# Landscape Connectivity and Dropout Stability of SGD Solutions for Over-parameterized Neural Networks

Alexander Shevchenko [1] Marco Mondelli [1]

## Abstract

The optimization of multilayer neural networks typically leads to a solution with zero training error, yet the landscape can exhibit spurious local minima and the minima can be disconnected. In this paper, we shed light on this phenomenon: we show that the combination of stochastic gradient descent (SGD) and over-parameterization makes the landscape of multilayer neural networks approximately connected and thus more favorable to optimization. More specifically, we prove that SGD solutions are connected via a piecewise linear path, and the increase in loss along this path vanishes as the number of neurons grows large. This result is a consequence of the fact that the parameters found by SGD are increasingly dropout stable as the network becomes wider. We show that, if we remove part of the neurons (and suitably rescale the remaining ones), the change in loss is independent of the total number of neurons, and it depends only on how many neurons are left. Our results exhibit a mild dependence on the input dimension: they are dimension-free for two-layer networks and require the number of neurons to scale linearly with the dimension for multilayer networks. We validate our theoretical findings with numerical experiments for different architectures and classification tasks.

## 1. Introduction

The recent successes of deep learning have two elements in common: *(i)* a local search algorithm, e.g., stochastic gradient descent (SGD), and *(ii)* an over-parameterized neural network. Even though the training problem can have several local minima (Auer et al., 1996) and is NP-hard in the worst case (Blum & Rivest, 1989), the optimization of an over-parameterized network via SGD typically leads to a solution that has small training error and generalizes well. This fact has led to a focus on the theoretical understanding of neural networks' optimization landscape (see, e.g., (Livni et al., 2014; Dauphin et al., 2014; Safran & Shamir, 2016; Pennington & Bahri, 2017) and the discussion in Section 2). However, most of the existing results either make strong assumptions on the model or do not provide a satisfactory scaling with respect to the parameters of the problem.

From the empirical viewpoint, it has been observed that, if we connect two minima of SGD with a line segment, the loss is large along this path (Goodfellow et al., 2015; Keskar et al., 2017). However, if the path is chosen in a more sophisticated way, one can connect the minima found by SGD via a piecewise linear path where the loss is approximately constant (Garipov et al., 2018; Draxler et al., 2018). These findings suggest that the minima of SGD are not isolated points in parameter space, but rather they are approximately connected. In the recent paper (Kuditipudi et al., 2019), mode connectivity of multilayer ReLU networks is proved by assuming generic properties of well-trained networks, i.e., dropout stability and noise stability.

In this work, we consider multilayer neural networks trained by one-pass (or online) SGD with the square loss. We show that, as the number of neurons increases, *(i)* the neural network becomes increasingly dropout stable, and *(ii)* the optimization landscape becomes increasingly connected between SGD solutions. We establish quantitative bounds on how much the loss changes after the dropout procedure and along the path connecting two SGD solutions, and we relate this change in loss to the total number of neurons, the size of the dropout pattern, and the input dimension. By doing so, we give a theoretical justification to the empirical observation that the barriers between local minima tend to disappear as the neural network becomes larger (Draxler et al., 2018). More specifically, our main contributions can be summarized as follows:

**Two-layer networks.** We consider the training of a two-layer neural network $\hat{y}(\boldsymbol{x}) = \frac{1}{N}\boldsymbol{a}^{\mathsf{T}}\sigma(\boldsymbol{W}\boldsymbol{x})$ with $N$ neurons. First, we study the dropout stability of SGD solutions, namely, we bound the change in loss when $N - M$ neurons are removed from the trained network and $M$ remaining

[1] Institute of Science and Technology, Austria. Correspondence to: Alexander Shevchenko <ashevche@ist.ac.at>.

*Proceedings of the $37^{th}$ International Conference on Machine Learning*, Online, PMLR 119, 2020. Copyright 2020 by the author(s).

neurons are suitably rescaled: *we show that the change in loss scales at most as $\sqrt{\log M/M}$, and therefore it does not depend on the number of neurons $N$ of the original network or on the dimension $d$ of the input*. Then, we characterize the landscape connectivity for the parameters obtained via SGD: *we show that pairs of SGD solutions are connected via a piecewise linear path, and the loss along this path is no larger than the loss at the extremes plus a term that scales as $\sqrt{\log N/N}$*. Let us emphasize that the two solutions of SGD are obtained by running the algorithm on different samples (from the same data distribution), for different initializations, and for the different number of iterations.

**Multilayer networks.** We consider the training of a general model of deep neural network with $L+1 \geq 4$ layers, where each hidden layer contains $N$ neurons. This model includes as a special case $\hat{\boldsymbol{y}}(\boldsymbol{x})$ which is equal to

$$\frac{1}{N}\boldsymbol{W}_{L+1}\sigma_L\left(\cdots\left(\frac{1}{N}\boldsymbol{W}_2\sigma_1\left(\boldsymbol{W}_1\boldsymbol{x}\right)\right)\cdots\right) \tag{1.1}$$

Our results are similar to those for two-layer networks: *(i) if we keep at least $M$ neurons in each layer, the change in loss scales at most as $\sqrt{(d+\log M)/M}$; (ii) pairs of SGD solutions are connected via a piecewise linear path, along which the loss does not increase more than $\sqrt{(d+\log N)/N}$*. In contrast with the two-layer case, these bounds are not dimension-free. However, the dependence on the input dimension $d$ is only linear, since the loss change vanishes as soon as $M, N \gg d$. We assume that, during SGD training, the parameters of the first and last layer are kept fixed, and they are regarded as random features (Rahimi & Recht, 2008). We believe that this assumption, as well as the requirement of having at least 4 layers, can be removed with an improved analysis.

The proofs of dropout stability build on recent results concerning the mean-field description of the SGD dynamics (Mei et al., 2019; Araújo et al., 2019), see also the discussion in Section 2. The proofs of landscape connectivity use ideas from (Kuditipudi et al., 2019).

**Organization of the paper.** In Section 2, we succinctly review related work. In Section 3, we present our rigorous results for two-layer networks: we first assume that the activation function $\sigma$ is bounded, and then we provide an extension to unbounded activations. In Section 4, we present our results for multilayer networks. In Section 5, we validate our findings with numerical experiments on fully-connected neural networks trained on MNIST and CIFAR-10 datasets. Finally, in Section 6 we discuss additional connections to the literature and give directions for future work. All the proofs are deferred to the appendices in the supplementary material, which also contain additional numerical results.

**Notation.** We use bold symbols for vectors $\boldsymbol{a}, \boldsymbol{b}$, and capitalized bold symbols for matrices $\boldsymbol{A}, \boldsymbol{B}$. We denote by $\|\boldsymbol{a}\|_2$ the norm of $\boldsymbol{a}$, by $\|\boldsymbol{A}\|_{\rm op}$ the operator norm of $\boldsymbol{A}$, by $\langle \boldsymbol{a}, \boldsymbol{b}\rangle$ the scalar product of $\boldsymbol{a}, \boldsymbol{b}$, and by $\boldsymbol{a} \odot \boldsymbol{b}$ the Hadamard (or entrywise) product of $\boldsymbol{a}, \boldsymbol{b}$. Given an integer $N$ and a real number $r \geq 1$, we set $[N] = \{1, \dots, N\}$ and $[r] = \{1, \dots, \lfloor r \rfloor\}$. Given a discrete set $\mathcal{A}$, we denote by $|\mathcal{A}|$ its cardinality.

## 2. Related Work

The landscape of several non-convex optimization problems has been studied in recent years, including empirical risk minimization (Mei et al., 2018a), low rank matrix problems (Ge et al., 2017), matrix completion (Ge et al., 2016), and semi-definite programs (Boumal et al., 2016). Motivated by the extraordinary success of deep learning, a growing literature is focusing on the loss surfaces of neural networks. Under strong assumptions, in (Choromanska et al., 2015) the loss function is related to a spin glass and it is shown that local minima are located in a well-defined band. It has been shown that local minima are globally optimal in various settings: deep linear networks (Kawaguchi, 2016); fully connected and convolutional neural networks with a wide layer containing more neurons than training samples (Nguyen & Hein, 2017; 2018); deep networks with more neurons than training samples and skip connections (Nguyen et al., 2019). Furthermore, if one of the layers is sufficiently wide, in (Nguyen, 2019b) it is shown that sublevel sets are connected. Similar results are proved for binary classification in (Liang et al., 2018a;b). In (Freeman & Bruna, 2017), a two-layer neural networks with ReLU activations is considered, and it is shown that the landscape becomes approximately connected as the number of neurons increases. However, the energy gap scales exponentially with the input dimension. In (Venturi et al., 2019), it is shown that there are no spurious valleys when the number of neurons is larger than the intrinsic dimension of the networks. However, for many standard architectures, the intrinsic dimension of the network is infinite.

In this paper, we take a different view and relate the problem to a recent line of work, which shows that the behavior of neural networks trained by SGD tends to a mean field limit, as the number of neurons grows. This phenomenon has been first studied in two-layer neural networks in (Mei et al., 2018b; Rotskoff & Vanden-Eijnden, 2018; Chizat & Bach, 2018; Sirignano & Spiliopoulos, 2018). In particular, in (Mei et al., 2018b), it is shown that the SGD dynamics is well approximated by a Wasserstein gradient flow, given that the number of neurons exceeds the data dimension. Improved and dimension-free bounds are provided in (Mei et al., 2019). Convergence to the global optimum is proved for noisy SGD in (Mei et al., 2018b; Chizat & Bach, 2018), without any explicit rate. A convergence rate which is exponential and dimension-free is proved in (Javanmard et al.,

2019) by exploiting the displacement convexity of the limit dynamics. An argument indicating convergence in a time polynomial in the dimension is provided in (Wei et al., 2018), but for a different type of continuous flow. Fluctuations around the mean field limit are also studied in (Rotskoff & Vanden-Eijnden, 2018; Sirignano & Spiliopoulos, 2019a). The multilayer case is tackled in (Nguyen, 2019a; Sirignano & Spiliopoulos, 2019b; Araújo et al., 2019; Nguyen & Pham, 2020a). In (Sirignano & Spiliopoulos, 2019b), it is considered a (less natural) model where the number of neurons grows one layer at a time. In (Nguyen, 2019a), a formalism is developed to describe the mean field limit, but the results are not rigorous. Rigorous bounds between the SGD dynamics and a limit stochastic process are established in (Araújo et al., 2019), where it is assumed that the first and last layer are not trained to simplify the analysis. A different approach based on the concept of neuronal embedding is put forward in (Nguyen & Pham, 2020a). In (Nguyen & Pham, 2020a), it is also provided a convergence result for three-layer networks, later generalized in the companion note (Nguyen & Pham, 2020b).

In a nutshell, existing mean-field analyses show that the dynamics of SGD is close to a limit stochastic process. However, the consequences of this fact remain largely unexplored, since the limit process is hard to analyze. In this work, we advance the mean-field theory of neural networks, and we provide the first theoretical guarantees on two phenomena widely observed in practice: dropout stability and mode connectivity of SGD solutions.

We remark that the mean-field regime considered in this paper is different from the "lazy training" regime that has recently received a lot of attention (Allen-Zhu et al., 2019a;b; Chizat et al., 2019; Du et al., 2018; 2019; Jacot et al., 2018; Li & Liang, 2018; Zou et al., 2018). In fact, in order to prove convergence of gradient descent in the lazy regime, it is crucially exploited that the parameters stay bounded in a certain region. On the contrary, in the mean field regime, the scaling of the gradient (see Eqs. (3.3) and (4.3)) ensures that the parameters move away from the initialization. The connection between the mean-field and the lazy regime is investigated in Section 4 of (Mei et al., 2019) and in the recent paper (Chen et al., 2020). We highlight that neural networks trained in the mean-field regime achieve results comparable to the state of the art for standard datasets, as demonstrated in the numerical results of Section 5.

## 3. Dropout Stability and Connectivity for Two-Layer Networks

### 3.1. Setup

We consider a two-layer neural network with $N$ neurons:

$$\hat{y}_N(\boldsymbol{x}, \boldsymbol{\theta}) = \tfrac{1}{N}\textstyle\sum_{i=1}^{N} a_i \sigma(\boldsymbol{x}, \boldsymbol{w}_i), \tag{3.1}$$

where $\boldsymbol{x} \in \mathbb{R}^d$ is a feature vector, $\hat{y}_N(\boldsymbol{x}, \boldsymbol{\theta}) \in \mathbb{R}$ is the output of the network, $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_N)$, with $\boldsymbol{\theta}_i = (a_i, \boldsymbol{w}_i) \in \mathbb{R}^{D+1}$, are the parameters of the network and $\sigma : \mathbb{R}^d \times \mathbb{R}^D \to \mathbb{R}$ is an activation function.

A typical example is $\sigma(\boldsymbol{x}, \boldsymbol{w}) = \sigma(\langle \boldsymbol{x}, \boldsymbol{w}\rangle)$, for a scalar function $\sigma : \mathbb{R} \to \mathbb{R}$. In order to incorporate a bias term in the hidden layer, one can simply add the feature 1 to $\boldsymbol{x}$ and adjust the shape of the parameters $\boldsymbol{w}_i$ accordingly. We are interested in minimizing the expected square loss (also known as population risk):

$$L_N(\boldsymbol{\theta}) = \mathbb{E}\Big\{\big(y - \hat{y}_N(\boldsymbol{x}, \boldsymbol{\theta})\big)^2\Big\}, \tag{3.2}$$

where the expectation is taken over $(\boldsymbol{x}, y) \sim \mathbb{P}$. To do so, we are given data $(\boldsymbol{x}_k, y_k)_{k\geq 0} \overset{\text{i.i.d.}}{\sim} \mathbb{P}$, and we learn the parameters of the network via stochastic gradient descent (SGD) with step size $s_k$:

$$\begin{aligned} \boldsymbol{\theta}_i^{k+1} &= \boldsymbol{\theta}_i^k - s_k N \cdot \mathrm{Grad}_i(\boldsymbol{\theta}^k), \\ \mathrm{Grad}_i(\boldsymbol{\theta}^k) &= \nabla_{\boldsymbol{\theta}_i}\big(y_k - \hat{y}_N(\boldsymbol{x}_k, \boldsymbol{\theta}^k)\big)^2, \end{aligned} \tag{3.3}$$

where $\boldsymbol{\theta}^k$ denotes the parameters after $k$ steps of SGD, and the parameters are initialized independently according to the distribution $\rho_0$. We consider a one-pass (or online) model, where each data point is used only once.

Given a neural network with parameters $\boldsymbol{\theta}$ and a subset $\mathcal{A}$ of $[N]$, the dropout network with parameters $\boldsymbol{\theta}_{\mathrm{S}}$ is obtained by setting to 0 the outputs of the neurons indexed by $[N]\setminus\mathcal{A}$ and by suitably rescaling the remaining outputs. Denote by $\hat{y}_{|\mathcal{A}|}(\boldsymbol{x}, \boldsymbol{\theta}_{\mathrm{S}})$ and $L_{|\mathcal{A}|}(\boldsymbol{\theta}_{\mathrm{S}})$ the output of the dropout network and its expected square loss, respectively. In formulas,

$$\begin{aligned} \hat{y}_{|\mathcal{A}|}(\boldsymbol{x}, \boldsymbol{\theta}_{\mathrm{S}}) &= \frac{1}{|\mathcal{A}|}\sum_{i\in\mathcal{A}} a_i \sigma(\boldsymbol{x}, \boldsymbol{w}_i), \\ L_{|\mathcal{A}|}(\boldsymbol{\theta}_{\mathrm{S}}) &= \mathbb{E}\left\{\big(y - \hat{y}_{|\mathcal{A}|}(\boldsymbol{x}, \boldsymbol{\theta}_{\mathrm{S}})\big)^2\right\}. \end{aligned} \tag{3.4}$$

Let us compare the original network (3.1) with the dropout network (3.4): $\boldsymbol{w}_i$ does not change, $a_i$ is rescaled by $|\mathcal{A}|/|N|$ and in (3.4) we sum over $|\mathcal{A}|$ neurons (while in (3.1) the sum is over $N$ neurons). This is equivalent to setting $|N| - |\mathcal{A}|$ neurons to zero and rescaling the others by a factor, as in (Kuditipudi et al., 2019).

We now define the notions of dropout stability and connectivity for network parameters.

**Definition 3.1** (Dropout stability)**.** *Given $\mathcal{A} \subseteq [N]$, we say that $\boldsymbol{\theta}$ is $\varepsilon_{\mathrm{D}}$-dropout stable if*

$$|L_N(\boldsymbol{\theta}) - L_{|\mathcal{A}|}(\boldsymbol{\theta}_{\mathrm{S}})| \leq \varepsilon_{\mathrm{D}}. \tag{3.5}$$

**Definition 3.2** (Connectivity)**.** *We say that two parameters $\boldsymbol{\theta}$ and $\boldsymbol{\theta}'$ are $\varepsilon_{\mathrm{C}}$-connected if there exists a continuous path* *in parameter space $\pi : [0, 1] \to \mathbb{R}^{D\times N}$, such that $\pi(0) = \boldsymbol{\theta}$ and $\pi(1) = \boldsymbol{\theta}'$ with*

$$L_N(\pi(t)) \leq \max(L_N(\boldsymbol{\theta}), L_N(\boldsymbol{\theta}')) + \varepsilon_{\rm C}. \tag{3.6}$$

### 3.2. Results for Bounded Activations

We make the following assumptions on the learning rate $s_k$, the data distribution $(\boldsymbol{x}, y) \sim \mathbb{P}$, the activation function $\sigma$, and the initialization $\rho_0$:

**(A1)** $s_k = \alpha\xi(k\alpha)$, where $\xi : \mathbb{R}_{\geq 0} \to \mathbb{R}_{>0}$ is bounded by $K_1$ and $K_1$-Lipschitz.
**(A2)** The response variables $y$ are bounded by $K_2$ and the gradient $\nabla_{\boldsymbol{w}}\sigma(\boldsymbol{x}, \boldsymbol{w})$ is $K_2$ sub-gaussian when $\boldsymbol{x} \sim \mathbb{P}$.
**(A3)** The activation function $\sigma$ is bounded by $K_3$ and differentiable, with gradient bounded by $K_3$ and $K_3$-Lipschitz.
**(A4)** The initialization $\rho_0$ is supported on $|a_i^0| \leq K_4$.

We are now ready to present our results, which are proved in Appendix A in the supplementary material.

**Theorem 1** (Two-layer)**.** *Assume that conditions **(A1)-(A4)** hold, and fix $T \geq 1$. Let $\boldsymbol{\theta}^k$ be obtained by running $k$ steps of the SGD algorithm (3.3) with data $\{(\boldsymbol{x}_j, y_j)\}_{j=0}^k \overset{\text{i.i.d.}}{\sim} \mathbb{P}$ and initialization $\rho_0$. Then, the following results hold:*

***(A)** Pick $\mathcal{A} \subseteq [N]$ independent of $\boldsymbol{\theta}^k$. Then, with probability at least $1-e^{-z^2}$, for all $k \in [T/\alpha]$, $\boldsymbol{\theta}^k$ is $\varepsilon_{\rm D}$-dropout stable with $\varepsilon_{\rm D}$ equal to*

$$Ke^{KT^3}\left(\frac{\sqrt{\log|\mathcal{A}|}+z}{\sqrt{|\mathcal{A}|}} + \sqrt{\alpha}\big(\sqrt{D+\log N}+z\big)\right), \tag{3.7}$$

*where the constant $K$ depends only on the constants $K_i$ of the assumptions.*

***(B)** Fix $T' \geq 1$ and let $(\boldsymbol{\theta}')^{k'}$ be obtained by running $k'$ steps of SGD with data $\{(\boldsymbol{x}'_j, y'_j)\}_{j=0}^{k'} \overset{\text{i.i.d.}}{\sim} \mathbb{P}$ and initialization $\rho'_0$ that satisfies **(A4)**. Then, with probability at least $1-e^{-z^2}$, for all $k \in [T/\alpha]$ and $k' \in [T'/\alpha]$, $\boldsymbol{\theta}^k$ and $(\boldsymbol{\theta}')^{k'}$ are $\varepsilon_{\rm C}$-connected with $\varepsilon_{\rm C}$ equal to*

$$Ke^{KT_{\max}^3}\left(\frac{\sqrt{\log N}+z}{\sqrt{N}} + \sqrt{\alpha}\big(\sqrt{D+\log N}+z\big)\right), \tag{3.8}$$

*where $T_{\max} = \max(T, T')$. Furthermore, the path connecting $\boldsymbol{\theta}^k$ with $(\boldsymbol{\theta}')^{k'}$ consists of 7 line segments.*

The result (A) characterizes the change in loss when only $|\mathcal{A}|$ neurons remain in the network. In particular, the change in loss scales as $\sqrt{\log|\mathcal{A}|/|\mathcal{A}|} + \sqrt{\alpha(D + \log N)}$, where $N$ is the total number of neurons, $D$ is the dimension of the neurons and $\alpha$ is the step size of SGD. This quantity vanishes as long as $|\mathcal{A}| \gg 1$ and $\alpha \ll 1/(D + \log N)$. Note that the number of training samples $k$ is such that $k\alpha$ is a constant. Thus, the condition $\alpha \ll 1/(D + \log N)$ implies that $k$ needs to scale only logarithmically with $N$. Furthermore, the condition $|\mathcal{A}| \gg 1$ implies that $|\mathcal{A}|$ does not need to scale with $N$, $D$. The proof builds on the machinery developed in (Mei et al., 2019) to provide a mean-field approximation to the dynamics of SGD. In (Mei et al., 2019), it is shown that, as $N \to \infty$ and $\alpha \to 0$, the parameters $\boldsymbol{\theta}^k$ obtained by running $k$ steps of SGD with step size $\alpha$ are close to $N$ i.i.d. particles that evolve according to a nonlinear dynamics at time $k\alpha$. Here, the idea is to show that *(i)* the parameters $\boldsymbol{\theta}_{\rm S}^k$ are also close to $|\mathcal{A}|$ such i.i.d. particles, and *(ii)* the quantities $L_N(\boldsymbol{\theta}^k)$ and $L_{|\mathcal{A}|}(\boldsymbol{\theta}_{\rm S}^k)$ concentrate to the same limit value, which represents the limit loss of the nonlinear dynamics.

The result (B) shows that we can connect two different solutions of SGD via a simple path. Note that the two solutions can be obtained by running SGD for the different number of iterations ($k' \neq k$), for different training datasets ($(\boldsymbol{x}_j, y_j) \neq (\boldsymbol{x}'_j, y'_j)$) and for different initializations of SGD ($\rho_0 \neq \rho'_0$). The proof uses ideas from (Kuditipudi et al., 2019). In that work, the authors consider a multilayer neural network with ReLU activations and show how to find a piecewise linear path between two solutions that are dropout stable with $|\mathcal{A}| = N/2$. In fact, $\varepsilon_{\rm C}$ has a similar scaling to $\varepsilon_{\rm D}$ after setting $|\mathcal{A}| = N/2$. We are also able to show (and, consequently, exploit) a more general notion of dropout stability for the trained network. In fact, (Kuditipudi et al., 2019) requires the existence of a single dropout pattern, while here we give a bound for any fixed dropout pattern (as long as it does not depend on SGD).

The bounds in Theorem 1 exhibit an exponential dependence on $T$. We remark that, in the mean-field regime, the number of samples $k$ is large, the step size $\alpha$ is small, and $T = k\alpha$ is a constant. In fact, $T$ is the evolution time of the limit stochastic process (which does not depend on $N$, $\alpha$). Empirically, the value of $T$ needed to achieve good accuracy is quite small: $T = 1$ gives $< 16\%$ error on CIFAR-10, see Section 5. The exponential dependence on $T$ is common to all existing mean-field analyses, and improving it is an open question. The assumptions on the learning rate, the data distribution and the initialization are mild and only require some regularity. The assumptions on the activation function are fulfilled in several practical settings: $\sigma(\boldsymbol{x}, \boldsymbol{w}) = \sigma(\langle\boldsymbol{x}, \boldsymbol{w}\rangle)$, where $\sigma : \mathbb{R} \to \mathbb{R}$ is, e.g., the sigmoid or the hyperbolic tangent.

### 3.3. Extension to Unbounded Activations

Note that Theorem 1 requires that the activation function is bounded. We can relax this assumption, at the cost of a less tight dependence on the time $T$ of the evolution. In particular, assume further that *(i)* the feature vectors $\boldsymbol{x}$ and the initialization $\rho_0$ are bounded, and that *(ii)* the loss at each step of SGD is uniformly bounded, i.e., $\max_j |y_j - \hat{y}_N(\boldsymbol{x}_j, \boldsymbol{\theta}^j)| \leq K_5$. This last requirement is reasonable, since the objective of SGD is to minimize such a loss. Then, the results of Theorem 1 hold also for unbounded $\sigma$, where the term $Ke^{KT^3}$ is replaced by a generic $K(T)$, which depends on $T$ and on the constants $K_i$ of the assumptions. The simulation results of Section 5 show that such a dependence on $T$ is mild in practical settings.

The formal statement and the proof of this result is contained in Appendix B in the supplementary material. The idea is to show that, if the parameters of the neural network are initialized with a bounded distribution, then they stay bounded for any finite time $T$ of the SGD evolution. Thus, the SGD evolution does not change if we substitute the unbounded activation function with a bounded one, and we can apply the results for bounded $\sigma$.

## 4. Dropout Stability and Connectivity for Multilayer Networks

### 4.1. Setup

We consider a neural network with $L + 1 \geq 4$ layers, where each hidden layer contains $N$ neurons. Given the input feature vector $\boldsymbol{x} \in \mathbb{R}^{d_0}$, the first layer activations $\boldsymbol{z}_{i_1}^{(1)}$ for $i_1 \in [N]$ have form

$$\sigma^{(0)}\left(\boldsymbol{x}, \boldsymbol{\theta}_{i_1}^{(0)}\right), \quad \boldsymbol{\theta}_{i_1}^{(0)} \in \mathbb{R}^{D_0}$$

the intermediate layer $\ell \in [L-1]$ activations $\boldsymbol{z}_{i_{\ell+1}}^{(\ell+1)}(\boldsymbol{x}, \boldsymbol{\theta})$ for $i_{\ell+1} \in [N]$ are defined as follows

$$\frac{1}{N}\sum_{i_\ell=1}^{N} \boldsymbol{a}_{i_\ell, i_{\ell+1}}^{(\ell)} \odot \sigma^{(\ell)}\left(\boldsymbol{z}_{i_\ell}^{(\ell)}(\boldsymbol{x}, \boldsymbol{\theta}), \boldsymbol{w}_{i_\ell, i_{\ell+1}}^{(\ell)}\right),$$
$$\boldsymbol{\theta}_{i_\ell, i_{\ell+1}}^{(\ell)} = (\boldsymbol{a}_{i_\ell, i_{\ell+1}}^{(\ell)}, \boldsymbol{w}_{i_\ell, i_{\ell+1}}^{(\ell)}) \in \mathbb{R}^{D_\ell + d_{\ell+1}},$$

and the output of network is given by

$$\widehat{\boldsymbol{y}}_N(\boldsymbol{x}, \boldsymbol{\theta}) = \frac{1}{N}\sum_{i_L=1}^{N} \boldsymbol{a}_{i_L}^{(L)} \odot \sigma^{(L)}\left(\boldsymbol{z}_{i_L}^{(L)}(\boldsymbol{x}, \boldsymbol{\theta}), \boldsymbol{w}_{i_L}^{(L)}\right),$$
$$\boldsymbol{\theta}_{i_L}^{(L)} = (\boldsymbol{a}_{i_L}^{(L)}, \boldsymbol{w}_{i_L}^{(L)}) \in \mathbb{R}^{D_L + d_{L+1}}, \quad i_L \in [N]. \tag{4.1}$$

Here, $\sigma^{(\ell)} : \mathbb{R}^{d_\ell} \times \mathbb{R}^{D_\ell} \to \mathbb{R}^{d_{\ell+1}}$ ($\ell \in \{0, \dots, L\}$) are the activation functions, and $\boldsymbol{\theta}$ contains the parameters of the network, which are $\boldsymbol{\theta}_{i_1}^{(0)}$, $\boldsymbol{\theta}_{i_\ell, i_{\ell+1}}^{(\ell)}$ and $\boldsymbol{\theta}_{i_L}^{(L)}$.

Note that (4.1) includes the model (1.1) as a special case. To see this, consider the following setting: pick $D_0 = d_0$ and stack the parameters $\boldsymbol{\theta}_{i_1}^{(0)} \in \mathbb{R}^{d_0}$ into the rows of the matrix $\boldsymbol{W}_1 \in \mathbb{R}^{N \times d_0}$; for $i \in [L-1]$, pick $D_\ell = 1$ and stack the scalar parameters $\boldsymbol{a}_{i_\ell, i_{\ell+1}}^{(\ell)} \in \mathbb{R}$ into the matrix $\boldsymbol{W}_{\ell+1} \in \mathbb{R}^{N \times N}$; pick $D_L = d_{L+1}$ and stack the parameters $\boldsymbol{a}_{i_L}^{(L)} \in \mathbb{R}^{d_{L+1}}$ into the columns of the matrix $\boldsymbol{W}_{L+1} \in \mathbb{R}^{d_{L+1} \times N}$; finally, assume that the activation function $\sigma^{(\ell)}$ does not depend on $\boldsymbol{w}_{i_\ell, i_{\ell+1}}^{(\ell)}$ for $\ell \in [L-1]$ and that $\sigma^{(L)}$ does not depend on $\boldsymbol{w}_{i_L}^{(L)}$. Then, in this setting, (4.1) can be reduced to (1.1).

We are interested in minimizing the expected square loss:

$$L_N(\boldsymbol{\theta}) = \mathbb{E}\left\{\left\|\boldsymbol{y} - \widehat{\boldsymbol{y}}_N(\boldsymbol{x}, \boldsymbol{\theta})\right\|_2^2\right\}, \tag{4.2}$$

where the expectation is taken over $(\boldsymbol{x}, \boldsymbol{y}) \sim \mathbb{P}$. To do so, we are given data $(\boldsymbol{x}_k, \boldsymbol{y}_k)_{k \geq 0} \overset{\text{i.i.d.}}{\sim} \mathbb{P}$, we run SGD with step size $s_k$ for the intermediate layers $\ell \in [L-1]$, and we fix first and last layer:

$$\boldsymbol{\theta}_{i_\ell, i_{\ell+1}}^{(\ell)}(k+1) = \boldsymbol{\theta}_{i_\ell, i_{\ell+1}}^{(\ell)}(k) - s_k N^2 \text{Grad}_{i_\ell, i_{\ell+1}}^{(\ell)}(\boldsymbol{\theta}(k)),$$
$$\text{Grad}_{i_\ell, i_{\ell+1}}^{(\ell)}(\boldsymbol{\theta}(k)) = \nabla_{\boldsymbol{\theta}_{i_\ell, i_{\ell+1}}^{(\ell)}}\left\|\boldsymbol{y}_k - \widehat{\boldsymbol{y}}_N(\boldsymbol{x}_k, \boldsymbol{\theta}(k))\right\|_2^2,$$
$$\boldsymbol{\theta}_{i_1}^{(0)}(k+1) = \boldsymbol{\theta}_{i_1}^{(0)}(k), \quad \boldsymbol{\theta}_{i_L}^{(L)}(k+1) = \boldsymbol{\theta}_{i_L}^{(L)}(k), \tag{4.3}$$

where $\boldsymbol{\theta}(k)$ contains the parameters of the network after $k$ steps of SGD. As in the two-layer setting, we consider a one-pass model and the parameters are initialized independently, i.e., $\{\boldsymbol{\theta}_{i_1}^{(0)}(0)\}_{i_1 \in [N]} \overset{\text{i.i.d.}}{\sim} \rho_0^{(0)}$, $\{\boldsymbol{\theta}_{i_\ell, i_{\ell+1}}^{(\ell)}(0)\}_{i_\ell, i_{\ell+1} \in [N]} \overset{\text{i.i.d.}}{\sim} \rho_0^{(\ell)}$, for $\ell \in [L-1]$, and $\{\boldsymbol{\theta}_{i_L}^{(L)}(0)\}_{i_L \in [N]} \overset{\text{i.i.d.}}{\sim} \rho_0^{(L)}$.

The gradients of $\widehat{\boldsymbol{y}}_N$ with respect to the parameters of the network can be computed via backpropagation (Goodfellow et al., 2016). By doing so (see Araújo et al. (2019, Section 3.3)), we obtain that $\boldsymbol{\theta}_{i_\ell, i_{\ell+1}}^{(\ell)}$ evolves at a time scale of $1/N^2$. Thus, we multiply the step size $s_k$ in (4.3) with the factor $N^2$ in order to avoid falling into the "lazy training" regime. In lazy training, the parameters hardly vary but the method still converges to zero training loss, and this regime has received a lot of attention recently (Jacot et al., 2018; Li & Liang, 2018; Zou et al., 2018; Du et al., 2018; 2019; Allen-Zhu et al., 2019b;a; Chizat et al., 2019). Let us emphasize that the SGD scalings in (3.3) and (4.3) imply that the parameters move as long as the product of the number of iterations with the step size is non-vanishing.

Note also that the parameters of layers $\ell = 0$ and $\ell = L$, i.e., $\{\boldsymbol{\theta}_{i_1}^{(0)}\}_{i_1 \in [N]}$ and $\{\boldsymbol{\theta}_{i_L}^{(L)}\}_{i_L \in [N]}$, stay fixed to their initial values. This is done for technical reasons. In fact, by computing the backpropagation equations, one obtains that $\boldsymbol{\theta}_{i_1}^{(0)}$ and $\boldsymbol{\theta}_{i_L}^{(L)}$ evolve at a time scale of $1/N$, which makes it challenging to analyze their trajectories. We regard the parameters $\boldsymbol{\theta}_{i_1}^{(0)}$ and $\boldsymbol{\theta}_{i_L}^{(L)}$ as random features (Rahimi & Recht, 2008) close to the input and the output.

Given a neural network with parameters $\boldsymbol{\theta}$ and subsets $\mathcal{A}_1, \dots, \mathcal{A}_L$ of $[N]$, the dropout network with parameters

$\boldsymbol{\theta}_{\mathrm{S}}$ is obtained by setting to 0 the outputs of the neurons indexed by $[N] \setminus \mathcal{A}_i$ at layer $i$ and by suitably rescaling the remaining outputs. With an abuse of notation, denote by $\widehat{\boldsymbol{y}}_{|\mathcal{A}|}\left(\boldsymbol{x}, \boldsymbol{\theta}_{\mathrm{S}}\right)$ and $L_{|\mathcal{A}|}(\boldsymbol{\theta}_{\mathrm{S}})$ the output of the dropout network and its expected square loss, respectively. In formulas, the dropout version of activations $\boldsymbol{z}_{i_{\ell+1}}^{(\ell+1)}\left(\boldsymbol{x}, \boldsymbol{\theta}_{\mathrm{S}}\right)$ of layer $\ell \in [L-1]$ for $i_{\ell+1} \in \mathcal{A}_{\ell+1}$ are given by

$$\frac{1}{|\mathcal{A}_\ell|} \sum_{i_\ell \in \mathcal{A}_\ell} \boldsymbol{a}_{i_\ell, i_{\ell+1}}^{(\ell)} \odot \sigma^{(\ell)}\left(\boldsymbol{z}_{i_\ell}^{(\ell)}\left(\boldsymbol{x}, \boldsymbol{\theta}_{\mathrm{S}}\right), \boldsymbol{w}_{i_\ell, i_{\ell+1}}^{(\ell)}\right),$$

the output of dropout network $\widehat{\boldsymbol{y}}_{|\mathcal{A}|}\left(\boldsymbol{x}, \boldsymbol{\theta}_{\mathrm{S}}\right)$ takes the form

$$\frac{1}{|\mathcal{A}_L|} \sum_{i_L \in \mathcal{A}_L} \boldsymbol{a}_{i_L}^{(L)} \odot \sigma^{(L)}\left(\boldsymbol{z}_{i_L}^{(L)}\left(\boldsymbol{x}, \boldsymbol{\theta}_{\mathrm{S}}\right), \boldsymbol{w}_{i_L}^{(L)}\right),$$

and, consequently, the expected square loss is defined by

$$L_{|\mathcal{A}|}(\boldsymbol{\theta}_{\mathrm{S}}) = \mathbb{E}\left\{\left\|\boldsymbol{y} - \widehat{\boldsymbol{y}}_{|\mathcal{A}|}\left(\boldsymbol{x}, \boldsymbol{\theta}_{\mathrm{S}}\right)\right\|_2^2\right\},$$

where $\boldsymbol{z}_{i_1}^{(1)}\left(\boldsymbol{x}, \boldsymbol{\theta}_{\mathrm{S}}\right) = \boldsymbol{z}_{i_1}^{(1)}\left(\boldsymbol{x}, \boldsymbol{\theta}\right)$ for $i_1 \in \mathcal{A}_1$. The definitions of dropout stability and connectivity are analogous to those for two-layer networks: *(i)* $\boldsymbol{\theta}$ is $\varepsilon_{\mathrm{D}}$-dropout stable if (3.5) holds; and *(ii)* $\boldsymbol{\theta}$ and $\boldsymbol{\theta}'$ are $\varepsilon_{\mathrm{C}}$-connected if they are connected by a continuous path in parameter space such that (3.6) holds.

### 4.2. Results

We make the following assumptions on the learning rate $s_k$, the data distribution $(\boldsymbol{x}, \boldsymbol{y}) \sim \mathbb{P}$, the activation functions $\sigma^{(\ell)}$, and the initializations $\rho_0^{(\ell)}$:

**(B1)** $s_k = \alpha\xi(k\alpha)$, where $\xi : \mathbb{R}_{\geq 0} \to \mathbb{R}_{>0}$ is bounded by $K_1$ and $K_1$-Lipschitz.
**(B2)** The response variables $\boldsymbol{y}$ are bounded by $K_2$.
**(B3)** For $\ell \in \{0, \ldots, L\}$, the activation function $\sigma^{(\ell)}$ is bounded by $K_3$, with Fréchet derivative bounded by $K_3$ and $K_3$-Lipschitz.
**(B4)** The initializations $\{\rho_0^{(\ell)}\}_{\ell=0}^L$ have finite first moment and they are supported on $\|\boldsymbol{a}_{i_\ell, i_{\ell+1}}^{(\ell)}(0)\|_2 \leq K_4$ for $\ell \in [L-1]$, and $\|\boldsymbol{a}_{i_L}^{(L)}(0)\|_2 \leq K_4$.

We are now ready to present our results, which are proved in Appendix C in the supplementary material.

**Theorem 2** (Multilayer)**.** *Assume that conditions **(B1)-(B4)** hold, let $\boldsymbol{\theta}(k)$ be obtained by running $k$ steps of the SGD algorithm (4.3) with data $\{(\boldsymbol{x}_j, \boldsymbol{y}_j)\}_{j=0}^k \overset{\text{i.i.d.}}{\sim} \mathbb{P}$ and initializations $\{\rho_0^{(\ell)}\}_{\ell=0}^L$, and define $T = k\alpha > 0$. Then, the following results hold:*

***(A)** Pick $\mathcal{A}_1, \ldots, \mathcal{A}_L \subseteq [N]$ independent of $\boldsymbol{\theta}(k)$. Then, with probability at least $1 - e^{-z^2}$, $\boldsymbol{\theta}(k)$ is $\varepsilon_{\mathrm{D}}$-dropout stable with $\varepsilon_{\mathrm{D}}$ equal to*

$$K(T, L)\left(\frac{\sqrt{d} + z}{\sqrt{A_{\min}}} + \sqrt{\frac{\log N}{N}} + \sqrt{\alpha}\left(\sqrt{d + \log N} + z\right)\right) \tag{4.4}$$

*where $A_{\min} = \min_{i \in [L]} |\mathcal{A}_i|$, $d = \max_{\ell \in \{0, \ldots, L+1\}} d_\ell$ and the constant $K(T, L)$ depends on $T, L$ and on the constants $K_i$ of the assumptions.*

***(B)** Let $\boldsymbol{\theta}'(k')$ be obtained by running $k'$ steps of the SGD algorithm (4.3) with data $\{(\boldsymbol{x}_j', \boldsymbol{y}_j')\}_{j=0}^{k'} \overset{\text{i.i.d.}}{\sim} \mathbb{P}$ and initializations $\{(\rho_0^{(\ell)})'\}_{\ell=0}^L$ that satisfy **(B4)**, and define $T' = k'\alpha > 0$. Then, with probability at least $1 - e^{-z^2}$, $\boldsymbol{\theta}(k)$ and $\boldsymbol{\theta}'(k')$ are $\varepsilon_{\mathrm{C}}$-connected with $\varepsilon_{\mathrm{C}}$ equal*

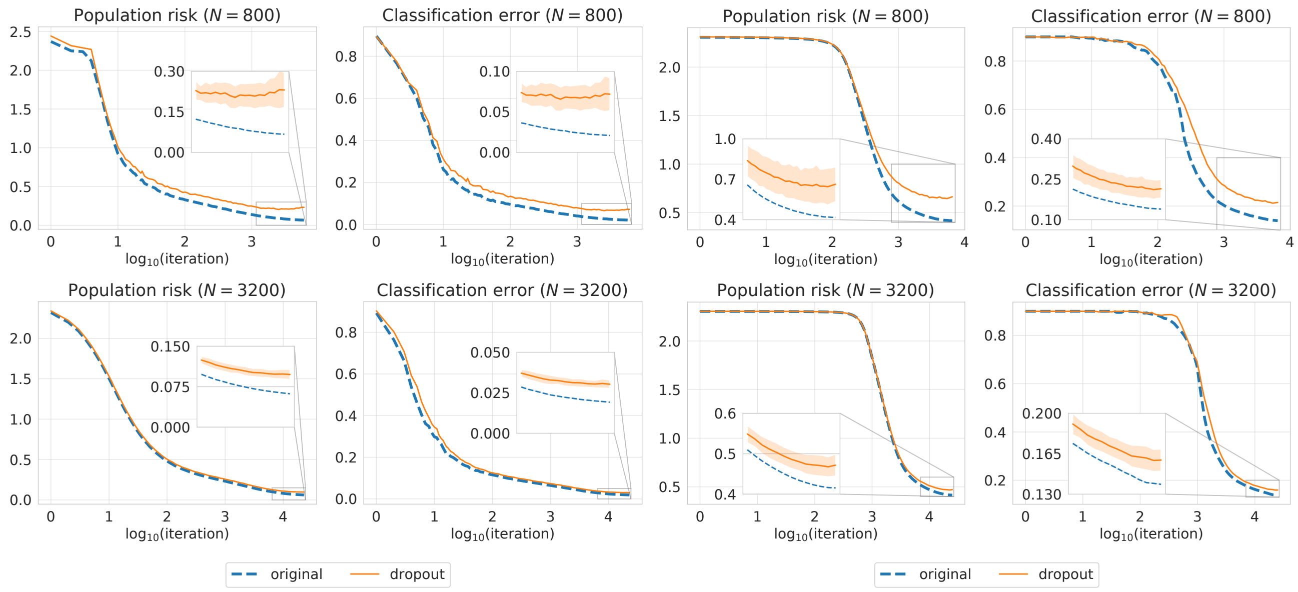
$$K(T_{\max}, L)\left(\frac{\sqrt{d + \log N} + z}{\sqrt{N}} + \sqrt{\alpha}\left(\sqrt{d + \log N} + z\right)\right) \tag{4.5}$$

*where $T_{\max} = \max(T, T')$.*

The results are similar in spirit to those of Theorem 1, but the analysis is more involved. We remark that, differently from the two-layer case, the ideal particles are not independent, see Remark 5.6 of (Araújo et al., 2019). We exploit a bound on the norm of the weights during training (see Lemma C.1 in Appendix C.1) and a bound on the *maximum* distance between SGD weights and weights of ideal particles. Our analysis improves upon (Araújo et al., 2019), where the bound is on the *average* distance between SGD and ideal-particle weights (compare (C.23) in Appendix C.1 and (10.1) in (Araújo et al., 2019)). This improvement is essential to show dropout stability. In fact, dropout stability requires dropping all weights associated to a subnetwork (and not just a given fraction of weights). The stronger guarantee on the distance to ideal particles leads to an extra $\log N$ in our bounds (compare Theorem 2 in this paper and (5.1) in (Araújo et al., 2019)). As concerns the proof of connectivity, we generalize the approach of (Kuditipudi et al., 2019), in order to analyze the model (4.1).

The bounds in Theorem 2 are not dimension-free (as in the two-layer case), but the dependence on the dimension $d$ is only linear. In fact, the loss change in (4.4) vanishes as long as $A_{\min} \gg d$, and $\alpha \ll 1/(d + \log N)$. The condition $A_{\min} \gg d$ implies that $A_{\min}$ needs to scale at least linearly with $d$, but does not scale with $N$. Furthermore, as in the two-layer case, the condition $\alpha \ll 1/(d + \log N)$ implies that the number of samples $k$ needs to scale only logarithmically with $N$.

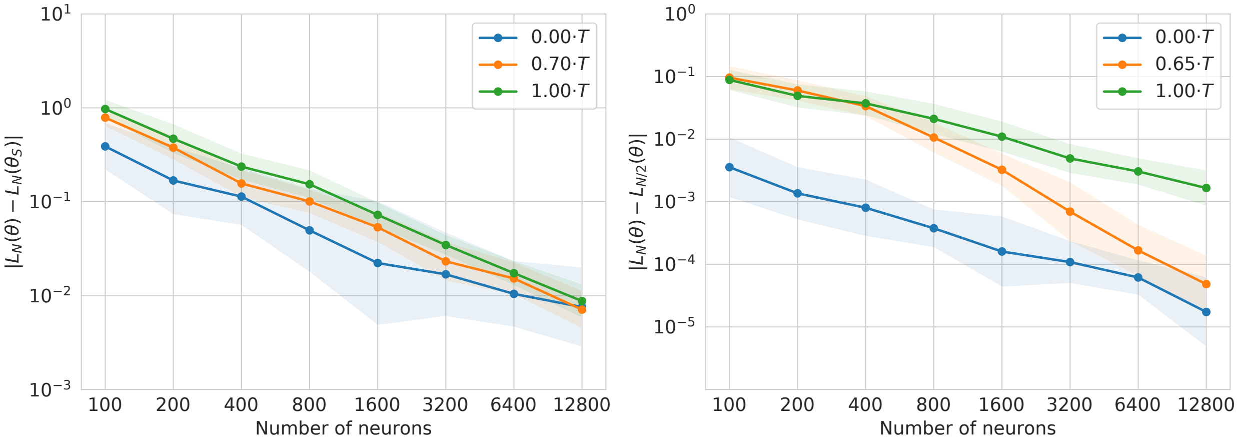
Compared to the two-layer case where there is no assumption on the initialization for $\boldsymbol{w}_i$, here we require a mild condition (finite first moment for $\rho_0^{(\ell)}$) in order to simplify the proof.



(a) MNIST, two-layer

(b) CIFAR-10, three-layer

*Figure 1.* Comparison of population risk and classification error between the trained network (blue dashed curve) and the dropout network (orange curve). In the full scale plot, we show the average values, and in the zoomed version we also provide the error bar.



(a) MNIST, two-layer

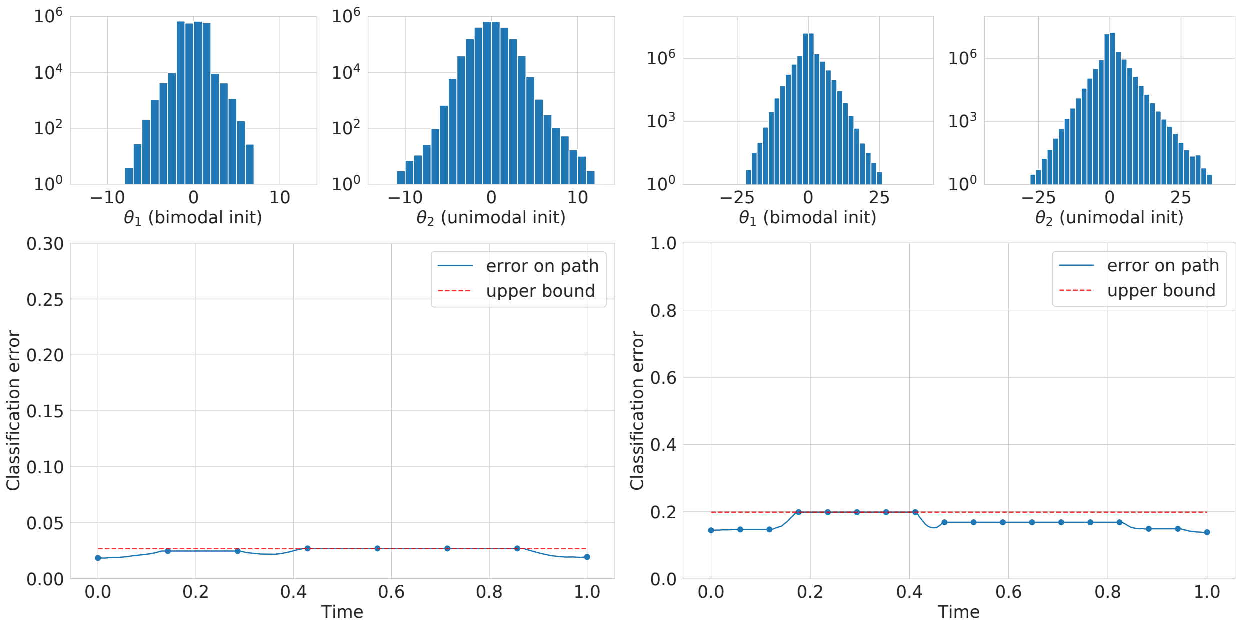
(b) CIFAR-10, three-layer

*Figure 2.* Change in loss after removing half of the neurons from each layer, as a function of the number of neurons $N$ of the full network.

## 5. Numerical Results

We consider two supervised learning tasks: *(a)* MNIST classification with the two-layer neural network (3.1); and *(b)* CIFAR-10 classification with the three-layer neural network (1.1). For MNIST, the input dimension is $d = 28 \times 28 = 784$ and we normalize pixel values to have zero mean and unit variance. For CIFAR-10, the input is given by VGG-16 features of dimension $d = 4 \times 4 \times 512 = 8192$. These features are computed by the convolutional layers of the VGG-16 network (Simonyan & Zisserman, 2015) pre-trained on the ImageNet dataset (Russakovsky et al., 2015). More specifically, we rescale the images to size $128 \times 128$, we rescale pixel values into the range $[-1, 1]$, and we feed them to the pre-trained VGG-16 network to extract the features. Qualitatively similar results (with larger classification error) are obtained by using fully connected networks directly on CIFAR-10 images.

For both tasks, the neural networks have ReLU activation functions, SGD aims at minimizing the cross-entropy loss, and the gradients are averaged over mini-batches of size 100.



(a) MNIST, two-layer

(b) CIFAR-10, three-layer

*Figure 3.* Classification error along a piecewise linear path that connects two SGD solutions $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$, with $N = 3200$. As predicted by the theory, the error along the path (blue curve) is no larger than the error of the two SGD solutions plus the change in loss due to the dropout of half of the neurons (red dashed curve).

In contrast with the setting of Section 4, all the layers of the neural network are trained. The scaling of the gradient updates follows (3.3) and (4.3): for the first and last layer, the gradient of the loss function is multiplied by a factor of $N$; for the middle layers, the gradient of the loss function is multiplied by a factor of $N^2$. This scaling ensures that the term in front of the learning rate $s_k$ does not depend on $N$, i.e., it is $\Theta(1)$ as $N$ goes large. The learning rate $s_k = \alpha\xi(k\alpha)$ does not depend on the time of the evolution, i.e., $\xi(t) = 1$. Furthermore, we set $\alpha = \alpha_0/N$, where $\alpha_0$ is a constant independent of $N$. We also set the number of training epochs to $k_0 \cdot N$, where $k_0$ is a constant independent of $N$. In this way, the product between the learning rate and the number of training epochs is the constant $T = k_0 \cdot \alpha_0$, which does not depend on $N$. The initializations of the parameters of the neural network are i.i.d. and do not depend on $N$, as in the setting described for the theoretical results. The population risk and the classification error are obtained by averaging over the test dataset. To measure statistics in the plots, i.e., average value and error bar at 1 standard deviation, we perform 20 independent trials of each experiment.

Figure 1 compares the performance of the trained network (blue dashed curve) and of the dropout network (orange curve), which is obtained by removing the second half of the neurons from each layer (and by suitably rescaling the remaining neurons). On the left, we report the results for MNIST, and on the right for CIFAR-10. For each classification task, we plot the population risk and the classification error for $N = 800$ and $N = 3200$. The networks are trained until the training loss has reached a plateau (0.062 for MNIST and 0.415 for CIFAR-10 when $N = 3200$). As expected, the performance of the dropout network improves with $N$, and it is very close to that of the trained network. For $N = 3200$, the classification error of the trained network is $< 2\%$ for MNIST and $< 14\%$ on CIFAR-10, and the classification error of the dropout network is $\approx 3\%$ on MNIST and $< 16\%$ on CIFAR-10.

Figure 2 plots the change in loss when only half of the neurons remain in the network, as a function of the total number of neurons $N$. For each classification task, we plot the change in loss at the beginning of training $(0 \cdot T)$, at an intermediate point where the population risk is still not too small $(\{0.65, 0.7\} \cdot T)$, and at the end of training $(1 \cdot T)$, where $T$ stands for the product of the learning rate and the total number of training epochs. The dependence between the change in loss and $N$ is essentially linear in log-log scale, as demonstrated by our theoretical results. Furthermore, the dependence on the time of the dynamics is quite mild.

Figure 3 shows that the optimization landscape is approximately connected when $N = 3200$. We plot the classification error along a piecewise linear path that connects two SGD solutions $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$ initialized with different distributions: the initial distribution of $\boldsymbol{\theta}_1$ is bimodal, while

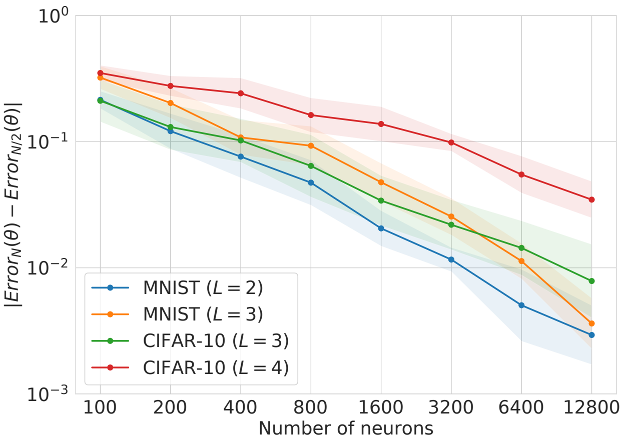


*Figure 4.* Change in classification error after removing half of the neurons from each layer, as a function of the number of neurons $N$ of the full network, at the end of training.

the initial distribution of $\boldsymbol{\theta}_2$ is unimodal. We also show the histograms of $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$, in order to highlight that one SGD solution cannot be obtained as a permutation of the other. As expected, the classification error along the path is roughly constant, since the network is dropout stable. More specifically, the error along the path (blue curve) is upper bounded by the error at the extremes plus the change in loss after dropping out half of the neurons of the network (red dashed curve).

Figure 4 plots the degradation in classification error due to the removal of half of the neurons from each layer. We consider neural networks at the end of training ($1 \cdot T$) and we report the performance degradation as a function of the number of neurons $N$ of the full network. We compare different architectures (two-layer, three-layer and four-layer neural networks) and classification tasks (MNIST and CIFAR-10). In all the cases considered, the performance degradation rapidly decreases, as the width of the network grows. When $N = 12800$, the classification error increases only *(i)* by $0.35\%$ for a two-layer network trained on MNIST, *(ii)* by $0.4\%$ for a three-layer network trained on MNIST, *(iii)* by $1\%$ for a three-layer network trained on CIFAR-10, and *(iv)* by $3.6\%$ for a four-layer network trained on CIFAR-10.

Additional experiments are presented in Appendix D in the supplementary material for the following learning tasks: classification of isotropic Gaussians with the two-layer neural network (3.1); MNIST classification with the three-layer neural network (1.1); CIFAR-10 classification with the four-layer neural network (1.1).

## 6. Discussion and Future Directions

The optimization landscape of neural networks can exhibit spurious local minima (Yun et al., 2018; Safran & Shamir, 2018), and its minima can be disconnected (Freeman & Bruna, 2017; Venturi et al., 2019; Kuditipudi et al., 2019). In this work, we show that these problematic scenarios are ruled out with SGD training and over-parametrization. In particular, we prove that the optimization landscape of SGD solutions is increasingly connected as the number of neurons grows. The explanation to this phenomenon has been hypothesized by some recent work: the SGD solutions have degrees of freedom to spare (Draxler et al., 2018) or, equivalently, they are dropout stable (Kuditipudi et al., 2019). We give theoretical grounding to this conjecture by proving that SGD solutions are dropout stable, i.e., that the loss does not change much when we remove even a large amount of neurons. In order to have meaningful bounds, the number of neurons does not need to be of the same order of the number of samples (cf. (Nguyen & Hein, 2017; 2018; Nguyen et al., 2019; Nguyen, 2019b)). Furthermore, our bounds are dimension-free for two-layer networks and they scale linearly with the dimension for multilayer networks (cf. (Freeman & Bruna, 2017)). Our analysis builds on a recent line of work showing that the dynamics of SGD tends to a mean field limit as the number of neurons increases (Mei et al., 2018b; 2019; Araújo et al., 2019). We believe that with these tools one could prove similar results also for noisy SGD and projected SGD.

The notion of dropout stability is closely related to the fact that neural networks have many redundant connections, and therefore they can be pruned with little performance loss, see, e.g., (Guo et al., 2016; Molchanov et al., 2017; Frankle & Carbin, 2019; Liu et al., 2019). However, it is difficult even to compare the relative merits of the different pruning techniques (Gale et al., 2019), let alone to understand the fundamental reasons leading to sparsity in neural networks. Thus, it would be interesting to investigate whether mean field approaches provide a more principled way of pruning deep neural networks.

## Acknowledgements

M. Mondelli was partially supported by the 2019 Lopez-Loreta Prize. The authors thank Phan-Minh Nguyen for helpful discussions and the IST Distributed Algorithms and Systems Lab for providing computational resources.

## References

Allen-Zhu, Z., Li, Y., and Liang, Y. Learning and generalization in overparameterized neural networks, going beyond two layers. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 6155–6166, 2019a.

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

Araújo, D., Oliveira, R. I., and Yukimura, D. A mean-field limit for certain deep neural networks. arXiv:1906.00193, 2019.

Auer, P., Herbster, M., and Warmuth, M. K. Exponentially many local minima for single neurons. In *Advances in neural information processing systems (NIPS)*, pp. 316–322, 1996.

Blum, A. and Rivest, R. L. Training a 3-node neural network is NP-complete. In *Advances in neural information processing systems (NIPS)*, pp. 494–501, 1989.

Boumal, N., Voroninski, V., and Bandeira, A. The non-convex burer-monteiro approach works on smooth semidefinite programs. In *Advances in Neural Information Processing Systems (NIPS)*, pp. 2757–2765, 2016.

Chen, Z., Cao, Y., Gu, Q., and Zhang, T. Mean-field analysis of two-layer neural networks: Non-asymptotic rates and generalization bounds. `arXiv:2002.04026`, 2020.

Chizat, L. and Bach, F. On the global convergence of gradient descent for over-parameterized models using optimal transport. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 3036–3046, 2018.

Chizat, L., Oyallon, E., and Bach, F. On lazy training in differentiable programming. In *Neural Information Processing Systems (NeurIPS)*, pp. 2933–2943, 2019.

Choromanska, A., Henaff, M., Mathieu, M., Arous, G. B., and LeCun, Y. The loss surfaces of multilayer networks. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 192–204, 2015.

Dauphin, Y. N., Pascanu, R., Gulcehre, C., Cho, K., Ganguli, S., and Bengio, Y. Identifying and attacking the saddle point problem in high-dimensional non-convex optimization. In *Advances in neural information processing systems (NIPS)*, pp. 2933–2941, 2014.

Draxler, F., Veschgini, K., Salmhofer, M., and Hamprecht, F. Essentially no barriers in neural network energy landscape. In *International Conference on Machine Learning (ICML)*, pp. 1308–1317, 2018.

Du, S. S., Lee, J. D., Li, H., Wang, L., and Zhai, X. Gradient descent finds global minima of deep neural networks. In *International Conference on Machine Learning (ICML)*, pp. 1675–1685, 2018.

Du, S. S., Zhai, X., Poczos, B., and Singh, A. Gradient descent provably optimizes over-parameterized neural networks. In *International Conference on Learning Representations (ICLR)*, 2019.

Frankle, J. and Carbin, M. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *International Conference on Learning Representations (ICLR)*, 2019.

Freeman, C. D. and Bruna, J. Topology and geometry of half-rectified network optimization. In *International Conference on Learning Representations (ICLR)*, 2017.

Gale, T., Elsen, E., and Hooker, S. The state of sparsity in deep neural networks. arXiv:1902.09574, 2019.

Garipov, T., Izmailov, P., Podoprikhin, D., Vetrov, D. P., and Wilson, A. G. Loss surfaces, mode connectivity, and fast ensembling of dnns. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 8789–8798, 2018.

Ge, R., Lee, J. D., and Ma, T. Matrix completion has no spurious local minimum. In *Advances in Neural Information Processing Systems (NIPS)*, pp. 2973–2981, 2016.

Ge, R., Jin, C., and Zheng, Y. No spurious local minima in nonconvex low rank problems: A unified geometric analysis. In *International Conference on Machine Learning (ICML)*, pp. 1233–1242, 2017.

Goodfellow, I., Bengio, Y., and Courville, A. *Deep learning*. MIT press, 2016.

Goodfellow, I. J., Vinyals, O., and Saxe, A. M. Qualitatively characterizing neural network optimization problems. In *International Conference on Learning Representations (ICLR)*, 2015.

Guo, Y., Yao, A., and Chen, Y. Dynamic network surgery for efficient DNNs. In *Advances In Neural Information Processing Systems (NIPS)*, pp. 1379–1387, 2016.

Jacot, A., Gabriel, F., and Hongler, C. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in neural information processing systems (NeurIPS)*, pp. 8571–8580, 2018.

Javanmard, A., Mondelli, M., and Montanari, A. Analysis of a two-layer neural network via displacement convexity. arXiv:1901.01375, 2019.

Kawaguchi, K. Deep learning without poor local minima. In *Advances in neural information processing systems (NIPS)*, pp. 586–594, 2016.

Keskar, N. S., Mudigere, D., Nocedal, J., Smelyanskiy, M., and Tang, P. T. P. On large-batch training for deep learning: Generalization gap and sharp minima. In *International Conference on Learning Representations (ICLR)*, 2017.

Kuditipudi, R., Wang, X., Lee, H., Zhang, Y., Li, Z., Hu, W., Arora, S., and Ge, R. Explaining landscape connectivity of low-cost solutions for multilayer nets. In *Advances in neural information processing systems (NeurIPS)*, 2019.

Li, Y. and Liang, Y. Learning overparameterized neural networks via stochastic gradient descent on structured data. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 8157–8166, 2018.

Liang, S., Sun, R., Lee, J. D., and Srikant, R. Adding one neuron can eliminate all bad local minima. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 4350–4360, 2018a.

Liang, S., Sun, R., Li, Y., and Srikant, R. Understanding the loss surface of neural networks for binary classification. In *International Conference on Machine Learning (ICML)*, pp. 2840–2849, 2018b.

Liu, Z., Sun, M., Zhou, T., Huang, G., and Darrell, T. Rethinking the value of network pruning. In *International Conference on Learning Representations (ICLR)*, 2019.

Livni, R., Shalev-Shwartz, S., and Shamir, O. On the computational efficiency of training neural networks. In *Advances in neural information processing systems (NIPS)*, pp. 855–863, 2014.

Mei, S., Bai, Y., and Montanari, A. The landscape of empirical risk for non-convex losses. *Annals of Statistics*, 46 (6A):2747–2774, 2018a.

Mei, S., Montanari, A., and Nguyen, P.-M. A mean field view of the landscape of two-layer neural networks. *Proceedings of the National Academy of Sciences*, 115(33): E7665–E7671, 2018b.

Mei, S., Misiakiewicz, T., and Montanari, A. Mean-field theory of two-layers neural networks: dimension-free bounds and kernel limit. In *Conference on Learning Theory (COLT)*, 2019.

Molchanov, D., Ashukha, A., and Vetrov, D. Variational dropout sparsifies deep neural networks. In *International Conference on Machine Learning (ICML)*, pp. 2498–2507, 2017.

Nguyen, P.-M. Mean field limit of the learning dynamics of multilayer neural networks. arXiv:1902.02880, 2019a.

Nguyen, P.-M. and Pham, H. T. A rigorous framework for the mean field limit of multilayer neural networks. arXiv:2001.11443, 2020a.

Nguyen, P.-M. and Pham, H. T. A note on the global convergence of multilayer neural networks in the mean field regime. arXiv:2006.09355, 2020b.

Nguyen, Q. On connected sublevel sets in deep learning. In *International Conference on Machine Learning (ICML)*, pp. 4790–4799, 2019b.

Nguyen, Q. and Hein, M. The loss surface of deep and wide neural networks. In *International Conference on Machine Learning (ICML)*, pp. 2603–2612, 2017.

Nguyen, Q. and Hein, M. Optimization landscape and expressivity of deep CNNs. In *International Conference on Machine Learning (ICML)*, pp. 3727–3736, 2018.

Nguyen, Q., Mukkamala, M. C., and Hein, M. On the loss landscape of a class of deep neural networks with no bad local valleys. In *International Conference on Learning Representations (ICLR)*, 2019.

Pennington, J. and Bahri, Y. Geometry of neural network loss surfaces via random matrix theory. In *International Conference on Machine Learning (ICML)*, pp. 2798–2806, 2017.

Rahimi, A. and Recht, B. Random features for large-scale kernel machines. In *Advances in neural information processing systems (NIPS)*, pp. 1177–1184, 2008.

Rotskoff, G. M. and Vanden-Eijnden, E. Neural networks as interacting particle systems: Asymptotic convexity of the loss landscape and universal scaling of the approximation error. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 7146–7155, 2018.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3): 211–252, 2015.

Safran, I. and Shamir, O. On the quality of the initial basin in overspecified neural networks. In *International Conference on Machine Learning (ICML)*, pp. 774–782, 2016.

Safran, I. and Shamir, O. Spurious local minima are common in two-layer ReLU neural networks. In *International Conference on Machine Learning (ICML)*, pp. 4430–4438, 2018.

Simonyan, K. and Zisserman, A. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations (ICLR)*, 2015.

Sirignano, J. and Spiliopoulos, K. Mean field analysis of neural networks. arXiv:1805.01053, 2018.

Sirignano, J. and Spiliopoulos, K. Mean field analysis of neural networks: A central limit theorem. *Stochastic Processes and their Applications*, 2019a.

Sirignano, J. and Spiliopoulos, K. Mean field analysis of deep neural networks. arXiv:1903.04440, 2019b.

Venturi, L., Bandeira, A. S., and Bruna, J. Spurious valleys in two-layer neural network optimization landscapes. *Journal of Machine Learning Research*, 20:1–34, 2019.

Wei, C., Lee, J. D., Liu, Q., and Ma, T. On the margin theory of feedforward neural networks. arXiv:1810.05369, 2018.

Yun, C., Sra, S., and Jadbabaie, A. A critical view of global optimality in deep learning. In *International Conference on Learning Representations (ICLR)*, 2018.

Zou, D., Cao, Y., Zhou, D., and Gu, Q. Stochastic gradient descent optimizes over-parameterized deep relu networks. arXiv:1811.08888, 2018.