Chapter 11

Wide neural networks

In this chapter we explore the dynamics of training neural networks of large width. Throughout we focus on the situation where we have data pairs

$$(\boldsymbol{x}_i, y_i) \in \mathbb{R}^d \times \mathbb{R} \qquad i \in \{1, \dots, m\}, \tag{11.0.1a}$$

and wish to train a neural network $\Phi(\boldsymbol{x}, \boldsymbol{w})$ depending on the input $\boldsymbol{x} \in \mathbb{R}^d$ and the parameters $\boldsymbol{w} \in \mathbb{R}^n$, by minimizing the square loss objective defined as

$$f(\boldsymbol{w}) := \sum_{i=1}^{m} (\Phi(\boldsymbol{x}_i, \boldsymbol{w}) - y_i)^2, \tag{11.0.1b}$$

which is a multiple of the empirical risk $\widehat{\mathcal{R}}_S(\Phi)$ in (1.2.3) for the sample $S = (\boldsymbol{x}_i, y_i)_{i=1}^m$ and the square-loss. We exclusively focus on gradient descent with a constant step size h, which yields a sequence of parameters $(\boldsymbol{w}_k)_{k \in \mathbb{N}}$. We aim to understand the evolution of $\Phi(\boldsymbol{x}, \boldsymbol{w}_k)$ as k progresses. For linear mappings $\boldsymbol{w} \mapsto \Phi(\boldsymbol{x}, \boldsymbol{w})$, the objective function (11.0.1b) is convex. As established in the previous chapter, gradient descent then finds a global minimizer. For typical neural network architectures, $\boldsymbol{w} \mapsto \Phi(\boldsymbol{x}, \boldsymbol{w})$ is not linear, and such a statement is in general not true.

Recent research has highlighted that neural network behavior tends to linearize in the parameters as network width increases [105]. This allows to transfer some of the results and techniques from the linear case to the training of neural networks. We start this chapter in Sections 11.1 and 11.2 by recalling (kernel) least-squares methods, which describe linear (in $\boldsymbol{w}$) models. Following [129], the subsequent sections explore why in the infinite width limit neural networks exhibit linear-like behavior. In Section 11.5.2 we formally introduce the linearization of $\boldsymbol{w} \mapsto \Phi(\boldsymbol{x}, \boldsymbol{w})$. Section 11.4 presents an abstract result showing convergence of gradient descent, under the condition that Φ does not deviate too much from its linearization. In Sections 11.5 and 11.6, we then detail the implications for wide neural networks for two (slightly) different architectures. In particular, we will prove that gradient descent can find global minimizers when applied to (11.0.1b) for networks of very large width. We emphasize that this analysis treats the case of strong overparametrization, specifically when network width increases while keeping the number of data points m fixed.

11.1 Linear least-squares

Arguably one of the simplest machine learning algorithms is linear least-squares regression. Given data (11.0.1a), linear regression tries to fit a linear function $\Phi(\boldsymbol{x}, \boldsymbol{w}) := \boldsymbol{x}^\top \boldsymbol{w}$ in terms of $\boldsymbol{w}$ by

minimizing $f(\boldsymbol{w})$ in (11.0.1b). With

$$\boldsymbol{A} = \begin{pmatrix} \boldsymbol{x}_1^\top \\ \vdots \\ \boldsymbol{x}_m^\top \end{pmatrix} \in \mathbb{R}^{m\times d} \qquad \text{and} \qquad \boldsymbol{y} = \begin{pmatrix} y_1 \\ \vdots \\ y_m \end{pmatrix} \in \mathbb{R}^m \tag{11.1.1}$$

it holds

$$f(\boldsymbol{w}) = \|\boldsymbol{A}\boldsymbol{w} - \boldsymbol{y}\|^2. \tag{11.1.2}$$

Remark 11.1. More generally, the ansatz $\Phi(\boldsymbol{x}, (\boldsymbol{w}, b)) := \boldsymbol{w}^\top \boldsymbol{x} + b$ corresponds to

$$\Phi(\boldsymbol{x}, (\boldsymbol{w}, b)) = (1, \boldsymbol{x}^\top) \begin{pmatrix} b \\ \boldsymbol{w} \end{pmatrix}.$$

Therefore, additionally allowing for a bias can be treated analogously.

The model $\Phi(\boldsymbol{x}, \boldsymbol{w}) = \boldsymbol{x}^\top \boldsymbol{w}$ is linear in both $\boldsymbol{x}$ and $\boldsymbol{w}$. In particular, $\boldsymbol{w} \mapsto f(\boldsymbol{w})$ is a convex function by Exercise 10.34, and we may apply the convergence results of Chapter 10 when using gradient based algorithms. If $\boldsymbol{A}$ is invertible, then f has a unique minimizer given by $\boldsymbol{w}_* = \boldsymbol{A}^{-1}\boldsymbol{y}$. If $\operatorname{rank}(\boldsymbol{A}) = d$, then f is strongly convex by Exercise 10.34, and there still exists a unique minimizer. If however $\operatorname{rank}(\boldsymbol{A}) < d$, then $\ker(\boldsymbol{A}) \neq \{\boldsymbol{0}\}$ and there exist infinitely many minimizers of f. To ensure uniqueness, we look for the **minimum norm solution** (or minimum 2-norm solution)

$$\boldsymbol{w}_* := \operatorname{argmin}_{\{\boldsymbol{w}\in\mathbb{R}^d \,|\, f(\boldsymbol{w}) \leq f(\boldsymbol{v}) \ \forall \boldsymbol{v}\in\mathbb{R}^d\}} \|\boldsymbol{w}\|. \tag{11.1.3}$$

The following proposition establishes the uniqueness of $\boldsymbol{w}_*$ and demonstrates that it can be represented as a superposition of the $(\boldsymbol{x}_i)_{i=1}^m$.

Proposition 11.2. *Let $\boldsymbol{A} \in \mathbb{R}^{m\times d}$ and $\boldsymbol{y} \in \mathbb{R}^m$ be as in* (11.1.1)*. There exists a unique minimum* 2*-norm solution of* (11.1.2)*. Denoting $\tilde{H} := \operatorname{span}\{\boldsymbol{x}_1, \dots, \boldsymbol{x}_m\} \subseteq \mathbb{R}^d$, it is the unique element*

$$\boldsymbol{w}_* = \operatorname{argmin}_{\tilde{\boldsymbol{w}}\in\tilde{H}} f(\tilde{\boldsymbol{w}}) \in \tilde{H}. \tag{11.1.4}$$

Proof. We start with existence and uniqueness. Let $C \subseteq \mathbb{R}^m$ be the space spanned by the columns of $\boldsymbol{A}$. Then C is closed and convex, and therefore $\boldsymbol{y}_* = \operatorname{argmin}_{\tilde{\boldsymbol{y}}\in C} \|\boldsymbol{y} - \tilde{\boldsymbol{y}}\|$ exists and is unique (this is a fundamental property of Hilbert spaces, see Theorem B.14). In particular, the set $M = \{\boldsymbol{w} \in \mathbb{R}^d \,|\, \boldsymbol{A}\boldsymbol{w} = \boldsymbol{y}_*\} \subseteq \mathbb{R}^d$ of minimizers of f is not empty. Clearly M is also closed and convex. By the same argument as before, $\boldsymbol{w}_* = \operatorname{argmin}_{\boldsymbol{w}_*\in M} \|\boldsymbol{w}_*\|$ exists and is unique.

It remains to show (11.1.4). Denote by $\boldsymbol{w}_*$ the minimum norm solution and decompose $\boldsymbol{w}_* = \tilde{\boldsymbol{w}} + \hat{\boldsymbol{w}}$ with $\tilde{\boldsymbol{w}} \in \tilde{H}$ and $\hat{\boldsymbol{w}} \in \tilde{H}^\perp$. We have $\boldsymbol{A}\boldsymbol{w}_* = \boldsymbol{A}\tilde{\boldsymbol{w}}$ and $\|\boldsymbol{w}_*\|^2 = \|\tilde{\boldsymbol{w}}\|^2 + \|\hat{\boldsymbol{w}}\|^2$. Since $\boldsymbol{w}_*$ is the minimal norm solution it must hold $\hat{\boldsymbol{w}} = \boldsymbol{0}$. Thus $\boldsymbol{w}_* \in \tilde{H}$. Finally assume there exists a minimizer $\boldsymbol{v}$ of f in $\tilde{H}$ different from $\boldsymbol{w}_*$. Then $\boldsymbol{0} \neq \boldsymbol{w}_* - \boldsymbol{v} \in \tilde{H}$, and since $\tilde{H}$ is spanned by the rows of $\boldsymbol{A}$ we have $\boldsymbol{A}(\boldsymbol{w}_* - \boldsymbol{v}) \neq \boldsymbol{0}$. Thus $\boldsymbol{y}_* = \boldsymbol{A}\boldsymbol{w}_* \neq \boldsymbol{A}\boldsymbol{v}$, which contradicts that $\boldsymbol{v}$ minimizes f. □

The condition of minimizing the 2-norm is a form of regularization. Interestingly, gradient descent converges to the minimum norm solution for the quadratic objective (11.1.2), as long as $\boldsymbol{w}_0$ is initialized within $\tilde{H} = \operatorname{span}\{\boldsymbol{x}_1, \dots, \boldsymbol{x}_m\}$ (e.g. $\boldsymbol{w}_0 = 0$). Therefore, it does not find an "arbitrary" minimizer but implicitly regularizes the problem in this sense. In the following $s_{\max}(\boldsymbol{A})$ denotes the maximal singular value of $\boldsymbol{A}$.

Theorem 11.3. *Let $\boldsymbol{A} \in \mathbb{R}^{m\times d}$ be as in (11.1.1), let $\boldsymbol{w}_0 = \tilde{\boldsymbol{w}}_0 + \hat{\boldsymbol{w}}_0$ where $\tilde{\boldsymbol{w}}_0 \in \tilde{H}$ and $\hat{\boldsymbol{w}}_0 \in \tilde{H}^\perp$. Fix $h \in (0, 1/(2s_{\max}(\boldsymbol{A})^2))$ and set*

$$\boldsymbol{w}_{k+1} := \boldsymbol{w}_k - h\nabla f(\boldsymbol{w}_k) \qquad \text{for all } k \in \mathbb{N} \tag{11.1.5}$$

with f in (11.1.2). Then

$$\lim_{k\to\infty} \boldsymbol{w}_k = \boldsymbol{w}_* + \hat{\boldsymbol{w}}_0.$$

We sketch the argument in case $\boldsymbol{w}_0 \in \tilde{H}$, and leave the full proof to the reader, see Exercise 11.32. Note that $\tilde{H}$ is the space spanned by the rows of $\boldsymbol{A}$ (or the columns of $\boldsymbol{A}^\top$). The gradient of the objective function equals

$$\nabla f(\boldsymbol{w}) = 2\boldsymbol{A}^\top(\boldsymbol{A}\boldsymbol{w} - \boldsymbol{y}).$$

Therefore, if $\boldsymbol{w}_0 \in \tilde{H}$, then the iterates of gradient descent never leave the subspace $\tilde{H}$. By Exercise 10.34 and Theorem 10.11, for small enough step size, it holds $f(\boldsymbol{w}_k) \to 0$. By Proposition 11.2 there only exists one minimizer in $\tilde{H}$, corresponding to the minimum norm solution. Thus $\boldsymbol{w}_k$ converges to the minimal norm solution.

11.2 Kernel least-squares

Let again $(\boldsymbol{x}_j, y_j) \in \mathbb{R}^d \times \mathbb{R}$, $j = 1, \dots, m$. In many applications linear models are too simplistic, and are not able to capture the true relation between $\boldsymbol{x}$ and y. Kernel methods allow to overcome this problem by introducing nonlinearity in $\boldsymbol{x}$, but retaining linearity in the parameter $\boldsymbol{w}$.

Let H be a Hilbert space with inner product $\langle\cdot,\cdot\rangle_H$, that is also referred to as the **feature space**. For a (typically nonlinear) **feature map** $\phi : \mathbb{R}^d \to H$, consider the model

$$\Phi(\boldsymbol{x}, \boldsymbol{w}) = \langle \phi(\boldsymbol{x}), \boldsymbol{w}\rangle_H \tag{11.2.1}$$

with $\boldsymbol{w} \in H$. If $H = \mathbb{R}^n$, the components of ϕ are referred to as features. With the objective function

$$f(\boldsymbol{w}) := \sum_{j=1}^{m} \big(\langle \phi(\boldsymbol{x}_j), \boldsymbol{w}\rangle_H - y_j\big)^2 \qquad \boldsymbol{w} \in H, \tag{11.2.2}$$

we wish to determine a minimizer of f. To ensure uniqueness and regularize the problem, we again consider the minimum H-norm solution

$$\boldsymbol{w}_* := \operatorname{argmin}_{\{\boldsymbol{w}\in H \,|\, f(\boldsymbol{w})\le f(\boldsymbol{v})\ \forall \boldsymbol{v}\in H\}} \|\boldsymbol{w}\|_H.$$

As we will see below, $\boldsymbol{w}_*$ is well-defined. We will call $\Phi(\boldsymbol{x}, \boldsymbol{w}_*) = \langle \phi(\boldsymbol{x}), \boldsymbol{w}_*\rangle_H$ the **kernel least squares estimator**. The nonlinearity of the feature map allows for more expressive models $\boldsymbol{x} \mapsto \Phi(\boldsymbol{x}, \boldsymbol{w})$ capable of capturing more complicated structures beyond linearity in the data.

Remark 11.4 (Gradient descent). Let $H = \mathbb{R}^n$ be equipped with the Euclidean inner product. Consider the sequence $(\boldsymbol{w}_k)_{k\in\mathbb{N}_0} \subseteq \mathbb{R}^n$ generated by gradient descent to minimize (11.2.2). Assuming sufficiently small step size, by Theorem 11.3 for $\boldsymbol{x} \in \mathbb{R}^d$

$$\lim_{k\to\infty} \Phi(\boldsymbol{x}, \boldsymbol{w}_k) = \langle \phi(\boldsymbol{x}), \boldsymbol{w}_* \rangle + \langle \phi(\boldsymbol{x}), \hat{\boldsymbol{w}}_0 \rangle \,. \tag{11.2.3}$$

Here, $\hat{\boldsymbol{w}}_0 \in \mathbb{R}^n$ denotes the orthogonal projection of $\boldsymbol{w}_0 \in \mathbb{R}^n$ onto $\tilde{H}^\perp$ where $\tilde{H} := \operatorname{span}\{\phi(\boldsymbol{x}_1), \dots, \phi(\boldsymbol{x}_m)\}$. Gradient descent thus yields the kernel least squares estimator plus $\langle \phi(\boldsymbol{x}), \hat{\boldsymbol{w}}_0 \rangle$. Notably, on the set

$$\{\boldsymbol{x} \in \mathbb{R}^d \,|\, \phi(\boldsymbol{x}) \in \operatorname{span}\{\phi(\boldsymbol{x}_1), \dots, \phi(\boldsymbol{x}_m)\}\}, \tag{11.2.4}$$

(11.2.3) thus coincides with the kernel least squares estimator independent of the initialization $\boldsymbol{w}_0$.

11.2.1 Examples

To motivate the concept of feature maps consider the following example from [153].

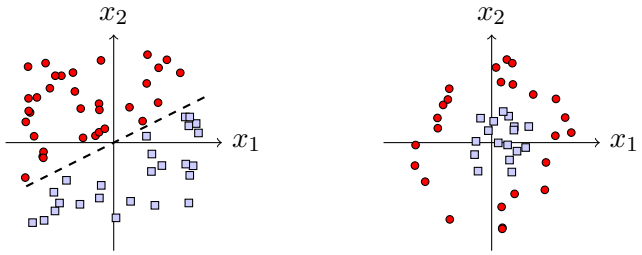
Example 11.5. Let $\boldsymbol{x}_i \in \mathbb{R}^2$ with associated labels $y_i \in \{-1, 1\}$ for $i = 1, \dots, m$. The goal is to find some model $\Phi(\cdot, \boldsymbol{w}) : \mathbb{R}^2 \to \mathbb{R}$, for which

$$\operatorname{sign}(\Phi(\boldsymbol{x}, \boldsymbol{w})) \tag{11.2.5}$$

predicts the label y of $\boldsymbol{x}$. For a linear (in $\boldsymbol{x}$) model

$$\Phi(\boldsymbol{x}, (\boldsymbol{w}, b)) = \boldsymbol{x}^\top \boldsymbol{w} + b,$$

the decision boundary of (11.2.5) equals $\{\boldsymbol{x} \in \mathbb{R}^2 \,|\, \boldsymbol{x}^\top \boldsymbol{w} + b = 0\}$ in $\mathbb{R}^2$. Hence, by adjusting $\boldsymbol{w}$ and b, (11.2.5) can separate data by affine hyperplanes in $\mathbb{R}^2$. Consider two datasets represented by light blue squares for $+1$ and red circles for -1 labels:



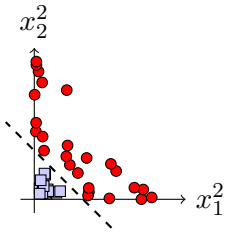
dataset 1 dataset 2

The first dataset is separable by an affine hyperplane as depicted by the dashed line. Thus a linear model is capable of correctly classifying all datapoints. For the second dataset this is not possible.

To enhance model expressivity, introduce a feature map $\phi : \mathbb{R}^2 \to \mathbb{R}^6$ via

$$\phi(\boldsymbol{x}) = (1, x_1, x_2, x_1x_2, x_1^2, x_2^2)^\top \in \mathbb{R}^6 \qquad \text{for all } \boldsymbol{x} \in \mathbb{R}^2. \tag{11.2.6}$$

For $\boldsymbol{w} \in \mathbb{R}^6$, this allows $\Phi(\boldsymbol{x}) = \boldsymbol{w}^\top \phi(\boldsymbol{x})$ to represent arbitrary polynomials of degree 2. With this kernel approach, the decision boundary of (11.2.5) becomes the set of all hyperplanes *in the feature space* passing through $\mathbf{0} \in \mathbb{R}^6$. Visualizing the last two features of the second dataset, we obtain



features 5 and 6 of dataset 2

Note how in the feature space $\mathbb{R}^6$, the datapoints are again separated by such a hyperplane. Thus, with the feature map in (11.2.6), the predictor (11.2.5) can perfectly classify all points also for the second dataset.

In the above example we chose the feature space $H = \mathbb{R}^6$. It is also possible to work with infinite dimensional feature spaces as the next example demonstrates.

Example 11.6. Let $H = \ell^2(\mathbb{N})$ be the space of square summable sequences and $\phi : \mathbb{R}^d \to \ell^2(\mathbb{N})$ some map. Fitting the corresponding model

$$\Phi(\boldsymbol{x}, \boldsymbol{w}) = \langle \phi(\boldsymbol{x}), \boldsymbol{w} \rangle_{\ell^2} = \sum_{i \in \mathbb{N}} \phi_i(\boldsymbol{x}) w_i$$

to data $(\boldsymbol{x}_i, y_i)_{i=1}^m$ requires to minimize

$$f(\boldsymbol{w}) = \sum_{j=1}^m \left(\Big(\sum_{i \in \mathbb{N}} \phi_i(\boldsymbol{x}_j) w_i \Big) - y_j \right)^2 \qquad \boldsymbol{w} \in \ell^2(\mathbb{N}).$$

Hence we have to determine an *infinite sequence* of parameters $(w_i)_{i \in \mathbb{N}}$.

11.2.2 Kernel trick

At first glance, computing a (minimal H-norm) minimizer $\boldsymbol{w}$ in the possibly infinite-dimensional Hilbert space H seems infeasible. The so-called *kernel trick* allows to do this computation. To explain it, we first revisit the foundational representer theorem.

Theorem 11.7 (Representer theorem)**.** *There is a unique minimum H-norm solution $\boldsymbol{w}_* \in H$ of (11.2.2). With $\tilde{H} := \operatorname{span}\{\phi(\boldsymbol{x}_1), \dots, \phi(\boldsymbol{x}_m)\}$ it equals the unique element*

$$\boldsymbol{w}_* = \operatorname{argmin}_{\tilde{\boldsymbol{w}} \in \tilde{H}} f(\tilde{\boldsymbol{w}}) \in \tilde{H}. \tag{11.2.7}$$

Proof. Let $\tilde{\boldsymbol{w}}_1, \dots, \tilde{\boldsymbol{w}}_n$ be a basis of $\tilde{H}$. If $\tilde{H} = \{0\}$ the statement is trivial, so we assume $1 \le n \le m$. Let $\boldsymbol{A} = (\langle \phi(\boldsymbol{x}_i), \tilde{\boldsymbol{w}}_j \rangle)_{ij} \in \mathbb{R}^{m \times n}$. Every $\tilde{\boldsymbol{w}} \in \tilde{H}$ has a unique representation $\tilde{\boldsymbol{w}} = \sum_{j=1}^n \alpha_j \tilde{\boldsymbol{w}}_j$ for some $\boldsymbol{\alpha} \in \mathbb{R}^n$. With this ansatz

$$f(\tilde{\boldsymbol{w}}) = \sum_{i=1}^m (\langle \phi(\boldsymbol{x}_i), \tilde{\boldsymbol{w}} \rangle - y_i)^2 = \sum_{i=1}^m \Big(\sum_{j=1}^n \langle \phi(\boldsymbol{x}_i), \tilde{\boldsymbol{w}}_j \rangle \, \alpha_j - y_i \Big)^2 = \|\boldsymbol{A}\boldsymbol{\alpha} - \boldsymbol{y}\|^2. \tag{11.2.8}$$

Note that $\boldsymbol{A} : \mathbb{R}^n \to \mathbb{R}^m$ is injective since for every $\boldsymbol{\alpha} \in \mathbb{R}^n \backslash \{0\}$ holds $\sum_{j=1}^n \alpha_j \tilde{\boldsymbol{w}}_j \in \tilde{H} \setminus \{0\}$ and hence $\boldsymbol{A}\alpha = (\left\langle \phi(\boldsymbol{x}_i), \sum_{j=1}^n \alpha_j \tilde{\boldsymbol{w}}_j \right\rangle)_{i=1}^m \neq 0$. Therefore, there exists a unique minimizer $\boldsymbol{\alpha} \in \mathbb{R}^n$ of the right-hand side of (11.2.8), and thus there exists a unique minimizer $\boldsymbol{w}_* \in \tilde{H}$ in (11.2.7).

For arbitrary $\boldsymbol{w} \in H$ we wish to show $f(\boldsymbol{w}) \geq f(\boldsymbol{w}_*)$, so that $\boldsymbol{w}_*$ minimizes f in H. Decompose $\boldsymbol{w} = \tilde{\boldsymbol{w}} + \hat{\boldsymbol{w}}$ with $\tilde{\boldsymbol{w}} \in \tilde{H}$ and $\hat{\boldsymbol{w}} \in \tilde{H}^\perp$, i.e. $\langle \phi(\boldsymbol{x}_j), \hat{\boldsymbol{w}} \rangle_H = 0$ for all $j = 1, \ldots, m$. Then, using that $\boldsymbol{w}_*$ minimizes f in $\tilde{H}$,

$$f(\boldsymbol{w}) = \sum_{j=1}^m (\langle \phi(\boldsymbol{x}_j), \boldsymbol{w} \rangle_H - y_j)^2 = \sum_{j=1}^m (\langle \phi(\boldsymbol{x}_j), \tilde{\boldsymbol{w}} \rangle_H - y_j)^2 = f(\tilde{\boldsymbol{w}}) \geq f(\boldsymbol{w}_*).$$

Finally, let $\boldsymbol{w} \in H$ be any minimizer of f in H different from $\boldsymbol{w}_*$. It remains to show $\|\boldsymbol{w}\|_H > \|\boldsymbol{w}_*\|_H$. Decompose again $\boldsymbol{w} = \tilde{\boldsymbol{w}} + \hat{\boldsymbol{w}}$ with $\tilde{\boldsymbol{w}} \in \tilde{H}$ and $\hat{\boldsymbol{w}} \in \tilde{H}^\perp$. As above $f(\boldsymbol{w}) = f(\tilde{\boldsymbol{w}})$ and thus $\tilde{\boldsymbol{w}}$ is a minimizer of f. Uniqueness of $\boldsymbol{w}_*$ in (11.2.7) implies $\tilde{\boldsymbol{w}} = \boldsymbol{w}_*$. Therefore $\hat{\boldsymbol{w}} \neq 0$ and $\|\boldsymbol{w}_*\|_H^2 < \|\tilde{\boldsymbol{w}}\|_H^2 + \|\hat{\boldsymbol{w}}\|_H^2 = \|\boldsymbol{w}\|_H$. □

Instead of looking for the minimum norm minimizer $\boldsymbol{w}_*$ in the Hilbert space H, by Proposition 11.2 it suffices to determine the unique minimizer in the at most m-dimensional subspace $\tilde{H}$ spanned by $\phi(\boldsymbol{x}_1), \ldots, \phi(\boldsymbol{x}_m)$. This significantly simplifies the problem. To do so we first introduce the notion of kernels.

Definition 11.8. A symmetric function $K : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ is called a **kernel** if for any $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m \in \mathbb{R}^d$ the **kernel matrix** $\boldsymbol{G} = (K(\boldsymbol{x}_i, \boldsymbol{x}_j))_{i,j=1}^m \in \mathbb{R}^{m \times m}$ is symmetric positive semidefinite.

Given a feature map $\phi : \mathbb{R}^d \to H$, it is easy to check that

$$K(\boldsymbol{x}, \boldsymbol{x}') := \left\langle \phi(\boldsymbol{x}), \phi(\boldsymbol{x}') \right\rangle_H \qquad \text{for all } \boldsymbol{x}, \boldsymbol{x}' \in \mathbb{R}^d,$$

defines a kernel. The corresponding kernel matrix $\boldsymbol{G} \in \mathbb{R}^{m \times m}$ is given by

$$G_{ij} = \langle \phi(\boldsymbol{x}_i), \phi(\boldsymbol{x}_j) \rangle_H = K(\boldsymbol{x}_i, \boldsymbol{x}_j).$$

With the ansatz $\boldsymbol{w} = \sum_{j=1}^m \alpha_j \phi(\boldsymbol{x}_j)$, minimizing the objective (11.2.2) in $\tilde{H}$ is equivalent to minimizing

$$\|\boldsymbol{G}\boldsymbol{\alpha} - \boldsymbol{y}\|^2, \tag{11.2.9}$$

in $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_m) \in \mathbb{R}^m$.

Proposition 11.9. *Let $\boldsymbol{\alpha} \in \mathbb{R}^m$ be any minimizer of* (11.2.9). *Then $\boldsymbol{w}_* = \sum_{j=1}^m \alpha_j \phi(\boldsymbol{x}_j)$ is the unique minimum H-norm solution of* (11.2.2).

Proposition 11.9, the proof of which is left as an exercise, suggests the following algorithm to compute the kernel least squares estimator:

(i) compute the kernel matrix $\boldsymbol{G} = (K(\boldsymbol{x}_i, \boldsymbol{x}_j))_{i,j=1}^m$,

(ii) determine a minimizer $\boldsymbol{\alpha} \in \mathbb{R}^m$ of $\|\boldsymbol{G}\boldsymbol{\alpha} - \boldsymbol{y}\|$,

(iii) evaluate $\Phi(\boldsymbol{x}, \boldsymbol{w}_*)$ via

$$\Phi(\boldsymbol{x}, \boldsymbol{w}_*) = \left\langle \phi(\boldsymbol{x}), \sum_{j=1}^m \alpha_j \phi(\boldsymbol{x}_j) \right\rangle_H = \sum_{j=1}^m \alpha_j K(\boldsymbol{x}, \boldsymbol{x}_j). \tag{11.2.10}$$

Thus, minimizing (11.2.2) and expressing the kernel least squares estimator does neither require explicit knowledge of the feature map ϕ nor of the minimum norm solution $\boldsymbol{w}_* \in H$. It is sufficient to choose a kernel map $K : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$; this is known as the kernel trick. Given a kernel K, we will therefore also refer to (11.2.10) as the kernel least squares estimator without specifying H or ϕ.

Example 11.10. Common examples of kernels include the **polynomial kernel**

$$K(\boldsymbol{x}, \boldsymbol{x}') = (\boldsymbol{x}^\top \boldsymbol{x}' + c)^r \qquad c \geq 0,\ r \in \mathbb{N},$$

the radial basis function (**RBF**) **kernel**

$$K(\boldsymbol{x}, \boldsymbol{x}') = \exp(-c\|\boldsymbol{x} - \boldsymbol{x}'\|^2) \qquad c > 0,$$

and the **Laplace kernel**

$$K(\boldsymbol{x}, \boldsymbol{x}') = \exp(-c\|\boldsymbol{x} - \boldsymbol{x}'\|) \qquad c > 0.$$

Remark 11.11. If $\Omega \subseteq \mathbb{R}^d$ is compact and $K : \Omega \times \Omega \to \mathbb{R}$ is a continuous kernel, then Mercer's theorem implies existence of a Hilbert space H and a feature map $\phi : \mathbb{R}^d \to H$ such that

$$K(\boldsymbol{x}, \boldsymbol{x}') = \big\langle \phi(\boldsymbol{x}), \phi(\boldsymbol{x}') \big\rangle_H \qquad \text{for all } \boldsymbol{x}, \boldsymbol{x}' \in \Omega,$$

i.e. K is the corresponding kernel. See for instance [216, Thm. 4.49].

11.3 Tangent kernel

Consider again a general model $\Phi(\boldsymbol{x}, \boldsymbol{w})$ with input $\boldsymbol{x} \in \mathbb{R}^d$ and parameters $\boldsymbol{w} \in \mathbb{R}^n$. The goal remains to minimize the square loss objective (11.0.1b) given the data (11.0.1a). If $\boldsymbol{w} \mapsto \Phi(\boldsymbol{x}, \boldsymbol{w})$ is not linear, then unlike in Sections 11.1 and 11.2, the objective function (11.0.1b) is in general not convex, and most results on first order methods in Chapter 10 are not directly applicable.

We now simplify the situation by *linearizing the model in $\boldsymbol{w} \in \mathbb{R}^n$ around the initialization*: Fixing $\boldsymbol{w}_0 \in \mathbb{R}^n$, let

$$\Phi^{\text{lin}}(\boldsymbol{x}, \boldsymbol{w}) := \Phi(\boldsymbol{x}, \boldsymbol{w}_0) + \nabla_{\boldsymbol{w}} \Phi(\boldsymbol{x}, \boldsymbol{w}_0)^\top (\boldsymbol{w} - \boldsymbol{w}_0) \qquad \text{for all } \boldsymbol{w} \in \mathbb{R}^n, \tag{11.3.1}$$

which is the first order Taylor approximation of Φ around the initial parameter $\boldsymbol{w}_0$. Introduce the notation

$$\delta_i := \Phi(\boldsymbol{x}_i, \boldsymbol{w}_0) - \nabla_{\boldsymbol{w}} \Phi(\boldsymbol{x}_i, \boldsymbol{w}_0)^\top \boldsymbol{w}_0 - y_i \qquad \text{for all } i = 1, \dots, m. \tag{11.3.2}$$

The square loss for the linearized model then reads

$$f^{\text{lin}}(\boldsymbol{w}) := \sum_{j=1}^{m} (\Phi^{\text{lin}}(\boldsymbol{x}_i, \boldsymbol{w}) - y_i)^2$$
$$= \sum_{j=1}^{m} (\langle \nabla_{\boldsymbol{w}} \Phi(\boldsymbol{x}_i, \boldsymbol{w}_0), \boldsymbol{w} \rangle + \delta_i)^2, \tag{11.3.3}$$

where $\langle \cdot, \cdot \rangle$ stands for the Euclidean inner product in $\mathbb{R}^n$. Comparing with (11.2.2), minimizing f^{lin} corresponds to a kernel least squares regression with feature map

$$\phi(\boldsymbol{x}) = \nabla_{\boldsymbol{w}} \Phi(\boldsymbol{x}, \boldsymbol{w}_0) \in \mathbb{R}^n.$$

The corresponding kernel is

$$\hat{K}_n(\boldsymbol{x}, \boldsymbol{x}') = \left\langle \nabla_{\boldsymbol{w}} \Phi(\boldsymbol{x}, \boldsymbol{w}_0), \nabla_{\boldsymbol{w}} \Phi(\boldsymbol{x}', \boldsymbol{w}_0) \right\rangle. \tag{11.3.4}$$

We refer to $\hat{K}_n$ as the empirical **tangent kernel**, as it arises from the first order Taylor approximation (the tangent) of the original model Φ around initialization $\boldsymbol{w}_0$. Note that the kernel depends on the choice of $\boldsymbol{w}_0$. As explained in Remark 11.4, training Φ^{lin} with gradient descent yields the kernel least-squares estimator with kernel $\hat{K}_n$ plus an additional term depending on $\boldsymbol{w}_0$.

Of course the linearized model Φ^{lin} only captures the behaviour of Φ for parameters $\boldsymbol{w}$ that are close to $\boldsymbol{w}_0$. If we assume for the moment that during training of Φ, the parameters remain close to initialization, then we can expect similar behaviour and performance of Φ and Φ^{lin}. Under certain assumptions, we will see in the next sections that this is precisely what happens, when the width of a neural network increases. Before we make this precise, in Section 11.4 we investigate whether gradient descent applied to $f(\boldsymbol{w})$ will find a *global* minimizer, under the assumption that Φ^{lin} is a good approximation of Φ.

11.4 Convergence to global minimizers

Intuitively, if $\boldsymbol{w} \mapsto \Phi(\boldsymbol{x}, \boldsymbol{w})$ is not linear but "close enough to its linearization" Φ^{lin} defined in (11.3.1), we expect that the objective function is close to a convex function and gradient descent can still find global minimizers of (11.0.1b). To motivate this, consider Figures 11.1 and 11.2 where we chose the number of training data $m = 1$ and the number of parameters $n = 1$. As we can see, essentially we require the difference of Φ and Φ^{lin} and of their derivatives to be small in a neighbourhood of w_0. The size of the neighbourhood crucially depends on the initial error $\Phi(\boldsymbol{x}_1, w_0) - y_1$, and on the size of the derivative $\frac{d}{dw}\Phi(\boldsymbol{x}_1, w_0)$.

For general m and n, we now make the required assumptions on Φ precise.

Assumption 11.12. Let $\Phi \in C^1(\mathbb{R}^d \times \mathbb{R}^n)$ and $\boldsymbol{w}_0 \in \mathbb{R}^n$. There exist constants $r > 0$, U, $L < \infty$ and $0 < \lambda_{\min} \leq \lambda_{\max} < \infty$ such that

(a) the kernel matrix of the empirical tangent kernel

$$(\hat{K}_n(\boldsymbol{x}_i, \boldsymbol{x}_j))_{i,j=1}^m = \left(\langle \nabla_{\boldsymbol{w}} \Phi(\boldsymbol{x}_i, \boldsymbol{w}_0), \nabla_{\boldsymbol{w}} \Phi(\boldsymbol{x}_j, \boldsymbol{w}_0) \rangle \right)_{i,j=1}^m \in \mathbb{R}^{m \times m} \tag{11.4.1}$$

is regular and its eigenvalues belong to $[\lambda_{\min}, \lambda_{\max}]$,

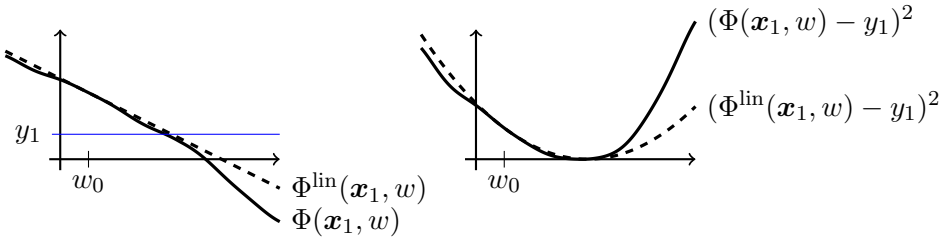


Figure 11.1: Graph of a model $w \mapsto \Phi(\boldsymbol{x}_1, w)$ and its linearization $w \mapsto \Phi^{\text{lin}}(\boldsymbol{x}_1, w)$ at the initial parameter w_0, s.t. $\frac{d}{dw}\Phi(\boldsymbol{x}_1, w_0) \neq 0$. If Φ and Φ^{lin} are close, then there exists w s.t. $\Phi(\boldsymbol{x}_1, w) = y_1$ (left). If the derivatives are also close, the loss $(\Phi(\boldsymbol{x}_1, w) - y_1)^2$ is nearly convex in w, and gradient descent finds a global minimizer (right).

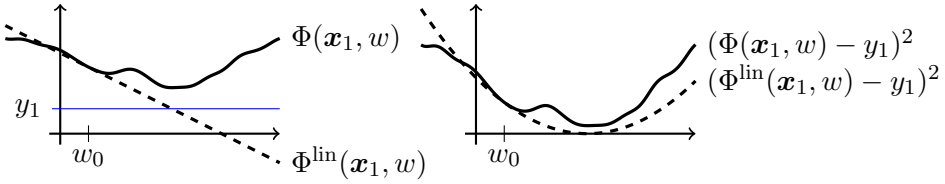


Figure 11.2: Same as Figure 11.1. If Φ and Φ^{lin} are not close, there need not exist w such that $\Phi(\boldsymbol{x}_1, w) = y_1$, and gradient descent need not converge to a global minimizer.

(b) for all $i \in \{1, \dots, m\}$ holds

$$\begin{aligned} \|\nabla_{\boldsymbol{w}} \Phi(\boldsymbol{x}_i, \boldsymbol{w})\| &\leq U && \text{for all } \boldsymbol{w} \in B_r(\boldsymbol{w}_0) \\ \|\nabla_{\boldsymbol{w}} \Phi(\boldsymbol{x}_i, \boldsymbol{w}) - \nabla_{\boldsymbol{w}} \Phi(\boldsymbol{x}_i, \boldsymbol{v})\| &\leq L\|\boldsymbol{w} - \boldsymbol{v}\| && \text{for all } \boldsymbol{w},\ \boldsymbol{v} \in B_r(\boldsymbol{w}_0), \end{aligned} \tag{11.4.2}$$

(c) and

$$L \leq \frac{\lambda_{\min}^2}{12 m^{3/2} U^2 \sqrt{f(\boldsymbol{w}_0)}} \qquad \text{and} \qquad r = \frac{2\sqrt{m} U}{\lambda_{\min}} \sqrt{f(\boldsymbol{w}_0)}. \tag{11.4.3}$$

The regularity of the kernel matrix in Assumption 11.12 (a) is equivalent to $(\nabla_{\boldsymbol{w}} \Phi(\boldsymbol{x}_i, \boldsymbol{w}_0)^\top)_{i=1}^m \in \mathbb{R}^{m \times n}$ having full rank $m \leq n$ (in particular we have at least as many parameters n as training data m). In the context of Figure 11.1, this means that $\frac{d}{dw}\Phi(\boldsymbol{x}_1, w_0) \neq 0$ and thus Φ^{lin} is a not a constant function. This condition guarantees that there exists $\boldsymbol{w}$ such that $\Phi^{\text{lin}}(\boldsymbol{x}_i, \boldsymbol{w}) = y_i$ for all $i = 1, \dots, m$. In other words, already the linearized model Φ^{lin} is sufficiently expressive to interpolate the data. Assumption 11.12 (b) formalizes the closeness condition of Φ and Φ^{lin}. Apart from giving an upper bound on $\nabla_{\boldsymbol{w}} \Phi(\boldsymbol{x}_i, \boldsymbol{w})$, it assumes $\boldsymbol{w} \mapsto \Phi(\boldsymbol{x}_i, \boldsymbol{w})$ to be L-smooth in a ball of radius $r > 0$ around $\boldsymbol{w}_0$, for all $i = 1, \dots, m$. This allows to control how far $\Phi(\boldsymbol{x}_i, \boldsymbol{w})$ and $\Phi^{\text{lin}}(\boldsymbol{x}_i, \boldsymbol{w})$ and their derivatives may deviate from each other for $\boldsymbol{w}$ in this ball. Finally Assumption 11.12 (c) ties together all constants, ensuring the full model to be sufficiently close to its linearization in a large enough neighbourhood of $\boldsymbol{w}_0$.

We are now ready to state the following theorem, which is a variant of [129, Thm. G.1]. In Section 11.5 we will see that its main requirement—Assumption 11.12—is satisfied with high probability for certain (wide) neural networks.

Theorem 11.13. *Let Assumption 11.12 be satisfied and fix a positive learning rate*

$$h \leq \frac{1}{\lambda_{\min} + \lambda_{\max}}. \tag{11.4.4}$$

Set for all $k \in \mathbb{N}$

$$\boldsymbol{w}_{k+1} = \boldsymbol{w}_k - h\nabla f(\boldsymbol{w}_k). \tag{11.4.5}$$

It then holds for all $k \in \mathbb{N}$

$$\|\boldsymbol{w}_k - \boldsymbol{w}_0\| \leq \frac{2\sqrt{m}U}{\lambda_{\min}}\sqrt{f(\boldsymbol{w}_0)} \tag{11.4.6a}$$

$$f(\boldsymbol{w}_k) \leq (1 - h\lambda_{\min})^{2k} f(\boldsymbol{w}_0). \tag{11.4.6b}$$

Proof. In the following denote the error in prediction by

$$E(\boldsymbol{w}) := (\Phi(\boldsymbol{x}_i, \boldsymbol{w}) - y_i)_{i=1}^m \in \mathbb{R}^m$$

such that

$$\nabla E(\boldsymbol{w}) = (\nabla_{\boldsymbol{w}} \Phi(\boldsymbol{x}_i, \boldsymbol{w}))_{i=1}^m \in \mathbb{R}^{m \times n}$$

and with the empirical tangent kernel $\hat{K}_n$ in Assumption 11.12

$$\nabla E(\boldsymbol{w}) \nabla E(\boldsymbol{w})^\top = (\hat{K}_n(\boldsymbol{x}_i, \boldsymbol{x}_j))_{i,j=1}^m \in \mathbb{R}^{m \times m}. \tag{11.4.7}$$

Moreover, (11.4.2) gives

$$\|\nabla E(\boldsymbol{w})\|^2 \leq \|\nabla E(\boldsymbol{w})\|_F^2 = \sum_{i=1}^m \|\nabla \Phi(\boldsymbol{x}_i, \boldsymbol{w})\|^2 \leq mU^2 \qquad \text{for all } \boldsymbol{w} \in B_r(\boldsymbol{w}_0), \tag{11.4.8a}$$

and similarly

$$\begin{aligned} \|\nabla E(\boldsymbol{w}) - \nabla E(\boldsymbol{v})\|^2 &\leq \sum_{i=1}^m \|\nabla_{\boldsymbol{w}} \Phi(\boldsymbol{x}_i, \boldsymbol{w}) - \nabla_{\boldsymbol{w}} \Phi(\boldsymbol{x}_i, \boldsymbol{v})\|^2 \\ &\leq mL^2 \|\boldsymbol{w} - \boldsymbol{v}\|^2 \qquad \text{for all } \boldsymbol{w},\ \boldsymbol{v} \in B_r(\boldsymbol{w}_0). \end{aligned} \tag{11.4.8b}$$

Denote $c := 1 - h\lambda_{\min} \in (0, 1)$. We use induction over k to prove

$$\sum_{j=0}^{k-1} \|\boldsymbol{w}_{j+1} - \boldsymbol{w}_j\| \leq h2\sqrt{m}U\|E(\boldsymbol{w}_0)\| \sum_{j=0}^{k-1} c^j, \tag{11.4.9a}$$

$$\|E(\boldsymbol{w}_k)\|^2 \leq \|E(\boldsymbol{w}_0)\|^2 c^{2k}, \tag{11.4.9b}$$

for all $k \in \mathbb{N}_0$ and where an empty sum is understood as zero. Since $\sum_{j=0}^\infty c^j = (1-c)^{-1} = (h\lambda_{\min})^{-1}$ and $f(\boldsymbol{w}_k) = \|E(\boldsymbol{w}_k)\|^2$, these inequalities directly imply (11.4.6).

The case $k = 0$ is trivial. For the induction step, assume (11.4.9) holds for some $k \in \mathbb{N}_0$.

Step 1. We show (11.4.9a) for $k+1$. The induction assumption and (11.4.3) give

$$\|\boldsymbol{w}_k - \boldsymbol{w}_0\| \le 2h\sqrt{m}U\|E(\boldsymbol{w}_0)\| \sum_{j=0}^{\infty} c^j = \frac{2\sqrt{m}U}{\lambda_{\min}}\sqrt{f(\boldsymbol{w}_0)} = r, \tag{11.4.10}$$

and thus $\boldsymbol{w}_k \in B_r(\boldsymbol{w}_0)$. Next

$$\nabla f(\boldsymbol{w}_k) = \nabla(E(\boldsymbol{w}_k)^\top E(\boldsymbol{w}_k)) = 2\nabla E(\boldsymbol{w}_k)^\top E(\boldsymbol{w}_k). \tag{11.4.11}$$

Using the iteration rule (11.4.5), the bound (11.4.8a), and (11.4.9b)

$$\begin{aligned}\|\boldsymbol{w}_{k+1} - \boldsymbol{w}_k\| &= 2h\|\nabla E(\boldsymbol{w}_k)^\top E(\boldsymbol{w}_k)\| \\ &\le 2h\sqrt{m}U\|E(\boldsymbol{w}_k)\| \\ &\le 2h\sqrt{m}U\|E(\boldsymbol{w}_0)\|c^k.\end{aligned}$$

This shows (11.4.9a) for $k+1$. In particular, as in (11.4.10) we conclude

$$\boldsymbol{w}_{k+1},\ \boldsymbol{w}_k \in B_r(\boldsymbol{w}_0). \tag{11.4.12}$$

Step 2. We show (11.4.9b) for $k+1$. Since E is continuously differentiable, there exists $\tilde{\boldsymbol{w}}_k$ in the convex hull of $\boldsymbol{w}_k$ and $\boldsymbol{w}_{k+1}$ such that

$$E(\boldsymbol{w}_{k+1}) = E(\boldsymbol{w}_k) + \nabla E(\tilde{\boldsymbol{w}}_k)(\boldsymbol{w}_{k+1} - \boldsymbol{w}_k) = E(\boldsymbol{w}_k) - h\nabla E(\tilde{\boldsymbol{w}}_k)\nabla f(\boldsymbol{w}_k),$$

and thus by (11.4.11)

$$\begin{aligned}E(\boldsymbol{w}_{k+1}) &= E(\boldsymbol{w}_k) - 2h\nabla E(\tilde{\boldsymbol{w}}_k)\nabla E(\boldsymbol{w}_k)^\top E(\boldsymbol{w}_k) \\ &= \big(\boldsymbol{I}_m - 2h\nabla E(\tilde{\boldsymbol{w}}_k)\nabla E(\boldsymbol{w}_k)^\top\big)E(\boldsymbol{w}_k),\end{aligned}$$

where $\boldsymbol{I}_m \in \mathbb{R}^{m\times m}$ is the identity matrix. We wish to show that

$$\|\boldsymbol{I}_m - 2h\nabla E(\tilde{\boldsymbol{w}}_k)\nabla E(\boldsymbol{w}_k)^\top\| \le c, \tag{11.4.13}$$

which then implies (11.4.9b) for $k+1$ and concludes the proof.

Using (11.4.8) and the fact that $\boldsymbol{w}_k$, $\tilde{\boldsymbol{w}}_k \in B_r(\boldsymbol{w}_0)$ by (11.4.12),

$$\begin{aligned}&\|\nabla E(\tilde{\boldsymbol{w}}_k)\nabla E(\boldsymbol{w}_k)^\top - \nabla E(\boldsymbol{w}_0)\nabla E(\boldsymbol{w}_0)^\top\| \\ &\quad\le \|\nabla E(\tilde{\boldsymbol{w}}_k)\nabla E(\boldsymbol{w}_k)^\top - \nabla E(\boldsymbol{w}_k)\nabla E(\boldsymbol{w}_k)^\top\| \\ &\qquad + \|\nabla E(\tilde{\boldsymbol{w}}_k)\nabla E(\boldsymbol{w}_k)^\top - \nabla E(\boldsymbol{w}_k)\nabla E(\boldsymbol{w}_0)^\top\| \\ &\qquad + \|\nabla E(\tilde{\boldsymbol{w}}_k)\nabla E(\boldsymbol{w}_0)^\top - \nabla E(\boldsymbol{w}_0)\nabla E(\boldsymbol{w}_0)^\top\| \\ &\quad\le 3mULr.\end{aligned}$$

Since the eigenvalues of $\nabla E(\boldsymbol{w}_0)\nabla E(\boldsymbol{w}_0)^\top$ belong to $[\lambda_{\min}, \lambda_{\max}]$ by (11.4.7) and Assumption 11.12 (a), as long as $h \le (\lambda_{\min} + \lambda_{\max})^{-1}$ we have

$$\begin{aligned}\|\boldsymbol{I}_m - 2h\nabla E(\tilde{\boldsymbol{w}}_k)\nabla E(\boldsymbol{w}_k)^\top\| &\le \|\boldsymbol{I}_m - 2h\nabla E(\boldsymbol{w}_0)\nabla E(\boldsymbol{w}_0)^\top\| + 6hmULr \\ &\le 1 - 2h\lambda_{\min} + 6hmULr \\ &\le 1 - 2h(\lambda_{\min} - 3mULr) \\ &\le 1 - h\lambda_{\min} = c,\end{aligned}$$

where we have used the equality for r and the upper bound for L in (11.4.3). □

Let us emphasize the main statement of Theorem 11.13. By (11.4.6b), full batch gradient descent (11.4.5) achieves zero loss in the limit, i.e. the data is interpolated by the limiting model. In particular, this yields convergence for the (possibly nonconvex) optimization problem of minimizing $f(\boldsymbol{w})$.

11.5 Training dynamics for LeCun initialization

In this and the next section we discuss the implications of Theorem 11.13 for wide neural networks. For ease of presentation we focus on shallow networks with only one hidden layer, but stress that similar considerations also hold for deep networks, see the bibliography section.

11.5.1 Architecture

Let $\Phi : \mathbb{R}^d \to \mathbb{R}$ be a neural network of depth one and width $n \in \mathbb{N}$ of type

$$\Phi(\boldsymbol{x}, \boldsymbol{w}) = \boldsymbol{v}^\top \sigma(\boldsymbol{U}\boldsymbol{x} + \boldsymbol{b}) + c. \tag{11.5.1}$$

Here $\boldsymbol{x} \in \mathbb{R}^d$ is the input, and $\boldsymbol{U} \in \mathbb{R}^{n\times d}$, $\boldsymbol{v} \in \mathbb{R}^n$, $\boldsymbol{b} \in \mathbb{R}^n$ and $c \in \mathbb{R}$ are the parameters which we collect in the vector $\boldsymbol{w} = (\boldsymbol{U}, \boldsymbol{b}, \boldsymbol{v}, c) \in \mathbb{R}^{n(d+2)+1}$ (with $\boldsymbol{U}$ suitably reshaped). For future reference we note that

$$\begin{aligned}
\nabla_{\boldsymbol{U}}\Phi(\boldsymbol{x}, \boldsymbol{w}) &= (\boldsymbol{v} \odot \sigma'(\boldsymbol{U}\boldsymbol{x} + \boldsymbol{b}))\boldsymbol{x}^\top \in \mathbb{R}^{n\times d} \\
\nabla_{\boldsymbol{b}}\Phi(\boldsymbol{x}, \boldsymbol{w}) &= \boldsymbol{v} \odot \sigma'(\boldsymbol{U}\boldsymbol{x} + \boldsymbol{b}) \in \mathbb{R}^n \\
\nabla_{\boldsymbol{v}}\Phi(\boldsymbol{x}, \boldsymbol{w}) &= \sigma(\boldsymbol{U}\boldsymbol{x} + \boldsymbol{b}) \in \mathbb{R}^n \\
\nabla_{c}\Phi(\boldsymbol{x}, \boldsymbol{w}) &= 1 \in \mathbb{R},
\end{aligned} \tag{11.5.2}$$

where $\odot$ denotes the Hadamard product. We also write $\nabla_{\boldsymbol{w}}\Phi(\boldsymbol{x}, \boldsymbol{w}) \in \mathbb{R}^{n(d+2)+1}$ to denote the full gradient with respect to all parameters.

In practice, it is common to initialize the weights randomly, and in this section we consider so-called LeCun initialization. The following condition on the distribution used for this initialization will be assumed throughout the rest of Section 11.5.

Assumption 11.14. The distribution $\mathcal{D}$ on $\mathbb{R}$ has expectation zero, variance one, and finite moments up to order eight.

To explicitly indicate the expectation and variance in the notation, we also write $\mathcal{D}(0, 1)$ instead of $\mathcal{D}$, and for $\mu \in \mathbb{R}$ and $\varsigma > 0$ we use $\mathcal{D}(\mu, \varsigma^2)$ to denote the corresponding scaled and shifted measure with expectation μ and variance ς^2; thus, if $X \sim \mathcal{D}(0, 1)$ then $\mu + \varsigma X \sim \mathcal{D}(\mu, \varsigma^2)$. LeCun initialization [127] sets the variance of the weights in each layer to be reciprocal to the input dimension of the layer, thereby normalizing the output variance across all network nodes. The initial parameters

$$\boldsymbol{w}_0 = (\boldsymbol{U}_0, \boldsymbol{b}_0, \boldsymbol{v}_0, c_0)$$

are thus randomly initialized with components

$$U_{0;ij} \overset{\text{iid}}{\sim} \mathcal{D}\Big(0, \frac{1}{d}\Big), \qquad v_{0;i} \overset{\text{iid}}{\sim} \mathcal{D}\Big(0, \frac{1}{n}\Big), \qquad b_{0;i},\ c_0 = 0, \tag{11.5.3}$$

independently for all $i = 1, \dots, n$, $j = 1, \dots, d$. For a fixed $\varsigma > 0$ one might choose variances ς^2/d and ς^2/n in (11.5.3), which would require only minor modifications in the rest of this section. Biases

are set to zero for simplicity, with nonzero initialization discussed in the exercises. All expectations and probabilities in Section 11.5 are understood with respect to this random initialization.

Example 11.15. Typical examples for $\mathcal{D}(0,1)$ are the standard normal distribution on $\mathbb{R}$ or the uniform distribution on $[-\sqrt{3},\sqrt{3}]$.

11.5.2 Neural tangent kernel

We begin our analysis by investigating the empirical tangent kernel

$$\hat{K}_n(\boldsymbol{x},\boldsymbol{z}) = \langle \nabla_{\boldsymbol{w}}\Phi(\boldsymbol{x},\boldsymbol{w}_0), \nabla_{\boldsymbol{w}}\Phi(\boldsymbol{z},\boldsymbol{w}_0)\rangle$$

of the shallow network (11.5.1). Scaled properly, it converges in the infinite width limit $n\to\infty$ towards a specific kernel known as the **neural tangent kernel** (NTK). Its precise formula depends on the architecture and initialization. For the LeCun initialization (11.5.3) we denote it by K^{LC}.

Theorem 11.16. *Let $R<\infty$ such that $|\sigma(x)|\le R\cdot(1+|x|)$ and $|\sigma'(x)|\le R\cdot(1+|x|)$ for all $x\in\mathbb{R}$. For any $\boldsymbol{x}$, $\boldsymbol{z}\in\mathbb{R}^d$ and $u_i \overset{\mathrm{iid}}{\sim} \mathcal{D}(0,1/d)$, $i=1,\dots,d$, it then holds*

$$\lim_{n\to\infty}\frac{1}{n}\hat{K}_n(\boldsymbol{x},\boldsymbol{z}) = \mathbb{E}[\sigma(\boldsymbol{u}^\top\boldsymbol{x})^\top\sigma(\boldsymbol{u}^\top\boldsymbol{z})] =: K^{\mathrm{LC}}(\boldsymbol{x},\boldsymbol{z})$$

almost surely.

Moreover, for every δ, $\varepsilon>0$ there exists $n_0(\delta,\varepsilon,R)\in\mathbb{N}$ such that for all $n\ge n_0$ and all $\boldsymbol{x}$, $\boldsymbol{z}\in\mathbb{R}^d$ with $\|\boldsymbol{x}\|$, $\|\boldsymbol{z}\|\le R$

$$\mathbb{P}\Big[\Big\|\frac{1}{n}\hat{K}_n(\boldsymbol{x},\boldsymbol{z}) - K^{\mathrm{LC}}(\boldsymbol{x},\boldsymbol{z})\Big\| < \varepsilon\Big] \ge 1-\delta.$$

Proof. Denote $\boldsymbol{x}^{(1)} = \boldsymbol{U}_0\boldsymbol{x}+\boldsymbol{b}_0\in\mathbb{R}^n$ and $\boldsymbol{z}^{(1)} = \boldsymbol{U}_0\boldsymbol{z}+\boldsymbol{b}_0\in\mathbb{R}^n$. Due to the initialization (11.5.3) and our assumptions on $\mathcal{D}(0,1)$, the components

$$x_i^{(1)} = \sum_{j=1}^d U_{0;ij}x_j \sim \boldsymbol{u}^\top\boldsymbol{x} \qquad i=1,\dots,n$$

are i.i.d. with finite pth moment (independent of n) for all $1\le p\le 8$. Due to the linear growth bound on σ and σ', the same holds for the $(\sigma(x_i^{(1)}))_{i=1}^n$ and the $(\sigma'(x_i^{(1)}))_{i=1}^n$. Similarly, the $(\sigma(z_i^{(1)}))_{i=1}^n$ and $(\sigma'(z_i^{(1)}))_{i=1}^n$ are collections of i.i.d. random variables with finite pth moment for all $1\le p\le 8$.

Denote $\tilde{v}_i = \sqrt{n}v_{0;i}$ such that $\tilde{v}_i \overset{\mathrm{iid}}{\sim}\mathcal{D}(0,1)$. By (11.5.2)

$$\frac{1}{n}\hat{K}_n(\boldsymbol{x},\boldsymbol{z}) = (1+\boldsymbol{x}^\top\boldsymbol{z})\frac{1}{n^2}\sum_{i=1}^n \tilde{v}_i^2\sigma'(x_i^{(1)})\sigma'(z_i^{(1)}) + \frac{1}{n}\sum_{i=1}^n\sigma(x_i^{(1)})\sigma(z_i^{(1)}) + \frac{1}{n}.$$

Since

$$\frac{1}{n}\sum_{i=1}^n \tilde{v}_i^2\sigma'(x_i^{(1)})\sigma'(z_i^{(1)}) \tag{11.5.4}$$

is an average over i.i.d. random variables with finite variance, the law of large numbers implies almost sure convergence of this expression towards

$$\begin{aligned}\mathbb{E}\big[\tilde{v}_i^2\sigma'(x_i^{(1)})\sigma'(z_i^{(1)})\big] &= \mathbb{E}[\tilde{v}_i^2]\mathbb{E}[\sigma'(x_i^{(1)})\sigma'(z_i^{(1)})] \\ &= \mathbb{E}[\sigma'(\boldsymbol{u}^\top \boldsymbol{x})\sigma'(\boldsymbol{u}^\top \boldsymbol{z})],\end{aligned}$$

where we used that $\tilde{v}_i^2$ is independent of $\sigma'(x_i^{(1)})\sigma'(z_i^{(1)})$. By the same argument

$$\frac{1}{n}\sum_{i=1}^{n}\sigma(x_i^{(1)})\sigma(z_i^{(1)}) \to \mathbb{E}[\sigma(\boldsymbol{u}^\top \boldsymbol{x})\sigma(\boldsymbol{u}^\top \boldsymbol{z})]$$

almost surely as $n \to \infty$. This shows the first statement.

The existence of n_0 follows similarly by an application of Theorem A.22. □

Example 11.17 (K^{LC} for ReLU)**.** Let $\sigma(x) = \max\{0, x\}$ and let $\mathcal{D}(0,1)$ be the standard normal distribution. For $\boldsymbol{x}$, $\boldsymbol{z} \in \mathbb{R}^d$ denote by

$$\theta = \arccos\left(\frac{\boldsymbol{x}^\top \boldsymbol{z}}{\|\boldsymbol{x}\|\|\boldsymbol{z}\|}\right)$$

the angle between these vectors. Then according to [36, Appendix A], it holds with $u_i \overset{\text{iid}}{\sim} \mathcal{D}(0,1)$, $i = 1, \dots, d$,

$$K^{\mathrm{LC}}(\boldsymbol{x}, \boldsymbol{z}) = \mathbb{E}[\sigma(\boldsymbol{u}^\top \boldsymbol{x})\sigma(\boldsymbol{u}^\top \boldsymbol{z})] = \frac{\|\boldsymbol{x}\|\|\boldsymbol{z}\|}{2\pi d}(\sin(\theta) + (\pi - \theta)\cos(\theta)).$$

11.5.3 Gradient descent

We now proceed similar as in [129, App. G], to show that Theorem 11.13 is applicable to the wide neural network (11.5.1) with high probability under random initialization (11.5.3). This will imply that gradient descent can find global minimizers when training wide neural networks. We work under the following assumptions on the activation function and training data.

Assumption 11.18. There exist $R < \infty$ and $0 < \lambda_{\min}^{\mathrm{LC}} \leq \lambda_{\max}^{\mathrm{LC}} < \infty$ such that

(a) for the activation function $\sigma : \mathbb{R} \to \mathbb{R}$ holds $|\sigma(0)|$, $\mathrm{Lip}(\sigma)$, $\mathrm{Lip}(\sigma') \leq R$,

(b) $\|\boldsymbol{x}_i\|$, $|y_i| \leq R$ for all training data $(\boldsymbol{x}_i, y_i) \in \mathbb{R}^d \times \mathbb{R}$, $i = 1, \dots, m$,

(c) the kernel matrix of the neural tangent kernel

$$(K^{\mathrm{LC}}(\boldsymbol{x}_i, \boldsymbol{x}_j))_{i,j=1}^m \in \mathbb{R}^{m\times m}$$

is regular and its eigenvalues belong to $[\lambda_{\min}^{\mathrm{LC}}, \lambda_{\max}^{\mathrm{LC}}]$.

We start by showing Assumption 11.12 (a) for the present setting. More precisely, we give bounds for the eigenvalues of the empirical tangent kernel.

Lemma 11.19. *Let Assumption 11.18 be satisfied. Then for every $\delta > 0$ there exists $n_0(\delta, \lambda_{\min}^{\mathrm{LC}}, m, R) \in \mathbb{R}$ such that for all $n \geq n_0$ with probability at least $1-\delta$ all eigenvalues of*

$$(\hat{K}_n(\boldsymbol{x}_i, \boldsymbol{x}_j))_{i,j=1}^m = \big(\langle \nabla_{\boldsymbol{w}} \Phi(\boldsymbol{x}_i, \boldsymbol{w}_0), \nabla_{\boldsymbol{w}} \Phi(\boldsymbol{x}_j, \boldsymbol{w}_0) \rangle \big)_{i,j=1}^m \in \mathbb{R}^{m \times m}$$

belong to $[n\lambda_{\min}^{\mathrm{LC}}/2, 2n\lambda_{\max}^{\mathrm{LC}}]$.

Proof. Denote $\hat{\boldsymbol{G}}_n := (\hat{K}_n(\boldsymbol{x}_i, \boldsymbol{x}_j))_{i,j=1}^m$ and $\boldsymbol{G}^{\mathrm{LC}} := (K^{\mathrm{LC}}(\boldsymbol{x}_i, \boldsymbol{x}_j))_{i,j=1}^m$. By Theorem 11.16, there exists n_0 such that for all $n \geq n_0$ holds with probability at least $1-\delta$ that

$$\left\| \boldsymbol{G}^{\mathrm{LC}} - \frac{1}{n} \hat{\boldsymbol{G}}_n \right\| \leq \frac{\lambda_{\min}^{\mathrm{LC}}}{2}.$$

Assuming this bound to hold

$$\frac{1}{n}\|\hat{\boldsymbol{G}}_n\| = \sup_{\substack{\boldsymbol{a} \in \mathbb{R}^m \\ \|\boldsymbol{a}\|=1}} \frac{1}{n} \|\hat{\boldsymbol{G}}_n \boldsymbol{a}\| \geq \inf_{\substack{\boldsymbol{a} \in \mathbb{R}^m \\ \|\boldsymbol{a}\|=1}} \|\boldsymbol{G}^{\mathrm{LC}} \boldsymbol{a}\| - \frac{\lambda_{\min}^{\mathrm{LC}}}{2} \geq \lambda_{\min}^{\mathrm{LC}} - \frac{\lambda_{\min}^{\mathrm{LC}}}{2} \geq \frac{\lambda_{\min}^{\mathrm{LC}}}{2},$$

where we have used that $\lambda_{\min}^{\mathrm{LC}}$ is the smallest eigenvalue, and thus singular value, of the symmetric positive definite matrix $\boldsymbol{G}^{\mathrm{LC}}$. This shows that the smallest eigenvalue of $\hat{\boldsymbol{G}}_n$ is larger or equal to $\lambda_{\min}^{\mathrm{LC}}/2$. Similarly, we conclude that the largest eigenvalue is bounded from above by $\lambda_{\max}^{\mathrm{LC}} + \lambda_{\min}^{\mathrm{LC}}/2 \leq \lambda_{\max}^{\mathrm{LC}}$. This concludes the proof. □

Next we check Assumption 11.12 (b). To this end we first bound the norm of a random matrix.

Lemma 11.20. *Let $\mathcal{D}(0,1)$ be as in Assumption 11.14, and let $\boldsymbol{W} \in \mathbb{R}^{n \times d}$ with $W_{ij} \overset{\text{iid}}{\sim} \mathcal{D}(0,1)$. Denote the fourth moment of $\mathcal{D}(0,1)$ by μ_4. Then*

$$\mathbb{P}\Big[\|\boldsymbol{W}\| \leq \sqrt{n(d+1)} \Big] \geq 1 - \frac{d\mu_4}{n}.$$

Proof. It holds

$$\|\boldsymbol{W}\| \leq \|\boldsymbol{W}\|_F = \Big(\sum_{i=1}^n \sum_{j=1}^d W_{ij}^2 \Big)^{1/2}.$$

The $\alpha_i := \sum_{j=1}^d W_{ij}^2$, $i = 1, \ldots, n$, are i.i.d. distributed with expectation d and finite variance dC, where $C \leq \mu_4$ is the variance of W_{11}^2. By Theorem A.22

$$\mathbb{P}\Big[\|\boldsymbol{W}\| > \sqrt{n(d+1)} \Big] \leq \mathbb{P}\Big[\frac{1}{n} \sum_{i=1}^n \alpha_i > d+1 \Big] \leq \mathbb{P}\Big[\Big| \frac{1}{n} \sum_{i=1}^n \alpha_i - d \Big| > 1 \Big] \leq \frac{d\mu_4}{n},$$

which concludes the proof. □

Lemma 11.21. *Let Assumption 11.18 (a) be satisfied with some constant R. Then there exists $M(R)$, and for all c, $\delta > 0$ there exists $n_0(c, d, \delta, R) \in \mathbb{N}$ such that for all $n \geq n_0$ it holds with probability at least $1 - \delta$*

$$\begin{aligned} \|\nabla_{\boldsymbol{w}} \Phi(\boldsymbol{x}, \boldsymbol{w})\| &\leq M\sqrt{n} && \text{for all } \boldsymbol{w} \in B_{cn^{-1/2}}(\boldsymbol{w}_0) \\ \|\nabla_{\boldsymbol{w}} \Phi(\boldsymbol{x}, \boldsymbol{w}) - \nabla_{\boldsymbol{w}} \Phi(\boldsymbol{x}, \boldsymbol{v})\| &\leq M\sqrt{n}\|\boldsymbol{w} - \boldsymbol{v}\| && \text{for all } \boldsymbol{w},\ \boldsymbol{v} \in B_{cn^{-1/2}}(\boldsymbol{w}_0) \end{aligned}$$

for all $\boldsymbol{x} \in \mathbb{R}^d$ with $\|\boldsymbol{x}\| \leq R$.

Proof. Due to the initialization (11.5.3), by Lemma 11.20 we can find $n_0(\delta, d)$ such that for all $n \geq n_0$ holds with probability at least $1 - \delta$ that

$$\|\boldsymbol{v}_0\| \leq 2 \qquad \text{and} \qquad \|\boldsymbol{U}_0\| \leq 2\sqrt{n}. \tag{11.5.5}$$

For the rest of this proof we fix arbitrary $\boldsymbol{x} \in \mathbb{R}^d$ and $n \geq n_0 \geq c^2$ such that

$$\|\boldsymbol{x}\| \leq R \qquad \text{and} \qquad n^{-1/2}c \leq 1.$$

We need to show that the claimed inequalities hold as long as (11.5.5) is satisfied. We will several times use that for all $\boldsymbol{p}$, $\boldsymbol{q} \in \mathbb{R}^n$

$$\|\boldsymbol{p} \odot \boldsymbol{q}\| \leq \|\boldsymbol{p}\|\|\boldsymbol{q}\| \qquad \text{and} \qquad \|\sigma(\boldsymbol{p})\| \leq R\sqrt{n} + R\|\boldsymbol{p}\|$$

since $|\sigma(x)| \leq R \cdot (1 + |x|)$. The same holds for σ'.

Step 1. We show the bound on the gradient. Fix

$$\boldsymbol{w} = (\boldsymbol{U}, \boldsymbol{b}, \boldsymbol{v}, c) \qquad \text{s.t.} \qquad \|\boldsymbol{w} - \boldsymbol{w}_0\| \leq cn^{-1/2}.$$

Using formula (11.5.2) for $\nabla_{\boldsymbol{b}} \Phi$ and the above inequalities

$$\begin{aligned} \|\nabla_{\boldsymbol{b}} \Phi(\boldsymbol{x}, \boldsymbol{w})\| &\leq \|\nabla_{\boldsymbol{b}} \Phi(\boldsymbol{x}, \boldsymbol{w}_0)\| + \|\nabla_{\boldsymbol{b}} \Phi(\boldsymbol{x}, \boldsymbol{w}) - \nabla_{\boldsymbol{b}} \Phi(\boldsymbol{x}, \boldsymbol{w}_0)\| \\ &= \|\boldsymbol{v}_0 \odot \sigma'(\boldsymbol{U}_0\boldsymbol{x})\| + \|\boldsymbol{v} \odot \sigma'(\boldsymbol{U}\boldsymbol{x} + \boldsymbol{b}) - \boldsymbol{v}_0 \odot \sigma'(\boldsymbol{U}_0\boldsymbol{x})\| \\ &\leq 2(R\sqrt{n} + 2R^2\sqrt{n}) + \|\boldsymbol{v} \odot \sigma'(\boldsymbol{U}\boldsymbol{x} + \boldsymbol{b}) - \boldsymbol{v}_0 \odot \sigma'(\boldsymbol{U}_0\boldsymbol{x})\|. \end{aligned} \tag{11.5.6}$$

Due to

$$\|\boldsymbol{U}\| \leq \|\boldsymbol{U}_0\| + \|\boldsymbol{U}_0 - \boldsymbol{U}\|_F \leq 2\sqrt{n} + cn^{-1/2} \leq 3\sqrt{n}, \tag{11.5.7}$$

the last norm in (11.5.6) is bounded by

$$\begin{aligned} &\|(\boldsymbol{v} - \boldsymbol{v}_0) \odot \sigma'(\boldsymbol{U}\boldsymbol{x} + \boldsymbol{b})\| + \|\boldsymbol{v}_0 \odot (\sigma'(\boldsymbol{U}\boldsymbol{x} + \boldsymbol{b}) - \sigma'(\boldsymbol{U}_0\boldsymbol{x}))\| \\ &\quad \leq cn^{-1/2}(R\sqrt{n} + R \cdot (\|\boldsymbol{U}\|\|\boldsymbol{x}\| + \|\boldsymbol{b}\|)) + 2R \cdot (\|\boldsymbol{U} - \boldsymbol{U}_0\|\|\boldsymbol{x}\| + \|\boldsymbol{b}\|) \\ &\quad \leq R\sqrt{n} + 3\sqrt{n}R^2 + cn^{-1/2}R + 2R \cdot (cn^{-1/2}R + cn^{-1/2}) \\ &\quad \leq \sqrt{n}(4R + 5R^2) \end{aligned}$$

and therefore

$$\|\nabla_{\boldsymbol{b}} \Phi(\boldsymbol{x}, \boldsymbol{w})\| \leq \sqrt{n}(6R + 9R^2).$$

For the gradient with respect to $\boldsymbol{U}$ we use $\nabla_{\boldsymbol{U}}\Phi(\boldsymbol{x},\boldsymbol{w}) = \nabla_{\boldsymbol{b}}\Phi(\boldsymbol{x},\boldsymbol{w})\boldsymbol{x}^\top$, so that

$$\|\nabla_{\boldsymbol{U}}\Phi(\boldsymbol{x},\boldsymbol{w})\|_F = \|\nabla_{\boldsymbol{b}}\Phi(\boldsymbol{x},\boldsymbol{w})\boldsymbol{x}^\top\|_F = \|\nabla_{\boldsymbol{b}}\Phi(\boldsymbol{x},\boldsymbol{w})\|\,\|\boldsymbol{x}\| \le \sqrt{n}(6R^2+9R^3).$$

Next

$$\begin{aligned}\|\nabla_{\boldsymbol{v}}\Phi(\boldsymbol{x},\boldsymbol{w})\| &= \|\sigma(\boldsymbol{U}\boldsymbol{x}+\boldsymbol{b})\| \\ &\le R\sqrt{n} + R\|\boldsymbol{U}\boldsymbol{x}+\boldsymbol{b}\| \\ &\le R\sqrt{n} + R\cdot(3\sqrt{n}R + cn^{-1/2}) \\ &\le \sqrt{n}(2R+3R^2),\end{aligned}$$

and finally $\nabla_c\Phi(\boldsymbol{x},\boldsymbol{w}) = 1$. In all, with $M_1(R) := (1+8R+12R^2)$

$$\|\nabla_{\boldsymbol{w}}\Phi(\boldsymbol{x},\tilde{\boldsymbol{w}})\| \le \sqrt{n}M_1(R).$$

Step 2. We show Lipschitz continuity. Fix

$$\boldsymbol{w} = (\boldsymbol{U},\boldsymbol{b},\boldsymbol{v},c) \qquad \text{and} \qquad \tilde{\boldsymbol{w}} = (\tilde{\boldsymbol{U}},\tilde{\boldsymbol{b}},\tilde{\boldsymbol{v}},\tilde{c})$$

such that $\|\boldsymbol{w}-\boldsymbol{w}_0\|$, $\|\tilde{\boldsymbol{w}}-\boldsymbol{w}_0\| \le cn^{-1/2}$. Then

$$\|\nabla_{\boldsymbol{b}}\Phi(\boldsymbol{x},\boldsymbol{w}) - \nabla_{\boldsymbol{b}}\Phi(\boldsymbol{x},\tilde{\boldsymbol{w}})\| = \|\boldsymbol{v}\odot\sigma'(\boldsymbol{U}\boldsymbol{x}+\boldsymbol{b}) - \tilde{\boldsymbol{v}}\odot\sigma'(\tilde{\boldsymbol{U}}\boldsymbol{x}+\tilde{\boldsymbol{b}})\|.$$

Using $\|\tilde{\boldsymbol{v}}\| \le \|\boldsymbol{v}_0\| + cn^{-1/2} \le 3$ and (11.5.7), this term is bounded by

$$\begin{aligned}&\|(\boldsymbol{v}-\tilde{\boldsymbol{v}})\odot\sigma'(\boldsymbol{U}\boldsymbol{x}+\boldsymbol{b})\| + \|\tilde{\boldsymbol{v}}\odot(\sigma'(\boldsymbol{U}\boldsymbol{x}+\boldsymbol{b}) - \sigma'(\tilde{\boldsymbol{U}}\boldsymbol{x}+\tilde{\boldsymbol{b}}))\| \\ &\quad\le \|\boldsymbol{v}-\tilde{\boldsymbol{v}}\|(R\sqrt{n} + R\cdot(\|\boldsymbol{U}\|\|\boldsymbol{x}\| + \|\boldsymbol{b}\|)) + 3R\cdot(\|\boldsymbol{x}\|\|\boldsymbol{U}-\tilde{\boldsymbol{U}}\| + \|\boldsymbol{b}-\tilde{\boldsymbol{b}}\|) \\ &\quad\le \|\boldsymbol{w}-\tilde{\boldsymbol{w}}\|\sqrt{n}(5R+6R^2).\end{aligned}$$

For $\nabla_{\boldsymbol{U}}\Phi(\boldsymbol{x},\boldsymbol{w})$ we obtain similar as in Step 1

$$\begin{aligned}\|\nabla_{\boldsymbol{U}}\Phi(\boldsymbol{x},\boldsymbol{w}) - \nabla_{\boldsymbol{U}}\Phi(\boldsymbol{x},\tilde{\boldsymbol{w}})\|_F &= \|\boldsymbol{x}\|\|\nabla_{\boldsymbol{b}}\Phi(\boldsymbol{x},\boldsymbol{w}) - \nabla_{\boldsymbol{b}}\Phi(\boldsymbol{x},\tilde{\boldsymbol{w}})\| \\ &\le \|\boldsymbol{w}-\tilde{\boldsymbol{w}}\|\sqrt{n}(5R^2+6R^3).\end{aligned}$$

Next

$$\begin{aligned}\|\nabla_{\boldsymbol{v}}\Phi(\boldsymbol{x},\boldsymbol{w}) - \nabla_{\boldsymbol{v}}\Phi(\boldsymbol{x},\tilde{\boldsymbol{w}})\| &= \|\sigma(\boldsymbol{U}\boldsymbol{x}+\boldsymbol{b}) - \sigma(\tilde{\boldsymbol{U}}\boldsymbol{x}-\tilde{\boldsymbol{b}})\| \\ &\le R\cdot(\|\boldsymbol{U}-\tilde{\boldsymbol{U}}\|\|\boldsymbol{x}\| + \|\boldsymbol{b}-\tilde{\boldsymbol{b}}\|) \\ &\le \|\boldsymbol{w}-\tilde{\boldsymbol{w}}\|(R^2+R)\end{aligned}$$

and finally $\nabla_c\Phi(\boldsymbol{x},\boldsymbol{w}) = 1$ is constant. With $M_2(R) := R+6R^2+6R^3$ this shows

$$\|\nabla_{\boldsymbol{w}}\Phi(\boldsymbol{x},\boldsymbol{w}) - \nabla_{\boldsymbol{w}}\Phi(\boldsymbol{x},\tilde{\boldsymbol{w}})\| \le \sqrt{n}M_2(R)\|\boldsymbol{w}-\tilde{\boldsymbol{w}}\|.$$

In all, this concludes the proof with $M(R) := \max\{M_1(R), M_2(R)\}$. □

Before coming to the main result of this section, we first show that the initial error $f(\boldsymbol{w}_0)$ remains bounded with high probability.

Lemma 11.22. *Let Assumption 11.18 (a), (b) be satisfied. Then for every $\delta > 0$ exists $R_0(\delta, m, R) > 0$ such that for all $n \in \mathbb{N}$*

$$\mathbb{P}[f(\boldsymbol{w}_0) \leq R_0] \geq 1 - \delta.$$

Proof. Let $i \in \{1, \dots, m\}$, and set $\boldsymbol{\alpha} := \boldsymbol{U}_0 \boldsymbol{x}_i$ and $\tilde{v}_j := \sqrt{n} v_{0;j}$ for $j = 1, \dots, n$, so that $\tilde{v}_j \overset{\text{iid}}{\sim} \mathcal{D}(0, 1)$. Then

$$\Phi(\boldsymbol{x}_i, \boldsymbol{w}_0) = \frac{1}{\sqrt{n}} \sum_{j=1}^{n} \tilde{v}_j \sigma(\alpha_j).$$

By Assumption 11.14 and (11.5.3), the $\tilde{v}_j \sigma(\alpha_j)$, $j = 1, \dots, n$, are i.i.d. centered random variables with finite variance bounded by a constant $C(R)$ independent of n. Thus the variance of $\Phi(\boldsymbol{x}_i, \boldsymbol{w}_0)$ is also bounded by $C(R)$. By Chebyshev's inequality, see Lemma A.21, for every $k > 0$

$$\mathbb{P}[|\Phi(\boldsymbol{x}_i, \boldsymbol{w}_0)| \geq k\sqrt{C}] \leq \frac{1}{k^2}.$$

Setting $k = \sqrt{m/\delta}$

$$\begin{aligned} \mathbb{P}\Big[\sum_{i=1}^{m} |\Phi(\boldsymbol{x}_i, \boldsymbol{w}_0) - y_i|^2 \geq m(k\sqrt{C} + R)^2 \Big] &\leq \sum_{i=1}^{m} \mathbb{P}\Big[|\Phi(\boldsymbol{x}_i, \boldsymbol{w}_0) - y_i| \geq k\sqrt{C} + R \Big] \\ &\leq \sum_{i=1}^{m} \mathbb{P}\Big[|\Phi(\boldsymbol{x}_i, \boldsymbol{w}_0)| \geq k\sqrt{C} \Big] \leq \delta, \end{aligned}$$

which shows the claim with $R_0 = m \cdot (\sqrt{Cm/\delta} + R)^2$. □

The next theorem is the main result of this section. It states that in the present setting gradient descent converges to a global minimizer and the limiting network achieves zero loss, i.e. interpolates the data. Moreover, during training the network weights remain close to initialization if the network width n is large.

Theorem 11.23. *Let Assumption 11.18 be satisfied, and let the parameters $\boldsymbol{w}_0$ of the neural network Φ in* (11.5.1) *be initialized according to* (11.5.3)*. Fix a learning rate*

$$h < \frac{2}{\lambda_{\min}^{\mathrm{LC}} + 4\lambda_{\max}^{\mathrm{LC}}} \frac{1}{n}$$

and with the objective function (11.0.1b) *let for all $k \in \mathbb{N}$*

$$\boldsymbol{w}_{k+1} = \boldsymbol{w}_k - h \nabla f(\boldsymbol{w}_k).$$

Then for every $\delta > 0$ there exist $C > 0$, $n_0 \in \mathbb{N}$ such that for all $n \geq n_0$ holds with probability at least $1 - \delta$ that for all $k \in \mathbb{N}$

$$\|\boldsymbol{w}_k - \boldsymbol{w}_0\| \leq \frac{C}{\sqrt{n}}$$
$$f(\boldsymbol{w}_k) \leq C\Big(1 - \frac{hn}{2\lambda_{\min}^{\mathrm{LC}}}\Big)^{2k}.$$

Proof. We wish to apply Theorem 11.13, which requires Assumption 11.12 to be satisfied. By Lemma 11.19, 11.21 and 11.22, for every $c > 0$ we can find n_0 such that for all $n \geq n_0$ with probability at least $1 - \delta$ we have $\sqrt{f(\boldsymbol{w}_0)} \leq \sqrt{R_0}$ and Assumption 11.12 (a), (b) holds with the values

$$L = M\sqrt{n}, \quad U = M\sqrt{n}, \quad r = cn^{-1/2}, \quad \lambda_{\min} = \frac{n\lambda_{\min}^{\mathrm{LC}}}{2}, \quad \lambda_{\max} = 2n\lambda_{\max}^{\mathrm{LC}}.$$

For Assumption 11.12 (c), it suffices that

$$M\sqrt{n} \leq \frac{n^2(\lambda_{\min}^{\mathrm{LC}}/2)^2}{12m^{3/2}M^2n\sqrt{R_0}} \qquad \text{and} \qquad cn^{-1/2} \geq \frac{2mM\sqrt{n}}{n}\sqrt{R_0}.$$

Choosing $c > 0$ and n large enough, the inequalities hold. The statement is now a direct consequence of Theorem 11.13. □

11.5.4 Proximity to linearized model

The analysis thus far was based on the linearization Φ^{lin} describing the behaviour of the full network Φ well in a neighbourhood of the initial parameters $\boldsymbol{w}_0$. Moreover, Theorem 11.23 states that the parameters remain in an $O(n^{-1/2})$ neighbourhood of $\boldsymbol{w}_0$ during training. This suggests that the trained full model $\lim_{k\to\infty} \Phi(\boldsymbol{x}, \boldsymbol{w}_k)$ yields predictions similar to the trained linearized model.

To describe this phenomenon, we adopt again the notations $\Phi^{\mathrm{lin}} : \mathbb{R}^d \times \mathbb{R}^n \to \mathbb{R}$ and f^{lin} from (11.3.1) and (11.3.3). Initializing $\boldsymbol{w}_0$ according to (11.5.3) and setting $\boldsymbol{p}_0 = \boldsymbol{w}_0$, gradient descent computes the parameter updates

$$\boldsymbol{w}_{k+1} = \boldsymbol{w}_k - h\nabla_{\boldsymbol{w}} f(\boldsymbol{w}_k), \qquad \boldsymbol{p}_{k+1} = \boldsymbol{p}_k - h\nabla_{\boldsymbol{w}} f^{\mathrm{lin}}(\boldsymbol{p}_k)$$

for the full and linearized models, respectively. Let us consider the dynamics of the prediction of the network on the training data. Writing

$$\Phi(\boldsymbol{X}, \boldsymbol{w}) := (\Phi(\boldsymbol{x}_i, \boldsymbol{w}))_{i=1}^m \in \mathbb{R}^m \qquad \text{such that} \qquad \nabla_{\boldsymbol{w}}\Phi(\boldsymbol{X}, \boldsymbol{w}) \in \mathbb{R}^{m\times n}$$

it holds

$$\nabla_{\boldsymbol{w}} f(\boldsymbol{w}) = \nabla_{\boldsymbol{w}} \|\Phi(\boldsymbol{X}, \boldsymbol{w}) - \boldsymbol{y}\|^2 = 2\nabla_{\boldsymbol{w}}\Phi(\boldsymbol{X}, \boldsymbol{w})^\top (\Phi(\boldsymbol{X}, \boldsymbol{w}) - \boldsymbol{y}).$$

Thus for the full model

$$\begin{aligned}\Phi(\boldsymbol{X}, \boldsymbol{w}_{k+1}) &= \Phi(\boldsymbol{X}, \boldsymbol{w}_k) + \nabla_{\boldsymbol{w}}\Phi(\boldsymbol{X}, \tilde{\boldsymbol{w}}_k)(\boldsymbol{w}_{k+1} - \boldsymbol{w}_k) \\ &= \Phi(\boldsymbol{X}, \boldsymbol{w}_k) - 2h\nabla_{\boldsymbol{w}}\Phi(\boldsymbol{X}, \tilde{\boldsymbol{w}}_k)\nabla_{\boldsymbol{w}}\Phi(\boldsymbol{X}, \boldsymbol{w}_k)^\top (\Phi(\boldsymbol{X}, \boldsymbol{w}_k) - \boldsymbol{y}), \end{aligned} \tag{11.5.8}$$

where $\tilde{\boldsymbol{w}}_k$ is in the convex hull of $\boldsymbol{w}_k$ and $\boldsymbol{w}_{k+1}$.

Similarly, for the linearized model with (cp. (11.3.1))

$$\Phi^{\mathrm{lin}}(\boldsymbol{X}, \boldsymbol{w}) := (\Phi^{\mathrm{lin}}(\boldsymbol{x}_i, \boldsymbol{w}))_{i=1}^m \in \mathbb{R}^m \qquad \text{and} \qquad \nabla_{\boldsymbol{p}} \Phi^{\mathrm{lin}}(\boldsymbol{X}, \boldsymbol{p}) = \nabla_{\boldsymbol{w}} \Phi(\boldsymbol{X}, \boldsymbol{w}_0) \in \mathbb{R}^{m \times n}$$

such that

$$\nabla_{\boldsymbol{p}} f^{\mathrm{lin}}(\boldsymbol{p}) = \nabla_{\boldsymbol{p}} \|\Phi^{\mathrm{lin}}(\boldsymbol{X}, \boldsymbol{p}) - \boldsymbol{y}\|^2 = 2 \nabla_{\boldsymbol{w}} \Phi(\boldsymbol{X}, \boldsymbol{w}_0)^\top (\Phi^{\mathrm{lin}}(\boldsymbol{X}, \boldsymbol{p}) - \boldsymbol{y})$$

and

$$\begin{aligned} \Phi^{\mathrm{lin}}(\boldsymbol{X}, \boldsymbol{p}_{k+1}) &= \Phi^{\mathrm{lin}}(\boldsymbol{X}, \boldsymbol{p}_k) + \nabla_{\boldsymbol{p}} \Phi^{\mathrm{lin}}(\boldsymbol{X}, \boldsymbol{p}_0)(\boldsymbol{p}_{k+1} - \boldsymbol{p}_k) \\ &= \Phi^{\mathrm{lin}}(\boldsymbol{X}, \boldsymbol{p}_k) - 2h \nabla_{\boldsymbol{w}} \Phi(\boldsymbol{X}, \boldsymbol{w}_0) \nabla_{\boldsymbol{w}} \Phi(\boldsymbol{X}, \boldsymbol{w}_0)^\top (\Phi^{\mathrm{lin}}(\boldsymbol{X}, \boldsymbol{p}_k) - \boldsymbol{y}). \end{aligned} \tag{11.5.9}$$

Remark 11.24. From (11.5.9) it is easy to see that with $\boldsymbol{A} := 2h \nabla_{\boldsymbol{w}} \Phi(\boldsymbol{X}, \boldsymbol{w}_0) \nabla_{\boldsymbol{w}} \Phi(\boldsymbol{X}, \boldsymbol{w}_0)^\top$ and $\boldsymbol{B} := \boldsymbol{I}_m - \boldsymbol{A}$ holds the explicit formula

$$\Phi^{\mathrm{lin}}(\boldsymbol{X}, \boldsymbol{p}_k) = \boldsymbol{B}^k \Phi^{\mathrm{lin}}(\boldsymbol{X}, \boldsymbol{p}_0) + \sum_{j=0}^{k-1} \boldsymbol{B}^k \boldsymbol{A} \boldsymbol{y}$$

for the prediction of the linear model in step k. Note that if $\boldsymbol{A}$ is regular and h is small enough, then $\boldsymbol{B}^k$ converges to the zero matrix as $k \to \infty$ and $\sum_{j=0}^{\infty} \boldsymbol{B}^k = \boldsymbol{A}^{-1}$ since this is a Neumann series.

Comparing the two dynamics (11.5.8) and (11.5.9), the difference only lies in the two $\mathbb{R}^{m \times m}$ matrices

$$2h \nabla_{\boldsymbol{w}} \Phi(\boldsymbol{X}, \tilde{\boldsymbol{w}}_k) \nabla_{\boldsymbol{w}} \Phi(\boldsymbol{X}, \boldsymbol{w}_k)^\top \qquad \text{and} \qquad 2h \nabla_{\boldsymbol{w}} \Phi(\boldsymbol{X}, \boldsymbol{w}_0) \nabla_{\boldsymbol{w}} \Phi(\boldsymbol{X}, \boldsymbol{w}_0)^\top.$$

Recall that the step size h in Theorem 11.23 scales like $1/n$.

Proposition 11.25. *Consider the setting of Theorem 11.23. Then there exists $C < \infty$, and for every $\delta > 0$ there exists n_0 such that for all $n \geq n_0$ holds with probability at least $1 - \delta$ that for all $k \in \mathbb{N}$*

$$\frac{1}{n} \|\nabla_{\boldsymbol{w}} \Phi(\boldsymbol{X}, \tilde{\boldsymbol{w}}_k) \nabla_{\boldsymbol{w}} \Phi(\boldsymbol{X}, \boldsymbol{w}_k)^\top - \nabla_{\boldsymbol{p}} \Phi(\boldsymbol{X}, \boldsymbol{p}_0) \nabla_{\boldsymbol{p}} \Phi(\boldsymbol{X}, \boldsymbol{p}_0)^\top\| \leq C n^{-1/2}.$$

Proof. Consider the setting of the proof of Theorem 11.23. Then for every $k \in \mathbb{N}$ holds $\|\boldsymbol{w}_k - \boldsymbol{w}_0\| \leq r$ and thus also $\|\tilde{\boldsymbol{w}}_k - \boldsymbol{w}_0\| \leq r$, where $r = cn^{-1/2}$. Thus Lemma 11.21 implies the norm to be bounded by

$$\begin{aligned} &\frac{1}{n} \|\nabla_{\boldsymbol{w}} \Phi(\boldsymbol{X}, \tilde{\boldsymbol{w}}_k) - \nabla_{\boldsymbol{p}} \Phi(\boldsymbol{X}, \boldsymbol{p}_0)\| \|\nabla_{\boldsymbol{w}} \Phi(\boldsymbol{X}, \boldsymbol{w}_k)^\top\| + \\ &\qquad \frac{1}{n} \|\nabla_{\boldsymbol{p}} \Phi(\boldsymbol{X}, \boldsymbol{p}_0)\| \|\nabla_{\boldsymbol{w}} \Phi(\boldsymbol{X}, \boldsymbol{w}_k)^\top - \nabla_{\boldsymbol{p}} \Phi(\boldsymbol{X}, \boldsymbol{p}_0)^\top\| \\ &\leq mM(\|\tilde{\boldsymbol{w}}_k - \boldsymbol{p}_0\| + \|\boldsymbol{w}_k - \boldsymbol{p}_0\|) \leq cmMn^{-1/2} \end{aligned}$$

which gives the statement. □

By Proposition 11.25 the two matrices driving the dynamics (11.5.8) and (11.5.9) remain in an $O(n^{-1/2})$ neighbourhood of each other throughout training. This allows to show the following proposition, which states that the prediction function learned by the network gets arbitrarily close to the one learned by the linearized version in the limit $n \to \infty$. The proof, which we omit, is based on Grönwall's inequality. See [105, 129].

Proposition 11.26. *Consider the setting of Theorem 11.23. Then there exists $C < \infty$, and for every $\delta > 0$ there exists n_0 such that for all $n \geq n_0$ holds with probability at least $1 - \delta$ that for all $\|\boldsymbol{x}\| \leq 1$*

$$\sup_{k \in \mathbb{N}} |\Phi(\boldsymbol{x}, \boldsymbol{w}_k) - \Phi^{\text{lin}}(\boldsymbol{x}, \boldsymbol{p}_k)| \leq C n^{-1/2}.$$

11.5.5 Connection to Gaussian processes

In the previous section, we established that for large widths, the trained neural network mirrors the behaviour of the trained linearized model, which itself is closely connected to kernel least-squares with the neural tangent kernel. Yet, as pointed out in Remark 11.4, the obtained model still strongly depends on the choice of random initialization $\boldsymbol{w}_0 \in \mathbb{R}^n$. We should thus understand both the model at initialization $\boldsymbol{x} \mapsto \Phi(\boldsymbol{x}, \boldsymbol{w}_0)$ and the model after training $\boldsymbol{x} \mapsto \Phi(\boldsymbol{x}, \boldsymbol{w}_k)$, as random draws of a certain distribution over functions. To make this precise, let us introduce Gaussian processes.

Definition 11.27. Let $(\Omega, \mathbb{P})$ be a probability space, and let $g : \mathbb{R}^d \times \Omega \to \mathbb{R}$. We call g a **Gaussian process** with mean function $m : \mathbb{R}^d \to \mathbb{R}$ and covariance function $c : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ if

(a) for each $\boldsymbol{x} \in \mathbb{R}^d$ holds $\omega \mapsto g(\boldsymbol{x}, \omega)$ is a random variable,

(b) for all $k \in \mathbb{N}$ and all $\boldsymbol{x}_1, \dots, \boldsymbol{x}_k \in \mathbb{R}^d$ the random variables $g(\boldsymbol{x}_1, \cdot), \dots, g(\boldsymbol{x}_k, \cdot)$ have a joint Gaussian distribution such that

$$(g(\boldsymbol{x}_1, \omega), \dots, g(\boldsymbol{x}_k, \omega)) \sim \mathrm{N}\Big(m(\boldsymbol{x}_i)_{i=1}^k, (c(\boldsymbol{x}_i, \boldsymbol{x}_j))_{i,j=1}^k\Big).$$

In words, g is a Gaussian process, if $\omega \mapsto g(\boldsymbol{x}, \omega)$ defines a collection of random variables indexed over $\boldsymbol{x} \in \mathbb{R}^d$, such that the joint distribution of $(g(\boldsymbol{x}_1, \cdot))_{j=1}^n$ is a Gaussian whose mean and variance are determined by m and c respectively. Fixing $\omega \in \Omega$, we can then interpret $\boldsymbol{x} \mapsto g(\boldsymbol{x}, \omega)$ as a random draw from a distribution over functions.

As first observed in [155], certain neural networks at initialization tend to Gaussian processes in the infinite width limit.

Proposition 11.28. *Consider depth-n networks Φ_n as in* (11.5.1) *with initialization* (11.5.3)*, and define with $u_i \overset{\text{iid}}{\sim} \mathcal{D}(0, 1/d)$, $i = 1, \dots, d$,*

$$c(\boldsymbol{x}, \boldsymbol{z}) := \mathbb{E}[\sigma(\boldsymbol{u}^\top \boldsymbol{x})\sigma(\boldsymbol{u}^\top \boldsymbol{z})] \qquad \text{for all } \boldsymbol{x},\ \boldsymbol{z} \in \mathbb{R}^d.$$

Then for all distinct $\boldsymbol{x}_1, \dots, \boldsymbol{x}_k \in \mathbb{R}^d$ it holds that

$$\lim_{n\to\infty} (\Phi_n(\boldsymbol{x}_1, \boldsymbol{w}_0), \dots, \Phi_n(\boldsymbol{x}_k, \boldsymbol{w}_0)) \sim \mathrm{N}(\boldsymbol{0}, (c(\boldsymbol{x}_i, \boldsymbol{x}_j))_{i,j=1}^k)$$

with weak convergence.

Proof. Set $\tilde{v}_i := \sqrt{n} v_{0,i}$ and $\tilde{\boldsymbol{u}}_i = (U_{0,i1}, \dots, U_{0,id}) \in \mathbb{R}^d$, so that $\tilde{v}_i \overset{\text{iid}}{\sim} \mathcal{D}(0, 1)$, and the $\tilde{\boldsymbol{u}}_i \in \mathbb{R}^d$ are also i.i.d., with each component distributed according to $\mathcal{D}(0, 1/d)$.

Then for any $\boldsymbol{x}_1, \dots, \boldsymbol{x}_k$

$$\boldsymbol{Z}_i := \begin{pmatrix} \tilde{v}_i \sigma(\tilde{\boldsymbol{u}}_i^\top \boldsymbol{x}_1) \\ \vdots \\ \tilde{v}_i \sigma(\tilde{\boldsymbol{u}}_i^\top \boldsymbol{x}_k) \end{pmatrix} \in \mathbb{R}^k \qquad i = 1, \dots, n,$$

defines n centered i.i.d. vectors in $\mathbb{R}^k$. By the central limit theorem, see Theorem A.24,

$$\begin{pmatrix} \Phi(\boldsymbol{x}_1, \boldsymbol{w}_0) \\ \vdots \\ \Phi(\boldsymbol{x}_k, \boldsymbol{w}_0) \end{pmatrix} = \frac{1}{\sqrt{n}} \sum_{j=1}^n \boldsymbol{Z}_i$$

converges weakly to $\mathrm{N}(\boldsymbol{0}, \boldsymbol{C})$, where

$$C_{ij} = \mathbb{E}[\tilde{v}_1^2 \sigma(\tilde{\boldsymbol{u}}_1^\top \boldsymbol{x}_i)\sigma(\tilde{\boldsymbol{u}}_1^\top \boldsymbol{x}_j)] = \mathbb{E}[\sigma(\tilde{\boldsymbol{u}}_1^\top \boldsymbol{x}_i)\sigma(\tilde{\boldsymbol{u}}_1^\top \boldsymbol{x}_j)].$$

This concludes the proof. □

In the sense of Proposition 11.28, the network $\Phi(\boldsymbol{x}, \boldsymbol{w}_0)$ converges to a Gaussian process as the width n tends to infinity. Using the explicit dynamics of the linearized network outlined in Remark 11.24, one can show that the linearized network after training also corresponds to a Gaussian process (for some mean and covariance function depending on the data, the architecture, and the initialization). As the full and linearized models converge in the infinite width limit, we can infer that wide networks post-training resemble draws from a Gaussian process, see [129, Sec. 2.3.1] and [45].

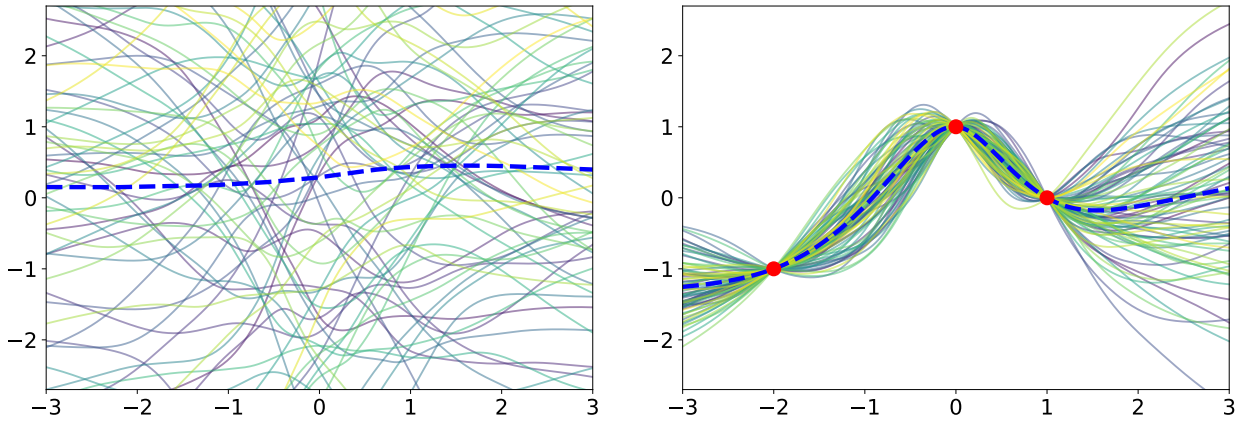
Rather than delving into the technical details of such statements, in Figure 11.3 we plot 80 different realizations of a neural network before and after training, i.e.

$$\boldsymbol{x} \mapsto \Phi(\boldsymbol{x}, \boldsymbol{w}_0) \qquad \text{and} \qquad \boldsymbol{x} \mapsto \Phi(\boldsymbol{x}, \boldsymbol{w}_k). \tag{11.5.10}$$

We chose the architecture as (11.5.1) with activation function $\sigma = \arctan(x)$, width $n = 250$ and initialization

$$U_{0;ij} \overset{\text{iid}}{\sim} \mathrm{N}\Big(0, \frac{3}{d}\Big), \qquad v_{0;i} \overset{\text{iid}}{\sim} \mathrm{N}\Big(0, \frac{3}{n}\Big), \qquad b_{0;i},\ c_0 \overset{\text{iid}}{\sim} \mathrm{N}(0, 2). \tag{11.5.11}$$

The network was trained on a dataset of size $m = 3$ with $k = 1000$ steps of gradient descent and constant step size $h = 1/n$. Before training, the network's outputs resemble random draws from a Gaussian process with a constant zero mean function. Post-training, the outputs show minimal variance at the data points, since they essentially interpolate the data, cp. Remark 11.4 and (11.2.4). They exhibit increased variance further from these points, with the precise amount depending on the initialization variance chosen in (11.5.11).



(a) before training (b) after training

Figure 11.3: 80 realizations of a neural network at initialization (a) and after training on the red data points (b). The blue dashed line shows the mean. Figure based on [129, Fig. 2].

11.6 Normalized initialization

Consider the gradient $\nabla_{\boldsymbol{w}}\Phi(\boldsymbol{x}, \boldsymbol{w}_0)$ as in (11.5.2) with LeCun initialization. Since the components of $\boldsymbol{v}$ behave like $v_i \overset{\text{iid}}{\sim} \mathcal{D}(0, 1/n)$, it is easy to check that in terms of the width n

$$\begin{aligned}
\mathbb{E}[\|\nabla_{\boldsymbol{U}}\Phi(\boldsymbol{x}, \boldsymbol{w}_0)\|] &= \mathbb{E}[\|(\boldsymbol{v} \odot \sigma'(\boldsymbol{U}\boldsymbol{x} + \boldsymbol{b}))\boldsymbol{x}^\top\|] &&= O(1)\\
\mathbb{E}[\|\nabla_{\boldsymbol{b}}\Phi(\boldsymbol{x}, \boldsymbol{w}_0)\|] &= \mathbb{E}[\|\boldsymbol{v} \odot \sigma'(\boldsymbol{U}\boldsymbol{x} + \boldsymbol{b})\|] &&= O(1)\\
\mathbb{E}[\|\nabla_{\boldsymbol{v}}\Phi(\boldsymbol{x}, \boldsymbol{w}_0)\|] &= \mathbb{E}[\|\sigma(\boldsymbol{U}\boldsymbol{x} + \boldsymbol{b})\|] &&= O(n)\\
\mathbb{E}[\|\nabla_{c}\Phi(\boldsymbol{x}, \boldsymbol{w}_0)\|] &= \mathbb{E}[|1|] &&= O(1).
\end{aligned}$$

As a result of this different scaling, gradient descent with step width $O(n^{-1})$ as in Theorem 11.23, will primarily train the weigths $\boldsymbol{v}$ in the output layer, and will barely move the remaining parameters $\boldsymbol{U}$, $\boldsymbol{b}$, and c. This is also reflected in the expression for the obtained kernel K^{LC} computed in Theorem 11.16, which corresponds to the contribution of the term $\langle \nabla_{\boldsymbol{v}}\Phi, \nabla_{\boldsymbol{v}}\Phi \rangle$.

Remark 11.29. For optimization methods such as ADAM, which scale each component of the gradient individually, the same does not hold in general.

LeCun initialization aims to normalize the variance of the output of all nodes at initialization (the forward dynamics). To also normalize the variance of the gradients (the backward dynamics), in this section we shortly dicuss a different architecture and initialization, consistent with the one used in the original NTK paper [105].

11.6.1 Architecture

Let $\Phi : \mathbb{R}^d \to \mathbb{R}$ be a depth-one neural network

$$\Phi(\boldsymbol{x}, \boldsymbol{w}) = \frac{1}{\sqrt{n}} \boldsymbol{v}^\top \sigma\Big(\frac{1}{\sqrt{d}} \boldsymbol{U}\boldsymbol{x} + \boldsymbol{b}\Big) + c, \tag{11.6.1}$$

with input $\boldsymbol{x} \in \mathbb{R}^d$ and parameters $\boldsymbol{U} \in \mathbb{R}^{n \times d}$, $\boldsymbol{v} \in \mathbb{R}^n$, $\boldsymbol{b} \in \mathbb{R}^n$ and $c \in \mathbb{R}$. We initialize the weights randomly according to $\boldsymbol{w}_0 = (\boldsymbol{U}_0, \boldsymbol{b}_0, \boldsymbol{v}_0, c_0)$ with parameters

$$U_{0;ij} \overset{\text{iid}}{\sim} \mathcal{D}(0,1), \qquad v_{0;i} \overset{\text{iid}}{\sim} \mathcal{D}(0,1), \qquad b_{0;i},\ c_0 = 0. \tag{11.6.2}$$

At initialization, (11.6.1), (11.6.2) is equivalent to (11.5.1), (11.5.3). However, for the gradient we obtain

$$\begin{aligned}
\nabla_{\boldsymbol{U}} \Phi(\boldsymbol{x}, \boldsymbol{w}) &= n^{-1/2} \Big(\boldsymbol{v} \odot \sigma'(d^{-1/2}\boldsymbol{U}\boldsymbol{x} + \boldsymbol{b})\Big) d^{-1/2} \boldsymbol{x}^\top \in \mathbb{R}^{n \times d} \\
\nabla_{\boldsymbol{b}} \Phi(\boldsymbol{x}, \boldsymbol{w}) &= n^{-1/2} \boldsymbol{v} \odot \sigma'\Big(d^{-1/2}\boldsymbol{U}\boldsymbol{x} + \boldsymbol{b})\Big) \in \mathbb{R}^n \\
\nabla_{\boldsymbol{v}} \Phi(\boldsymbol{x}, \boldsymbol{w}) &= n^{-1/2} \sigma(d^{-1/2}\boldsymbol{U}\boldsymbol{x} + \boldsymbol{b}) \in \mathbb{R}^n \\
\nabla_{c} \Phi(\boldsymbol{x}, \boldsymbol{w}) &= 1 \in \mathbb{R}.
\end{aligned} \tag{11.6.3}$$

Contrary to (11.5.2), the three gradients with $O(n)$ entries are all scaled by the factor $n^{-1/2}$. This leads to a different training dynamics.

11.6.2 Neural tangent kernel

We compute again the neural tangent kernel. Unlike for LeCun initialization, there is no $1/n$ scaling required to obtain convergence of

$$\hat{K}_n(\boldsymbol{x}, \boldsymbol{z}) = \langle \nabla_{\boldsymbol{w}} \Phi(\boldsymbol{x}, \boldsymbol{w}_0), \nabla_{\boldsymbol{w}} \Phi(\boldsymbol{z}, \boldsymbol{w}_0) \rangle$$

as $n \to \infty$. Here and in the following we consider the setting (11.6.1)–(11.6.2) for Φ and $\boldsymbol{w}_0$. This is also referred to as the NTK initialization, we denote the kernel by K^{NTK}. Due to the different training dynamics, we obtain additional terms in the NTK compared to Theorem 11.23.

Theorem 11.30. *Let $R < \infty$ such that $|\sigma(x)| \leq R \cdot (1 + |x|)$ and $|\sigma'(x)| \leq R \cdot (1 + |x|)$ for all $x \in \mathbb{R}$, and let $\mathcal{D}$ satisfy Assumption 11.14. For any $\boldsymbol{x}$, $\boldsymbol{z} \in \mathbb{R}^d$ and $u_i \overset{\text{iid}}{\sim} \mathcal{D}(0, 1/d)$, $i = 1, \dots, d$, it then holds*

$$\begin{aligned}
\lim_{n \to \infty} \hat{K}_n(\boldsymbol{x}, \boldsymbol{z}) &= \Big(1 + \frac{\boldsymbol{x}^\top \boldsymbol{z}}{d}\Big) \mathbb{E}[\sigma'(\boldsymbol{u}^\top \boldsymbol{x})^\top \sigma'(\boldsymbol{u}^\top \boldsymbol{z})] + \mathbb{E}[\sigma(\boldsymbol{u}^\top \boldsymbol{x})^\top \sigma(\boldsymbol{u}^\top \boldsymbol{z})] + 1 \\
&=: K^{\text{NTK}}(\boldsymbol{x}, \boldsymbol{z})
\end{aligned}$$

almost surely.

Proof. Denote $\boldsymbol{x}^{(1)} = \boldsymbol{U}_0\boldsymbol{x} + \boldsymbol{b}_0 \in \mathbb{R}^n$ and $\boldsymbol{z}^{(1)} = \boldsymbol{U}_0\boldsymbol{z} + \boldsymbol{b}_0 \in \mathbb{R}^n$. Due to the initialization (11.6.2) and our assumptions on $\mathcal{D}(0,1)$, the components

$$x_i^{(1)} = \sum_{j=1}^{d} U_{0;ij} x_j \sim \boldsymbol{u}^\top \boldsymbol{x} \qquad i = 1, \dots, n$$

are i.i.d. with finite pth moment (independent of n) for all $1 \leq p \leq 8$, and the same holds for the $(\sigma(x_i^{(1)}))_{i=1}^n$, $(\sigma'(x_i^{(1)}))_{i=1}^n$, $(\sigma(z_i^{(1)}))_{i=1}^n$, and $(\sigma'(z_i^{(1)}))_{i=1}^n$.

Then

$$\hat{K}_n(\boldsymbol{x}, \boldsymbol{z}) = \Big(1 + \frac{\boldsymbol{x}^\top \boldsymbol{z}}{d}\Big) \frac{1}{n} \sum_{i=1}^{n} v_i^2 \sigma'(x_i^{(1)}) \sigma'(z_i^{(1)}) + \frac{1}{n} \sum_{i=1}^{n} \sigma(x_i^{(1)}) \sigma(z_i^{(1)}) + 1.$$

By the law of large numbers and because $\mathbb{E}[v_i^2] = 1$, this converges almost surely to $K^{\mathrm{NTK}}(\boldsymbol{x}, \boldsymbol{z})$.

The existence of n_0 follows similarly by an application of Theorem A.22. □

Example 11.31 (K^{NTK} for ReLU)**.** Let $\sigma(x) = \max\{0, x\}$ and let $\mathcal{D}(0, 1/d)$ be the centered normal distribution on $\mathbb{R}$ with variance $1/d$. For $\boldsymbol{x}$, $\boldsymbol{z} \in \mathbb{R}^d$ holds by [36, Appendix A] (also see Exercise 11.36), that with $u_i \overset{\text{iid}}{\sim} \mathcal{D}(0, 1/d)$, $i = 1, \dots, d$,

$$\mathbb{E}[\sigma'(\boldsymbol{u}^\top \boldsymbol{x}) \sigma'(\boldsymbol{u}^\top \boldsymbol{z})] = \frac{\pi - \arccos\left(\frac{\boldsymbol{x}^\top \boldsymbol{z}}{\|\boldsymbol{x}\| \|\boldsymbol{z}\|}\right)}{2\pi}.$$

Together with Example 11.17, this yields an explizit formula for K^{NTK} in Theorem 11.30.

For this network architecture and under suitable assumptions on $\mathcal{D}$, similar arguments as in Section 11.5 can be used to show convergence of gradient descent to a global minimizer and proximity of the full to the linearized model. We refer to the literature in the bibliography section.

Bibliography and further reading

The discussion on linear and kernel regression in Sections 11.1 and 11.2 is quite standard, and can similarly be found in many textbooks. For more details on kernel methods we refer for instance to [41, 205]. The neural tangent kernel and its connection to the training dynamics was first investigated in [105] using an architecture similar to the one in Section 11.6. Since then, many works have extended this idea and presented differing perspectives on the topic, see for instance [2, 55, 5, 35]. Our presentation in Sections 11.4, 11.5, and 11.6 primarily follows [129] who also discussed the case of LeCun initialization. Especially for the main results in Theorem 11.13 and Theorem 11.23, we largely follow the arguments in this paper. The above references additionally treat the case of deep networks, which we have omitted here for simplicity. The explicit formula for the NTK of ReLU networks as presented in Examples 11.17 and 11.31 was given in [36]. The observation that neural networks at initialization behave like Gaussian processes presented in Section 11.5.5 was first made in [155]. For a general reference on Gaussian processes see the textbook [186]. When only training the last layer of a network (in which the network is affine linear), there are strong links to random feature methods [184]. Recent developements on this topic can also be found in the literature under the name "Neural network Gaussian processes", or NNGPs for short [128, 46].

Exercises

Exercise 11.32. Prove Theorem 11.3.

Hint: Assume first that $\boldsymbol{w}_0 \in \ker(\boldsymbol{A})^\perp$ (i.e. $\boldsymbol{w}_0 \in \tilde{H}$). For $\operatorname{rank}(\boldsymbol{A}) < d$, using $\boldsymbol{w}_k = \boldsymbol{w}_{k-1} - h\nabla f(\boldsymbol{w}_{k-1})$ and the singular value decomposition of $\boldsymbol{A}$, write down an explicit formula for $\boldsymbol{w}_k$. Observe that due to $1/(1-x) = \sum_{k\in\mathbb{N}_0} x^k$ for all $x \in (0,1)$ it holds $\boldsymbol{w}_k \to \boldsymbol{A}^\dagger \boldsymbol{y}$ as $k \to \infty$, where $\boldsymbol{A}^\dagger$ is the Moore-Penrose pseudoinverse of $\boldsymbol{A}$.

Exercise 11.33. Let $\boldsymbol{x}_i \in \mathbb{R}^d$, $i = 1, \dots, m$. Show that there exists a "feature map" $\phi : \mathbb{R}^d \to \mathbb{R}^m$, such that for any configuration of labels $y_i \in \{-1, 1\}$, there always exists a hyperplane in $\mathbb{R}^m$ separating the two sets $\{\phi(\boldsymbol{x}_i) \,|\, y_i = 1\}$ and $\{\phi(\boldsymbol{x}_i) \,|\, y_i = -1\}$.

Exercise 11.34. Consider the RBF kernel $K : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$, $K(x, x') := \exp(-(x - x')^2)$. Find a Hilbert space H and a feature map $\phi : \mathbb{R} \to H$ such that $K(x, x') = \langle \phi(\boldsymbol{x}), \phi(\boldsymbol{x}') \rangle_H$.

Exercise 11.35. Let $n \in \mathbb{N}$ and consider the polynomial kernel $K : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$, $K(\boldsymbol{x}, \boldsymbol{x}') = (1 + \boldsymbol{x}^\top \boldsymbol{x}')^r$. Find a Hilbert space H and a feature map $\phi : \mathbb{R}^d \to H$, such that $K(\boldsymbol{x}, \boldsymbol{x}') = \langle \phi(\boldsymbol{x}), \phi(\boldsymbol{x}') \rangle_H$.

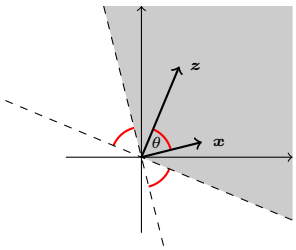
Hint: Use the multinomial formula.

Exercise 11.36. Let $u_i \overset{\text{iid}}{\sim} \mathrm{N}(0, 1)$ be i.i.d. standard Gaussian distributed random variables for $i = 1, \dots, d$. Show that for all nonzero $\boldsymbol{x}$, $\boldsymbol{z} \in \mathbb{R}^d$

$$\mathbb{E}[\mathbb{1}_{[0,\infty)}(\boldsymbol{u}^\top \boldsymbol{x}) \mathbb{1}_{[0,\infty)}(\boldsymbol{u}^\top \boldsymbol{z})] = \frac{\pi - \theta}{2\pi}, \qquad \theta = \arccos\Big(\frac{\boldsymbol{x}\boldsymbol{z}^\top}{\|\boldsymbol{x}\| \, \|\boldsymbol{z}\|}\Big).$$

This shows the formula for the ReLU NTK with Gaussian initialization as discussed in Example 11.31.

Hint: Consider the following sketch



Exercise 11.37. Consider the network (11.5.1) with LeCun initialization as in (11.5.3), but with the biases instead initialized as

$$c,\ b_i \overset{\text{iid}}{\sim} \mathcal{D}(0, 1) \qquad \text{for all } i = 1, \dots, n. \tag{11.6.4}$$

Compute the corresponding NTK as in Theorem 11.23. Moreover, compute the NTK also for the normalized network (11.6.1) with initialization (11.6.2) as in Theorem 11.30, but replace again the bias initialization with that given in (11.6.4).