An Empirical Verification of Wide Networks Theory

Dario Balboni[1]
dario.balboni@sns.it
Davide Bacciu[2]
bacciu@di.unipi.it

[1] Scuola Normale Superiore
Pisa, Italy
[2] Università di Pisa
Pisa, Italy

Abstract

In recent years many theories explaining the behavior of Wide Neural Networks have been proposed, focusing on relations of wide networks with Neural Tangent Kernels and on devising a novel optimization theory for overparameterized models. However, despite the efforts, real-world models are still not well-understood.

To this aim, we empirically measure crucial quantities for neural networks in the more realistic setting of mildly overparameterized models and in three main areas: conditioning of the optimization process, training speed, and generalization of the obtained models. We analyze the obtained results and highlight discrepancies between existing theories and realistic models, to guide future works on theoretical refinements.

Our contribution is exploratory in nature and aims to encourage the development of mixed theoretical-practical approaches, where experiments are quantitative and aimed at measuring fundamental quantities of the existing theories.

1 Introduction

A staggering aspect about neural networks is that they are seemingly able to overfit the training sample and yet generalize to unseen data. Similar behaviour has been actually observed in other learning paradigms in overparameterized settings, such as with linear regression [3] and kernel methods [4, 29], but neural networks are certainly the model characterized by the most surprising predictive performance on real-world data.

Several recent works [9, 16, 28, 32] have attempted to explain the theoretical underpinnings of why neural networks learn and generalize. Inspired by the relation of infinitely wide neural networks with Neural Tangent Kernel [16, 19], much of this research has focused on developing a theory for *very wide networks*, i.e. where the number of parameters $m = \Theta(n^\alpha)$ is polynomially bigger than the number of examples n [2, 10, 21, 25] with $\alpha > 1$, a condition which is not usually satisfied in real-world models[1]. While these theoretical results are being obtained under increasingly milder (and hence more realistic) overparametrization assumptions, they are typically expressed only in terms of asymptotic rates, and it is difficult to determine whether these results help in explaining the success of actually deployed models.

[1] Consider, for instance, VGG19 which has 144M parameters distributed on 19 layers and was trained on 1.3M images: already a quadratic polynomial would require an order of 1 trillion parameters per layer for VGG, which is clearly unattainable for current practice. On the other hand VGG19 is mildly overparameterized, i.e. the number of parameters per layer is slightly greater than the number of used images.

Another line of research focuses on strongly experimental work [20], which is exclusively tailored to obtain empirical suggestions for practitioners. Our approach is instead focused on measuring crucial quantities during training of realistic networks with the aim to inform theoretical research and to expose points of discrepancy.

The Problem. Neural networks are function approximators usually trained via Empirical Risk Minimization, but they cannot be understood via classical optimization theory because they are extremely non-convex [23]. In recent years many researchers have considered alternatives to convexity, concentrating on variants of the Polyak-Łojasiewicz (PL) condition [26] like PL* functions by Liu et al. [22] and the Proxy-PL condition by Frei and Gu [12].

For the PL condition many linear convergence results for first-order optimization methods are known [7, 14, 17], and the PL coefficient (akin to strong-convexity coefficient) can be lower bounded in a region nearby the network random initialization [22], thus providing convergence guarantees if the network weights remain in the vicinity of initial weights, which is the case for very wide networks. While being a very promising theory, it is currently not clear if such theorems do hold for smaller networks like those deployed in practice.

Paper Contribution. Our aim is to precisely analyze which points of the current theories fail to hold empirically when applied to mildly overparameterized networks, i.e. networks in which the number of weights grows linearly with respect to the number of examples $m \simeq cn$. This paper is, to our knowledge, the first one to perform a detailed empirical analysis of abstract theories tailored at exposing the uneffectiveness of existing theories in certain areas.

We perform quantitative measures of key local quantities related to conditioning, convergence and optimization in the PL function theory to check the impact of reducing the networks width from polynomial in the number of examples to linear. We show how such measures can characterize the training progress of real neural models on machine vision tasks providing, in Section 5, an empirical analysis that compares the prediction of the theory with the observed behavior of trained models, which enables us to spot phenomena that aren't currently fully explained.

Our work builds on theories in the area of PL functions and their applicability to neural networks; in particular we consider a general theory of PL functions convergence by Liu et al. [22], stability bounds by Charles and Papailiopoulos [6] for generalization, and the work by Agarwal et al. [1] for the analysis of conditioning.

Differences with related works. Known applications of the theory of Polyak-Łojasiewicz functions to neural networks are only concerned with the theoretical side. For example, Liu et al. [22] deals with theoretical convergence issues on wide networks and do not consider the interplay between conditioning and network depth; we focus instead on empirical convergence monitoring and generalization on mildly overparameterized networks, operating far from the kernel regime, and on realistic models. Moreover we consider quantitative issues like convergence speed of real networks which are not addressed by Liu et al. [22].

For what concerns the empirical analysis on wide neural networks, Lee et al. [20] performed a large-scale experiment to study the performance of finite-width networks compared with their infinite limits, as predicted by the NTK theory [16]. Our approach differs in that we focus on adherence of the empirical behavior to the presented theory at the level of single optimization steps, while [20] restricts to comparing the final outcomes of the optimization process for various finite- and infinite-width models, and thus mainly serves to guide empirical practitioners.

2 Convergence

We introduce the background on PL functions that we use throughout the paper, referring the reader to Karimi et al. [17] for known convergence results. We remark that the presented theory concerning convergence speeds has an analogue in the two-sided PL setting of [31], thereby extending its possible applications to minimax optimization problems, such as those present in Generative Adversarial Networks [13] optimization.

Definition 1 (PL Condition)**.** *Given a function $f : X \subseteq \mathbb{R}^m \to \mathbb{R}$ we say that f is μ-PL iff*

$$\forall x \in X \quad \frac{1}{2}\|\nabla f(x)\|^2 \geq \mu(f(x) - f^*) \tag{1}$$

where $f^* := \inf_{x \in X} f(x)$.

The Polyak-Łojasiewicz condition basically states that the norm of the gradient at a point controls the minimality gap at the current point, and thus for this class of functions necessarily $\nabla f(x) = 0$ implies that x is a global optimum in X.

PL functions enjoy the useful property of exponential convergence to a point of minimum value via common first-order otimization methods [7, 14, 17]. In this work we only consider minimization via gradient descent, but the extension to other algorithms is standard.

Lemma 1 (Convergence speed and radius for PL functions, [17])**.** *Let $f : X \subseteq \mathbb{R}^m \to \mathbb{R}$ be μ-PL and L-smooth. Choose an initial point $x_0 \in X$ and let the sequence of iterates evolve according to the rule*

$$x^{k+1} = x^k - \frac{1}{L}\nabla f(x^k). \tag{2}$$

Letting $\gamma := 1 - \frac{\mu}{L}$, the optimality gap decreases exponentially following the formula

$$f(x^{k+1}) - f^* \leq \gamma\Big(f(x^k) - f^*\Big). \tag{3}$$

Moreover, the distance from the initial point is bounded by

$$\frac{1}{L}\left\|x^{k+1} - x^0\right\| \leq \sqrt{\frac{2(f(x^0) - f^*)}{L}}\frac{1}{1 - \sqrt{\gamma}} \tag{4}$$

Additionally, PL functions theory encompasses convex optimization because strongly convex functions satisfy the Polyak-Łojasiewicz condition with the same coefficient [17]. Details of the measure of PL coefficients are reported in Appendix B.1.

3 Generalization

We expose a generalization bound for PL functions by Charles and Papailiopoulos [6] and we highlight our contribution in how the bound's quantities can be measured on real networks.

Recall that, given a labeled dataset $S = \{z_i = (x_i, y_i) \mid i = 1, \ldots, n\}$ with examples sampled i.i.d. $(x_i, y_i) \sim \mathcal{D}$ from a distribution $\mathcal{D} \in \mathcal{P}(X \times Y)$ and a learning algorithm $\mathcal{A}$, one can define $W \ni w_S := \mathcal{A}(S)$ the algorithm's output on S. Let $\ell : X \times Y \times Y \to \mathbb{R}$ be a convex loss function, and $f : X \times W \to Y$ the considered model. Then the empirical training error is:

$$R_S(w) = \frac{1}{|S|}\sum_{(x,y) \in S} \ell(x, y, f(x; w)). \tag{5}$$

In order to parallel classical convex optimization theory we rely on the cited result [6] that employs the notion of stability [11] to prove a generalization bound for PL functions.

Lemma 2 (Generalization Bound for PL risk, [6, Theorem 3])**.** *Suppose that for every randomly extracted dataset S of n elements, R_S is μ-PL, and that the empirical risk is optimized by gradient descent. Let w^*_S be a point of minimum value towards which gradient descent is converging, whose existance is guaranteed by PL-ness of the objective, i.e. assume that $\left|R_S(w_S) - R_S(w^*_S)\right| \leq \varepsilon_A$. Moreover assume that $\ell(x,y,f(x;\cdot))$ is G-Lipschitz, then for any δ with probability at least $1-\delta$ we have the estimates*

$$\beta_{ptw} \leq 2\sqrt{\varepsilon_A}\sqrt{\frac{2G^2}{\mu}} + \frac{1}{n-1}\frac{2G^2}{\mu} \tag{6}$$

$$|R_{\mathcal{D}}(w_S) - R_S(w_S)| \leq \sqrt{\frac{M^2 + 12Mn\beta_{ptw}}{2n\delta}} \tag{7}$$

where $0 \leq \ell(x,y,y') \leq M$ and $R_{\mathcal{D}}(w) := \mathbb{E}_{(x,y)\sim\mathcal{D}}[\ell(x,y,f(x;w))]$.

The exposed bound in Equation (6) gives us a way to quantitatively measure stability on real neural networks, by locally estimating their Lipschitz coefficient G and their PL constant μ as detailed in Appendix B.1, to produce Figure 4 that we will comment in Section 5.

Let us briefly comment on the role of the PL constant μ and of small-deviations of the model with respect to its parameters and its inputs as captured by the Lipschitz coefficient G and its smoothness constant L: as we can see in the definition of γ in Lemma 1 a smaller smoothness constant and a higher PL coefficient are of benefit to fast convergence.

Concerning generalization we notice similarly the importance of a large μ and a small Lipschitz coefficient in the quantity $2G^2/\mu$, which appears twice in Equation (6).

We observe moreover that the first term in Equation (6) depends on the amount of optimization performed by the algorithm, and vanishes for a perfect fitting of the training data, which suggests a connection between overfitting and generalization for PL models.

4 Conditioning

Conditioning[2] is extremely important in general optimization theory, since it provides valuable information about the speed of convergence to the objective; in the case of neural network it is related to the PL coefficient, as the following Lemma shows:

Lemma 3 (Convergence and Conditioning, [22, Theorem 4.1])**.** *Let $F : \Omega \subseteq \mathbb{R}^m \to \mathbb{R}^n$ be a function and $\mathcal{L} : \mathbb{R}^n \to \mathbb{R}$ a μ-PL and L-smooth function on $F(\Omega)$. Define $K(\theta) := \nabla F(\theta)^T \nabla F(\theta)$, which is a $n \times n$ positive semidefinite matrix and let $\lambda_* := \min_{\theta\in\Omega} \lambda_{min}(K(\theta))$, $\lambda^* = \max_{\theta\in\Omega} \lambda_{max}(K(\theta))$. Then $h := \mathcal{L} \circ F$ is $\mu\lambda_*$-PL and $L\lambda^*$-smooth on Ω since*

$$\begin{aligned} \|\nabla h(\theta)\|^2 &= \left\|\nabla \mathcal{L}(F(\theta))^T K(\theta) \nabla \mathcal{L}(F(\theta))\right\| && (8) \\ &\geq \lambda_{min}(K(\theta)) \|\nabla \mathcal{L}(F(\theta))\|^2 && (9) \\ &\geq \lambda_* \mu(\mathcal{L}(F(\theta)) - \mathcal{L}_*) = \lambda_* \mu(h(\theta) - h_*). && (10) \end{aligned}$$

where $\mathcal{L}_* := \min_{\zeta\in F(\Omega)} \mathcal{L}(\zeta) = \min_{\theta\in\Omega} \mathcal{L}(F(\theta)) = h_*$. The bound on smoothness can be proved in a similar way by majorization with $\lambda_{max}(K(\theta))$.

[2] Let us recall that the conditioning number $\kappa(M)$ of a rectangular matrix M is the ratio between its highest singular value $\sigma_{\max}(M) = \lambda_{\max}(\sqrt{M^T M})$ and its lowest singular value $\sigma_{\min}(M) = \lambda_{\min}(\sqrt{M^T M})$.

Agarwal et al. [1] provide results on conditioning at initialization and during training of wide neural networks, namely that conditioning improves exponentially with the depth of the network. Their results hold for very wide networks, where the networks' weights displace from their initialization can be bounded with $O(1/\sqrt{m})$, m being the number of weights in a single layer, and thus when the Neural Tangent Kernel well-approximates its infinite-width deterministic limit [16].

Due to the difficulty of extracting quantitative results for smaller networks from the work of Agarwal et al. [1], we present the ideas of their arguments to give the reader a bit of context on the qualitative analysis that will be performed on this matter in the experimental section.

Conditioning in Neural Networks Let $X \subseteq \mathbb{R}^d$ and let n samples $z_i = (x_i, y_i) \sim \mathcal{D} \in \mathcal{P}(X \times Y)$ be extracted from a distribution $\mathcal{D}$, and let $f^m : X \times \Theta \to \mathbb{R}^k$ be a function representing the first m layers of a neural network, which is parametric in the weights $\theta \in \Theta \subseteq \mathbb{R}^t$. Let us define the function $F^m : \Theta \subseteq \mathbb{R}^t \to \mathbb{R}^{nk}$ by $F^m(\theta) := (f^m(x_1;\theta), \dots, f^m(x_n;\theta))$.

We are then interested in studying the eigenvalues of the two different Gram matrices

$$G_{ij}^{m;\theta} := \left\langle f^m(x_i;\theta), f^m(x_j;\theta) \right\rangle, \qquad K^{m;\theta} := \nabla F^m(\theta)^T \nabla F^m(\theta), \tag{11}$$

which are related with network optimization speed [1], since $G^{m;\theta}$ is the kernel matrix when the neural network is interpreted as a feature extractor and only the last layer is trained; and $K^{m;\theta}$ is instead connect to the PL constant of the network via Lemma 3.

The main observation in the proof of Agarwal et al. [1] is the following relation between the entries of two G matrices at different layers:

$$G^{m+1;\theta} = \hat{\sigma}(G^{m;\theta}) \tag{12}$$

where $\hat{\sigma}$ is the dual activation function defined by Daniely et al. [8] and is applied entrywise.

This observation allows to control the entries in the G matrices at different layers: off-diagonal entries converge to zero when iterating, while diagonal entries remain fixed at their initial values; this observation allows to bound the highest and lowest eigenvalues, and a similar idea applies to the K matrix via its relations with the deterministic limit of the Neural Tangent Kernel. More details can be found in Appendix A.3.

Conditioning during Training Because of Lemma 3, convergence speed is ultimately connected with the minimum eigenvalue of the K matrix encountered during training, and it is thus interesting to study how conditioning evolves during the optimization trajectory.

Existing theories bound conditioning during training using the distance between weights at the current time and at the beginning of training, making use of the well-known fact that for invertible linear operators A and B such that $\sigma_{\min}(A)\|A-B\|_{\text{op}} < 1$ one has (see [24]):

$$|\sigma_{\min}(A) - \sigma_{\min}(B)| \leq \|A-B\|_{\text{op}} \leq \|A-B\|_F, \tag{13}$$

where $\|\cdot\|_{\text{op}}$ is the operator norm and $\|\cdot\|_F$ is the Frobenius matrix norm.

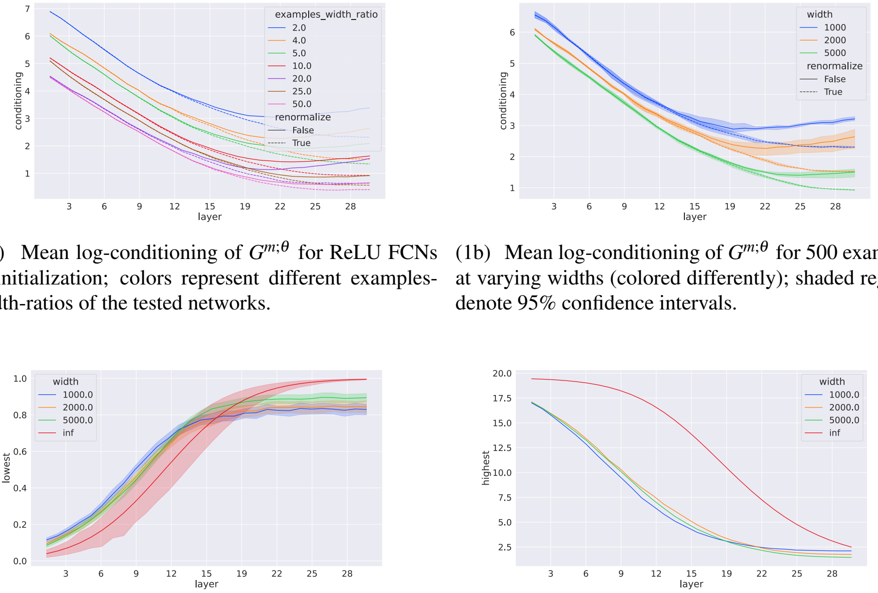
Equation (13) can then be used with $A = G^{m;\theta(t_0)}$ and $B = G^{m;\theta(0)}$ to obtain the required bound, as the right hand side can be expanded in terms of distance between weights [22]. Such an approach is essentially pessimistic, as it assumes that conditioning is optimal at the start and that it degrades during training.

5 Empirical Analysis

In this Section we describe and analyze the performed experiments to check how well real-world models can be described by the exposed theories.

Experimental Setting. We train several mildly overparameterized FCNs on random subsets of CIFAR10 [18][3]. The networks are initialized with Gaussian Kaiming initialization [15] to preserve the variance of activations in the forward pass; activation functions are normalized according to Agarwal et al. [1], i.e. having zero mean and unitary variance on standardized normal inputs. Input data is normalized to satisfy a scaled requirement of the unitary norm required by Agarwal et al. [1] such that $\|x_i\| = \sqrt{m}$, where m is the width of the first layer.

Conditioning at Initialization. We consider FCN networks consisting of 30 layers of varying widths (1000, 2000, 5000), different activation functions (ReLU and Tanh), over multiple numbers of randomly extracted examples (100, 200, 500, 1000), either renormalizing[4] after application of each layer or not, and averaging on three random seeds. We measure conditioning of $G^{m;\theta}$ at initialization over all layers. The experiment has been run on a Tesla V100 PCIe 16GB GPU.



(1a) Mean log-conditioning of $G^{m;\theta}$ for ReLU FCNs at initialization; colors represent different examples-width-ratios of the tested networks.

(1b) Mean log-conditioning of $G^{m;\theta}$ for 500 examples at varying widths (colored differently); shaded regions denote 95% confidence intervals.

(1c) Normalized lowest eigenvalue for ReLU networks of different widths with renormalization. Infinite width network line has been obtained using Lemma 12.

(1d) Normalized highest eigenvalue for ReLU networks of different widths with renormalization. To make the graph more clear only the mean is reported.

[3] Usage of a single dataset to validate the theory has been dictated by the heavy computational requirements of the experiments. Such a choice should nonetheless not impact the validity of the presented results, because the tested theories do not contain free parameters to be fitted on the dataset, and experiments are repeated for multiple random extractions of a subset of the data, thus leaving less chance for an overfit of the results to the dataset.

[4] i.e. rescaling each datapoint such that its norm is the square root of the number of neurons in the layer.

Figure (1a) shows that conditioning at initialization effectively decreases exponentially even for finite-width networks; it is smaller for higher width-examples ratio, and continues to decrease at each layer when the examples are renormalized; on the other hand, if the examples are not renormalized, we can find a layer after which conditioning starts to increase again, probably due to small-width effects in the sampling of gaussian weights that deviate from the distribution assumptions of Agarwal et al. [1][5]. Moreover, larger widths allow to reach a smaller conditioning for the same sample size (Figure (1b)) and also allow for a latter departure between the normalizing and non-normalizing behavior.

In Figure (1c) we can see that the lowest eigenvalue (with a rescaling by square root of width computed accordingly to Appendix B.2) effectively increases as it passes through different layers, also at finite widths. However it is also clear that it produces a marked increase in the first layers and it stops at farther layers, with higher values for wider networks. Analogously we see an inverse trend for normalized highest eigenvalue in Figure (1d). These observations highlight the need for a more thoughtful conditioning theory for finite-width networks so that useful hints can be obtained for an optimal tradeoff between computational resources and convergence speed with respect to network width and normalization layers.

Training Speed and Conditioning. In the second experiment we consider networks of 6 and 9 FCN layers, of width 500, with different activation functions (ReLU, Tanh), and train them using full-batch gradient descent (without momentum) with mean-squared-error loss and cross-entropy loss[6] over 80 epochs, with various learning rates (0.0005, 0.001, 0.005, 0.01) and different number of randomly sampled examples (50, 100, 250, 500) over three seeds. To have meaningful results on networks that actually learn, the analysis has been conducted only on those configurations that have reached accuracy $> 12\%$ at the end of training. The experiment has been run on an A100 SXM4 40GB GPU.

We find that the upper bound on loss decrease given by Lemma 1 matches well the actual loss decreases at later epochs, while in initial epochs the estimate is too conservative (Figure (2a)).

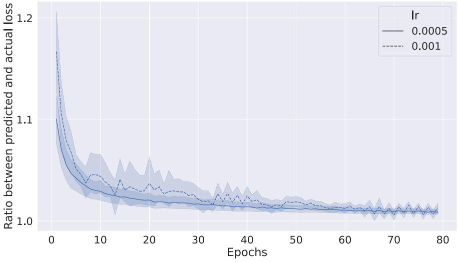
Figure (2c) and Figure (2d) show how for networks trained with cross-entropy loss, conditioning worsens very slowly during training, while for higher learning rates it initially increases very rapidly, but then decreases steadily, ending up lower than the start. The reader can see how pessimistic are current estimates of eigenvalues based on Equation (13) during training in Figure (2b), in which we can see how the bound degrades significantly, and even becomes vacuous for an high number of examples. We emphasize the importance for theoretical research to look at possible explanations for this behaviour, which could greatly simplify the study of neural networks optimization, as the main theoretical difficulties in providing convergence guarantees lie in the possibility that the lowest eigenvalue may tend to zero during training.

Practical Conditioning Proxy. Given our findings about conditioning at initialization (Figure (1a)) and the importance of conditioning for training speed, in principle we would like to avoid training a network that is too deep for its width, which has the drawback of raising

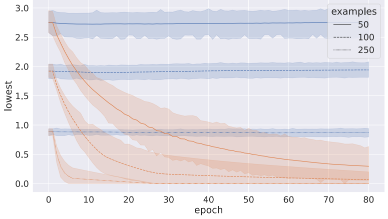
[5] According to the original paper renormalization shouldn't have any effect, but they effectively consider infinite-width networks implicitly in the definition of the dual activation function (see Lemma 11).

[6] We include cross-entropy since it is the most common loss for classification networks, even though it is not a PL function. To make it tractable we report the theory of weakly-PL functions in Appendix A.2.

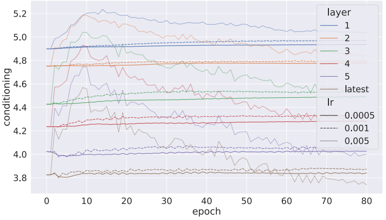
[8] The "loss prediction ratio" refers to the ratio of the actual training loss and the bound calculated according to Lemma 1 where the PL and smoothness coefficients are empirically calculated on the network at the previous epoch. Local PL coefficients calculations are detailed in Appendix B.1.



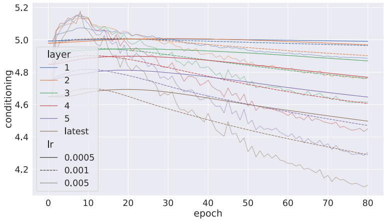
(2a) Ratio between predicted and measured loss decrease[8] at single epochs for 6-layers ReLU networks trained under MSE at different learning rates.



(2b) Progression of the lowest eigenvalues of $G^{m;\theta}$ for ReLU networks trained with MSE is in blue; lower bounds obtained using Equation (13) are in orange.



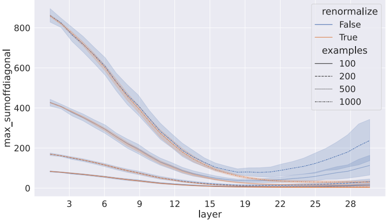
(2c) Progression of log-conditioning for ReLU networks trained under Categorical Cross-Entropy.



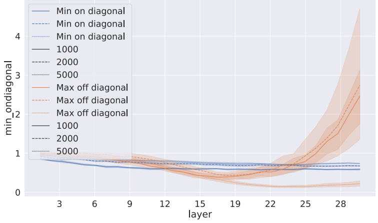
(2d) Progression of log-conditioning for Tanh networks trained under Categorical Cross-Entropy.

the conditioning from some point onward. Measuring conditioning directly is inefficient, as it requires to solve the eigenvalue problem for very big square matrices, whose size is the product of the networks width and the number of examples, making it impractical.

As we have already observed in Section 4, we expect off-diagonal entries of $G^{m;\theta}$ to go to zero as the signal propagates throught the layers, hence we propose to measure this proxy information instead of measuring eigenvalues directly, as this can be used to bound the highest and lowest eigenvalues via Gershgorin Circle Theorems, greatly simplifying practical approximated measurements of conditioning[9].



(3a) Maximum row-sum of the off-diagonal entries of the matrix $G^{m;\theta}$.

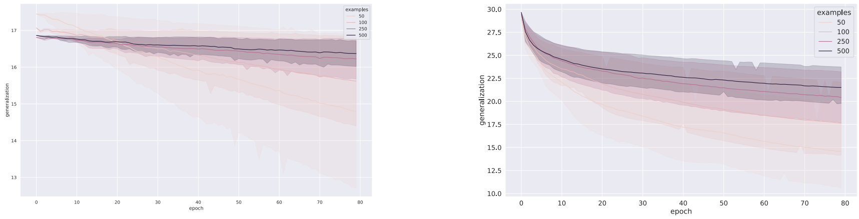


(3b) Minimum value of the on-diagonal entries and the maximum value of the off-diagonal entries of $G^{m;\theta}$.

[9] A cheaper measure of network conditioning at a certain point in its training trajectory may be useful to stop training when the expected perfomance gains in terms of loss decrease are not justifyied by their expected costs. Moreover it could be used in the architecture search paradigm to filter out networks that have worse conditioning at initialization, as it is indicative of the loss optimization possibilities for that architecture.

In Figure (3a) and Figure (3b) the difference in behaviour between the normalized layers and the non-normalized ones is extremely evident, and their divergence point aligns perfectly with the raise in conditioning that we have already observed in Figure (1a), making these measures a good proxy even for smaller networks than those observed by Agarwal et al. [1].

Generalization. By measuring the local PL coefficients and estimates of the network Lipschitz coefficients, we are able to use Lemma 2 and compute a generalization bound for networks trained on MSE loss. Figure 4 shows the expected marked decrease with increasing optimization. Despite this, the obtained bounds are vacuous[10].



(4a) Generalization bounds for ReLU networks trained under MSE on logits.

(4b) Generalization bounds for ReLU networks trained under MSE.

Figure 4: Generalization bounds obtained using the estimates in Lemma 2

6 Conclusions

The performed experiments on realistically-sides models have exposed some discrepancies with the available theories. We expose the most important ones in the authors opinion, and briefly comment on their possible consequences.

- The different behavior of conditioning with respect to renormalized and non-renormalized networks (Figure (1a)), contrary to the observations of Agarwal et al. [1] which only seem to hold on very wide networks, as hinted by the role that width plays in delaying the difference between the two settings (Figure (1b)).

  A better understanding of this phenomenon could lead to find optimal network widths in the trade-off between computational requirements and better loss convergence, as well as the possibility to remove normalization layers on early layers of the networks, which doesn't seem to bring conditioning benefits according to Figure (1a).

- The unexpected finding that conditioning worsens very slowly (and can possibly improve) during training, as current theories assume pessimistically that it worsens the farther the networks parameters are from their initialization (Figures (2b) to (2d)).

[10] To the authors opinion, this outcome could be due both to the lack of sharpness on generalization estimates based on stability, as well as by the small number of examples used in each test (more examples per test could give non-vacuous estimates, even though these would require a lot more memory than what is currently available on typical GPUs) or by the large number of weights of the FCN architecture, which impact a lot on measures that depend on Lipschitz constants. We thus find that the results on generalization are inconclusive, and that more experiments are necessary to understand the reasons behind the vacuousness of the bounds.

If this finding were confirmed by other experiments and theoretical reasons, existing theories would simplify considerably, as convergence guarantees all rely on conditioning not worsening too much during training.

- The fact that the available generalization bounds give vacuous estimates (Figure 4)[10]. Stricter generalization bounds could give guarantees to all the settings in which neural models are deployed in critical tasks, such as in automotive or in clinical screening.

This work is, to our knowledge, the first that tries to explicitly quantify abstract theories about the inner working of neural networks, and to compare the bounds obtained with real-world experiments. We hope that such an approach can provide useful feedback to theoretical researchers by pointing out aspects that are not yet completely explained.

Limitations. There are, in general, very few testable theories expressed in quantitative terms, which necessarily makes it more difficult to test theories on deployed models. We think that this is an interesting research field in which to invest, and which is still relatively unexplored. Benefits of such an approach include a faster feedback loop between theory and practice, and a stronger focus on real-world models, which, as we have highlighted in the current exposition, do not behave like very wide networks in regards to multiple aspects.

Future Work. Let us conclude by delineating possible research directions for future works.

Conditioning. Future works should consider extending the theory on conditioning to finite networks, thereby including an estimation of the variance due to randomness in finite-width initialization, a reconsideration of the role of normalization (which seems more useful in finite-width settings), and focus on alternative ways to bound conditioning during training, where it does not worsen as much as previously thought. Moreover, we think that the search for better conditioning can provide a principled justification to weight initialization strategies that make networks train better, thus connecting this work with a line of theoretical investigations by Schoenholz et al., Xiao et al. [28, 30].

Convergence. For what concerns convergence, the current theory has good predictability at later epochs, while in the initial epochs the networks perform much better than the theoretical predictions. This phenomenon has to be investigated in more depth, as there could be significant gains from having a better model of how networks behave in the first training epochs, e.g. to discard models that exhibit worse PL constants during an architecture search.

Generalization. In contrast with the partial results obtained concerning convergence and conditioning, the tested generalization theory gives vacuous bounds. We highlight the need to assess the cause of this behavior and to develop alternative approaches.

Alternatives to the Polyak-Łojasiewicz condition. We highlight how future theories should also consider variations of the PL condition like Proxy-PLness [12], and the weak-PL condition [7], since multiple useful loss functions only satisfy the latter.

Extensions to other architectures. Extensions of the presented theories to CNN, ResNets and Transformers are needed to form a more complete picture of network behavior. Those can be investigated within the same framework, but require different theoretical calculations in Lemma 9, Lemma 11 and an adaptation of the results of Agarwal et al. [1] for each architecture, and are thus left to future works.

References

[1] Naman Agarwal, Pranjal Awasthi, and Satyen Kale. A deep conditioning treatment of neural networks. In *Algorithmic Learning Theory*, pages 249–305. PMLR, 2021.

[2] Zeyuan Allen-Zhu, Yuanzhi Li, and Zhao Song. A convergence theory for deep learning via over-parameterization. In *International Conference on Machine Learning*, pages 242–252. PMLR, 2019.

[3] Peter L Bartlett, Philip M Long, Gábor Lugosi, and Alexander Tsigler. Benign overfitting in linear regression. *Proceedings of the National Academy of Sciences*, 117(48): 30063–30070, 2020.

[4] Mikhail Belkin, Siyuan Ma, and Soumik Mandal. To understand deep learning we need to understand kernel learning. In *International Conference on Machine Learning*, pages 541–549. PMLR, 2018.

[5] Rajendra Bhatia. *Matrix analysis*, volume 169. Springer Science & Business Media, 2013.

[6] Zachary Charles and Dimitris Papailiopoulos. Stability and generalization of learning algorithms that converge to global optima. In *International Conference on Machine Learning*, pages 745–754, 2018.

[7] Dominik Csiba and Peter Richtárik. Global convergence of arbitrary-block gradient methods for generalized polyak-łojasiewicz functions. *arXiv preprint arXiv:1709.03014*, 2017.

[8] Amit Daniely, Roy Frostig, and Yoram Singer. Toward deeper understanding of neural networks: The power of initialization and a dual view on expressivity. *arXiv preprint arXiv:1602.05897*, 2016.

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

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

[11] Andre Elisseeff, Theodoros Evgeniou, Massimiliano Pontil, and Leslie Pack Kaelbing. Stability of randomized learning algorithms. *Journal of Machine Learning Research*, 6(1), 2005.

[12] Spencer Frei and Quanquan Gu. Proxy convexity: A unified framework for the analysis of neural networks trained by gradient descent. *arXiv preprint arXiv:2106.13792*, 2021.

[13] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27:2672–2680, 2014.

[14] Charles Guille-Escuret, Manuela Girotti, Baptiste Goujaud, and Ioannis Mitliagkas. A study of condition numbers for first-order optimization. In *International Conference on Artificial Intelligence and Statistics*, pages 1261–1269. PMLR, 2021.

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

[16] Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in neural information processing systems*, pages 8571–8580, 2018.

[17] Hamed Karimi, Julie Nutini, and Mark Schmidt. Linear convergence of gradient and proximal-gradient methods under the polyak-łojasiewicz condition. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 795–811, 2016.

[18] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[19] 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.

[20] Jaehoon Lee, Samuel S Schoenholz, Jeffrey Pennington, Ben Adlam, Lechao Xiao, Roman Novak, and Jascha Sohl-Dickstein. Finite versus infinite neural networks: an empirical study. *arXiv preprint arXiv:2007.15801*, 2020.

[21] Dawei Li, Tian Ding, and Ruoyu Sun. On the benefit of width for neural networks: Disappearance of bad basins. *arXiv*, pages arXiv–1812, 2018.

[22] Chaoyue Liu, Libin Zhu, and Mikhail Belkin. Toward a theory of optimization for over-parameterized systems of non-linear equations: the lessons of deep learning. *arXiv preprint arXiv:2003.00307*, 2020.

[23] Chaoyue Liu, Libin Zhu, and Mikhail Belkin. Loss landscapes and optimization in over-parameterized non-linear systems and neural networks. *Applied and Computational Harmonic Analysis*, 59:85–116, 2022.

[24] Haifeng Ma. Construction of some generalized inverses of operators between banach spaces and their selections, perturbations and applications. 2012.

[25] Samet Oymak and Mahdi 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.

[26] Boris Teodorovich Polyak. Gradient methods for minimizing functionals. *Zhurnal Vychislitel'noi Matematiki i Matematicheskoi Fiziki*, 3(4):643–653, 1963.

[27] Mark Rudelson and Roman Vershynin. Smallest singular value of a random rectangular matrix. *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences*, 62(12):1707–1739, 2009.

[28] Samuel S Schoenholz, Justin Gilmer, Surya Ganguli, and Jascha Sohl-Dickstein. Deep information propagation. 2017.

[29] Alexander Tsigler and Peter L Bartlett. Benign overfitting in ridge regression. *arXiv preprint arXiv:2009.14286*, 2020.

[30] Lechao Xiao, Yasaman Bahri, Jascha Sohl-Dickstein, Samuel Schoenholz, and Jeffrey Pennington. Dynamical isometry and a mean field theory of cnns: How to train 10,000-layer vanilla convolutional neural networks. In *International Conference on Machine Learning*, pages 5393–5402. PMLR, 2018.

[31] Junchi Yang, Negar Kiyavash, and Niao He. Global convergence and variance-reduced optimization for a class of nonconvex-nonconcave minimax problems. *arXiv preprint arXiv:2002.09621*, 2020.

[32] Zitong Yang, Yaodong Yu, Chong You, Jacob Steinhardt, and Yi Ma. Rethinking bias-variance trade-off for generalization of neural networks. In *International Conference on Machine Learning*, pages 10767–10777. PMLR, 2020.

A Additional Theory and Useful Lemmas

In this Appendix we report useful lemmas that complement the main text about composition of lipschitz and smoothness constants, theory of minimization of weakly PL functions, and the ideas about the theory of conditioning.

A.1 Lemmas on Lipschitz and Smoothness constant

In what follows we define the smoothness and lipschitz constant of a function $f: X \to Y$ by:

$$\left\|f(x) - f(x')\right\| \leq G_f \left\|x - x'\right\| \tag{14}$$

$$\left\|\nabla f(x) - \nabla f(x')\right\| \leq L_f \left\|x - x'\right\| \tag{15}$$

Lemma 4 (Smoothness Constant and Composition)**.** *Let $f: X \to Y$ and $g: Y \to Z$, and define $h(x) = g(f(x))$. Then we have $G_h \leq G_g G_f$ and $L_h \leq G_f^2 L_g + G_g L_f$.*

Proof. For the smoothness constant we have:

$$\begin{aligned} \|\nabla h_x - \nabla h_{x'}\| &= \left\|\nabla g_{f(x)} \nabla f_x - \nabla g_{f(x')} \nabla f_x + \nabla g_{f(x')} \nabla f_x - \nabla g_{f(x')} \nabla f_{x'}\right\| && (16) \\ &\leq \|\nabla f_x\| \left\|\nabla g_{f(x)} - \nabla g_{f(x')}\right\| + \left\|\nabla g_{f(x')}\right\| \|\nabla f_x - \nabla f_{x'}\| && (17) \\ &\leq \left(G_f^2 L_g + G_g L_f\right) \left\|x - x'\right\|. && (18) \end{aligned}$$

The result is on the other hand well-known for the Lipschitz constant. □

Lemma 5. *The mean-squared-error loss has $G \leq 2MSE(x)$, $L \leq N$, where N is the number of dimensions. Moreover it is a 1-strongly convex function.*

Proof. Recall that for the mean-squared-error loss we have:

$$\ell(x,y) = \frac{1}{2} \sum_i (x_i - y_i)^2 \tag{19}$$

$$\frac{\partial \ell}{\partial x_n} = x_n - y_n \tag{20}$$

$$\frac{\partial^2 \ell}{\partial x_n \partial x_m} = \delta_{nm} \tag{21}$$

and thus we obtain that $\sum_n \left|\frac{\partial \ell}{\partial x_n}\right|^2 = 2\ell(x,y)$, $\sum_{n,m} \left|\frac{\partial^2 \ell}{\partial x_n \partial x_m}\right|^2 = N$, and it is a 1-strongly convex function because the Hessian is always positive definite and $H \geq I$. □

Lemma 6. *The cross-entropy loss has $L, G \leq 2$ and is a weakly convex function.*

Proof. Recall that cross-entropy satifyies

$$\ell(x,k) = -x_k + \log\left(\sum_j e^{x_j}\right) \tag{22}$$

$$\frac{\partial \ell}{\partial x_n}(x,k) = -\delta_{n,k} + \frac{e^{x_n}}{\sum_j e^{x_j}} \tag{23}$$

$$\frac{\partial^2 \ell}{\partial x_m \partial x_n}(x,k) = \frac{\delta_{mn} e^{x_n} \left(\sum_j e^{x_j}\right) - e^{x_m} e^{x_n}}{\left(\sum_j e^{x_j}\right)^2} \tag{24}$$

and one easily obtains that $\sum_n \left|\frac{\partial \ell}{\partial x_n}(x,k)\right|^2 \leq 2$, $\sum_{n,m} \left|\frac{\partial^2 \ell}{\partial x_m \partial x_n}(x,k)\right|^2 \leq 2$. Weak convexity stems from the positivity of the hessian matrix: let us call $t_i = \frac{e^{x_i}}{\sum_j e^{x_j}}$, then we have seen that $H_{nm} = \frac{\partial^2 \ell}{\partial x_m \partial x_n}(x,k) = \delta_{mn} t_n - t_n t_m$, and thus we have

$$u^T H u = \sum_{ij} H_{ij} u_i u_j = \sum_i t_i u_i^2 - \sum_{ij} t_i t_j u_i u_j = \sum_i t_i u_i^2 - \left(\sum_j t_j u_j\right)^2 \geq 0 \tag{25}$$

by Cauchy-Schwartz inequality applied to $\sqrt{t_i}$ and $\sqrt{t_i} u_i$, since $\sum_i t_i = 1$. □

Definition 2 (Matrix Norm)**.** *Let A be a square matrix and consider the norm induced by the p-vector norm:*

$$\|A\|_{p\to p} := \sup_{v \in V} \frac{\|Av\|_p}{\|v\|_p}. \tag{26}$$

Definition 3 (Spectral Radius)**.** *Let A be an $n \times n$ square matrix with eigenvalues $\lambda_1, \ldots, \lambda_n \in \mathbb{C}$. We define its spectral radius $\rho(A)$ by*

$$\rho(A) := \max\left\{ ,_{i=1}^{n} |\lambda_i| \right\} = \max\left\{ |\lambda_1|, \ldots, |\lambda_n| \right\} \tag{27}$$

as the maximum absolute value of its eigenvalues.

Lemma 7 (Matrix Norm Facts)**.** *In what follows let $p, q \in \overline{\mathbb{R}}$ be an Hölder conjugate pair, i.e. $\frac{1}{p} + \frac{1}{q} = 1$. Then the matrix norm satisfies the following:*

1. $\|A\|_{p\to p} = \|A^T\|_{q\to q}$
2. $\|AB\|_{p\to p} \leq \|A\|_{p\to p} \|B\|_{p\to p}$
3. $\rho(A) \leq \|A\|_{p\to p}$
4. $\|A\|_{2\to 2}^2 = \rho(A^T A)$
5. $\|A\|_{\infty\to\infty} = \max_i \left(\sum_j |A_{ij}|\right)$

Proof. Omitted because they are known facts. For a proof see Bhatia [5]. □

Lemma 8 (Interpolation for matrix norms)**.** *Let $p, q \in \overline{\mathbb{R}}$ be an Hölder conjugated pair, i.e. $\frac{1}{p} + \frac{1}{q} = 1$. Then it holds:*

$$\|A\|_{2\to 2}^2 \leq \|A\|_{p\to p} \|A\|_{q\to q}. \tag{28}$$

Proof. Using the facts enumerated in Lemma 7 we get

$$\|A\|_{2\to 2}^2 = \rho(A^T A) \leq \|A^T A\|_{p\to p} \leq \|A^T\|_{p\to p} \|A\|_{p\to p} = \|A\|_{q\to q} \|A\|_{p\to p}. \tag{29}$$

In particular we notice that $\|A\|_{2\to 2}^2 \leq \|A\|_{1\to 1} \|A\|_{\infty\to\infty}$. □

Lemma 9 (Lipschitz and Smoothness constant for a layer)**.** *Let us consider a FCN layer $f(x) = M\phi(x)$ where the activation function ϕ is applied entrywise. Suppose that it receives inputs whose coordinates are always inside the interval I; then its Lipschits and Smoothness constants are:*

$$G_f = \sigma_{max}(M) \cdot \max_{t \in I} \left|\phi'(t)\right| \tag{30}$$

$$L_f = \max_{ij} \left|M_{ij}\right| \cdot \max_{t \in I} \left|\phi''(t)\right|. \tag{31}$$

Proof. It is easy to calculate the layer Lipschitz constant with respect to euclidean norm since it is a composition of a linear layer M and of the non-linear function ϕ, and we obtain the desired result.

The calculation of the Smoothness constant is more involved, and we start by writing out fully the layer-defining equations:

$$f(x)_i = \sum_j M_{ij}\phi(x_j) \tag{32}$$

$$\frac{\partial f(x)_i}{\partial x_j} = M_{ij}\phi'(x_j) \tag{33}$$

$$\frac{\partial^2 f(x)_i}{\partial x_j \, \partial x_j} = \delta_{jk} M_{ij}\phi''(x_j) \tag{34}$$

At this point we can make use of Equation (28) to obtain the desired result

$$\left\|\nabla^2 f(x)_i\right\|_{2\to 2} = \sqrt{\|\nabla^2 f(x)_i\|_{1\to 1} \|\nabla^2 f(x)_i\|_{\infty\to\infty}} \tag{35}$$

$$= \left\|\nabla^2 f(x)_i\right\| 1 \to 1 \tag{36}$$

$$= \max_j \left|M_{ij}\right| \cdot \max_{t \in I} \left|\phi''(t)\right| \tag{37}$$

where $\left\|\nabla^2 f(x)_i\right\|_{1\to 1} = \left\|\nabla^2 f(x)_i\right\|_{\infty\to\infty}$ because $\nabla^2 f(x)_i$ is symmetric. □

The above lemmas provide a way to compute the worst expected eigenvalue when the network weights are trained, using Equation (13), since we can now use Lemma 9 to compute the Lipschitz and Smoothness constants of the layer and Lemma 4 iteratively to bound those of the whole network.

Being based on iterative estimates, it is obvious that the obtained quantities could be off from the real Lipschitz and Smoothness constant by quite a bit, and thus also the obtained eigenvalue and generalization estimates obtained. We leave the question of how to improve those bounds to future works.

A.2 Weakly PL Functions

In the main text we have exposed the theory about strongly PL functions for ease of exposition; in this Section we instead present the theory of weak PL functions which originated in the work of Csiba and Richtárik [7], and which is needed to analyze the results of the experiments on the cross-entropy loss. We expose the main results about weak PL functions, and refer the interested reader to the cited work.

Definition 4 (Weakly PL function, [7])**.** *A function $f : X \to \mathbb{R}$ is weakly μ-PL on $\Omega \subseteq X$ if there $\exists x^* \in X^*$ the set of global minimizer such that*

$$\forall x, y \in \Omega \quad \|\nabla f(x)\| \|x - x^*\| \geq \sqrt{\mu}(f(x) - f^*) \tag{38}$$

where $f^* := \min_{x \in \Omega} f(x)$.

It can be easily observed that every weakly convex function is weakly PL with $\mu = 1$. Moreover we have the following lemma on the decrease of the objective during the optimization process:

Lemma 10 (Gradient Descent on Weakly PL functions, [7, Lemma 3])**.** *Given an L-smooth, weakly μ-PL function $f : X \to \mathbb{R}$ and chosen an initial point x_0, we let the sequence of iterates evolve according to the rule:*

$$x_{k+1} = x_k - \frac{1}{L} \nabla f(x_k). \tag{39}$$

Then the optimality gap decreases following the formula

$$f(x_{k+1}) - f^* \leq \left(1 - \frac{\mu(f(x_k) - f^*)}{2L\|x_k - x_*\|^2}\right)(f(x_k) - f^*). \tag{40}$$

Specificities about how the quantities in Equation (40) have been measured are discussed in Appendix B.1.

A.3 Conditioning Ideas

In this Section we detail the ideas in the work of Agarwal et al. [1] that allow us to obtain bounds on the conditioning of the various network layers.

In what follows, we will assume that the inputs $(x_i)_{i=1,\dots,n}$ to our neural network are all of unitary norm and their pairwise product satisfies $\left|x_i \cdot x_j\right| \leq 1 - \delta$. We being by introducing the definition of the dual activation function, due to Daniely et al. [8].

Definition 5 (Dual Activation, [8])**.** *Given an activation function σ, we define the dual activation function $\hat{\sigma} : [-1, 1] \to \mathbb{R}$ by*

$$\hat{\sigma}(\rho) = \mathbb{E}_{(X,Y) \sim \Sigma_\rho}[\sigma(X)\sigma(Y)] \qquad \textit{where } \Sigma_\rho = \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}. \tag{41}$$

Lemma 11 (Dual Activation and Infinite-Width Networks, [8])**.** *$\hat{\sigma}$ corresponds to the function that is calculated by freshly initialized networks layers of infinite-width on a given pair of inputs. In other words, define by*

$$\tilde{K}^m_{ij} := \lim_{t \to \infty} K^{m;\theta}_{ij} \tag{42}$$

the NTK deterministic limit, where t is the dimension of the space Θ. Then we have that

$$\tilde{K}^{m+1} = \hat{\sigma}\left(\tilde{K}^m\right) \tag{43}$$

where $\hat{\sigma}$ is applied entrywise.

Lemma 11 gives us an idea about what to expect when depth is increased in a wide neural network. Infact, when the space of parameters is so large that $K^{m;\theta}_{ij} \simeq \tilde{K}^m_{ij}$, Lemma 11 gives us a way to know the entries of the K matrix and thus to bound its conditioning.

Eigenvalue Bounding Sketch We begin by exposing a way to give bounds on the smallest eigenvalue for iterative applications of the dual activation to a given positive definite matrix:

Lemma 12 (Eigenvalue lower bound lemma [1, Lemma 23])**.** *Let $H \in \mathbb{R}^{n\times n}$ a positive-definite matrix, i.e. $H \geq \delta I_n$ and all values $= 1$ on the diagonal. Let $f : \mathbb{R} \to \mathbb{R}$ be an analytic function whose series expansion in zero has only positive coefficients, and let $f[H]$ be the application of f to each entry of the matrix. Then $f[H] \geq (f(1) - f(1-\delta))I_n$.*

By our assumptions, $K^{0;\theta}$ is positive definite with ones on the diagonal, and we can verify that for all commonly considered activation functions, $\hat{\sigma}$ is analytic on the real line with non-negative coefficients in its power series expansion, so that Lemma 12 above applies.

On the other hand, we can bound the highest eigenvalue using Gershgorin circle theorems if we know how to bound the off-diagonal entries of the K matrix. To this aim we introduce the definition of distortion, that essentially quantifies how much $\hat{\sigma}$ acts as a contraction when its inputs are a bit distant from one.

Definition 6 (Maximal Distortion of Dual Activation, [1])**.** *Given a dual activation function $\hat{\sigma}$, we define its distortion at level δ as:*

$$M_\delta := \sup\{\hat{\sigma}(\rho) \mid |\rho| \leq 1-\delta\}. \tag{44}$$

As the reader can imagine, iteratively applying $\hat{\sigma}$ to the set of pairwise products of inputs $D_{ij} = \langle x_i, x_j \rangle$ leaves the diagonal terms unchanged since $\hat{\sigma}(1) = 1$, while it increasingly shrinks the off-diagonal terms to zero, in an amount that is quantified by its distortion.

It is then clear that any matrix chain such that $S^{m+1} = \hat{\sigma}(S^m)$ essentially converges toward the identity matrix. Agarwal et al. [1] are able to extract the convergence rates from such reasonments and thereby establish the stronger result of exponential convergence of both the $G^{m;\theta}$ and $K^{m;\theta}$ matrices to conditioning of one. For more details we refer the reader to the already cited works [1, 8].

B Additional Experimental Measures and Results

In this Section we detail the measurements procedures and the results of the experiments.

B.1 Measure of PL Coefficients

The empirical measure of the global PL coefficient (μ in Equation (1)) of a function is hard because one has to guarantee that

$$\forall x \in X \quad \mu \leq \frac{\|\nabla f(x)\|^2}{2(f(x)-f^*)}, \tag{45}$$

which essentially requires to know the function f globally.

What we did was instead to measure the local PL coefficients $\mu : X \to \mathbb{R}$ defined by

$$\mu(x) := \frac{\|\nabla f(x)\|^2}{2(f(x)-f^*)}, \tag{46}$$

where $f^* = 0$ in our case because we know we are in the setting of overparameterized models. This may at first seem like a rough simplification which could lead to vastly incorrect results; we thus motivate more in depth this choice.

Indeed, under appropriate smoothness of f, the function $\mu : X \to \mathbb{R}$ is itself smooth, and thus it is at least locally stable. To quantify how much stable this quantity is in a local neighbourhood of a point, one can leverage the fact that, in our case, the function to minimize is a sum of independent terms, i.e.

$$\mu(x) = \frac{\mathbb{E}_{z\sim\mathcal{D}}\left[\|\nabla_x g(z;x)\|^2\right]}{2\mathbb{E}_{z\sim\mathcal{D}}[g(z;x)]} \tag{47}$$

where $f(x) = \sum_{i=1,\dots,n} g(z_i;x)$, which effectively smoothes out the PL coefficient function.

Moreover, it is not hard to devise a convergence theorem akin to Lemma 1 using local coefficients $\mu(x)$, which effectively shows their usefulness in predicting the behavior of the optimization process.

Lemma 13. *Consider a function $f : X \to \mathbb{R}$ which is L-smooth. Then the gradient method with a step-size of $1/L$*

$$x_{k+1} = x_k - \frac{1}{L}\nabla f(x_k) \tag{48}$$

satifies the equation

$$f(x_k) - f^* \leq (f(x_0) - f^*)\prod_{i=0}^{k-1}\left(1 - \frac{\mu(x_i)}{L}\right) \tag{49}$$

which gives a global convergence rate.

Proof. By using the update rule in the L-smoothness condition we get

$$f(x_{k+1}) - f(x_k) \leq -\frac{1}{2L}\|\nabla f(x_k)\|^2. \tag{50}$$

Now by using the local PL inequality we get

$$f(x_{k+1}) - f(x_k) \leq -\frac{\mu(x_k)}{L}(f(x_k) - f^*) \tag{51}$$

Rearranging and subtracting f^* from both sides gives $f(x_{k+1}) - f^* \leq \left(1 - \frac{\mu(x_k)}{L}\right)(f(x_k) - f^*)$, which can be applied recursively to obtain the stated result. □

B.2 Scaling of the eigenvalues

We are interested in understanding the change in conditioning due to varying the width of the network and the number of examples used to train it. To this aim we compare the measured eigenvalues to the ones predicted by random matrix theory. In particular if we treat the matrix $F^m(\theta)$ as a random Gaussian matrix, using the result of Rudelson and Vershynin [27] we would expect that $\lambda_{\max} \sim \sqrt{\text{width}\cdot\text{examples}}$ and $\lambda_{\min} \sim \sqrt{\text{width}} - \sqrt{\text{examples}}$ and thus

$$\kappa(G^{m;\theta}) = \frac{\lambda_{\max}}{\lambda_{\min}} \sim \frac{1}{\sqrt{\frac{1}{\text{examples}}} - \sqrt{\frac{1}{\text{width}}}} \tag{52}$$

and this is exactly what we can observe in Figure 5, which shows that such relation holds at every layer.

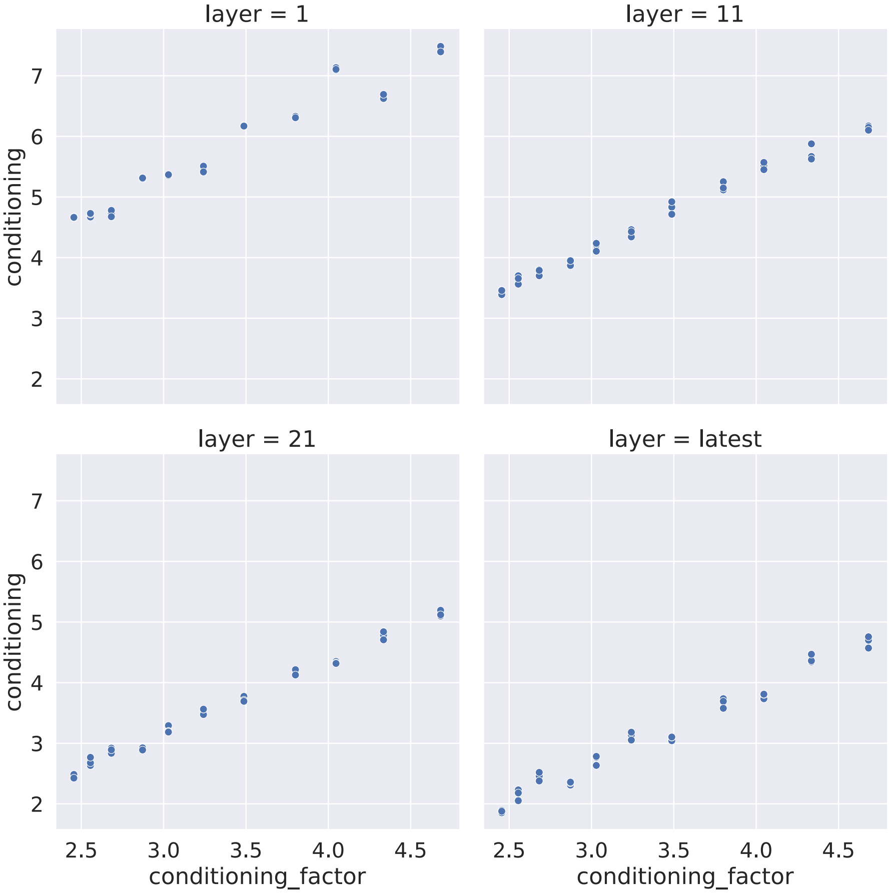
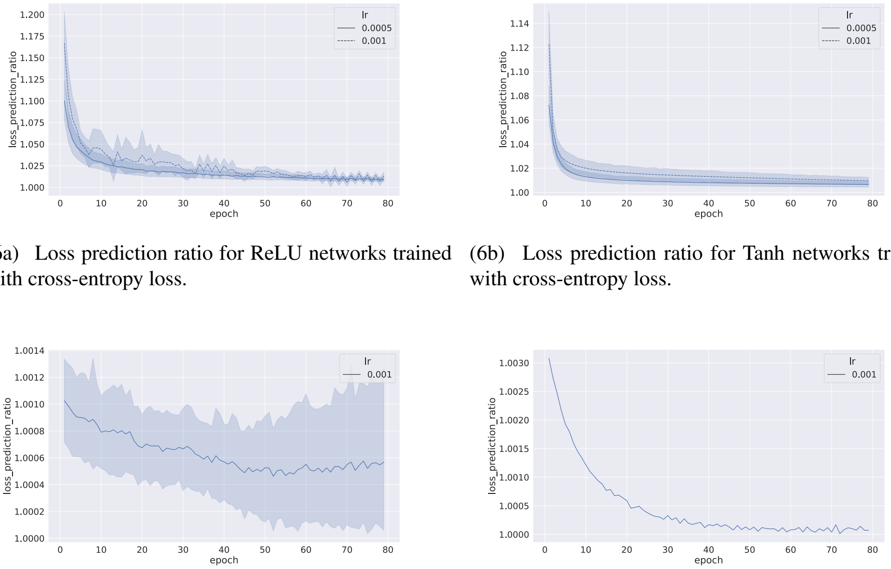


Figure 5: Plot of logarithmic conditioning for non-renormalized Tanh networks with respect to the factor $\frac{\sqrt{\text{examples}\cdot\text{width}}}{\sqrt{\text{width}}-\sqrt{\text{examples}}}$.

B.3 Prediction of training loss with cross-entropy

Using Lemma 10 for the cross-entropy loss and estimating $\|x_k - x^*\| \simeq 2\frac{f(x_k)-f^*}{G}$ where G is the Lipschitz constant of the function f, we obtain the results showed in Figure (6a) and Figure (6b).

We can observe how the prediction accuracy are similar to what we have obtained for the mean-squared-error loss: our estimates are too conservative for the first epochs, but align well with successive epochs. Further analyses are needed to perfectly align theoretical predictions with experimental ones.



(6a) Loss prediction ratio for ReLU networks trained with cross-entropy loss.

(6b) Loss prediction ratio for Tanh networks trained with cross-entropy loss.

(6c) Loss prediction ratio for ReLU networks trained with MSE.

(6d) Loss prediction ratio for Tanh networks trained with MSE.

Figure 6: Ratio between single-epoch loss prediction and actual loss.

B.4 Generalization

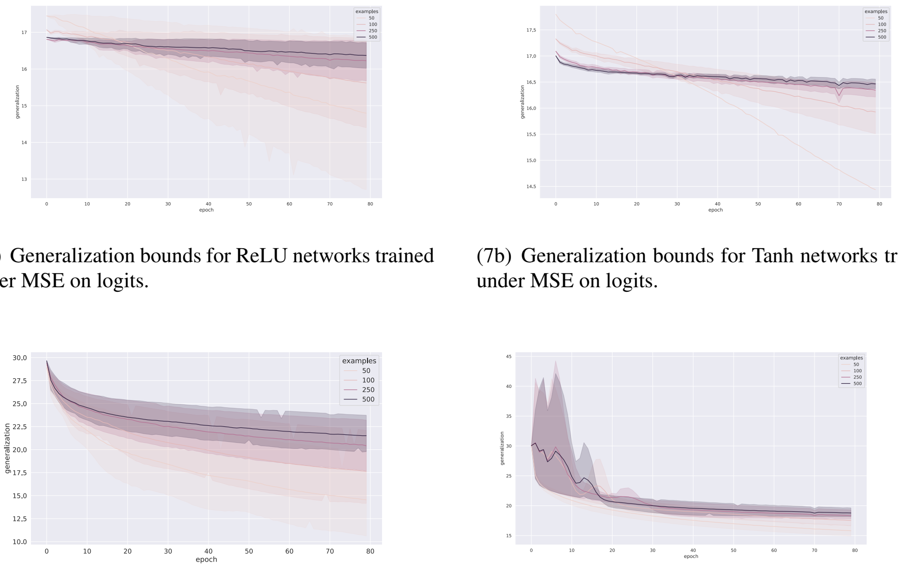
The estimates arising from Lemma 2 are vacuous[12] as can be seen in Figure 7. Further investigation on this issue is left to future work.

B.5 Eigenvalues during Training

We report in this section all the graphs about the behavior of the minimum measured eigenvalues of the $G^{m;\theta}$ matrix during training. As it can be seen from the Figures, they don't decrease as much as predicted, effectively remaining constant in the case of MSE loss and also slightly improving over time for cross-entropy loss.

We also report in Figure 9 multiple graphs about the lowest predicted eigenvalues according to Equation (13) to show how much those bounds are pessimistic. We can notice that in multiple cases the bounds even become vacuous after the first epochs of training.

[12] The loss has a value between zero and one, while the predicted bounds are well over twenty.



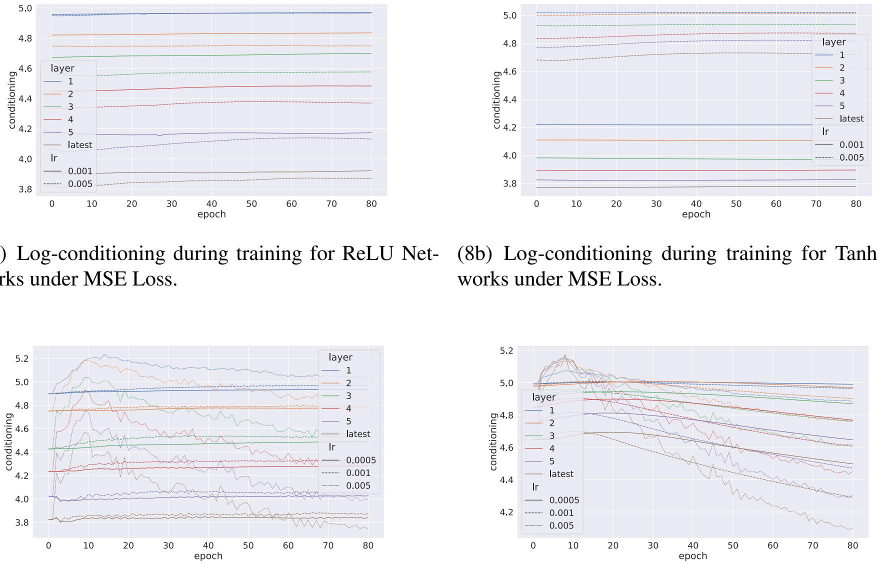
(7a) Generalization bounds for ReLU networks trained under MSE on logits.

(7b) Generalization bounds for Tanh networks trained under MSE on logits.

(7c) Generalization bounds for ReLU networks trained under MSE.

(7d) Generalization bounds for Tanh networks trained under MSE.

Figure 7: Generalization bounds obtained using the estimates in Lemma 2



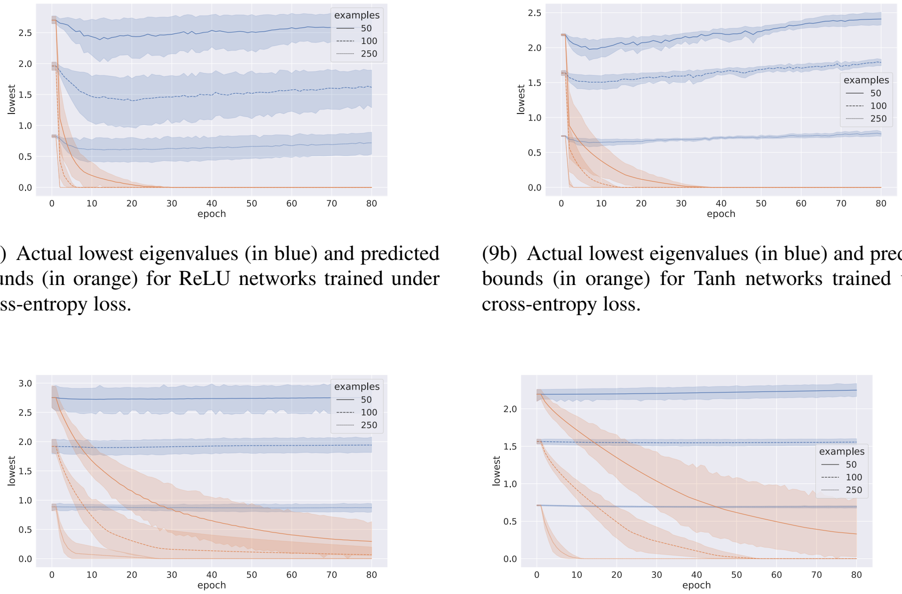
(8a) Log-conditioning during training for ReLU Networks under MSE Loss.

(8b) Log-conditioning during training for Tanh Networks under MSE Loss.

(8c) Log-conditioning during training for ReLU Networks under cross-entropy Loss.

(8d) Log-conditioning during training for Tanh Networks under cross-entropy Loss.

Figure 8: Behavior of log-conditioning of $G^{m;\theta}$ during training.



(9a) Actual lowest eigenvalues (in blue) and predicted bounds (in orange) for ReLU networks trained under cross-entropy loss.

(9b) Actual lowest eigenvalues (in blue) and predicted bounds (in orange) for Tanh networks trained under cross-entropy loss.

(9c) Actual lowest eigenvalues (in blue) and predicted bounds (in orange) for ReLU networks trained under MSE loss.

(9d) Actual lowest eigenvalues (in blue) and predicted bounds (in orange) for Tanh networks trained under MSE loss.

Figure 9: Lowest eigenvalues and predicted bounds during training.