Algorithmic Stability of Minimum-Norm Interpolating Deep Neural Networks

Anonymous Author(s)
Affiliation
Address
email

Abstract

Algorithmic stability is a classical framework for analyzing the generalization error of learning algorithms. It predicts that an algorithm is likely to have a small generalization error if it is insensitive to small perturbations in the training set such as the removal or replacement of a training point. While stability has been demonstrated for numerous well-known algorithms, this framework has had limited success in analyses of neural networks. In this paper we study the algorithmic stability of deep ReLU neural networks that achieve zero training error using parameters with the smallest L_2 norm, also known as the minimum-norm interpolation, a phenomenon that can be observed in overparameterized models trained by gradient-based algorithms. We find that such networks are stable when they contain a (possibly small) stable sub-network, followed by a layer with a low-rank weight matrix. The low-rank assumption is inspired by recent empirical and theoretical results which demonstrate that training deep neural networks is biased towards low-rank weight matrices, for minimum-norm interpolation and weight-decay regularization. Furthermore, we present a series of experiments supporting our finding that a trained deep neural network often consists of a stable sub-network and several final low-rank layers.

1 Introduction

The stochastic gradient descent (SGD) family of algorithms has emerged as the go-to tool for training machine learning models on a vast amount of data. While the generalization ability of such algorithms is reasonably well-understood when learning involves solving convex or quasi-convex optimization problems, the picture is much less clear for non-convex learning problems involving overparameterized models, such as that of training a deep neural network. Yet, the empirical evidence strongly suggests that the generalization ability of SGD algorithms remains high despite overparameterization in non-convex settings. These observations appear to be in conflict with the classical learning-theoretic *complexity-fit tradeoff* viewpoint [Bousquet et al., 2004, Zhang et al., 2021].

In recent years, there has been growing interest in the hypothesis that the favorable generalization performance of optimization algorithms can be accounted for by the algorithm's *implicit bias* or inherent model capacity control. A prominent idea in this setting is that of *minimum-complexity interpolation*, which states that the algorithm finds the simplest solution among those achieving zero (or near-zero) training error. While this phenomenon is well established in the context of linear regression (e.g. a pseudo-inverse solution to the least-squares problem recovers parameters with a minimum L_2 norm), the suggestion that it also applies to the training of a deep overparameterized ReLU neural networks is only rather recent. In particular, it has been shown theoretically that the gradient flow (GF) algorithm (gradient descent can be seen as a discretization of GF) asymptotically

Submitted to the Mathematics of Modern Machine Learning Workshop at NeurIPS 2024. Do not distribute.

fits an interpolating ReLU neural network with the minimum L_2 norm of the parameters [Lyu and Li, 2019, Ji and Telgarsky, 2020, Phuong and Lampert, 2020]. GF is thought to be a good proxy for gradient descent algorithms because the approximation error introduced by discretization remains controlled under certain regularity assumptions [Elkabetz and Cohen, 2021]. Recent works have attempted to reconcile minimum-norm interpolation with classical generalization theories based on the uniform convergence principle such as Rademacher complexity [Bartlett and Mendelson, 2002, Bartlett et al., 2017]. Indeed, dimension-free bounds have provided some explanation as to why we do not observe overfitting in the absence of label noise [Ji and Telgarsky, 2019, Telgarsky, 2022], and observe only *benign overfitting* otherwise [Tsigler et al., 2020, Koehler et al., 2021, Frei et al., 2023].

In this paper, we explore the generalization capabilities of minimum-norm interpolating neural networks from the alternative viewpoint of *algorithmic stability*. Stable learning algorithms are insensitive to small perturbations (e.g. removal or replacement of data points) of the training set and generalize well under mild assumptions [Bousquet and Elisseeff, 2002, Shalev-Shwartz et al., 2010]. Moreover, stability analysis provides valuable insights beyond generalization, such as controlling the variance of the algorithm, which is crucial for uncertainty quantification methods like bootstrapping [Elisseeff et al., 2005]. Many algorithms have been shown to be stable, including nonparametric predictors (e.g. nearest-neighbors) [Devroye and Wagner, 1979], minimizers of strongly convex problems (such as the ridge regression estimator) [Bousquet and Elisseeff, 2002], GD-type algorithms minimizing convex and smooth objectives [Hardt et al., 2016, Lei and Ying, 2020], as well as quasi-convex objectives [Charles and Papailiopoulos, 2018, Richards and Kuzborskij, 2021].

Despite these advances, success has so far been limited in the context of neural networks. Although several works have analyzed the stability of SGD in the non-convex setting, their results come with significant limitations. First, vacuous bounds are typically obtained when the number of training steps (or time) far exceeds the sample size, i.e. $t \gg n$ [Hardt et al., 2016, Kuzborskij and Lampert, 2018, Richards and Rabbat, 2021, Wang et al., 2023]; at the same time, for a large enough model capacity, we expect interpolation to happen when $t \gg n$. Second, strong assumptions such as Lipschitzness of the loss function *in the parameters* or penalization of the objective [Hardt et al., 2016, Kuzborskij and Lampert, 2018, Farghly and Rebeschini, 2021] are often required. Third, these works assume *parameter stability*, meaning that parameters are expected to remain close (typically in the Euclidean distance) if the training set is perturbed slightly; this is often unrealistic in the context of neural networks since the same predictor can be expressed using very different parameters thanks to symmetries in the weight matrices and because of non-convexity of the objective function. Finally, these works focus on optimization aspects, which seldom reveal insight into the structural properties of neural networks obtained by stable algorithms.

Our contributions In this paper, we hypothesize that stability of deep nonlinear networks originates in the early layers, and is preserved throughout the subsequent layers. Specifically, we study the algorithmic stability of minimum-norm interpolation with deep ReLU neural networks, and identify sufficient conditions for stability. In particular, such interpolations of neural networks are stable if they contain a contiguous, *stable sub-network* (for instance, the few first layers), and this sub-network is followed by at least one *low-rank weight matrix*. See Fig. 1 for an illustration.

For the data-generating process, we assume the existence of a neural network with the same architecture that can interpolate the data using bounded weights, although not necessarily with the minimal norm. This implies a setting where data cannot be arbitrarily complex ensuring that weights of such a network remain bounded as the training set grows.

Our analysis is inspired by recent theoretical and empirical findings suggesting that deep interpolating neural networks exhibit a low-rank weight matrix structure [Frei et al., 2022, Timor et al., 2023, Galanti et al., 2023]. Thus, the idea behind the proof is to ensure that if a stable sub-network exists, the remaining low-rank weight matrices will preserve its stability. Our proof separates the concepts of low-rank bias and sub-network stability into distinct phenomena. This separation enables isolated analysis, simplifying the study of stability in neural networks to the potentially easier task of understanding why stable sub-networks exist. Furthermore, we conduct a series of experiments in which we observe that stable prefix sub-networks indeed occur in practice.

In our analysis we also take a step towards addressing some limitations in the literature discussed earlier. Algorithmically, interpolating neural networks studied here can be obtained as solutions of a GF and so, unlike previous results, our findings hold for $t \gg n$ and potentially $t \to \infty$. In

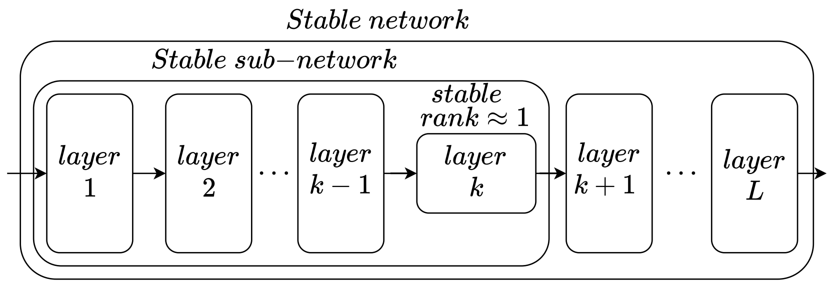

Figure 1: **A diagram of our main result.** Our main result relies on three assumptions: a) the data is expressible by a network with finite norm (Assumption 1), b) the minimum-norm interpolating ReLU neural network contains at least one layer with a low stable rank matrix (**??**) and c) the sub-network is stable (Hypothesis 1). Under these three assumptions, we show in Theorem 1 that the full network is also stable.

contrast to existing works, the analysis does not require regularity assumptions such as Lipschitzness or smoothness in the parameters.

2 Preliminaries

Neural networks. We consider fully-connected neural networks with ReLU activations, L layers and uniform width d for hidden layers, no bias, and a real-valued output. In this setting, network $N : \mathbb{R}^{d_0} \to \mathbb{R}$ is defined by weight matrices $\{\mathbf{W}_\ell\}_{1 \le \ell \le L}$, one matrix per layer, with $\mathbf{W}_\ell \in \mathbb{R}^{d_\ell \times d_{\ell-1}}$, $d_\ell = d$ for each $1 \le \ell \le L-1$, and $d_L = 1$. Network N computes its output on an input $\mathbf{x} \in \mathbb{R}^{d_0}$ by computing a *pre-activation* vector $\mathbf{h}_\ell$ and a *post-activation* vector $\mathbf{y}_\ell$ for each layer ℓ starting from $\mathbf{y}_0 = \mathbf{x}$ as follows, where the activation function ϕ_ℓ is $\mathrm{ReLU}(x) = \max\{x, 0\}$ applied to vectors element-wise for every layer $1 \le \ell \le L-1$ and ϕ_L is the identity function for layer L:

$$\mathbf{h}_\ell = \mathbf{W}_\ell \cdot \mathbf{y}_{\ell-1} \qquad \mathbf{y}_\ell = \phi_\ell(\mathbf{h}_\ell). \tag{1}$$

We do not explicitly consider the bias term, however it can be modelled by appending 1 to $\mathbf{y}_{\ell-1}$.

The network's output $N(\mathbf{x})$ is given by $y_L \in \mathbb{R}$ and the weights $\mathbf{W}_L$ are referred to as the *readout weights*. The collection of all parameters is denoted as θ, and we note $N(\cdot) = N(\,\cdot\,;\,\theta)$ to emphasize the dependency in the parameters. Network N is an instance of a neural architecture $\mathbb{A} = \langle L, d, d_0 \rangle$ determined by the number of layers L, the width d of the hidden layers and the input dimension d_0.

In the following we use notation $N^{1:k-1} : \mathbb{R}^{d_0} \to \mathbb{R}^{d_{k-1}}$ to denote a neural network with parameters $\{\mathbf{W}_\ell\}_{1 \le \ell \le k-1}$ obtained from N by removing all layers with $\ell > k-1$.

A useful property of ReLU is *positive homogeneity*: $\mathrm{ReLU}(\alpha x) = \alpha\, \mathrm{ReLU}(x)$ for each $x \in \mathbb{R}$ and $\alpha \ge 0$. In particular, we have $N(\alpha \mathbf{x};\, \theta) = \alpha\, N(\mathbf{x};\, \theta)$ and $N(\mathbf{x};\, \alpha\theta) = \alpha^L\, N(\mathbf{x};\, \theta)$.

Stable rank. The Frobenius norm of a matrix $\mathbf{A} \in \mathbb{R}^{p \times q}$ with singular values s_j for $1 \le j \le \min(p, q)$ is given by $\|\mathbf{A}\|_F = (\sum_j s_j^2)^{\frac{1}{2}}$. The spectral norm of $\mathbf{A}$, that is $\|\mathbf{A}\|_2$ is given by the largest absolute value of its singular values. The stable rank of matrix $\mathbf{A}$ is defined as $\mathrm{S}(\mathbf{A}) = \|\mathbf{A}\|_F / \|\mathbf{A}\|_2$. In particular, matrix $\mathbf{A}$ has stable rank 1 if and only if its rank (defined in the usual way) is also one [Tropp, 2015]; in this case, its singular value with largest absolute value is also the only non-zero singular value. We will sometimes refer to the Frobenius norm of a neural network N as that of the matrix obtained by concatenating all weight matrices of N.

Training set. A training set $(\mathbf{X}, \mathbf{y})$ consists of n examples $(\mathbf{x}_i, y_i)$ sampled i.i.d. from an unknown distribution p on $\mathcal{X} \times \{-1, +1\}$ where the input space $\mathcal{X}$ is an Euclidean ball of radius one. Furthermore, we denote by $(\mathbf{X}^{(j)}, \mathbf{y}^{(j)})$ the training dataset obtained from $(\mathbf{X}, \mathbf{y})$ by re-sampling j-th example according to p independently and we note $(\mathbf{x}^{(j)}, y^{(j)})$ the new sample. Finally, we say that N interpolates $(\mathbf{X}, \mathbf{y})$ if $N(\mathbf{x}_i) = y_i$ for each $i \in [n]$.

Minimum-norm interpolating neural network. Given a neural architecture $\mathbb{A} = \langle L, d, d_0 \rangle$ and a training set $(\mathbf{X}, \mathbf{y})$, we assume that we have access to an algorithm $\mathcal{T}$ which returns the parameters of a minimum-norm interpolating neural network instantiating $\mathbb{A}$. We then denote

$$\mathcal{T}(\mathbf{X}, \mathbf{y}) = \arg\min_{\theta} \left\{ \|\theta\|^2 \ : \ \forall i \in [n] \quad N(\mathbf{x}_i;\, \theta) = y_i \right\} .$$

Finally, we denote the parameters obtained by algorithm $\mathcal{T}$ as $\hat{\theta} = \mathcal{T}(\mathbf{X}, \mathbf{y})$.

Training to interpolation and minimum-norm solutions are common assumptions which represent idealized views of gradient-based training algorithms (in particular with weight decay) [Timor et al., 2023, Galanti et al., 2023]. For example, in the limit of infinite training time, gradient flow on ReLU networks converges to a minimum-norm interpolant [Lyu and Li, 2019, Ji and Telgarsky, 2020].

Algorithmic stability. Algorithm $\mathcal{A}$ is β-uniformly ϵ-stable [Kutin and Niyogi, 2002] with respect to a data distribution p if, for each training set $(\mathbf{X}, \mathbf{y})$ sampled from p, the following holds for $i \in [n]$, where $\hat{\theta} = \mathcal{A}(\mathbf{X}, \mathbf{y})$ and $\hat{\theta}^{(i)} = \mathcal{A}(\mathbf{X}^{(i)}, \mathbf{y}^{(i)})$:

$$\mathbb{P}\left(\left| N(\mathbf{x};\, \hat{\theta}) - N(\mathbf{x};\, \hat{\theta}^{(i)}) \right| \leq \epsilon \right) \geq 1 - \beta$$

where $\mathbb{P}()$ is taken with respect to the jointly distributed $(\mathbf{X}, \mathbf{y}, \mathbf{x}^{(i)}, y^{(i)}, \mathbf{x}, y)$.

This notion of stability is weaker than the well-known notion of *ϵ-uniform stability* [Bousquet and Elisseeff, 2002]: $\sup_{\mathbf{X},\mathbf{y},\mathbf{x},y,i} |N(\mathbf{x};\, \hat{\theta}) - N(\mathbf{x};\, \hat{\theta}^{(i)})| \leq \epsilon$, however both coincide for $\beta = 0$ almost surely. Often, we will say that the algorithm is stable when for a fixed β, $\epsilon = \mathcal{O}_{n\to\infty}(n^{-\alpha})$ for some $\alpha > 0$. We will occasionally abuse terminology and speak of a *stable* minimum-norm interpolating neural network $N(\,\cdot\,;\, \hat{\theta})$, meaning that algorithm $\mathcal{T}$ generating $\hat{\theta}$ is stable.

Stability implies generalization. Let $f : \mathbb{R}^2 \to [0, M]$ be a fixed known loss function. Then the *risk* and the *empirical risk* of the predictor parameterized by θ are respectively defined as $R(\theta) = \int f(N(\mathbf{x};\, \theta), y)\, \mathrm{d}p(\mathbf{x}, y)$ and $\hat{R}(\theta) = \frac{1}{n}\sum_{i=1}^{n} f(N(\mathbf{x}_i;\, \theta), y_i)$. It is known that if $\hat{\theta}$ is generated by an ϵ-uniformly-stable algorithm $\mathcal{A}$, there exist universal constants $c_1, c_2 > 0$ such that for any $\delta \in (0, 1)$ [Feldman and Vondrak, 2018, Bousquet et al., 2020],

$$\mathbb{P}\left(|R(\hat{\theta}) - \hat{R}(\hat{\theta})| \leq c_1 \ln(n) \ln(1/\delta)\, \epsilon + c_2\, M \sqrt{\frac{\ln(1/\delta)}{n}} \right) \geq 1 - \delta\, . \tag{2}$$

Hence, the gap between the population loss and empirical loss is controlled with high probability as long as the algorithm is stable.

3 From sub-network stability to prediction stability

In this section, we present our main result and discuss its implications. Our results depend on one technical assumption and one main hypothesis, which we introduce next. The first assumption concerns the complexity of the learning problem [Timor et al., 2023]:

Assumption 1 (B-admissible training set)**.** *Given a finite $B > 0$, we call a training set B-admissible if there exists a neural network N of architecture $\langle L^*, d, d_0 \rangle$ for some $L^* \geq 2$ with parameters θ^* such that $N(\mathbf{x}_i;\, \theta^*) = y_i$ for $i \in [n]$ and its weight matrices satisfy $\max_k \|\mathbf{W}_k^*\|_F \leq B$.*

Under this assumption, the training set can be viewed as generated by a hypothetical teacher network whose weight matrices have bounded norms. Note that Assumption 1 is only meaningful in learning scenarios where data cannot be arbitrarily complex (otherwise, one can construct instances such that $B \to \infty$ as $n \to \infty$).

We next define our notion of sub-network.

Definition 1 (Sub-network)**.** *Given $2 \leq k \leq L - 1$, consider the following decomposition of weight matrix $\mathbf{W}_k \in \mathbb{R}^{d \times d}$: $\mathbf{W}_k = \lambda_k \mathbf{u}_k {\mathbf{v}_k}^\top + \mathbf{W}_k'$ where $\lambda_k > 0$ is the leading eigenvalue of $\mathbf{W}_k$, $\mathbf{u}_k, \mathbf{v}_k$ are unitary vectors, $\mathbf{u}_k$ is the leading eigenvector of $\mathbf{W}_k$ and $\mathbf{v}_k$ is the leading eigenvector of $\mathbf{W}_k^T$. Then, a **sub-network** at position k is defined as*

$$f_k(\mathbf{x};\, \theta) := \mathbf{v}_k^\top N^{1:k-1}(\mathbf{x};\, \theta)\, . \tag{3}$$

Invoking the recent results from Timor et al. [2023], it is easy to see that under Assumption 1, a trained deep network contains at least one layer with low stable rank.

Lemma 1. *Suppose that datasets are B-admissible according to Assumption 1. Given $a > 0$ and $\epsilon = M/n^{-\alpha}$, for some $M \geq 0, \alpha > 0$, there exists $L > L^*$ and $1 \leq k \leq L-1$ such that with parameters $\hat{\theta}$ generated by algorithm $\mathcal{T}$ for architecture $\mathbb{A} = \langle L, \mathrm{d}, \mathrm{d}_0 \rangle$, the following holds:*

$$\mathrm{S}(\hat{\mathbf{W}}_k) \leq 1 + a\,\epsilon\,. \tag{4}$$

Finally, we state our main hypothesis, namely the existence of a stable sub-network.

Hypothesis 1 (β-uniformly ϵ-stable sub-network)**.** *Given $\beta \in [0,1]$ and $\epsilon = M/n^{-\alpha}$, for some $M \geq 0, \alpha > 0$, and $L > L^*$, there exists $2 \leq k \leq L-1$ such that with parameters $\hat{\theta}$ and $\hat{\theta}^{(i)}$ generated by algorithm $\mathcal{T}$ for architecture $\mathbb{A} = \langle L, \mathrm{d}, \mathrm{d}_0 \rangle$, the following holds:*

$$\mathbb{P}\left(\left| f_k(\mathbf{x};\, \hat{\theta}) - f_k(\mathbf{x};\, \hat{\theta}^{(i)}) \right| \leq \epsilon \right) \geq 1 - \beta\,.$$

In support of this hypothesis, we present empirical evidence in Figures 2 and 3: trained FCNs contain sub-networks whose stability is on par with that of the full network.

Now we present our main result (a complete statement is given in Theorem 1):

Main result *(sketch). Let $a > 0$ and $\epsilon = n^{-\alpha}$ for some $\alpha > 0$ and suppose that there exist (B, k, L, β) such that Assumption 1, and Hypothesis 1 are satisfied. Then, assuming that sample size satisfies $n = \Omega(\max(a\,B^L, B^{L-k+1}))$, there is a universal constant $C > 0$ such that $\mathcal{T}$ is β-uniformly*

$$C\left(1 + B^{2L-k+1} + a\,B^{3L-k+1}\right)\epsilon \text{ — stable}\,.$$

The main implication of the above result is that the stability of the entire minimum-norm interpolating neural network is controlled by the stability of its sub-network. This fact is non-trivial: since $\hat{\theta}$ and $\hat{\theta}^{(i)}$ are different (thus in particular the deeper layers $\hat{\mathbf{W}}_{k+1}, \ldots, \hat{\mathbf{W}}_L$ and $\hat{\mathbf{W}}_{k+1}^{(i)}, \ldots, \hat{\mathbf{W}}_L^{(i)}$), there is no *a priori* reason to believe that they would preserve any stability that originates in early layers. The proof requires the key observation that the stable sub-network is followed by a layer with a weight matrix of a low stable rank, which will preserve the signal as it propapagates into deeper layers. While this assumption might initially seem strong, we observe it to hold in practice (Fig. 2). Furthermore, recently Timor et al. [2023] showed that the stable rank decays roughly at the rate $B^{\frac{L}{k}}$.[1] In our case, the stable rank decay rate is assumed to be $a\,\epsilon$ with a free parameter $a \geq 0$.

Note that while the overall stability scales linearly with the stability of the sub-network, the bound is attenuated by a B-dependent factor. One extreme (pessimistic) case is $a \gg 0$, that is when weight matrices are sufficiently far from rank-one. Then, the factor in the worst case becomes of order B^{3L}. Intuitively, this occurs because, learning all perturbation matrices $(\mathbf{W}_k^{\epsilon})_k$ (see **??**) becomes unavoidable. In another extreme case of rank-one matrices and a shallow stable sub-network ($a = 0, k = 2$), we have an overall stability bound of order $B^{2L}\epsilon$.

In a more optimistic scenario, a is sufficiently small such that weight matrices have a stable-rank close to one. In this case we have a dominant factor B^{2L-k+1}, which captures the cost of training a neural network that is deeper than the sub-network. In particular, observe that the deeper the stable sub-network is (increasing k), the smaller the cost. For a deep stable sub-network ($k = L-1$), we reach a factor B^L which is similar to dimension-free analysis of deep ReLU neural networks [Golowich et al., 2018].

Finally, in Figure 3, we provide some empirical validation of our main hypothesis and our main result by showing that the prediction stability of the entire network is roughly of the same order (with respect to n) as that of the sub-network stability: this is captured by similar slopes in the log-scale, which corresponds to $-\alpha$ in $\epsilon = n^{-\alpha}$ assumption.

4 Conclusions, limitations, and future work

In this work, we have studied sufficient conditions for algorithmic stability in minimum-norm interpolating deep ReLU neural networks. This study opens up several interesting avenues for future

[1] In fact, they showed a slightly stronger upper bound on the harmonic mean: $\frac{k}{\sum_{j=1}^{k}(1/\mathrm{S}(\widehat{\mathbf{W}}_j))} \leq B^{\frac{L}{k}}$.

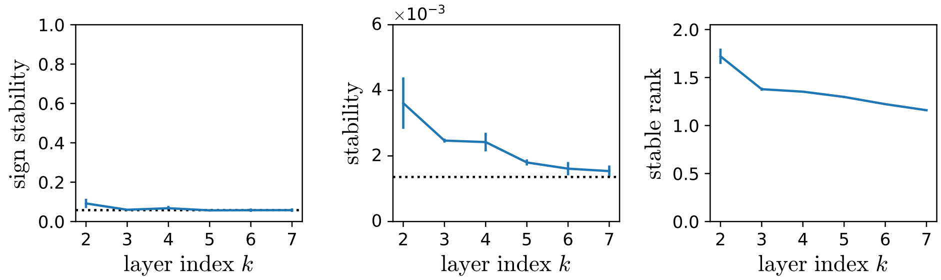

Figure 2: **Stability of sub-networks and stable rank of the layers.** We trained an 8-layer FCN on a uniformly drawn 10^4 MNIST sample by minimizing a mean square error (MSE) loss to near zero, classifying the first 5 classes as -1 and others as 1. We performed multiple trials, where each trial is with identical initialization and a different portion of the training set is replaced for each trial. The error bars are 1 standard deviation of the trial. Using the models, we measured the sign stability **(left)**, i.e. $|\text{sign}(f_k(\mathbf{x}; \hat{\theta})) - \text{sign}(f_k(\mathbf{x}; \hat{\theta}^{(i)}))|$, stability **(middle)**, and the stable rank of weight matrix **(right)** for each sub-network f_k for $2 \leq k \leq 7$. The horizontal dotted lines are the (sign) stability of the full network. For the details of the experiment, link to our code, and additional experiments on Fashion-MNIST, see Appendix E.

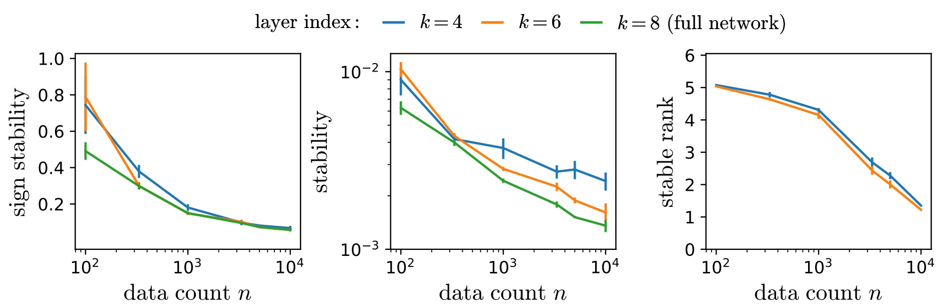

Figure 3: **Stability as a function of number of data points** We followed the same setting as in Fig. 2 while varying training data set sizes. Both the sign stability **(left)** and stability **(middle)** of sub-networks (blue and orange) decay at a rate similar to that of the full network (green). The stable rank of weight matrices **(right)** also decreases as a function of n, suggesting that **??** holds in the large n limit. Observe that the slopes for the sub-networks and the full network are similar which validates that the respective stabilities have the same dependency in n (Theorem 1).

research. One of the sufficient conditions we examined is the existence of a stable sub-network, which we did not prove theoretically and which remains an open question. In related areas, such as the lottery ticket hypothesis, the existence of a well-performing sub-network has been shown theoretically, which might inspire the use of similar techniques to prove the existence of a stable sub-network. Finally, an interesting open question is bridging the gap between the stability of minimum-norm interpolation in neural networks under GF and actual optimization algorithms, such as SGD. While there are several promising directions, a complete picture incorporating stability analysis might require additional arguments [Poggio et al., 2020, Elkabetz and Cohen, 2021].

One possible limitation of our analysis is that the stability bound involves a B-dependent factor which scales as B^{2L-k+1} in the best case. For the case $k = L - 1$, it might be possible to achieve a better factor B^L which would be in line with Rademacher complexity analysis [Golowich et al., 2018].

Even though we empirically validate that our assumptions hold for bias-free FCNs trained on some datasets, whether the condition holds in larger architectures trained on more complex datasets requires more empirical exploration. In the future, we aim to extend our results to more complex scenarios and empirically explore the limit in which deep neural networks satisfy our assumptions.

References

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

Peter L Bartlett, Dylan J Foster, and Matus J Telgarsky. Spectrally-normalized margin bounds for neural networks. *Advances in Neural Information Processing Systems*, 30, 2017.

Stéphane Boucheron, Gábor Lugosi, and Pascal Massart. *Concentration inequalities: A nonasymptotic theory of independence*. Oxford University Press, 2013.

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

Olivier Bousquet, Stéphane Boucheron, and Gábor Lugosi. *Introduction to Statistical Learning Theory*, pages 169–207. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004. ISBN 978-3-540-28650-9. doi: 10.1007/978-3-540-28650-9_8. URL https://doi.org/10.1007/978-3-540-28650-9_8.

Olivier Bousquet, Yegor Klochkov, and Nikita Zhivotovskiy. Sharper bounds for uniformly stable algorithms. In *Conference on Computational Learning Theory (COLT)*, 2020.

Zachary Charles and Dimitris Papailiopoulos. Stability and generalization of learning algorithms that converge to global optima. In *International Conference on Machine Learing (ICML)*, 2018.

Luc Devroye and Terry Wagner. Distribution-free inequalities for the deleted and holdout error estimates. *IEEE Transactions on Information Theory*, 25(2):202–207, 1979.

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

Omer Elkabetz and Nadav Cohen. Continuous vs. discrete optimization of deep neural networks. *Advances in Neural Information Processing Systems*, 2021.

Cong Fang, Hangfeng He, Qi Long, and Weijie J Su. Layer-peeled model: Toward understanding well-trained deep neural networks. *arXiv preprint arXiv:2101.12699*, 4, 2021.

Tyler Farghly and Patrick Rebeschini. Time-independent generalization bounds for sgld in non-convex settings. *Advances in Neural Information Processing Systems*, 2021.

Vitaly Feldman and Jan Vondrak. Generalization bounds for uniformly stable algorithms. *Advances in Neural Information Processing Systems*, 31, 2018.

Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *International Conference on Learning Representations (ICLR)*, 2018.

Jonathan Frankle, Gintare Karolina Dziugaite, Daniel Roy, and Michael Carbin. Linear mode connectivity and the lottery ticket hypothesis. In *International Conference on Machine Learing (ICML)*, 2020.

Spencer Frei, Gal Vardi, Peter Bartlett, Nathan Srebro, and Wei Hu. Implicit bias in leaky relu networks trained on high-dimensional data. In *International Conference on Learning Representations (ICLR)*, 2022.

Spencer Frei, Gal Vardi, Peter Bartlett, and Nathan Srebro. Benign overfitting in linear classifiers and leaky relu networks from kkt conditions for margin maximization. In *Conference on Computational Learning Theory (COLT)*. PMLR, 2023.

Tomer Galanti, Zachary S. Siegel, Aparna Gupte, and Tomaso Poggio. Characterizing the implicit bias of regularized sgd in rank minimization, 2023.

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

XY Han, Vardan Papyan, and David L Donoho. Neural collapse under mse loss: Proximity to and dynamics on the central path. In *International Conference on Learning Representations (ICLR)*, 2021.

Moritz Hardt, Ben Recht, and Yoram Singer. Train faster, generalize better: Stability of stochastic gradient descent. In *International Conference on Machine Learing (ICML)*, 2016.

Ziwei Ji and Matus Telgarsky. Polylogarithmic width suffices for gradient descent to achieve arbitrarily small test error with shallow relu networks. In *International Conference on Learning Representations (ICLR)*, 2019.

Ziwei Ji and Matus Telgarsky. Directional convergence and alignment in deep learning. *Advances in Neural Information Processing Systems*, 2020.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015.

Frederic Koehler, Lijia Zhou, Danica J Sutherland, and Nathan Srebro. Uniform convergence of interpolators: Gaussian width, norm bounds and benign overfitting. *Advances in Neural Information Processing Systems*, 2021.

Samuel Kutin and Partha Niyogi. Almost-everywhere algorithmic stability and generalization error. In *Uncertainty in Artificial Intelligence (UAI)*, 2002.

Ilja Kuzborskij and Christoph H Lampert. Data-Dependent Stability of Stochastic Gradient Descent. In *International Conference on Machine Learing (ICML)*, 2018.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Yunwen Lei and Yiming Ying. Fine-grained analysis of stability and generalization for stochastic gradient descent. In *International Conference on Machine Learing (ICML)*, 2020.

Kaifeng Lyu and Jian Li. Gradient descent maximizes the margin of homogeneous neural networks. In *International Conference on Learning Representations (ICLR)*, 2019.

Eran Malach, Gilad Yehudai, Shai Shalev-Schwartz, and Ohad Shamir. Proving the lottery ticket hypothesis: Pruning is all you need. In *International Conference on Machine Learing (ICML)*, 2020.

Dustin G. Mixon, Hans Parshall, and Jianzong Pi. Neural collapse with unconstrained features. *Sampling Theory, Signal Processing, and Data Analysis*, 20(2):11, 7 2022.

Laurent Orseau, Marcus Hutter, and Omar Rivasplata. Logarithmic pruning is all you need. *Advances in Neural Information Processing Systems*, 2020.

Vardan Papyan, XY Han, and David L Donoho. Prevalence of neural collapse during the terminal phase of deep learning training. *Proceedings of the National Academy of Sciences of the United States of America*, 117(40):24652–24663, 2020.

Mary Phuong and Christoph H Lampert. The inductive bias of relu networks on orthogonally separable data. In *International Conference on Learning Representations (ICLR)*, 2020.

Tomaso Poggio, Andrzej Banburski, and Qianli Liao. Theoretical issues in deep networks. *Proceedings of the National Academy of Sciences of the United States of America*, 117(48):30039–30045, 2020.

Akshay Rangamani, Lorenzo Rosasco, and Tomaso Poggio. For interpolating kernel machines, minimizing the norm of the erm solution maximizes stability. *Analysis and Applications*, 21(01): 193–215, 2023.

Dominic Richards and Ilja Kuzborskij. Stability & generalisation of gradient descent for shallow neural networks without the neural tangent kernel. In *Advances in Neural Information Processing Systems*, 2021.

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

Keitaro Sakamoto and Issei Sato. Analyzing lottery ticket hypothesis from pac-bayesian theory perspective. *Advances in Neural Information Processing Systems*, 2022.

Matan Schliserman and Tomer Koren. Stability vs implicit bias of gradient methods on separable data and beyond. In *Conference on Learning Theory*, 2022.

Shai Shalev-Shwartz, Ohad Shamir, Nathan Srebro, and Karthik Sridharan. Learnability, stability and uniform convergence. *Journal of Machine Learning Research*, 11:2635–2670, 2010.

Matus Telgarsky. Feature selection and low test error in shallow low-rotation relu networks. In *International Conference on Learning Representations (ICLR)*, 2022.

Nadav Timor, Gal Vardi, and Ohad Shamir. Implicit regularization towards rank minimization in relu networks. In *Algorithmic Learning Theory (ALT)*, 2023.

Joel A Tropp. An introduction to matrix concentration inequalities. *Foundations and Trends® in Machine Learning*, 8(1-2):1–230, 2015.

Alexander Tsigler, Gabor Lugosi, Peter Bartlett, and Phil Long. Benign overfitting in linear regression. *PNAS*, 117(48):30063–30070, 2020.

Puyu Wang, Yunwen Lei, Di Wang, Yiming Ying, and Ding-Xuan Zhou. Generalization guarantees of gradient descent for multi-layer neural networks. *arXiv preprint arXiv:2305.16891*, 2023.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3):107–115, 2021.

A Applications of the main result.

The stability bound we presented can also used in some applications. In particular, when combined with Eq. (2), it implies a high-probability bound on the generalization error:

$$|R(\hat{\theta}) - \hat{R}(\hat{\theta})| = \tilde{\mathcal{O}}\left(\left(1 + B^{2L-k+1} + a\, B^{3L-k+1}\right)\epsilon + \frac{1}{\sqrt{n}}\right) .$$

Under assumption $\epsilon = n^{-\alpha}$ for $\alpha > 0$, the generalization error converges 0 as $n \to \infty$ in probability.

Finally, the stability bound can also be used to give a bound on the variance of a trained neural network, which is not normally achievable through the uniform convergence bounds. Controlling the variance of trained predictors is interesting in the context of uncertainty quantification (e.g., through ensembling, as discussed in [Elisseeff et al., 2005]). In particular, Efron-Stein inequality [Boucheron et al., 2013] implies that

$$\mathrm{Var}(N(\hat{\theta})) \leq \frac{C^2}{2}\left(1 + B^{2L-k+1} + a\, B^{3L-k+1}\right)^2 n\, \epsilon^2 .$$

Once again, assuming $\epsilon = n^{-\alpha}$ it turns out that the variance converges to 0 asymptotically only if $\alpha > 1/2$, that is when sub-network is very stable with $\epsilon < 1/\sqrt{n}$.

B Proof of the main result

Theorem 1. *Suppose that datasets are B-admissible according to Assumption 1. Let $\epsilon = M/n^{-\alpha}$, for some $M \geq 0, \alpha > 0$ and let $a > 0$, consider $L > L'$ and $2 \leq k \leq L-1$ that satisfy (4). Suppose that the sub-network is β-uniformly ϵ-stable according to Hypothesis 1. Then, assuming that the sample size satisfies $n \geq \max\left(a \cdot 2MB^L,\, 4MB^{L-k+1}\right)^{\frac{1}{\alpha}}$, $\mathcal{T}$ is β-uniformly ϵ'-prediction stable with*

$$\epsilon' = (1 + 8B^{2L-k+1})\,\epsilon + 2a\left(B^L + B^{2L-k+1}\epsilon + 4B^{3L-k+1}\right)\,\epsilon .$$

The proof relies on Lemmas 1, 2 and 3, shown Appendix D. First, we discuss high-level proof ideas.

Some proof ideas For simplicity consider a rank-one case ($a = 0$). The key Lemma 2 shows that if a deep ReLU network interpolates the data, then the prediction done at the rank-one layer is maintained throughout the rest of the layers. Consider a decomposition $\mathbf{W}_k = \lambda_k \mathbf{u}_k \mathbf{v}_k^\top$, where $\mathbf{v}_k$ defines a hyperplane which separates inputs in the feature space of the previous layer given by $N^{1:k-1}(\,\cdot\,, \hat{\theta})$. Then, multiplication by $\lambda_k \mathbf{u}_k$ and the propagation through all subsequent layers is just a rescaling of the predictions to fit the labels. In other words, the sign of $f_k(\mathbf{x}) = \mathbf{v}_k^\top N^{1:k-1}(\mathbf{x}, \hat{\theta})$ for each $\mathbf{x}$ already determines the final output and the prediction is done at the sub-network level (see Fig. 6 in Appendix E for further discussion).

Next, in Lemma 3, we show that if a deep ReLU network is a minimum-norm interpolant of the data, the intermediate prediction of data points at the sub-network level must have substantial margin γ. Intuitively, if the margin were infinitesimally small, this would require a large contribution in (one of the weights of) the subsequent layers to compensate and yield an output of order 1, which is forbidden by the fact that the network has minimum norm.

Lemma 2. *Suppose that there exist $k \leq L-1$ and $\epsilon \geq 0$ that satisfy **??**. Then, there exists a bounded function $b(\,\cdot\,;\theta)$, and $C_\theta^+, C_\theta^+ > 0$ (independent from the input), such that, for all $\mathbf{x} \in \mathbb{R}^{d_0}$ the following is true:*

1. *If $f_k(\mathbf{x};\, \theta) > 0$, then $N(\mathbf{x}) + b(\mathbf{x};\, \theta) \cdot \epsilon = C_\theta^+ \cdot f_k(\mathbf{x};\, \theta)$.*
2. *If $f_k(\mathbf{x};\, \theta) \leq 0$, then $N(\mathbf{x}) + b(\mathbf{x};\, \theta) \cdot \epsilon = C_\theta^- \cdot f_k(\mathbf{x};\, \theta)$.*
3. *For all $\mathbf{x} \in \mathbb{R}^{d_0}$, $|b(\mathbf{x};\, \theta)| \leq a\, \left(\prod_{j=1}^L \|\mathbf{W}_j\|_2\right)$.*
4. *$\left\|\mathbf{W}_k^\epsilon\, N^{1:k-1}(\mathbf{x})\right\| \leq a\, \epsilon\, \prod_{j=1}^k \|\mathbf{W}_j\|_2$.*

Lemma 3. *Suppose that datasets are B-admissible according to Assumption 1. Suppose that there exist $k \leq L-1$ and $\epsilon = M/n^{-\alpha}$, for some $M \geq 0, \alpha > 0$, that satisfy* **??**. *Let $\hat{\theta} = \mathcal{T}(\mathbf{X}, \mathbf{y})$ and $\hat{\theta}^{(i)} = \mathcal{T}(\mathbf{X}^{(i)}, \mathbf{y}^{(i)})$ and assume that $|y_i| \geq 1$ for all $i \in [n]$. Then, assuming that the sample size satisfies $n \geq \left(2 \cdot M \cdot a \cdot B^L\right)^{\frac{1}{\alpha}}$, for any $\gamma \leq 1/(2 \cdot B^{L-k+1})$, and $\mu \geq B^{k-1}$,*

$$\gamma \leq \left| f_k(\mathbf{x}_i;\, \hat{\theta}) \cdot y_i \right| \leq \mu \qquad (\forall i \in [n]).$$

Proof of Theorem 1. In the following let $(\mathbf{x}, y)$ be the point replacing the ith example in the training set $(\mathbf{X}, \mathbf{y})$ and so by interpolation we have $y = N(\mathbf{x};\, \hat{\theta}^{(i)})$.

By Lemma 2 there exist $C_{\hat{\theta}}, C_{\hat{\theta}^{(i)}}$ independent from the input and $b(\,\cdot\,;\, \hat{\theta})$, $b(\,\cdot\,;\, \hat{\theta}^{(i)})$ such that

$$N(\mathbf{x};\, \hat{\theta}) + b(\mathbf{x};\, \hat{\theta}) \cdot \epsilon = C_{\hat{\theta}} \cdot f_k(\mathbf{x};\, \hat{\theta}) \;, \quad N(\mathbf{x};\, \hat{\theta}^{(i)}) + b(\mathbf{x};\, \hat{\theta}^{(i)}) \cdot \epsilon = C_{\hat{\theta}^{(i)}} \cdot f_k(\mathbf{x};\, \hat{\theta}^{(i)}) \;. \tag{5}$$

Observe that,

$$\begin{aligned}
&\left| N(\mathbf{x};\, \hat{\theta}) - N(\mathbf{x};\, \hat{\theta}^{(i)}) \right| - \left| \left(b(\mathbf{x};\, \hat{\theta}) - b(\mathbf{x};\, \hat{\theta}^{(i)}) \right) \cdot \epsilon \right| \\
&\leq \left| N(\mathbf{x};\, \hat{\theta}) - N(\mathbf{x};\, \hat{\theta}^{(i)}) + (b(\mathbf{x};\, \hat{\theta}) - b(\mathbf{x};\, \hat{\theta}^{(i)})) \cdot \epsilon \right| \\
&\leq \left| C_{\hat{\theta}} \cdot f_k(\mathbf{x};\, \hat{\theta}) - C_{\hat{\theta}} \cdot f_k(\mathbf{x};\, \hat{\theta}^{(i)}) \right| + \left| C_{\hat{\theta}} \cdot f_k(\mathbf{x};\, \hat{\theta}^{(i)}) - C_{\hat{\theta}^{(i)}} \cdot f_k(\mathbf{x};\, \hat{\theta}^{(i)}) \right| \\
&\leq \left| C_{\hat{\theta}} \right| \cdot \left| f_k(\mathbf{x};\, \hat{\theta}) - f_k(\mathbf{x};\, \hat{\theta}^{(i)}) \right| + \left| f_k(\mathbf{x};\, \hat{\theta}^{(i)}) \right| \cdot \left| C_{\hat{\theta}} - C_{\hat{\theta}^{(i)}} \right| \\
&\leq \left| C_{\hat{\theta}} \right| \cdot \epsilon + \left| f_k(\mathbf{x};\, \hat{\theta}^{(i)}) \right| \cdot \left| C_{\hat{\theta}} - C_{\hat{\theta}^{(i)}} \right| \qquad \text{(By sub-network stability assumption)} \\
&\leq \left| C_{\hat{\theta}} \right| \cdot \epsilon + \mu \cdot \left| C_{\hat{\theta}} - C_{\hat{\theta}^{(i)}} \right| \;. \qquad \text{(By Lemma 3)}
\end{aligned}$$

First note that by Lemma 2,

$$|b(\mathbf{x};\, \hat{\theta})| \leq a \prod_{j=1}^{L} \|\widehat{\mathbf{W}}_j\|_2 \leq a\, B^L =: b$$

where $\|\widehat{\mathbf{W}}_j\|_2 \leq B$ comes as in the first part of the proof of Lemma 3. Similarly, $|b(\mathbf{x};\, \hat{\theta}^{(i)})| \leq b$. Then,

$$\left| \left(b(\mathbf{x};\, \hat{\theta}) - b(\mathbf{x};\, \hat{\theta}^{(i)}) \right) \cdot \epsilon \right| \leq 2\, a\, \epsilon\, B^L \;.$$

Now, the idea is to use Eq. (5) to control $|C_{\hat{\theta}}|$ and to control $|C_{\hat{\theta}} - C_{\hat{\theta}^{(i)}}|$ in terms of difference of sub-network predictions and invoke sub-network stability once again. W.l.o.g., let $i \neq 1$ and so

$$\begin{aligned}
C_{\hat{\theta}} &= \frac{N(\mathbf{x}_1;\, \hat{\theta}) + b(\mathbf{x}_1;\, \hat{\theta}) \cdot \epsilon}{f_k(\mathbf{x}_1;\, \hat{\theta})} = \frac{y_1 + b(\mathbf{x}_1;\, \hat{\theta}) \cdot \epsilon}{f_k(\mathbf{x}_1;\, \hat{\theta})} \;, \\
C_{\hat{\theta}^{(i)}} &= \frac{N(\mathbf{x}_1;\, \hat{\theta}^{(i)}) + b(\mathbf{x}_1;\, \hat{\theta}^{(i)}) \cdot \epsilon}{f_k(\mathbf{x}_1;\, \hat{\theta}^{(i)})} = \frac{y_1 + b(\mathbf{x}_1;\, \hat{\theta}^{(i)}) \cdot \epsilon}{f_k(\mathbf{x}_1;\, \hat{\theta}^{(i)})} \;.
\end{aligned}$$

So, by Lemma 2 and Lemma 3 (with γ defined therein),

$$|C_{\hat{\theta}}| \leq \frac{|y_1| + b \cdot \epsilon}{|f_k(\mathbf{x}_1;\, \hat{\theta})|} \leq \frac{|y_1| + b \cdot \epsilon}{\gamma} \;.$$

Now we turn our attention to the gap:

$$\begin{aligned}
\left| C_{\hat{\theta}} - C_{\hat{\theta}^{(i)}} \right| &= \left| \frac{y_1 + b(\mathbf{x}_1;\, \hat{\theta}) \cdot \epsilon}{f_k(\mathbf{x}_1;\, \hat{\theta})} - \frac{y_1 + b(\mathbf{x}_1;\, \hat{\theta}^{(i)}) \cdot \epsilon}{f_k(\mathbf{x}_1;\, \hat{\theta}^{(i)})} \right| \\
&\leq |y_1| \left| \frac{f_k(\mathbf{x}_1;\, \hat{\theta}) - f_k(\mathbf{x}_1;\, \hat{\theta}^{(i)})}{f_k(\mathbf{x}_1;\, \hat{\theta}) \cdot f_k(\mathbf{x}_1;\, \hat{\theta}^{(i)})} \right| + \left| \frac{b(\mathbf{x}_1;\, \hat{\theta})}{f_k(\mathbf{x}_1;\, \hat{\theta})} - \frac{b(\mathbf{x}_1;\, \hat{\theta}^{(i)})}{f_k(\mathbf{x}_1;\, \hat{\theta}^{(i)})} \right| \cdot \epsilon \\
&\overset{(a)}{\leq} \frac{|y_1| \cdot \epsilon}{\left| f_k(\mathbf{x}_1;\, \hat{\theta}) \cdot f_k(\mathbf{x}_1;\, \hat{\theta}^{(i)}) \right|} + \frac{b \cdot \epsilon}{\left| f_k(\mathbf{x}_1;\, \hat{\theta}) \cdot f_k(\mathbf{x}_1;\, \hat{\theta}^{(i)}) \right|} \\
&\overset{(b)}{\leq} \frac{2\,(|y_1| + b)\,\epsilon}{(\gamma - \epsilon)_+^2 + \gamma^2 - \epsilon^2}
\end{aligned}$$

where in step (a) we used the assumption that sub-network is ϵ-stable and Lemma 2, while step (b) amounts to lower-bounding the denominator, and deferred till the end of the proof.

Putting all together we have

$$\left|N(\mathbf{x};\,\hat{\theta}) - N(\mathbf{x};\,\hat{\theta}^{(i)})\right| \leq 2 \cdot b \cdot \epsilon + \frac{1 + b \cdot \epsilon}{\gamma} \cdot \epsilon + \mu \cdot \frac{2\,(1+b)\,\epsilon}{(\gamma - \epsilon)_+^2 + \gamma^2 - \epsilon^2} \,.$$

Now, let's require $\epsilon \leq \gamma/2$: By assumption we have that $\epsilon = (M/n)^\alpha$ and so our requirement is equivalent to $n \geq ((2M)/\gamma)^{\frac{1}{\alpha}}$. In overall, this gives

$$\left|N(\mathbf{x};\,\hat{\theta}) - N(\mathbf{x};\,\hat{\theta}^{(i)})\right| \leq (2b+1)\epsilon + \frac{b}{\gamma}\,\epsilon^2 + 2\,(1+b)\,\frac{\mu}{\gamma^2}\,\epsilon \,.$$

We conclude by using the bounds on γ and μ in Lemma 3, and the one on b which comes by Lemma 2.

Proof of step (b) By Lemma 3 we have that $\left|f_k(\mathbf{x};\,\hat{\theta}^{(i)})\right| \geq \gamma$. Now, given the above, the fact that the sub-network is ϵ-stable, and triangle inequality we have

$$\epsilon \geq \left|f_k(\mathbf{x};\,\hat{\theta}) - f_k(\mathbf{x};\,\hat{\theta}^{(i)})\right| \geq \left|f_k(\mathbf{x};\,\hat{\theta}^{(i)})\right| - \left|f_k(\mathbf{x};\,\hat{\theta})\right| \geq \gamma - \left|f_k(\mathbf{x};\,\hat{\theta})\right| \,.$$

This gives

$$\begin{aligned}
&\left(f_k(\mathbf{x};\,\hat{\theta}) - f_k(\mathbf{x};\,\hat{\theta}^{(i)})\right)^2 \leq \epsilon^2 \\
\iff \quad & f_k(\mathbf{x};\,\hat{\theta})^2 + f_k(\mathbf{x};\,\hat{\theta}^{(i)})^2 - \epsilon^2 \leq 2\,f_k(\mathbf{x};\,\hat{\theta}) \cdot f_k(\mathbf{x};\,\hat{\theta}^{(i)}) \\
\implies \quad & \frac{(\gamma-\epsilon)_+^2 + \gamma^2 - \epsilon^2}{2} \leq f_k(\mathbf{x};\,\hat{\theta}) \cdot f_k(\mathbf{x};\,\hat{\theta}^{(i)}) \,.
\end{aligned}$$

□

C Additional related work

Algorithmic stability The stability of interpolating kernel least-squares has been studied by Rangamani et al. [2023] who established a connection to stability of the pseudo-inverse, which is indeed a minimum-norm interpolation, and which is in turn controlled by the smallest non-zero eigenvalue of the kernel matrix. In this paper we look into a rather different setting of interpolation with neural networks rather than kernel machines, and provide sufficient conditions for such stability.

Recently, Schliserman and Koren [2022] analyzed the performance of gradient descent type methods on convex learning problems given linearly separable data. The setting of our paper can be interpreted as noise-free labels and so in case of classification, we have separability as well, however the decision boundary can be non-linear (as we can conclude from Assumption 1).

Neural collapse Recently, a phenomenon called neural collapse (NC) [Papyan et al., 2020, Han et al., 2021] has been observed where the post-activations (and weights of a final layer) of a well-trained neural network, appear to be clustered in a low-dimensional subspace. Theoretical studies on the cause of NC mostly rely on unconstrained feature models [Fang et al., 2021, Mixon et al., 2022] where the product of two matrices is trained under gradient flow: the two matrices train to a low-rank structure where their ranks equal the dimension of the outputs.

Lottery ticket hypothesis The existence of a small good neural network within a large deep neural network has been proposed before as a prominent hypothesis in neural network learning. In particular the *lottery ticket hypothesis* [Frankle and Carbin, 2018] posits that deep neural networks contain small sub-networks which could be trained in isolation and lead to comparable performance, and that these sub-networks happen to be favored by standard initializations The *lottery ticket hypothesis* has however only been analyzed in some restricted settings [Frankle et al., 2020, Malach et al., 2020, Orseau et al., 2020, Sakamoto and Sato, 2022]. It is still an active area of theoretical research. In this paper we explore a related concept, where the quality of the sub-network is captured by its stability, which establishes a link to analysis of the generalization error.

D Omitted proofs

D.1 Proof of Lemma 1

Invoking Theorem 4 in Timor et al. [2023], we have that $\hat{\theta}$ obtained by algorihtm $\mathcal{T}$ for architecture $\mathbb{A} = \langle L, d, d_0 \rangle$ verifies:

$$\frac{L}{\sum_{k=1}^{L} \left(\mathrm{S}(\hat{\mathbf{W}}_k))\right)^{-1}} \leq B^{\frac{L^*}{L}}$$

thus

$$\frac{1}{L} \sum_{k=1}^{L} \left(\mathrm{S}(\hat{\mathbf{W}}_k))\right)^{-1} \geq \frac{1}{B^{\frac{L^*}{L}}}$$

Since the quantity on the left is an average, there must exist $1 \leq k \leq L$ such that

$$\left(\mathrm{S}(\hat{\mathbf{W}}_k))\right)^{-1} \geq \frac{1}{B^{\frac{L^*}{L}}} \tag{6}$$

thus

$$\mathrm{S}(\hat{\mathbf{W}}_k)) \leq B^{\frac{L^*}{L}} \ .$$

By setting $L \geq L^* \frac{\log(B)}{\log(1+a\cdot\epsilon)}$, we ensure that:

$$\mathrm{S}(\hat{\mathbf{W}}_k)) \leq 1 + a \cdot \epsilon \ .$$

We note $\hat{\mathbf{W}}_k = \lambda_k \mathbf{u}_k \mathbf{v}_k^T + \hat{\mathbf{W}}_k^\epsilon$ its decomposition following Definition 1.

D.2 Proof of Lemma 2

Throughout the proof we drop dependence on θ, e.g. $N(\mathbf{x}) \equiv N(\mathbf{x};\, \theta)$.

Introduce

$$b(\mathbf{x}) := \frac{1}{\epsilon} \cdot \left(N^{k+1:L}\left(\mathrm{ReLU}\left(\lambda_k \mathbf{u}_k \mathbf{v}_k^\top N^{1:k-1}(\mathbf{x})\right)\right) - N(\mathbf{x})\right)$$

and we can thus write:

$$N(\mathbf{x}) + b(\mathbf{x}) \cdot \epsilon = N^{k+1:L}\left(\mathrm{ReLU}\left(\lambda_k \mathbf{u}_k \mathbf{v}_k^\top N^{1:k-1}(\mathbf{x})\right)\right) \ .$$

Showing statement 1. First consider the case when $\mathbf{v}_k^\top N^{1:k-1}(\mathbf{x}) > 0$. Now, using the fact that ReLU is positively-homogeneous,

$$\mathrm{ReLU}\left(\lambda_k \mathbf{u}_k \mathbf{v}_k^\top N^{1:k-1}(\mathbf{x})\right) = \mathbf{v}_k^\top N^{1:k-1}(\mathbf{x}) \cdot \mathrm{ReLU}(\lambda_k \mathbf{u}_k)$$

Therefore, using positive-homogeneity once again,

$$\begin{aligned} N(\mathbf{x}) &= N^{k+1:L}\left(\mathbf{v}_k^\top N^{1:k-1}(\mathbf{x}) \cdot \mathrm{ReLU}(\lambda_k \mathbf{u}_k)\right) - b(\mathbf{x}) \cdot \epsilon \\ &= \mathbf{v}_k^\top N^{1:k-1}(\mathbf{x}) \cdot \underbrace{N^{k+1:L}\left(\mathrm{ReLU}(\lambda_k \mathbf{u}_k)\right)}_{C^+} - b(\mathbf{x}) \cdot \epsilon \end{aligned}$$

where we note that C^+ can take a different sign since $C^+ = \mathbf{W}_L^\top N^{k+1:L-1}\left(\mathrm{ReLU}(\lambda_k \mathbf{u}_k)\right)$.

Showing statement 2. Now, considering an alternative case $\mathbf{v}_k^\top N^{1:k-1}(\mathbf{x}) \leq 0$, positive-homogeneity once again gives

$$\mathrm{ReLU}\left(\lambda_k \mathbf{u}_k \mathbf{v}_k^\top N^{1:k-1}(\mathbf{x})\right) = -\mathbf{v}_k^\top N^{1:k-1}(\mathbf{x}) \cdot \mathrm{ReLU}(-\lambda_k \mathbf{u}_k)$$

and so

$$\begin{aligned} N(\mathbf{x}) &= N^{k+1:L}\left(-\mathbf{v}_k^\top N^{1:k-1}(\mathbf{x}) \cdot \mathrm{ReLU}(-\lambda_k \mathbf{u}_k)\right) - b(\mathbf{x}) \cdot \epsilon \\ &= \mathbf{v}_k^\top N^{1:k-1}(\mathbf{x}) \cdot \underbrace{\left(-N^{k+1:L}\left(\mathrm{ReLU}(-\lambda_k \mathbf{u}_k)\right)\right)}_{C^-} - b(\mathbf{x}) \cdot \epsilon \ . \end{aligned}$$

Showing statement 3 and 4. It suffices to prove that for any input $\mathbf{x} \in \mathbb{R}^{d_0}$,

$$\left|N(\mathbf{x}) - N^{k+1:L}\left(\mathrm{ReLU}\left(\lambda_k \mathbf{u}_k \mathbf{v}_k^\top N^{1:k-1}(\mathbf{x})\right)\right)\right| \leq a\,\epsilon\, B^L\,.$$

By 1-Lipschitzness of ReLU, we have:

$$\begin{aligned}
&\|\mathrm{ReLU}\left(\mathbf{W}_k\ N^{1:k-1}(\mathbf{x})\right) - \mathrm{ReLU}\left(\lambda_k \mathbf{u}_k \mathbf{v}_k^\top N^{1:k-1}(\mathbf{x})\right)\| \\
&\quad \leq \|\mathbf{W}_k^\epsilon\ N^{1:k-1}(\mathbf{x})\| \\
&\quad \leq \|\mathbf{W}_k^\epsilon\|_F\ \|N^{1:k-1}(\mathbf{x})\| \\
&\quad \leq a\,\epsilon\,\|\mathbf{W}_k\|_F\,\|N^{1:k-1}(\mathbf{x})\|\,.
\end{aligned}$$

where the last inequality comes by the following observation:

$$\mathrm{S}(\mathbf{W}_k) \leq 1 + a\,\epsilon \quad \Longrightarrow \quad \|\mathbf{W}_k^\epsilon\|_F \leq a\,\epsilon\,\|\mathbf{W}_k\|_F\,.$$

At the same time, note that by 1-Lipschitzness of ReLU, and Cauchy-Schwartz inequality we have that for any $\mathbf{z}, \mathbf{z}' \in \mathbb{R}^{d_k}$,

$$|N^{k+1:L}(\mathbf{z}) - N^{k+1:L}(\mathbf{z}')| \leq \|\mathbf{z} - \mathbf{z}'\| \prod_{j=k+1}^{L} \|\mathbf{W}_j\|_F\,.$$

Combining the above we have

$$\begin{aligned}
&\left|N(\mathbf{x}) - N^{k+1:L}\left(\mathrm{ReLU}\left(\lambda_k \mathbf{u}_k \mathbf{v}_k^\top N^{1:k-1}(\mathbf{x})\right)\right)\right| \\
&\quad = \left|N^{k+1:L}\left(\mathrm{ReLU}\left(\mathbf{W}_k\ N^{1:k-1}(\mathbf{x})\right)\right) - N^{k+1:L}\left(\mathrm{ReLU}\left(\lambda_k \mathbf{u}_k \mathbf{v}_k^\top N^{1:k-1}(\mathbf{x})\right)\right)\right| \\
&\quad \leq \left(\prod_{j=k}^{L} \|\mathbf{W}_j\|_F\right) \cdot a \cdot \epsilon\,\|N^{1:k-1}(\mathbf{x})\| \\
&\quad \leq \left(\prod_{j=1}^{L} \|\mathbf{W}_j\|_F\right) \cdot a \cdot \epsilon
\end{aligned}$$

where the last inequality comes by Cauchy-Schwartz inequality, realizing that $\mathrm{ReLU}(|x|) = |x|$, and the fact that $\|\mathbf{x}\| \leq 1$. ◻

D.3 Proof of Lemma 3

We will require the following:

Lemma 4 (Timor et al. [2023, Lemma 14])**.** *Let $\hat{\theta} = \mathcal{T}(\mathbf{X}, \mathbf{y})$. Then, for every $1 \leq i < j \leq L$ we have $\|\widehat{\mathbf{W}}_i\|_F = \|\widehat{\mathbf{W}}_j\|_F$.*

First, notice that $N(\cdot, \hat{\theta})$ is a minimum-norm interpolant and so Lemma 4, $\leq e^{\frac{k-1}{2}}\ B^{L-k+1}$. the weight matrices $\widehat{\mathbf{W}}_\ell$ have the same norm and thus verify:

$$\|\widehat{\mathbf{W}}_\ell\|_F^2 = \frac{1}{L}\sum_{\ell=1}^{L} \|\widehat{\mathbf{W}}_\ell\|_F^2 \leq \frac{1}{L}\sum_{\ell=1}^{L} \|\mathbf{W}_\ell^*\|_F^2 \leq \frac{1}{L} \cdot L \cdot B^2$$

which gives us $\|\widehat{\mathbf{W}}_\ell\|_F \leq B$.

Proof of a lower bound. Using the fact that $N(\mathbf{x}_i;\, \hat{\theta}) = y_i$ and assumption that $|y_i| \geq 1$ for each i, we have:

$$\begin{aligned}
1 \leq |y_i| &= |N(\mathbf{x}_i;\, \hat{\theta})| \\
&= \left|N^{k+1:L}\left(\mathrm{ReLU}\left(\hat{\lambda}_k \hat{\mathbf{u}}_k \hat{\mathbf{v}}_k^\top\ N^{1:k-1}(\mathbf{x}) + \widehat{\mathbf{W}}_k^\epsilon\ N^{1:k-1}(\mathbf{x})\right)\right)\right| \\
&\overset{(a)}{\leq} \left|\mathrm{ReLU}\left(\hat{\lambda}_k \hat{\mathbf{u}}_k \hat{\mathbf{v}}_k^\top\ N^{1:k-1}(\mathbf{x}) + \widehat{\mathbf{W}}_k^\epsilon\ N^{1:k-1}(\mathbf{x})\right)\right| \prod_{\ell=k+1}^{L} \|\widehat{\mathbf{W}}_\ell\|_2 \\
&\overset{(b)}{\leq} \left(\left|\hat{\mathbf{v}}_k^\top N^{1:k-1}(\mathbf{x}_i;\, \hat{\theta})\right| \|\hat{\mathbf{u}}_k\| |\hat{\lambda}_k| + \left\|\widehat{\mathbf{W}}_k^\epsilon\ N^{1:k-1}(\mathbf{x}_i;\, \hat{\theta})\right\|\right) \prod_{\ell=k+1}^{L} \|\widehat{\mathbf{W}}_\ell\|_F
\end{aligned}$$

where steps (a), (b) comes by the Cauchy-Schwartz inequality and the fact that $\mathrm{ReLU}(|x|) = |x|$.

Thus,

$$\frac{1}{B^{L-k}} \leq \left| \hat{\mathbf{v}}_k^\top N^{1:k-1}(\mathbf{x}_i;\, \hat{\theta}) \right| \|\hat{\mathbf{u}}_k\| |\hat{\lambda}_k| + \left\| \widehat{\mathbf{W}}_k^\epsilon \, N^{1:k-1}(\mathbf{x}_i;\, \hat{\theta}) \right\| \tag{7}$$

Furthermore by Lemma 2,

$$\left\| \widehat{\mathbf{W}}_k^\epsilon \, N^{1:k-1}(\mathbf{x}_i;\, \hat{\theta}) \right\| \leq a\,\epsilon \prod_{j=1}^{k} \|\mathbf{W}_k^\epsilon\|_2 \leq a\,\epsilon\, B^k \,.$$

By assumption of the Lemma we have $n \geq \left(2\, M\, a\, B^L\right)^{\frac{1}{\alpha}}$, and therefore $\epsilon = \frac{M}{n^\alpha} \leq \frac{1}{2 \cdot a \cdot B^L}$, which gives us that:

$$\left\| \widehat{\mathbf{W}}_k^\epsilon \, N^{1:k-1}(\mathbf{x}_i;\, \hat{\theta}) \right\| \leq \frac{1}{2\, B^{L-k}} \,.$$

Plugged into Eq. (7), and using the fact that $|\hat{\lambda}_k| \leq \|\widehat{\mathbf{W}}_k\|_2 \leq \|\widehat{\mathbf{W}}_k\|_F \leq B$ and using that $\|\hat{\mathbf{u}}_k\| = 1$, we have:

$$\left| \hat{\mathbf{v}}_k^\top N^{1:k-1}(\mathbf{x}_i;\, \hat{\theta}) \right| \geq \frac{1}{2\, B^{L-k+1}} \,.$$

Proof of an upper bound. Similarly, using the Cauchy-Schwartz inequality,

$$|f_k(\mathbf{x};\, \hat{\theta})| = \left| \hat{\mathbf{v}}_k^\top N^{1:k-1}(\mathbf{x}_i;\, \hat{\theta}) \right| \leq \|\hat{\mathbf{v}}_k\| \|\mathbf{x}_i\| \prod_{\ell=1}^{k-1} \|\widehat{\mathbf{W}}_\ell\|_F \leq B^{k-1} \,.$$

□

E Experiments

Code The code for our experiment is available at `https://anonymous.4open.science/r/stability_min_norm_neurips2024`

Dataset We used binarized MNIST [LeCun et al., 1998] and Fashion-MNIST [Xiao et al., 2017] datasets where we assigned target value -1 for the first 5 classes and 1 for all other classes. The images were flattened to 1-dimensional vectors.

Architecture For the experiment, we used a bias-free FCN with a width of 100 and a depth of 8. The model had a scalar output, in which MSE loss was used to classify between labels with output values -1 (first 5 classes) and 1 (latter 5 classes). The weight matrix W_k was initialized with Gaussian distribution with standard deviation $1/\sqrt{N_{k-1}}$.

Training All models were trained with Adam [Kingma and Ba, 2015] at a learning rate of 0.001 until it reached over 99% training accuracy. We used a weight decay of 0.005 to obtain a minimum norm solution. At epochs 60 and 120, we divided the learning rate and weight decay by 5.

Stability To calculate the stability, we prepared 5 models with identical architectures and initialization trained on $10,000$ mutually exclusive data points from MNIST. For all models, we perform SVD on the weight matrix of the k^{th} layer (W_k) to obtain the largest right eigenvector v_k and the subnetwork f_k (Eq. (3)). Because the output of subnetworks can vary up to a scale, we normalize the subnetwork such that $\sum_{\mathbf{x} \in \mathrm{TEST}} f_k(\mathbf{x};\, \theta^{(i)})^2 = 1$.

The stability was measured by the absolute difference of the f_k on the test set.[2]

$$\text{Stability} \approx \frac{1}{4} \sum_{i=2}^{5} \sum_{\mathbf{x} \in \mathrm{TEST}} |f_k(\mathbf{x};\, \theta^{(1)}) - f_k(\mathbf{x};\, \theta^{(i)})|.$$

[2]The empirical v_k may contain a global sign difference; we took the smallest stability obtained from $-v_k$ and v_k.

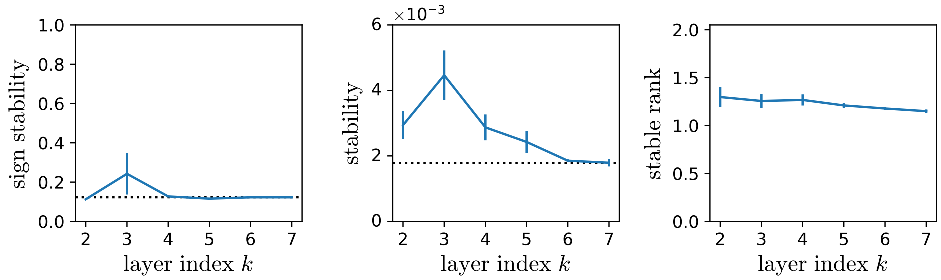

Figure 4: **Stability of subnetworks and stable rank of the layers (Fashion-MNIST).** We repeat the experiment of Fig. 2, but on Fashion-MNIST dataset. In agreement with the experiment for MNIST, FCN contains stable subnetwork and low-rank layers.

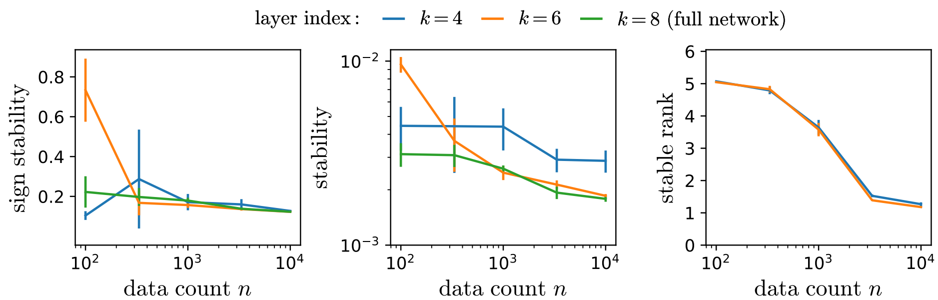

Figure 5: **Stability as a function of data points (Fashion-MNIST)** We repeat the experiment of Fig. 3, but on Fashion-MNIST dataset. As n increases, the stability of the subnetwork f_6 (orange) is similar to the stability of the whole network (green) and the stable rank of W_6 also converges to 1.

Likewise, the sign stability was assessed by the occurrences of identical labels.

$$\text{Sign stability} := \frac{1}{4}\sum_{i=2}^{5}\sum_{\mathbf{x}\in\text{TEST}} |\text{sign}(f_k(\mathbf{x};\,\theta^{(1)})) - \text{sign}(f_k(\mathbf{x};\,\theta^{(i)}))|.$$

Computing resource Our experiments require a few minutes to 2 hours of training on a GPU (RTX 3070 8GB) depending on the number of data points. All experiments require less than 3 GB of memory. Experiments with a few data points were trained on CPU cluster which contains the following CPUs: Intel(R) Core(TM) i5-7500, i7-9700K, i7-8700; and Intel(R) Xeon(R) Silver 4214R, Gold 5220R, Silver 4310, Gold 6226R, E5-2650 v2, E5-2660 v3, E5-2640 v4, Gold 5120, Gold 6132.

E.1 Additional experiments

In Figs. 4 and 5, we repeat the experiments for Figs. 2 and 3 for Fashion-MNIST and obtain equivalent results. In Fig. 6, we plot the performance of the sub-networks in Fig. 4.

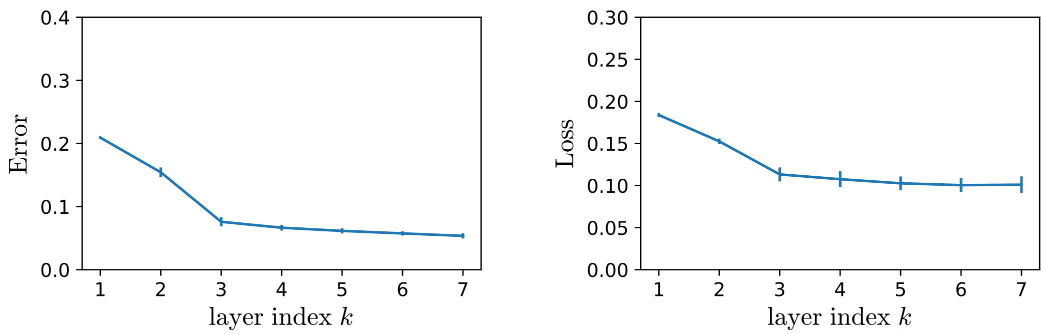

Figure 6: **Layer compression.** Using the models trained for Fig. 2, we send the output of the k^{th} layer to an auxiliary linear layer, which is trained using Adam over 200 epochs. We measure the test error **(left)** and the test loss **(right)**, which resembles the stability in Fig. 2.