Stability & Generalisation of Gradient Descent for Shallow Neural Networks without the Neural Tangent Kernel

Anonymous Author(s)
Affiliation
Address
email

Abstract

We revisit on-average algorithmic stability of Gradient Descent (GD) for training overparameterised shallow neural networks and prove new generalisation and excess risk bounds without the Neural Tangent Kernel (NTK) or Polyak-Łojasiewicz (PL) assumptions. In particular, we show oracle type bounds which reveal that the generalisation and excess risk of GD is controlled by an interpolating network with the shortest GD path from initialisation (in a sense, an interpolating network with the smallest relative norm). While this was known for kernelised interpolants, our proof applies directly to networks trained by GD without intermediate kernelisation. At the same time, by relaxing oracle inequalities developed here we recover existing NTK-based risk bounds in a straightforward way, which demonstrates that our analysis is tighter. Finally, unlike most of the NTK-based analyses we focus on regression with label noise and show that GD with early stopping is *consistent*.

1 Introduction

A canonical problem in statistical learning has the learner given a tuple of independently sampled training examples $S = (z_1, \ldots, z_n)$, where each example $z_i = (\mathbf{x}_i, y_i)$ consists of an *input* $\mathbf{x}_i$ distributed according to an unknown distribution P_X supported on an Euclidean ball $\mathcal{B}_2^d(C_x)$ of radius C_x, and a *label* generated by the *regression function* as

$$y_i = f^\star(\mathbf{x}_i) + \varepsilon_i \qquad i \in [n] \, .$$

Where here $f^\star : \mathcal{B}_2^d(C_x) \to [-1, 1]$ is some ground truth function and ε_i is a zero-mean bounded noise with variance $\sigma^2 \geq 0$, such that $|y_i| = C_y$ almost surely for a constant $C_y \geq 0$. Based on training examples the goal of the learner is to select *parameters* $\mathbf{W}$ from some parameter space $\mathcal{W}$ in order to minimise the *statistical risk*

$$\mathcal{L}(\mathbf{W}) \stackrel{\text{def}}{=} \frac{1}{2}\mathbf{E}_{S,\mathbf{x},y}\left[\left(f_{\mathbf{W}}(\mathbf{x}) - y\right)^2\right]$$

where $(\mathbf{x}, y)$ is an independent copy of $(\mathbf{x}_1, y_1)$ and $f_{\mathbf{W}}$ is a predictor parameterised by $\mathbf{W}$. In this paper we will focus on a *shallow neural network* predictor that takes the form

$$f_{\mathbf{W}}(\mathbf{x}) \stackrel{\text{def}}{=} \sum_{k=1}^{m} u_k \phi\left(\langle \mathbf{W}_k, \mathbf{x}\rangle\right) \qquad \left(\mathbf{x} \in \mathbb{R}^d, \ \mathbf{W} \in \mathbb{R}^{d\times m}\right)$$

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

defined with respect to some *activation function* $\phi : \mathbb{R} \to \mathbb{R}$, fixed *output layer* $\mathbf{u} \in \{\pm 1/\sqrt{m}\}^m$, and a tunable (possibly randomised) *hidden layer* $\mathbf{W}$. In particular, we will consider $f_{\mathbf{W}_T}$, where the hidden layer is obtained by minimising an empirical proxy of $\mathcal{L}$ called the *empirical risk* $\mathcal{L}_S(\mathbf{W}) \stackrel{\text{def}}{=} \frac{1}{2n}\sum_{i=1}^n (f_{\mathbf{W}}(\mathbf{x}_i) - y_i)^2$ by running a *Gradient Descent (GD)* procedure: For $t \in [T]$ steps with initial parameters $\mathbf{W}_0$ and a step size $\eta > 0$, we have iterates $\mathbf{W}_{t+1} = \mathbf{W}_t - \eta \nabla \mathcal{L}_t(\mathbf{W}_t)$ where $\nabla \mathcal{L}_t(\mathbf{W})$ is the first order derivative of $\mathcal{L}_t(\mathbf{W})$.

Understanding the behaviour of the statistical risk for neural networks has been a long-standing topic of interest in the statistical learning theory [Anthony and Bartlett, 1999]. The standard approach to this problem is based on *uniform* bounds on the *generalisation gap*

$$\epsilon^{\text{Gen}}(\mathbf{W}_T) \stackrel{\text{def}}{=} \mathcal{L}(\mathbf{W}_T) - \mathcal{L}_S(\mathbf{W}_T) \,,$$

which, given a parameter space $\mathcal{W}$, involves controlling the gap for the worst possible choice of $\mathbf{W} \in \mathcal{W}$ under some unknown data distribution. The theory then typically leads to the capacity-based (Rademacher complexity, VC-dimension, or metric-entropy based) bounds which hold with high probability (w.h.p.) over S [Bartlett and Mendelson, 2002, Golowich et al., 2018]: [1]

$$\sup_{\mathbf{W} \in \mathcal{W}} |\epsilon^{\text{Gen}}(\mathbf{W})| \lesssim \sqrt{\frac{\text{capacity}(\mathcal{W})}{n}} \,. \tag{1}$$

Thus, if one could simultaneously control the empirical risk and the capacity of the class of neural networks, one could control the statistical risk. Unfortunately, controlling the empirical risk $\mathcal{L}_T$ turns out to be a challenging part here since it is non-convex, and thus, it is not clear whether GD can minimise it up to a desired precision.

This issue has attracted considerable attention in recent years with numerous works [Du et al., 2018, Lee et al., 2019, Allen-Zhu et al., 2019, Oymak and Soltanolkotabi, 2020] demonstrating that *overparameterised* shallow networks (in a sense $m \gtrsim \text{poly}(n)$) trained on subgaussian inputs converge to global minima exponentially fast, namely, $\mathcal{L}_S(\mathbf{W}_T) \lesssim (1 - \eta \cdot d/n)^T$. Loosely speaking, these proofs are based on the idea that a sufficiently overparameterised network trained by GD initialised at $\mathbf{W}_0$ with Gaussian entries, predicts closely to a solution of a Kernelised Least-Squares (KLS) formulation minimised by GD, where the kernel function called the *Neural Tangent Kernel (NTK)* [Jacot et al., 2018] is implicitly given by the activation function. This connection explains the observed exponential rate in case of shallow neural networks: For the NTK kernel matrix $\mathbf{K}$ the convergence rate of GD is $(1 - \eta\lambda_{\min}(\mathbf{K}))^T$, where $\lambda_{\min}(\mathbf{K})$ is its smallest eigenvalue, and as it turns out, for subgaussian inputs $\lambda_{\min}(\mathbf{K}) \gtrsim d/n$ [Bartlett et al., 2021]. [2] Naturally, the convergence was exploited to state bounds on the statistical risk: Arora et al. [2019] showed that for noise-free regression ($\sigma^2 = 0$) when $T \gtrsim 1/(\eta\lambda_{\min}(\mathbf{K}))$,

$$\mathcal{L}(\mathbf{W}_T) \lesssim \sqrt{\frac{\langle \mathbf{y}, (n\mathbf{K})^{-1}\mathbf{y}\rangle}{n}} \,. \tag{2}$$

Clearly, the bound is non-vacuous whenever $\langle \mathbf{y}, \mathbf{K}^{-1}\mathbf{y}\rangle \lesssim n^{2\alpha}$ for some $\alpha \in [0, 1)$, and Arora et al. [2019] present several examples of smooth target functions which satisfy this. More generally, for $f^\star$ which belongs to the Reproducing kernel Hilbert space (RKHS) induced by NTK one has $\langle \mathbf{y}, \mathbf{K}^{-1}\mathbf{y}\rangle \leq \|f^\star\|^2_{\mathcal{H}_{\text{NTK}}}$ [Schölkopf and Smola, 2002]. The norm-based control of the risk is standard in the literature on kernels, and thus, one might wonder to which extent neural networks are kernel methods in disguise? At the same time, some experimental and theoretical evidence [Bai and Lee, 2019, Seleznova and Kutyniok, 2020, Suzuki and Akiyama, 2021] suggest that connection to kernels might be good only at explaining the behaviour of very wide networks, much more overparameterised than those used in practice. Therefore, an interesting possibility is to develop alternative ways to analyse generalisation in neural networks: Is there a more straightforward kernel-free optimisation-based perspective?

[1] Throughout this paper, we use $f \lesssim g$ to say that there exists a universal constant $c > 0$ and some $k \in \mathbb{N}$ such that $f \leq cg \log^k(g)$ holds uniformly over all arguments.

[2] Which is a tightest known bound, Oymak and Soltanolkotabi [2020] prove a looser bound without distributional assumption on the inputs.

In this paper we take a step in this direction and explore a kernel-free approach, which at the same time avoids worst-case type uniform bounds such as Eq. (1). In particular, we focus on the notion of the *algorithmic stability*: If an algorithm is insensitive to replacement (or removal) of an observation in a training tuple, then it must have a small generalisation gap. Thus, a natural question is whether GD is sufficiently stable when training overparameterised neural networks.

The stability of GD (and its stochastic counterpart) when minimising convex and non-convex smooth objective functions was first explored by Hardt et al. [2016]. Specifically, for a time-dependent choice of a step size $\eta_t = 1/t$ and a problem-dependent constant $\alpha \in (0, 1)$ they show that

$$\mathbf{E}\left[\epsilon^{\text{Gen}}(\mathbf{W}_T) \mid \mathbf{W}_0, \mathbf{u}\right] \lesssim \ln(T) n^{-\alpha} .$$

Unfortunately, when combined with the NTK-based convergence rate of the empirical risk we have a vacuous bound since $\mathcal{L}_S(\mathbf{W}_T) \lesssim 1$. [3] This is because the stability is enforced through a quickly decaying step size rather than by exploiting a finer structure of the loss, and turns out to be insufficient to guarantee the convergence of the empirical risk. That said, several works Charles and Papailiopoulos [2018], Lei and Ying [2021] have proved stability bounds exploiting an additional structure in mildly non-convex losses. Specifically, they studied the stability of GD minimising a *gradient-dominated* empirical risk, meaning that for all $\mathbf{W}$ in some neighbourhood $\mathcal{W}$ and a problem-dependent quantity μ, it is assumed that $\mathcal{L}_S(\mathbf{W}) - \min_{\mathbf{W}'} \mathcal{L}_S(\mathbf{W}') \leq \|\nabla \mathcal{L}_S(\mathbf{W})\|^2/(2\mu)$. Having iterates of GD within $\mathcal{W}$ and assuming that the gradient is ρ-Lipschitz, this allows to show

$$\mathbf{E}\left[\epsilon^{\text{Gen}}(\mathbf{W}_T) \mid \mathbf{W}_0, \mathbf{u}\right] \lesssim \frac{\rho}{\mu} \cdot \frac{1}{n} .$$

As it turns out, the condition is satisfied for the iterates of GD training overparameterised networks [Du et al., 2018] with high probability over $\mathbf{W}_0$. The key quantity that controls the bound is $\rho/\mu \lesssim 1/\lambda_{\min}(\mathbf{K})$ which can be interpreted as a *condition number* of the NTK matrix. However, it is known that for subgaussian inputs the condition number behaves as n/d which renders the bound vacuous [Bartlett et al., 2021].

2 Our Contributions

In this paper we revisit algorithmic stability of GD for training overparameterised shallow neural networks, and prove new risk bounds **without the Neural Tangent Kernel (NTK) or Polyak-Łojasiewicz (PL)** machinery. In particular, we first show a bound on the generalisation gap and then specialise it to state risk bounds for regression with and without label noise. In the case of learning with noise we demonstrate that GD with a form of early stopping is *consistent*, meaning that the risk asymptotically converges to the noise rate σ^2.

Our analysis brings out a key quantity, which controls all our bounds, the *Regularised Empirical Risk Minimisation (R-ERM) Oracle* defined as

$$\Delta_S^{\text{oracle}} \stackrel{\text{def}}{=} \min_{\mathbf{W} \in \mathbb{R}^{d \times m}} \mathcal{L}_S(\mathbf{W}) + \mathcal{O}\left(\frac{\|\mathbf{W} - \mathbf{W}_0\|_F^2}{\eta T}\right) \quad \text{as} \quad \eta T \to \infty , \tag{3}$$

which means that Δ_S^{oracle} is essentially an empirical risk of solution closest to initialisation (an interpolant when for m large enough). We first consider a bound on the generalisation gap.

2.1 Generalisation Gap

For simplicity of presentation in the following we assume that the activation function ϕ, its first and second derivatives are bounded. Assuming parameterisation $m \gtrsim (\eta T)^5$ we show (Corollary 1) that the expected generalisation gap is bounded as

$$\mathbf{E}\left[\epsilon^{\text{Gen}}(\mathbf{W}_T) \mid \mathbf{W}_0, \mathbf{u}\right] \leq C \cdot \frac{\eta T}{n}\left(1 + \frac{\eta T}{n}\right) \mathbf{E}\left[\Delta_S^{\text{oracle}} \mid \mathbf{W}_0, \mathbf{u}\right] , \tag{4}$$

[3] If $\eta_s = \frac{1}{s}$ we have $\mathcal{L}_S(\mathbf{W}_T) \lesssim \exp(\mu \sum_{j=1}^{T} \frac{1}{j}) \approx \frac{1}{T^\mu}$, thus, if $\mu \approx \frac{1}{n}$ we then require $T \sim \epsilon^{-n}$ for $\mathcal{L}_S(\mathbf{W}_T) \lesssim \epsilon$. Plugging this into the Generalisation Error bound we get that $\log(T) n^{-\alpha} = n^{1-\alpha} \log(1/\epsilon)$ which is vacuous as n grows.

where C is a constant independent from n, T, η.

Dependence of m on the total number of steps T might appear strange at first, however things clear out once we pay attention to the scaling of the bound. Setting the step size η to be constant, $T = n^\alpha$, and overparameterisation $m \gtrsim n^{5\alpha}$ for some free parameter $\alpha \in (0, 1]$ we have

$$\mathbf{E}\left[\epsilon^{\mathrm{Gen}}(\mathbf{W}_T) \mid \mathbf{W}_0, \mathbf{u}\right] = \mathcal{O}\left(\frac{1}{n^\alpha} \cdot \mathbf{E}\left[\|\hat{\mathbf{W}} - \mathbf{W}_0\|_F^2 \;\middle|\; \mathbf{W}_0, \mathbf{u}\right]\right) \quad \text{as} \quad n \to \infty\,,$$

where $\hat{\mathbf{W}}$ is chosen as parameters of a *minimal-norm interpolating network*, in a sense $\hat{\mathbf{W}} \in \operatorname{argmin}_{\mathbf{W} \in \mathbb{R}^{d \times m}} \{\|\mathbf{W} - \mathbf{W}_0\|_F^2 \mid \mathcal{L}_S(\mathbf{W}) = 0\}$. Thus, as Eq. (4) suggests, the generalisation gap is controlled by a minimal relative norm of an interpolating network. In comparison, previous work in the NTK setting obtained results of a similar type where in place of $\hat{\mathbf{W}}$ one has an interpolating minimal-norm solution to an NTK least-squares problem. Specifically, in Eq. (2) the generalisation gap is controlled by $\left\langle \mathbf{y}, (n\mathbf{K})^{-1}\mathbf{y} \right\rangle$, which is a squared norm of such solution.

The second observation we make is that there is a trade-off between convergence rate and overparameterisation. Indeed, decreasing α also decreases the exponent of overparameterisation, but also decreases the rate at which generalisation gap approaches 0.

The generalisation gap of course only tells us a partial story, since the behavior of the empirical risk is unknown. Next, we take care of this and present bounds *excess risk* bounds.

2.2 Risk Bound without Label Noise

We first present a bound on the statistical risk which does not use the usual NTK arguments to control the empirical risk. For now assume that $\sigma^2 = 0$ meaning theat there is no label noise and randomness is only in the inputs.

NTK-free Risk Bound. In Corollary 2 we show that the risk is bounded as

$$\mathbf{E}\left[\mathcal{L}(\mathbf{W}_T) \mid \mathbf{W}_0, \mathbf{u}\right] \leq \left(1 + C \cdot \frac{\eta T}{n}\left(1 + \frac{\eta T}{n}\right)\right) \mathbf{E}\left[\Delta_S^{\mathrm{oracle}} \mid \mathbf{W}_0, \mathbf{u}\right]\,. \tag{5}$$

Note that this bound looks very similar compared to the bound on generalisation gap. The difference lies in a "1+" term which accounts for the fact that $\mathcal{L}(\mathbf{W}_T) \leq \Delta_S^{\mathrm{oracle}}$ as we show in Lemma 1.

As before we let the step size be constant, set $T = n^\alpha$, and let overparameterisation be $m \gtrsim n^{5\alpha}$ for some $\alpha \in (0, 1]$. Then our risk bound implies that

$$\mathbf{E}\left[\mathcal{L}(\mathbf{W}_T) \mid \mathbf{W}_0, \mathbf{u}\right] = \mathcal{O}\left(\frac{1}{n^\alpha} \cdot \mathbf{E}\left[\|\hat{\mathbf{W}} - \mathbf{W}_0\|_F^2 \;\middle|\; \mathbf{W}_0, \mathbf{u}\right]\right) \quad \text{as} \quad n \to \infty\,,$$

which comes as a simple consequence of bounding $\Delta_S^{\mathrm{oracle}}$ while choosing $\hat{\mathbf{W}}$ to be parameters of a minimal relative norm interpolating network as before. Recall that $y_i = f^\star(\mathbf{x}_i)$: The bound suggests that target functions $f^\star$ are only learnable if $n^{-\alpha}\mathbf{E}\left[\|\hat{\mathbf{W}} - \mathbf{W}_0\|_F^2 \;\middle|\; \mathbf{W}_0, \mathbf{u}\right] \to 0$ as $n \to \infty$ for a fixed α and input distribution. A natural question is whether a class of such functions is non-empty. Indeed, the NTK theory suggests that it is not and in the following we will recover NTK-based results by relaxing our oracle inequality.

Similarly as in Section 2.1 we choose $T = n^\alpha$ and so we require $m \gtrsim n^{5\alpha}$, which trades off overparameterisation and convergence rate through the choice of $\alpha \in (0, 1]$. To the best of our knowledge this is the first result of this type, where one can achieve a smaller overparameterisation compared to the NTK literature at the expense of a slower convergence rate of the risk. Finally, unlike the NTK setting we do not require randomisation of the initialisation and the risk bound we obtain holds for any $\mathbf{W}_0$. [4]

[4] In the NTK literature randomisation of $\mathbf{W}_0$ is required to guarantee that $\lambda_{\min}(\mathbf{K}) > 0$ [Du et al., 2018].

Comparison to NTK-based Risk Bound. Our generalisation bound can be naturally relaxed to obtain NTK-based empirical risk convergence rates. In this case we observe that our analysis is general enough to recover results of Arora et al. [2019] (up to a difference that our bounds hold in expectation rather than in high probability).

By looking at Eq. (5), our task is in controlling Δ_S^{oracle}: by its definition in Eq. (3) we can see that it can be bounded by the regularised empirical risk of any model, and we choose a Moore-Penrose pseudo-inverse solution to the least-squares problem supplied with the NTK feature map $\mathbf{x} \mapsto (\nabla_{\mathbf{W}} f_{\mathbf{W}}(\mathbf{x}))(\mathbf{W}_0)$. Here, entries of $\mathbf{W}_0$ are drawn from $\mathcal{N}(0, \nu_{\text{init}}^2)$ for some appropriately chosen ν_{init}^2, and entries of the outer layer $\mathbf{u}$ are distributed according to the uniform distribution over $\{\pm 1/\sqrt{m}\}$, independently from each other and the data. It is not hard to see that aforementioned pseudo-inverse solution $\mathbf{W}^{\text{pinv}}$ will have zero empirical risk, and so we are left with controlling $\|\mathbf{W}^{\text{pinv}} - \mathbf{W}_0\|_F^2$ as can be seen from the definition of Δ_S^{oracle}. Note that it is straightforward to do since $\mathbf{W}^{\text{pinv}}$ has an analytic form. In Theorem 2 we carry out these steps and show that with high probability over initialisation $(\mathbf{W}_0, \mathbf{u})$,

$$\Delta_S^{\text{oracle}} = \tilde{\mathcal{O}}_P\left(\frac{1}{n}\left\langle \mathbf{y}, (n\mathbf{K})^{-1}\mathbf{y}\right\rangle\right) \quad \text{as} \quad n \to \infty\,,$$

which combined with Eq. (5) recovers the result of Arora et al. [2019]. Note that we had to *relax* the oracle inequality, which suggests that the risk bound we give here is *tighter* compared to the NTK-based bound. Similarly, Arora et al. [2019] demonstrated that $\|\mathbf{W}_T - \mathbf{W}_0\|_F^2 = \tilde{\mathcal{O}}_P\left(\left\langle \mathbf{y}, (n\mathbf{K})^{-1}\mathbf{y}\right\rangle\right)$ holds w.h.p. over initialisation and ReLU activation function, however their proof requires a much more involved "coupling" argument where iterates $(\mathbf{W}_t)_{t=1}^T$ are shown to be close to the GD iterates minimising a KLS problem with NTK matrix.

2.3 Risk Bound with Label Noise and Consistency

So far we considered regression with a random inputs, but without label noise. In this section we use our bounds to we show that GD with early stopping is *consistent* in the regression with label noise. Here labels are generated as $y_i = f^\star(\mathbf{x}_i) + \varepsilon_i$ where zero-mean random variables $(\varepsilon_i)_{i=1}^n$ are i.i.d., almost surely *bounded*,[5] and $\mathbf{E}[\varepsilon_1^2] = \sigma^2$.

For now we will use the same settings of parameters as in Section 2.2: Recall that that η is constant, $T = n^\alpha$, $m \gtrsim n^{5\alpha}$ for a free parameter $\alpha \in (0, 1]$. In addition assume a *well-specified* scenario which means that m is large enough such that for some subset of parameters $\mathcal{W}$ we achieve $\mathcal{L}(\mathbf{W}^\star) = \sigma^2$ for some parameters $\mathbf{W}^\star$. Employing our risk bound of Eq. (5), we relax R-ERM oracle as

$$\mathbf{E}[\Delta_S^{\text{oracle}} \mid \mathbf{W}_0, \mathbf{u}] \lesssim \sigma^2 + n^{-\alpha}\|\mathbf{W}^\star - \mathbf{W}_0\|_F^2\,, \quad \text{where} \quad \mathbf{W}^\star \in \operatorname{argmin}_{\mathbf{W}\in\mathbb{R}^{d\times m}} \mathcal{L}(\mathbf{W})\,.$$

Then we immediately have an α-dependent risk bound

$$\mathbf{E}\left[\mathcal{L}(\mathbf{W}_T) \mid \mathbf{W}_0, \mathbf{u}\right] \leq \left(1 + \frac{C}{n^{1-\alpha}}\right)\left(\sigma^2 + \mathcal{O}\left(\frac{\|\mathbf{W}^\star - \mathbf{W}_0\|_F^2}{n^\alpha}\right)\right) \quad \text{as} \quad n \to \infty\,. \tag{6}$$

It is important to note that for any tuning $\alpha \in (0, 1)$, as long as $\mathbf{W}^\star$ is constant in n (which is the case in the parametric setting), we have $\mathbf{E}\left[\mathcal{L}(\mathbf{W}_T) \mid \mathbf{W}_0, \mathbf{u}\right] \to \sigma^2$ as $n \to \infty$, and so we achieve *consistency*. Adaptivity to the noise achieved here is due to *early stopping*: Indeed, recall that we take $T = n^\alpha$ steps and having smaller α mitigates the effect of the noise as can be seen from Eq. (6). This should come as no surprise as early stopping is well-known to have a regularising effect in GD methods [Yao et al., 2007]. The current literature on the NTK deals with this through a kernelisation perspective by introducing an explicit $L2$ regularisation [Hu et al., 2021], while risk bound of [Arora et al., 2019] designed for a noise-free setting would be vacuous in this case.

2.4 Future Directions and Limitations

We presented generalisation and risk bounds for shallow nets controlled by the Regularised Empirical Risk Minimisation oracle Δ_S^{oracle}. By straightforward relaxation of Δ_S^{oracle}, we showed that the risk

[5] We assume boundedness of labels throughout our analysis to ensure that $\mathcal{L}_S(\mathbf{W}_0)$ is bounded almost surely for simplicity. This can be relaxed by introducing randomised initialisation, e.g. as in [Du et al., 2018].

can be controlled by the minimal (relative) norm of an interpolating network which is tighter than the NTK-based bounds. There are several interesting venues which can be explored based on our results. For example, by assuming a specific form of a target function $f^\star$ (e.g., a "teacher" network), one possibility is to characterise the minimal norm. This would allow us to understand better which such target function are *learnable* by GD.

One limitation of our analysis is smoothness of the activation function and its boundedness. While boundedness can be easily dealt with by randomising $(\mathbf{W}_0, \mathbf{u})$, extending our analysis to non-smooth activations (such as ReLU $\phi(x) = \max(x, 0)$) appears to be non-trivial because our stability analysis crucially relies on the control of the smallest eigenvalue of the Hessian. Another peculiarity of our analysis is a time-dependent parameterisation requirement $m \gtrsim (\eta T)^5$. Making parameterisation n-dependent introduces an implicit early-stopping condition, which helped us to deal with label noise. On the other hand, such requirement might be limiting when dealing with noiseless interpolating learning where one could require to have $\eta T \to \infty$ while keeping n finite.

3 Main Result and Proof Sketch

This section presents both the main results as well as a proof sketch. Precisely, Section 3.1 presents the main result and Section 3.2 the proof sketch.

3.1 Main Results

We formally make the following assumption regarding the regularity of the activation function.

Assumption 1 (Activation)**.** *The activation $\phi(u)$ continuous and twice differentiable with constant $B_\phi, B_{\phi'}, B_{\phi''} \geq 0$ bounding $\phi(u) \leq B_\phi$, $\phi'(u) \leq B_{\phi'}$ and $\phi''(u) \leq B_{\phi''}$ for any $u \in \mathbb{R}$.*

This is satisfied for sigmoid as well as hyperbolic tangent activations.

Assumption 2 (Inputs, labels, and the loss function)**.** *For constants $C_x, C_y, C_0 > 0$, inputs belong to $\mathcal{B}_2^d(C_x)$, labels belong to $[-C_y, C_y]$, and loss is uniformely bounded by C_0 almost surely.*

A straightforward consequence of the above is smoothness of the loss (see Eq. (11) for the proof).

Proposition 1 (Smoothness)**.** *$\nabla \mathcal{L}_S$ is ρ-Lipschitz, where $\rho \stackrel{\text{def}}{=} C_x^2(B_{\phi'}^2 + B_{\phi''}B_\phi + {}^{B_{\phi''}C_y}/\sqrt{m})$.*

Given these assumptions we are ready to present the main theorem a bound on for the gradient descent iterates.

Theorem 1 (Generalisation Error)**.** *Consider Assumptions 1 and 2. Fix $t > 0$. If $\eta \leq 1/(2\rho)$ and*

$$m \geq 144(\eta t)^2 C_x^4 C_0^2 B_{\phi''}^2 \left(4B_{\phi'}C_x\sqrt{\eta t} + \sqrt{2}\right)^2 \tag{7}$$

$$\text{then} \qquad \mathbf{E}\left[\epsilon^{Gen}(\mathbf{W}_{t+1}) \,\middle|\, \mathbf{W}_0, \mathbf{u}\right] \leq b\left(\frac{\eta}{n} + \frac{\eta^2 t}{n^2}\right)\sum_{j=0}^{t}\mathbf{E}\left[\mathcal{L}_S(\mathbf{W}_j) \mid \mathbf{W}_0, \mathbf{u}\right]$$

where $b = 16e^3 C_x^{\frac{3}{2}} B_{\phi'}^2 (1 + C_x^{\frac{3}{2}} B_{\phi'}^2)$.

Proof. For full proof see Appendix C with sketch proof in Section 3.2. □

Theorem 1 then provides a formal upper bound on the Generalisation Gap of a shallow neural network trained with GD. Specifically, provided the network is sufficiently wide the generalisation gap can be controlled through both: the step size and number of iterations through the factor alongside the trajectory taken by gradient through the Empirical Risk $\sum_{t=0}^{T}\mathbf{E}[\mathcal{L}_S(\mathbf{W}_t) \mid \mathbf{W}_0, \mathbf{u}]$. Similar bounds for the test performance of graident descent have been given by Lei and Ying [2020], although in our case the risk is non-convex, and thus, a more delicate bound on the Optimisation Gap is required. This is summarised within the lemma, which is presented in an "oracle" inequality manner.

Lemma 1 (Optimisation Error)**.** *Consider Assumptions 1 and 2. Fix $t > 0$. If $\eta \leq 1/(2\rho)$, then*

$$\frac{1}{t}\sum_{j=0}^{t}\mathcal{L}_S(\mathbf{W}_j) \leq \min_{\mathbf{W}\in\mathbb{R}^{d\times m}}\left\{\mathcal{L}_S(\mathbf{W}) + \frac{\|\mathbf{W}-\mathbf{W}_0\|_F^2}{\eta t} + \frac{\widetilde{b}\|\mathbf{W}-\mathbf{W}_0\|_F^3}{\sqrt{m}}\right\} + \widetilde{b}C_0\cdot\frac{(\eta t)^{\frac{3}{2}}}{\sqrt{m}}$$

where $\widetilde{b} = C_x^2 B_{\phi''}\left(B_{\phi'}C_x + C_0\right)$.

Proof. Full proof is given in Appendix B with sketch proof in Section 3.2. □

By combining Theorem 1 and Lemma 1 we get the following where Δ_S^{oracle} is defined in Eq. (3):

Corollary 1. *Assume the same as in Theorem 1 and Lemma 1. Then,*

$$\mathbf{E}\left[\epsilon^{\text{Gen}}(\mathbf{W}_T) \mid \mathbf{W}_0\right] \leq C\cdot\frac{\eta T}{n}\left(1+\frac{\eta T}{n}\right)\mathbf{E}\left[\Delta_S^{\text{oracle}} \mid \mathbf{W}_0\right]$$

where Δ_S^{oracle} is defined in Eq. (3) and C is a constant independent from n, T, η.

The above combined with Lemma 1, and the fact that $\mathcal{L}_S(\mathbf{W}_T) = \min_{t\in[T]}\mathcal{L}_S(\mathbf{W}_t)$ gives us:

Corollary 2. *Assume the same as in Theorem 1 and Lemma 1. Then,*

$$\mathbf{E}\left[\mathcal{L}(\mathbf{W}_T) \mid \mathbf{W}_0\right] \leq \left(1 + C\cdot\frac{\eta T}{n}\left(1+\frac{\eta T}{n}\right)\right)\mathbf{E}\left[\Delta_S^{\text{oracle}} \mid \mathbf{W}_0\right] .$$

Finally, Δ_S^{oracle} is controlled by the norm of the NTK solution, which establishes connection to the NTK-based risk bounds:

Theorem 2 (Connection between Δ_S^{oracle} and NTK)**.** *Consider Assumption 1 and that $\eta T = n$. Moreover, assume that entries of $\mathbf{W}_0$ are i.i.d., that $\lambda_{\min}(\mathbf{K}) \gtrsim 1/n$, and assume that $\mathbf{u} \sim \text{unif}\left(\{\pm 1/\sqrt{m}\}\right)^m$ independently from all sources of randomness. Then, with probability least $1-\delta$ for $\delta \in (0,1)$, over $(\mathbf{W}_0, \mathbf{u})$,*

$$\Delta_S^{\text{oracle}} = \tilde{\mathcal{O}}_P\left(\frac{1}{n}\left\langle \mathbf{y}, (n\mathbf{K})^{-1}\mathbf{y}\right\rangle\right) \quad \text{as} \quad n\to\infty .$$

Proof. The proof is given in Appendix D. □

3.2 Proof Sketch of Theorem 1 and Lemma 1

Throughout this sketch let expectation be understood as $\mathbf{E}[\cdot] = \mathbf{E}[\cdot \mid \mathbf{W}_0, \mathbf{u}]$ and $\ell(\mathbf{w}, (\mathbf{x}, y)) \stackrel{\text{def}}{=} \frac{1}{2}(\langle \mathbf{w}, \mathbf{x}\rangle - y)^2$. Let us begin with the Generalisation Gap bound, for which we use stability Bousquet and Elisseeff [2002]. Specifically, for $i \in [n]$ consider the training tuple with the ith data point resampled: $S^{(i)} \stackrel{\text{def}}{=} (z_1, \ldots, z_{i-1}, \widetilde{z}_i, z_{i+1}, \ldots, z_n)$ where $\widetilde{z}_i$ is an independent copy of z_i. With $\mathbf{W}_s^{(i)}$ denoting gradient descent with the dataset with the resampled data point $S^{(i)}$, the Generalisation Gap can be rewritten

$$\mathbf{E}[\mathcal{L}(\mathbf{W}_T) - \mathcal{L}_S(\mathbf{W}_T)] = \frac{1}{n}\sum_{i=1}^{n}\mathbf{E}[\ell(\mathbf{W}_s, \widetilde{z}_i) - \ell(\mathbf{W}_s^{(i)}, \widetilde{z}_i)]$$

The equality suggests that if the model does not vary much when a data point is resampled $\mathbf{W}_s \approx \mathbf{W}_s^{(i)}$, then the Generalisation Gap will be small. Indeed, prior work Hardt et al. [2016], Kuzborskij and Lampert [2018] has been dedicated to proving uniform bounds on L_2 norm $\max_{i\in[n]}\|\mathbf{W}_s - \mathbf{W}_s^{(i)}\|_2$. In our case, the more refined squared L_2 expected stability is considered $\frac{1}{n}\sum_{i=1}^{n}\mathbf{E}[\|\mathbf{W}_s - \mathbf{W}_s^{(i)}\|_2^2]$. To this end, we follow Lei and Ying [2020] and simply plug in the definition of the gradient iterates and apply young's inequality $(a+b)^2 \leq (1+\frac{1}{t})a^2 + (1+t)b^2$ to get for $i \in [n]$

$$\|\mathbf{W}_{t+1} - \mathbf{W}_{t+1}^{(i)}\|_2^2 \leq \frac{\eta^2(1+t)}{n^2}\left(\|\nabla\ell(\mathbf{W}_t, z_i)\|_2^2 + \|\nabla\ell(\mathbf{W}_t^{(i)}, \widetilde{z}_i)\|_2^2\right)$$
$$(1+t^{-1})\underbrace{\|\mathbf{W}_t - \mathbf{W}_t^{(i)} - \eta\left(\nabla\mathcal{L}_{S^{(i)}}(\mathbf{W}_t) - \nabla\mathcal{L}_{S^{(i)}}(\mathbf{W}_t^{(i)})\right)\|_2^2}_{\text{Expansiveness of the Gradient Update}} .$$

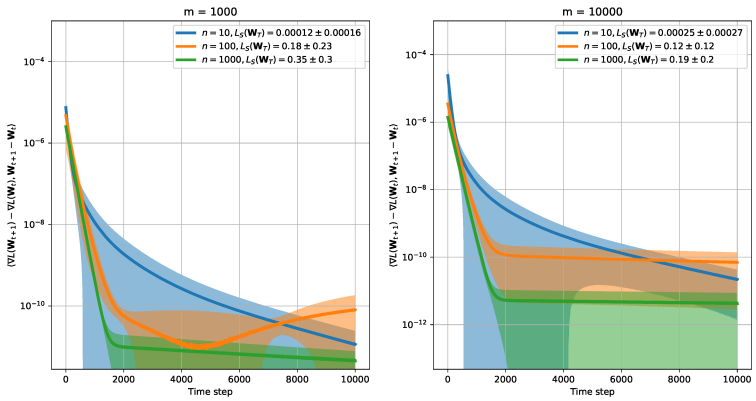

Figure 1: The gradient operator of an overparameterised shallow neural network is almost-monotone: Here, inputs are uniformly distributed on a 10-sphere and labels are generated by $\mathbf{x} \mapsto \frac{e^{\langle \mathbf{W}^\star, \mathbf{x} \rangle}}{1+e^{\langle \mathbf{W}^\star, \mathbf{x} \rangle}}$ where $\mathbf{W}^\star$ is once sampled from $\mathcal{N}(\mathbf{0}, \mathbf{I}_{10})$ at the beginning of the training. The initialisation $\mathbf{W}_0$ is sampled from $\mathcal{N}(\mathbf{0}, \mathbf{I}_{10})$ and each experiment is performed 10 times (which is reflected by standard deviation). The shallow network is then trained by GD with $\eta = 1$, $T = 10^4$, and sigmoid activation.

Following prior stability analyses of gradient descent Hardt et al. [2016], Kuzborskij and Lampert [2018], Lei and Ying [2020], Richards and Rabbat [2021], the expansiveness of the gradient update then needs to be bounded. For this we note Richards and Rabbat [2021] which observed that the negative Eigenvalues of the loss's Hessian $\nabla^2 \mathcal{L}_{S^{(i)}}(\cdot) \in \mathbb{R}^{dm \times dm}$ can control gradient expansiveness. In our case, following a more delicate analysis as the loss is not Lipschitz, we show for some $\epsilon \geq 0$

$$\min_{\|\mathbf{u}\|_2 \leq 1, \alpha \in [0,1]} \left\langle \mathbf{u}, \nabla^2 \mathcal{L}_{S^{(i)}}(\mathbf{W}(\alpha)) \mathbf{u} \right\rangle \geq -\epsilon$$

where the Hessian is evaluated along the line $\mathbf{W}(\alpha) = \mathbf{W}_t + \alpha(\mathbf{W}_t^{(i)} - \mathbf{W}_t)$ and $\epsilon \lesssim \sqrt{\frac{\eta t}{m}}$. A consequence of this is that the gradient operator $\nabla \mathcal{L}_{S^{(i)}}(\cdot)$ is then locally approximately monotone:

$$\begin{aligned}\langle \mathbf{W}_t - \mathbf{W}_t^{(i)}, \nabla \mathcal{L}_{S^{(i)}}(\mathbf{W}_t) - \nabla \mathcal{L}_{S^{(i)}}(\mathbf{W}_t^{(i)}) \rangle \geq \eta \|\nabla \mathcal{L}_{S^{(i)}}(\mathbf{W}_t) - \nabla \mathcal{L}_{S^{(i)}}(\mathbf{W}_t^{(i)})\|_2^2 \\ - \epsilon \|\mathbf{W}_t - \mathbf{W}_t^{(i)} - \eta(\nabla \mathcal{L}_{S^{(i)}}(\mathbf{W}_t) - \nabla \mathcal{L}_{S^{(i)}}(\mathbf{W}_t^{(i)}))\|_2^2.\end{aligned}$$

This result, crucial to our proof, is supported by an empirical evidence: In Fig. 1 we present a synthetic experiment where we train a shallow neural network with sigmoid activation, and demonstrate that $\nabla \mathcal{L}_S(\cdot)$ is indeed monotone under overparameterisation. Moreover, its monotonicity seems to "stabilise" as we increase the number of parameters.

Applying the monotonicity when opening the squared norm of the gradient expansiveness allows us to arrive at the recursion

$$\begin{aligned}\|\mathbf{W}_{t+1} - \mathbf{W}_{t+1}^{(i)}\|_2^2 &\leq \frac{1+t^{-1}}{1-2\eta\epsilon} \|\mathbf{W}_t - \mathbf{W}_t^{(i)}\|_2^2 + \frac{2\eta^2 t}{n^2} \left(\|\nabla \ell(\mathbf{W}_t, z_i)\|_2^2 + \|\nabla \ell(\mathbf{W}_t^{(i)}, \widetilde{z}_i)\|_2^2 \right) \\ &\leq \left(\frac{1+t^{-1}}{1-2\eta\epsilon} \right)^t \frac{2\eta^2 t}{n^2} \sum_{j=0}^{t} \left(\|\nabla \ell(\mathbf{W}_j, z_i)\|_2^2 + \|\nabla \ell(\mathbf{W}_j^{(i)}, \widetilde{z}_i)\|_2^2 \right).\end{aligned}$$

Note to control the exponential term is $\left(\frac{1}{1-2\eta\epsilon}\right)^t \lesssim \exp(\eta t \epsilon)$ we then require $\epsilon = \sqrt{\frac{\eta t}{m}} \lesssim \frac{1}{\eta t}$, which is satisfied when $m \gtrsim (\eta t)^3$. Meanwhile, from properties of the squared loss we have $\|\nabla \ell(\mathbf{W}_j, z_i)\|_2^2 \lesssim \ell(\mathbf{W}_j, z_i)$, and thus, when summing over $i \in [n]$ we recover the Empirical risk $\frac{1}{n} \sum_{i=1}^n \|\nabla \ell(\mathbf{W}_j, z_i)\|_2^2 \lesssim \mathcal{L}_S(\mathbf{W}_j)$. Finally, noting that $\mathbf{E}[\ell(\mathbf{W}_j, z_i)] = \mathbf{E}[\ell(\mathbf{W}_j^{(i)}, \widetilde{z}_i)]$ we then arrive at the following bound on the expected squared L_2 stability

$$\frac{1}{n} \sum_{i=1}^n \mathbf{E}[\|\mathbf{W}_{t+1} - \mathbf{W}_{t+1}^{(i)}\|_2^2] \lesssim \frac{\eta^2 t}{n^2} \sum_{j=0}^{t} \mathbf{E}[\mathcal{L}_S(\mathbf{W}_j)].$$

Note that the Generalisation Gap is directly controlled by the Optimisation Performance along the path of the iterates, matching the shape of bounds by Lei and Ying [2020]. Using smoothness of the loss as well as that $\mathcal{L}_S(\mathbf{W}_T) \leq \frac{1}{T}\sum_{j=0}^{T}\mathcal{L}_S(\mathbf{W}_j)$, the risk is bounded when $m \gtrsim (\eta t)^3$ as

$$\mathbf{E}[\mathcal{L}(\mathbf{W}_T)] \lesssim \left(1 + \frac{\eta T}{n}\left(1 + \frac{\eta T}{n}\right)\right)\frac{1}{T}\sum_{j=0}^{T}\mathbf{E}[\mathcal{L}_S(\mathbf{W}_j)]\,.$$

We must now bound the Optimisation Error averaged across the iterates. Following standard optimisation arguments for gradient descent on a smooth objective we then get

$$\mathcal{L}_S(\mathbf{W}_{t+1}) \leq \mathcal{L}_S(\mathbf{W}_t) - \frac{\eta}{2}\|\nabla\mathcal{L}_S(\mathbf{W}_t)\|_2^2$$

At this point convexity would usually be applied to upper bound $\mathcal{L}_S(\mathbf{W}_t) \leq \mathcal{L}_S(\hat{\mathbf{W}}) - \langle\nabla\mathcal{L}_S(\mathbf{W}_t), \hat{\mathbf{W}} - \mathbf{W}_t\rangle$ where $\hat{\mathbf{W}}$ is a minimiser of $\mathcal{L}_S(\cdot)$. The analysis differs from this approach in two respects. Firstly, we do not consider a minimiser of $\mathcal{L}_S(\cdot)$ but a general point $\mathbf{W}$, which is then optimised over at the end. Secondly, the objective is not convex, and therefore, we leverage that the Hessian negative Eigenvalues are on the order of $-\frac{1}{\sqrt{m}}$ to arrive at the upper bound

$$\mathcal{L}_S(\mathbf{W}_t) \lesssim \underbrace{\mathcal{L}_S(\mathbf{W}) - \langle\nabla\mathcal{L}_S(\mathbf{W}_t), \mathbf{W} - \mathbf{W}_t\rangle}_{\text{Convex Component}} + \underbrace{\frac{1}{\sqrt{m}}\|\mathbf{W} - \mathbf{W}_t\|_2^3}_{\text{Negative Hessian Eigenvalues}}$$

After this point, the standard optimisation analysis for gradient descent can be performed i.e. $-\langle\nabla\mathcal{L}_S(\mathbf{W}_t), \mathbf{W} - \mathbf{W}_t\rangle - \frac{\eta}{2}\|\mathcal{L}_S(\mathbf{W}_t)\|_2^2 = \frac{1}{\eta}\left(\|\mathbf{W} - \mathbf{W}_t\|_2^2 - \|\mathbf{W} - \mathbf{W}_{t+1}\|_2^2\right)$, which then yields to a telescoping sum over $t = 0, 1, \ldots, T-1$. This leads to the upper bound for $\mathbf{W} \in \mathbb{R}^{m\times d}$

$$\frac{1}{T}\sum_{j=0}^{T}\mathbf{E}[\mathcal{L}_S(\mathbf{W}_j)] \lesssim \mathcal{L}_S(\mathbf{W}) + \frac{\|\mathbf{W} - \mathbf{W}_0\|_2^2}{\eta T} + \frac{1}{\sqrt{m}}\cdot\frac{1}{T}\sum_{j=0}^{T}\|\mathbf{W} - \mathbf{W}_j\|_2^3$$

where we plug this into the generalisation bound and take a minimum over $\mathbf{W} \in \mathbb{R}^{m\times d}$.

4 Additional Related Literature

Stability of Gradient Descent. Algorithmic stability [Bousquet and Elisseeff, 2002] has been used to investigate the generalisation performance of GD in a number of works [Hardt et al., 2016, Kuzborskij and Lampert, 2018, Chen et al., 2018, Lei and Ying, 2020, Richards and Rabbat, 2021]. While near optimal bounds are achieved for convex losses, the non-convex case aligning with this work is more challenging. Within [Hardt et al., 2016, Kuzborskij and Lampert, 2018] in particular a restrictive $1/t$ step size is required if t iterations of gradient descent are performed. This was partially alleviated within [Richards and Rabbat, 2021] which demonstrated that the magnitude of Hessian's negative eigenvalues can be leveraged to allow for much larger step sizes to be taken. In this work we also utilise the magnitude of Hessian's negative Eigenvalue, although a more delicate analysis is required as both: the gradient is potentially unbounded in our case; and the Hessian's negative eigenvalues need to be bounded when evaluated across multiple points. We note the scaling of the Hessian network has also previously been observed within [Liu et al., 2020]. Stability-based generalisation bounds were also shown for certain types of non-convex functions, such Polyak-Łojasiewicz functions [Charles and Papailiopoulos, 2018, Lei and Ying, 2021] and functions with strict saddles among stationary points [Gonen and Shalev-Shwartz, 2017]. In line with our results, Rangamani et al. [2020] shows that for interpolating kernel predictors, minimizing the norm of the ERM solution minimizes stability.

Certain notions of stability, such the uniform stability (taking $\sup$ over the data rather than expectation) allows to prove **high-probability risk bounds (h.p.)** [Bousquet and Elisseeff, 2002] which are known to be optimal up to log-terms [Feldman and Vondrak, 2019, Bousquet et al., 2020]. In this work we show bounds in expectation, or equivalently, control the first moment of a generalisation gap. To prove a h.p. bounds while enjoying benefits of a stability in expectation, one would need to control higher order moments [Maurer, 2017, Abou-Moustafa and Szepesvári, 2019], however often this is done through a higher-order uniform stability: Since our proof closely relies on the stability in expectation, it is not clear whether our results can be trivially stated with high probability.

References

K. Abou-Moustafa and Cs. Szepesvári. An exponential efron-stein inequality for l_q stable learning rules. In *Algorithmic Learning Theory (ALT)*, pages 31–63. PMLR, 2019.

Z. Allen-Zhu, Y. Li, and Z. Song. A convergence theory for deep learning via over-parameterization. In *International Conference on Machine Learing (ICML)*, pages 242–252. PMLR, 2019.

M. Anthony and P. L. Bartlett. *Neural network learning: Theoretical foundations*. Cambridge University Press, 1999.

S. Arora, S. Du, W. Hu, Z. Li, and R. Wang. Fine-grained analysis of optimization and generalization for overparameterized two-layer neural networks. In *International Conference on Machine Learing (ICML)*, pages 322–332. PMLR, 2019.

Y. Bai and J. D. Lee. Beyond linearization: On quadratic and higher-order approximation of wide neural networks. In *International Conference on Learning Representations (ICLR)*, 2019.

P. L. Bartlett and S. Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3(Nov):463–482, 2002.

P. L. Bartlett, A. Montanari, and A. Rakhlin. Deep learning: a statistical viewpoint. *arXiv preprint arXiv:2103.09177*, 2021.

O. Bousquet and A. Elisseeff. Stability and generalization. *Journal of Machine Learning Research*, 2:499–526, 2002.

O. Bousquet, Y. Klochkov, and N. Zhivotovskiy. Sharper bounds for uniformly stable algorithms. In *Conference on Computational Learning Theory (COLT)*, pages 610–626. PMLR, 2020.

Z. Charles and D. Papailiopoulos. Stability and generalization of learning algorithms that converge to global optima. In *International Conference on Machine Learing (ICML)*, pages 745–754. PMLR, 2018.

Y. Chen, C. Jin, and B. Yu. Stability and convergence trade-off of iterative optimization algorithms. *arXiv preprint arXiv:1804.01619*, 2018.

S. S. Du, X. Zhai, B. Poczos, and A. Singh. Gradient descent provably optimizes over-parameterized neural networks. In *International Conference on Learning Representations (ICLR)*, 2018.

V. Feldman and J. Vondrak. High probability generalization bounds for uniformly stable algorithms with nearly optimal rate. In *Conference on Computational Learning Theory (COLT)*, pages 1270–1279. PMLR, 2019.

N. Golowich, A. Rakhlin, and O. Shamir. Size-independent sample complexity of neural networks. In *Conference on Computational Learning Theory (COLT)*, 2018.

A. Gonen and S. Shalev-Shwartz. Fast rates for empirical risk minimization of strict saddle problems. In *Conference on Computational Learning Theory (COLT)*, pages 1043–1063. PMLR, 2017.

M. Hardt, B. Recht, and Y. Singer. Train faster, generalize better: Stability of stochastic gradient descent. In *International Conference on Machine Learing (ICML)*, pages 1225–1234. PMLR, 2016.

T. Hu, W. Wang, C. Lin, and G. Cheng. Regularization matters: A nonparametric perspective on overparametrized neural network. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 829–837. PMLR, 2021.

A. Jacot, F. Gabriel, and C. Hongler. Neural tangent kernel: convergence and generalization in neural networks. In *Conference on Neural Information Processing Systems (NeurIPS)*, pages 8580–8589, 2018.

I. Kuzborskij and C. H. Lampert. Data-dependent stability of stochastic gradient descent. In *International Conference on Machine Learing (ICML)*, pages 2815–2824. PMLR, 2018.

Jaehoon Lee, Lechao Xiao, Samuel S Schoenholz, Yasaman Bahri, Roman Novak, Jascha Sohl-Dickstein, and Jeffrey Pennington. Wide neural networks of any depth evolve as linear models under gradient descent. *arXiv preprint arXiv:1902.06720*, 2019.

Y. Lei and Y. Ying. Fine-grained analysis of stability and generalization for stochastic gradient descent. In *International Conference on Machine Learing (ICML)*, pages 5809–5819. PMLR, 2020.

Y. Lei and Y. Ying. Sharper generalization bounds for learning with gradient-dominated objective functions. In *International Conference on Learning Representations (ICLR)*, 2021.

C. Liu, L. Zhu, and M. Belkin. On the linearity of large non-linear models: when and why the tangent kernel is constant. *Conference on Neural Information Processing Systems (NeurIPS)*, 33, 2020.

A. Maurer. A second-order look at stability and generalization. In *Conference on Computational Learning Theory (COLT)*, pages 1461–1475. PMLR, 2017.

Y. Nesterov. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media, 2003.

S. Oymak and M. Soltanolkotabi. Toward moderate overparameterization: Global convergence guarantees for training shallow neural networks. *IEEE Journal on Selected Areas in Information Theory*, 1(1):84–105, 2020.

A. Rangamani, L. Rosasco, and T. Poggio. For interpolating kernel machines, minimizing the norm of the ERM solution minimizes stability. 2020.

D. Richards and M. Rabbat. Learning with gradient descent and weakly convex losses. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2021.

B. Schölkopf and A. J. Smola. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. The MIT Press, 2002.

M. Seleznova and G. Kutyniok. Analyzing finite neural networks: Can we trust neural tangent kernel theory? *arXiv preprint arXiv:2012.04477*, 2020.

T. Suzuki and S. Akiyama. Benefit of deep learning with non-convex noisy gradient descent: Provable excess risk bound and superiority to kernel methods. In *International Conference on Learning Representations (ICLR)*, 2021.

Y. Yao, L. Rosasco, and A. Caponnetto. On early stopping in gradient descent learning. *Constructive Approximation*, 26(2):289–315, 2007.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] These are summarised in Section 2.4.
   (c) Did you discuss any potential negative societal impacts of your work? [N/A] We believe that presented research should be categorized as basic research and we are not targeting any specific application area. Theorems may inspire new algorithms and theoretical investigation. The algorithms presented here can be used for many different

applications and a particular use may have both positive or negative impacts. We are not aware of any immediate short term negative implications of this research and we believe that a broader impact statement is not required for this paper.

(d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]

2. If you are including theoretical results...

(a) Did you state the full set of assumptions of all theoretical results? [Yes] Assumptions are stated throughout Section 2 and in statements of theorems in Section 3.

(b) Did you include complete proofs of all theoretical results? [Yes] The main paper presents proof sketch in Section 3.2 while all the proofs with details are deferred to the supplementary material.

3. If you ran experiments...

(a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] Small synthetic experiment with instructions in Section 3.2.

(b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]

(c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]

(d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [No] We present a small synthetic experiment in Section 3.2 which can be run on any contemporary laptop.

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...

(a) If your work uses existing assets, did you cite the creators? [N/A]

(b) Did you mention the license of the assets? [N/A]

(c) Did you include any new assets either in the supplemental material or as a URL? [N/A]

(d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]

5. If you used crowdsourcing or conducted research with human subjects...

(a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]