Which Minimizer Does My Neural Network Converge To?

Manuel Nonnenmacher[1,2][✉], David Reeb[1], and Ingo Steinwart[2]

[1] Bosch Center for Artificial Intelligence (BCAI), 71272 Renningen, Germany, `manuel.nonnenmacher@de.bosch.com`
[2] Institute for Stochastics and Applications, University of Stuttgart, 70569 Stuttgart, Germany

Abstract. The loss surface of an overparameterized neural network (NN) possesses many global minima of zero training error. We explain how common variants of the standard NN training procedure change the minimizer obtained. First, we make explicit how the size of the initialization of a strongly overparameterized NN affects the minimizer and can deteriorate its final test performance. We propose a strategy to limit this effect. Then, we demonstrate that for adaptive optimization such as AdaGrad, the obtained minimizer generally differs from the gradient descent (GD) minimizer. This adaptive minimizer is changed further by stochastic mini-batch training, even though in the non-adaptive case, GD and stochastic GD result in essentially the same minimizer. Lastly, we explain that these effects remain relevant for less overparameterized NNs. While overparameterization has its benefits, our work highlights that it induces sources of error absent from underparameterized models.

Keywords: Overparameterization · Optimization · Neural Networks.

1 Introduction

Overparameterization is a key ingredient in the success of neural networks (NNs), thus modern NNs have become ever more strongly overparameterized. As much as this has helped increase NN performance, overparameterization has also caused several puzzles in our theoretical understanding of NNs, especially with regards to their good optimization behavior [36] and favorable generalization properties [37]. In this work we shed light on the optimization behavior, identifying several caveats.

More precisely, we investigate how the obtained minimizer can change depending on the NN training procedure – we consider common techniques like adjusting the initialization size, the use of adaptive optimization, and stochastic gradient descent (SGD).

These training choices can have significant impact on the final test performance, see Fig. 1. While some of these peculiar effects had been observed experimentally [34,38], we explain and quantify them in a general setting. Note further that this effect is absent from the more commonly studied underparameterized models, whose minimizer is generically unique ([31] and App. A(see [26])).

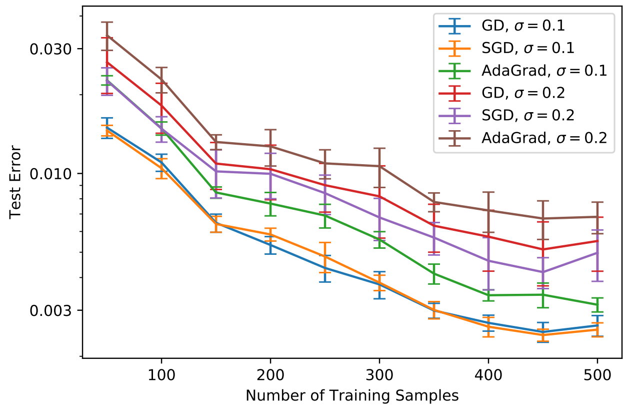

Fig. 1. The test performance of an overparameterized NN depends considerably on the optimization method (GD, SGD, AdaGrad) and on the initialization size σ, even though all nets have been trained to the same low empirical error of 10^{-5}. Shown are results on MNIST 0 vs. 1 under squared loss in 5 repetitions of each setting, varying the degree of overparameterization by changing the training set size. Our theoretical results explain and quantify these differences between training choices.

Our analysis makes use of the improved understanding of the training behavior of strongly overparameterized NNs which has been achieved via the Neural Tangent Kernel (NTK) [20,22,15]. Through this connection one can show that overparameterized NNs trained with gradient descent (GD) converge to a minimizer which is an interpolator of low complexity w.r.t. the NTK [7]. We also extend our analysis to less overparameterized NNs by using linearizations at later training times instead of the NTK limit (Sect. 6).

Our contributions are as follows:

– We explain quantitatively how the size of initialization impacts the trained overparameterized NN and its test performance. While the influence can generally be severe, we suggest a simple algorithm to detect and mitigate this effect (Sect. 3).
– We prove that the choice of adaptive optimization method changes the minimizer obtained and not only the training trajectory (Sect. 4). This can significantly affect the test performance. As a technical ingredient of independent interest we prove that strongly overparameterized NNs admit a linearization under adaptive training similar to GD and SGD training [22,14,1].

- We show that the batch size of mini-batch SGD affects the minimizer of adaptively trained NNs, in contast to the non-adaptive setting, where the SGD minimizer is virtually the same as in full-batch (GD) training (Sect. 5).
- Our theoretical findings are confirmed by extensive experiments on different datasets, where we investigate the effect of the changed minimizer on the test performance (Sect. 7).

2 Background

Throughout, the N training points $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N \subset B_1^d(0) \times \mathbb{R}$ have inputs from the d-dimensional unit ball $B_1^d(0) := \{x \in \mathbb{R}^d : \|x\|_2 \leq 1\}$, and one-dimensional outputs for simplicity. $\mathcal{D}$ is called *non-degenerate* if $x_i \nparallel x_j$ for $i \neq j$. We often view the training inputs $X = (x_1, \ldots, x_N)^T \in \mathbb{R}^{N \times d}$ and corresponding labels $Y = (y_1, \ldots, y_N)^T \in \mathbb{R}^N$ in matrix form. For any function g on $\mathbb{R}^d$, we define $g(X)$ by row-wise application of g.

The output of a fully-connected NN with L layers and parameters θ is denoted $f_\theta^{\mathrm{NN}}(x) = h^L(x)$ with

$$\begin{aligned} h^l(x) &= \frac{\sigma}{\sqrt{m_l}} W^l a(h^{l-1}) + \sigma b^l \quad \text{for } l = 2, \ldots, L, \\ h^1(x) &= \frac{\sigma}{\sqrt{m_1}} W^1 x + \sigma b^1, \end{aligned} \tag{1}$$

where $a : \mathbb{R} \to \mathbb{R}$ is the activation function, applied component-wise. We assume a either to have bounded second derivative [14] or to be the ReLU function $a(h) = \max\{h, 0\}$ [1]. Layer l has m_l neurons ($m_0 = d$, $m_L = 1$); we assume $m_l \equiv m$ constant for all hidden layers $l = 1, \ldots, L-1$ for simplicity.[3] $W^l \in \mathbb{R}^{m_l \times m_{l-1}}$ and $b^l \in \mathbb{R}^{m_l}$ are the NN weights and biases, for a total of $P = \sum_{l=1}^L (m_{l-1}+1) m_l$ real parameters in $\theta = [W^{1:L}, b^{1:L}]$. We keep the parameter scaling σ as an explicit scalar parameter, that can be varied [20]. The parametrization (1) together with a *standard normal initialization* $W_{i,j}^l, b_i^l \sim \mathcal{N}(0,1)$ (sometimes with *zero biases* $b_i^l = 0$) is the *NTK-parametrization* [14]. This is equivalent to the standard parametrization (i.e., no prefactors in (1)) and initialization $W_{i,j}^l \sim \mathcal{N}(0, \sigma^2/m_l)$, $b_i^l \sim \mathcal{N}(0, \sigma^2)$ [18] in a NN forward pass, while in gradient training the two parametrizations differ by a width-dependent scaling factor of the learning rate [20,22].

We mostly consider the squared error $|\hat{y} - y|^2/2$ and train by minimizing its empirical loss

$$L_{\mathcal{D}}(f_\theta) = \frac{1}{2N} \sum_{(x,y) \in \mathcal{D}} |f_\theta(x) - y|^2 = \frac{1}{2N} \|f_\theta(X) - Y\|_2^2.$$

We train by *discrete* update steps, i.e. starting from initialization $\theta = \theta_0$ the parameters are updated via the discrete iteration $\theta_{t+1} = \theta_t - \eta U_t[\theta_t]$ for $t =$

[3] For unequal hidden layers, the infinite-width limit (below) is $\min_{1 \leq l \leq L-1}\{m_l\} \to \infty$ [22].

$0, 1, \ldots$, where U_t is some function of the (past and present) parameters and η the learning rate. Gradient descent (GD) training uses the present loss gradient $U_t[\theta_t] = \nabla_\theta L_\mathcal{D}(f_\theta)|_{\theta=\theta_t}$; for adaptive and stochastic gradient methods, see Sects. 4 and 5.

A central object in our study is the feature map $\phi(x) := \nabla_\theta f_\theta^{\mathrm{NN}}(x)\big|_{\theta=\theta_0} \in \mathbb{R}^{1\times M}$ associated with a NN f_θ^{NN} at its initialization θ_0. With this we will consider the NN's linearization around θ_0 as

$$f_\theta^{\mathrm{lin}}(x) \; := \; f_{\theta_0}^{\mathrm{NN}}(x) + \phi(x)(\theta - \theta_0). \tag{2}$$

On the other hand, ϕ gives rise to the so-called neural tangent kernel (NTK) $K : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ by $K(x, x') := \phi(x)\phi(x')^T$ [20,22]. We use the associated kernel norm to define the *minimum complexity interpolator* of the data $\mathcal{D}$:

$$f^{\mathrm{int}} := \operatorname*{arg\,min}_{f\in\mathcal{H}_K} \|f\|_{\mathcal{H}_K} \qquad \text{subject to} \;\; Y = f(X), \tag{3}$$

where $\mathcal{H}_K$ is the reproducing kernel Hilbert space (RKHS) associated with K [7]. Its explicit solution is $f^{\mathrm{int}}(x) = \phi(x)\phi(X)^T \left(\phi(X)\phi(X)^T\right)^{-1} Y$ (App. B (see [26])). Here, $\phi(X)\phi(X)^T = K(X, X)$ is invertible for "generic" X and θ_0 in the overparameterized regime $P \geq N$. Technically, we always assume that the infinite-width kernel $\Theta(x, x') := \lim_{m\to\infty} K(x, x')$ has positive minimal eigenvalue $\lambda_0 := \lambda_{\min}(\Theta(X, X)) > 0$ on the data (this $\lim_{m\to\infty}$ exists in probability [20]). $\lambda_0 > 0$ holds for non-degenerate $\mathcal{D}$ and standard normal initialization with zero biases [14]; for standard normal initialization (with normal biases) it suffices that $x_i \neq x_j$ for $i \neq j$.

With these prerequisites, we use as a technical tool the fact that a strongly overparameterized NN stays close to its linearization during GD training (for extensions, see Thm. 7 More precisely, the following holds:

Lemma 1. ([22,1]) *Denote by θ_t and $\tilde\theta_t$ the parameter sequences obtained by gradient descent on the NN f_θ^{NN} (1) and on its linearization $f_{\tilde\theta}^{\mathrm{lin}}$ (2), respectively, starting from the same initialization $\theta_0 = \tilde\theta_0$ and with sufficiently small step size η. There exists some $C = \mathrm{poly}(1/\delta, N, 1/\lambda_0, 1/\sigma)$ such that for all $m \geq C$ and for all $x \in B_1^d(0)$ it holds with probability at least $1 - \delta$ over the random initialization θ_0 that:* $\sup_t |f_{\theta_t}^{\mathrm{NN}}(x) - f_{\tilde\theta_t}^{\mathrm{lin}}(x)|^2 \leq O(1/m)$.

3 Impact of initialization

In this section we quantify theoretically how the initialization θ_0 influences the final NN trained by gradient descent (GD), and in particular its test error or risk. As a preliminary result, we give an analytical expression for the GD-trained NN, which becomes exact in the infinite-width limit:

Theorem 1. *Let f^{NN} be the fully converged solution of an L-layer ReLU-NN (1), trained by gradient descent under squared loss on non-degenerate data $\mathcal{D} = (X, Y)$. There exists $C = \mathrm{poly}(1/\delta, N, 1/\lambda_0, 1/\sigma)$ such that whenever there are $m \geq C$*

neurons in each hidden layer, it holds for any $x \in B_1^d(0)$ that, with probability at least $1-\delta$ over standard normal initialization θ_0,

$$f^{\text{NN}}(x) \;= \phi(x)\phi(X)^T\big(\phi(X)\phi(X)^T\big)^{-1}Y + \frac{1}{L}\phi(x)\big(\mathbb{1} - P_{\mathcal{F}}\big)\theta_0 \,+\, O\Big(\frac{1}{\sqrt{m}}\Big), \tag{4}$$

where $P_{\mathcal{F}} := \phi(X)^T\big(\phi(X)\phi(X)^T\big)^{-1}\phi(X)$ is the projector onto the data feature subspace.

Results similar to Thm. 1 have been found in other works before [20,21,38]. Our result has a somewhat different form compared to them and makes the dependence on the initialization θ_0 more explicit. For this, we simplified the Y-independent term by using the property $f_\theta^{\text{NN}}(x) = \frac{1}{L}\langle\theta, \nabla_\theta f_\theta^{\text{NN}}(x)\rangle$ of ReLU-NNs (Lemma 4 (see [26])). Further, our Thm. 1 is proven for discrete, rather than continuous, update steps. The proof in App. F.1 (see [26]) first solves the dynamics of the linearized model (2) iteratively and recovers the first two terms in the infinite training time limit. Finally, Lemma 1 gives $O(1/\sqrt{m})$-closeness to $f^{\text{NN}}(x)$ in the strongly overparameterized regime.

The expression (4) for the converged NN has two main parts. The first term is just the minimum complexity interpolator $f^{\text{int}}(x) = \phi(x)\phi(X)^T(\phi(X)\phi(X)^T)^{-1}Y$ from Eq. (3), making the solution interpolate the training set $\mathcal{D}$ perfectly. Note, $f^{\text{int}}(x)$ is virtually independent of the random initialization θ_0 in the strongly overparameterized regime since $\phi(x)\phi(x')^T$ converges in probability as $m \to \infty$ (Sect. 2); intuitively, random θ_0's yield similarly expressive features $\phi(x) \in \mathbb{R}^P$ as $P \to \infty$.

The second term in (4), however, depends on θ_0 explicitly. More precisely, it is proportional to the part $(\mathbb{1}-P_{\mathcal{F}})\theta_0$ of the initialization that is orthogonal to the feature subspace $\mathcal{F} = \text{span}\{\phi(x_1), \ldots, \phi(x_N)\}$ onto which $P_{\mathcal{F}}$ projects. This term is present as GD alters $\theta_t \in \mathbb{R}^P$ only along the N-dimensional $\mathcal{F}$. It vanishes on the training inputs $x = X$ due to $\phi(X)(\mathbb{1} - P_{\mathcal{F}}) = 0$.

Our main concern is now the extent to which the *test* error is affected by θ_0 and thus in particular by the second term $\phi(x)(\mathbb{1}-P_{\mathcal{F}})\theta_0/L$. Due to its Y-independence, this term will generally harm test performance. While this holds for large initialization scaling σ, small σ suppresses this effect:

Theorem 2. *Under the prerequisites of Thm. 1 and fixing a test set $\mathcal{T} = (X_{\mathcal{T}}, Y_{\mathcal{T}})$ of size $N_{\mathcal{T}}$, there exists $C = \text{poly}(N, 1/\delta, 1/\lambda_0, 1/\sigma, N_{\mathcal{T}})$ such that for $m \geq C$ the test error $L_{\mathcal{T}}(f^{\text{NN}})$ of the trained NN satisfies the following bounds, with probability at least $1-\delta$ over the standard normal initialization with zero biases,*

$$\begin{aligned}\sqrt{L_{\mathcal{T}}(f^{\text{NN}})} &\geq \sigma^L J(X_{\mathcal{T}}) - \sqrt{L_{\mathcal{T}}(f^{\text{int}})} - O\Big(\frac{1}{\sqrt{m}}\Big)\,,\\ \sqrt{L_{\mathcal{T}}(f^{\text{NN}})} &\leq \sqrt{L_{\mathcal{T}}(f^{\text{int}})} + O\Big(\sigma^L + \frac{1}{\sqrt{m}}\Big),\end{aligned} \tag{5}$$

where $J(X_{\mathcal{T}})$ is independent of the initialization scaling σ and $J(X_{\mathcal{T}}) > 0$ holds almost surely. With standard normally initialized biases, the same bounds hold with both σ^L replaced by σ.

The lower bound in (5) shows that big initialization scalings σ leave a significant mark $\sim \sigma^L$ on the the test error, while the final *training* error of f^{NN} is always 0 due to strong overparameterization. This underlines the importance of good initialization schemes, as they do not merely provide a favorable starting point for training, but impact which minimizer the weights converge to. To understand the scaling, note that the features scale with σ like $\phi_\sigma(x) = \sigma^L \phi_1(x)$ (the behavior is more complex for standard normal biases, see App. F.2 (see [26]). The first term in (4) is thus invariant under σ, while the second scales as $\sim \sigma^L$.

The main virtue of the upper bound in (5) is that the harmful influence of initialization can be reduced by adjusting σ and m simultaneously. To show this, App. F.3 (see [26]) takes care to bound the second term in (4) on the test set $\|\phi(X_{\mathcal{T}})(1 - P_{\mathcal{F}})\theta_0\|/\sqrt{N_{\mathcal{T}}} \leq O(\sigma^L)$ *independently* of ϕ's dimension P, which would grow with m. Note further that the kernel interpolator f^{int} in (5) with loss $L_{\mathcal{T}}(f^{\mathrm{int}})$ was recently found to be quite good empirically [5] and theoretically [24,4].

Based on these insights into the decomposition (4) and the scaling of $L_{\mathcal{T}}(f^{\mathrm{NN}})$ with σ, we suggest the following algorithm to mitigate the potentially severe influence of the initialization on the test error in large NNs as much as possible: *(a)* randomly sample an initialization θ_0 and train $f^{\mathrm{NN}'}$ using a standard scaling $\sigma' \simeq O(1)$ (e.g. [18]); *(b)* train $f^{\mathrm{NN}''}$ with the same θ_0 and a somewhat smaller $\sigma'' < \sigma'$ (e.g. by several ten percent); *(c)* compare the losses on a validation set $\mathcal{V}$: if $L_{\mathcal{V}}(f^{\mathrm{NN}''}) \approx L_{\mathcal{V}}(f^{\mathrm{NN}'})$ then finish with step (e), else if $L_{\mathcal{V}}$ decreases by a significant margin then continue; *(d)* repeat from step (b) with successively smaller $\sigma''' < \sigma''$ until training becomes impractically slow; *(e)* finally, return the trained f^{NN} with smallest validation loss.

It is generally not advisable to start training (or the above procedure) with a too small $\sigma < O(1)$ due to the vanishing gradient problem which leads to slow training, even though the upper bound in (5) may suggest very small σ values to be beneficial from the viewpoint of the test error. A “antisymmetrical initialization” method was introduced in [38] to reduce the impact of the initialization-dependence on the test performance by doubling the network size; this however also increases the computational cost.

Note further that the above theorems do not hold exactly anymore for σ too small due to the $m \geq \mathrm{poly}(1/\sigma)$ requirement; our experiments (Fig. 1 and Sect. 7) however confirm the predicted σ-scaling even for less strongly overparameterized NNs.

4 Impact of adaptive optimization

We now explain how the choice of adaptive optimization method affects the minimizer to which overparameterized models converge. The discrete weight update step for adaptive gradient training methods is

$$\theta_{t+1} = \theta_t - \eta D_t \nabla_\theta L_{\mathcal{D}}(\theta)\big|_{\theta=\theta_t}, \tag{6}$$

where the "adaptive matrices" $D_t \in \mathbb{R}^{P\times P}$ are prescribed by the method. This generalizes GD, which is obtained by $D_t \equiv \mathbb{1}$, and includes AdaGrad [16] via $D_t = \Big(\mathrm{diag}\big(\sum_{u=0}^t g_u g_u^T\big)\Big)^{-1/2}$ with the loss gradients $g_t = \nabla_\theta L(\theta)\big|_{\theta_t} \in \mathbb{R}^P$, as well as RMSprop [19] and other adaptive methods. We say that a sequence of adaptive matrices concentrates around $D \in \mathbb{R}^{P\times P}$ if there exists $Z \in \mathbb{R}$ such that $\|D_t - D\|_{\mathrm{op}}/D_{\max} \le Z/\sqrt{m}$ holds for all $t \in \mathbb{N}$, where $D_{\max} := \sup_t \|D_t\|_{\mathrm{op}}$; this is the simplest assumption under which we can generalize Thm. 1, but in general we only need that the linearization during training holds approximately (App. F.4 (see [26])).

The following result gives a closed-form approximation to strongly overparameterized NNs during training by Eq. (6). Note that this overparameterized adaptive case was left unsolved in [31]. The closed-form expression allows us to illustrate via explicit examples that the obtained minimizer can be markedly different from the GD minimizer, see Example 1 below.

Theorem 3. *Given a NN f_θ^{NN} (1) and a non-degenerate training set $\mathcal{D} = (X, Y)$ for adaptive gradient training (6) under squared loss with adaptive matrices $D_t \in \mathbb{R}^{P\times P}$ concentrated around some D, there exists $C = \mathrm{poly}(N, 1/\delta, 1/\lambda_0, 1/\sigma)$ such that for any width $m \ge C$ of the NN and any $x \in B_1^d(0)$ it holds with probability at least $1-\delta$ over the random initialization that*

$$\begin{aligned} f_{\theta_t}^{\mathrm{NN}}(x) \;&= \phi(x)A_t\left[\mathbb{1}\;-\prod_{u=t-1}^{0}\left(\mathbb{1}-\frac{\eta}{N}\phi(X)D_u\phi(X)^T\right)\right]\left(Y - f_{\theta_0}^{\mathrm{NN}}(X)\right), \\ &\quad + f_{\theta_0}^{\mathrm{NN}}(x) + \phi(x)B_t + O\Big(\frac{1}{\sqrt{m}}\Big), \end{aligned} \tag{7}$$

where $A_t = D_{t-1}\phi(X)^T\left(\phi(X)D_{t-1}\phi(X)^T\right)^{-1}$ and

$$B_t = \sum_{v=2}^{t}(A_{v-1}-A_v)\cdot\left[\mathbb{1}-\prod_{w=v-2}^{0}\left(\mathbb{1}-\frac{\eta}{N}\phi(X)D_w\phi(X)^T\right)\right]\cdot\left(Y - f_{\theta_0}^{\mathrm{NN}}(X)\right).$$

To interpret this result, notice that on the training inputs X we have $\phi(X)A_t = \mathbb{1}$ and therefore $\phi(X)B_t = 0$, so that the dynamics $f_{\theta_t}^{\mathrm{NN}}(X)$ on the training data simplifies significantly: When the method converges, i.e. $\prod_{u=t-1}^{0}(\dots) \to 0$ as $t\to\infty$, we have $f_{\theta_t}^{\mathrm{NN}}(X) \to Y + O(1/\sqrt{m})$, meaning that the training labels are (almost) perfectly interpolated at convergence. Even at convergence, however, the interpolating part $f_{\theta_0}^{\mathrm{NN}}(x) + \phi(x)A_t(Y - f_{\theta_0}^{\mathrm{NN}}(X))$ depends on D_t (via A_t) for test points x. In addition to that, the term $\phi(x)B_t$ in (7) is "path-dependent" due to the sum $\sum_{v=2}^{t}(\dots)$ and takes account of changes in the A_t (and thus, D_t) matrices during optimization, signifying changes in the geometry of the loss surface. The proof of Thm. 3 in App. F.4 (see [26]) is based on a generalization of the linearization Lemma 1 for strongly overparameterized NNs to adaptive training methods (Thm. 7 in App. G (see [26])).

We make the dependence of the trained NN (7) on the choice of adaptive method explicit by the following example of AdaGrad training.

Example 1. Perform adaptive optimization on a ReLU-NN with adaptive matrices D_t according to the AdaGrad prescription with

$$g_t = a_t g \quad \text{where} \quad g \in \mathbb{R}^P,\ a_t \in \mathbb{R}, \tag{8}$$

i.e. we assume that the gradients point all in the same direction g with some decay behavior set by the a_t. We choose this simple setting for illustration purposes, but it occurs e.g. for heavily overparameterized NNs with $N = 1$ training point, where $g = \phi(x_1)^T$ (a similar setting was used in [36]). The adaptive matrices are then $D_t = D\left(\sum_{i=0}^t a_i^2\right)^{-1/2}$ with $D = \left(\mathrm{diag}(gg^T)\right)^{-1/2}$. As D_t evolves only with scalar factors, $A_t = D\phi(X)^T\left(\phi(X)D\phi(X)^T\right)^{-1}$ is constant and thus $B_t = 0$. We can then explicitly evaluate Thm. 3 and use similar arguments as in Thm. 2 to collect the $f_{\theta_0}^{\mathrm{NN}}$-terms into $O(\sigma^L)$, to write down the minimizer at convergence:

$$f^{\mathrm{NN}}(x) \ = \ \phi(x)D\phi(X)^T\left(\phi(X)D\phi(X)^T\right)^{-1}Y + O\left(\sigma^L\right) + O\left(1/\sqrt{m}\right). \tag{9}$$

This example shows explicitly that the minimizer obtained by adaptive gradient methods in overparameterized NNs can be different from the GD minimizer, which results by setting $D = \mathbb{1}$. This difference is not seen on the training inputs X, where Eq. (9) always evaluates to Y, but only at test points x. In fact, any function of the form $f^{\mathrm{NN}}(x) = \phi(x)w + O(\sigma^L + 1/\sqrt{m})$ that interpolates the data, $\phi(X)w = Y$, can be obtained as the minimizer by a judicious choice of D. In contrast to the toy example in Wilson et al. [34], our Example 1 does not require a finely chosen training set and is just a special case of the more general Thm. 3.

Another way to interpret Example 1 is that this adaptive method converges to the interpolating solution of minimal complexity w.r.t. a kernel $K_D(x, x') = \phi(x)D\phi(x')^T$ different from the kernel $K(x, x') = \phi(x)\phi(x')^T$ associated with GD (see Sect. 2 and Thm. 1). Thus, unlike in Sect. 3 where the disturbing term can in principle be diminished by initializing with small variance, adaptive training directly changes the way in which we interpolate the data and is harder to control.

Note that the situation is different for underparameterized models, where in fact the same (unique) minimizer is obtained irrespectively of the adaptive method chosen (see App. A [26] and [31]).

5 Impact of stochastic optimization

Here we investigate the effect SGD has on the minimizer to which the NN converges. The general update step of adaptive mini-batch SGD with adaptive matrices $D_t \in \mathbb{R}^{P\times P}$ is given by

$$\theta_{t+1} = \theta_t - \eta D_t \nabla_\theta L_{\mathcal{D}_{B_t}}(\theta)\big|_{\theta=\theta_t}, \tag{10}$$

where $\mathcal{D}_{B_t}$ contains the data points of the t-th batch B_t. Further, we denote by X_{B_t} the corresponding data matrix obtained by zeroing all rows outside B_t. Ordinary SGD corresponds to $D_t = \mathbb{1}$.

The main idea behind the results of this section is to utilize the fact that ordinary SGD can be written in the form of an adaptive update step by using the adaptive matrix $D_{B_t} := \frac{N}{|B_t|}\left(\phi(X_{B_t})\right)^T\left(\phi(X)\phi(X)^T\right)^{-1}\phi(X)$ in Eq. (6). Combining this re-writing of mini-batch SGD with the adaptive update rule, we can write Eq. (10) as $\theta_{t+1} = \theta_t - \eta D_t D_{B_t}\nabla_\theta L_{\mathcal{D}}(\theta)|_{\theta=\theta_t}$. Now applying a similar approach as for Thm. 3 leads to the following result:

Theorem 4 (informal). *Let f^{NN}_{adGD} and f^{NN}_{adSGD} be fully trained strongly overparameterized NNs trained on the empirical squared loss with adaptive GD and adaptive SGD, respectively. Then, for sufficiently small learning rate η it holds with high probability that:*

(a) If $D_t = \mathrm{const}$, then $\left|f^{\mathrm{NN}}_{adGD}(x) - f^{\mathrm{NN}}_{adSGD}(x)\right| \leq O(1/\sqrt{m})$.
(b) If D_t changes during training, the minimizers f^{NN}_{adGD} and f^{NN}_{adSGD} differ by a path- and batch-size-dependent contribution on top of the $O(1/\sqrt{m})$ linearization error.

Part (a) shows that GD and SGD lead to basically the same minimizer if the adaptive matrices D_t do not change during training. This is the case in particular for vanilla (S)GD, where $D_t = \mathbb{1}$. Part (b) on the other hand shows that for adaptive methods with varying adaptive matrices, the two NN minimizers obtained by GD and mini-batch SGD differ by a path-dependent contribution, where the path itself can be influenced by the batch size. We expect this effect to be smaller for more overparameterized models since then the adaptive matrices are expected to be more concentrated. For the formal version of Thm. 4 see App. C (see [26]).

One of the prerequisites of Thm. 4 is a small learning rate, but it is straightforward to generalize the results to any strongly overparameterized NN (in the NTK-regime) with a learning rate schedule such that the model converges to a minimizer of zero training loss (see App. F.5 (see [26])).

6 Beyond strong overparameterization

The previous three sections explain the impact of common training techniques on the obtained minimizer for strongly overparameterized NNs. In practice, NNs are usually less overparameterized or trained with a large initial learning rate, both of which can cause $\phi(X)$, and thus also $K(X, X)$, to change appreciably during training. This happens especially during the initial stages of training, where weight changes are most significant. The question thus arises how the theoretical results of Sects. 3–5 transfer to less overparameterized NNs. (Note that experimentally, the effects do still appear in less overparameterized NNs, see Fig. 1.)

The basis of our theoretical approach is the validity of Lemma 1 and its cousins like Thm. 7, which build on the fact that the weights of a strongly overparameterized NN do not change significantly during training, i.e. $\|\theta_t - \theta_0\|_2$ remains small for all $t > 0$. For less overparameterized NNs this does not hold in general. One can circumvent this by selecting a later training iteration $T > 0$ such that $\|\theta_t - \theta_T\|_2$ is small enough for all $t > T$. One can always find such T, assuming that θ_t converges as $t \to \infty$. Next, to proceed with a similar analysis as before, we linearize the NN around iteration T instead of Eq. (2):

$$f_\theta^{\mathrm{lin},T}(x) \ := \ f_{\theta_T}^{\mathrm{NN}}(x) + \phi_T(x)(\theta - \theta_T), \tag{11}$$

where $\phi_T(x) := \nabla_\theta f_\theta^{\mathrm{NN}}(x)\big|_{\theta=\theta_T}$ are the NN features at training time T, assumed such that $\|\theta_t - \theta_T\|_2$ is sufficiently small for all $t > T$. Assuming further that $\lambda_0^T := \lambda_{\min}(\phi_T(X)\phi_T(X)^T) > 0$, gives straightforward adaptations of Thms. 1, 3, and 4 with features $\phi_T(x)$ and valid at times $t \geq T$. The main difference is that the results are no longer probabilistic but rather conditional statements.

To demonstrate how this observation can be applied, assume now that we are given two versions θ_T and θ'_T of a NN, trained with two different training procedures up to iteration T. In case that θ_T and θ'_T are (significantly) different, then the adapted version of either Thm. 1 or 3 shows that both models generally converge to different minimizers; this effect would persist even if the two NNs were trained by the same procedure for $t > T$. On the contrary, if θ_T and θ'_T were the same (or similar) at iteration T, then Thm. 4 suggests that the minimizers will nevertheless differ when θ_T is updated with adaptive (S)GD and θ'_T with vanilla (S)GD. While our results only consider the impact on the minimizer obtained after an initial training period, there may exist further effects not considered in our analysis during the initial training steps $t < T$, where the features change.

7 Experiments

Here we demonstrate our theoretical findings experimentally. We perform experiments on three different data sets, MNIST (here), Fashion-MNIST (App. E (see [26])), and CIFAR10 (Fig. 5), using their 0 and 1 labels. The first experiment (Fig. 2) investigates the effect which the initialization size can have on the test performance of NNs, confirming the results of Sect. 3 qualitatively and quantitatively. Additionally, the behavior of the test (validation) error with σ demonstrates the effectiveness of the error mitigation algorithm described in Sect. 3. The second experiment (Fig. 3) demonstrates the significant difference in test performance between NNs trained with vanilla GD and the adaptive optimization methods AdaGrad and Adam (Sect. 4). The third experiment (Fig. 4) illustrates that for non-adaptive SGD there is only weak dependency on the batch-size and ordering of the datapoints, whereas for adaptive optimization with mini-batch SGD the dependence is noticeable (Sect. 5).

These first three experiments are run with one and two hidden-layer NNs of different widths m. In line with our framework and with other works on

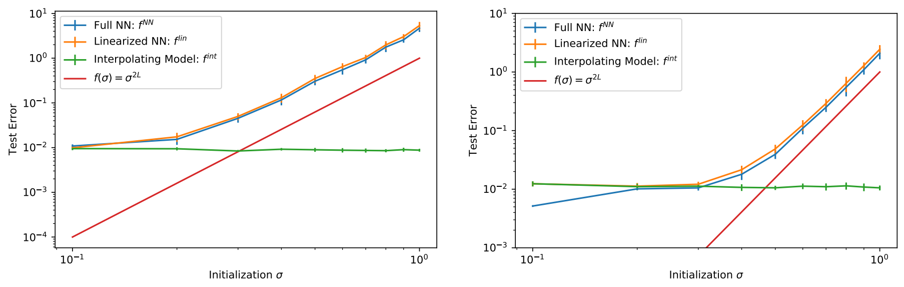

Fig. 2. Impact of initialization. The impact of initialization on the test performance of trained overparameterized NNs is illustrated for $L = 2$ (width $m = 4000$, left) and $L = 3$ layers ($m = 2000$, right). At small initialization size σ, the trained f^{NN} is close to the interpolating model f^{int} (Sect. 3), which is virtually independent of the initialization θ_0 and σ. At larger σ, the test error grows strongly $\sim \sigma^{2L}$, with depth-dependent exponent (Thm. 2). All NNs were trained to the same low empirical error and are close to their linearizations f^{lin}, verifying that indeed we are in the overparameterized limit. Our results underline that initialization does not merely provide a favorable starting point for optimization, but can strongly impact NN test performance.

overparameterized NNs we minimize the weights of the NN w.r.t. the empirical squared loss on a reduced number of training samples ($N = 100$), to make sure that overparameterization $m \gg N$ is satisfied to a high degree. We train all the NNs to a very low training error ($< 10^{-5}$) and then compare the mean test error from 10 independently initialized NNs with error bars representing the standard deviation[4].

In addition to the effects of the three settings described in the previous paragraph, Fig. 1 illustrates that similar effects appear in less overparameterized settings as well. Furthermore, while some of the theoretical conditions are not satisfied for modern architectures such as ResNets or networks trained with the cross-entropy loss, Fig. 5 shows that comparable effects appear in these settings as well.

8 Related work

The fact that NN initialization can influence the average test performance was pointed out in [38] and [36], who investigated this experimentally, but did not quantify it. The method suggested by [38] to reduce this effect doubles the number of NN parameters, significantly increasing the computational cost. Another method to reduce this effect was suggested by [11] with the “doubling trick”, which was shown to potentially harm the training dynamics [38]. [35] also

[4] More details on the settings needed to reproduce the experiments can be found in App. D (see [26]).

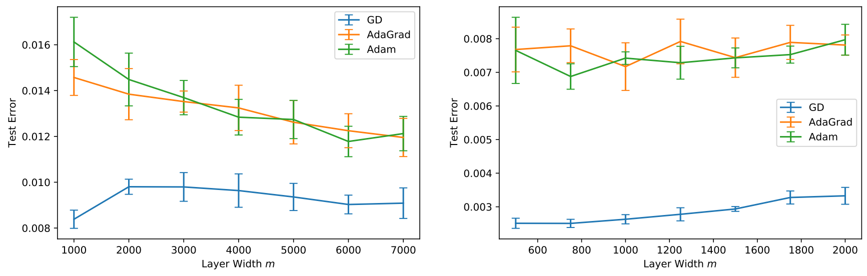

Fig. 3. Impact of adaptive optimization. It is shown that the test performance depends strongly on the choice of adaptive optimization method, for a variety of overparameterized NNs with $L = 2$ (left) and $L = 3$ layers (right) over different widths m. Even though all models were trained to the same low empirical error, there is a significant performance gap between vanilla GD and the adaptive methods, underlining the results of Sect. 4. Thus, while adaptive methods for overparameterized NNs may improve the training dynamics, they also affect the minimizers obtained.

investigates the impact of different initialization scalings but require deep and wide NNs, while we only requires wide NNs. Furthermore, [34] and [36] observed a significant test performance gap between GD and adaptive optimization methods for overparameterized NNs. While [34] does provide an analytical toy example for this in a linear model, our analysis of NNs is general and holds for generic training data. [31] attempts to explain such gaps for overparameterized linear models, but lacks the explicit expressions Eqs. (7), (9) we find. [28] looks into the implicit bias of AdaGrad only for the cross-entropy loss and [3] investigates the generalization behaviour of natural gradient descent under noise. While we on the other hand investigate the impact of general adaptive training methods for the squared loss on the minimizer obtained.

The convergence of the training error under SGD in comparison to GD has been the subject of many works, recently also in the context of strongly overparameterized NNs [25,30,1,10]. We, on the other hand, investigate whether the minimizer found by SGD is similar to the GD minimizer on a test set. Another concept closely linked to this is the implicit bias of (S)GD on the trained NN, certain aspects of which have been investigated in [32,17,29,27,9]. Our work elucidates the implicit bias caused by the random NN initialization and by the optimizer.

We use interpolating kernel methods [7] and the idea of an overparameterized learning regime which does not suffer from overfitting [25,8,6]. Bounds on their test performance have been derived by [24] and [4]. [5] on the other hand demonstrates their good performance experimentally, but does not investigate how the scale of NN initialization or nonzero NN biases influence the test behavior (our Sect. 3). While [33] also explores different minimum norm solutions they only consider

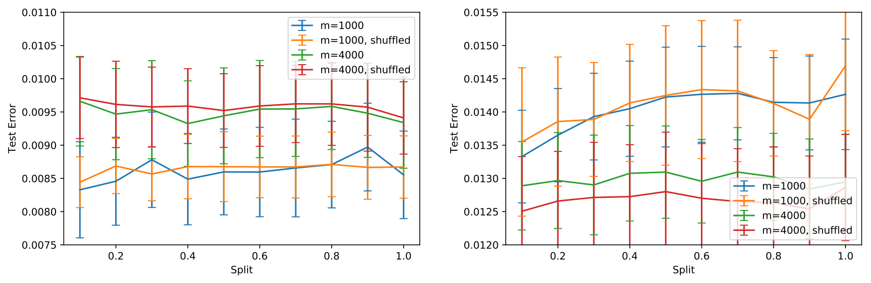

Fig. 4. Impact of stochastic training. Without adaptive optimization (left), SGD-trained NNs have basically the same test error as GD-trained ones (split = 1.0), for a large range of mini-batch sizes and both with and without shuffling the training set (Thm. 4(a)). In contrast to this, when using adaptive optimization methods (AdaGrad, right) the test-performance becomes dependent on the batch-size (Thm. 4(b)). The mini-batch size is given as the split ratio of the training set size, with vanilla GD corresponding to 1.0. The NNs shown have $L = 2$ layers and widths m as indicated.

linear models and different loss functions and do not focus on the impact of different training techniques as we do.

Our analysis uses the Neural Tangent Kernel (NTK) limit [20], where the NN width scales polynomially with the training set size. In this regime, it is found that NNs converge to arbitrarily small training error under (S)GD. The first works to investigate this were [12], [23] and [15]. Later, [1], [2], and [14] extended these results to deep NNs, CNNs and RNNs. The recent contribution by [22], building upon [21] and [13], explicitly solves the training dynamics.

9 Discussion

We have explained theoretically how common choices in the training procedure of overparameterized NNs leave their footprint in the obtained minimizer and in the resulting test performance. These theoretical results provide an explanation for previous experimental observations [34,36], and are further confirmed in our dedicated experiments.

To identify and reduce the harmful influence of the initialization on the NN test performance, we suggest a new algorithm motivated by the bounds in Thm. 2. The potentially harmful influence of adaptive optimization on the test performance, however, cannot be reduced so easily if one wants to keep the beneficial effects on training time. Indeed, current adaptive methods experimentally seem to have worse test error than SGD ([34] and Fig. 3). Therefore, adaptive methods for overparameterized models should be designed not only to improve convergence to any minimum, but this minimum should also be analyzed form the perspective of statistical learning theory.

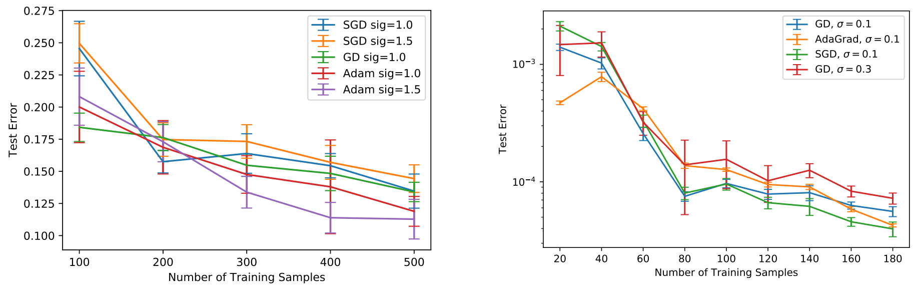

Fig. 5. ResNet networks and cross-entropy loss. The left figure shows the test performance of ResNet-20 on the Cifar10 data set using "airplaine" vs. "automobile" labels (encoded as 0 and 1) trained with squared loss, for different optimization methods (GD, SGD, AdaGrad) and initialization sizes σ, over the training set size. The figure on the right shows the same setting as in Fig. 1 but instead of the squared loss the network is trained using the cross-entropy loss. The shown results are averaged over 5 repetitions for each setting and all models are trained to the same low empirical error of 10^{-5}.

While our theory applies to NNs trained with squared loss, we believe the same effects to appear in NNs trained with cross-entropy loss (see Fig. 5). And while we show that in the less overparameterized regime, the minimizer is still dependent on training choices, it remains an open question to disentangle how exactly the learned data-dependent features contribute to the effects explained in our work.

References

1. Allen-Zhu, Z., Li, Y., Song, Z.: A convergence theory for deep learning via over-parameterization. In: International Conference on Machine Learning. pp. 242–252 (2019)
2. Allen-Zhu, Z., Li, Y., Song, Z.: On the convergence rate of training recurrent neural networks. In: Advances in Neural Information Processing Systems. pp. 6673–6685 (2019)
3. Amari, S., Ba, J., Grosse, R.B., Li, X., Nitanda, A., Suzuki, T., Wu, D., Xu, J.: When does preconditioning help or hurt generalization? In: International Conference on Learning Representations (2021), https://openreview.net/forum?id=S724o4_WB3
4. Arora, S., Du, S., Hu, W., Li, Z., Wang, R.: Fine-grained analysis of optimization and generalization for overparameterized two-layer neural networks. In: International Conference on Machine Learning. pp. 322–332 (2019)
5. Arora, S., Du, S.S., Hu, W., Li, Z., Salakhutdinov, R.R., Wang, R.: On exact computation with an infinitely wide neural net. In: Advances in Neural Information Processing Systems. pp. 8139–8148 (2019)
6. Belkin, M., Hsu, D.J., Mitra, P.: Overfitting or perfect fitting? risk bounds for classification and regression rules that interpolate. In: Advances in Neural Information Processing Systems. pp. 2300–2311 (2018)

7. Belkin, M., Ma, S., Mandal, S.: To understand deep learning we need to understand kernel learning. In: International Conference on Machine Learning. pp. 541–549 (2018)
8. Belkin, M., Rakhlin, A., Tsybakov, A.B.: Does data interpolation contradict statistical optimality? In: The 22nd International Conference on Artificial Intelligence and Statistics. pp. 1611–1619 (2019)
9. Bietti, A., Mairal, J.: On the inductive bias of neural tangent kernels. In: Advances in Neural Information Processing Systems. pp. 12873–12884 (2019)
10. Borovykh, A.: The effects of optimization on generalization in infinitely wide neural networks. ICML Workshop (2019)
11. Chizat, L., Oyallon, E., Bach, F.: On lazy training in differentiable programming. In: Advances in Neural Information Processing Systems. pp. 2937–2947 (2019)
12. Daniely, A.: SGD learns the conjugate kernel class of the network. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (eds.) Advances in Neural Information Processing Systems 30, pp. 2422–2430. Curran Associates, Inc. (2017)
13. De Matthews, A., Hron, J., Rowland, M., Turner, R., Ghahramani, Z.: Gaussian process behaviour in wide deep neural networks. In: 6th International Conference on Learning Representations, ICLR 2018-Conference Track Proceedings (2018)
14. Du, S., Lee, J., Li, H., Wang, L., Zhai, X.: Gradient descent finds global minima of deep neural networks. In: International Conference on Machine Learning. pp. 1675–1685 (2019)
15. Du, S.S., Zhai, X., Poczos, B., Singh, A.: Gradient descent provably optimizes over-parameterized neural networks. In: International Conference on Learning Representations (2019), https://openreview.net/forum?id=S1eK3i09YQ
16. Duchi, J., Hazan, E., Singer, Y.: Adaptive subgradient methods for online learning and stochastic optimization. Journal of machine learning research **12**(Jul), 2121–2159 (2011)
17. Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. In: Proceedings of the thirteenth international conference on artificial intelligence and statistics. pp. 249–256 (2010)
18. He, K., Zhang, X., Ren, S., Sun, J.: Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In: Proceedings of the IEEE international conference on computer vision. pp. 1026–1034 (2015)
19. Hinton, G.: Lecture 6 of the online course "neural networks for machine learning". Lecture 6 of the online course "Neural Networks for Machine Learning" (2012), https://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf
20. Jacot, A., Gabriel, F., Hongler, C.: Neural tangent kernel: Convergence and generalization in neural networks. In: Advances in neural information processing systems. pp. 8571–8580 (2018)
21. Lee, J., Sohl-Dickstein, J., Pennington, J., Novak, R., Schoenholz, S., Bahri, Y.: Deep neural networks as gaussian processes. In: International Conference on Learning Representations (2018), https://openreview.net/forum?id=B1EA-M-0Z
22. Lee, J., Xiao, L., Schoenholz, S., Bahri, Y., Novak, R., Sohl-Dickstein, J., Pennington, J.: Wide neural networks of any depth evolve as linear models under gradient descent. In: Advances in neural information processing systems. pp. 8570–8581 (2019)
23. Li, Y., Liang, Y.: Learning overparameterized neural networks via stochastic gradient descent on structured data. In: Advances in Neural Information Processing Systems. pp. 8157–8166 (2018)
24. Liang, T., Rakhlin, A.: Just interpolate: Kernel "ridgeless" regression can generalize. To appear in The Annals of Statistics (preprint arXiv:1808.00387) (2019)

25. Ma, S., Bassily, R., Belkin, M.: The power of interpolation: Understanding the effectiveness of SGD in modern over-parametrized learning. In: International Conference on Machine Learning. pp. 3325–3334 (2018)
26. Nonnenmacher, M., Reeb, D., Steinwart, I.: Which minimizer does my neural network converge to? arXiv preprint arXiv:2011.02408 (2020)
27. Oymak, S., Soltanolkotabi, M.: Overparameterized nonlinear learning: Gradient descent takes the shortest path? In: International Conference on Machine Learning. pp. 4951–4960 (2019)
28. Qian, Q., Qian, X.: The implicit bias of adagrad on separable data. arXiv preprint arXiv:1906.03559 (2019)
29. Rahaman, N., Baratin, A., Arpit, D., Draxler, F., Lin, M., Hamprecht, F., Bengio, Y., Courville, A.: On the spectral bias of neural networks. In: International Conference on Machine Learning. pp. 5301–5310 (2019)
30. Sankararaman, K.A., De, S., Xu, Z., Huang, W.R., Goldstein, T.: The impact of neural network overparameterization on gradient confusion and stochastic gradient descent. arXiv preprint arXiv:1904.06963 (2019)
31. Shah, V., Kyrillidis, A., Sanghavi, S.: Minimum norm solutions do not always generalize well for over-parameterized problems. arXiv preprint arXiv:1811.07055 (2018)
32. Soudry, D., Hoffer, E., Nacson, M.S., Gunasekar, S., Srebro, N.: The implicit bias of gradient descent on separable data. The Journal of Machine Learning Research **19**(1), 2822–2878 (2018)
33. Vaswani, S., Babanezhad, R., Gallego, J., Mishkin, A., Lacoste-Julien, S., Roux, N.L.: To each optimizer a norm, to each norm its generalization. arXiv preprint arXiv:2006.06821 (2020)
34. Wilson, A.C., Roelofs, R., Stern, M., Srebro, N., Recht, B.: The marginal value of adaptive gradient methods in machine learning. In: Advances in Neural Information Processing Systems. pp. 4148–4158 (2017)
35. Xiao, L., Pennington, J., Schoenholz, S.: Disentangling trainability and generalization in deep neural networks. In: International Conference on Machine Learning. pp. 10462–10472. PMLR (2020)
36. Zhang, C., Bengio, S., Hardt, M., Mozer, M.C., Singer, Y.: Identity crisis: Memorization and generalization under extreme overparameterization. In: International Conference on Learning Representations (2020), https://openreview.net/forum?id=B1l6y0VFPr
37. Zhang, C., Bengio, S., Hardt, M., Recht, B., Vinyals, O.: Understanding deep learning requires rethinking generalization. In: International Conference on Learning Representations (2017), https://openreview.net/forum?id=Sy8gdB9xx
38. Zhang, Y., Xu, Z.Q.J., Luo, T., Ma, Z.: A type of generalization error induced by initialization in deep neural networks. In: Mathematical and Scientific Machine Learning. pp. 144–164. PMLR (2020)