(1)}, c) + f(x, \theta_0). \tag{A94}$$

Then $g(x, (\overline{\gamma}, \overline{u}, \overline{v}))$ *satisfies* $g''(x, (\overline{\gamma}, \overline{u}, \overline{v})) = f''(x, \theta_0)$ *for* $x \notin S$ *and for* $x \in S$ *it is the solution of the following problem:*

$$\begin{aligned} \min_{h \in C^2(S)} \quad & \int_S \frac{(h''(x) - f''(x, \theta_0))^2}{\zeta(x)} \, \mathrm{d}x \\ \text{subject to} \quad & h(x_j) = y_j, \quad j = 1, \ldots, M. \end{aligned} \tag{A95}$$

J Proof of Proposition 7 and remarks on Proposition 8

Proof of Proposition 7. Let $p_{\mathcal{W},\mathcal{C}}$ and $p_{\mathcal{W},\mathcal{B}}$ denote the joint density functions of $(\mathcal{W}, \mathcal{C})$ and $(\mathcal{W}, \mathcal{B})$, respectively. We have

$$p_{\mathcal{W},\mathcal{C}}(W, C) = \left| \frac{\partial(W, -WC)}{\partial(W, C)} \right| p_{\mathcal{W},\mathcal{B}}(W, -WC) = |W| p_{\mathcal{W},\mathcal{B}}(W, -WC), \tag{A96}$$

and

$$\begin{aligned} \mathbb{E}(W^2 | C = x) p_{\mathcal{C}}(x) &= \int_{\mathbb{R}} W^2 p_{\mathcal{W},\mathcal{C}}(W | \mathcal{C} = x) \, \mathrm{d}W p_{\mathcal{C}}(x) \\ &= \int_{\mathbb{R}} W^2 p_{\mathcal{W},\mathcal{C}}(W, x) \, \mathrm{d}W \\ &= \int_{\mathbb{R}} |W|^3 p_{\mathcal{W},\mathcal{B}}(W, -Wx) \, \mathrm{d}W. \end{aligned} \tag{A97}$$

□

Proof of Proposition 8. The construction is given in the statement of the proposition. □

Remark A14 (Remark to Proposition 8, sampling the initial parameters). The variables $(\mathcal{W}, \mathcal{B})$ can be sampled by first sampling C from $p_{\mathcal{C}}(x) = \frac{1}{Z} \frac{1}{\varrho(x)}$, then independently sampling W from a standard Gaussian distribution and setting $B = -WC$. In this construction, in general $\mathcal{W}$ and $\mathcal{B}$ are not independent.

Intuitively, if we want the output function to be smooth at a certain point x_0, we can let the conditional distribution of $\mathcal{W}$ given $\mathcal{C}$ be concentrated around zero for $\mathcal{C} = x_0$, or we can let the probability density function of $\mathcal{C}$ to be small at $\mathcal{C} = x_0$. Note that $p_{\mathcal{C}}$ is the breakpoint density at initialization. The form of this has been studied for uniform initialization by Sahs et al. (2020). We provide the explicit form of the smoothness penalty function for several types of initialization in Appendix K.

Remark A15 (Remark to Proposition 8, independent initialization). Note that constructing an arbitrary curvature penalty function will necessitate in general a non-independent joint distribution of $\mathcal{W}$ and $\mathcal{B}$. If $\mathcal{W}$ and $\mathcal{B}$ are required to be independent random variables, (A97) gives

$$\zeta(x) = \mathbb{E}(W^2|C = x)p_{\mathcal{C}}(x) = \int_{\mathbb{R}} |W|^3 p_{\mathcal{W}}(W) p_{\mathcal{B}}(-Wx)\, \mathrm{d}W.$$

Given a desired function for the left hand side, we can still try to solve for the parameter densities. This type of integral equation problem has been studied (Nasim, 1973) and one can write a formal solution, although it is not always clear whether it will be a density.

K Proof of Theorem 9

We prove the statement for the three considered types of initialization distributions in turn.

Proof of Theorem 9. Gaussian initialization. Using (A97), we have

$$\begin{aligned}
\mathbb{E}(W^2|C = x)p_{\mathcal{C}}(x) &= \int_{\mathbb{R}} |W|^3 p_{\mathcal{W}}(W) p_{\mathcal{B}}(-Wx)\mathrm{d}W \\
&= \int_{\mathbb{R}} |W|^3 \frac{1}{\sqrt{2\pi}\sigma_w} e^{-\frac{W^2}{2\sigma_w^2}} \frac{1}{\sqrt{2\pi}\sigma_b} e^{-\frac{W^2 x^2}{2\sigma_b^2}} \mathrm{d}W \\
&= \frac{1}{2\pi\sigma_w\sigma_b} \int_{\mathbb{R}} |W|^3 e^{-(\frac{1}{2\sigma_w^2} + \frac{x^2}{2\sigma_b^2})W^2} \mathrm{d}W \\
&= \frac{1}{2\pi\sigma_w\sigma_b} \int_{\mathbb{R}} |W|^3 e^{-(\frac{1}{2\sigma_w^2} + \frac{x^2}{2\sigma_b^2})W^2} \mathrm{d}W
\end{aligned} \tag{A98}$$

Let $\sigma^2 = 1/\left(\frac{1}{\sigma_w^2} + \frac{x^2}{\sigma_b^2}\right)$, then

$$\begin{aligned}
\mathbb{E}(W^2|C = x)p_{\mathcal{C}}(x) &= \frac{\sigma}{\sqrt{2\pi}\sigma_w\sigma_b} \int_{\mathbb{R}} |W|^3 \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{W^2}{2\sigma^2}} \mathrm{d}W \\
&= \frac{\sigma}{\sqrt{2\pi}\sigma_w\sigma_b} \sigma^3 \cdot 2 \cdot \sqrt{\frac{2}{\pi}} \\
&= \frac{\sigma}{\sqrt{2\pi}\sigma_w\sigma_b} \sigma^3 \cdot 2 \cdot \sqrt{\frac{2}{\pi}} \\
&= \frac{2\sigma^4}{\pi\sigma_w\sigma_b} \\
&= \frac{2\sigma_w^3\sigma_b^3}{\pi(\sigma_b^2 + x^2\sigma_w^2)^2}.
\end{aligned} \tag{A99}$$

Then

$$\begin{aligned}
\zeta(x) &= \mathbb{E}(W^2|C = x)p_{\mathcal{C}}(x) \\
&= \frac{2\sigma_w^3\sigma_b^3}{\pi(\sigma_b^2 + x^2\sigma_w^2)^2}.
\end{aligned} \tag{A100}$$

□

Proof of Theorem 9. Binary-uniform initialization. Since $\mathcal{W}$ is either -1 or 1, $\mathbb{E}(\mathcal{W}^2|\mathcal{C} = x) = 1$ for any $x \in \mathrm{supp}(\nu_{\mathcal{C}})$. Since $\mathcal{B} \sim \mathcal{U}(-a_b, a_b)$, it is easy to check $-\mathcal{B}/\mathcal{W} \sim \mathcal{U}(-a_b, a_b)$. So $\zeta(x) = 1/2a_b,\ x \in [-a_b, a_b]$. □

Proof of Theorem 9. Uniform initialization. According to Theorem 1 in Sahs et al. (2020), the density function $p_{\mathcal{C}}(c)$ of $\nu_{\mathcal{C}}$ is

$$p_{\mathcal{C}}(c) = \frac{1}{4a_w a_b}\left(\min\left\{\frac{a_b}{|c|}, a_w\right\}\right)^2, \quad c \in \text{supp}(\nu_{\mathcal{C}}). \tag{A101}$$

When $|c| \leq \frac{a_b}{a_w}$, then $p_{\mathcal{C}}(c) = \frac{1}{4a_w a_b}(a_w)^2$. It means that $p_{\mathcal{C}}(c)$ is constant when $|c| \leq \frac{a_b}{a_w}$.

Let $p_{\mathcal{W},\mathcal{B}}(W^{(1)}, b)$ denote the density function of μ, $p_{\mathcal{W},\mathcal{C}}(W^{(1)}, c)$ denote the density function of ν, so

$$\begin{aligned} p_{\mathcal{W},\mathcal{C}}(W^{(1)}, c) &= p_{\mathcal{W},\mathcal{B}}(W^{(1)}, -cW^{(1)})\frac{\partial b}{\partial c} \\ &= \frac{1}{4a_w a_b}\mathbb{1}_{W^{(1)}\in[-a_w,a_w]} \cdot \mathbb{1}_{-cW^{(1)}\in[-a_b,a_b]} \cdot (-W^{(1)}) \end{aligned} \tag{A102}$$

Here $\mathbb{1}_a$ is the indicator function which equals to 1 when condition a is true, and 0 otherwise. Then density function $p_{\mathcal{W}|\mathcal{C}}(W^{(1)}|c)$ of the conditional distribution $\nu_{\mathcal{W}|\mathcal{C}=c}$ is

$$\begin{aligned} p_{\mathcal{W}|\mathcal{C}}(W^{(1)}|c) &= \frac{p_{\mathcal{W},\mathcal{C}}(W^{(1)}, c)}{p_{\mathcal{C}}(c)} \\ &= \frac{\frac{1}{4a_w a_b}\mathbb{1}_{W^{(1)}\in[-a_w,a_w]} \cdot \mathbb{1}_{-cW^{(1)}\in[-a_b,a_b]} \cdot (-W^{(1)})}{p_{\mathcal{C}}(c)} \end{aligned} \tag{A103}$$

When $|c| \leq \frac{a_b}{a_w}$, $|-cW^{(1)}| \leq \frac{a_b}{a_w}a_w = a_b$. So $-cW^{(1)} \in [-a_b, a_b]$ is true and $\mathbb{1}_{-cW^{(1)}\in[-a_b,a_b]} = 1$. Combined with the fact that $p_{\mathcal{C}}(c)$ is constant when $|c| \leq \frac{a_b}{a_w}$, we have $p_{\mathcal{W}|\mathcal{C}}(W^{(1)}|c)$ is independent of c when $|c| \leq \frac{a_b}{a_w}$. So $\mathbb{E}(\mathcal{W}^2|\mathcal{C} = c)$ is constant when $|c| \leq \frac{a_b}{a_w}$. Since $\frac{a_b}{a_w} \geq L$, $\mathbb{E}(\mathcal{W}^2|\mathcal{C} = c)$ and $p_{\mathcal{C}}(c)$ are constant when $c \in [-L, L]$. Then $\zeta(x) = \mathbb{E}(W^2|C = x)p_{\mathcal{C}}(x)$ is constant when $c \in [-L, L]$. □

L Equivalence of our characterization and NTK norm minimization

In this section we demonstrate that NTK norm minimization (Zhang et al., 2019), which characterizes the implicit bias of training a linearized model by gradient descent, is equivalent to our characterization in Section 4. Following Jacot et al. (2018), Zhang et al. (2019) show that gradient descent can be regarded as a kernel gradient descent in function space, whereby the kernel is given by the NTK. Then for a linearized model, gradient descent finds the global minimum that is closest to the initial output function in the corresponding reproducing kernel Hilbert space (RKHS). Let $\tilde{\Theta}_n$ be the empirical neural tangent kernel of training only the output layer, i.e.

$$\begin{aligned} \tilde{\Theta}_n(x_1, x_2) &= \frac{1}{n}\nabla_{W^{(2)}} f(x_1, \theta_0)\nabla_{W^{(2)}} f(x_2, \theta_0)^T \\ &= \frac{1}{n}\sum_{i=1}^{n} \nabla_{W_i^{(2)}} f(x_1, \theta_0)\nabla_{W_i^{(2)}} f(x_2, \theta_0) \\ &= \frac{1}{n}\sum_{i=1}^{n} [W_i^{(1)}x_1 + b_i^{(1)}]_+[W_i^{(1)}x_2 + b_i^{(1)}]_+. \end{aligned} \tag{A104}$$

As $n \to \infty$, $\tilde{\Theta}_n \to \tilde{\Theta}$, where

$$\tilde{\Theta}(x_1, x_2) = \int_{\mathbb{R}^2} [W^{(1)}x_1 + b^{(1)}]_+[W^{(1)}x_2 + b^{(1)}]_+ \, \mathrm{d}\mu(W^{(1)}, b). \tag{A105}$$

Equivalently, using the notation in Section 4.2, we have

$$\tilde{\Theta}(x_1, x_2) = \int_{\mathbb{R}^2} [W^{(1)}(x_1 - c)]_+[W^{(1)}(x_2 - c)]_+ \, \mathrm{d}\nu(W^{(1)}, c). \tag{A106}$$

Next, Zhang et al. (2019) construct a RKHS $\mathcal{H}_{\tilde{\Theta}}(S)$ by kernel $\tilde{\Theta}$, and the inner product of the RKHS is denoted by $\langle \cdot, \cdot \rangle_{\tilde{\Theta}}$. Then $\mathcal{H}_{\tilde{\Theta}}(S)$ satisfies:

$$\text{(i)} \quad \forall x \in S, \tilde{\Theta}(\cdot, x) \in \mathcal{H}_{\tilde{\Theta}}(S); \tag{A107}$$

$$\text{(ii)} \quad \forall x \in S, \forall f \in \mathcal{H}_{\tilde{\Theta}}, \langle f(\cdot), \tilde{\Theta}(\cdot, x) \rangle_{\tilde{\Theta}} = f(x); \tag{A108}$$

$$\text{(iii)} \quad \forall x, y \in S, \langle \tilde{\Theta}(\cdot, x), \tilde{\Theta}(\cdot, y) \rangle_{\tilde{\Theta}} = \tilde{\Theta}(x, y). \tag{A109}$$

Here the domain is $S = \operatorname{supp}(\zeta) \cap [\min_i x_i, \max_i x_i]$, which is the same as in Theorem 1 and Theorem 6. Using the reproducing kernel Hilbert space, Zhang et al. (2019) prove that $f^{\text{lin}}(x, \widetilde{\omega}_\infty)$ (defined in Section 3.3) is the solution of the following optimization problem:

$$\min_{g \in \mathcal{H}_{\tilde{\Theta}}(S)} \|g\|_{\tilde{\Theta}_n} \quad \text{s.t. } g(x_j) = y_j,\ j = 1, \ldots, M. \tag{A110}$$

As the width n tends to infinity, the above optimization problem becomes

$$\min_{g \in \mathcal{H}_{\tilde{\Theta}}(S)} \|g\|_{\tilde{\Theta}} \quad \text{s.t. } g(x_j) = y_j,\ j = 1, \ldots, M. \tag{A111}$$

In Section 4, we show that $f^{\text{lin}}(x, \widetilde{\omega}_\infty)$ is the solution of the optimization problem (15) in function space. As width n tends to infinity, the optimization problem (15) becomes (A62), which we repeat below:

$$\begin{aligned} \min_{\alpha \in C(\mathbb{R}^2)} \quad & \int_{\mathbb{R}^2} \alpha^2(W^{(1)}, b)\, \mathrm{d}\mu(W^{(1)}, b) \\ \text{subject to} \quad & \int_{\mathbb{R}^2} \alpha(W^{(1)}, b)[W^{(1)} x_j + b]_+ \, \mathrm{d}\mu(W^{(1)}, b) = y_j, \quad j = 1, \ldots, M. \end{aligned} \tag{A112}$$

Since optimization problems (A111) and (A112) both characterize the implicit bias of training a linearized model by gradient descent, they must have the same solution in function space. We express this formally in the following theorem:

Theorem A16 (Equivalence of our variational problem and NTK norm minimization)**.** *Assume that optimization problems* (A111) *and* (A112) *are both feasible. Suppose $\overline{\alpha}$ is the solution of* (A112)*, and consider the corresponding output function:*

$$\overline{g}(x) = \int_{\mathbb{R}^2} \overline{\alpha}(W^{(1)}, b)[W^{(1)} x + b]_+ \, \mathrm{d}\mu(W^{(1)}, b). \tag{A113}$$

Then $\overline{g}(x)$ restricted on S is the solution of the optimization problem (A111)*.*

Next, we give a standalone proof of this theorem using the property of kernel norm. The proof gives us an idea of what the kernel norm actually looks like.

Proof of Theorem A16. Since $\overline{\alpha}(W^{(1)}, b)$ is the solution of (A112), according to (A70) in the proof of Theorem 5,

$$\overline{\alpha}(W^{(1)}, b) = -\frac{1}{2} \sum_{j=1}^{M} \lambda_j [W^{(1)} x_j + b]_+ \quad \forall (W^{(1)}, b) \in \mathbb{R}^2 \tag{A114}$$

for some constants $\lambda_j, j = 1, \ldots, M$. Then we write $\overline{\alpha}(W^{(1)}, b)$ in the following form:

$$\overline{\alpha}(W^{(1)}, b) = \int_S h(x)[W^{(1)} x + b]_+ \mathrm{d}x, \tag{A115}$$

where $h(x)$ can be a combination of Dirac delta functions. Then substitute (A115) into the expression of $\overline{g}(x)$ (A113) to obtain

$$\begin{aligned} \overline{g}(x) &= \int_{\mathbb{R}^2 \times S} h(\tilde{x})[W^{(1)} \tilde{x} + b]_+ [W^{(1)} x + b]_+ \, \mathrm{d}\mu(W^{(1)}, b) \mathrm{d}\tilde{x} \\ &= \int_S h(\tilde{x}) \tilde{\Theta}(x, \tilde{x}) \mathrm{d}\tilde{x}, \end{aligned} \tag{A116}$$

where we use the expression of the NTK in equation (A105). Then

$$\begin{aligned}\langle g(x), g(x)\rangle_{\tilde{\Theta}} &= \langle g(x), \int_S h(\tilde{x})\tilde{\Theta}(x,\tilde{x})\mathrm{d}\tilde{x}\rangle_{\tilde{\Theta}}\mathrm{d}\tilde{x} \\ &= \int_S h(\tilde{x})\langle g(x), \tilde{\Theta}(x,\tilde{x})\rangle_{\tilde{\Theta}}\mathrm{d}\tilde{x} \\ &= \int_S h(\tilde{x})g(\tilde{x})\mathrm{d}\tilde{x} \quad \text{(here we use the property of RKHS norm (A108))} \\ &= \int_{S\times S} h(\tilde{x})h(\bar{x})\tilde{\Theta}(\tilde{x},\bar{x})\mathrm{d}\tilde{x}\mathrm{d}\bar{x} \quad \text{(use (A116))}.\end{aligned} \tag{A117}$$

On the other hand, using (A115), the objective of (A112) becomes

$$\begin{aligned}&\int_{S^2} \overline{\alpha}^2(W^{(1)},b)\ \mathrm{d}\mu(W^{(1)},b) \\ =&\int_{S\times S\times\mathbb{R}^2} h(\tilde{x})[W^{(1)}\tilde{x}+b]_+ h(\bar{x})[W^{(1)}\bar{x}+b]_+\ \mathrm{d}\tilde{x}\mathrm{d}\bar{x}\mathrm{d}\mu(W^{(1)},b) \\ =&\int_{S\times S} h(\tilde{x})h(\bar{x})\int_{\mathbb{R}^2}[W^{(1)}\tilde{x}+b]_+[W^{(1)}\bar{x}+b]_+\mathrm{d}\mu(W^{(1)},b)\ \mathrm{d}\tilde{x}\mathrm{d}\bar{x} \\ =&\int_{S\times S} h(\tilde{x})h(\bar{x})\tilde{\Theta}(\bar{x},\tilde{x})\ \mathrm{d}\tilde{x}\mathrm{d}\bar{x} \quad \text{(use (A105))}.\end{aligned} \tag{A118}$$

Comparing (A117) and (A118), we have that optimization problems (A111) and (A112) are equivalent if $\alpha(W^{(1)},b)$ has the form (A115) and $g(x)$ has the form (A116). Moreover, if every function $g\in\mathcal{H}_{\tilde{\Theta}}(S)$ can be approximated by the shallow network, we can find $\alpha(W^{(1)},b)$ in form of (A115) such that $g(x)$ is expressed in the form of (A116). In this sense we show that optimization problems (A111) and (A112) are equivalent. □

In Section 4.2, we relax the optimization problem (16) to (18) in order to characterize the implicit bias in function space. This relaxation can also be done in the NTK norm minimization setting. It means that we can equivalently relax the problem (A111) to the following problem:

$$\min_{g\in\mathcal{H}_{\tilde{\Theta}}(S),u\in\mathbb{R},v\in\mathbb{R}} \|g-ux-v\|_{\tilde{\Theta}} \quad \text{s.t. } g(x_j)=y_j,\ j=1,\dots,M. \tag{A119}$$

Then the optimization problems (18) and (A119) are equivalent. Theorem 6 shows that (18) and (21) have the same solution on the set $S=\mathrm{supp}(\zeta)\cap[\min_i x_i,\max_i x_i]$. Then we have that optimization problems (A119) and (21) are equivalent, which means that

$$\min_{u\in\mathbb{R},v\in\mathbb{R}} \|g-ux-v\|_{\tilde{\Theta}} = \int_S \frac{(g''(x))^2}{\zeta(x)}\ \mathrm{d}x, \quad \forall g\in\mathcal{H}_{\tilde{\Theta}}(S). \tag{A120}$$

Next, we directly prove the above equation (A120). Given function $g\in\mathcal{H}_{\tilde{\Theta}}(S)$, let $h=\mathrm{argmin}_{h\in\mathcal{H}_{\tilde{\Theta}}(S)}\|h\|_{\tilde{\Theta}}$, s.t. $h=g-ux-v$ for some $u\in\mathbb{R},v\in\mathbb{R}$. Then according to optimality of h, we have $\langle h,x\rangle_{\tilde{\Theta}}=0$ and $\langle h,1\rangle_{\tilde{\Theta}}=0$. Consider the space $W=\{h\in\mathcal{H}_{\tilde{\Theta}}(S): \langle h,x\rangle_{\tilde{\Theta}}=0,\langle h,1\rangle_{\tilde{\Theta}}=0\}$, which is the orthogonal complement of $\mathrm{span}\{1,x\}$. Then h is the projection of g on W. Since $h=g-ux-v$, $h''=g''$. So we can reformulate the equation (A120) which we want to prove in the following theorem:

Theorem A17 (Explicit form of the kernel norm)**.** *The kernel norm on the space $W=\{h\in\mathcal{H}_{\tilde{\Theta}}(S): \langle h,x\rangle_{\tilde{\Theta}}=0,\langle h,1\rangle_{\tilde{\Theta}}=0\}$ is given as follows:*

$$\|h\|_{\tilde{\Theta}}^2 = \int_S \frac{(h''(x))^2}{\zeta(x)}\ \mathrm{d}x, \quad \forall h\in W. \tag{A121}$$

This theorem gives the explicit form of the kernel norm in a subspace of $\mathcal{H}_{\tilde{\Theta}}(S)$. Next we prove the above theorem using the property of kernel norm.

Proof of Theoem A17. Let $\tilde{\Theta}_x(\cdot)=\tilde{\Theta}(\cdot,x)$. We can find the orthogonal projection of $\tilde{\Theta}_x$ on space W, which we denote by $\tilde{\Theta}_{x,W}$. Then we only need to prove that $\langle h,\tilde{\Theta}_{x,W}\rangle_{\tilde{\Theta}}=\int_S\frac{h''(y)\tilde{\Theta}''_{x,W}(y)}{\zeta(y)}\ \mathrm{d}y$ for any $h\in W$ and $x\in S$.

First, $\tilde{\Theta}_{x,W} = \tilde{\Theta}_x - ux - v$ for some constant $u, v \in \mathbb{R}$. Since $h \in W$, $\langle h, 1\rangle_{\tilde{\Theta}} = 0$ and $\langle h, x\rangle_{\tilde{\Theta}} = 0$. Then

$$\begin{aligned}\langle h, \tilde{\Theta}_{x,W}\rangle_{\tilde{\Theta}} &= \langle h, \tilde{\Theta}_x - ux - v\rangle_{\tilde{\Theta}} \\ &= \langle h, \tilde{\Theta}_x\rangle_{\tilde{\Theta}} - u\langle h, x\rangle_{\tilde{\Theta}} - v\langle h, 1\rangle_{\tilde{\Theta}} \\ &= \langle h, \tilde{\Theta}_x\rangle_{\tilde{\Theta}} \\ &= h(x) \quad \text{(use the reproducing property of the kernel (A108)).}\end{aligned} \tag{A122}$$

Next, using the notation from Section 4.2 we have

$$\begin{aligned}\tilde{\Theta}''_{x,W}(y) &= (\tilde{\Theta}_x(y) - uy - v)'' \\ &= \tilde{\Theta}_x(y)'' \\ &= \frac{\partial^2}{\partial y^2}\tilde{\Theta}(x, y) \\ &= \frac{\partial^2}{\partial y^2}\int_{\mathbb{R}^2}[W^{(1)}(x-c)]_+[W^{(1)}(y-c)]_+ \,\mathrm{d}\nu(W^{(1)}, c) \quad \text{(use (A106))} \\ &= \frac{\partial^2}{\partial y^2}\int_{\mathbb{R}^2}(W^{(1)})^2[\mathrm{sign}(W^{(1)})(x-c)]_+[\mathrm{sign}(W^{(1)})(y-c)]_+ \,\mathrm{d}\nu_{\mathcal{W}|\mathcal{C}=c}(W^{(1)})\mathrm{d}\nu_{\mathcal{C}}(c) \\ &= \frac{\partial^2}{\partial y^2}\int_{\mathbb{R}}\big(\mathbb{E}(\mathcal{W}^2\mathbb{1}(\mathcal{W}\geq 0)|\mathcal{C}=c)[x-c]_+[y-c]_+ \\ &\qquad +\mathbb{E}(\mathcal{W}^2\mathbb{1}(\mathcal{W}<0)|\mathcal{C}=c)[c-x]_+[c-y]_+\big)\,p_{\mathcal{C}}(c)\,\mathrm{d}c \\ &= \int_{\mathbb{R}}\Bigg(\mathbb{E}(\mathcal{W}^2\mathbb{1}(\mathcal{W}\geq 0)|\mathcal{C}=c)[x-c]_+\frac{\partial^2}{\partial y^2}[y-c]_+ \\ &\qquad +\mathbb{E}(\mathcal{W}^2\mathbb{1}(\mathcal{W}<0)|\mathcal{C}=c)[c-x]_+\frac{\partial^2}{\partial y^2}[c-y]_+\Bigg)\,p_{\mathcal{C}}(c)\,\mathrm{d}c \\ &= \int_{\mathbb{R}}\big(\mathbb{E}(\mathcal{W}^2\mathbb{1}(\mathcal{W}\geq 0)|\mathcal{C}=c)[x-c]_+\delta(y-c) \\ &\qquad +\mathbb{E}(\mathcal{W}^2\mathbb{1}(\mathcal{W}<0)|\mathcal{C}=c)[c-x]_+\delta(y-c)\big)\,p_{\mathcal{C}}(c)\,\mathrm{d}c \\ &= \big(\mathbb{E}(\mathcal{W}^2\mathbb{1}(\mathcal{W}\geq 0)|\mathcal{C}=y)[x-y]_+ + \mathbb{E}(\mathcal{W}^2\mathbb{1}(\mathcal{W}<0)|\mathcal{C}=y)[y-x]_+\big)\,p_{\mathcal{C}}(y).\end{aligned} \tag{A123}$$

Then

$$\begin{aligned}&\int_S \frac{h''(y)\tilde{\Theta}''_{x,W}(y)}{\zeta(y)}\,\mathrm{d}y \\ &= \int_S \frac{h''(y)\left(\mathbb{E}(\mathcal{W}^2\mathbb{1}(\mathcal{W}\geq 0)|\mathcal{C}=y)[x-y]_+ + \mathbb{E}(\mathcal{W}^2\mathbb{1}(\mathcal{W}<0)|\mathcal{C}=y)[y-x]_+\right)p_{\mathcal{C}}(y)}{\zeta(y)}\,\mathrm{d}y \\ &= \int_S \frac{h''(y)\left(\mathbb{E}(\mathcal{W}^2\mathbb{1}(\mathcal{W}\geq 0)|\mathcal{C}=y)[x-y]_+ + \mathbb{E}(\mathcal{W}^2\mathbb{1}(\mathcal{W}<0)|\mathcal{C}=y)[y-x]_+\right)}{\mathbb{E}(\mathcal{W}^2|\mathcal{C}=y)}\,\mathrm{d}y \\ &= \int_S \frac{\mathbb{E}(\mathcal{W}^2\mathbb{1}(\mathcal{W}\geq 0)|\mathcal{C}=y)}{\mathbb{E}(\mathcal{W}^2|\mathcal{C}=y)}h''(y)[x-y]_+ + \frac{\mathbb{E}(\mathcal{W}^2\mathbb{1}(\mathcal{W}<0)|\mathcal{C}=y)}{\mathbb{E}(\mathcal{W}^2|\mathcal{C}=y)}h''(y)[y-x]_+\,\mathrm{d}y.\end{aligned} \tag{A124}$$

Now we regard $\int_S \frac{h''(y)\tilde{\Theta}''_{x,W}(y)}{\zeta(y)}\,\mathrm{d}y$ as a function of x, then

$$\begin{aligned}
&\frac{\partial^2}{\partial x^2}\int_S \frac{h''(y)\tilde{\Theta}''_{x,W}(y)}{\zeta(y)}\,\mathrm{d}y\\
=&\frac{\partial^2}{\partial x^2}\int_S \frac{\mathbb{E}(\mathcal{W}^2\mathbb{1}(\mathcal{W}\geq 0)|\mathcal{C}=y)}{\mathbb{E}(\mathcal{W}^2|\mathcal{C}=y)}h''(y)[x-y]_+ + \frac{\mathbb{E}(\mathcal{W}^2\mathbb{1}(\mathcal{W}<0)|\mathcal{C}=y)}{\mathbb{E}(\mathcal{W}^2|\mathcal{C}=y)}h''(y)[y-x]_+\,\mathrm{d}y\\
=&\int_S \frac{\mathbb{E}(\mathcal{W}^2\mathbb{1}(\mathcal{W}\geq 0)|\mathcal{C}=y)}{\mathbb{E}(\mathcal{W}^2|\mathcal{C}=y)}h''(y)\delta(x-y) + \frac{\mathbb{E}(\mathcal{W}^2\mathbb{1}(\mathcal{W}<0)|\mathcal{C}=y)}{\mathbb{E}(\mathcal{W}^2|\mathcal{C}=y)}h''(y)\delta(y-x)\,\mathrm{d}y\\
=&\frac{\mathbb{E}(\mathcal{W}^2\mathbb{1}(\mathcal{W}\geq 0)|\mathcal{C}=x)}{\mathbb{E}(\mathcal{W}^2|\mathcal{C}=x)}h''(x) + \frac{\mathbb{E}(\mathcal{W}^2\mathbb{1}(\mathcal{W}<0)|\mathcal{C}=x)}{\mathbb{E}(\mathcal{W}^2|\mathcal{C}=x)}h''(x)\\
=&h''(x).
\end{aligned} \tag{A125}$$

From the definition of the space W, we see that the second derivative uniquely determines the element in W. Since $h\in W$, in order to show that $\int_S \frac{h''(y)\tilde{\Theta}''_{x,W}(y)}{\zeta(y)}\,\mathrm{d}y = h(x)$, we only need to show $\int_S \frac{h''(y)\tilde{\Theta}''_{x,W}(y)}{\zeta(y)}\,\mathrm{d}y \in W$, i.e. $\langle\int_S \frac{h''(y)\tilde{\Theta}''_{x,W}(y)}{\zeta(y)}\,\mathrm{d}y, 1\rangle_{\tilde{\Theta}} = 0$ and $\langle\int_S \frac{h''(y)\tilde{\Theta}''_{x,W}(y)}{\zeta(y)}\,\mathrm{d}y, x\rangle_{\tilde{\Theta}} = 0$. Then

$$\begin{aligned}
\langle\int_S \frac{h''(y)\tilde{\Theta}''_{x,W}(y)}{\zeta(y)}\,\mathrm{d}y, 1\rangle_{\tilde{\Theta}} =&\langle\int_S \frac{h''(y)\frac{\partial^2}{\partial y^2}\tilde{\Theta}(x,y)}{\zeta(y)}\,\mathrm{d}y, 1\rangle_{\tilde{\Theta}}\\
=&\langle\int_S \frac{h''(y)\lim_{h\to 0}\frac{\tilde{\Theta}(x,y+h)-2\tilde{\Theta}(x,y)+\tilde{\Theta}(x,y-h)}{h^2}}{\zeta(y)}\,\mathrm{d}y, 1\rangle_{\tilde{\Theta}}\\
=&\lim_{h\to 0}\langle\int_S \frac{h''(y)\frac{\tilde{\Theta}(x,y+h)-2\tilde{\Theta}(x,y)+\tilde{\Theta}(x,y-h)}{h^2}}{\zeta(y)}\,\mathrm{d}y, 1\rangle_{\tilde{\Theta}}\\
=&\lim_{h\to 0}\int_S \frac{h''(y)\frac{\langle\tilde{\Theta}(x,y+h),1\rangle_{\tilde{\Theta}}-2\langle\tilde{\Theta}(x,y),1\rangle_{\tilde{\Theta}}+\langle\tilde{\Theta}(x,y-h),1\rangle_{\tilde{\Theta}}}{h^2}}{\zeta(y)}\,\mathrm{d}y\\
=&\lim_{h\to 0}\int_S \frac{h''(y)\frac{y+h-2y+y-h}{h^2}}{\zeta(y)}\,\mathrm{d}y\\
=&0.
\end{aligned} \tag{A126}$$

Similarly we can show that $\langle\int_S \frac{h''(y)\tilde{\Theta}''_{x,W}(y)}{\zeta(y)}\,\mathrm{d}y, x\rangle_{\tilde{\Theta}} = 0$. This concludes the proof. □

M Gradient descent trajectory and trajectory of smoothing splines

In the following we discuss the relation between the trajectory of functions obtained by gradient descent training of a neural network and a trajectory of solutions to the variational problem with the data fitting constraints replaced by a MSE for decreasing smoothness regularization strength. This Lagrange version of the variational problem is solved by so-called smoothing splines. Smoothing splines have been studied intensively in the literature and in particular they can be written explicitly. We give the explicit form of the solution for the trajectory in the context of our discussion.

M.1 Regularized regression and early stopping

Bishop (1995) shows that for linear regression with quadratic loss, early stopping and L_2 regularization lead to similar solutions. Let us recall some details of his analysis, before proceeding with our particular setting. He considers the loss function $E(\mathbf{w}) = \|X\mathbf{w}-\mathbf{y}\|_2^2$, where $X = [\mathbf{x}_1,\ldots,\mathbf{x}_M]^T$ is the matrix of training inputs, $\mathbf{y} = [y_1,\ldots,y_M]^T$ is the vector of training outputs, and $\mathbf{w}$ is the weight vector of the linear model. Next the loss function can be written in the form of a quadratic

function:

$$\begin{aligned} E(W) &= \|X\mathbf{w} - \mathbf{y}\|_2^2 \\ &= \mathbf{w}^T X^T X \mathbf{w} - 2\mathbf{y}^T X \mathbf{w} + \mathbf{y}^T \mathbf{y} \\ &= \mathbf{w}^T X^T X \mathbf{w} - 2\mathbf{y}^T X \mathbf{w} + \mathbf{y}^T \mathbf{y} \\ &= \frac{1}{2}(\mathbf{w} - \mathbf{w}^*)^T H (\mathbf{w} - \mathbf{w}^*) + E_0, \end{aligned} \tag{A127}$$

where $H = 2X^T X$, E_0 is the minimum of the loss function, and $\mathbf{w}^*$ is the minimizer. The eigenvalues and eigenvectors of H are as follows:

$$H\mathbf{u}_j = \lambda_j \mathbf{u}_j. \tag{A128}$$

Then expand $\mathbf{w}$ and $\mathbf{w}^*$ in terms of the eigenvectors of H:

$$\mathbf{w} = \sum_j w_j \mathbf{u}_j, \qquad \mathbf{w}^* = \sum_j w_j^* \mathbf{u}_j. \tag{A129}$$

For the L_2 regularized regression problem, consider the regularized loss function $\tilde{E}(\mathbf{w}) = E(\mathbf{w}) + c\|\mathbf{w}\|_2^2$. Denote the minimizer by $\mathbf{w} = \tilde{\mathbf{w}}$ and consider its expansion as $\tilde{\mathbf{w}} = \sum_j \tilde{w}_j \mathbf{u}_j$. Bishop (1995) shows that

$$\tilde{w}_j = \frac{\lambda_j}{\lambda_j + c} w_j^*. \tag{A130}$$

For early stopping, consider the gradient descent on $E(\mathbf{w})$ with zero initial weight vector:

$$\begin{aligned} \mathbf{w}^{(\tau)} &= \mathbf{w}^{(\tau-1)} - \eta \nabla E \\ &= \mathbf{w}^{(\tau-1)} - \eta H(\mathbf{w}^{(\tau-1)} - \mathbf{w}^*), \\ \mathbf{w}^{(0)} &= \mathbf{0}. \end{aligned} \tag{A131}$$

Writing $\mathbf{w}^{(\tau)} = \sum_j w_j^{(\tau)} \mathbf{u}_j$, then

$$w_j^{(\tau)} = (1 - (1 - \eta\lambda_j)^\tau) w_j^*. \tag{A132}$$

Note that $1 - (1 - \eta\lambda_j)^\tau \to 1 - e^{-\eta\tau\lambda_j}$ as $\eta \to 0$. Hence choosing a sufficiently small learning rate, approximately we have

$$w_j^{(\tau)} = (1 - e^{-\eta\tau\lambda_j}) w_j^*. \tag{A133}$$

From (A130) and (A133), Bishop (1995) observes that if c is much larger than λ_j, then the regularized solution has coordinate $\tilde{w}_j$ close to 0, and similarly if $1/(\eta\tau)$ is much larger than λ_j, then the early-stopping solution has coordinate $w_j^{(\tau)}$ close to the initial value 0. We note that analogous observations apply when the regularization term has a reference point different from zero, $c\|\mathbf{w} - \overline{\mathbf{w}}\|_2^2$, and the gradient descent iteration is initialized at a point different from zero, $\mathbf{w}^{(0)} = \overline{\mathbf{w}}$.

Now we want to take a closer look at the trajectories. Consider the following two functions:

$$h_1(x) = \frac{\lambda_j}{\lambda_j + x}, \qquad h_2(x) = 1 - e^{-\lambda_j / x}. \tag{A134}$$

Actually we can verify that $h_1(0) = h_2(0) = 1$ and $\lim_{x\to\infty} \frac{h_1(x)}{h_2(x)} = 1$. It implies that these two functions are close to each other on $[0, \infty)$. Figure A7 shows the plot of functions $h_1(x)$ and $h_2(x)$.

Now we choose the coefficient of regularization $c = \frac{1}{\eta\tau}$. Comparing (A130) and (A133), and using the fact that $h_1(x)$ and $h_2(x)$ are close to each other on $[0, \infty)$, we show that early stopping and L_2 regularization lead to similar solutions across different values of $c = \frac{1}{\eta\tau}$.

Back to our problem, we repeat the gradient descent procedures (10) here:

$$\widetilde{W}_0^{(2)} = \overline{W}^{(2)}, \quad \widetilde{W}_{t+1}^{(2)} = \widetilde{W}_t^{(2)} - \eta \nabla_{W^{(2)}} L^{\text{lin}}(\widetilde{\omega}_t). \tag{A135}$$

It is actually minimizing the following loss function of $W^{(2)} - \overline{W}$:

$$E(W^{(2)} - \overline{W}) = \sum_{j=1}^{M} \left(\sum_{i=1}^{n} (W_i^{(2)} - \overline{W}_i^{(2)}) [W_i^{(1)} x_j + b_i]_+ - (y_j - f(x_j, \theta_0)) \right)^2. \tag{A136}$$

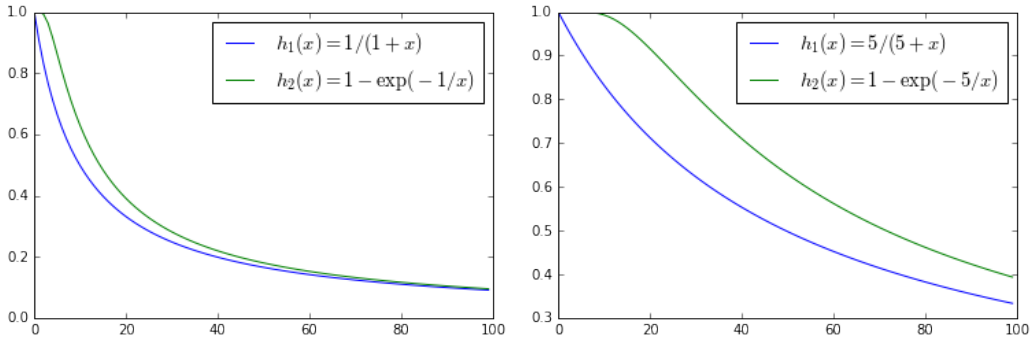

Figure A7: Plot of functions $h_1(x)$ and $h_2(x)$. The left panel plots the two function when $\lambda_j = 1$. The right panel plots the two function when $\lambda_j = 5$.

Here we change the variable from $W^{(2)}$ to $W^{(2)} - \overline{W}$. Then $W_t^{(2)} - \overline{W} = 0$ when $t = 0$, so that gradient descent start from the zero initial weight vector. Since the above model is linear with respect to $W^{(2)} - \overline{W}$, we can apply the above argument about early stopping and L_2 regularization. Suppose that we use learning rate μ_n for the neural network of width n. We show that the solution $\widetilde{W}_t^{(2)}$ at iteration t is close to the minimizer of the following regularized optimization problem:

$$\min_{W^{(2)}} \sum_{j=1}^{M} \left(\sum_{i=1}^{n} (W_i^{(2)} - \overline{W}_i^{(2)})[W_i^{(1)} x_j + b_i]_+ - (y_j - f(x_j, \theta_0)) \right)^2 + c\|W^{(2)} - \overline{W}\|_2^2, \quad \text{(A137)}$$

where $c = \frac{1}{\eta_n t}$. Using the same approach and notation as in Section 4, the optimization problem (A137) is equivalent to

$$\begin{aligned} \min_{\alpha_n \in C(\mathbb{R}^2)} \quad & \sum_{j=1}^{M} \left(\int_{\mathbb{R}^2} \alpha_n(W^{(1)}, b)[W^{(1)} x_j + b]_+ \, \mathrm{d}\mu_n(W^{(1)}, b) - y_j \right)^2 \\ & + \frac{1}{n\eta_n t} \int_{\mathbb{R}^2} \alpha_n^2(W^{(1)}, b) \, \mathrm{d}\mu_n(W^{(1)}, b), \end{aligned} \quad \text{(A138)}$$

where we use the ASI trick (see Appendix C.2). Here (A138) has an extra factor $\frac{1}{n}$ compared to (A137). This is because we define $\alpha_n(W_i^{(1)}, b_i) = n(W_i^{(2)} - \overline{W}_i^{(2)})$. According to Theorem 2, $\eta_n \leq \frac{1}{Kn\sqrt{M}\lambda_{\max}(\hat{\Theta}_n)}$ is sufficient in order to ensure convergence. Then we suppose that $\eta_n = \bar{\eta}/n$, where $\bar{\eta}$ is a constant so that the requirement on the learning rate in Theorem 2 is satisfied. The limit of the optimization problem (A138) as the width n tends to infinity is:

$$\begin{aligned} \min_{\alpha \in C(\mathbb{R}^2)} \quad & \sum_{j=1}^{M} \left(\int_{\mathbb{R}^2} \alpha(W^{(1)}, b)[W^{(1)} x_j + b]_+ \, \mathrm{d}\mu(W^{(1)}, b) - y_j \right)^2 \\ & + \frac{1}{\bar{\eta} t} \int_{\mathbb{R}^2} \alpha^2(W^{(1)}, b) \, \mathrm{d}\mu(W^{(1)}, b). \end{aligned} \quad \text{(A139)}$$

Following the same reasoning of Section 4.2, we relax the optimization problem (A139) to the following one:

$$\begin{aligned} \min_{\alpha \in C(\mathbb{R}^2), u \in \mathbb{R}, v \in \mathbb{R}} \quad & \sum_{j=1}^{M} \left(u x_j + v + \int_{\mathbb{R}^2} \alpha(W^{(1)}, b)[W^{(1)} x_j + b]_+ \, \mathrm{d}\mu(W^{(1)}, b) - y_j \right)^2 \\ & + \frac{1}{\bar{\eta} t} \int_{\mathbb{R}^2} \alpha^2(W^{(1)}, b) \, \mathrm{d}\mu(W^{(1)}, b). \end{aligned} \quad \text{(A140)}$$

Using the same technique and notation as in Theorem 6, we can prove that the solution of (A140) actually solves the following optimization problem:

$$\min_{h \in C^2(S)} \sum_{j=1}^{M} [h(x_j) - y_j]^2 + \frac{1}{\bar{\eta} t} \int_S \frac{(h''(x))^2}{\zeta(x)} \, \mathrm{d}x. \quad \text{(A141)}$$

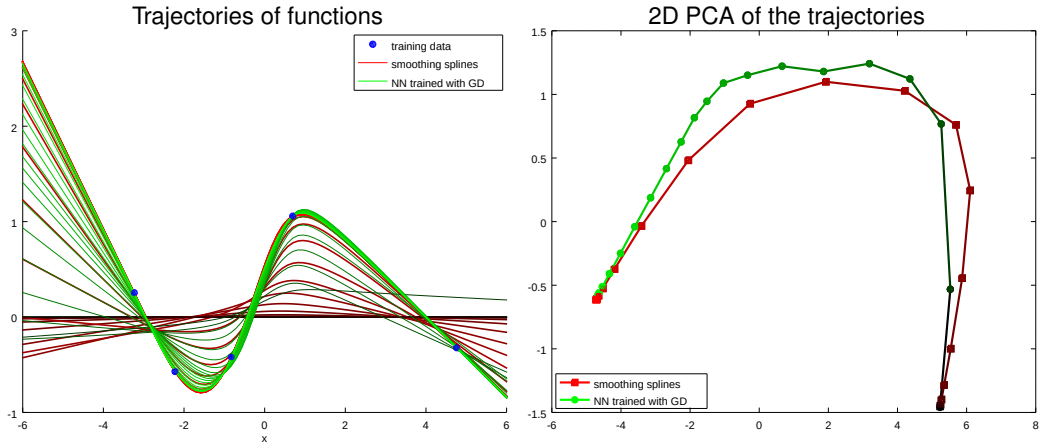

Figure A8: Trajectories of functions obtained by gradient descent training a neural network and by smoothing splines of the training data with decreasing regularization strength (from dark to bright). The left panel plots 20 functions along each trajectory. The right panel shows the same functions in a two dimensional PCA representation. With asymmetric initialization of the network parameters and adjusting the training data by ordinary linear regression, both trajectories start at the zero function. The trajectories are not equivalent, but are close, and both converge to the same (spatially adaptive) cubic spline interpolation of the training data (in the limit of infinite wide networks). Here we used a large network with $n = 2000$ hidden units and Gaussian initialization $\mathcal{W} \sim N(0, 1)$, $\mathcal{B} \sim N(0, 1)$. The results are similar for smaller networks and different initializations.

Then in order to study the trajectory of gradient descent, we can study the optimization problem (A141) with varying t. Figure A8 illustrates smoothing spline and gradient descent trajectories. The solution of (A141) is called spatially adaptive smoothing spline. Here the curvature penalty function is $\frac{1}{\bar{\eta}t}\frac{1}{\zeta(x)}$, with time dependent smoothness regularization coefficient $\frac{1}{\bar{\eta}t}$. Next, we give out the solution of (A141) in the following two cases: (1) uniform case (ζ is constant over domain S); (2) spatially adaptive case (ζ is not constant over domain S).

Remark A18 (Spectral bias). We have thus that the gradient descent optimization trajectory can be described approximately by a trajectory of smoothing splines which gradually relaxes the smoothness regularization (relative to initialization) until perfectly fitting the training data. If the function at initialization is at the zero function, e.g. by ASI, then the regularization is on the function itself. Hence the result provides a theoretical explanation for the spectral bias phenomenon that has been observed by Rahaman et al. (2019). The spectral bias is that lower frequencies are learned first.

M.2 Trajectory of smoothing splines with uniform curvature penalty

Suppose the reciprocal curvature penalty is constant $\zeta(x) \equiv z$ on the domain S. Let $\lambda = \frac{1}{\bar{\eta}tz}$. Then (A141) becomes the following optimization problem:

$$\min_{h \in C^2(S)} \sum_{j=1}^{M} [h(x_j) - y_j]^2 + \lambda \int_S (h''(x))^2 \, \mathrm{d}x. \tag{A142}$$

German (2001) gives the explicit form of the minimizer $\hat{h}$ of (A142), which is called a smoothing spline. The minimizer $\hat{h}$ is a natural cubic spline with knots at the sample points $x_1, \ldots, x_M$. The smoothing spline does not fit the training data exactly, but rather it balances fitting and smoothness. The smoothing parameter $\lambda \geq 0$ controls the trade off between fitting and roughness. The values of the smoothing spline at the knots can be obtained as

$$(\hat{h}(x_1), \ldots, \hat{h}(x_M))^\top = (I + \lambda A)^{-1} Y. \tag{A143}$$

The matrix A has entries $A_{ij} = \int_S h_i''(x) h_j''(x) \, \mathrm{d}x$, where h_i are spline basis functions which satisfy $h_i(x_j) = 0$ for $j \neq i$ and $h_i(x_j) = 1$ for $j = i$. German (2001) gives out a rather explicit form of

matrix A, which is an $M \times M$ matrix given by $A = \Delta^T W^{-1} \Delta$. Here Δ is an $(M-2) \times M$ matrix of second differences with elements:

$$\Delta_{ii} = \frac{1}{h_i}, \quad \Delta_{i,i+1} = -\frac{1}{h_i} - \frac{1}{h_{i+1}}, \quad \Delta_{i,i+2} = \frac{1}{h_{i+1}}.$$

And W is an $(M-2) \times (M-2)$ symmetric tri-diagonal matrix with elements:

$$W_{i-1,i} = W_{i,i-1} = \frac{h_i}{6}, \quad W_{i,i} = \frac{h_i + h_{i+1}}{3}, \text{ here } h_i = x_{i+1} - x_i.$$

As $\lambda \to 0$, the smoothing spline converges to the interpolating spline, and as $\lambda \to \infty$, it converges to the linear least squares estimate.

M.3 Trajectory of spatially adaptive smoothing splines

Let the curvature penalty $\rho(x) = \frac{1}{\bar{\eta} t} \frac{1}{\zeta(x)} \frac{1}{M}$. Then (A141) can be written as

$$\min_{h \in W_2(S)} \frac{1}{M} \sum_{i=1}^{M} [h(x_j) - y_j]^2 + \int_S \rho(x)(h''(x))^2 \, \mathrm{d}x, \tag{A144}$$

where $W_2(S) = \{f \colon f, f' \text{ absolutely continuous and } f'' \in L^2(S)\}$, with $L^2(S)$ the square integrable functions over the domain S. Abramovich and Steinberg (1996); Pintore et al. (2006) give out the solution of (A144) explicitly, which is called a spatially adaptive smoothing spline.

According to Pintore et al. (2006), the solution can be derived in terms of an appropriate RKHS representation of W_2^0 with inner product $\langle f, g \rangle_\rho = \int f''(x) g''(x) \rho(x) \, \mathrm{d}x$. Here $W_0^2(S) = W_2(S) \cap B_2(S)$, where $W_2(S)$ is defined above, and $B_2(S) = \{f : f(0) = f'(0) = 0\}$. Notice that when defining $B_2(S)$ we need $0 \in S$. Actually we can choose any point in S. Pintore et al. (2006) define $B_2(S)$ in this way just for simplicity. Then the kernel of the space $W_0^2(S)$ is given by

$$K_\rho(x_1, x_2) = \int_S \rho(u)^{-1} [x_1 - u]_+ [x_2 - u]_+ \mathrm{d}u. \tag{A145}$$

Then the minimizer $\hat{h}$ of (A144) is given by

$$\hat{h}(x) = \sum_{j=1}^{M} c_j K_\rho(x_j, x) + a + bx. \tag{A146}$$

Now define the $M \times M$ matrix

$$\Sigma_\rho = \{K_\rho(x_i, x_j)\}_{i,j=1,\ldots,M}, \tag{A147}$$

and the $M \times 2$ matrix

$$T = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_M \end{bmatrix}. \tag{A148}$$

Denote the vector of coefficients $\mathbf{c} = (c_1, \ldots, c_M)^T$ and the vector of output values $\mathbf{y} = (y_1, \ldots, y_M)^T$. Then the coefficients in (A146) satisfy the following conditions:

$$\Sigma_\rho \left[(\Sigma_\rho + MI)\mathbf{c} + T \begin{pmatrix} a \\ b \end{pmatrix} \right] = \Sigma_\rho \mathbf{y} \quad \text{and} \quad T^\top \left[\Sigma_\rho \mathbf{c} + T \begin{pmatrix} a \\ b \end{pmatrix} \right] = T^\top \mathbf{y}. \tag{A149}$$

After solving for (A149), we get the values of $\mathbf{c}$, a and b. Plug them into (A146), then we get the exact form of the minimizer of (A144).

N Solution to the variational problems after training

N.1 Interpolating splines with uniform curvature penalty

Theorem 9 (b) and (c) show that for certain distributions of $(\mathcal{W}, \mathcal{B})$, ζ is constant. In this case problem (21) is solved by the cubic spline interpolation of the data with natural boundary conditions (Ahlberg et al., 1967).

Theorem A19 (Ahlberg et al. 1967)**.** *For training samples* $\{(x_i, y_i)\}_{i=1}^M$*, suppose* $x_j \in S$*,* $j = 1, \ldots, M$*. Then cubic spline interpolation of data* $\{(x_i, y_i)\}_{i=1}^M$ *with natural boundary condition is the solution of*

$$\begin{aligned} \min_{h \in C^2(S)} \quad & \int_S (h''(x))^2 \mathrm{d}x \\ \text{subject to} \quad & h(x_j) = y_j, \quad j = 1, \ldots, m. \end{aligned} \tag{A150}$$

As already mentioned in Appendix M, cubic spline interpolation is a finite dimensional linear problem and can be solved exactly. A cubic spline is a piecewise polynomial of order 3 with $(M-1)$ pieces. The j-th piece has the form $S_j(x) = a_j + b_j x + c_j x^2 + d_j x^3$, $j = 1, \cdots, M-1$. These $(M-1)$ pieces satisfy equations $S_i(x_i) = y_i$, $S_i(x_{i+1}) = y_{i+1}$, $i = 1, \cdots, M-1$ and $S_i'(x_{i+1}) = S_{i+1}'(x_{i+1})$, $S_i''(x_{i+1}) = S_{i+1}''(x_{i+1})$, $i = 1, \cdots, M-2$, and $S_1''(x_1) = S_{M-1}''(x_M) = 0$. Hence computing the spline amounts to solving a linear system in $4(M-1)$ indeterminates.

N.2 Spatially adaptive interpolating splines

In the case that ζ is not constant, we can still give out the form of the solution to the variational problem (21) by using the result in Appendix M.3. We add a coefficient λ before the regularization term in the optimization problem (A144) and choose $\rho(x) = \frac{1}{\zeta(x)}$. Then we get

$$\min_{h \in W_2(S)} \frac{1}{M} \sum_{i=1}^M \left[h(x_j) - y_j\right]^2 + \lambda \int_S \frac{1}{\zeta(x)} (h''(x))^2 \, \mathrm{d}x. \tag{A151}$$

As $\lambda \to 0$, the minimizer of (A151) converges to the solution of the following optimization problem:

$$\min_{h \in W^2(S)} \int_S \frac{(h''(x))^2}{\zeta(x)} \, \mathrm{d}x \quad \text{s.t.} \quad h(x_j) = y_j, \quad j = 1, \ldots, m, \tag{A152}$$

which is exactly the variational problem (21). According to Appendix M, the solution of (A151) is given by:

$$\hat{h}^{(\lambda)}(x) = \sum_{j=1}^M c_j^{(\lambda)} K_{\frac{\lambda}{\zeta}}(x_j, x) + a^{(\lambda)} + b^{(\lambda)} x. \tag{A153}$$

And the vector $\mathbf{c}^{(\lambda)} = (c_1^{(\lambda)}, \cdots, c_M^{(\lambda)})^T$, $a^{(\lambda)}$ and $b^{(\lambda)}$ satisfy the following conditions:

$$\Sigma_{\frac{\lambda}{\zeta}} \left[(\Sigma_{\frac{\lambda}{\zeta}} + MI)\mathbf{c}^{(\lambda)} + T \begin{pmatrix} a^{(\lambda)} \\ b^{(\lambda)} \end{pmatrix} \right] = \Sigma_{\frac{\lambda}{\zeta}} \mathbf{y} \quad \text{and} \quad T^\top \left[\Sigma_{\frac{\lambda}{\zeta}} \mathbf{c}^{(\lambda)} + T \begin{pmatrix} a^{(\lambda)} \\ b^{(\lambda)} \end{pmatrix} \right] = T^\top \mathbf{y}, \tag{A154}$$

where $K_{\frac{\lambda}{\zeta}}$, $\Sigma_{\frac{\lambda}{\zeta}}$ and T are defined in (A145), (A147) and (A148). Since

$$\begin{aligned} K_{\frac{\lambda}{\zeta}}(x_1, x_2) &= \int_S \left(\frac{\lambda}{\zeta}\right)^{-1} [x_1 - u]_+ [x_2 - u]_+ \mathrm{d}u \\ &= \lambda^{-1} \int_S \left(\frac{1}{\zeta}\right)^{-1} [x_1 - u]_+ [x_2 - u]_+ \mathrm{d}u \\ &= \lambda^{-1} K_{\frac{1}{\zeta}}(x_1, x_2) \end{aligned} \tag{A155}$$

Also $\Sigma_{\frac{\lambda}{\zeta}} = \lambda^{-1} \Sigma_{\frac{1}{\zeta}}$. Then we let $\bar{c}_j^{(\lambda)} = \lambda^{-1} c_j^{(\lambda)}$ and $\bar{\mathbf{c}}^{(\lambda)} = \lambda^{-1} \mathbf{c}^{(\lambda)}$. So we can rewrite (A153) and (A154) as

$$\hat{h}^{(\lambda)}(x) = \sum_{j=1}^M \bar{c}_j^{(\lambda)} K_{\frac{1}{\zeta}}(x_j, x) + a^{(\lambda)} + b^{(\lambda)} x, \tag{A156}$$

where $\bar{\mathbf{c}}^{(\lambda)}$, $a^{(\lambda)}$ and $b^{(\lambda)}$ satisfy the following conditions:

$$\Sigma_{\frac{1}{\zeta}} \left[(\Sigma_{\frac{1}{\zeta}} + \lambda MI)\bar{\mathbf{c}}^{(\lambda)} + T \begin{pmatrix} a^{(\lambda)} \\ b^{(\lambda)} \end{pmatrix} \right] = \Sigma_{\frac{1}{\zeta}} \mathbf{y} \quad \text{and} \quad T^\top \left[\Sigma_{\frac{1}{\zeta}} \bar{\mathbf{c}}^{(\lambda)} + T \begin{pmatrix} a^{(\lambda)} \\ b^{(\lambda)} \end{pmatrix} \right] = T^\top \mathbf{y}, \tag{A157}$$

Now as $\lambda \to 0$, (A156) and (A157) become:

$$\hat{h}^{(0^+)}(x) = \sum_{j=1}^{M} \bar{c}_j^{(0^+)} K_{\frac{1}{\zeta}}(x_j, x) + a^{(0^+)} + b^{(0^+)} x, \tag{A158}$$

where $\bar{\mathbf{c}}^{(0^+)}$, $a^{(0^+)}$ and $b^{(0^+)}$ satisfy the following conditions:

$$\Sigma_{\frac{1}{\zeta}} \left[\Sigma_{\frac{1}{\zeta}} \bar{\mathbf{c}}^{(0^+)} + T \begin{pmatrix} a^{(0^+)} \\ b^{(0^+)} \end{pmatrix} \right] = \Sigma_{\frac{1}{\zeta}} \mathbf{y} \quad \text{and} \quad T^\top \left[\Sigma_{\frac{1}{\zeta}} \bar{\mathbf{c}}^{(\lambda)} + T \begin{pmatrix} a^{(0^+)} \\ b^{(0^+)} \end{pmatrix} \right] = T^\top \mathbf{y}, \tag{A159}$$

(A158) and (A159) give out the solution of (A151) as $\lambda \to 0$, which is also the solution to the variational problem (21).

O Possible generalizations

O.1 Multi-dimensional inputs

We have focused on 1D regression problems, but of course we are also interested in describing the implicit bias of gradient descent for multi-dimensional regression problems. Some of our results are independent of the input space dimension, and others can be generalized as we discuss in the following.

Consider a shallow neural network with d inputs. Use the same notation as in Section 4, and let $\mathbf{W}^{(1)}$ be the d-dimensional vector sampled from a d-dimensional random vector $\boldsymbol{\mathcal{W}}$. Then optimization problem (15) becomes:

$$\begin{aligned} \min_{\alpha_n \in C(\mathbb{R}^d \times \mathbb{R})} \quad & \int_{\mathbb{R}^d \times \mathbb{R}} \alpha_n^2(\mathbf{W}^{(1)}, b) \, \mathrm{d}\mu_n(\mathbf{W}^{(1)}, b) \\ \text{subject to} \quad & \int_{\mathbb{R}^d \times \mathbb{R}} \alpha_n(\mathbf{W}^{(1)}, b)[\langle \mathbf{W}^{(1)}, \mathbf{x}_j \rangle + b]_+ \, \mathrm{d}\mu_n(\mathbf{W}^{(1)}, b) = y_j, \quad j = 1, \dots, M, \end{aligned} \tag{A160}$$

where $\mathbf{W}^{(1)}$ becomes a d-dimensional vector. The limit of the problem (A160) as width $n \to \infty$ is

$$\begin{aligned} \min_{\alpha \in C(\mathbb{R}^d \times \mathbb{R})} \quad & \int_{\mathbb{R}^d \times \mathbb{R}} \alpha^2(\mathbf{W}^{(1)}, b) \, \mathrm{d}\mu(\mathbf{W}^{(1)}, b) \\ \text{subject to} \quad & \int_{\mathbb{R}^d \times \mathbb{R}} \alpha(\mathbf{W}^{(1)}, b)[\langle \mathbf{W}^{(1)}, \mathbf{x}_j \rangle + b]_+ \, \mathrm{d}\mu(\mathbf{W}^{(1)}, b) = y_j, \quad j = 1, \dots, M, \end{aligned} \tag{A161}$$

Similar to Section 4, we can relax the optimization problem (A161) to

$$\begin{aligned} \min_{\substack{\alpha \in C(\mathbb{R}^d \times \mathbb{R}), \\ \mathbf{u} \in \mathbb{R}^{d+1}, v \in \mathbb{R}}} \quad & \int_{\mathbb{R}^d \times \mathbb{R}} \alpha^2(\mathbf{W}^{(1)}, b) \, \mathrm{d}\mu(\mathbf{W}^{(1)}, b) \\ \text{subject to} \quad & \int_{\mathbb{R}^d \times \mathbb{R}} \alpha(\mathbf{W}^{(1)}, b)[\langle \mathbf{W}^{(1)}, \mathbf{x}_j \rangle + b]_+ \, \mathrm{d}\mu(\mathbf{W}^{(1)}, b) + \langle \mathbf{u}, \mathbf{x}_j \rangle + v = y_j, \ j = 1, \dots, M \end{aligned} \tag{A162}$$

Let $g(\mathbf{x}, \alpha) = \int_{\mathbb{R}^d \times \mathbb{R}} \alpha(\mathbf{W}^{(1)}, b)[\langle \mathbf{W}^{(1)}, \mathbf{x} \rangle + b]_+ \, \mathrm{d}\mu(\mathbf{W}^{(1)}, b) + \langle \mathbf{u}, \mathbf{x}_j \rangle + v$ be the function represented by the infinite-width network. Then the Laplacian $\Delta g(\mathbf{x}, \alpha) = \sum_{i=1}^{d} \partial_{x_i}^2 g(\mathbf{x}, \alpha)$ is given by

$$\begin{aligned} \Delta g(\mathbf{x}, \alpha) &= \int_{\mathbb{R}^{d+1}} \alpha(\mathbf{W}^{(1)}, b) \|\mathbf{W}^{(1)}\|_2^2 \, \delta(\langle \mathbf{W}^{(1)}, \mathbf{x} \rangle + b) \, \mathrm{d}\mu(\mathbf{W}^{(1)}, b) \\ &= \int_{\mathbb{R}} \left(\int_{\mathbb{R}^d} \alpha(\mathbf{W}^{(1)}, b) \|\mathbf{W}^{(1)}\|_2^2 \, \delta(\langle \mathbf{W}^{(1)}, \mathbf{x} \rangle + b) \, \mathrm{d}\mu_{\boldsymbol{\mathcal{W}} | \mathcal{B}=b}(\mathbf{W}^{(1)}) \right) \mathrm{d}\mu_{\mathcal{B}}(b) \end{aligned} \tag{A163}$$

where δ denotes the Dirac delta function; $\mu_{\mathcal{B}}$ denote the distribution of $\mathcal{B}$ which has a density function $p_{\mathcal{B}}(b)$, and $\mu_{\boldsymbol{\mathcal{W}} | \mathcal{B}=b}$ the conditional distribution of $\boldsymbol{\mathcal{W}}$ given $\mathcal{B} = b$. If we assume that $\boldsymbol{\mathcal{W}}$ and $\mathcal{B}$ are

independent, we can further simplify (A163):

$$\begin{aligned}\Delta g(\mathbf{x},\alpha) &= \int_{\mathbb{R}} \left(\int_{\mathbb{R}^d} \alpha(\mathbf{W}^{(1)},b)\|\mathbf{W}^{(1)}\|_2^2\, \delta(\langle \mathbf{W}^{(1)},\mathbf{x}\rangle + b)\, \mathrm{d}\mu_{\boldsymbol{\mathcal{W}}}(\mathbf{W}^{(1)}) \right) \mathrm{d}\mu_{\mathcal{B}}(b) \\ &= \int_{\mathbb{R}^d} \left(\int_{\mathbb{R}} \alpha(\mathbf{W}^{(1)},b)\|\mathbf{W}^{(1)}\|_2^2\, \delta(\langle \mathbf{W}^{(1)},\mathbf{x}\rangle + b) p_{\mathcal{B}}(b)\mathrm{d}b \right) \mathrm{d}\mu_{\boldsymbol{\mathcal{W}}}(\mathbf{W}^{(1)}) \\ &= \int_{\mathbb{R}^d} \alpha(\mathbf{W}^{(1)},-\langle \mathbf{W}^{(1)},\mathbf{x}\rangle)\|\mathbf{W}^{(1)}\|_2^2\, p_{\mathcal{B}}(-\langle \mathbf{W}^{(1)},\mathbf{x}\rangle)\, \mathrm{d}\mu_{\boldsymbol{\mathcal{W}}}(\mathbf{W}^{(1)}),\end{aligned} \tag{A164}$$

where $\mu_{\boldsymbol{\mathcal{W}}}$ denote the distribution of $\boldsymbol{\mathcal{W}}$. If we further assume that $\|\mathbf{W}^{(1)}\|_2 = 1$, which means we sample from the sphere $\mathbb{S}^{d-1}$ when doing initialization. Also assume that $\mu_{\boldsymbol{\mathcal{W}}}$ has a density function $p_{\boldsymbol{\mathcal{W}}}(\mathbf{W}^{(1)})$ on $\mathbb{S}^{d-1}$. Then (A164) becomes

$$\Delta g(\mathbf{x},\alpha) = \int_{\mathbb{S}^{d-1}} \alpha(\mathbf{W}^{(1)},-\langle \mathbf{W}^{(1)},\mathbf{x}\rangle)\, p_{\mathcal{B}}(-\langle \mathbf{W}^{(1)},\mathbf{x}\rangle) p_{\boldsymbol{\mathcal{W}}}(\mathbf{W}^{(1)})\, \mathrm{d}\mathbf{W}^{(1)}, \tag{A165}$$

Let $\beta(\mathbf{W}^{(1)},b) = \alpha(\mathbf{W}^{(1)},-b)\, p_{\mathcal{B}}(-b)\, p_{\boldsymbol{\mathcal{W}}}(\mathbf{W}^{(1)})$, then

$$\Delta g(\mathbf{x},\alpha) = \int_{\mathbb{S}^{d-1}} \beta(\mathbf{W}^{(1)},\langle \mathbf{W}^{(1)},\mathbf{x}\rangle)\, \mathrm{d}\mathbf{W}^{(1)}, \tag{A166}$$

Actually the right-hand side of (A166) is precisely the dual Radon transform of β. According to (Ongie et al., 2020, Lemma 3),

$$\beta = -\frac{1}{2(2\pi)^{d-1}} \mathcal{R}\{(-\Delta)^{(d+1)/2} g(\cdot,\alpha)\}, \tag{A167}$$

where $\mathcal{R}$ is the Radon transform which is defined by

$$\mathcal{R}\{f\}(\boldsymbol{\omega},b) \coloneqq \int_{\langle \boldsymbol{\omega},\mathbf{x}\rangle = b} f(\mathbf{x})\mathrm{d}s(\mathbf{x}), \quad (\boldsymbol{\omega},b) \in \mathbb{S}^{d-1}\times\mathbb{R}, \tag{A168}$$

where $\mathrm{d}s(\mathbf{x})$ represents integration with respect to $(d-1)$-dimensional surface measure on the hyperplane $\langle \boldsymbol{\omega},\mathbf{x}\rangle = b$. The power of the negative Laplacian $(-\Delta)^{(d+1)/2}$ in (A167) is the operator defined in Fourier domain by

$$\widehat{(-\Delta)^{(d+1)/2} f}(\boldsymbol{\xi}) = \|\boldsymbol{\xi}\|^{d+1} \widehat{f}(\boldsymbol{\xi}) \tag{A169}$$

When $d+1$ is a even number, $(-\Delta)^{(d+1)/2}$ is the same as applying the negative Laplacian $(d+1)/2$ times, while if $d+1$ is odd it is a pseudo-differential operator given by convolution with a singular kernel.

Then according to (A167) and the definition of β, we have

$$\alpha(\mathbf{W}^{(1)},b) = -\frac{\mathcal{R}\{(-\Delta)^{(d+1)/2} g(\cdot,\alpha)\}(\mathbf{W}^{(1)},-b)}{2(2\pi)^{d-1} p_{\mathcal{B}}(b)\, p_{\boldsymbol{\mathcal{W}}}(\mathbf{W}^{(1)})}, \tag{A170}$$

Then plug (A170) into the objective of (A162),

$$\begin{aligned}&\int_{\mathbb{R}^d\times\mathbb{R}} \alpha^2(\mathbf{W}^{(1)},b)\, \mathrm{d}\mu(\mathbf{W}^{(1)},b) \\ =&\int_{\mathbb{R}^d\times\mathbb{R}} \left(\frac{\mathcal{R}\{(-\Delta)^{(d+1)/2} g(\cdot,\alpha)\}(\mathbf{W}^{(1)},-b)}{2(2\pi)^{d-1} p_{\mathcal{B}}(b)\, p_{\boldsymbol{\mathcal{W}}}(\mathbf{W}^{(1)})} \right)^2 \mathrm{d}\mu(\mathbf{W}^{(1)},b) \\ =&\int_{\mathbb{S}^{d-1}\times\mathbb{R}} \left(\frac{\mathcal{R}\{(-\Delta)^{(d+1)/2} g(\cdot,\alpha)\}(\mathbf{W}^{(1)},-b)}{2(2\pi)^{d-1} p_{\mathcal{B}}(b)\, p_{\boldsymbol{\mathcal{W}}}(\mathbf{W}^{(1)})} \right)^2 p_{\mathcal{B}}(b)\, p_{\boldsymbol{\mathcal{W}}}(\mathbf{W}^{(1)})\, \mathrm{d}\mathbf{W}^{(1)}\mathrm{d}b \\ =&\int_{\mathbb{S}^{d-1}\times\mathbb{R}} \frac{\left(\mathcal{R}\{(-\Delta)^{(d+1)/2} g(\cdot,\alpha)\}(\mathbf{W}^{(1)},-b)\right)^2}{4(2\pi)^{2(d-1)} p_{\mathcal{B}}(b)\, p_{\boldsymbol{\mathcal{W}}}(\mathbf{W}^{(1)})}\, \mathrm{d}\mathbf{W}^{(1)}\mathrm{d}b\end{aligned} \tag{A171}$$

Then similar to Theorem 6, we show that under the assumptions that: (1) $\boldsymbol{\mathcal{W}}$ and $\mathcal{B}$ are independent; (2) distribution of $\mathcal{B}$ which has a density function $p_{\mathcal{B}}(b)$; (3) $\mathbb{P}(\|\boldsymbol{\mathcal{W}}\|_2 = 1) = 1$; (4) $\mu_{\boldsymbol{\mathcal{W}}}$ has a

density function $p_{\boldsymbol{\mathcal{W}}}(\mathbf{W}^{(1)})$ on $\mathbb{S}^{d-1}$, the solution of (A162) in function space actually solves the following optimization problem:

$$\begin{aligned} \min_{g \in C(\mathbb{R}^d)} \quad & \int_{\mathbb{S}^{d-1}\times\mathbb{R}} \frac{\left(\mathcal{R}\{(-\Delta)^{(d+1)/2} g\}(\mathbf{W}^{(1)}, -b)\right)^2}{4(2\pi)^{2(d-1)} p_{\mathcal{B}}(b)\, p_{\boldsymbol{\mathcal{W}}}(\mathbf{W}^{(1)})} \, \mathrm{d}\mathbf{W}^{(1)} \mathrm{d}b \\ \text{subject to} \quad & g(\mathbf{x}_j) = y_j, \quad j = 1, \ldots, M, \end{aligned} \tag{A172}$$

The optimization problem (A172) characterizes the implicit bias of the gradient descent in function space for multi-dimensional setting. The details of Radon transform and how to make it well-defined are shown in the work of Ongie et al. (2020). We omit the details here.

Zhang et al. (2019) obtained a characterization in terms of the minimization of a kernel norm in function space. This result is also valid for multidimensional inputs. In Appendix L we proved the equivalence between kernel norm minimization and our results in the one-dimensional setting. In the multi-dimensional setting, it will be interesting to show that the kernel norm is equivalent to the objective in (A172) under some conditions.

Lastly, in the multi-dimensional setting the breakpoint density in one-dimensional setting is replaced by a density of the locus of non-linearity of the represented functions, which has been studied by Hanin and Rolnick (2019).

O.2 Other activation functions

We have focused on networks with ReLUs. The ReLU is special in that the second derivative of ReLU is a delta function. For other activation functions the variational problem on function space will look different.

The paper by Parhi and Nowak (2019) considers different types of activation functions σ. These are then related to different types of linear operators L in the definition of the smoothness regularizer. Here L and σ satisfy $\mathrm{L}\sigma = \delta$, i.e. σ is a Green's function of L. Suppose σ is homogeneous. Then Parhi and Nowak (2019) show that minimizing the weight "norm"[3] of two-layer neural networks with activation function σ is actually minimizing 1-norm of $\mathrm{L}f$ where f is the output function of the neural network.

The approach in Parhi and Nowak (2019) can be combined with our analysis. So if for example we replace the ReLU by another homogeneous activation, we can replace the operator accordingly and get an analogous result. Use the same notation as in Section 4, and let σ be the activation function, where we assume that σ is a Green's function of a linear operator L. Then optimization problem (15) becomes:

$$\begin{aligned} \min_{\alpha_n \in C(\mathbb{R}^2)} \quad & \int_{\mathbb{R}^2} \alpha_n^2(W^{(1)}, b) \, \mathrm{d}\mu_n(W^{(1)}, b) \\ \text{subject to} \quad & \int_{\mathbb{R}^2} \alpha_n(W^{(1)}, b)\sigma(W^{(1)} x_j + b) \, \mathrm{d}\mu_n(W^{(1)}, b) = y_j, \quad j = 1, \ldots, M. \end{aligned} \tag{A173}$$

The limit of the problem (A173) as width $n \to \infty$ is

$$\begin{aligned} \min_{\alpha \in C(\mathbb{R}^2)} \quad & \int_{\mathbb{R}^2} \alpha^2(W^{(1)}, b) \, \mathrm{d}\mu(W^{(1)}, b) \\ \text{subject to} \quad & \int_{\mathbb{R}^2} \alpha(W^{(1)}, b)\sigma(W^{(1)} x_j + b) \, \mathrm{d}\mu(W^{(1)}, b) = y_j, \quad j = 1, \ldots, M. \end{aligned} \tag{A174}$$

As in Section 4.2, we can change the variables and relax the optimization problem (A174) to

$$\begin{aligned} \min_{\substack{\gamma \in C(\mathbb{R}^2), \\ p \in C(\mathbb{R})}} \quad & \int_{\mathbb{R}^2} \gamma^2(W^{(1)}, c) \, \mathrm{d}\nu(W^{(1)}, c) \\ \text{subject to} \quad & p(x_j) + \int_{\mathbb{R}^2} \gamma(W^{(1)}, c)\sigma\left(W^{(1)}(x_j - c)\right) \, \mathrm{d}\nu(W^{(1)}, c) = y_j, \quad j = 1, \ldots, M \\ & \mathrm{L}\, p \equiv 0. \end{aligned} \tag{A175}$$

[3] Here the form of "norm" depends on the degree of homogeneity of the activation σ. We use quotation marks here because the weight "norm" is a generalized notion of norm. It may not satisfy the property of norm.

If the activation function σ is ReLU, p is a linear function. Then (A175) becomes the optimization problem (18). Define the output function g of the neural network by

$$g(x,(\gamma,p)) = p(x) + \int_{\mathbb{R}^2} \gamma(W^{(1)},c)[W^{(1)}(x-c)]_+ \,\mathrm{d}\nu(W^{(1)},c).$$

Assume that the activation function σ is homogeneous of degree k, i.e. $\sigma(ax) = a^k\sigma(x)$ for all $a > 0$. Similar to (A82), we have

$$\begin{aligned}
(\mathrm{L}g)(x,(\gamma,p)) &= \mathrm{L}\left(\int_{\mathbb{R}^2} \gamma(W^{(1)},c)\left|W^{(1)}\right|^k \sigma\left(\mathrm{sign}(W^{(1)})\cdot(x-c)\right)\,\mathrm{d}\nu(W^{(1)},c)\right) \\
&= \int_{\mathbb{R}^2} \gamma(W^{(1)},c)\left|W^{(1)}\right|^k \delta(x-c)\,\mathrm{d}\nu(W^{(1)},c) \\
&= \int_{\mathrm{supp}(\nu_{\mathcal{C}})} \left(\int_{\mathbb{R}} \gamma(W^{(1)},c)\left|W^{(1)}\right|^k \,\mathrm{d}\nu_{\mathcal{W}|\mathcal{C}=c}(W^{(1)})\right)\delta(x-c)\,\mathrm{d}\nu_{\mathcal{C}}(c) \qquad \text{(A176)} \\
&= \int_{\mathrm{supp}(\nu_{\mathcal{C}})} \left(\int_{\mathbb{R}} \gamma(W^{(1)},c)\left|W^{(1)}\right|^k \,\mathrm{d}\nu_{\mathcal{W}|\mathcal{C}=c}(W^{(1)})\right)\delta(x-c)p_{\mathcal{C}}(c)\mathrm{d}c \\
&= p_{\mathcal{C}}(x)\int_{\mathbb{R}} \gamma(W^{(1)},x)\left|W^{(1)}\right|^k \,\mathrm{d}\nu_{\mathcal{W}|\mathcal{C}=x}(W^{(1)}).
\end{aligned}$$

Then similar to Theorem 6, we show that the solution of (A175) in function space actually solves the following optimization problem:

$$\min_{h\in C^2(S)} \int_S \frac{\left((\mathrm{L}h)(x)\right)^2}{\zeta(x)}\,\mathrm{d}x \quad \text{s.t.} \quad h(x_j) = y_j, \quad j = 1,\ldots,m, \qquad \text{(A177)}$$

where $\zeta(x) = p_{\mathcal{C}}(x)\mathbb{E}(\mathcal{W}^{2k}|\mathcal{C}=x)$ and $S = \mathrm{supp}(\zeta)\cap[\min_i x_i, \max_i x_i]$.

O.3 Deep networks and other architectures

For deep networks of L layers, if we only train the output layer, then we actually train a linear model. We can construct a two-layer neural network with activation σ which is a $(L-1)$-layer neural network. Then training this two-layer network is equivalent to training only the output layer of the deep network. So we can use the arguments in Section O.2 about the other activation functions. However, it remains unclear how we can find out the operator L corresponding to this activation σ.

In the case of shallow networks, we show that training only the output layer is similar to training all parameters. Our analysis of shallow networks is based on this. However, in the case of a deep network, training only the output layer is no longer similar to training all parameters. If we train all model parameters, the results from Lee et al. (2019) show that the model still is approximated by a linearized model. The result on kernel norm minimization (Zhang et al., 2019) holds in this case. It will be interesting to study the explicit form of the kernel norm, and extensions of our analysis to the case of training all parameters of deep networks.

O.4 Other loss functions

We have focused on the implicit bias of gradient descent for regression. For this type of problems, one often considers a loss function (per example) which has a single finite minimum. Roughly speaking, our description of the bias is in terms of smoothness properties of the solution functions. There are various works on the implicit bias of gradient descent for classification problems, e.g. Soudry et al. (2018). The implicit bias is often formulated in terms of maximum margins.

In our analysis, some theorems require that the loss function is mean square error (MSE). In Theorem 3, the gradient flow is a linear differential equation if we use MSE. If we use a different loss, this will be more complicated. However, we think that the result will generalize. We are also using the result from Lee et al. (2018), which is based on MSE. According to them it is not clear whether their result will still apply for other loss functions. Theorem 5 and Theorem 6 are about a variational problem that is derived from Theorem 2, in relation to the minimization of $\|\theta-\theta_2\|_2$. Theorem 2 remains valid for other loss functions beside MSE. To sum up, if we can generalize the Theorem 3 and the result of Lee et al. (2018) to other loss functions, we can generalize our main result in Theorem 1 to other loss functions.

O.5 Other optimization procedures

It would be interesting to extend the analysis to modifications of the basic gradient descent optimization procedure. The implicit bias of different optimization methods has been studied by Gunasekar et al. (2018a) covering some instances of mirror descent, natural gradient descent, Adam, and steepest descent with respect to different potentials and norms. In particular, they show that the implicit bias of coordinate descent corresponds to the minimization of the 1-norm of the weights. It will be interesting to work out the explicit form of these descriptions in function space.

The implicit bias of SGD has been studied in a series of articles, taking a different perspective to the implicit bias of gradient descent. This has been linked to the shape of the optimization landscape, with smaller mini-batch or larger step size leading to a bias towards wider minima of the objective function (Keskar et al., 2017; Wu et al., 2017; Dinh et al., 2017). Further, this is related to stability and robustness. Here again, it will be interesting to take an NTK perspective to analyze SGD (Allen-Zhu et al., 2019) and explore kernel norm minimization and explicit forms of the regularization in function space under SGD.