UNIVERSITÀ DEGLI STUDI DI TRENTO

DIPARTIMENTO DI MATEMATICA

Laurea Magistrale in Matematica

Weights regularity for residual neural networks in the depth limit

Candidato:

Giorgio Susanna

Relatore:

Michele Coghi

Contents

Chapter 1

Introduction

Diverse philosophical perspectives named deep learning in different ways, from cybernetics (from the 1940 to the 1960) to connectionism (from the 1980 and 1990s) and finally deep learning since the early 2000[19]. However, two crucial factors played a pivotal role throughout deep learning's evolution: larger datasets and models consistently drive breakthroughs in capability. Dataset size increased from 60000 training images in MNIST in 1994 [13], to roughly 10 billion images in Open Images [30] and ImageNet [12] in the 2010s. A similar trajectory is evident within model's depth. For example, in 2012, AlexNet [29] won the ImageNet Large Scale Visual Recognition Challenge (ILSVRC), it was only 13 layers in depth. Within a mere 3 years, Residual Networks revolutionized the State-Of-The-Art (SOTA) models in image recognition allowing for training of models of 154 layers in depth [23]. The merits of this architecture are corroborated by empirical results, as proved by the ILSVRC2015 victory [23]. In particular, they showed that the proposed architecture dramatically outperformed previous SOTA models in image recognition on two widespread dataset, namely ImageNet and CIFAR-10 [28]. The substantial size increase of the aforementioned datasets required a new approach to facilitate the efficient training of deeper models. Deeper models, while intuitively advantageous, presented a critical challenge: vanishing and exploding gradients (see [5] [18] for a more throughout overview). Even the deployment of well-established techniques, such as batch normalization [26] and dropout layers [40], proved surprisingly insufficient in mitigating these issues. Consequently, deep models underperformed against their shallow counterparts. Hence, the question whether deep models are really needed is rather interesting [2]. However, theoretically the former should be more general since the parameter space is larger and exhibit significantly greater non-linearity in the output. It was shown in 1969 [35], shallow neural networks without non-linear activation function are not capable to model the totality of boolean operators. But deep models are universal

approximators as it was first observed in [10] [24]. More recently, constructive bounds on the number of layers and neurons necessary to represent a target function were discovered (see [6] [11]), both implying an upper bound on the scaling in depth of $\lceil \log d \rceil$ with d being the size of each layer. Thanks to the introduction of Residual Networks, the authors of [23] enabled the training of deeper models by introducing "skip connections" or also referred to "shortcut connections". Therefore, the investigation into the depth of the model becomes somewhat superfluous, as this method demonstrably facilitates efficient training even for deep models. A fundamental understanding of ResNet hinges upon the key role played by "skip connections".

In a feed forward neural network, the output of an hidden layer is passed as the input of the next. In case of ResNet, the skip connection enhances the passed information by adding the previous hidden layer result. In a more rigorous fashion, denoting with h_t the hidden state at layer t,

$$h_{t+1} = \sigma(W h_t),$$

represents the forward pass in a Neural network with an activation function σ and trainable matrix weights W of consistent dimension with the input h_t and h_{t+1}.

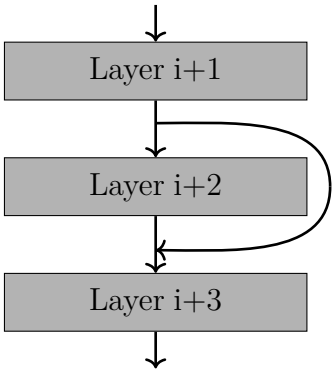


Figure 1.1: Example of Skip connection.

A skip connection with a skip of length 1, instead changes the forward pass in the following manner

$$h_{t+1} = h_t + \sigma(W h_t).$$

Therefore, within a supervised context, the purpose of the learning procedure in a ResNets is to estimate the increment at each step t in order to ensure that the output corresponds to the expected label. Conversely, a feed forward neural network optimization of the parameters is to estimate the "correct" hidden state t knowing the previous, which intuitively appears as more arduous. Another key observation that contrast the two architectures concerns the gradient. This is of substantial interest since the optimization algorithm are generally gradient-based, as we explain in chapter 2. For simplicity, we consider the identity activation function and a skip connection of length 1, which may be depicted as in Figure 1.1. Then, the computation of each hidden state is defined as

$$h_{t+1} = (I + W_t)h_t.$$

The gradient of the loss function with respect to the parameters W, once decom-

posed by chain-rule, is mainly composed by the product of the terms

$$\frac{\partial h_{t+1}}{\partial h_t} = I + W_t. \tag{1.1}$$

Contrariwise, in the case of a feed forward neural network and the identity activation function, each hidden state is defined as

$$h_{t+1} = W_t h_t,$$

and the gradient is composed mainly by the product of the following terms

$$\frac{\partial h_{t+1}}{\partial h_t} = W_t. \tag{1.2}$$

The difference between (1.1) and (1.2), albeit being minor, has two main advantages:

1. If W_t is ill-conditioned, the matrix $I + W_t$ is generally better conditioned.
2. In a deep feed forward neural network a matrix $W_t \approx 0$, at a fixed layer t, could damp the gradient for all the layers before t, hence making the training procedure ineffective.

Thus, the ResNet model seem to perform better throughout the learning procedure using gradient-based algorithms. From a geometrical perspective, ResNets improve the loss landscape by making it more convex as observed in [34]. Addi-

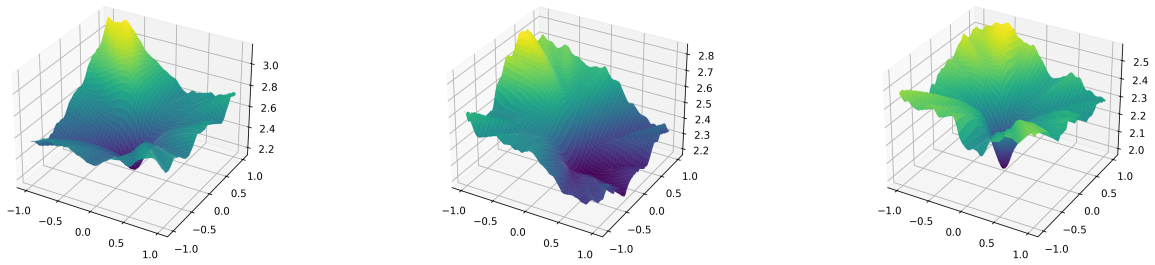


Figure 1.2: Representation of the loss landscape near a local minima via 3 different random gaussian direction in the parameter space using filter normalized plots as described in [34].

tionally, under random initialization of the parameters, the loss function seems to behave analogously to a Brown noise, instead of a white noise, leading to better

optimization results [3]. Recently, ResNets forward pass has been modified to include a regularizing constant at each increment, as described in Stable ResNet [22]. The forward pass to compute the $k+1$ hidden state, is evaluated as

$$h_{k+1} = h_k + \lambda_{k,L}\sigma(W_k h_k), \tag{1.3}$$

which stabilizes the gradient and ensures expressivity even in the infinite depth limit. An insightful analysis on the more appropriate value of the scaling $\lambda_{k,L}$, under the assumption of a uniform value over k, was conducted in [8]. They discovered that in case of a scaling $\lambda_{k,L} = L^{-\alpha}$ and weights initialization of order $L^{-\beta}$, the parameters were such that $\alpha + \beta \approx 1$, which is an assumption we use in chapter 3. In this introduction we highlighted the main practical potentials of this deep learning model, as well as introduced the main theoretical advantages from a quite loose perspective. However, its dynamic is not rigorously well-understood and a general framework has not been developed yet, although various proposals exist, e.g. in [42] they describe neural networks as gaussian processes. ResNet were proposed to be studied through the lenses of controlled dynamical system as in [14] [21], since the forward step resembles the forward Euler step approximation, as we will see in chapter 2. In the following we improve upon the results of [9], proving that the weights trajectories are of bounded 1 variation in analogous conditions. Moreover, we extend the result to a general loss function, instead of considering only the mean squared error (MSE) loss. Additionally, we verify numerically the assumptions and results obtained in chapter 3 on two distinct dataset in chapter 4. The results of chapter 3 establish that the controlled system associated to a ResNet model dwells in the "finite variation regime", hence in our setting, the stability results for a ResNet model should be consistent with [4, Theorem 2.2]. In particular, the author would like to thank Nikolas Tapia for the fruitful discussions and insights.

Chapter 2

ResNet model

This chapter aims to provide additional context to the results of chapter 3 and chapter 4. In particular, we describe a simplified model of a Residual Network specified by the following forward pass for an input data $x_i \in \mathbb{R}^d$,

$$\begin{cases} h_{k+1}^{x_i} = h_k^{x_i} + \sigma(W_k^{(L)} h_k), \quad k = 0, \dots, L-1 \\ h_0^{x_i} = x_i. \end{cases} \tag{2.1}$$

with $W_k^{(L)} \in \mathbb{R}^{d \times d}$ and $h_k \in \mathbb{R}^d$ for every $k \in 0, \dots, L-1$. We identify with $\sigma \colon \mathbb{R} \to \mathbb{R}$ an activation function, e.g. tanh, and with a little abuse of notation, we denote $\sigma(v) = (\sigma(v_1), \dots, \sigma(v_d))^\top$ for a vector $v \in \mathbb{R}^d$. This model is cut to the bone: it does not account for the bias term; it assumes that each layer has the same dimension, and each layer is a feed forward-like layer.

Remark 2.0.1. The bias term can be easily added without further introducing complexity, but by enhancing each layer with an additional "neuron". The following feed-forward step with bias

$$h_{k+1} = \sigma(W_k h_k + b_k), \tag{2.2}$$

can be linearised using

$$\widetilde{W}_k = \begin{pmatrix} W_k & \mathbf{b}_k^\top \\ 0_{1\times d} & 1 \end{pmatrix}, \quad \widetilde{h}_k = \begin{pmatrix} h_k \\ 1 \end{pmatrix},$$

with $\mathbf{b}_k = (b_k, \dots, b_k)^\top \in \mathbb{R}^d$. Then, it holds

$$\widetilde{h}_{k+1} = \sigma(\widetilde{W}_k \widetilde{h}_k), \quad \pi_h(\widetilde{h}_{k+1}) = \sigma(W_k h_k + b_k), \tag{2.3}$$

with π_h being the projection of the first d coordinates. A similar result holds in case of a ResNet forward step by denoting

$$\widetilde{W}_k = \begin{pmatrix} W_k & \mathbf{b}_k^\top \\ 0_{1\times d} & 0 \end{pmatrix}$$

and assuming $\sigma(0) = 0$ which is the case for most activation function, e.g. relu, tanh.

Remark 2.0.2. Despite the possibility of heterogeneous layer sizes, homogeneous layer sizes can reproduce the same results if carefully chosen. Notably, even in scenarios where layer dimensions are highly heterogeneous, such as transitioning from 3 to 5 and subsequently regressing to 2, it suffices to adopt the maximum dimensionality, i.e. 5, and "kill" the neurons whose weights correspond to unutilized units in the original architecture.

Therefore, from Remark 2.0.1 and Remark 2.0.2, this "cut-to-the-bone" model transcends mere "toy model" status by capturing intricate behavior comparable to highly engineered models. The model of chapter 3 discussion is a slight modification of the ResNet model described in (2.1):

$$\begin{cases} h_{k+1}^{x_i} = h_k^{x_i} + \delta_L \sigma(W_k^{(L)} h_k), \quad k = 0, \dots, L-1 \\ h_0^{x_i} = x_i, \end{cases} \tag{2.4}$$

with $\delta_L = L^{-\alpha}$ for $\alpha \in (0, 1]$. The scalar δ_L was introduced in [22] and [8] in which they studied the self-regularization properties on the parameters $W^{(L)}$ and the effectiveness of this additional scaling. In particular, thanks to this coefficient the connection to ODE is quite evident as we will prove in section 2.1. In the following, the model is discussed in the framework of supervised learning. This means that we assume $x_i \in \mathbb{R}^d$ as a input data and $y_i \in \mathbb{R}^d$ as a label for the given input x_i. In particular, we express the quality of a model with respect to a loss function $\ell \colon \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}_{\geq 0}$, which evaluates the discrepancy between predicted and true labels ($h_L^{x_i}$ and y_i respectively) for each input sample, then we average the result of the loss over the training dataset. The learning procedure of the parameters W of a deep neural network model, is thus defined by a minimization problem of the average of the loss function computed over the labelled dataset regarded as a function of the parameters W, i.e.

$$J(W) = \frac{1}{N} \sum_{i=1}^{N} \ell(y_i, \widehat{y}_i(W)), \tag{2.5}$$

where we denoted $\widehat{y}_i(W) = h_L^{x_i}$ as a more evocative notation for the output of the model with the given parameters $W \in \mathbb{R}^{L \times d \times d}$. A minima is generally reached through successive iteration of gradient descent with respect to the functional $J(W)$, although different procedure not relying on the gradient exists. Thanks to the work in back-propagation algorithm [32] [31] [39], i.e. an efficient way to estimate the gradient in a deep neural network, the optimization procedure is computationally feasible via an optimization algorithm dependent on the gradient. The most notable methods are gradient descent and stochastic gradient descent, which we briefly discuss in section 2.2.

2.1 ResNet and ODE

Deep neural network models have recently achieved remarkable successes. However being an old idea, they still lack a theoretical foundational framework for the model understanding and interpretability. For these reason in [14] was proposed the viewpoint of control problems to access to a deeper meaning to supervised learning procedures in the context of deep learning models. Moreover, for residual networks, in [4, Appendix A] and [21, Section 2.1], they propose a description of the forward step through the definition of a controlled differential equation. The inkling resides in the similarity between the forward Euler step and a ResNet forward pass. Namely, for an ODE

$$\frac{\mathrm{d}h_t}{\mathrm{d}t} = \sigma(W_t h_t), \quad t \in [0,T]$$

with $h \in \mathbb{R}^d$ and a path $W\colon [0,T] \to \mathbb{R}^{d\times d}$, which are applied component-by-component to a function $\sigma\colon \mathbb{R} \to \mathbb{R}$, i.e. $\mathbb{R}^d \ni v \mapsto \sigma(v) = (\sigma(v_1), \dots, \sigma(v_d))^\top \in \mathbb{R}^d$. Without loss of generality we assume $T = 1$, from here onwards. Then, the Euler forward step is given by

$$h_{t_{i+1}} = h_{t_i} + \sigma(W_{t_i} h_{t_i})\Delta t, \tag{2.6}$$

with $\Delta t = t^L_{i+1} - t^L_i$ for a succession of L equidistant points $0 = t^L_1 < \cdots < t^L_L = T$, i.e. $t^L_k = {}^k\!/_L$. It is clear that (2.6) is practically identical to (2.4) if $\delta_L = L^{-1}$. The only difference consists in a re-indexing of the hidden states: by rewriting $h_k = h_{t^L_k}$, which associates each hidden state h_k to the approximation at time t^L_k of (2.6) using the explicit forward Euler step. Eventually, for $k \in \{0, \dots, L-1\}$,

$$h_{k+1} = h_k + \sigma(W_{t^L_k} h_k)\frac{1}{L}. \tag{2.7}$$

A simple observation, reveals that controlling the path W, controls h_{k+1}. A more elegant formulation via a controlled system can be obtained by juxtaposing h and W regarded as a vector, i.e. $\widetilde{h} = (h^\top, \mathrm{vec}(W)^\top)^\top$ where we used the notation

$$\mathrm{vec}(W) = (W_{11}, W_{12}, \dots, W_{dd})^\top \in \mathbb{R}^{d^2}.$$

Additionally, we use the following notation,

Notation 2.1.1. Let E be a Banach space and $u\colon [0,1] \to E$ a path, we denote

$$u_{s,t} \coloneqq u_t - u_s \tag{2.8}$$

for $s, t \in [0,1]$.

To rewrite more compactly (2.7) under a new control $\widetilde{W}$,

$$\begin{cases} \widetilde{h}_{k+1}(\widetilde{x}_i) = \widetilde{h}_k(\widetilde{x}_i) + F(\widetilde{h}_k(\widetilde{x}_i))\widetilde{W}^L_{k,k+1}, \quad k = 0, \dots, L-1 \\ \widetilde{h}_0(\widetilde{x}_i) = \widetilde{x}_i = (x_i^\top, \mathrm{vec}(W_0)^\top)^\top. \end{cases} \tag{2.9}$$

With $\widetilde{h}_\cdot \in \mathbb{R}^{d^2+d}$,

$$F(\widetilde{h}) = \begin{pmatrix} 0_{d\times d^2} & \sigma(\pi_W(\widetilde{h})\pi_h(\widetilde{h})) \\ I_{d^2\times d^2} & 0_{d^2\times 1} \end{pmatrix},$$

where we denoted with $\pi_W(\widetilde{h})$ the projection of $\widetilde{h}$ with respect to the coordinates $d+1, \dots, d^2$ and $\pi_h(\widetilde{h})$ the projection of the first d coordinates of $\widetilde{h}$. The path $\widetilde{W}^L_k \in \mathbb{R}^{d^2+1}$, is given by $\widetilde{W}^L_{k,\mu} = (W_{t^L_k})_{\lceil \mu/d \rceil \mu \mod d}$ for $\mu = 1, \dots, d^2$ and $\widetilde{W}^L_{k,d^2+1} = t^L_k$. Then, we define the linear interpolation of (2.9),

$$\widetilde{h}^L_t(x_i) = \widetilde{x}_i + \sum_{k=0}^{\lfloor Lt \rfloor} F(\widetilde{h}^L_{t^L_k})\widetilde{W}_{t^L_k, t^L_{k+1}\wedge t}, \quad t \in (0,1], \tag{2.10}$$

The idea is to prove that the system (2.10) converges to the solution of the following Cauchy problem

$$\begin{cases} dH_t = F(H_t)d\widetilde{W}_t, \quad t \in (0,1] \\ H_0 = \widetilde{x}_i, \end{cases} \tag{2.11}$$

But first we need to introduce the notation for the space of continuous function between normed spaces and uniform norm.

Definition 2.1.2. Let $(E, \|\cdot\|_E)$ and $(G, \|\cdot\|_G)$ be two Banach spaces. A function $F\colon E \to G$ is said continuous if for any $\varepsilon > 0$ there exists $\delta > 0$ such that

$$\|x - y\|_E < \delta \implies \|F(x) - F(y)\|_G < \varepsilon. \tag{2.12}$$

We denote the uniform norm as

$$\|F\|_\infty \coloneqq \sup_{x\in E} \|F(x)\|_G \tag{2.13}$$

Additionally, we denote with $C(E, F)$ the space of continuous uniformly bounded functions F, i.e. $\|F\|_\infty < \infty$.

Finally, we introduce the notation of the p-variation semi-norm, although our cases are for $p = 1$.

Definition 2.1.3. Given a function $g\colon [0,1] \to (M, \|\cdot\|)$ with M being a normed space, we denote the $\|g\|_{p-\mathrm{var};[0,1]}$ as the p-variation semi-norm of g,

$$\|g\|_{p-\mathrm{var};[0,1]} \coloneqq \left(\sup_{\pi \text{ partition of } [0,1]} \sum_{i=0}^{\#\pi-1} \|g(s_{i+1}) - g(s_i)\|^p \right)^{\frac{1}{p}} \tag{2.14}$$

where we denoted by $\{s_0, \dots, s_{\#\pi-1}\}$ the elements of the partition π. Additionally, we will use the notation $C^{p-\mathrm{var}}([0,1], M)$ to denote the set of finite p-variation paths g over the interval $[0,1]$.

Theorem 2.1.4 (Existence)**.** *In the context of* (2.11)*, let*

1. $F \in C(\mathbb{R}^{d^2+d}, \mathbb{R}^{d^2+d} \otimes (\mathbb{R}^{d^2+1})^*)$*. Let* $M > 0$*, such that* $\|F\|_\infty \leq M$*.*
2. $\widetilde{W} \in C([0,1], \mathbb{R}^{d^2+1})$ *with* $\|\widetilde{W}\|_{1-var;[0,1]} < \infty$*.*

Then, there exists a sub-sequence of $\{\widetilde{h}_t^{L_n}\}_{n=1}^\infty$ *that converges uniformly to a solution* H *of the Cauchy problem* (2.11)*.*

Proof. Trivially it holds, for any choice of $L \in \mathbb{N}$ and $s, t \in [0,1]$

$$\|\widetilde{h}_{s,t}^L\| \leq M \|\widetilde{W}\|_{1-\mathrm{var};[s,t]}, \tag{2.15}$$

furthermore,

$$\|\widetilde{h}^L\|_{\infty;[0,1]} \leq \|\widetilde{x}_i\| + M \|\widetilde{W}\|_{1-\mathrm{var};[0,1]}, \tag{2.16}$$

i.e. the sequence $\widetilde{h}^L$ is uniformly bounded. From (2.15), we can deduce that $\{\widetilde{h}^L\}_{L=1}^\infty$ is equi-continuous because the control given by the 1-variation semi-norm is continuous. Therefore, we can apply Arzela-Ascoli's theorem to find a limit point H such that a sub-sequence $\widetilde{h}^{L_n} \to H$ uniformly. We are left to prove that H is the solution of (2.11). We note that, if $t \in (0,1]$, then

$$\widetilde{h}_{0,t}^L - \int_0^t F(\widetilde{h}_u^L)\, \mathrm{d}\widetilde{W}_u = \sum_{k=0}^{\lfloor Lt \rfloor} \int_{t_k^L}^{t_{k+1}^L \wedge t} \left(F(\widetilde{h}_{t_k^L}^L) - F(\widetilde{h}_u^L) \right) \mathrm{d}\widetilde{W}_u. \tag{2.17}$$

Once we acknowledge that for continuous functions Riemann-Stieltjes integrals coincides with Lebesgue-Stieltjes integrals, we can use dominated convergence on $\widetilde{h}^{L_n}$. Indeed, the succession $\widetilde{h}^{L_n}$ is uniformly bounded by an L^1 function by (2.16) and the point-wise convergence $\widetilde{h}_t^{L_n} \to H_t$ for $t \in [0,1]$ is due to the uniform

convergence. It yields, applying (2.17),

$$\begin{aligned} H_{0,t} - \int_0^t F(H_u)\,\mathrm{d}\widetilde{W}_u &= \lim_{n\to\infty}\left(\widetilde{h}^{L_n}_{0,t} - \int_0^t F(\widetilde{h}^{L_n}_u)\,\mathrm{d}\widetilde{W}_u\right) \\ &= \lim_{n\to\infty}\sum_{k=0}^{\lfloor L_n t\rfloor}\int_{t_k^{L_n}}^{t_{k+1}^{L_n}\wedge t}\left(F(\widetilde{h}^{L_n}_{t_k^{L_n}}) - F(\widetilde{h}^{L_n}_u)\right)\mathrm{d}\widetilde{W}_u \\ &\leq \lim_{n\to\infty}\sup_{u\in[0,1]}\|F(\widetilde{h}^{L_n}_{t_{k_u}^{L_n}}) - F(\widetilde{h}^{L_n}_u)\|\sum_{k=0}^{\lfloor L_n t\rfloor}\|\widetilde{W}\|_{1-\mathrm{var};[t_k^{L_n},t_{k+1}^{L_n}\wedge t]} \end{aligned}$$

where we introduced the notation $t_{k_u}^{L_n}$ to represent the largest element in the partition $\{t_k^{L_n}\}_{k=0}^{\infty}$ such that $t_k^{L_n} \leq u$. Then, by continuity of F and due to the decreasing mesh size as $n \to \infty$, it yields

$$\begin{aligned} \lim_{n\to\infty}\sup_{u\in[0,1]}\|F(\widetilde{h}^{L_n}_{t_{k_u}^{L_n}}) - F(\widetilde{h}^{L_n}_u)\|\sum_{k=0}^{\lfloor L_n t\rfloor}\|\widetilde{W}\|_{1-\mathrm{var};[t_k^{L_n},t_{k+1}^{L_n}\wedge t]} \\ \leq \lim_{n\to\infty}\sup_{u\in[0,1]}\|F(\widetilde{h}^{L_n}_{t_{k_u}^{L_n}}) - F(\widetilde{h}^{L_n}_u)\|\|\widetilde{W}\|_{1-\mathrm{var};[0,t]} = 0. \end{aligned}$$

Because $\|\widetilde{W}\|_{1-\mathrm{var};[0,1]} < \infty$ and F is continuous. Eventually proving that H is the solution of (2.11). □

Under the additional constraint of F being Lipschitz, we are able to prove the uniqueness of the solution for the problem described in (2.11), by a Picard-Lindelöf-type of argument,

Theorem 2.1.5 (Uniqueness)**.** *In the context of* (2.11)*. Let* $M > 0$ *and*

1. $F \in C(\mathbb{R}^{d^2+d}, \mathbb{R}^{d^2+d}\otimes(\mathbb{R}^{d^2+1})^*)$ *and* M*-Lipschitz, i.e.* $\forall a, b \in \mathbb{R}^{d^2+d}$ *then*

$$\|F(a) - F(b)\| \leq M\|a - b\|.$$

2. $\widetilde{W} \in C([0,1], \mathbb{R}^{d^2+1})$ *with* $\|\widetilde{W}\|_{1-var;[0,1]} < \infty$.

Then, (2.11) *admits a unique solution.*

Proof. Let $C([s,t], \mathbb{R}^{d^2+d})$ for $[s,t] \subset [0,1]$. We denote the initial condition $x \in \mathbb{R}^{d^2+d}$ and the map Ψ

$$\Psi\colon C([s,t], \mathbb{R}^{d^2+d}) \to C([s,t], \mathbb{R}^{d^2+d}), \quad H \mapsto \left(v \mapsto x + \int_s^v F(H_u)\,\mathrm{d}\widetilde{W}_u\right).$$

For any $H, H' \in C([s,t], \mathbb{R}^{d^2+d})$ and $v \in [s,t]$, it holds

$$\begin{aligned}\|\Psi(H)_v - \Psi(H')_v\| &\leq \int_s^v \|F(H_u) - F(H'_u)\| \, \|\mathrm{d}\widetilde{W}_u\| \\ &\leq \int_s^v M\|H_u - H'_u\| \, \|\mathrm{d}\widetilde{W}_u\| \\ &\leq M\|H - H'\|_\infty \|\widetilde{W}\|_{1-\mathrm{var};[s,t]}.\end{aligned}$$

This shows that the map Ψ is Lipschitz continuous in the space $C([s,t], \mathbb{R}^{d^2+d})$ endowed with the supremum norm and with Lipschitz constant determined by the Lipschitz constant of F and the 1-variation of $\widetilde{W}_u$. We note that $\|\widetilde{W}\|_{1-\mathrm{var};[0,\cdot]} \colon t \mapsto \|\widetilde{W}\|_{1-\mathrm{var};[0,t]}$ is non-decreasing and uniformly continuous on $[0,1]$ (see [16, Proposition 1.12]). Therefore, for any $\varepsilon > 0$ there exists $\delta > 0$ such that

$$|t-s| < \delta \implies \left| \|\widetilde{W}\|_{1-\mathrm{var};[0,t]} - \|\widetilde{W}\|_{1-\mathrm{var};[0,s]} \right| < \varepsilon.$$

Choosing $\varepsilon = 1/M$, there exists δ_M such that for each interval $[s,t] \subset [0,1]$ with $|t-s| < \delta_M$,

$$\left| \|\widetilde{W}\|_{1-\mathrm{var};[0,t]} - \|\widetilde{W}\|_{1-\mathrm{var};[0,s]} \right| < \frac{1}{M}.$$

Therefore, we conclude that Ψ is a contraction on $C([s,t], \mathbb{R}^{d^2+d})$ for $[s,t] \subset [0,1]$ and $|t-s| < \delta_M$. By Banach fixed-point theorem, for any initial condition x, Ψ has a unique fixed point. Eventually concluding that there exists a function $H \in C([s,t], \mathbb{R}^{d^2+d})$ such that

$$H_v = \Psi(H)_v = x + \int_s^v F(H_u) \, \mathrm{d}\widetilde{W}_u,$$

i.e. H is a solution of (2.11) on any interval $[s,t] \subset [0,1]$ with a fixed size $|t-s| < \delta_M$. By glueing the solutions over each interval of size δ_M, we prove the claim. □

Remark 2.1.6. From a practical perspective, a possible choice for the activation function to obtain both existence and uniqueness of the solution via the previous results, namely Theorem 2.1.5 and Theorem 2.1.4, is to choose $\sigma(x) = \tanh(x)$. The induced function $F \colon \mathbb{R}^{d^2+d} \to \mathbb{R}^{d^2+d} \otimes (\mathbb{R}^{d^2+1})^*$ is bounded because tanh is. It is trivially continuous, because $\tanh \in C^\infty(\mathbb{R})$. However, F is not Lipschitz in $\mathbb{R}^{d^2+d}$, albeit being locally Lipschitz, which is strong enough for our purposes. In particular, if we set a ball of fixed radius $K > 0$,

$$B(0,K) = \left\{ x \in \mathbb{R}^{d^2+d} \mid \|x\| < K \right\}. \tag{2.18}$$

Let $K > 0$ denote

$$\|H\|_\infty \leq \|H_0\| + M\|\widetilde{W}\|_{1-\text{var};[0,1]} = K,$$

which is a uniform bound for any solution H of (2.11). Then, for any elements $H, A \in B(0, K)$,

$$\begin{aligned}\|F(H) - F(A)\| &\leq \text{Lip}_\sigma \|\pi_W(H) - \pi_W(A)\| \|\pi_h(H)\| \\ &\quad + \text{Lip}_\sigma \|\pi_h(H) - \pi_h(A)\| \|\pi_W(A)\| \\ &\leq 2\text{Lip}_\sigma K \|H - A\|,\end{aligned}$$

where we denoted with Lip_σ the Lipschitz constant of the activation function $\sigma = \tanh$. In short, F is Lipschitz on $B(0, K)$. Moreover, any solution of (2.11) belongs to $B(0, K)$ at any time. Thus, considering as initial point $H \in C([s, t], \mathbb{R}^{d^2+d})$ such that $\|H\|_\infty \in B(0, K)$, we can proceed iteratively apply Ψ to H. This procedure has to converge to the fixed point, because the map Ψ is Lipschitz continuous and $\Psi(H) \in C([s, t], \mathbb{R}^{d^2+d})$ and $\|\Psi(H)\|_\infty < K$. Concluding that the uniqueness is preserved with little modification to the proof of Theorem 2.1.5.

2.1.1 Young differential equation connection

In this subsection we want to describe a slight modification to the previous results with a rougher path W in order to better describe the dynamics in case the coefficient $\delta_L = L^{-\alpha}$ for $\alpha \in (1/2, 1]$. However, before delving into further details, we need to introduce additional definitions.

Definition 2.1.7. Let $(E, \|\cdot\|_E)$ be a Banach space. We define the α-Hölder semi-norm for $\alpha \geq 0$ for a path $u_t \colon [0, 1] \to (E, \|\cdot\|_E)$ on the interval $[s, t] \subset [0, 1]$ as

$$\|u\|_{\alpha;[s,t]} \coloneqq \sup_{\substack{k<l \\ k,l\in[s,t]}} \frac{\|u_{l,k}\|_E}{|k - l|^\alpha}. \tag{2.19}$$

We will use the notation $C^\alpha([0, 1], E)$ for the set of α-Hölder paths u over the interval $[0, 1]$. And introduce $C^\alpha_x([0, 1], E)$ the set of α-Hölder paths u that starts from x, i.e. $u(0) = x$.

We introduce an additional notation for a uniform modification of the previous semi-norm as it plays an important role in the following proofs.

Definition 2.1.8. Let $(E, \|\cdot\|_E)$ be a Banach space. We define the α-Hölder semi-norm for $\alpha \geq 0$ and a generic interval of length $\tau \in (0, 1]$ for a path $u_t \colon [0, 1] \to (E, d)$ as

$$\|u\|_{\alpha;\tau} \coloneqq \sup_{\substack{|k-l|<\tau \\ k,l\in[0,1]}} \frac{\|u_{l,k}\|_E}{|k - l|^\alpha}. \tag{2.20}$$

The finiteness of the α-Hölder semi-norm is a stronger condition than having finite $1/\alpha$-variation (see Definition 2.1.3), since the first assumes continuity of the function, albeit there's a strong connection between the two definitions, see [16, Chapter 5]. As we have already mentioned, in [22] and [8], it was proposed the introduction of coefficients $\delta_L \simeq \Delta_t$ in analogy to (2.6) to further enhance the forward step of the model described in (2.1). In [9] the suggested scaling is given by $\Delta_t = L^{-\alpha}$ for $\alpha = 1/2$ which is uniformly dependent on the number of layers L. In order to achieve by analogy to the forward Euler step these coefficients dependent on L^α with $\alpha \in (1/2, 1]$, the scaling may be represented as further path $W\colon [0,1] \to \mathbb{R}^{d\times d}$ such that $\|W\|_{\alpha;[0,1]} \leq 1$, then

$$h_{t_{i+1}} = h_{t_i} + \sigma(h_{t_i})^\top W_{t_i^L, t_{i+1}^L}, \tag{2.21}$$

with t_i^L being equidistant points such that $|t_{i+1}^L - t_i^L| = \frac{1}{L}$. Then, we should expect, under strict conditions on the activation function σ, that there exists a path $U\colon [0,1] \to \mathbb{R}^{d\times d}$ that

$$\|h_{t_{i+1}^L}\| = \|h_{t_i^L} + \sigma(h_{t_i^L})^\top W_{t_i^L, t_{i+1}^L}\| \approx \left\| h_{t_i^L} + \frac{1}{L^\alpha}\sigma(U_{t_i^L} h_{t_i^L}) \right\|.$$

This is especially clear under the constraints $\|h\| \ll 1$ and $\sigma(x) \approx x$ in a neighborhood of 0. In the ODE case, a similar model to (2.21) was described in [15]. In this case the ODE is considered a Young Differential Equation (YDE), since $\alpha \in (1/2, 1]$ for which the integral is well-defined; see [17] for a thorough review of the framework. We state for reference the result that establishes the existence and well-definedness of the integral even for paths in the Young regime.

Theorem 2.1.9. *Let $\alpha, \beta \in (0,1)$ such that $\alpha + \beta > 1$. Denote with $X \in C^\alpha([0,1];\mathbb{R}^d)$ and $Y \in C^\beta([0,1];\mathbb{R}^d)$ two paths. Then, for any $[s,t] \subset [0,1]$, the integral $\int_s^t Y_r\, dX_r$ is well defined and the following inequality holds*

$$\left\| \int_s^t Y_r\, dX_r - Y_s X_{s,t} \right\| \leq C(\alpha,\beta)\|Y\|_{\beta;[s,t]}\|X\|_{\alpha;[s,t]}|t-s|^{\alpha+\beta}. \tag{2.22}$$

Proof. See [17, Theorem 2.7, Property 3]. □

We slightly rewrite (2.21) by considering $h_{t_k^L} = h_k^L$ to resemble the forward step of a residual network,

$$\begin{cases} h_{k+1}(x_i) = h_k(x_i) + \sigma(h_k(x_i))^\top W_{k,k+1}^L, \quad k = 0, \dots, L-1 \\ h_0(x_i) = x_i. \end{cases}$$

with $W^L_{k,k+1} = W_{t^L_k, t^L_{k+1}}$ for a path $W\colon [0,1] \to \mathbb{R}^{d\times d}$ such that for $\alpha \in (1/2, 1]$ we have $\|W\|_{\alpha;[0,1]} < B$. We are now able to rewrite a continuous version of the previous forward step,

$$h^L_t(x_i) \coloneqq x_i + \int_0^t \sigma(h^L_u)\,\mathrm{d}W_u - \sum_{k=0}^{\lfloor Lt\rfloor} \int_{t^L_k}^{t\wedge t^L_{k+1}} \left(\sigma(h^L_u)^\top - \sigma(h^L_{t^L_k})^\top\right)\mathrm{d}W_u, \tag{2.23}$$

and for ease of notation, we denote $\psi^L(t) = \sum_{k=0}^{\lfloor Lt\rfloor} \int_{t^L_k}^{t\wedge t^L_{k+1}} \left(\sigma(h^L_u)^\top - \sigma(h^L_{t^L_k})^\top\right)\mathrm{d}W_u$ which represents the error term with respect to the solution of the true YDE (2.24).

Theorem 2.1.10. *Let $\alpha \in (1/2, 1]$,*

1. *$W \in C^\alpha([0,1], \mathbb{R}^{d\times d})$.*
2. *$\sigma \in C^2(\mathbb{R}, \mathbb{R})$ such that the function σ and its derivative are uniformly bounded, i.e. there exists $M > 0$ such that $\|\sigma\|_\infty, \|\sigma^{(1)}\|_\infty \le M$.*

Then, there exists a solution H_t of (2.24),

$$\begin{cases} dH_t = \sigma(H_t)^\top dW_t \\ H_0 = x_i. \end{cases} \tag{2.24}$$

Moreover, there exists a subsequence of $\{h^{L_k}\}_{k=1}^\infty$ that converges uniformly to the solution H as $k \to \infty$.

Proof. We give a proof of the existence by an approximation result of the forward Euler step, hence proving that there exists convergence of the "ResNet" forward model and the YDE solution. For ease of notation we denote $h^L_t = h^L_t(x_i)$ given that the starting point is fixed to x_i. We first have to bound $\|\psi^L_{s,t}\| \lesssim |s-t|^\alpha L^{1-2\alpha}$, to deduce an upper bound on $\|\psi^L_{s,t}\|_{\alpha;\tau}$ for some fixed radius $\tau > 1/L$. We begin by estimating in trivial occurrences of the value s, t, namely $s = t^L_k$, $t = t^L_{k+1}$, then

$$\|\psi^L_{t^L_k, t^L_{k+1}}\| \le \left\| \int_{t^L_k}^{t^L_{k+1}} \sigma(h^L_u)^\top - \sigma(h^L_{t^L_k})^\top \,\mathrm{d}W_u \right\| \le C(\alpha)\|\sigma(h^L)\|_{\alpha;\tau}\|W\|_{\alpha;\tau}\frac{1}{L^{2\alpha}},$$

by Theorem 2.1.9, where we denoted with $C(\alpha) = C(\alpha, \alpha)$ for brevity. Next we consider two times in the same interval, i.e. $s, t \in [t^L_k, t^L_{k+1}]$, then

$$\begin{aligned} \|\psi^L_{s,t}\| &= \left\| \int_s^t \sigma(h^L_u)^\top - \sigma(h^L_{t^L_k})^\top \,\mathrm{d}W_u \right\| \\ &\le \left\| \int_s^t \sigma(h^L_u)^\top - \sigma(h^L_s)^\top + \sigma(h^L_s)^\top - \sigma(h^L_{t^L_k})^\top \,\mathrm{d}W_u \right\| \\ &= \left\| \int_s^t \sigma(h^L_u)^\top - \sigma(h^L_s)^\top \,\mathrm{d}W_u \right\| + \|\sigma(h^L_s)^\top - \sigma(h^L_{t^L_k})^\top\|\|W_{s,t}\|. \end{aligned}$$

In a similar fashion to the previous estimate, it holds

$$\begin{aligned}\left\|\int_s^t \sigma(h_u^L)^\top - \sigma(h_s^L)^\top \,\mathrm{d}W_u\right\| &+ \|\sigma(h_s^L) - \sigma(h_{t_k^L}^L)\|\|W_{s,t}\| \\ &\leq C(\alpha)\|\sigma(h^L)\|_{\alpha;\tau}\|W\|_{\alpha;\tau}|t-s|^{2\alpha} + \frac{\|\sigma(h^L)\|_{\alpha;\tau}}{L^\alpha}\|W\|_{\alpha;\tau}|t-s|^\alpha \\ &\leq \|\sigma(h^L)\|_{\alpha;\tau}\|W\|_{\alpha;\tau}\left(\frac{|t-s|}{L}\right)^\alpha (C(\alpha)+1),\end{aligned}$$

that is

$$\|\psi_{s,t}^L\| \leq (C(\alpha)+1)\|\sigma(h^L)\|_{\alpha;\tau}\|W\|_{\alpha;\tau}\left(\frac{|t-s|}{L}\right)^\alpha.$$

Finally, we consider s,t in two distinct intervals. We consider k and $k+m$ for $m>0$ and $k\geq 0$, then $s\in[t_k^L, t_{k+1}^L]$ and $t\in[t_{k+m}^L, t_{k+m+1}^L]$ such that $|t-s|<\tau$, and we observe that

$$m = L|t_{k+m}^L - t_k^L| \leq L|t-s|.$$

The estimate for $\psi_{s,t}^L$ follows using the previous results,

$$\begin{aligned}\|\psi_{s,t}^L\| &\leq \|\psi_{s,t_{k+1}^L}^L\| + \|\psi_{t_{k+m},t}^L\| + \sum_{l=1}^{m-1}\|\psi_{t_{k+l},t_{k+l+1}^L}^L\| \\ &\leq \|\sigma(h^L)\|_{\alpha;\tau}\|W\|_{\alpha;\tau}\left(2(C(\alpha)+1)\left(\frac{|t-s|}{L}\right)^\alpha + (m-1)C(\alpha)\frac{1}{L^{2\alpha}}\right) \\ &\leq \|\sigma(h^L)\|_{\alpha;\tau}\|W\|_{\alpha;\tau}\left(2(C(\alpha)+1)\left(\frac{|t-s|}{L}\right)^\alpha + C(\alpha)\frac{|t-s|}{L^{2\alpha-1}}\right) \\ &\leq 2(C(\alpha)+1)\|\sigma(h^L)\|_{\alpha;\tau}\|W\|_{\alpha;\tau}|t-s|^\alpha\left(\frac{1}{L^\alpha}+\frac{1}{L^{2\alpha-1}}\right),\end{aligned}$$

noting that $2\alpha-1\leq\alpha$ because $\alpha\in(1/2,1]$, it yields

$$\|\psi_{s,t}^L\| \leq 4(C(\alpha)+1)\|\sigma(h^L)\|_{\alpha;\tau}\|W\|_{\alpha;\tau}|t-s|^\alpha L^{1-2\alpha}. \tag{2.25}$$

Therefore, the α-Hölder semi-norm of ψ^L in a radius τ is bounded by

$$\|\psi^L\|_{\alpha;\tau} \leq 4(C(\alpha)+1)\|\sigma(h^L)\|_{\alpha;\tau}\|W\|_{\alpha;\tau}L^{1-2\alpha}. \tag{2.26}$$

We proceed to prove that also h^L is bounded uniformly in α-Hölder semi-norm.

From (2.23) for any $|s-t| < \tau$,

$$\begin{aligned}\|h^L_{s,t}\| &= \left\| \int_s^t \sigma(h^L_u)^\top \,\mathrm{d}W_u - \psi^L_{s,t} \right\| \\ &\leq \|W\|_{\alpha;\tau}|t-s|^\alpha \left(\|\sigma\|_\infty + C(\alpha)\|\sigma(h^L)\|_{\alpha;\tau}|t-s|^\alpha \right) \\ &\quad + 4(C(\alpha)+1)\|\sigma(h^L)\|_{\alpha;\tau}\|W\|_{\alpha;\tau}|t-s|^\alpha L^{1-2\alpha} \\ &\leq \|W\|_{\alpha;\tau}|t-s|^\alpha \left(\|\sigma\|_\infty + C(\alpha)\|\sigma\|_{C^1}\|h^L\|_{\alpha;\tau}\tau^\alpha \right) \\ &\quad + 4(C(\alpha)+1)\|\sigma\|_{C^1}\|h^L\|_{\alpha;\tau}\|W\|_{\alpha;\tau}|t-s|^\alpha L^{1-2\alpha}.\end{aligned}$$

Choosing τ and L respectively small enough and large enough,

$$C(\alpha)\tau^\alpha\|W\|_{\alpha;\tau}\|\sigma\|_{C^1} \leq \frac{1}{4}, \quad 4(C(\alpha)+1)\|\sigma\|_{C^1}\|W\|_{\alpha;\tau}L^{1-2\alpha} \leq \frac{1}{4},$$

It holds

$$\|h^L_{s,t}\| \leq \|W\|_{\alpha;\tau}\|\sigma\|_\infty|t-s|^\alpha + \frac{1}{2}\|h^L\|_{\alpha;\tau}|t-s|^\alpha, \tag{2.27}$$

thus diving by $|t-s|^\alpha$ and taking the supremum over $t \neq s$ such that $|t-s| < \tau$, it yields

$$\|h^L\|_{\alpha;\tau} \leq 2\|W\|_{\alpha;\tau}\|\sigma\|_\infty. \tag{2.28}$$

Thanks to the bounds found in (2.26) and (2.28), we are ready to prove the convergence of h^L to a solution H. We note that the subspace $C^\alpha_0([0,1];\mathbb{R}^d)$ is a Banach space endowed with the semi-norm $\|\cdot\|_{\alpha;\tau}$ (see [16, Theorem 5.25]). We further observe that in case the starting point is 0, the semi-norm becomes a norm. From (2.28), we have that the sequence $h^L - x_i$ is uniformly bounded C^α_0 and equi-continuous as a consequence of the estimate (2.28). Thus, by [16, Proposition 5.28], there exists a subsequence of $h^{L_k} - x_i$ such that it converges to a $h - x_i \in C^\alpha([0,1],\mathbb{R}^d)$ in $C^{\alpha-\varepsilon}([0,1],\mathbb{R}^d)$ for any $\varepsilon > 0$. Since, $h^L - x_i$ solves (2.23) and $\|\psi^L\|_{\alpha;\tau} \to 0$ as $L \to \infty$, we conclude that $h - x_i$ is a solution of (2.24) by continuity of the Young integral. □

Remark 2.1.11. Note that in case $\sigma(x) = \tanh(Ux)$ for any fixed matrix $U \in \mathbb{R}^{d\times d}$ the requirements of Theorem 2.1.10 are trivially satisfied.

2.2 Gradient Descent

In the introduction of this chapter, we provided an informal overview of the supervised learning task. In this section, we want to cover the basics of the most widespread gradient based optimization algorithms, namely gradient descent as it will be a core requirement for the understanding of chapter 3. The inkling of

gradient descent is the following: consider a convex function $f\colon \mathbb{R}^d \to \mathbb{R}$, we select a point x_0 and then iteratively compute x_t as described below

$$x_{t+1} = x_t - \eta(t)\nabla f(x_t), \quad t \in \mathbb{N}, \tag{2.29}$$

with $\eta(t)$ being a learning rate. For proper learning rates $\eta(t)$, the algorithm converges to the global minimum of the function f. The procedure described in (2.29) in the context of a non-convex function f, reaches a stationary point, namely a point $x \in \mathbb{R}^d$ such that $\nabla f(x) = 0$. In general a global minima might not exists for a non-convex function, moreover stationary points are not global minima, therefore this optimization procedure seems bound to fail. Despite their apparent non-linearity, empirical and theoretical evidence suggests that the optimization landscape of neural networks exhibits surprising properties. Notably, the work of [27] demonstrates that under mild assumptions and for the mean squared error (MSE) loss function, the following properties hold true for neural networks of any width and depth:

1. the loss function is non-convex and non-concave;
2. every local minimum is a global minimum;
3. every critical point that is not a global minimum is a saddle point.

The conjecture proved in [27] finds support in the empirical observations presented in [20]. Assuming a stochastic initialization, [1] proves that gradient descent optimization, for various deep neural networks model, achieves arbitrarily small error for common loss functions with a polynomial number of steps of gradient descent. Moreover, different architectures display different loss function with respect to the parameter space. Therefore, the loss landscape for different models may be regarded as "more convex" than in others. That is a key observation in [34], in which, thanks to a fine-tuned normalization of the parameter space, named filter normalization, the dimensionality reduction visualization seems compatible with theoretical results. Namely, the loss landscape seems rather convex near a local minima once we introduce skip connections to a deep neural network (see for example Figure 2.1). After these premises, we give a more in-depth overview of the gradient descent algorithm in the context of a general neural network model, then apply it to the model specified in (2.4). Let $L \in \mathbb{N}_{>0}$ be the number of layers of a neural network model, $\{d_k\}_{k=0}^{L}$ be the dimension of each hidden state h_k. We describe the forward step procedure given an input data $x \in \mathbb{R}^{d_0}$,

$$\begin{cases} h_{k+1} = F_k(h_k, W_k) & k = 0, \dots, L-1, \\ h_0 = x. \end{cases}$$

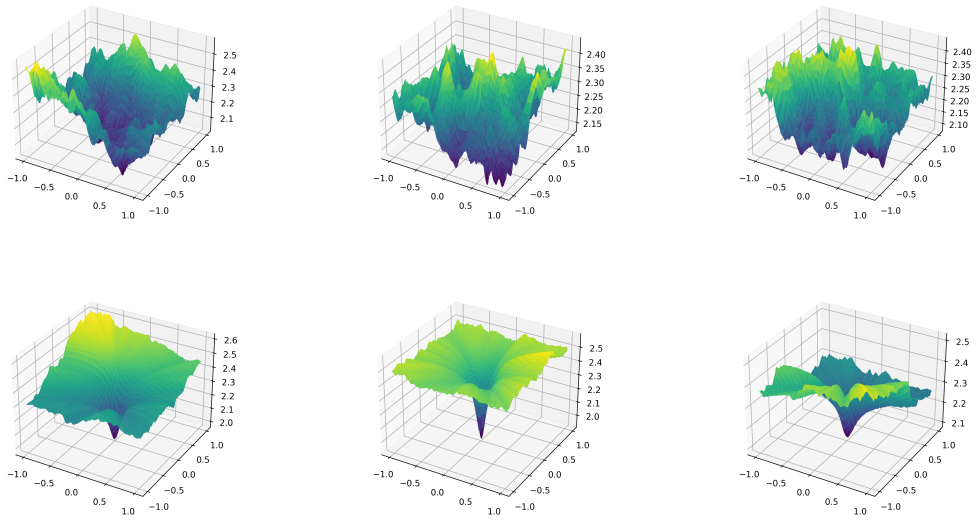


Figure 2.1: Figure depicts loss landscapes using the filter normalization method in [34]. Top panels show landscapes for six distinct directions without skip connections, while bottom panels depict the loss landscapes with skip connections.

where

$$F_k \colon \mathbb{R}^{d_k} \times \mathbb{R}^{d_k \times d_{k+1}} \to \mathbb{R}^{d_{k+1}}, \quad h_k \in \mathbb{R}^{d_k}, \quad W_k \in \mathbb{R}^{d_{k+1} \times d_k}.$$

In supervised learning, the training dataset is crucial. For this reason, we specify that the dataset comprises input-output pairs, denoted as $(x_i, y_i) \in \mathbb{R}^{d_0} \oplus \mathbb{R}^{d_L}$, where:

1. x_i represents the d_0-dimensional input vector of the i-th data point.
2. y_i represents the corresponding label or output value for x_i.
3. N represents the total number of data points belonging to the training dataset.

Then, the sequence of matrices $\{W_k\}_{k=0}^{L}$ are the parameters to optimize using gradient descent. We need to chose a loss function $\ell \colon \mathbb{R}^{d_L} \times \mathbb{R}^{d_L} \to \mathbb{R}_{\geq 0}$, e.g. the MSE-loss $\ell(y, \widehat{y}) = \frac{1}{2}\|y - \widehat{y}\|^2$, and consider the average of the functional as described in (2.5). Then, each layer is optimized using the gradient descent rule and choosing an appropriate learning rate $\eta(t) \in \mathbb{R}_{>0}$,

$$W_k(t+1) = W_k(t) - \eta(t) \nabla_k J(W(t)), \tag{2.30}$$

where we denote the gradient

$$\nabla_k J(W(t)) = \begin{pmatrix} \frac{\partial J(W(t))}{\partial W_{k,11}} & \cdots & \frac{\partial J(W(t))}{\partial W_{k,1d_{k+1}}} \\ \vdots & \ddots & \vdots \\ \frac{\partial J(W(t))}{\partial W_{k,d_k 1}} & \cdots & \frac{\partial J(W(t))}{\partial W_{k,d_k d_{k+1}}} \end{pmatrix}.$$

The introduction of back-propagation allowed for fast computation of the gradients, which heavily relies on the chain-rule composition of the gradient. Namely, by chain rule, the gradient can be simplified as

$$\begin{aligned} \nabla_k J(W(t)) &= \frac{1}{N} \sum_{i=i}^{N} \nabla_{\widehat{y}} \ell(y_i, \widehat{y}_i(W(t)))^\top \prod_{j=1}^{L-k-1} \frac{\partial h_{L-j+1}^{x_i}}{\partial h_{L-j}}(W(t)) \frac{\partial h_{k+1}^{x_i}}{\partial W_k}(W(t)) \\ &= \frac{1}{N} \sum_{i=i}^{N} \nabla_{\widehat{y}} \ell(y_i, \widehat{y}_i(W(t)))^\top M_{k+1}^{x_i}(W(t)) \frac{\partial h_{k+1}^{x_i}}{\partial W_k}(W(t)). \quad (2.31) \end{aligned}$$

where we introduced, for notational purposes, $M_k^{x_i}(W(t))$ for $k = 1, \ldots, L-1$ as

$$M_k^{x_i}(W(t)) = \prod_{j=1}^{L-k} \frac{\partial h_{L-j+1}^{x_i}}{\partial h_{L-j}}(W(t)). \quad (2.32)$$

Despite lacking guaranteed convergence to a global minimum, gradient descent (as described in (2.30)), remains prevalent due to its simplicity and efficiency. But various extensions tries to overcome its limitations, namely the high memory usage to store the gradients and the rate of convergence. Recognizing this limitation and aiming to accelerate convergence to local minima, researchers have developed numerous sophisticated optimization algorithms. For example, momentum-based optimizers, also known as Polyak's heavy ball method [38], address the inherent oscillations of gradient descent by incorporating additional information, such as past gradients. The simplest implementation of gradient descent with momentum is specified for each training step $t \in \mathbb{N}_{>0}$ in the following equations

$$\begin{cases} m(t+1) = \mu(t) m(t) + \eta(t) \nabla J(W(t)), \\ W(t+1) = W(t) - m(t+1). \end{cases}$$

where we introduced the momentum variable m and $\mu(t)$ specifying the relevance of the momentum at each step t. On the other hand, to alleviate the memory usage, and leveraging the stochastic nature of the learning process, gradient descent variants deviate from computing the gradient across the entire training dataset. In case of billion or trillion of data points, the gradient computation is reduced to few

data-points for each step, instead of considering the whole training dataset at once, pivotally improving the computational performance of the training procedure. In this regards, there are mainly two variants: stochastic gradient descent (SGD), which utilizes at each step a single data point, and batch gradient descent, which employs a fixed-size subset of the training dataset, named batch, to compute the gradient, offering a trade-off between computational efficiency and a more stable gradient than SGD. We describe a step of batch gradient descent in (2.2) for completeness,

$$W(t+1) = W(t) - \eta(t)\frac{1}{B}\sum_{i=1}^{B} \nabla_W \ell(y_{X_i}, \widehat{y}_{X_i}(W(t)))$$

with $B < N$ the size of the batch, $X \colon \Omega \to \mathbb{R}^B$ being a random variable which uniformly samples from the training dataset of N points.

2.2.1 Gradient for the ResNet model

We defer the definitions for the canonical orthonormal basis, which we denote with $\{e_i\}_{i=1}^d \subset \mathbb{R}^d$ as well as the tensor product definition to Appendix A. In the last part of this section we compute the gradient of the loss for the model described in (2.4). Using a similar notation to (2.32) for the Residual Network model (2.4), we have

$$M_k^{x_i}(W(t)) = \prod_{j=1}^{L-k} (I_d + \delta_L \operatorname{diag} \nabla\sigma(W_{L-j}(t) h_{L-j}^{x_i}(W(t)))W_{L-j}(t)), \tag{2.33}$$

where we denoted for a given vector $v \in \mathbb{R}^d$,

$$\nabla\sigma(v) = \begin{pmatrix} \frac{\partial\sigma(v_1)}{\partial v_1} \\ \vdots \\ \frac{\partial\sigma(v_d)}{\partial v_d} \end{pmatrix}, \quad \operatorname{diag} v = \begin{pmatrix} v_1 & 0 & \cdots & 0 \\ 0 & v_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & v_d \end{pmatrix}.$$

Then, using (2.31) for a single labelled data $(x, y) \in \mathbb{R}^d \oplus \mathbb{R}^d$

$$\frac{\partial \ell(y, \widehat{y})}{\partial W_k} = \nabla_{\widehat{y}} \ell(y, \widehat{y})^\top M_{k+1} \frac{\partial h_{k+1}}{\partial W_k}, \tag{2.34}$$

where we used a single data point and for notational purposes we omit the initial data point x as a dependency for M_{k+1} and h_k for $k = 0, \ldots, L$. By linearity of

the differential operator, it is a trivial feat to generalize the previous result to N distinct data points. We can rewrite the term $\frac{\partial h_{k+1}}{\partial W_k}$ more explicitly,

$$\begin{aligned}\frac{\partial h_{k+1}}{\partial W_k} &= \delta_L \frac{\partial \sigma(W_k(t)h_k(W(t)))}{\partial W_k} \\ &= \delta_L \sum_{i=1}^{d} \frac{\partial \sigma_i((W_k(t)h_k(W(t)))_i)}{\partial W_k} e_i \\ &= \delta_L \sum_{i,j=1}^{d} \frac{\partial \sigma_i((W_k(t)h_k(W(t)))_i)}{\partial (W_k h_k)_j} \frac{\partial (W_k(t)h_k(W(t)))_j}{\partial W_{k,j}} e_j \otimes e_i \\ &= \delta_L \sum_{i,j,l=1}^{d} \frac{\partial \sigma_i((W_k(t)h_k(W(t)))_i)}{\partial (W_k h_k)_j} \frac{\partial (W_k(t)h_k(W(t)))_j}{\partial W_{k,j,l}} e_l \otimes e_j \otimes e_i\end{aligned}$$

noting that $\frac{\partial \sigma_i((W_k h_k)_i)}{\partial (W_k h_k)_j} \neq 0$ if and only if $i = j$, it holds

$$\begin{aligned}\frac{\partial h_{k+1}}{\partial W_k} &= \delta_L \sum_{i,l=1}^{d} \frac{\partial \sigma_i((W_k(t)h_k(W(t))_i)}{\partial (W_k h_k)_i} \frac{\partial (W_k(t)h_k(W(t)))_i}{\partial W_{k,i,l}} e_l \otimes e_i \otimes e_i \\ &= \delta_L \sum_{i,l=1}^{d} \sigma^{(1)}(W_{k,i}(t)h_k(W(t)))h_{k,l}(W(t))e_l \otimes e_i \otimes e_i. \quad (2.35)\end{aligned}$$

Finally, we are able to rewrite (2.34) for the ResNet model (2.4) more explicitly,

$$\begin{aligned}\frac{\partial \ell(y, \widehat{y})}{\partial W_k} &= \nabla_{\widehat{y}} \ell(y, \widehat{y})^\top \prod_{j=1}^{L-k-1} (I_d + \delta_L \operatorname{diag} \nabla \sigma(W_{L-j}(t)h_{L-j}(W(t)))W_{L-j}(t)) \\ &\quad \cdot \delta_L \sum_{i,l=1}^{d} \sigma^{(1)}(W_{k,i}(t)h_k(W(t)))h_{k,l}(W(t))e_l \otimes e_i \otimes e_i. \quad (2.36)\end{aligned}$$

Thus, when we are considering the parameters $W^{(L)} \in \mathbb{R}^{L \times d \times d}$ in the context of the gradient descent evolution, we append a "time"-dependent parameter identifying the steps of gradient descent applied to the initial parameters, namely

$$\begin{aligned}W_k^{(L)}(t+1) &= W_k^{(L)}(t) - \eta(t) \nabla_k J(W^{(L)}(t)) \\ &= W_k^{(L)}(t) - \eta(t) \sum_{i=1}^{N} \frac{\partial \ell(y_i, \widehat{y}_i(W^{(L)}(t)))}{\partial W_k}, \quad (2.37)\end{aligned}$$

with $\frac{\partial \ell(y, \widehat{y}(W^{(L)}(t)))}{\partial W_k}$ as denoted in (2.36), and specify that $\widehat{y}(W^{(L)}(t)))$ is dependent on the parameters of the model at step t of the optimization procedure.

Chapter 3

Weights regularity

The analogy between ResNet and Euler approximation steps of an ordinary differential equation plays a crucial role in the model's understanding as first introduced in [14]. In section 2.1, we established fundamental existence and uniqueness results for neural networks under well-defined assumptions on the weights and activation function regularity (see Theorem 2.1.4 and Theorem 2.1.5). In Remark 2.1.6 we highlighted that, when employing the tanh activation function within the ResNet architecture (2.4) for $\delta_L = L^{-1}$, a subsequence of the forward pass converges, as L approaches infinity, to the unique solution of the corresponding ordinary differential equation (2.11). This finding strengthen the theoretical connection between deep learning models and continuous dynamical systems, paving the way for further potential benefits for interpretability and control. However, to guarantee *a priori* bounds in the ODE setting, a bound of the regularity of the *control* is required, which in our case corresponds to the weights of the Residual Neural Network, see section 2.1. The outline of this chapter is to define initial conditions of the weights W, which is based on a similar assumptions and regime due to [9]. Then, in the context of a supervised learning task optimized through gradient descent, we prove that the weights norm as a trajectory are of bounded 1-variation, i.e. Corollary 3.4.5. Additionally, the 1-variation of the weights, regarded as path in d^2 dimension, is always bounded, as shown in Theorem 3.4.2. Moreover, if we consider the path described by the weights as linearly interpolated we achieve a convergence result via a Arzela-Ascoli's type of argument in Theorem 3.4.7. The results are then followed by chapter 4, in which we test numerically the prediction of the theoretical results.

3.1 ResNet model

In this chapter we consider the ResNet model described in (2.4). In particular, the coefficient δ_L scales with respect to L, namely $\delta_L = L^{-\alpha}$ where α is chosen according to Assumption 3.1.1 (iii). Moreover, we rely on gradient descent optimization (see subsection 2.2.1) in the context of supervised learning as described in chapter 2. Namely, we use the same notation of (2.5) to denote the average of the loss function over a given dataset with N points satisfying Assumption 3.1.1 (i). The norms are the euclidean norms for the appropriate order of tensor in over the vector space $\mathbb{R}^d$, as described in Appendix A. We note that the notation $\|W\|_\infty = \max_{0,\dots,L-1} \|W\|$ is consistent with the notation of the uniform norm (2.13), when we consider W as a linearly interpolated path with respect to the layers. Therefore, with a little abuse of notation we denote with $\|W\|_\infty = \max_{0,\dots,L-1} \|W\|$.

Assumption 3.1.1.

(i) For each input data and label, respectively $x_i, y_i \in \mathbb{R}^d$, for $i = 1, \dots, N$, we assume them to be normalized, i.e. $\|x_i\| \leq 1$,$\|y_i\| \leq 1$;

(ii) The activation function $\sigma \in C^1(\mathbb{R})$, and additionally, $|\sigma(x)| \leq |x|$, $|\sigma^{(1)}(z)| \leq 1$ and $\sigma^{(1)}$ is 1-Lipschitz continuous;

(iii) Let $\nu, \gamma \in (0, \infty)$, $\alpha, \beta \in [0, 1]$ such that $\alpha + \beta = 1$. We assume the weights at initialization satisfy the following conditions

$$\|W(0)\|_\infty = \max_{k=0,\dots,L-1} \|W_k(0)\| \leq \frac{\nu}{2} L^{-\beta};$$
$$\|W_{k,k+1}(0)\| = \|W_{k+1}(0) - W_k(0)\| \leq \frac{\nu}{2e} L^{-1/\gamma-\beta} \quad \forall k = 0, \dots, L-2.$$

Remark 3.1.2. At first glance, Assumption 3.1.1 (ii) seem rather restrictive. However, we can easily see that $\sigma(z) = \tanh(z)$ satisfy Assumption 3.1.1 (ii). Moreover, the constraint on $\|\sigma^{(1)}\| \leq 1$ and $\sigma^{(1)}$ to be 1-Lipschitz may be further generalized to any positive constant $C > 0$, i.e. $\|\sigma^{(1)}\| \leq C$ and $\sigma^{(1)}$ is C-Lipschitz. Since our main goal is an asymptotic estimate for L approaching ∞, and the constant term C, will appear only as a constant term multiplied by a $\mathcal{O}(L^{-\varepsilon})$ for $\varepsilon > 0$, in the iterative formulas of Lemma 3.2.3 and Lemma 3.2.4 under Assumption 3.1.1 (iii).

3.2 General inference and gradient bounds

This section describes the good properties of the model (2.4) with respect to the norm described in section II. Moreover, in case Assumption 3.1.1 (iii) is satisfied, it

is easy to check that (3.1) and (3.2) respectively establish that the model's output and the gradient are "non-exploding".

Lemma 3.2.1 (Boundedness of hidden layers)**.** *Let $W^{(L)} \in \mathbb{R}^{L\times d\times d}$ be the weights of a ResNet as described above. Under Assumption 3.1.1 (ii), it holds*

$$\|h_k^x\| \leq \|x\| \exp\left(\delta_L \sum_{i=0}^{k-1} \|W_i^{(L)}\|\right), \quad k = 0, \dots, L \tag{3.1}$$

$$\|M_k^x\| \leq \sqrt{d} \exp\left(\delta_L \sum_{i=1}^{L-k} \|W_{L-i}^{(L)}\|\right), \quad k = 0, \dots, L \tag{3.2}$$

Proof. For ease of notation we identify $W = W^{(L)}$ and $h_k^x = h_k$ for each $k \in \{0, \dots L\}$. The proof is a simple matter of adding and subtracting each hidden layer up to the initial input,

$$\begin{aligned}\log \|h_k\| &= \log \|x\| + \sum_{i=1}^{k} \log\left(\frac{\|h_i\|}{\|h_{i-1}\|}\right) \\ &\leq \log \|x\| + \sum_{i=1}^{k} \log\left(1 + \delta_L \frac{\|\sigma(W_{i-1}h_{i-1})\|}{\|h_{i-1}\|}\right).\end{aligned}$$

Because $|\sigma(z)| \leq |z|$, we conclude that $\|\sigma(W_{i-1}h_{i-1})\| \leq \|W_{i-1}\|\|h_{i-1}\|$, hence using the inequality $\log(1+x) \leq x$, we achieve the bound

$$\log \|h_k\| \leq \log \|x\| + \delta_L \sum_{i=0}^{k-1} \|W_i\|.$$

Taking the exponential gives the claim.
The proof of the boundedness of $\|M_k^x\|$ follows in the same spirit, by considering the estimate against the canonical orthonormal basis e_m, where m ranges between $1, \dots, d$,

$$\begin{aligned}\|M_k^x e_m\| &= \left\| \prod_{i=1}^{L-k} (e_m + \delta_L \operatorname{diag} \nabla\sigma(W_{L-i}h_{L-i}^x) W_{L-i} e_m) \right\| \\ &\leq \prod_{i=1}^{L-k} \left\| e_m + \delta_L \operatorname{diag} \nabla\sigma(W_{L-i}h_{L-i}^x) W_{L-i} e_m \right\| \\ &\leq \prod_{i=1}^{L-k} 1 + \delta_L \left\| \operatorname{diag} \nabla\sigma(W_{L-i}h_{L-i}^x) W_{L-i} e_m \right\| \\ &\leq \prod_{i=1}^{L-k} 1 + \delta_L \|W_{L-i}\| \leq \exp\left(\sum_{i=1}^{L-k} \delta_L \|W_{L-i}\|\right),\end{aligned}$$

where in the last step we used the trivial inequality $\log(1+x) \leq x$. Therefore, $\|M_k^x\| \leq \sqrt{d}\exp\left(\sum_{i=1}^{L-k} \delta_L \|W_{L-i}\|\right)$ as claimed. □

Lemma 3.2.2. *Let $L \geq 3$ and $W^{(L)} \in \mathbb{R}^{L\times d\times d}$ the weights of a ResNet as described above. Under Assumption 3.1.1 (ii) and (iii), it holds*

$$\|h_k^x\| \geq \|x\| \exp\left(\delta_L \sum_{j=0}^{k-1} \|W_j^{(L)}\| \left(1 - 4\delta_L \|W_j^{(L)}\| - 2\delta_L^3 \|W_j^{(L)}\|^3\right)\right), \quad k = 1, \dots, L \tag{3.3}$$

Proof. For ease of notation we identify $W = W^{(L)}$ and $h_k^x = h_k$ for each $k \in \{0, \dots L\}$. The proof of the lower bound (3.3) requires a more careful estimate than the upper bound proof in Lemma 3.2.1, albeit the main idea stays unchanged. By decomposing the ℓ^2 norm in $\mathbb{R}^d$ into an inner-product, i.e. $\|h_j\|^2 = \langle h_j, h_j\rangle$, it holds

$$\begin{aligned}
\log\|h_k\| &= \log\|x\| + \frac{1}{2}\sum_{j=1}^{k} \log\left(\frac{\|h_j\|^2}{\|h_{j-1}\|^2}\right) \\
&= \log\|x\| + \frac{1}{2}\sum_{j=1}^{k} \log\left(1 + \underbrace{\frac{2\delta_L}{\|h_{j-1}\|^2}\langle h_{j-1}, \sigma(W_{j-1}h_{j-1})\rangle + \delta_L^2 \frac{\|\sigma(W_{j-1}h_{j-1})\|^2}{\|h_{j-1}\|^2}}_{\Delta_j}\right),
\end{aligned}$$

Noting that $\|h_j\|^2\Delta_j \geq 2\delta_L \langle \sigma(W_{j-1}h_{j-1}), h_{j-1}\rangle$, then by Cauchy-Schwarz and the assumption $|\sigma(z)| \leq |z|$, we conclude

$$\Delta_j \geq -2\delta_L\|W_{j-1}\| \geq -2/3, \tag{3.4}$$

because $\|W\|_\infty \leq L^{-\beta}$, $\alpha + \beta = 1$ and $L \geq 3$. Thanks to (3.4) the inequality $\log(1+\Delta_j) \geq -2\Delta_j^2 + \Delta_j$ is satisfied. Moreover,

$$\Delta_j \leq 2\delta_L\|W_{j-1}\| + \delta_L^2\|W_{j-1}\|^2,$$

hence, $\Delta_j^2 \leq 4\delta_L^2\|W_{j-1}\|^2 + 2\delta_L^4\|W_{j-1}\|^4$, and

$$\begin{aligned}
\log\|x\| + \frac{1}{2}\sum_{j=1}^{k}\log(1+\Delta_j) &\geq \log\|x\| + \frac{1}{2}\sum_{j=1}^{k} -2\Delta_j^2 + \Delta_j \\
&\geq \log\|x\| - \delta_L \sum_{j=0}^{k-1} \|W_j\|\left(1 + 4\delta_L\|W_j\| + 2\delta_L^3\|W_j\|^3\right).
\end{aligned}$$

Finally, we conclude that

$$\|h_k\| \geq \|x\| \exp\left(-\delta_L \sum_{j=0}^{k-1} \|W_j\| \left(1 + 4\delta_L \|W_j\| + 2\delta_L^3 \|W_j\|^3\right)\right). \qquad \square$$

Before delving into the main results regarding the regularity of the path described by the weights of a Residual network, we need to consider the effect of gradient descent on the weights.

Lemma 3.2.3. *Let $W^{(L)} \in \mathbb{R}^{L\times d\times d}$ and denote the gradient descent update as described in* (2.37)*. Under Assumption 3.1.1 (ii), it holds*

$$\begin{aligned}\|W_k^{(L)}(t+1)\| &\leq \|W_k^{(L)}(t)\| \\ &+ \eta(t)\delta_L\sqrt{d} \max_{j=1,\dots,N} \|M_{k+1}^{x_j}(W(t))\| \|h_k^{x_j}\| \sqrt{\frac{1}{N}\sum_{i=1}^{N} \|\nabla_{\widehat{y}}\ell(y_i, \widehat{y}_i(W(t)))\|^2}\end{aligned} \tag{3.5}$$

denoting

$$G_{k,\infty}^t = \sqrt{d} \max_{j=1,\dots,N} \|M_{k+1}^{x_j}(W(t))\| \sqrt{\frac{1}{N}\sum_{i=1}^{N} \|\nabla_{\widehat{y}}\ell(y_i, \widehat{y}_i(W(t)))\|^2}, \tag{3.6}$$

we have

$$\|W_k^{(L)}(t+1)\| \leq \|W_k^{(L)}(t)\| + \eta(t)\delta_L G_{k,\infty}^t \max_{j=1,\dots,N} \|h_k^{x_j}(W(t))\| \tag{3.7}$$

Proof. It is easy to see that once the term

$$\|\nabla_k J(W^{(L)}(t))\|^2,$$

is bounded, we prove the claim. For ease of notation, we denote with $W_k^{(L)}(t) = W_k$ for $k = 0, \dots, L-1$, unless otherwise specified, because we are dealing at a fixed learning step t. The following inequalities are given by triangle inequality, Jensen's and properties of the euclidean norm in the appropriate space,

$$\begin{aligned}\|\nabla_k J(W(t))\|^2 &= \left\| \frac{1}{N}\sum_{i=1}^{N} \nabla_{\widehat{y}}\ell(y, \widehat{y})^\top M_{k+1} \frac{\partial h_{k+1}}{\partial W_k} \right\|^2 \\ &\leq \frac{1}{N}\sum_{i=1}^{N} \left\| \nabla_{\widehat{y}}\ell(y, \widehat{y})^\top M_{k+1} \frac{\partial h_{k+1}}{\partial W_k} \right\|^2 \\ &\leq \frac{1}{N}\sum_{i=1}^{N} \|\nabla_{\widehat{y}}\ell(y_i, \widehat{y}_i)\|^2 \|M_{k+1}^{x_i}\|^2 \left\| \frac{\partial h_{k+1}^{x_i}(W(t))}{\partial W_k(t)} \right\|^2.\end{aligned}$$

We know from (2.35) and Assumption 3.1.1 (ii), that

$$\left\|\frac{\partial h_{k+1}}{\partial W_k}\right\| = \left\|\delta_L \sum_{i,l=1}^{d} \sigma^{(1)}(W_{k,i}h_k)h_{k,l}e_l \otimes e_i \otimes e_i\right\| \\ \leq d\delta_L^2 \|\sigma^{(1)}\|_\infty^2 \|h_k\|^2 \leq d\delta_L^2 \|h_k\|^2,$$

substituting it in the previous computation, it holds

$$\frac{1}{N}\sum_{i=1}^{N} \|\nabla_{\widehat{y}}\ell(y_i,\widehat{y}_i)\|^2 \|M_{k+1}^{x_i}\|^2 \left\|\frac{\partial h_{k+1}^{x_i}(W(t))}{\partial W_k(t)}\right\|^2 \\ \leq \frac{1}{N}\sum_{i=1}^{N} \|\nabla_{\widehat{y}}\ell(y_i,\widehat{y}_i)\|^2 \|M_{k+1}^{x_i}\|^2 d\delta_L^2 \|h_k^{x_i}\|^2.$$

Finally, taking the uniform estimate with respect to the input data of the hidden states h^{x_i} and M^{x_i},

$$\|\nabla_k J(W(t))\|^2 \leq d\delta_L^2 \max_{j=1,\dots,N} \|M_{k+1}^{x_j}\|^2 \|h_k^{x_j}\|^2 \frac{1}{N}\sum_{i=1}^{N} \|\nabla_{\widehat{y}}\ell(y_i,\widehat{y}_i)\|^2$$

By triangle inequality, then

$$\begin{aligned}\|W_k(t+1)\| &\leq \|W_k(t) + \eta(t)\nabla_k J(W(t))\| \\ &\leq \|W_k(t)\| + \eta(t)\|\nabla_k J(W(t))\| \\ &\leq \|W_k(t)\| + \sqrt{d}\delta_L \eta(t) \max_{j=1,\dots,N} \|M_{k+1}^{x_j}\| \|h_k^{x_j}\| \sqrt{\frac{1}{N}\sum_{i=1}^{N} \|\nabla_{\widehat{y}}\ell(y_i,\widehat{y}_i)\|^2},\end{aligned}$$

proving the claim. □

Lemma 3.2.4 (Layers weights increase relation via training)**.** *Let $W^{(L)} \in \mathbb{R}^{L\times d\times d}$. Let the weights $W^{(L)}$ evolve using the gradient descent dynamics as described in* (2.37)*. Under Assumption 3.1.1 (ii), it holds*

$$\|W_{k,k+1}(t+1)\| \leq \|W_{k,k+1}(t)\| \left(1 + \eta(t)\delta_L 2\sqrt{2} G_{k+1,\infty}^t \max_{j=1,\dots,N} \|h_k^{x_j}(W(t))\|^2\right) \\ + \eta(t)\delta_L^2 \sqrt{2} G_{k+1,\infty}^t \max_{j=1,\dots,N} \|h_k^{x_j}(W(t))\| \\ \cdot \left(2\|W_{k+1}(t)\|\|W_k(t)\|\|h_k^{x_j}(W(t))\| + \sqrt{2}\|W_k(t)\| + \|W_{k+1}(t)\|^2 \|h_{k+1}^{x_j}(W(t))\|\right). \tag{3.8}$$

where we denoted $G_{k,\infty}^t$ as in (3.6)*.*

Proof. In the following, we omit the training time, as it is always related to a fixed time t, e.g. $W_k(t) = W_k$. The proof consist in proving an upper bound on

$$\|\nabla_{k+1} J(W(t)) - \nabla_k J(W(t))\|^2,$$

then, substitute it in the following triangle inequality estimate,

$$\|W_{k,k+1}(t+1)\| \leq \|W_{k,k+1}(t)\| + \eta(t)\|\nabla_{k+1} J(W(t)) - \nabla_k J(W(t))\|.$$

because $W_k(t+1) = W_k(t) - \eta(t)\nabla_k J(W(t))$ by gradient descent update rule. Noting that

$$\begin{aligned}
&\nabla_{k+1} J(W(t)) - \nabla_k J(W(t)) \\
&= \frac{1}{N}\sum_{i=1}^{N} \nabla \ell(y_i, \widehat{y}_i)^\top M_{k+2}^{x_i} \left(\frac{\partial h_{k+2}^{x_i}}{\partial W_{k+1}} - \frac{\partial h_{k+1}^{x_i}}{\partial W_k} - \delta_L \operatorname{diag} \nabla\sigma(W_{k+1} h_{k+1}^{x_i}) W_{k+1} \frac{\partial h_{k+1}^{x_i}}{\partial W_k} \right).
\end{aligned}$$

It is easy to estimate the residual of the hidden states,

$$\begin{aligned}
&\frac{\partial h_{k+2}^{x_i}}{\partial W_{k+1}} - \frac{\partial h_{k+1}^{x_i}}{\partial W_k} \\
&= \delta_L \sum_{n,m=1}^{d} \left\{ \sigma^{(1)}\left(W_{k+1,n} h_{k+1}^{x_i}\right) h_{k+1,m}^{x_i} - \sigma^{(1)}\left(W_{k,n} h_k^{x_i}\right) h_{k,m}^{x_i} \right\} e_m \otimes e_n \otimes e_n \\
&= \delta_L \sum_{n,m=1}^{d} \left\{ (\sigma^{(1)}\left(W_{k+1,n} h_{k+1}^{x_i}\right) - \sigma^{(1)}\left(W_{k,n} h_k^{x_i}\right)) h_{k,m}^{x_i} + \right. \\
&\qquad \left. + \delta_L \sigma(W_{k,m} h_k^{x_i}) \sigma^{(1)}(W_{k+1,n} h_{k+1}^{x_i}) \right\} e_m \otimes e_n \otimes e_n
\end{aligned}$$

where we used the equality $h_{k+1} = h_k + \delta_L \sigma(W_k h_k)$. Using triangle inequality and the properties of the norm,

$$\begin{aligned}
&\left\| \frac{\partial h_{k+2}^{x_i}}{\partial W_{k+1}} - \frac{\partial h_{k+1}^{x_i}}{\partial W_k} \right\| \\
&\leq \delta_L \left\| \sum_{n,m=1}^{d} (\sigma^{(1)}\left(W_{k+1,n} h_{k+1}^{x_i}\right) - \sigma^{(1)}\left(W_{k,n} h_k^{x_i}\right)) h_{k,m}^{x_i} e_m \otimes e_n \otimes e_n \right\| \\
&\qquad + \delta_L^2 \left\| \sum_{m,n=1}^{d} \sigma(W_{k,m} h_k^{x_i}) \sigma^{(1)}(W_{k+1,n} h_{k+1}^{x_i}) e_m \otimes e_n \otimes e_n \right\| \\
&\leq \delta_L \left\| \sum_{n,m=1}^{d} (\sigma^{(1)}\left(W_{k+1,n} h_{k+1}^{x_i}\right) - \sigma^{(1)}\left(W_{k,n} h_k^{x_i}\right)) e_m \otimes e_n \otimes e_n \right\| \|h_k^{x_i}\| \\
&\qquad + \delta_L^2 \left\| \sum_{m,n=1}^{d} \sigma(W_{k,m} h_k^{x_i}) \sigma^{(1)}(W_{k+1,n} h_{k+1}^{x_i}) e_m \otimes e_n \otimes e_n \right\|
\end{aligned}$$

Finally, noting that

$$\left\| \sum_{n,m=1}^{d} (\sigma^{(1)}\left(W_{k+1,n} h_{k+1}^{x_i}\right) - \sigma^{(1)}\left(W_{k,n} h_k^{x_i}\right)) e_m \otimes e_n \otimes e_n \right\| \leq \sqrt{d} \left\| \nabla\sigma\left(W_{k+1} h_{k+1}^{x_i}\right) - \nabla\sigma\left(W_k h_k^{x_i}\right) \right\| \quad (3.9)$$

Using the hypothesis on the regularity of the activation functions, we can bound the second term with

$$\delta_L^2 \left\| \sum_{m,n=1}^{d} \sigma^{(1)}(W_{k+1,n} h_{k+1}^{x_i}) \sigma(W_{k,m} h_k^{x_i}) e_m \otimes e_n \otimes e_n \right\| \leq \delta_L^2 \sqrt{d} \|\sigma(W_k h_k^{x_i})\| \leq \delta_L^2 \sqrt{d} \|W_k\| \|h_k^{x_i}\|. \quad (3.10)$$

It follows in an analogous manner the estimate of the following term

$$\left\| \delta_L \operatorname{diag} \nabla\sigma(W_{k+1} h_{k+1}^{x_i}) W_{k+1} \frac{\partial h_{k+1}^{x_i}}{\partial W_k} \right\| \leq \sqrt{d} \delta_L^2 \|W_{k+1}\|^2 \|h_{k+1}^{x_i}\| \|h_k^{x_i}\| \quad (3.11)$$

Finally, taking the norm, using the estimates (3.9), (3.10) and (3.11), yields

$$\begin{aligned}
&\|\nabla_{k+1} J(W(t)) - \nabla_k J(W(t))\|^2 \\
&= \left\| \frac{1}{N} \sum_{i=1}^{N} \nabla\ell(y_i, \widehat{y}_i)^\top M_{k+2}^{x_i} \left(\frac{\partial h_{k+2}^{x_i}}{\partial W_{k+1}} - \frac{\partial h_{k+1}^{x_i}}{\partial W_k} - \delta_L \operatorname{diag} \nabla\sigma(W_{k+1} h_{k+1}^{x_i}) W_{k+1} \frac{\partial h_{k+1}^{x_i}}{\partial W_k} \right) \right\|^2 \\
&\leq \frac{1}{N} \sum_{i=1}^{N} \|\nabla\ell(y_i, \widehat{y}_i)\|^2 \|M_{k+2}^{x_i}\|^2 \Big(\left\| \frac{\partial h_{k+2}^{x_i}}{\partial W_{k+1}} - \frac{\partial h_{k+1}^{x_i}}{\partial W_k} \right\|^2 + \left\| \delta_L \operatorname{diag} \nabla\sigma(W_{k+1} h_{k+1}^{x_i}) W_{k+1} \frac{\partial h_{k+1}^{x_i}}{\partial W_k} \right\|^2 \Big) \\
&\leq \frac{1}{N} \sum_{i=1}^{N} \|\nabla\ell(y_i, \widehat{y}_i)\|^2 \|M_{k+2}^{x_i}\|^2 \Big(4\delta_L^2 d \left\| \nabla\sigma\left(W_{k+1} h_{k+1}^{x_i}\right) - \nabla\sigma\left(W_k h_k^{x_i}\right) \right\|^2 \|h_k^{x_i}\|^2 \\
&\qquad + 2d\delta_L^4 \|h_k^{x_i}\|^2 \left(2\|W_k\|^2 + \|W_{k+1}\|^4 \|h_{k+1}^{x_i}\|^2\right) \Big).
\end{aligned}$$

Using the fact that $\sigma^{(1)}(z)$ is 1-Lipschitz, we are able to further bound the first term of the previous estimate by $W_{k,k+1}$

$$\begin{aligned}
\left\| \nabla\sigma\left(W_{k+1} h_{k+1}^{x_i}\right) - \nabla\sigma\left(W_k h_k^{x_i}\right) \right\| &\leq \|W_{k+1} h_{k+1}^{x_i} - W_k h_k^{x_i}\| \\
&\leq \|W_{k,k+1}\| \|h_k^{x_i}\| + \|W_{k+1}\| \|h_{k,k+1}^{x_i}\| \\
&\leq \|W_{k,k+1}\| \|h_k^{x_i}\| + \delta_L \|W_{k+1}\| \|W_k\| \|h_k^{x_i}\|,
\end{aligned}$$

by the previous estimate it holds,

$$\begin{aligned}
&\|\nabla_{k+1}J(W) - \nabla_k J(W)\|^2 \\
&\leq \frac{1}{N}\sum_{i=1}^{N} \|\nabla\ell(y_i,\widehat{y}_i)\|^2\|M_{k+2}^{x_i}\|^2\Big(8\delta_L^2 d\|W_{k,k+1}\|^2\|h_k^{x_i}\|^4 + 8\delta_L^4 d\|W_{k+1}\|^2\|W_k\|^2\|h_k^{x_i}\|^4 \\
&\qquad + 2d\delta_L^4\|h_k^{x_i}\|^2\left(2\|W_k\|^2 + \|W_{k+1}\|^4\|h_{k+1}^{x_i}\|^2\right)\Big).
\end{aligned}$$

Finally, taking the square root and taking the maximum over the input data,

$$\begin{aligned}
&\|\nabla_{k+1}J(W) - \nabla_k J(W)\| \leq \delta_L 2\sqrt{2}G_{k+1,\infty}^t \max_{j=1,\dots,N}\|h_k^{x_j}\|^2\|W_{k,k+1}\| \\
&+\delta_L^2\sqrt{2}G_{k+1,\infty}^t \max_{j=1,\dots,N}\|h_k^{x_j}\|\left(2\|W_{k+1}\|\|W_k\|\|h_k^{x_j}\| + \sqrt{2}\|W_k\| + \|W_{k+1}\|^2\|h_{k+1}^{x_j}\|\right)
\end{aligned}$$

with

$$G_{k,\infty}^t = \sqrt{d}\max_{j=1,\dots,N}\|M_{k+1}^{x_j}(W(t))\|\sqrt{\frac{1}{N}\sum_{i=1}^{N}\|\nabla_{\widehat{y}}\ell(y_i,\widehat{y}_i(W(t)))\|^2}$$

as in Lemma 3.2.3, proving the claim. □

3.3 Main regularity result

Theorem 3.3.3 is the main contribution of this section. However, to ensure its comprehension, we first have to take a detour to establish the key distinctions from prior state of the art result, namely [9, Theorem 3.5]. We restate the result with all its assumption for additional clarity.

Theorem 3.3.1 (Theorem 3.5 of [9])**.** *Let L be large enough. Under the assumption that there exists a constant $c_0 > 0$ such that*

(i) Smooth activation functions: $\sigma \in C^2(\mathbb{R})$, $\sigma^{(1)}(0) = 1$ and for all $z \in \mathbb{R}$ $|\sigma(z)| \leq |z|$, $|\sigma^{(1)}(z)| \leq 1$ and $|\sigma^{(2)}(z)| \leq 1$.

(ii) Scaling factor $\delta_L = L^{-1/2}$.

(iii) Separated unit data: $\|x_i\| = \|y_i\| = 1$ and $\forall i \neq j$,

$$|\langle x_i, x_j\rangle| \leq (8N)^{-1}e^{-4c_0}. \tag{3.12}$$

(iv) The weights are initialized as

$$\sup_{k,m}\|W_{k,m}^{(L)}(0)\| \leq \frac{2^{-9/2}}{\sqrt{dN}}e^{-4.2c_0}L^{-1}. \tag{3.13}$$

(v) Small initial loss:

$$J_L(W^{(L)}(0)) \leq \frac{c_0^2 e^{-8.2c_0}}{2^{15}3^2 dN^2} \tag{3.14}$$

Let the parameters $W^{(L)}(t)$ evolve according to the gradient descent dynamics with learning rate $\eta_L(t)$ until time $T_L \in \mathbb{N}$, chosen in such a way that for each $t = 0, \dots, T_L - 1$, we have

$$\eta_L(t) \leq \frac{e^{-10.5c_0}}{160Nd}, \quad \sum_{t=0}^{T_L-1} \eta_L(t) \leq d^{-1} \log L. \tag{3.15}$$

Then, for each $t = 0, \dots, T_L$, we have

$$\max_{k=1,\dots,L} \|W_k(t)\| \leq c_0 L^{-1/2},$$

$$\max_{k=1,\dots,L-1} \|W_{k+1}(t) - W_k(t)\| \leq \frac{2^{-7/2}}{\sqrt{N}} e^{-4.2c_0} L^{-1}$$

$$J_L(W(t)) \leq \exp\left(-\frac{1}{32N} e^{-2c_0} \sum_{s=0}^{t-1} \eta_L(s)\right) J_0 + 34dc_0^4 e^{6.4c_0} \sum_{s=0}^{t-1} \eta_L(s) L^{-1} J_0 \tag{3.16}$$

with $J_L(W(t))$ being the average of the MSE-loss over the labelled data.

Assumption 3.1.1 (i)-(iii) are rather similar to the assumptions (i),(iii) and (iv) of Theorem 3.3.1. We improved upon the result by using a lower regular activation function σ and generalized the hypothesis (ii) of Theorem 3.3.1 by allowing a scaling $\delta_L = L^{-\alpha}$, for $\alpha \in (1/2, 1]$. Then, a key improvement consists in the removal of the separated data assumption, namely (3.12). Because it implies that the labelled dataset used for the training procedure should be constrained by the inequality $d > N^4$, i.e. the number of features must be far larger than the number of data points in the labelled dataset. However, (3.12) is critical to the local convergence result obtained in [9, Theorem 3.5], namely the upper bound on the loss (3.16). Hence, we were not able to achieve a sharp estimate of $J(W(t))$ in the gradient descent dynamic, albeit irrelevant to our result. Therefore, we used a different approach by assuming a more practical hypothesis, namely

Assumption 3.3.2. Under Assumption 3.1.1 (i) and (iii), there exists a constant $C_\ell > 0$ that satisfies

$$\|\nabla_{\widehat{y}} J(W)\|^2 \leq \frac{1}{N} \sum_{i=1}^{N} \|\nabla_{\widehat{y}} \ell(y_i, \widehat{y}_i(W))\|^2 \leq C_\ell < \infty. \tag{3.17}$$

And obtained a similar result to Theorem 3.3.1

Theorem 3.3.3. *Under Assumption 3.1.1 and Assumption 3.3.2 with* $\nu = 1$, $\gamma \in (1/2\alpha, 1]$ *and* $\alpha, \beta \in [0,1]$ *such that* $\alpha + \beta = 1$ *and* $\alpha > \beta$. *Let* $M > 0$ *such that* $L > M$ *is large enough. The parameters* $W^{(L)}(t)$ *evolve according to the gradient descent algorithm* (2.30) *with learning rate* $\eta(t)$ *and* $T_L \in \mathbb{N}$ *epochs, such that there exists a constant* $Q > 0$ *and for* $t = 0, \dots, T_L - 1$, *it holds*

$$\eta(t) \leq 1, \qquad T_L \leq Q \log L. \tag{3.18}$$

Then, for each $t = 0, \dots, T_L$ *we have*

$$\max_k \|W_k(t)\| \leq L^{-\beta},$$
$$\max_k \|W_{k+1}(t) - W_k(t)\| \leq L^{-1/\gamma - \beta}.$$

Theorem 3.3.3 is freed from the requirement in (3.12) and it is satisfied by a larger class of loss functions. At first glance, Assumption 3.3.2 might seem rather restrictive. However, thanks to Lemma 3.2.2 and Lemma 3.2.1 and simple computations, for commonly used loss functions ℓ, Assumption 3.3.2 holds. We first check in case of a generalization of a commonly used loss function for regression tasks. Namely, we define the p-loss function with $p \in [2, \infty)$ as

$$\ell_p(y, \widehat{y}) \coloneqq \frac{1}{p}\|y - \widehat{y}\|_p^p,$$

where we defined the p-norm for a vector $v \in \mathbb{R}^d$ as

$$\|v\|_p^p = \sum_{i=1}^{d} |v_i|^p.$$

Then, it is easy to verify that

$$\|\nabla_{\widehat{y}} \ell_p(y, \widehat{y})\|_2 \leq \|\nabla_{\widehat{y}} \ell_p(y, \widehat{y})\|_q, \quad \frac{1}{p} + \frac{1}{q} = 1,$$

because $q \leq 2$ and $\|x\|_2 \leq \|x\|_q$ for any $1 \leq q \leq 2$. Finally, it is straightforward to compute

$$\|\nabla_{\widehat{y}} \ell_p(y, \widehat{y})\|_q^q = p\ell_p(y, \widehat{y}),$$

therefore,

$$\|\nabla_{\widehat{y}} \ell_p(y, \widehat{y})\|_2 \leq (p\ell_p(y, \widehat{y}))^{1/q},$$

proving that

$$\frac{1}{N}\sum_{i=1}^{N} \|\nabla_{\widehat{y}} \ell_p(y_i, \widehat{y}_i)\|_2^2 \leq \frac{1}{N}\sum_{i=1}^{N} (p\ell_p(y_i, \widehat{y}_i))^{2/q}$$

as long as the following estimate holds for $\delta_L \sum_{k=0}^{L-1} \|W_k\| \leq 1$,

$$\frac{1}{N}\sum_{i=1}^{N}(p\ell_p(y_i,\widehat{y}_i))^{2/q} \leq \frac{2^{(2p-2)/q}}{N}\sum_{i=1}^{N}(\|y_i\|_p^p + \|\widehat{y}_i\|_p^p)^{2/q} \leq C < \infty,$$

by (3.1) of Lemma 3.2.1.

An analogous result could be proved for Cross Entropy, namely using the one-hot encoding for each label $y \in \mathbb{R}^C$. Namely if y represents the j-th class, then

$$y_i = \begin{cases} 0 & \text{if } i \neq j, \\ 1 & \text{otherwise.} \end{cases}$$

We denote with $[y] = j \in \{1, \dots, C\}$ where C represents the number of different classes. The Cross-Entropy loss is expressed as the average over the losses ℓ_{CE}

$$\ell_{\text{CE}}(\widehat{y}^n, y^n) = -\frac{1}{C}\log\left(\frac{\exp(\widehat{y}^n_{[y^n]})}{\sum_{c=1}^{C}\exp(\widehat{y}^n_c)}\right),$$

for each data point $n \in \{1, \dots, N\}$. Therefore, noting that for each $k \in \{1, \dots, C\}$,

$$\begin{aligned}\left|\frac{\partial}{\partial \widehat{y}^n_k}\ell_{\text{CE}}(\widehat{y}^n, y^n)\right|^2 &= \left|-\frac{\chi_{\{k=[y^n]\}}}{C}\left(1 - \frac{\exp(y_k)}{\sum_{c=1}^{C}\exp(y_c)}\right) + \frac{\chi_{\{k\neq[y^n]\}}}{C}\left(\frac{\exp(y_k)}{\sum_{c=1}^{C}\exp(y_c)}\right)\right|^2 \\ &\leq \frac{4}{C^2}\max\left\{\left|1 - \frac{\exp(y_k)}{\sum_{c=1}^{C}\exp(y_c)}\right|^2, \left|\frac{\exp(y_k)}{\sum_{c=1}^{C}\exp(y_c)}\right|^2\right\} \\ &\leq \frac{4}{C^2}.\end{aligned}$$

Hence,

$$\|\nabla_{\widehat{y}^n}\ell_{\text{CE}}(\widehat{y}^n, y^n)\|^2 \leq \frac{4}{C}$$

and taking the average, we conclude

$$\frac{1}{N}\sum_{n=1}^{N}\|\nabla_{\widehat{y}^n}\ell_{\text{CE}}(\widehat{y}^n, y^n)\|^2 \leq \frac{4}{C} < \infty.$$

We note that in case of the Cross Entropy loss, the assumption on the scaling on W is altogether not necessary. In Assumption 3.1.1, we assumed only that $\|x_i\|, \|y_i\| \leq 1$, but without a huge loss of the dataset generality, we could additionally assume the existence of $K > 0$ such that $\|y_i\|\|x_i\|^{-1} \leq K$. This assumption is not restrictive as we can always scale the y_is by $K \min_{i=1,\dots,N}\|x_i\|$,

unless $\min_{i=1,\dots,N} \|x_i\| = 0$. In this regime, we can prove that the Kullback-Leibler divergence is another loss function satisfying Assumption 3.3.2. By definition,

$$\ell_{\mathrm{KL}}(y_i, \widehat{y}_i) \coloneqq y_i \log\left(\frac{y_i}{\widehat{y}_i}\right), \tag{3.19}$$

hence

$$\frac{\mathrm{d}}{\mathrm{d}\widehat{y}_i}\ell_{\mathrm{KL}}(y_i, \widehat{y}_i) = -\frac{y_i}{\widehat{y}_i}. \tag{3.20}$$

But it is easy to check that $|\widehat{y}_i|^2 \geq \|x_i\|^2 e^{-6}$ since

$$\|h_L^{x_i}\| \geq \|x_i\| \exp\left(-1 - 4L^{-1} - 2L^{-3}\right) \geq \|x_i\| \exp(-3). \tag{3.21}$$

where we used the statement of Lemma 3.2.2. Hence, we require $L \geq 3$ and $W^{(L)}$ to satisfy Assumption 3.1.1 (iii) for $\beta + \alpha = 1$, but there is no limit on γ and ν. Therefore, under the assumption that there exists $K > 0$ such that $\|y_i\|\|x_i\|^{-1} \leq K$ for any choice of $i = 1, \dots, N$, we conclude

$$\frac{1}{N}\sum_{i=1}^{N}\left(\frac{\mathrm{d}}{\mathrm{d}\widehat{y}_i}\ell_{\mathrm{KL}}(y_i, \widehat{y}_i)\right)^2 \leq Ke^6 < \infty. \tag{3.22}$$

From this example, we observe that the rescaling of the dataset is quite relevant to Assumption 3.3.2. Moreover, it hints to the possibility that the result of Theorem 3.3.3 may be further generalized to a wider class of loss functions by using slightly stronger assumptions on the dataset. Before proving the result Theorem 3.3.3, we need to prove the following trivial lemma.

Lemma 3.3.4 (Discrete Grönwall inequality)**.** *Let* $\{u_n\}, \{v_n\}, \{w_n\} \subset \mathbb{R}_{\geq 0}$. *If* $e_{n+1} \leq u_n e_n + v_n$ *for each* $n > 0$*, then*

$$e_n \leq \left(\prod_{n'=0}^{n-1} u_{n'}\right) e_0 + \sum_{n'=0}^{n-1}\left(\prod_{n''=n'+1}^{n-1} u_{n''}\right) v_{n'}$$

Proof. By iteratively rewriting the relation $e_{n+1} \leq u_n e_n + v_n$ for each $m \leq n$, yields the claim. □

We have proved that Assumption 3.3.2 is general enough for most practical purposes, thus we conclude this section with the proof of Theorem 3.3.3.

Proof of Theorem 3.3.3. We assume L fixed and large enough. For $t = 0$, the conditions are trivially satisfied. Thus, we can start the inductive proof by iterating through the 2 statements.

Bound on layers norm. By the recursive relation in Lemma 3.2.3,

$$\begin{aligned}\|W_k(t+1)\| &\leq \|W_k(0)\| + \sum_{s=0}^{t} \eta(s) L^{-\alpha} G^s_{k,\infty} \max_{j=1,\dots,N} \|h^{x_j}_k(s)\| \\ &\leq \frac{1}{2} L^{-\beta} + L^{-\alpha} \sum_{s=0}^{t} \eta(s) d\sqrt{C_\ell} \exp(2) \\ &\leq \frac{1}{2} L^{-\beta} + L^{-\alpha} Q \log(L) d\sqrt{C_\ell} \exp(2). \end{aligned} \tag{3.23}$$

The estimate in (3.23) comes from Lemma 3.2.1 applied with the inductive hypothesis that $\|W(s)\|_\infty \leq L^{-\beta}$ for $s < t+1$ and the fact that $\alpha+\beta = 1$. Moreover, in a similar manner we obtain $G^s_{k,\infty} < d\sqrt{C_\ell} e < \infty$ uniformly over $s < t$ and $k = 0, \dots, L-1$ by Lemma 3.2.1 and the inductive hypothesis on $\|W_k(s)\|$. Clearly the logarithmic term comes from the assumption $t+1 \leq Q \log L$ from (3.18). Finally, from (3.23) and applying the previous estimates on $G^s_{k,\infty}$, we can choose L large enough that the following lower bound is satisfied

$$L^{-\alpha+\beta} \log(L) \leq \frac{1}{2Q} \frac{1}{2d\sqrt{C_\ell} \exp(2)}. \tag{3.24}$$

Because $L^{-\alpha+\beta} = O(L^{-\varepsilon})$ for some $\varepsilon > 0$ and $\log L = o(L^\rho)$ for any choice of $\rho > 0$. Therefore, we can find L large enough to satisfy the inequality. Concluding that

$$\|W_k(t+1)\| \leq L^{-\beta}.$$

Moreover, the bound (3.24) is uniform in the training step, hence we can fix L large enough throughout the proof by induction.

Bound on layers' increments. Using the discrete Grönwall inequality (Lemma 3.3.4), with respect to the recurrent relation of Lemma 3.2.4, which we report for clarity

$$\begin{aligned}&\|W_{k,k+1}(t+1)\| \leq \|W_{k,k+1}(t)\| \left(1 + \eta(t)\delta_L 2\sqrt{2} G^t_{k+1,\infty} \max_{j=1,\dots,N} \|h^{x_j}_k\|^2\right) \\ &+\eta(t)\delta_L^2 \sqrt{2} G^t_{k+1,\infty} \max_{j=1,\dots,N} \|h^{x_j}_k\| \left(2\|W_{k+1}\|\|W_k\|\|h^{x_j}_k\| + \sqrt{2}\|W_k\| + \|W_{k+1}\|^2 \|h^{x_j}_{k+1}\|\right).\end{aligned}$$

we obtain,

$$\|W_{k,k+1}(t+1)\| \leq \prod_{s=0}^{t}(1+\eta(s)L^{-\alpha}2\sqrt{2}G^{s}_{k+1,\infty}\max_{i=j,\dots,N}\|h^{x_j}_k(s)\|^2)\|W_{k,k+1}(0)\|+$$
$$+\sum_{s=0}^{t}\left(\prod_{r=s+1}^{t}(1+\eta(r)L^{-\alpha}2\sqrt{2}G^{r}_{k+1,\infty}\max_{i=j,\dots,N}\|h^{x_j}_k(r)\|^2)\right)\cdot$$
$$\cdot\,\eta(s)\delta_L^2\sqrt{2}G^{s}_{k+1,\infty}\max_{j=1,\dots,N}\|h^{x_j}_k(s)\|\Big(2\|W_{k+1}(s)\|\|W_k(s)\|\|h^{x_j}_k(s)\|$$
$$+\sqrt{2}\|W_k(s)\|+\|W_{k+1}(s)\|^2\|h^{x_j}_{k+1}(s)\|\Big).$$

The previous inequality is quite unmanageable. For this reason, we use the trivial inequality $1+x\leq\exp(x)$, it yields

$$\prod_{s=0}^{t}(1+\eta(s)L^{-\alpha}2\sqrt{2}G^{s}_{k+1,\infty}\max_{i=j,\dots,N}\|h^{x_j}_k(s)\|^2)\|W_{k,k+1}(0)\|$$
$$\leq\exp\left(\sum_{s=0}^{t}\eta(s)L^{-\alpha}2\sqrt{2}G^{s}_{k+1,\infty}\max_{i=j,\dots,N}\|h^{x_j}_k(s)\|^2\right).$$

By inductive hypothesis, $\|W_k(s)\|\leq L^{-\beta}$ for each $k\in\{0,\dots,L-1\}$ and, $s\in\{0,\dots,t\}$. Therefore, we can easily deduce using Lemma 3.2.1, that it holds

$$\|M^{x_j}_k(s)\|\leq\sqrt{d}e,\quad \|h^{x_j}_k(s)\|\leq e,$$

for $s\in\{0,\dots,t\}$, $k\in\{0,\dots,L\}$. Moreover, by the previous estimates $G^{s}_{k+1,\infty}\leq de\sqrt{C_\ell}$. Hence, it yields

$$\|W_{k,k+1}(t+1)\|\leq\exp\left(2d\sqrt{2C_\ell}e^3QL^{-\alpha}\log(L)\right)\Big(\|W_{k,k+1}(0)\|$$
$$+d\sqrt{2C_\ell}e^2QL^{-2\alpha-\beta}(3eL^{-\beta}+\sqrt{2})\log(L)\Big),$$

where we used the fact that $\eta(s)\leq 1$ and $t\leq Q\log L$. By the initial condition on the increments, i.e. $\|W_{k,k+1}(0)\|\leq\frac{1}{2e}L^{-1/\gamma-\beta}$, we have

$$\|W_{k,k+1}(t+1)\|\leq\exp\left(2d\sqrt{2C_\ell}e^3QL^{-\alpha}\log(L)\right)\Big(\frac{1}{2e}L^{-1/\gamma-\beta}$$
$$+d\sqrt{2C_\ell}e^2QL^{-2\alpha-\beta}(3eL^{-\beta}+\sqrt{2})\log(L)\Big).$$

Finally, we observe that for L large enough,

$$\exp\left(2d\sqrt{2C_\ell}e^3QL^{-\alpha}\log(L)\right)\leq e\iff L^{-\alpha}\log(L)\leq\frac{1}{2d\sqrt{2C_\ell}e^3Q}.$$

Analogously, for $1/\gamma < 2\alpha$, there exists L large enough that

$$\frac{1}{2} + d\sqrt{2C_\ell}e^3 Q L^{-2\alpha+1/\gamma}(3eL^{-\beta} + \sqrt{2})\log(L) \leq 1$$

that is,

$$L^{-2\alpha+1/\gamma}\log(L) \leq \frac{1}{2d\sqrt{2C_\ell}e^3 Q(3eL^{-\beta} + \sqrt{2})}. \tag{3.25}$$

Hence, for L large enough, we conclude

$$\|W_{k,k+1}(t+1)\| \leq L^{-1/\gamma-\beta}$$

Finally, noting that the bounds in L are uniform for each training step and we proved the induction step for each statement, we conclude the proof of the claims by induction. □

Remark 3.3.5. We note that under the condition $\eta(s) \lesssim L^{\alpha-\beta-\varepsilon}$ for any $\varepsilon > 0$ and $s \leq T_L$, the boundedness results as claimed in Theorem 3.3.3 is still satisfied. Indeed, from the proof of Theorem 3.3.3, we require

$$L^{-\alpha+\beta}\sum_{s=1}^{Q\log L}\eta(s) \lesssim 1,$$
$$L^{-2\alpha+\frac{1}{\gamma}}\sum_{s=1}^{Q\log L}\eta(s) \lesssim 1,$$

from (3.24) and (3.25) respectively and noting that the $Q\log L$ term was obtained from $\sum_{s=1}^{Q\log L}\eta(s)$. Therefore, if $\gamma \in (1/2\alpha, 1]$, which will be a common requirement for the regularity results in the next sections, a practical choice of $\eta(s)$ is given by $L^{\alpha-\beta-\varepsilon}$.

Theorem 3.3.3 implies that distance between two layers stay bounded if they are bounded at initialization and we train the neural network less than $Q\log L$. This is far from ideal, because it was proved in [1] that to achieve a small error with sufficiently high probability, a polynomial in L of time of steps might be necessary. The property described in Theorem 3.3.3 is conserved as L approaches infinity. Therefore in case $\alpha = 1$, if the sequence $W^{(L)}$ were to converge to a path in $W^* \in C([0,1], \mathbb{R}^{d^2})$, its 1-variation should be bounded in order to be able to apply the existence and uniqueness result described in section 2.1. In the next section, we try with two different interpolation of the path of the weights to obtain general results on the convergence of a sub-sequence under the weak regime described by Assumption 3.1.1 and Assumption 3.3.2.

3.4 Weights regularity I

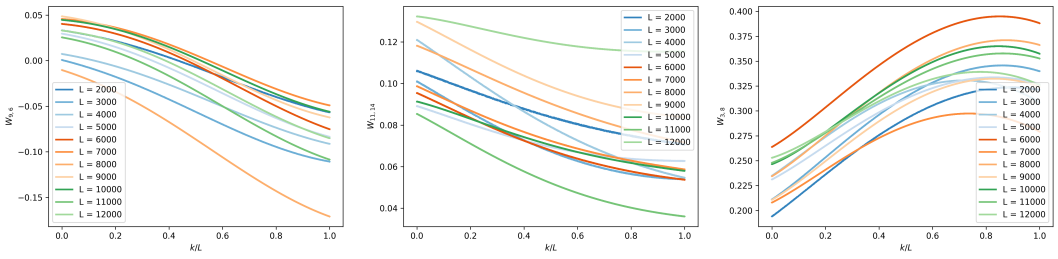
Theorem 3.3.3 enables us to prove the estimates in the depth-limit of the 1-variation of the parameters $W^{(L)}$ regarded as trajectories indexed by layer in the studied regime of Assumption 3.1.1 and Assumption 3.3.2. The most simple method is to consider the trajectories as $\mathcal{W}_t^{(L)} := L^\beta W^{(L)}_{\lfloor (L-1)t \rfloor}(T_L)$ for $t \in [0,1]$ for $L > 1$. Moreover, we denote with $t_k^L = {}^k/_{(L-1)}$ for $k = 0, \dots, L-1$, thus $\mathcal{W}^{(L)}_{t_k^L} = L^\beta W_k(T_L)$. In order to provide more context to the results of this section, we present it in the context of [9, Proposition 3.7], which we report for clarity, noting that it shares the same assumptions of Theorem 3.3.1:

Proposition 3.4.1 (Proposition 3.7 of [9])**.** *Let $W^{(L)}(t)$ be the parameters following the gradient descent dynamics. Assume there exists $W^* := [0,1] \to \mathbb{R}^{d\times d}$ such that*

$$\sup_{s\in[0,1]} L^{1/2} \| L^{1/2} W^{(L)}_{\lfloor Ls \rfloor}(T_L) - W^*_s \| \to 0, \quad \text{as } L \to \infty. \tag{3.26}$$

Then, the scaling limit W^* has finite 2-variation.

The condition (3.26) is rather strong as observed in [9], and we were not able to consistently verify it. It is clear that visual representations might suggest the assumption (3.26), e.g. see that the paths in Figure 3.1 are following a similar trajectory[1]. However, since we were not able to verify it formally nor empirically



(a) Paths of $L^\beta W^{(L)}_{9,6}(T_L)$ for the hyper-parameters CIFAR10-B12t.

(b) Paths of $L^\beta W^{(L)}_{11,14}(T_L)$ for the hyper-parameters CIFAR10-B22t.

(c) Paths of $L^\beta W^{(L)}_{3,8}(T_L)$ for the hyper-parameters CIFAR10-B33t.

Figure 3.1: Example trajectories of random weights in case of CIFAR10 models with the specified hyper-parameters as in Table 4.1. The trajectories are rescaled by L^β for the respective value of β. On the x-axis is specified the layer normalized by the depth of the network, while on the y-axis the value of the respective coordinate of the path after T_L training steps.

[1] We dismiss the methodology and additional images to section III for cohesiveness.

in a robust manner, we decided to not use (3.26), but prefer to prove weaker results with respect to $\mathcal{W}^{(L)}$, e.g. Corollary 3.4.5. However, a partial explanation of this behaviour is discussed in Theorem 3.4.7 by using a different definition of the trajectories described by the weights.

Theorem 3.4.2. *Under Assumption 3.1.1 and Assumption 3.3.2 with* $\nu = 1$, $\gamma \in (1/2\alpha, 1]$ *and* $\alpha > 1/2$. *Then, there exists* $K > 0$ *such that*

$$\sup_{L\in\mathbb{N}_0} \|\mathcal{W}^{(L)}\|_{1-var;[0,1]} < K. \tag{3.27}$$

Proof. We note that either the supremum is reached for some finite $L < M$ for some $M > 0$ or it is approaching the supremum for L approaching infinity. If the supremum is reached for some finite value of L, then clearly the 1-variation is finite since the function $\mathcal{W}_s^L$ for $s \in [0,1]$ is a step-wise function with L increments. Therefore, we are left with the case L approaching infinity. In this case, we note that by Theorem 3.3.3 there exists $M > 0$, such that for any $L > M$,

$$L^\beta \|W_{k+1}^{(L)} - W_k^{(L)}\| \leq L^{-1/\gamma}.$$

Thus, we consider (3.28)

$$\sup_{\pi_\delta \text{ partition of } [0,1]} \sum_{i=0}^{\#\pi_\delta - 1} \left\| \mathcal{W}_{s_i,s_{i+1}}^{(L)} \right\|. \tag{3.28}$$

for a partition π of mesh size of up to $\delta > 0$ for a fixed value of $L > M$. We note that if (3.28) is bounded uniformly in δ for any choice of $\delta > 0$, in particular we have that the 1-variation is bounded. If $\delta > L^{-1}$ then the mesh size of the partition is too large to account for all the increments of the step-function defined by $\mathcal{W}^{(L)}$. Contrariwise, if $\delta \leq L^{-1}$ the partition see all the non-trivial increments, but adds also trivial increments given by $\mathcal{W}_{u,v}^{(L)}$ with $u < v \in (t_k^L, t_{k+1}^L)$ for some $k \in \{0, \dots, L-1\}$. Eventually, we conclude that the largest possible value of (3.28) is achieved when $\delta = L^{-1}$. Moreover, in the last case we can compute such value, as long as $L > M$, i.e.

$$\begin{aligned}\sup_{\pi_\delta \text{ partition of } [0,1]} \sum_{i=0}^{\#\pi_\delta - 1} L^\beta \left\| W_{\lfloor (L-1)s_{i+1} \rfloor}^{(L)} - W_{\lfloor (L-1)s_i \rfloor}^{(L)} \right\| &\leq \sum_{i=0}^{L-1} L^\beta \left\| W_{k+1}^{(L)} - W_k^{(L)} \right\| \\ &\leq \sum_{i=0}^{L-1} L^{-1/\gamma} \leq 1,\end{aligned}$$

because $\gamma \in (1/2\alpha, 1]$. □

We proved that weights regarded as paths have finite 1-variation when $\alpha > 1/2$, whereas in Proposition 3.4.1 proves that the limit, if it exists, it has finite 2-variation with $\alpha = 1/2$. This sudden change may be caused by the fact that when $\alpha > 1/2$, the variation of the trajectory of the weights cannot increase significantly via the learning procedure via gradient descent when L is large enough. Moreover, it could be that Proposition 3.4.1 is not sharp. An additional effect could be caused by the initialization described in Assumption 3.1.1 (iii). Namely, we have that $\|W_{k,k+1}\| \lesssim L^{-\rho} = L^{-1/\gamma-\beta}$ for $\rho \in (1,\infty)$ when $\gamma \in (0,1]$. Further adding to the regularity of the trajectories at "startup".

Remark 3.4.3. The result in Theorem 3.4.2 should not be deemed extraordinary. Under Assumption 3.1.1 and Assumption 3.3.2 there is a trivial example for which the 1-variation is bounded. Namely, if we were to consider the labelled dataset $(x_i, f(x_i)) \in \mathbb{R}^d \oplus \mathbb{R}^d$, with f being the identity function. Because Assumption 3.1.1 (iii) scales the parameters close to 0 as L approaches infinity, and a trivial minima is given by $W_k = 0$ for $k = 0, \dots, L-1$. It seems reasonable that gradient descent would let the weights converge to the minima under appropriate learning rates. Moreover, the path described by the parameters is of finite p-variation for $p \in (0,\infty)$.

In a similar way, because we do not want to rely on the (3.26) which assumes the existence of the limit, we limit to the lim sup with respect to the norm of the trajectories which shall exists.

Theorem 3.4.4. *Under Assumption 3.1.1 and Assumption 3.3.2 with $\nu = 1$, $\gamma \in (1/2\alpha, 1]$ and $\alpha > 1/2$. Let*

$$\mathcal{W}_t^* := \limsup_{L\to\infty} \|\mathcal{W}_t^{(L)}\|, \quad t \in [0,1]. \tag{3.29}$$

Then,

$$\limsup_{\delta\to 0^+} \sup_{\pi_\delta \text{ partition of } [0,1]} \sum_{i=0}^{\#\pi_\delta - 1} \left| \mathcal{W}_{s_{i+1}}^* - \mathcal{W}_{s_i}^* \right| < \infty, \tag{3.30}$$

where π_δ represents a generic partition of $[0,1]$ with mesh size upper bounded by $\delta > 0$.

Proof. In order to prove the result we consider any partition π_δ of specified mesh size $\delta > 0$. Noting that

$$\begin{aligned}\sum_{i=0}^{\#\pi_\delta - 1} \left| \mathcal{W}_{s_{i+1}}^* - \mathcal{W}_{s_i}^* \right| &= \sum_{i=0}^{\#\pi_\delta - 1} \left| \limsup_{L\to\infty} \|\mathcal{W}_{s_{i+1}}^{(L)}\| - \limsup_{L\to\infty} \|\mathcal{W}_{s_i}^{(L)}\| \right| \\ &\leq \sum_{i=0}^{\#\pi_\delta - 1} \limsup_{L\to\infty} \|\mathcal{W}_{s_i,s_{i+1}}^{(L)}\|\end{aligned}$$

by inverse triangle inequality. We know by Theorem 3.3.3, that there exists $M > 0$ such that for any $L > M$, it holds

$$L^{\beta}\|W_{k+1}^{L} - W_{k}^{L}\| \leq L^{-1/\gamma},$$

thus,

$$\|\mathcal{W}_{s}^{(L)} - \mathcal{W}_{t}^{(L)}\| \leq L^{-1/\gamma+1}|s-t| + L^{-1/\gamma}c, \quad c \in (0,1).$$

Therefore, each partition of fixed mesh size $\delta > 0$, is bounded in the following trivial manner

$$\sum_{i=0}^{\#\pi_\delta - 1} \limsup_{L\to\infty} \|\mathcal{W}_{s_i,s_{i+1}}^{(L)}\| \leq \sum_{i=0}^{\#\pi_\delta - 1} \limsup_{L\to\infty} L^{-1/\gamma+1}|s_{i+1} - s_i| + L^{-1/\gamma}c_i.$$

Noting that each term of the summation is bounded by $|s_{i+1} - s_i|$ if $1/\gamma \geq 1$ as L approaches ∞, we conclude

$$\limsup_{\delta\to 0^+} \sup_{\pi_\delta \text{ partition of } [0,1]} \sum_{i=0}^{\#\pi_\delta - 1} \left|\mathcal{W}_{s_{i+1}}^{*} - \mathcal{W}_{s_i}^{*}\right| \leq \limsup_{\delta\to 0^+} \sup_{\pi_\delta \text{ partition of } [0,1]} \sum_{i=0}^{\#\pi_\delta - 1} |s_{i+1} - s_i| = 1. \quad \square$$

Corollary 3.4.5. *Under the same assumptions of Theorem 3.4.4, the path $\mathcal{W}^*$ as defined in Theorem 3.4.4, has finite 1-variation.*

Proof. It is a corollary to [16, Proposition 5.6] and Theorem 3.4.4. $\square$

However, the paths described by $\mathcal{W}^{(L)}$ are not continuous. A continuous modification of $\mathcal{W}^{(L)}$ would allow for the usage of Arzela-Ascoli's type of argument. For this reason, we consider the paths described by a linear interpolation between two weights layer. We denote with $\mathfrak{W}_t^{(L)}$ the linear interpolation of the path $W^{(L)}$ rescaled by L^{β}. Namely, for $k = 0, \ldots, L-1$, we denote for $t \in [t_k^L, t_{k+1}^L]$

$$\mathfrak{W}_t^{(L)} = L^{\beta}\left((1 - L(t - t_k^L))W_k(T_L) + L(t - t_k^L)W_{k+1}(T_L)\right), \tag{3.31}$$

with $t_k^L = k/L$. Then an analogous result to Theorem 3.4.2 can be established,

Proposition 3.4.6. *Under Assumption 3.1.1 and Assumption 3.3.2 with $\nu = 1$, $\gamma \in (1/2\alpha, 1]$ and $\alpha > 1/2$. Then, there exists $K > 0$ such that*

$$\sup_{L\in\mathbb{N}_0} \|\mathfrak{W}^{(L)}\|_{1-var;[0,1]} < K.$$

Proof. We note that either the supremum is reached for some finite $L < M$ for some $M > 0$ or it is approaching the supremum for L approaching infinity. If the supremum is reached for a finite value of L, then clearly the 1-variation is finite. Therefore, we are left with the case L approaching infinity. We note that by Theorem 3.3.3 there exists $M > 0$, such that for any $L > M$,

$$L^{\beta}\|W_{k+1}^{(L)} - W_{k}^{(L)}\| \leq L^{-1/\gamma}.$$

Thus, we consider (3.32)

$$\sup_{\pi_\delta \text{ partition of } [0,1]} \sum_{i=0}^{\#\pi_\delta - 1} \left\| \mathfrak{W}_{s_i, s_{i+1}}^{(L)} \right\|, \tag{3.32}$$

for a partition π of mesh size of up to $\delta > 0$ for a fixed value of $L > M$. We note that if we prove that for any choice of $\delta > 0$, we value of (3.32) is bounded uniformly in δ, then in particular the 1-variation is bounded. We observe that $\mathfrak{W}^L$ is a continuous function defined over L intervals of length $1/L$. In each interval, any coordinate of the path is either monotonically increasing or decreasing. We know that triangle inequality becomes an equality if and only if the arguments are linearly dependent. In case of a linearly interpolated function, we observe that the sum of the increments between a "mid-point" and the extrema is the same as the increment between the extrema, since the points are on the same line, i.e. are linearly dependent. To be more precise, we denote with m_t for $t \in (0, 1)$,

$$m_t = tv + (1-t)u, \quad u, v \in \mathbb{R}^d$$

then, we can see that $v - m_t$ and $m_t - u$ are linearly dependent, because there exists $a \in \mathbb{R}$ such that

$$v - m_t = a(m_t - u) \iff a(v-u) = \frac{1}{t}(1-t)(v-u).$$

Therefore, $\|v - u\| = \|v - m_t\| + \|m_t - u\|$ by triangle equality. A similar consideration holds for $\mathfrak{W}^{(L)}$ when considering "in-between" points. Namely, for any $t \in (t_k^L, t_{k+1}^L)$ for $k = 0, \ldots, L-1$, it holds

$$\|\mathfrak{W}_{t_{k+1}^L}^{(L)} - \mathfrak{W}_{t_k^L}^{(L)}\| = \|\mathfrak{W}_{t_{k+1}^L}^{(L)} - \mathfrak{W}_{t}^{(L)}\| + \|\mathfrak{W}_{t_k^L}^{(L)} - \mathfrak{W}_{t}^{(L)}\|,$$

by triangle equality. Furthermore, if a partition of mesh size $\delta < L^{-1}$, is such that it does not include the points t_k^L, we can always add them and obtain a partition with a larger variation. Let $\pi = \{s_i\}_{i=1}^N$ be a partition of mesh size $\delta < L^{-1}$ such that $t_k^L \notin \pi$, then, by triangle inequality

$$\sum_{i=0}^{\#\pi - 1} \left\| \mathfrak{W}_{s_i, s_{i+1}}^{(L)} \right\| \leq \sum_{i=0}^{\#\pi' - 1} \left\| \mathfrak{W}_{s_i', s_{i+1}'}^{(L)} \right\|,$$

with $\pi' = \pi \cup \{t_k^L\}$. Thus, without loss of generality we consider the partition of $[0,1]$ with mesh size δ such that $\delta \geq 1/L$. Additionally, we note that if $\delta > L^{-1}$, then the partition cannot see all the increments of the function. Eventually, we conclude that the largest possible value of (3.32) is achieved when $\delta = L^{-1}$. We are able to compute the value as long as $L > M$, i.e.

$$\begin{aligned}\sup_{\pi \text{ partition of } [0,1]} \sum_{i=0}^{\#\pi_\delta - 1} \left\| \mathfrak{W}^{(L)}_{s_i, s_{i+1}} \right\| &\leq \sum_{i=0}^{L-1} L^{\beta} \left\| W^{(L)}_{k+1} - W^{(L)}_{k} \right\| \\ &\leq \sum_{i=0}^{L-1} L^{-1/\gamma} \leq 1,\end{aligned}$$

because $\gamma \in (1/2\alpha, 1]$. □

Theorem 3.4.7. *Under Assumption 3.1.1 and Assumption 3.3.2 with $\nu = 1$, $\gamma \in (1/2\alpha, 1]$ and $\alpha > 1/2$. Then, there exists a sub-sequence $\mathfrak{W}^{(L_k)}$ such that it converges to a path $\mathfrak{W} \in C^{1-var}([0,1], \mathbb{R}^{d^2})$ in p-variation for $p > 1$.*

Proof. The goal is to prove the statement by using a Arzela-Ascoli's type of result, namely [16, Proposition 5.28]. In Proposition 3.4.6 we proved that the sequence is bounded and the supremum is finite in 1-variation. Then, we are only left to prove that the sequence $\mathfrak{W}^L$ is equi-continuous to conclude the proof of the claim. We note that we can restrict to the case of $L > M$ with $M > 0$ such that Theorem 3.3.3 is valid. We fix $\varepsilon > 0$ and without loss of generality assume $\varepsilon < M^{-1}$. Then, for any choice of $L > M$, and $s, t \in [t_k^L, t_{k+1}^L]$, it holds

$$\begin{aligned}\|\mathfrak{W}^{(L)}_{s,t}\| &\leq L^{\beta+1}|s-t| \|W^{(L)}_{k,k+1}\| \\ &\leq L^{1-1/\gamma}|s-t|,\end{aligned}$$

where we used the definition (3.31). Choosing any $s, t \in [t_k^M, t_{k+1}^M]$ such that $s \in [t_k^L, t_{k+1}^L]$ and $t \in [t_{k+l}^L, t_{k+l+1}^L]$, we observe that

$$\begin{aligned}\|\mathfrak{W}^{(L)}_{s,t}\| &= \|\mathfrak{W}^{(L)}_{t^L_{k+l},t} + \mathfrak{W}^{(L)}_{t^L_{k+l-1},t^L_{k+l}} + \cdots + \mathfrak{W}^{(L)}_{s,t^L_{k+1}}\| \\ &\leq \|\mathfrak{W}^{(L)}_{t^L_{k+l},t}\| + \|\mathfrak{W}^{(L)}_{s,t^L_{k+1}}\| + \sum_{r=1}^{l-1} \|\mathfrak{W}^{(L)}_{t^L_{k+r},t^L_{k+r+1}}\| \\ &\leq L^{-1/\gamma}\left(L|t - t^L_{k+l}| + L|s - t^L_{k+1}| + (l-1)\right).\end{aligned}$$

Because $l - 1 = L|t^L_{k+l} - t^L_{k+1}|$, hence

$$\begin{aligned}\|\mathfrak{W}^{(L)}_{s,t}\| &\leq L^{1-1/\gamma}\left(|t - t^L_{k+l}| + |s - t^L_{k+1}| + |t^L_{k+l} - t^L_{k+1}|\right) \\ &\leq L^{1-1/\gamma}|t-s|.\end{aligned}$$

Under the constraint that $\gamma^{-1} \geq 1$ and $M \geq 1$, we have that $L^{1-1/\gamma} < M^{1-1/\gamma} \leq 1$. Therefore, once $\varepsilon < M^{-1}$, choosing $\delta \leq \varepsilon$ leads to $\|\mathfrak{W}_{s,t}^{(L)}\| < \varepsilon$, for any choice of $L > M$. Eventually, we proved the claim that the sequence $\left\{\mathfrak{W}^{(L)}\right\}_{L>M}^{\infty}$ is equi-continuous. □

Previously, we hinted that numerically we could not verify (3.26), although intuitively the trajectories seem bound to converge to a similar path (see section IV for additional plots). This discrepancy might stem from the convergence of a specific sub-sequence, as pointed out by the proof of Theorem 3.4.7. Another constraint of the numerical approach we used to validate (3.26), may be due to the too few layers, which could be limiting for the previous asymptotic results.

3.5 Weights regularity II

In this section we aim to obtain the same result of Theorem 3.3.3 even in case $\alpha = \beta = 1/2$. However, this is not trivial and requires a stronger assumption than Assumption 3.3.2.

Assumption 3.5.1. Under Assumption 3.1.1 (i) and (iii) and the weights $W^{(L)}$ updates via the gradient descent algorithm with a learning rate $\eta(r)$, there exist $\rho, C_\ell > 0$ satisfying

$$\frac{1}{N}\sum_{i=1}^{N} \|\nabla \ell_{\widehat{y}}(y_i, \widehat{y}_i(W(t)))\|^2 \leq \frac{C_\ell}{(t+1)^\rho}, \quad t \geq 0. \tag{3.33}$$

There is no clear path to prove Assumption 3.5.1 in a rigorous manner in the regime described by Assumption 3.1.1. For this reason, we test it empirically in section 4.3 to verify in case of CIFAR10 dataset, even in case of a more realistic model in subsection 4.3.1.

Theorem 3.5.2. *Under Assumption 3.1.1 and Assumption 3.5.1 with $\nu = 1/5$, $\gamma \in [1/2\alpha, 1]$, $\alpha + \beta = 1$ and $\alpha \geq \beta$. Let $M > 0$ such that $L > M$ is large enough. The parameters $W^{(L)}(t)$ evolve according to the gradient descent dynamics* (2.30) *with learning rate $\eta(t)$ and $T_L \in \mathbb{N}$ epochs, such that there exists a constant $Q > 0$ and for $t = 0, \ldots, T_L - 1$, we have*

$$\sum_{r=1}^{T_L} \eta(r) \frac{\sqrt{C_\ell}}{(r+1)^{\rho/2}} \leq \Lambda < \frac{1}{10d} \tag{3.34}$$

Then, for each $t = 0, \dots, T_L$ *it holds*

$$\|W_\cdot(t)\|_\infty \leq L^{-\beta},$$
$$\|W_{k,k+1}(t)\| \leq \left(\frac{1}{10} + \frac{\sqrt{2}e^3}{10}(3e+\sqrt{2})\right) L^{-\frac{1}{\gamma}-\beta}, \quad \forall k \in \{0, \dots, L-2\}.$$

Remark 3.5.3. If the condition (3.33) is satisfied with $\eta(r) = \frac{Q}{(r+1)^{1-\rho/2+\varepsilon}}$ for some constant $Q, \varepsilon > 0$. Then, by known results on the convergence of the harmonic series, it holds

$$\sum_{r=1}^{T_L} \eta(r)\sqrt{\frac{C_\ell}{(r+1)^{\rho/2}}} \leq \sum_{r=1}^{T_L} \frac{Q\sqrt{C_\ell}}{(r+1)^{1+\varepsilon}} < \frac{1}{10d}.$$

Then, we can choose $Q = Q_\varepsilon \leq (10\zeta(1+\varepsilon)d\sqrt{C_\ell})^{-1}$ for any choice of $\varepsilon > 0$ where we used the notation $\zeta(s)$ to denote the classical Riemann zeta function. From numerical experiments in section 4.3, $\rho \ll 1$ hence we expect the learning rates $\eta(t)$ to be quite small. Moreover, we note that T_L may be as large as desired and not dependent on L.

Proof of Theorem 3.5.2. The proof follows *mutatis mutandis* the result in Theorem 3.3.3. For a more succinct exposition, the instances where the rationale mirrors the preceding proof of Theorem 3.3.3 are removed. Our focus shifts to the little adjustments necessary to correct the previous proof under the new assumptions. For $t = 0$, the conditions are trivially satisfied. Thus, we can start the inductive proof by iterating through the 2 statements.

Bound on layers norm. By the recursive relation in Lemma 3.2.3,

$$\begin{aligned}
\|W_k(t+1)\| &\leq \|W_k(0)\| + \sum_{s=0}^{t} \eta(s) L^{-\alpha} G^s_{k,\infty} \max_{j=1,\dots,N} \|h_k^{x_j}\| \\
&\leq \frac{1}{10} L^{-\beta} + L^{-\alpha} \sum_{r=0}^{t} \eta(r) \frac{\sqrt{C_\ell}}{(r+1)^{\rho/2}} \sqrt{d} \exp(2) \\
&\leq \frac{1}{10} L^{-\beta} + L^{-\alpha} \Lambda d \exp(2)
\end{aligned}$$

where the last estimate is given by the fact that $\alpha \geq \beta$. Moreover, $G^s_{k,\infty} < \sqrt{C_\ell}/(s+1)^{\rho/2} d \exp(1) < \infty$ uniformly over s, k by Lemma 3.2.1 and the inductive hypothesis on $\|W_k(s)\|$. Finally, we need to prove the following inequality, to prove the claim

$$\frac{1}{10} L^{-\beta} + L^{-\alpha} \Lambda d\sqrt{d} \exp(2) \leq L^{-\beta}.$$

Because the worst case scenario is for $\alpha = \beta = 1/2$, as the inequality becomes independent of L; we note that as $d\sqrt{d}\Lambda < 1/10$ by assumption, hence

$$\frac{1}{10} + \frac{\exp(2)}{10} < 1,$$

finally proving the inequality,

$$\|W_k(t+1)\| \leq L^{-\beta}.$$

Bound on layers' increments. Using the discrete Grönwall inequality (Lemma 3.3.4), with respect to the recurrent relation

$$\|W_{k,k+1}(t+1)\| \leq \|W_{k,k+1}(t)\| \left(1 + \eta(t)\delta_L 2\sqrt{2} G^t_{k+1,\infty} \max_{j=1,\dots,N} \|h_k^{x_j}\|^2\right)$$
$$+\eta(t)\delta_L^2 \sqrt{2} G^t_{k+1,\infty} \max_{j=1,\dots,N} \|h_k^{x_j}\| \left(2\|W_{k+1}\|\|W_k\|\|h_k^{x_j}\| + \sqrt{2}\|W_k\| + \|W_{k+1}\|^2 \|h_{k+1}^{x_j}\|\right).$$

we obtain,

$$\|W_{k,k+1}(t+1)\| \leq \prod_{s=0}^{t} (1 + \eta(s) L^{-\alpha} 2\sqrt{2} G^s_{k+1,\infty} \max_{i=j,\dots,N} \|h_k^{x_j}(s)\|^2) \|W_{k,k+1}(0)\| +$$
$$+ \sum_{s=0}^{t} \left(\prod_{r=s+1}^{t} (1 + \eta(r) L^{-\alpha} 2\sqrt{2} G^r_{k+1,\infty} \max_{i=j,\dots,N} \|h_k^{x_j}(r)\|^2) \right) \cdot$$
$$\cdot \eta(s)\delta_L^2 \sqrt{2} G^s_{k+1,\infty} \max_{j=1,\dots,N} \|h_k^{x_j}(s)\| \Big(2\|W_{k+1}(s)\|\|W_k(s)\|\|h_k^{x_j}(s)\|$$
$$+ \sqrt{2}\|W_k(s)\| + \|W_{k+1}(s)\|^2 \|h_{k+1}^{x_j}(s)\| \Big).$$

Using the inequality $1 + x \leq \exp(x)$ and substituting the fact that $\|W_k(s)\| \leq L^{-\beta}$ for each $k \in \{0, \dots, L-1\}$ and, $s \in \{0, \dots, t\}$ by inductive hypothesis, we can easily check that $\|M^{x_j}(s)_k\|, \|h_k^{x_j}(s)\| \leq \exp(1)$ in the same range of s and k as before. Moreover, in the similar manner to the previous estimates $G^s_{k+1,\infty} < \sqrt{C_\ell}/(s+1)^{\rho/2} d \exp(1)$. Hence, it yields

$$\|W_{k,k+1}(t+1)\| \leq \exp\left(2d\sqrt{2}e^3 L^{-\alpha}\Lambda\right) \Big(\|W_{k,k+1}(0)\|$$
$$+ d\sqrt{2}\Lambda e^2 L^{-2\alpha-\beta}(3eL^{-\beta} + \sqrt{2}) \Big).$$

By initial condition of $\|W_{k,k+1}(0)\| \leq \frac{1}{10e} L^{-1/\gamma-\beta}$, we have

$$\|W_{k,k+1}(t+1)\| \leq \exp\left(2d\sqrt{2}e^3 L^{-\alpha}\Lambda\right) \Big(\frac{1}{10e} L^{-1/\gamma-\beta}$$
$$+ d\sqrt{2}\Lambda e^2 L^{-2\alpha-\beta}(3eL^{-\beta} + \sqrt{2}) \Big).$$

Finally, we observe that there exists L large enough,

$$\exp\left(2d\sqrt{2}e^3L^{-\alpha}\Lambda\right) \leq e \iff L^{-\alpha} \leq \frac{10}{2\sqrt{2}e^3},$$

because $\Lambda d \leq 1/10$. Moreover, we note that if the inequality is satisfied for $\alpha = \beta$ and $1/\gamma = 2\alpha$, it holds for $\alpha > \beta$ and $1/\gamma < 2\alpha$. We consider only the former case, in which the inequality becomes, under the hypothesis that $\Lambda d \leq 1/10$,

$$\frac{1}{10}L^{-1/\gamma-\beta} + \frac{\sqrt{2}e^3}{10}L^{-2\alpha-\beta}(3eL^{-\beta} + \sqrt{2}) \leq \left(\frac{1}{10} + \frac{\sqrt{2}e^3}{10}(3e + \sqrt{2})\right) L^{-1/\gamma-\beta}$$

$$\Updownarrow$$

$$\frac{1}{10} + \frac{\sqrt{2}e^3}{10}(3eL^{-\beta} + \sqrt{2}) \leq \frac{1}{10} + \frac{\sqrt{2}e^3}{10}(3e + \sqrt{2}),$$

Hence, for $L > 1$ we note that $3eL^{-\beta} \leq 3e$, proving at time $t + 1$

$$\|W_{k,k+1}(t+1)\| \leq \left(\frac{1}{10} + \frac{\sqrt{2}e^3}{10}(3e + \sqrt{2})\right) L^{-1/\gamma-\beta}$$

Finally concluding the proof of all the claims by induction. □

Remark 3.5.4. Because the statements of Theorem 3.3.3 and Theorem 3.5.2 are the same up to a constant value, the same regularity results of the weights hold, namely results Corollary 3.4.5, and Theorem 3.4.7 even with the choice of the parameters $\gamma = 1/2\alpha$ and $\alpha = \beta$ at initialization. Concluding that even in case of a very slow convergence rate, i.e. $\rho \ll 1$ of the average of the gradient loss function, we should expect the weights path to be of bounded 1-variation in the depth-limit once we choose a learning rate appropriately and in the specified regimes of Assumption 3.1.1 and Assumption 3.5.1.

Chapter 4

Numerical Experiments

In this chapter we defer all simulations and visualizations of chapter 3 results. In particular, we show that the result Corollary 3.4.5 is satisfied for large L even in an experimental setting and using a wide gamut of different hyper-parameters. Analogously, we check that the 1-variation of the weights is bounded as expressed in Theorem 3.4.2 and Theorem 3.4.7. Moreover, we verify Assumption 3.5.1 consistency with real world examples and not only on theoretical models, in order to guarantee realistic assumptions for Theorem 3.5.2. However, prior to embarking in the experimental results, we begin with a concise exposition of the experimental apparatus. The model's forward pass is described in equation (2.4), but minor details must be changed to account for the data dimensionality. For this reason, we introduce U_{in}, U_{out} as embedding of the data to and from the inner ResNet model. Hence, the forward pass is described as

$$\begin{cases} h_0^{x_i} = U_{\text{in}} x_i \\ h_{j+1}^{x_i} = h_j^{x_i} + \delta_L \sigma(W_j h_j^{x_i}) \qquad j = 1, \dots, L-1 \\ h_L^{x_i} = U_{\text{out}}(h_{L-1}^{x_i} + \delta_L \sigma(W_{L-1} h_{L-1}^{x_i})) \end{cases} \tag{4.1}$$

For example, in case of CIFAR10[1], each data point represents a colored image as a vector $x_i \in \mathbb{R}^{32\times32\times3}$. For computational reasons, we limit weights to very low dimensions, i.e. $W_i \in \mathbb{R}^{d\times d}$ with $d = 15$. Thus, the embedding should be of dimension $U_{\text{in}} \in \mathbb{R}^{32\times32\times3\times d}$. Concluding the example, the embedding $U_{\text{out}} \in \mathbb{R}^{d\times c}$ with $c = 10$ being the value of distinct classes to classify the initial objects. The training procedure is via full gradient descent with a constant learning rate η, on the trainable parameters W_k for $k \in \{0, \dots, L-1\}$, while the embedding are set to constant non-trainable values. The initialization is either set to a constant initial value or sampled from a truncated normal distribution. In case of a constant

[1]The datasets used are described in section 4.1.

initial value, it is determined by the parameters L, γ and α. Namely, to satisfy condition Assumption 3.1.1 (iii), we use for each layer $k = 0, \dots, L-1$,

$$W_{k,ij} = \frac{1}{d^2} L^{-\frac{1}{\gamma}-\beta}, \quad \forall i, j \in \{1, \dots, d\}\,. \tag{4.2}$$

Thus, in this case, no stochastic behaviour is involved in the learning procedure, except for the embeddings' weights which are sampled from a uniform distribution on $[-\sqrt{\varphi}, \sqrt{\varphi}]$, with φ^{-1} being the number of input features. Therefore, we select a seed for each simulation in order to preserve the same embedding throughout the different models selection of the parameter L to highlight eventual trends in the depth-limit. Similarly, for the truncated normal case, we initialize it with a truncated normal distribution whose mean and variance are respectively 0 and $\frac{1}{d^2} L^{-\frac{1}{\gamma}-\beta}$, and truncated in the range $[-\frac{1}{d^2} L^{-\frac{1}{\gamma}-\beta}, \frac{1}{d^2} L^{-\frac{1}{\gamma}-\beta}]$ to satisfy Assumption 3.1.1 (iii). For ease of notation we collect each model selection of hyper-parameters and dataset in Table 4.1 and Table 4.2. The nomenclature for each model selection of hyper-parameters is encoded via a letter B, in case the model hyper-parameters are coherent with the assumptions of Theorem 3.3.3 or Theorem 3.5.2, whilst the model denoted with a letter C do not. The letter t or c at the end of the model selection name identifies the type of initialization for the training parameters, respectively truncated and constant initialization. For example, the hyper-parameters described by CIFAR10-B11c denote a model whose depth-limit should be bounded, trained on the CIFAR10 dataset with a constant initialization.

ModelClass	d	α	γ	η	T_L	σ
CIFAR10-B11	15	0.975	1.0	$100L^{\beta}$	$5\log L$	tanh
CIFAR10-B12	15	0.975	1.0	$L^{\alpha-\beta}$	$5\log L$	tanh
CIFAR10-B21	15	0.75	1.0	$20L^{\beta}$	$5\log L$	tanh
CIFAR10-B22	15	0.75	1.0	$L^{\alpha-\beta}$	$5\log L$	tanh
CIFAR10-B31	15	0.5	1.0	L^{β}	$5\log L$	tanh
CIFAR10-B32	15	0.5	1.0	$0.01L^{\beta}$	$5\log L$	tanh
CIFAR10-B33	15	0.5	1.0	$L^{\alpha-\beta}$	$5\log L$	tanh
CIFAR10-C11	15	0.25	1.0	L^{β}	$5\log L$	tanh
CIFAR10-C12	15	0.25	1.0	$L^{\alpha-\beta}$	$5\log L$	tanh
CIFAR10-C21	15	0.49	2.0	L^{β}	$5\log L$	tanh
CIFAR10-C22	15	0.49	2.0	$L^{\alpha-\beta}$	$5\log L$	tanh

Table 4.1: ResNet Models classes and corresponding hyper-parameters for CIFAR10 data module.

ModelClass	d	α	γ	η	T_L	σ
C10k-B11	1	0.975	1.0	L^β	$5 \log L$	tanh
C10k-B12	1	0.975	1.0	$100L^\beta$	$5 \log L$	tanh
C10k-B13	1	0.975	1.0	$L^{\alpha-\beta}$	$5 \log L$	tanh
C10k-B21	1	0.75	1.0	L^β	$5 \log L$	tanh
C10k-B22	1	0.75	1.0	$20L^\beta$	$5 \log L$	tanh
C10k-B23	1	0.75	1.0	$20L^{\alpha-\beta}$	$5 \log L$	tanh
C10k-B31	1	0.5	1.0	L^β	$5 \log L$	tanh
C10k-B32	1	0.5	1.0	$0.01L^\beta$	$5 \log L$	tanh
C10k-B33	1	0.5	1.0	$L^{\alpha-\beta}$	$5 \log L$	tanh
C10k-C11	1	0.25	1.0	L^β	$5 \log L$	tanh
C10k-C12	1	0.25	1.0	$0.01L^\beta$	$5 \log L$	tanh
C10k-C13	1	0.25	1.0	$L^{\alpha-\beta}$	$5 \log L$	tanh
C10k-C21	1	0.49	2	L^β	$5 \log L$	tanh
C10k-C22	1	0.49	2	$0.01L^\beta$	$5 \log L$	tanh
C10k-C23	1	0.49	2	$L^{\alpha-\beta}$	$5 \log L$	tanh

Table 4.2: ResNet Models classes and corresponding hyper-parameters for C10k data module.

We choose the learning rates according to Remark 3.3.5. Nonetheless, we tried different scaling of the learning rates. Moreover, as we will see, in case $\beta \approx 0$, when $\eta = L^\beta$, the learning rate is practically constant and close to 1, throughout the considered value of $L \in \{2000, \dots, 12000\}$. Hence, the training procedure does not impinge on the regularity of $\mathcal{W}^{(L)}$ because no meaningful optimization is modifying the weights. The learning rate $L^{\alpha-\beta}$ seems to achieve the best results in training, with an accuracy of about 30% in very few steps in case of CIFAR10 dataset. The accuracy is quite large noting that the embedding are set to random non-trainable values; the input has more than 10 times the number of dimensions than the ResNet hidden states; no data augmentation is used and the architecture is a theoretical abstraction of more convoluted models. In the following, we do not include any discussion of the loss or accuracy of the model as it is tangential to the results regarding the regularity of the weights.

4.1 Datasets

In the following we train the ResNet models on two quite diverse dataset. We denote the first as C10k, which is a synthetic dataset consisting of uniformly random points $N = 10000$ in the range $[-1, 1]$ and the target function is the

constant function $f(x) = 0.01$. In this case, it was proved in [25] that Residual Networks with a relu activation converges. Hence, this is expected to be the best case scenario for our model. Despite the activation function not set to relu, we should expect quite consistent results for most of the chosen hyper-parameters. Our second dataset of choice is CIFAR10[2], which is expected to better capture

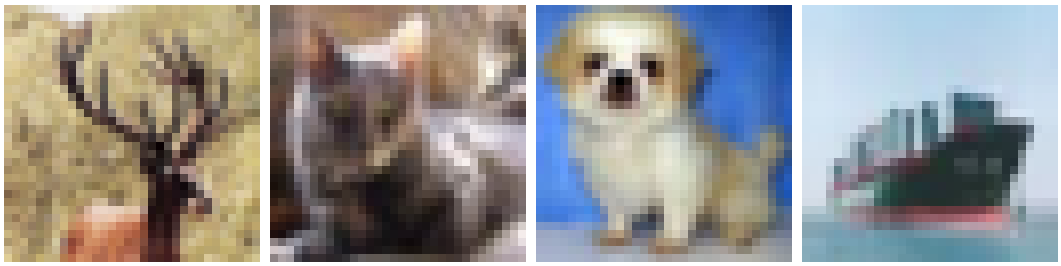


Figure 4.1: Example of the training dataset CIFAR10, with 4 out of 10 the classes, respectively from left to right: a deer, a cat, a dog and a ship.

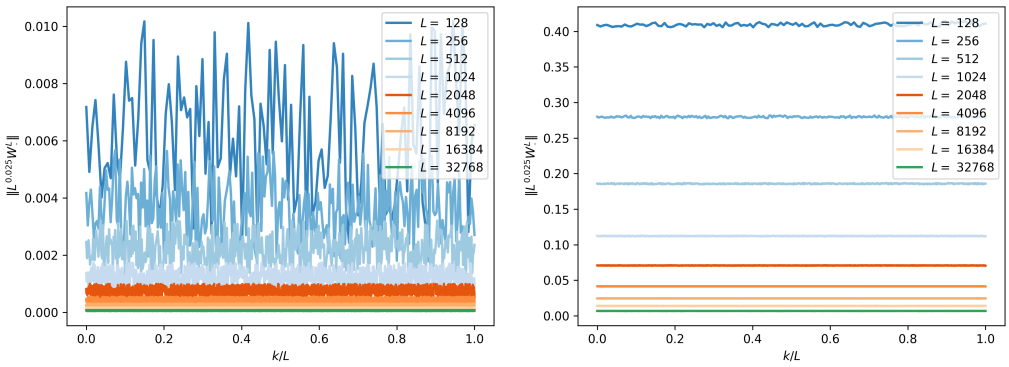
real-life scenario complexities. The complete training dataset consists of over 50000 32×32 coloured images that represents 10 different objects. In order to fasten the training procedure and reduce memory overhead caused by full gradient descent steps, we restrict to 25% of the training dataset, unless otherwise specified.

4.2 Weights regularity

In this section, we aim to give numerical examples consistent with the results in Corollary 3.4.5, Theorem 3.4.2, and Proposition 3.4.6 and we further investigate whether the assumptions are sharp. The 1-variation is computed after training via full gradient descent on the specified dataset and using the hyper-parameters specified by the model's name. The algorithm used to compute the 1-variation is described in [7], albeit in the case $p = 1$ there is no difference between the algorithm and the naive implementation of summing all the increments of a path[3]. For $p > 1$, the proposed method exhibits a significant speedup compared to the brute-force approach of checking all possible sub-increments. This efficiency advantage was relevant during the initial stages of experimentation, in which we successfully verified the boundedness of the 2-variation property, which was proved in [9]. In the following experiments, for each hyper-parameters choice, we trained 3 models with a different random seed to account for the stochastic behaviour of the embedding in the models and the sampling of the used data.

[2]The dataset is available at https://www.cs.toronto.edu/~kriz/cifar.html.
[3]an implementation is available at https://github.com/ntapiam/rust-pvar

4.2.1 C10k



(a) C10k-B11t (b) C10k-B12t

Figure 4.2: Trajectories of $\|\mathcal{W}^{(L)}\|$ of the specified models hyper-parameters. For $L > 4096$ we see a clear dampening in the trajectories oscillation behaviour.

Before showing the results, we motivate the usage of $\alpha = 0.975$ as a mere test case to verify the correctness of Theorem 3.3.3. Indeed, from the proof of Theorem 3.3.3, we deduce

$$L^{\beta-\alpha}\log(L) \leq \frac{1}{4dQ\sqrt{C_\ell}\exp(2)}. \tag{4.3}$$

In case $\alpha = 0.975$, $Q = 5$, $d = 1$ and noting that

$$\frac{1}{N}\sum_{i=1}^{N}\|\nabla_{\widehat{y}}\ell_2(y_i,\widehat{y}_i)\|_2^2 \leq \frac{2}{N}\sum_{i=1}^{N}(\|y_i\|_2^2+,\|\widehat{y}_i)\|_2^2) \leq 2,$$

where we denoted with $\ell_2(y_i,\widehat{y}_i) = 2^{-1}\|y_i-\widehat{y}_i\|_2^2$ the MSE loss. We should see that $\operatorname{argmax}_L \|\mathcal{W}^{(L)}\|_{1-\text{var}} < 4000$. That is, in case of C10k-B11, C10k-B12, class models we should expect when the number of layers is larger than 4000, the 1-variation decreases. We should not expect the same for C10k-B13, since the learning rate is chosen $L^{\alpha-\beta}$ and from Remark 3.3.5, we know that $\eta \lesssim L^{\alpha-\beta-\varepsilon}$ for any $\varepsilon > 0$. Indeed, the models C10k-B11c, C10k-B12c, C10k-B11t, C10k-B12t achieve their maximum before $L = 4096$, which is consistent with the reported data using both a constant and a truncated normal initialization. The data is reported in Table 4.3.

L	C10k-B11c	C10k-B11t	C10k-B12c	C10k-B12t	C10k-B13c	C10k-B13t
128	$7.883 \cdot 10^{-6}$	0.3359	0.2152	0.3361	0.1925	0.3361
256	$1.118 \cdot 10^{-6}$	0.3247	0.1215	0.3248	0.2077	0.3247
512	$2.161 \cdot 10^{-7}$	0.3314	0.06351	0.3314	0.2284	0.3312
1024	$2.105 \cdot 10^{-7}$	0.3217	0.02592	0.3217	0.2343	0.3214
2048	$1.453 \cdot 10^{-7}$	0.3206	0.01007	0.3155	0.2565	0.3242
4096	$2.664 \cdot 10^{-7}$	0.3203	0.002808	0.3123	0.2656	0.3318
8192	$3.19 \cdot 10^{-7}$	0.3171	0.0006477	0.3116	0.2787	0.3371
16384	$3.336 \cdot 10^{-7}$	0.3155	0.0001158	0.3124	0.2844	0.3445
32768	$2.817 \cdot 10^{-8}$	0.3136	$3.639 \cdot 10^{-5}$	0.3133	0.2908	0.3485

Table 4.3: Computation of $\|\mathcal{W}^{(L)}\|_{1-\text{var};[0,1]}$ in case of the different specified model against C10k dataset. The maximum value out of 3 sample models is reported.

A depiction of the trajectories of $\|\mathcal{W}^{(L)}\|$ are shown in Figure 4.2.
As we can see, for $L < 4000$, the paths are very unsteady, but their jumps magnitude steadily decrease as L increases. For $L \geq 2^{11}$, the paths seem straight lines. This is rather confusing, since there is no significant decrease in 1-variation in the respective table. However, this is due to the resolution of the y-axis in the plots. In light of the trajectory depicted on the right, we see that the jumps magnitude diminishes but the jumps are more frequent, thus almost completely balancing the 1-variation. A similar behaviour occurs for larger L.

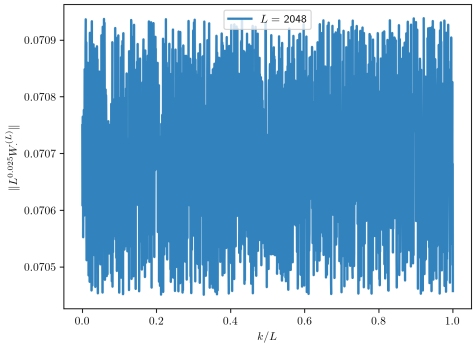
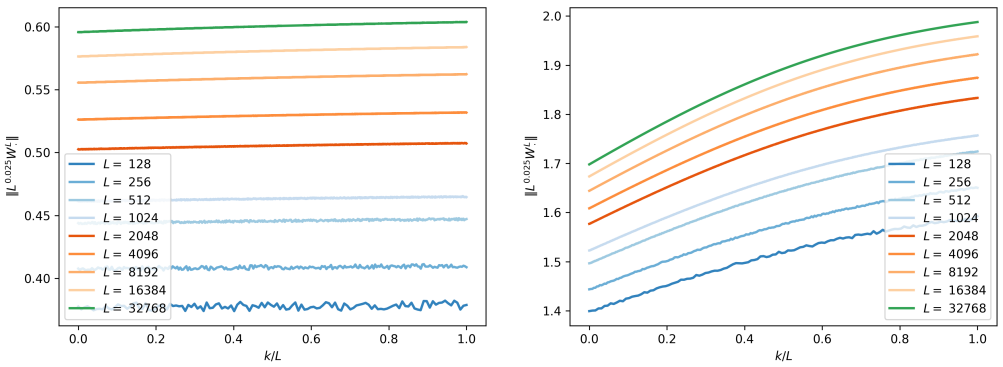


Figure 4.3: $\|\mathcal{W}^{(2048)}\|$ for the hyper-parameters C10k-B12t

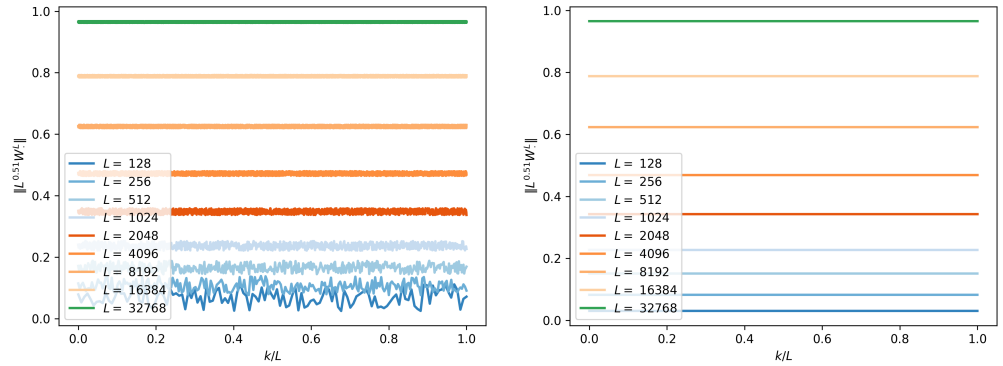
The result of Theorem 3.3.3 seems tight in the setting of a learning rate L^α with $\alpha \approx 1$, since from Table 4.3 we can see that even for $L > 4096$ the models C10k-B13c, C10k-B13t there is a slow but sure increasing trend. However, the trajectories increase in roughness, described by $\mathcal{W}^{(L)}$ is not quite evident from Figure 4.4. The theoretical bound we discussed in (4.3) for this dataset and $\alpha = 0.975$, seems to hold for all the other choice of hyper-parameters as long as the learning rate is set not too large. Indeed, we see that in case of C10k-C11c and C10k-C11t, which are respectively reported in Table 4.4, the values of $\|\mathcal{W}^{(L)}\|_{1-\text{var};[0,1]}$ are monotonically increasing. But, as we change the learning rate, e.g. in C10k-C12c, C10k-C13c, the 1-variation decreases as L increase. Concluding that the learning procedure affect significantly the weights regularity. Another case in which the 1-variation is



(a) C10k-B13t (b) C10k-B13t

Figure 4.4: Two realisation of the trajectories $\|\mathcal{W}^{(L)}\|$ of the specified model hyperparameters.

not decreasing or almost stationary is for $\gamma = 2$, namely in the models C10k-C21t, C10k-C22t, C10k-C23t. But, a modification to the initialization induces a change in the 1-variation behaviour, as we can see from the data regarding the hyperparameters C10k-C21c, C10k-C22c, C10k-C23c. This is further evident by the trajectories plot, e.g. in Figure 4.5. However, this should not be deemed unexpected:



(a) C10k-C22t (b) C10k-C22c

Figure 4.5: Representation of the trajectories of $\|\mathcal{W}^{(L)}\|$ in case of two different initialization.

the constant value initialization is such that $\|W_{k,k+1}\| = 0$, whereas the truncated normal is expected to be more rough, with increments $\|W_{k,k+1}\| \approx L^{-1/\gamma-\beta}$. The

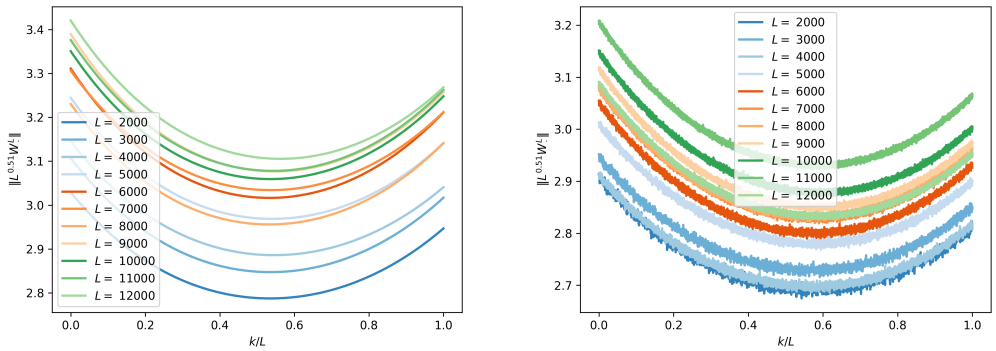
different behaviours justify that under stricter conditions, we can assume $\gamma > 1$ and still obtain finite 1-variation of $\mathcal{W}^{(L)}$ as L approaches infinity. For the other

L	C10k-C11c	C10k-C11t	C10k-C13c	C10k-C13t	C10k-C23c	C10k-C23t
128	$1.621 \cdot 10^{-5}$	0.6667	0.0002309	0.3364	0.002322	3.806
256	0.0006409	0.766	$8.583 \cdot 10^{-5}$	0.325	0.001285	5.201
512	0.00531	0.8511	$3.362 \cdot 10^{-5}$	0.3316	0.0007176	7.503
1024	122	123	$2.444 \cdot 10^{-5}$	0.3218	0.0003761	10.3
2048	889.1	889.2	$3.946 \cdot 10^{-5}$	0.3209	0.0002147	14.52
4096	1839	1849	$7.331 \cdot 10^{-5}$	0.3204	0.0001513	20.5
8192	4453	4442	$6.33 \cdot 10^{-5}$	0.3172	0.0001808	28.71
16384	8485	8471	0.0003606	0.3155	0.0003512	40.39
32768	$1.454 \cdot 10^{4}$	$1.457 \cdot 10^{4}$	0.0004196	0.3136	0.0006418	56.76

Table 4.4: Computation of $\|\mathcal{W}^{(L)}\|_{1-\text{var};[0,1]}$ in case of the different specified model against C10k dataset. The maximum value out of 3 sample models is reported.

choices of the hyper-parameters there is no interesting behaviour to report. The data appears to be rather consistent with the theoretical results; for completeness we report the observed data in section III.

4.2.2 CIFAR10



(a) CIFAR10-C22c (b) CIFAR10-C22t

Figure 4.6: Depiction of trajectories of $\|\mathcal{W}^{(L)}\|$ with different initialization confronted.

An analogous analysis can be performed on the CIFAR10 dataset, employing the hyper-parameters outlined in Table 4.1. With this dataset, the verification of

the bound in (4.3) in L becomes impractical even for large values of alpha, i.e. 0.975. Hyper-parameter settings like CIFAR10-B11 and CIFAR10-B12 require L to exceed $2.5 \cdot 10^5$, highlighting the limitations of the theoretical results in a more practical setting. While the previous dataset offered insights on the validity of the

L	CIFAR10-B12c	CIFAR10-B21c	CIFAR10-B22c	CIFAR10-B32c	CIFAR10-B33c
2000	1.583	5.519	2.971	1.444	3.16
3000	1.7	5.381	3.136	1.91	3.296
4000	1.642	5.53	3.117	2.149	3.271
5000	1.755	5.316	3.318	2.593	3.469
6000	1.802	5.33	3.536	3.034	3.696
7000	1.801	5.396	3.42	3.104	3.541
8000	1.795	5.02	3.392	3.248	3.518
9000	1.878	5.21	3.636	3.635	3.76
10000	1.875	5.019	3.563	3.67	3.67
11000	1.867	5.028	3.564	3.799	3.685
12000	1.85	5.056	3.559	3.873	3.673
L	CIFAR10-B12t	CIFAR10-B21t	CIFAR10-B22t	CIFAR10-B32t	CIFAR10-B33t
2000	5.524	8.067	6.378	5.886	6.497
3000	5.542	7.928	6.574	6.065	6.709
4000	5.549	7.731	6.519	6.142	6.624
5000	5.566	7.693	6.583	6.26	6.677
6000	5.544	7.762	6.611	6.395	6.711
7000	5.574	7.681	6.651	6.504	6.734
8000	5.561	7.72	6.627	6.564	6.708
9000	5.569	7.642	6.678	6.684	6.753
10000	5.591	7.707	6.766	6.859	6.859
11000	5.568	7.714	6.687	6.821	6.759
12000	5.597	7.556	6.781	6.975	6.859

Table 4.5: Computation of $\|\mathcal{W}^{(L)}\|_{1-\text{var};[0,1]}$ in case of the different specified model on CIFAR10 dataset. The maximum value out of 3 sample models is reported.

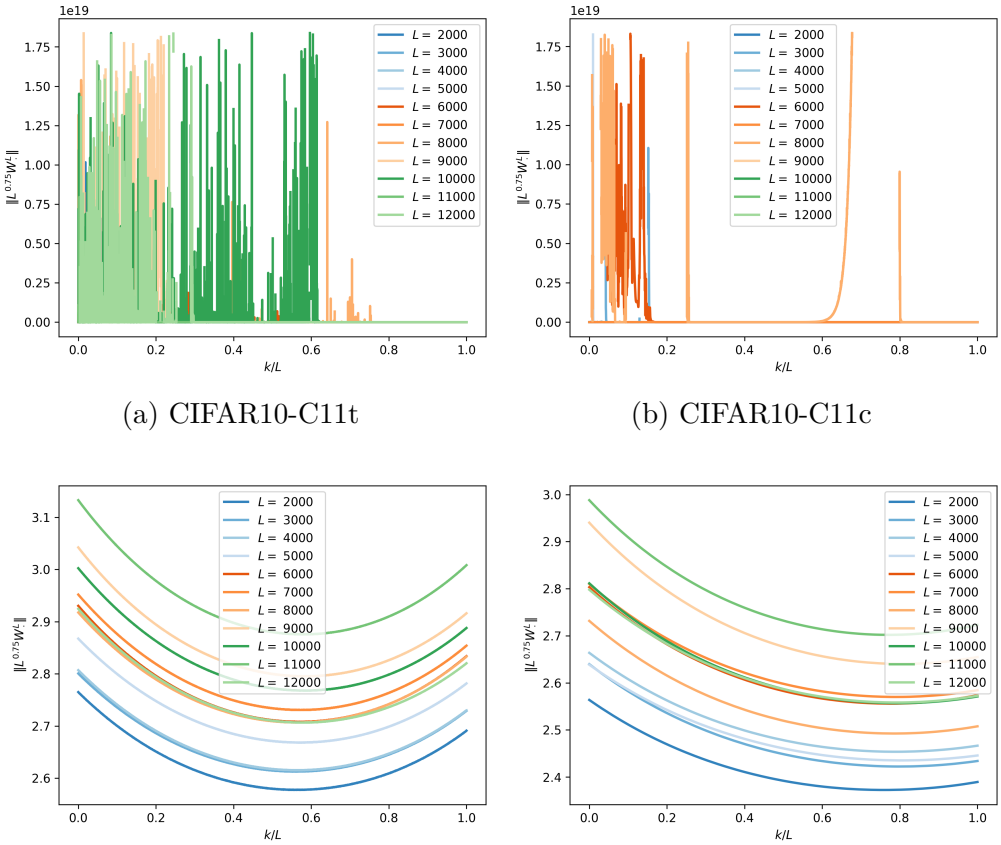
results in chapter 3, CIFAR10, describes a real-world scenario with greater variety, strengthening our understanding of the results generality. Notably, even for smaller value of L than the lower bound obtainable by plugging-in the value of the hyper-parameters in (4.3), the regularity of the weights regarded as paths, remains small whenever the hyper-parameters are chosen to satisfy Theorem 3.4.2 hypothesis. However, further investigation into minor differences is needed. From Table 4.5, we note that the maximum value of $\|\mathcal{W}\|_{1-\text{var};[0,1]}$ is reached well below the expected bound on L used in the proof of Theorem 3.3.3. In most cases, we achieve

a maximum for $L \leq 10000$. The attainment of a maximum already for such values of L is due to the constant initialization, as we have already observed in the C10k dataset. This initialization vastly facilitates the regularity of the trajectories, since the initial 1-variation is 0. Concluding, that the initialization is a determining factor in the final regularity of the trajectories. When initializing using the truncated normal distribution, a definitive upper bound for the path variation seems elusive. In this case, the 1-variation exhibits non-monotonic behaviour in L, either increasing or by exhibiting an oscillating increase, see Table 4.5.

L	CIFAR10-C21c	CIFAR10-C22c	CIFAR10-C21t	CIFAR10-C22t
2000	1087	3.39	689.4	255.9
3000	1390	3.518	1758	313.1
4000	2031	3.493	1485	361.6
5000	8010	3.687	3906	404.4
6000	4269	3.911	$3.167 \cdot 10^4$	443
7000	6406	3.747	4856	478.6
8000	4095	3.731	$1.369 \cdot 10^4$	511.5
9000	$1.517 \cdot 10^6$	3.968	9430	542.5
10000	9159	3.888	9713	571.7
11000	8571	3.911	$8.589 \cdot 10^6$	599.7
12000	$6.569 \cdot 10^5$	3.883	$1.174 \cdot 10^4$	626.4

Table 4.6: Computation of $\|\mathcal{W}^{(L)}\|_{1-\text{var};[0,1]}$ in case of the different specified model on CIFAR10 dataset. The maximum value out of 3 sample models is reported.

The reported values emphasize the sharpness of the result, namely that L must be far larger than practical occurrences. To further solidify our observations, let us consider the edge cases CIFAR10-C22c and CIFAR10-C22t, which are outside the range of Theorem 3.4.2's hypothesis (namely $\gamma > 1$ and $\alpha < \beta$). For these model hyper-parameters, the paths exhibit quite distinct behaviours, see Figure 4.6. As expected from similar remarks in the case of the C10k dataset, under constant initialization, the paths appear quite regular, whereas truncated normal initialization yields highly irregular trajectories. Solely due to an excessively large learning rate, the model's weights show uncontrolled growth in 1-variation, indicated by the divergence of the 1-variation (see Table B.3). This phenomenon persists even when the hyper-parameters deviate from the assumptions of Theorem 3.3.3, highlighting the crucial role of the learning rate. The erratic behaviour observed in the paths associated with hyper-parameters CIFAR10-C11 (Figure 4.7a) and CIFAR10-C12 (Figure 4.7b) serves as a quite effective evidence. Therefore, the optimization procedure is fundamental in the regularity of the paths $\mathcal{W}^{(L)}$ as the initialization is, as we have previously observed. Interestingly, variations in the parameter α do



(a) CIFAR10-C11t

(b) CIFAR10-C11c

(c) CIFAR10-C12t

(d) CIFAR10-C12c

Figure 4.7: The different behaviour of the trajectories $\|\mathcal{W}^{(L)}\|$ as the learning rate changes for $\alpha = 0.25$.

not appear to significantly impact the regularity of the paths, provided we adjust the learning rate accordingly. This observation is reinforced by the model with hyper-parameters CIFAR10-C12, where the 1-variation remains unsteadily increasing and attain the maximum value with both initialization, at $L = 9000$ and $L = 10000$. In turn, the trajectories are more regular as depicted in Figure 4.7c and Figure 4.7d. This last observation may suggest potential refinements to Theorem 3.3.3.

4.3 Convergence rate hypothesis

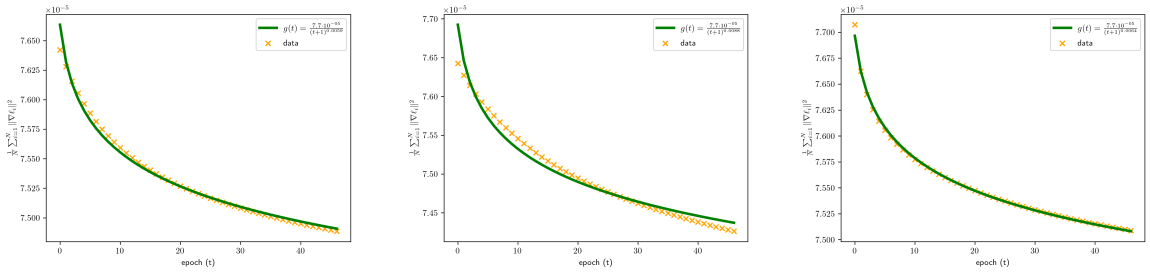
In section 3.5, we assumed condition (3.33) on the average of the norm of the gradients. We show that in copious cases it seems rather consistent with what has been observed on CIFAR10 dataset. In truth, we check that the upper bound may be limited to a stricter condition. Considering the following function

$$\Omega^{N,\ell}(t) = \frac{1}{N}\sum_{i=1}^{N} \|\nabla_{\widehat{y}}\ell(y_i, \widehat{y}_i(W(t)))\|^2 \tag{4.4}$$

where N, ℓ depends of the dataset used and $\widehat{y}_i(W(t))$ is the output of the Residual Neural Network as described in (4.1) at the training step t. Numerical evidences suggest

$$\Omega^{N,\ell}(t) \approx \frac{C}{(t+1)^{\rho}} \tag{4.5}$$

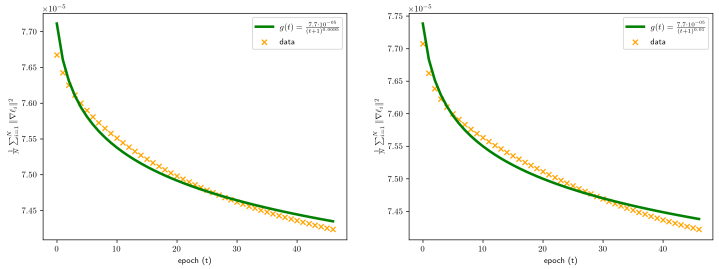
for some choice of the parameters C, ρ. In the following, we use `scipy.curve_fit`[41]



(a) CIFAR10-B12t, $L = 12000$.

(b) CIFAR10-B32t, $L = 11000$.

(c) CIFAR10-B12t, $L = 10000$.

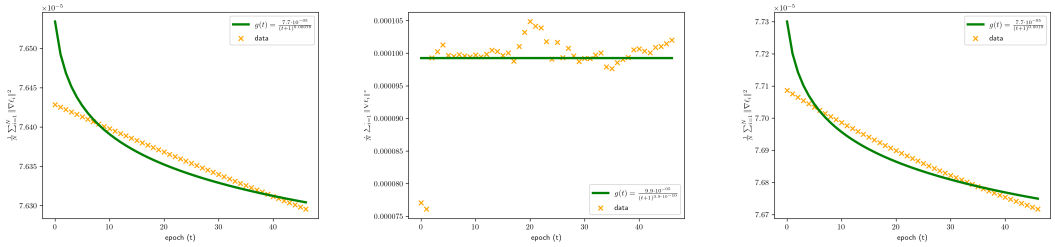


(d) CIFAR10-B22c, $L = 12000$.

(e) CIFAR10-C12t, $L = 10000$.

Figure 4.8: Constrained non-linear least-squares regression to fit the function $g(t) = {}^{C}/_{(t+1)^{\rho}}$ to the data of $\Omega^{12500,\text{CrossEntropy-loss}}$ in case of CIFAR10 dataset.

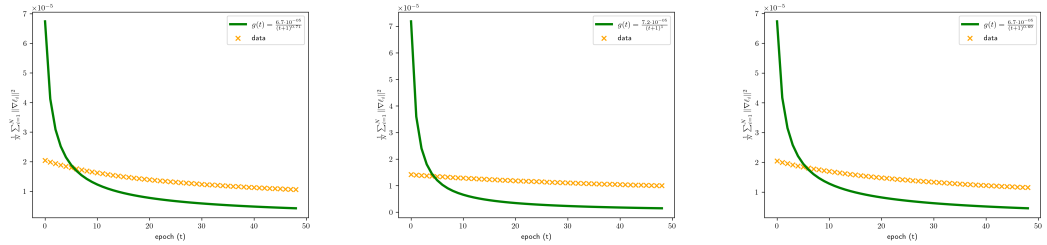
for the fitting procedure of the function $\Omega^{N,\ell}(t)$ with respect to the function specified in (4.5) with parameters bounded in a positive interval $C, \rho \in [0, \infty)$. The procedure `scipy.curve_fit` uses a Trust Region Reflective algorithm. However, a good choice of the trust region is necessary since if it is too large, the model minimizer may be far from the minimizer of the objective function in the region; contrariwise if the region is too small the Trust Region Reflective algorithm misses an opportunity to take a substantial step that will move it far closer to the minimizer of the objective function[36, Chapter 4]. For this reason, we tried two different trust region: $A = [0, 1] \times [0, 1]$ and $B = [0, 3] \times [0, 3]$ for the parameters C, ρ. When we estimated the parameters C and ρ, in case of the models trained on CIFAR10, and a converging trend was evident, we obtained a very low standard deviation using both A and B as trust region. Therefore, we used the



(a) CIFAR10-B11t (b) CIFAR10-C11t (c) CIFAR10-B11c

Figure 4.9: Constrained non-linear least-squares regression to fit the function $g(t) = {}^{C}/_{(t+1)^{\rho}}$ to the data of $\Omega^{12500,\text{CrossEntropy-loss}}$ in case of CIFAR10 dataset. All models are $L = 10000$

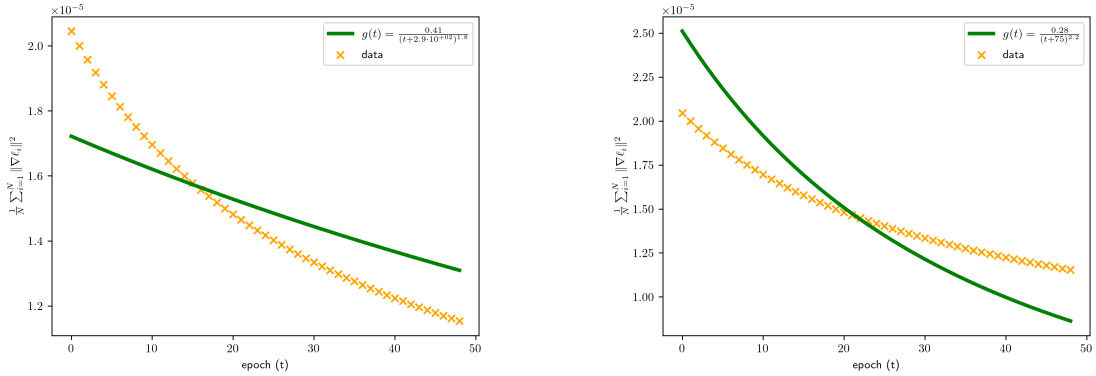
largest of the two, i.e. B, as the preferred trust region. Indeed, we can see that the results seem rather appealing as shown in Figure 4.8. The standard deviation of the parameters in the case of ResNet is significantly smaller than the estimated value. For example, in case of Figure 4.8a, we have the estimated parameters $\widehat{C} \approx 7.6638 \cdot 10^{-5}$ and $\widehat{\rho} \approx 5.9349 \cdot 10^{-3}$, with their respective standard deviation being $\widehat{\sigma}_C \approx 2.1183 \cdot 10^{-8}$ and $\widetilde{\sigma}_\rho \approx 9.1146 \cdot 10^{-5}$. In all cases pictured in Figure 4.8, we see that the tail of the data points is below the estimated trend, thus proving that the condition (3.33) is satisfied at the end of the training. We do not include all the estimated parameters values and their standard deviation for brevity, albeit we include a further analysis in case of a more realistic model below. However, in case there is no clear convergence trend, the condition in (3.33) does not appear realistic as showcased in Figure 4.9. Therefore, it seems clear that the model convergence affects whether condition (3.33) holds or not as expected by the definition of $\Omega(t)$. For the sake of precision, it is not be a mere dependence on the convergence, but more of a gamut depending on the convergence rate. Indeed,



(a) C10k-C13t (b) C10k-C23t (c) C10k-B33t

Figure 4.10: Constrained non-linear least-squares regression to fit the function $g(t) = {}^{C}/_{(t+1)^{\rho}}$ to the data of $\Omega^{10000,\text{MSE-loss}}$ in case of C10k dataset. All models are for $L = 16384$.

Assumption 3.5.1 does not hold as portrayed in Figure 4.10, although the loss on the training dataset is well-below 0.004 and diminishes throughout the training procedure. In this case, it could be caused by the fact that the loss is already quite small at initialization. In a certain sense, this could be intuitively explained as if we were at the end of the learning procedure. Therefore, the intuition to model this behaviour may be to consider the class of functions $g(t) = {}^{C}/_{(t+T_0)^{\rho}}$ with an additional shift parameter $T_0 > 1$. But by further adding a shift parameter T_0,



(a) Model C10k-B33t with the parameter $T_0 \in [0, 500]$.

(b) Model C10k-C33t with the parameter $T_0 \in [0, 100]$.

Figure 4.11: Constrained non-linear least-squares regression to fit the function $g(t) = {}^{C}/_{(t+T_0)^{\rho}}$ to the data of $\Omega^{10000,\text{MSE-loss}}$ in case of C10k dataset. All models are for $L = 16384$

with trust region $T_0 \in [0, 500]$ we obtain $T_0 \approx 288.57$ with $\widehat{\sigma}_{T_0} \approx 2507.9$ in case of C10k-B33t with $L = 16384$ as in Figure 4.11a. Trying with a smaller trust

region lead to unreliable models, e.g. for $T_0 \in [0, 100]$ we obtain $T_0 \approx 74.849$ with standard deviation $\widehat{\sigma}_{T_0} \approx 102.12$, see Figure 4.11b. Concluding that we cannot verify the stronger hypothesis, namely (4.5), even with the additional shift parameter. Moreover, the weaker condition assumed in section 3.5 seems not convincing since there is no strong decrement in the last epochs of Ω. Concluding that the assumption (3.33) is clearly dependent on the convergence of the model, besides the model and the dataset.

4.3.1 Realistic `resnet` model

This part is devolved to test Assumption 3.5.1 in case of a more realistic Residual Network model. We test it on the CIFAR10 dataset and show that a similar behaviour of $\Omega(t)$ can be considered even in case of batch gradient descent optimization. The model architecture is described as in [23], but we apply few tweaks to include the self-regularizing property of δ_L used in the theoretical results discussed in chapter 3. Namely, each block's forward pass is given by, for $t = 0, \dots N_{\text{blocks}} - 1$

$$\begin{cases} p_t = \delta_L \operatorname{relu}(F^1_{\text{BN}}(F^1_{\text{CN}}(h_t))) \\ q_t = \delta_L F^2_{\text{BM}}(F^2_{\text{CN}}(p_t)) \\ h_{t+1} = \operatorname{relu}(h_t + q_t) \end{cases} \tag{4.6}$$

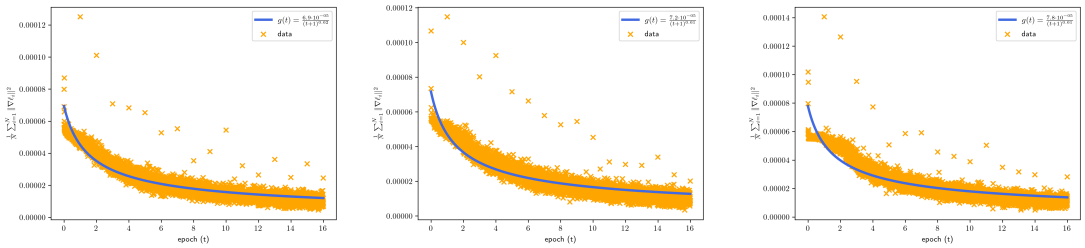
with $F^i_{\text{BM}}, F^i_{\text{CN}}$ respectively denoting a Batch normalization layer [26] and a Convolution layer [31]; $\delta_L = L^\alpha$ with $\alpha = 0.75$ and $L = 56, 104, 224$, i.e. with $9, 17, 37$ blocks respectively for each of the convolution layer filters of size 16, 32 and 64. Hereafter, we denote with `resnet56`, `resnet104`, `resnet224` respectively the aforementioned model with 56, 104 and 224 layers. In [23] they employ different data augmentations, however we do not implement most of the data augmentations, because we are not trying to compete with a highly engineered model, but show that in a similar setting to the one described in chapter 3, hypothesis (3.33) holds. We use batch gradient descent optimization instead of gradient descent for computational purposes, with batch size 128 and a constant learning rate $\eta(t) = L^{\alpha-\beta-0.05} = L^{0.45}$. This choice is similar to the scaling of the learning rate we used in the previous section, which is another evidence that the learning

rate fixed is quite effective.
We used the following simple data augmentation for training: 4 pixels are padded on each side, and a 32×32 crop is randomly sampled from the padded image, we achieve decent results in training as shown in Table 4.7 on the right, with only 16 epochs of training or equivalently 6256 steps.

L	Test Accuracy (%)
56	82.03
104	84.49
224	81.54

Table 4.7: Accuracy of the trained `resnet` model.

Table 4.8 shows that the estimated values are significant, since the standard deviation is quite low. To reduce over-fitting because of the noise introduced by the SGD training procedure, we considered the result of $\Omega_{\text{SGD}}(t)$ for $t \in (0, 16)$ where the time t is considered continuous since at t-th SGD step we are at epoch time $t/391$ because each epoch is composed by 391 batches.



(a) Model `resnet56` with $L = 56$ layers.

(b) Model `resnet104` with $L = 104$ layers.

(c) Model `resnet224` with $L = 224$ layers.

Figure 4.12: Constrained non-linear least-squares regression to fit the function $g(t) = C/(t+1)^{\rho}$ to the data of $\Omega_{\text{SGD}}^{\text{CrossEntropy-loss}}$ in case of CIFAR10 dataset.

L	$\widehat{C}$	$\widehat{\rho}$	$\widehat{\sigma}_C$	$\widehat{\sigma}_\rho$
56	$6.9462 \cdot 10^{-5}$	$6.1903 \cdot 10^{-1}$	$2.6181 \cdot 10^{-7}$	$2.5190 \cdot 10^{-3}$
104	$7.1716 \cdot 10^{-5}$	$6.0978 \cdot 10^{-1}$	$2.9322 \cdot 10^{-7}$	$2.7127 \cdot 10^{-3}$
224	$7.7803 \cdot 10^{-5}$	$6.1095 \cdot 10^{-1}$	$3.4228 \cdot 10^{-7}$	$2.9215 \cdot 10^{-3}$

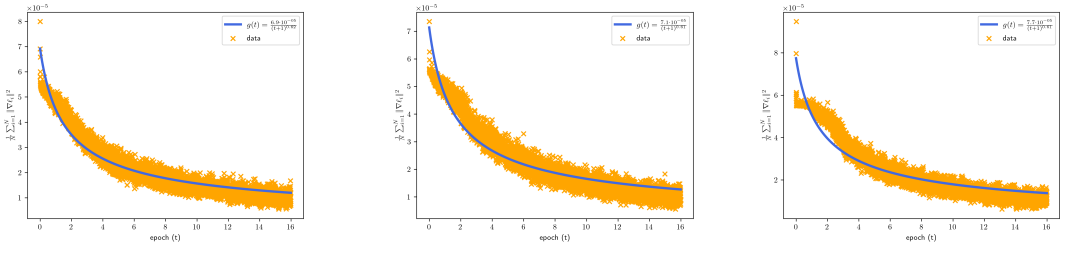
Table 4.8: Estimated parameters C, ρ for `resnet56`, `resnet104` and `resnet224` models trained on CIFAR10.

There are few data points scattered that may increase the value of C and ρ, but the fitting procedure seems still consistent to (3.33). We perform an additional filtering of the data, by leaving out the maximum and minimum value in a window of size 27, because of the intrinsic stochastic nature of batch gradient descent.

Indeed, during training a batch may be either a "poor" batch, namely a batch over-represented by a category of objects that are hard to classify for the `resnet` model, e.g. cats and dogs are generally hard to distinguish, or a "good" batch, e.g. only consisting of cats. This filtering slightly decrease the standard deviation of the estimated parameters $\widehat{C}, \widehat{\rho}$ as shown in Table 4.9 and the plots are qualitatively more appealing Figure 4.13.

L	$\widehat{C}$	$\widehat{\rho}$	$\widehat{\sigma}_C$	$\widehat{\sigma}_\rho$
56	$6.9202 \cdot 10^{-5}$	$6.1956 \cdot 10^{-1}$	$2.2533 \cdot 10^{-7}$	$2.1781 \cdot 10^{-3}$
104	$7.1352 \cdot 10^{-5}$	$6.0962 \cdot 10^{-1}$	$2.5640 \cdot 10^{-7}$	$2.3852 \cdot 10^{-3}$
224	$7.7403 \cdot 10^{-5}$	$6.1076 \cdot 10^{-1}$	$3.1174 \cdot 10^{-7}$	$2.6756 \cdot 10^{-3}$

Table 4.9: Estimated parameters C, ρ for `resnet56`, `resnet104` and `resnet224` models trained on CIFAR10 after removing minimum and maximum values in a window of size 27.



(a) Model resnet56 with $L = 56$ layers.

(b) Model resnet104 with $L = 104$ layers.

(c) Model resnet224 with $L = 224$ layers.

Figure 4.13: Constrained non-linear least-squares regression to fit the function $g(t) = {}^{C}/_{(t+1)^{\rho}}$ to the data of $\Omega_{\text{SGD}}^{\text{CrossEntropy-loss}}$ in case of CIFAR10 dataset. The data is filtered leaving out the maximum and minimum value for each interval of 27 data points.

The additional filtering did not highlight any unexpected result, but the consistency between the two methods strengthened the robustness of Assumption 3.5.1. Concluding that the assumption (3.33) could be more general, e.g. not related to the optimizer used, and could lead to faster and better training procedure. For example, a stopping criterion based on the first few mini-batches to estimate the parameters C, ρ and deduce the number of epochs required to achieve a loss value bounded by ε.

Appendix A

Miscellanea of multidimensional results

I Tensor product

Definition I.1 (Tensor product)**.** The tensor product $(V \otimes W, g)$ of two vector spaces V, W is a vector space and a bilinear map $g\colon V \times W \to V \otimes W$ such that, for every bilinear map $f\colon V \times W \to P$, there is a unique linear map φ, such that the following diagram commutes.

$$\begin{array}{ccc} & V \times W & \\ {\scriptstyle g}\swarrow & & \searrow{\scriptstyle f} \\ V \otimes W & \xrightarrow[\varphi]{} & P \end{array} \tag{A.1}$$

In particular, it is easy to see that in case V and W are two finite dimensional vector spaces with vector basis $\{v_i\}_{i=1}^{d_V}$ and $\{w_i\}_{i=1}^{d_W}$, then $\dim V \otimes W = d_V \cdot d_W$ and $\{v_i \otimes w_j\}_{i,j=1}^{d_V, d_W}$ is a basis of the tensor product vector space. In particular, any element $a \in V \otimes W$ can be represented as

$$a = \sum_{i,j=1}^{d_V, d_W} a_{ij} v_i \otimes w_j. \tag{A.2}$$

This notation can be easily generalized to the tensor product of k-vector spaces $V_1, \ldots, V_k$. Denoting with $\{v_i^j\}_{i=1}^{d_{V_j}}$ a basis for the vector space V_j with dimension d_j, then if $a \in V_1 \otimes \ldots V_k$,

$$a = \sum_{i_1,\ldots,i_k=1}^{d_{V_1},\ldots,d_{V_k}} a_{i_1 \ldots i_k} v_{i_1}^1 \otimes \cdots \otimes v_{i_k}^k.$$

For simplicity, we consider vector spaces $\mathbb{R}^d$ with $d > 0$, although the tensor product may be defined on modules as described in [33]. We denote $\{e_i\}_{i=1}^d$ the canonical orthonormal basis on $\mathbb{R}^d$, namely if we identify with $(v)_j$ the j-th coordinate of a vector $v \in \mathbb{R}^d$, then

$$(e_k)_j = \begin{cases} 1 & \text{if} j = k \\ 0 & \text{otherwise.} \end{cases}$$

For simplicity, we introduce the notation $(\mathbb{R}^d)^{\otimes k}$ to denote the tensor product of k spaces $\mathbb{R}^d$, with the convention that for $k = 0$ we have $(\mathbb{R}^d)^{\otimes 0} = \mathbb{R}$. Then, an element $T \in (\mathbb{R}^d)^{\otimes k}$ may be represented with respect to the canonical orthonormal basis as

$$T = \sum_{1 \le i_1, \dots, i_k \le d} T_{i_1,\dots,i_k} e_{i_1} \otimes \cdots \otimes e_{i_k}.$$

Definition I.2. Given a Hilbert space H over $\mathbb{R}$, we denote with H^* the dual vector space

$$H^* \coloneqq \{g \colon H \to \mathbb{R} \mid g \text{ linear and continuous}\}.$$

In particular we see that in case of finite dimensional vector spaces on $\mathbb{R}^d$, the canonical basis for the dual vector space may be defined starting from the canonical orthonormal basis by

$$e_i^* \colon \mathbb{R}^d \to \mathbb{R}, \quad e_i^*(e_j) = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{otherwise} \end{cases}$$

Finally, we note that finite dimensional spaces over $\mathbb{R}$ are self-dual and $((\mathbb{R}^d)^*)^{\otimes k}) \simeq (\mathbb{R}^d)^{\otimes k}$. This remark is of particular relevance since it allows to consider tensors as operators or as a "multidimensional array" once we fix an orthonormal basis.

II Norm

Because each element $T \in (\mathbb{R}^d)^{\otimes k}$ could be thought of as an element in a vector space $\mathbb{R}^{d^k}$, it becomes straightforward to consider the canonical inner product of $\mathbb{R}^{d^k}$ and extend it to $(\mathbb{R}^d)^{\otimes k}$. Indeed, $(\mathbb{R}^d)^{\otimes k}$ is a Hilbert space whose inner-product is defined as

$$\langle T, S \rangle_{(\mathbb{R}^d)^{\otimes k}} \coloneqq \sum_{1 \le i_1, \dots, i_k \le d} T_{i_1,\dots,i_k} S_{i_1,\dots,i_k},$$

whose associated norm is denoted with $\|T\|^2_{(\mathbb{R}^d)^{\otimes k}} = \langle T, T \rangle_{(\mathbb{R}^d)^{\otimes k}}$. Abusing notation, we denote $\|T\| = \|T\|_{(\mathbb{R}^d)^{\otimes k}}$ as the aforementioned norm for $k \in \mathbb{N}$, without distinguishing the power of the tensor product. We only require that it is compatible with the argument's dimension. In chapter 3 we only require upper bounds,

therefore this change in notation, is not obfuscating since the "sub-multiplicativity" property holds as proved in the following lemma.

Lemma II.1. *Let $n, k \in \mathbb{N}$ and $n > k$ we denote $T \in (\mathbb{R}^d)^{\otimes k} \otimes ((\mathbb{R}^d)^{\otimes(n-k)})^*$, and $v \in (\mathbb{R}^d)^{\otimes(n-k)}$, then*

$$\|Tv\| \leq \|T\|\|v\| \tag{A.3}$$

Proof. We denote

$$T = \sum_{i_1,\dots,i_n=1}^{d} t_{i_1\dots i_n} e_{i_1} \otimes \cdots \otimes e_{i_k} \otimes e^*_{i_{k+1}} \otimes \cdots \otimes e^*_n, \quad v = \sum_{i_1,\dots,i_{n-k}=1}^{d} v_{i_1\dots i_{n-k}} e_{i_1} \otimes \cdots \otimes e_{i_{n-k}}$$

Then, by Cauchy-Schwarz inequality,

$$\begin{aligned}
\|Tv\|^2 &= \left\| \sum_{i_1,\dots,i_k=1}^{d} \sum_{i_{k+1},\dots,i_n=1}^{d} t_{i_1\dots i_n} v_{i_{k+1}\dots i_{n-k}} e_{i_1} \otimes \cdots \otimes e_{i_k} \right\|^2 \\
&= \sum_{i_1,\dots,i_k=1}^{d} \left(\sum_{i_{k+1},\dots,i_n=1}^{d} t_{i_1\dots i_n} v_{i_{k+1}\dots i_{n-k}} \right)^2 \\
&\leq \sum_{i_1,\dots,i_k=1}^{d} \left(\sum_{i_{k+1},\dots,i_n=1}^{d} t^2_{i_1\dots i_n} \right) \left(\sum_{i_{k+1},\dots,i_n=1}^{d} v^2_{i_{k+1}\dots i_{n-k}} \right) \\
&= \|T\|^2 \|v\|^2.
\end{aligned}$$

□

Remark II.2. Lemma II.1 it is quite general, albeit we use it in case of $n \leq 3$ and $k < n$. Let,

1. $T \in (\mathbb{R}^d)^{\otimes 2} \otimes (\mathbb{R}^d)^*$, $S \in \mathbb{R}^d \otimes (\mathbb{R}^d)^* \otimes (\mathbb{R}^d)^*$.
2. $A \in \mathbb{R}^d \otimes (\mathbb{R}^d)^*$, $B \in \mathbb{R}^d \otimes \mathbb{R}^d$.
3. $v \in \mathbb{R}^d$.

Then,

$$\|Av\| \leq \|A\|\|v\| \tag{A.4}$$
$$\|Tv\| \leq \|T\|\|v\| \tag{A.5}$$
$$\|SB\| \leq \|S\|\|B\| \tag{A.6}$$

Since, the same procedure of Lemma II.1 holds, we show only the proof in case of (A.4). We denote with

$$A = \sum_{i,j=1}^{d} a_{ij} e_i \otimes e_j^*, \quad v = \sum_{i=1}^{d} v_i e_i$$

and the proof follows clearly

$$\begin{aligned} \|Av\|^2 &= \left\| \sum_{i=1}^{d} \sum_{j=1}^{d} a_{ij} v_j e_i \right\|^2 = \sum_{i=1}^{d} \left(\sum_{j=1}^{d} a_{ij} v_j \right)^2 \\ &\leq \sum_{i=1}^{d} \left(\sum_{j=1}^{d} a_{ij}^2 \right) \left(\sum_{j=1}^{d} v_j^2 \right) = \|A\|^2 \|v\|^2. \end{aligned}$$

which is exactly the proof that $\|Av\|_2 \leq \|A\|_F \|v\|_2$ with $\|\cdot\|_F$ being the Frobenious norm and $\|\cdot\|_2$ being the euclidean norm in $\mathbb{R}^d$. Moreover, we see that we can compose the inequalities, for example from (A.4) and (A.5), it holds

$$\|T(Av)\| \leq \|T\| \|Av\| \leq \|T\| \|A\| \|v\|.$$

We proceed to show that in case of activation functions that are sub-linear, the application by component of the function preserves the sub-linearity property through the norm described in this appendix.

Lemma II.3. *Let $\sigma \colon \mathbb{R} \to \mathbb{R}$ be a Lipschitz function with Lipschitz constant $L > 0$ and $\sigma(0) = 0$. Let $v \in \mathbb{R}^d$ and denote with $\sigma(v) = (\sigma(v_1), \ldots, \sigma(v_d))^\top$. Then,*

$$\|\sigma(v)\| \leq L \|v\|.$$

Proof. The proof is straightforward and builds upon the proof of Lemma II.1,

$$\begin{aligned} \|\sigma(v)\|^2 &= \left\| \sum_{i=1}^{d} \sigma(v_i) e_i \right\| = \sum_{i=1}^{d} \sigma(v_i)^2 \\ &\leq \sum_{i=1}^{d} L^2 v_i^2 = L^2 \|v_i\|^2. \end{aligned}$$

□

Remark II.4. Note that by Lemma II.1, we also have

$$\|\sigma(Av)\| \leq L \|A\| \|v\|$$

under the constraints of Lemma II.3, which is of particular utility in chapter 3.

Appendix B

Data and implementation

The code used for the experimental results of chapter 4 is available at https://github.com/ladezai/residual-network. The experiments relies heavily on PyTorch [37] and PyTorch Lightning frameworks, though the complete list of libraries used is listed in Table B.1. For more information regarding the usage of the scripts contained in the repository, we defer to `README.md` of https://github.com/ladezai/residual-network. The model's checkpoints are available at https://drive.google.com/drive/folders/1WYcLeW_T_h-Vjg61Ov-NAyElF9ud59cMt?usp=sharing[1]. In the following, we give a brief over-view of the repository's structure.

Framework	Repository Link
PyTorch	Github Link
PyTorch Lightning	Github Link
Tensorboard	Github Link
SciPy	Github Link
Numpy	Github Link
p-var	Github Link

Table B.1: Framework used for the experiments.

I Model training

The Residual Network model is implemented in `resnet.py` as described in (2.4). Various hyper-parameters are available to further experiment, such as non-constant initialization, modifications to the skip blocks and different activation functions.

[1]The repository is roughly 6GB.

The dataset used, i.e. CIFAR10 and C10k, are loaded through the `dataset.py`. The parameters for the CIFAR10 dataset were `training_data_percentage` = 0.25, while for C10k we used `num_samples` = 10^5 and `c` = 0.01. The training procedure and the Command Line Interface (CLI) in `main.py` via the Pytorch Lightning CLI. In `main.py` is implemented the logging of the p-variation for a given p, $p/2$ and $3p/4$, the computation of the average norm of the loss gradients, as in (3.33), at each gradient descent step. For the experiments using the `resnet` model described in [23] for the CIFAR10 dataset, we used the script `resnet_cifar10.py` that train the model as described in subsection 4.3.1, logs the average norm of the loss gradients, as in (3.33), for each stochastic gradient descent step and test the accuracy.

II Loss landscape and paths visualization

The images of the loss landscape were generated using `visualization.py`, which provides a CLI to easily render a loss landscape surface from a model checkpoint. The model hyper-parameter used in Figure 1.2 and Figure 2.1 are reported in Table B.2; when the figure is said to not have the skip connection, the residual network becomes a plain feed-forward neural network. Although, the parameter α does not play a role in the forward pass when there is no skip-connection, it plays an indirect role as $\beta = 1 - \alpha$ and β is used in the initialization. Both the models were trained on CIFAR10 dataset using 50% of the entire test-dataset, i.e. roughly 25000 images. For the visualization, we used 3 different seeds to obtain 3

d	L	α	γ	η	T_L	σ	skip-connection	initialization
50	20	0.7	2.0	2.0	50	tanh	Yes	truncated normal
50	20	1.0	100	2.0	50	tanh	No	truncated normal

Table B.2: ResNet-like model hyper-parameters.

different viewpoints on the loss landscape for both models as shown in Figure 2.1. The trajectories of the weights norm, e.g. Figure 4.2, Figure 4.5, plotted the linear interpolation of the points $\|L^\beta W_k^{(L)}\|$ with respect to the k index layer. For these visualizations, we implemented the script `norm_path.py`. The script computes the p-variation of the paths of the norms and the p-variation of the path $L^\beta W_k^{(L)}$ as well. The curve fitting procedure via Trust Region Reflective algorithm using SciPy's implementation, is available in `curve_fit_avg_grad.py`. The script provides an estimate of the parameters with their standard deviance and standard error, finally stores the plots of the curve with respect to the data-points, e.g. Figure 4.13. The images of specified in Figure 3.1 portray the evolution of selected

weights throughout the depth of the ResNet rescaled by their respective L^{β}. The code is available in the script `path_weights.py`.

III Additional 1-variation data

For completeness in this section, we collected the data related to the computation of the 1-variation of the models described in chapter 4, which we did not include for brevity reasons in section 4.2.

L	CIFAR10-B11c	CIFAR10-B11t	CIFAR10-B31c	CIFAR10-B31t
2000	0.166	5.691	536.1	354.3
3000	0.1083	5.703	300.6	1071
4000	0.06997	5.708	583.5	2337
5000	0.05813	5.713	779.4	2383
6000	0.04629	5.714	1909	1139
7000	0.03621	5.716	2333	1171
8000	0.03023	5.716	5528	4604
9000	0.02719	5.716	$7.256 \cdot 10^5$	2115
10000	0.02207	5.715	3210	6041
11000	0.01896	5.716	6599	4098
12000	0.01613	5.716	4875	5877
L	CIFAR10-C11c	CIFAR10-C11t	CIFAR10-C12c	CIFAR10-C12t
2000	∞	∞	3.123	6.48
3000	∞	∞	3.264	6.693
4000	∞	∞	3.303	6.641
5000	∞	∞	3.447	6.665
6000	∞	∞	3.716	6.722
7000	∞	∞	3.552	6.74
8000	∞	∞	3.481	6.688
9000	∞	∞	3.858	6.809
10000	∞	$2.445 \cdot 10^{31}$	3.672	6.86
11000	∞	∞	3.801	6.821
12000	∞	∞	3.641	6.842

Table B.3: Computation of $\|\mathcal{W}^{(L)}\|_{1-\text{var};[0,1]}$ in case of the different specified model on CIFAR10 dataset. The maximum value out of 3 sample models is reported.

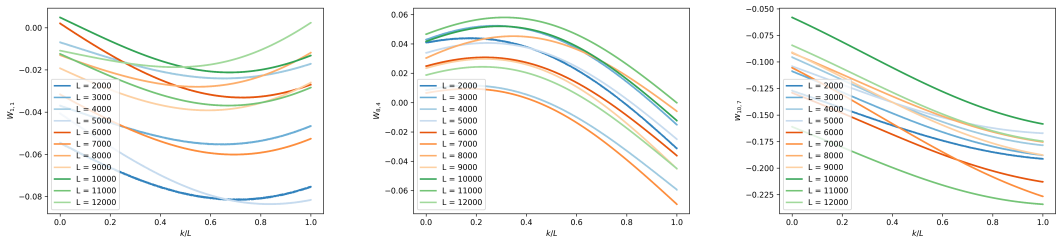
L	C10k-B21c	C10k-B22c	C10k-B23c	C10k-B31c	C10k-B32c	C10k-B33c
128	0.005953	0.06785	0.02784	0.001726	$8.756 \cdot 10^{-5}$	0.002555
256	0.00384	0.0593	0.02221	0.001436	0.0001119	0.001429
512	0.002566	0.04745	0.01805	0.001128	0.000129	0.0008157
1024	0.00151	0.03385	0.01359	0.0007843	0.0001181	0.0004325
2048	0.001007	0.02458	0.01096	0.0005569	0.0001158	0.0002506
4096	0.0006001	0.01671	0.008315	0.0003412	0.0001075	0.0001514
8192	0.0003758	0.01135	0.0064	0.0002062	0.0001628	0.0001799
16384	0.0002548	0.007459	0.004783	0.0001657	0.0002629	0.0002731
32768	0.0002724	0.004921	0.003621	0.0007312	0.0005465	0.0005312
L	C10k-B21t	C10k-B22t	C10k-B23t	C10k-B31t	C10k-B32t	C10k-B33t
128	0.336	0.3375	0.3364	0.3385	0.3359	0.3364
256	0.3248	0.3254	0.325	0.3262	0.3248	0.325
512	0.3314	0.3317	0.3316	0.3323	0.3315	0.3316
1024	0.3218	0.3219	0.3218	0.3222	0.3218	0.3218
2048	0.3192	0.3172	0.3167	0.3232	0.3207	0.3208
4096	0.3193	0.3174	0.3174	0.3203	0.3203	0.3203
8192	0.3165	0.3149	0.315	0.3179	0.3172	0.3172
16384	0.3151	0.3139	0.314	0.3156	0.3155	0.3155
32768	0.3133	0.3133	0.3133	0.3133	0.3136	0.3136
L	C10k-C12c	C10k-C12t	C10k-C21c	C10k-C21t	C10k-C22c	C10k-C22t
128	0.0006413	0.3375	0.001847	3.831	0.0001022	3.801
256	$6.199 \cdot 10^{-5}$	0.326	0.001514	5.221	0.0001326	5.197
512	$6.676 \cdot 10^{-5}$	0.3324	0.001166	7.521	0.0001502	7.5
1024	0.0003242	0.3223	0.0007892	10.31	0.0001354	10.3
2048	0.001385	0.3289	0.0005322	14.66	0.0001289	14.52
4096	0	0.3202	0.001684	20.65	0.0001322	20.5
8192	0	0.3117	0.004433	28.85	0.0001849	28.71
16384	0	0.3135	0.01412	40.53	0.0004213	40.39
32768	0	2.477	0.03595	56.92	0.00091	56.76

Table B.4: Computation of $\|\mathcal{W}^{(L)}\|_{1-\text{var};[0,1]}$ in case of the different specified model on C10k dataset. The maximum value out of 3 sample models is reported.

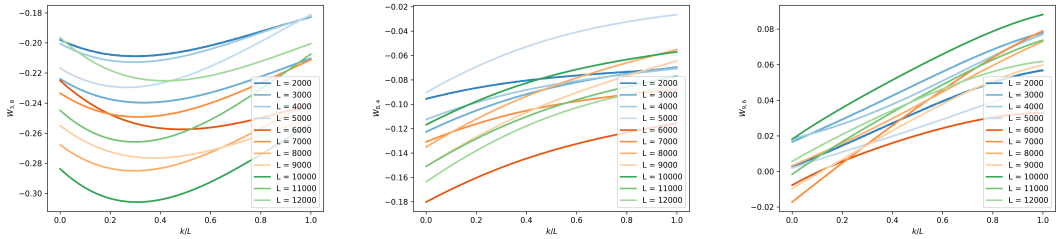
IV Additional images

We provide additional visuals to enhance the credibility of Theorem 3.4.7. We further observe that the trajectories initial point is not subtracted to the paths, although the convergence result is in the 1-variation semi-norm, for visualization purposes. We note that even with the same hyper-parameters but different value of

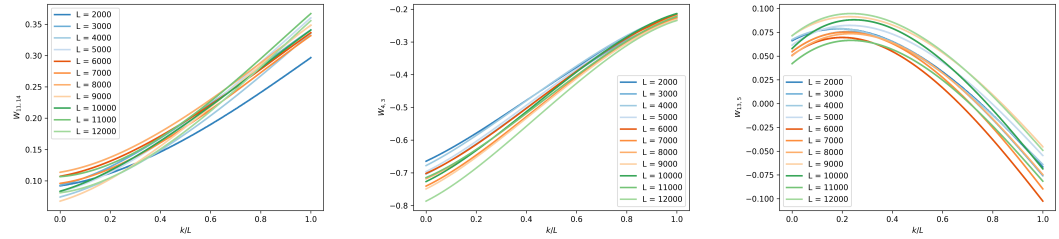
L, changes the trajectories. However, most of the times the trend is rather similar or even indistinguishable up to a translation. The only notable difference is with regard to the CIFAR10-C22t where the paths are visibly rougher as expected from previous analysis in chapter 4.



(a) CIFAR10-B12t

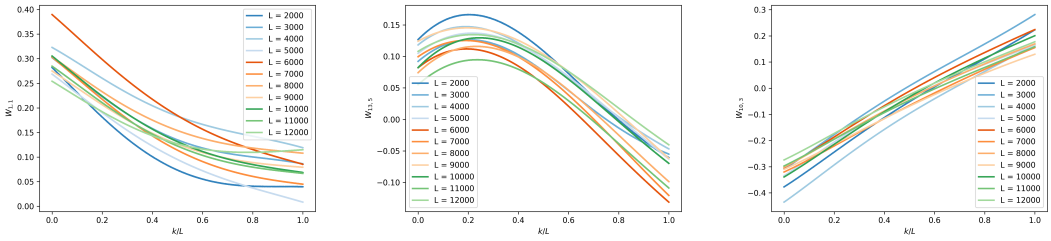


(b) CIFAR10-B22t

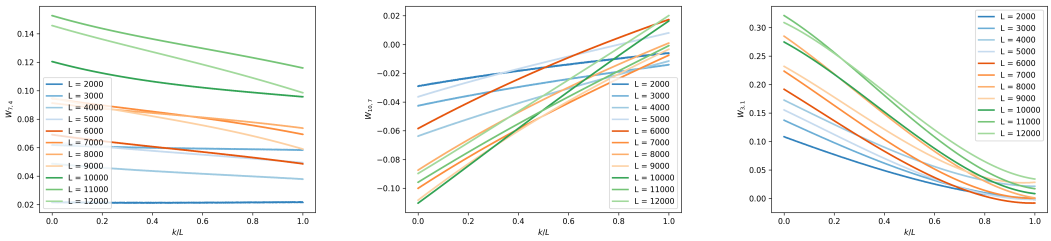


(c) CIFAR10-B33t

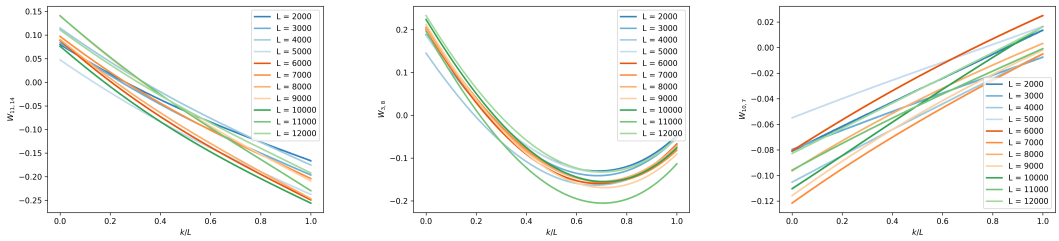
Figure B.1: Trajectory of the paths $\mathfrak{W}$ for a selected choice of coordinates and the specified model hyper-parameters.



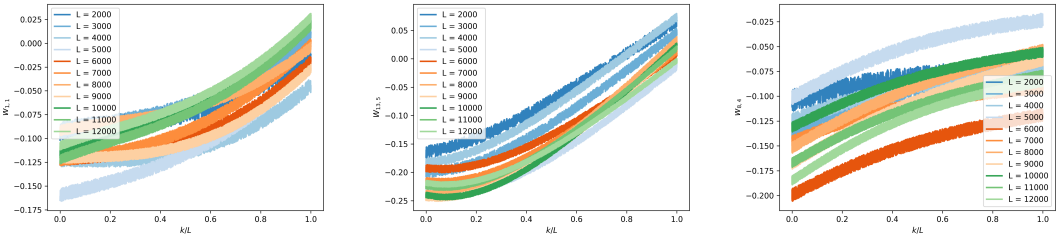
(a) CIFAR10-B21t



(b) CIFAR10-B32t



(c) CIFAR10-C12t



(d) CIFAR10-C22t

Figure B.2: Trajectory of the paths $\mathfrak{W}$ for a selected choice of coordinates and the specified model hyper-parameters.

Bibliography

[1] Zeyuan Allen-Zhu, Yuanzhi Li, and Zhao Song. "A Convergence Theory for Deep Learning via Over-Parameterization". In: (2018). arXiv: 1811.03962. URL: http://arxiv.org/abs/1811.03962.

[2] Lei Jimmy Ba and Rich Caruana. "Do Deep Nets Really Need to be Deep?" In: *CoRR* abs/1312.6184 (2013). arXiv: 1312.6184. URL: http://arxiv.org/abs/1312.6184.

[3] David Balduzzi et al. *The Shattered Gradients Problem: If resnets are the answer, then what is the question?* 2018. arXiv: 1702.08591 [cs.NE].

[4] Christian Bayer, Peter K. Friz, and Nikolas Tapia. "Stability of Deep Neural Networks via Discrete Rough Paths". In: *SIAM Journal on Mathematics of Data Science* 5.1 (Feb. 2023), pp. 50–76. DOI: 10.1137/22m1472358. URL: https://doi.org/10.1137%2F22m1472358.

[5] Y. Bengio, P. Simard, and P. Frasconi. "Learning long-term dependencies with gradient descent is difficult". In: *IEEE Transactions on Neural Networks* 5.2 (1994), pp. 157–166. DOI: 10.1109/72.279181.

[6] Mose Blanchard and M. Amine Bennouna. "The Representation Power of Neural Networks: Breaking the Curse of Dimensionality". In: *CoRR* abs/2012.-05451 (2020). arXiv: 2012.05451. URL: https://arxiv.org/abs/2012.05451.

[7] Vygantas Butkus and Rimas Norvaia. "Computation of p-variation". English. In: *Lith. Math. J.* 58.4 (2018), pp. 360–378. ISSN: 0363-1672. DOI: 10.1007/s10986-018-9414-3.

[8] Alain-Sam Cohen, Rama Cont, Alain Rossier, and Renyuan Xu. *Scaling Properties of Deep Residual Networks*. 2021. arXiv: 2105.12245 [cs.LG].

[9] Rama Cont, Alain Rossier, and RenYuan Xu. *Convergence and Implicit Regularization Properties of Gradient Descent for Deep Residual Networks*. 2023. arXiv: 2204.07261 [cs.LG].

[10] G. Cybenko. "Approximation by superpositions of a sigmoidal function". English. In: *Math. Control Signals Syst.* 2.4 (1989), pp. 303–314. ISSN: 0932-4194. DOI: 10.1007/BF02551274.

[11] Tim De Ryck, Samuel Lanthaler, and Siddhartha Mishra. "On the approximation of functions by tanh neural networks". In: *Neural Networks* 143 (2021), pp. 732–750. ISSN: 0893-6080. DOI: https://doi.org/10.1016/j.neunet.2021.08.015. URL: https://www.sciencedirect.com/science/article/pii/S0893608021003208.

[12] Jia Deng et al. "ImageNet: A large-scale hierarchical image database". In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. 2009, pp. 248–255. DOI: 10.1109/CVPR.2009.5206848.

[13] Li Deng. "The mnist database of handwritten digit images for machine learning research". In: *IEEE Signal Processing Magazine* 29.6 (2012), pp. 141–142.

[14] Weinan E. "A proposal on machine learning via dynamical systems". English. In: *Commun. Math. Stat.* 5.1 (2017), pp. 1–11. ISSN: 2194-6701. DOI: 10.1007/s40304-017-0103-z.

[15] Weinan E, Chao Ma, and Lei Wu. *The Barron Space and the Flow-induced Function Spaces for Neural Network Models*. 2021. arXiv: 1906.08039 [cs.LG].

[16] Peter K. Friz and Nicolas B. Victoir. *Multidimensional stochastic processes as rough paths. Theory and applications.* English. Vol. 120. Camb. Stud. Adv. Math. Cambridge: Cambridge University Press, 2010. ISBN: 978-0-521-87607-0. DOI: 10.1017/CBO9780511845079.

[17] Lucio Galeati. "Nonlinear Young differential equations: a review". English. In: *J. Dyn. Differ. Equations* 35.2 (2023), pp. 985–1046. ISSN: 1040-7294. DOI: 10.1007/s10884-021-09952-w.

[18] Xavier Glorot and Yoshua Bengio. "Understanding the difficulty of training deep feedforward neural networks". In: *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*. Ed. by Yee Whye Teh and Mike Titterington. Vol. 9. Proceedings of Machine Learning Research. Chia Laguna Resort, Sardinia, Italy: PMLR, 13–15 May 2010, pp. 249–256. URL: https://proceedings.mlr.press/v9/glorot10a.html.

[19] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. http://www.deeplearningbook.org. MIT Press, 2016.

[20] Ian J. Goodfellow, Oriol Vinyals, and Andrew M. Saxe. *Qualitatively characterizing neural network optimization problems*. 2015. arXiv: 1412.6544 [cs.NE].

[21] Eldad Haber, Lars Ruthotto, and Elliot Holtham. "Learning across scales - A multiscale method for Convolution Neural Networks". In: *CoRR* abs/1703.-02009 (2017). arXiv: 1703.02009. URL: http://arxiv.org/abs/1703.02009.

[22] Soufiane Hayou et al. "Stable ResNet". In: *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*. Ed. by Arindam Banerjee and Kenji Fukumizu. Vol. 130. Proceedings of Machine Learning Research. PMLR, Apr. 2021, pp. 1324–1332. URL: https://proceedings.mlr.press/v130/hayou21a.html.

[23] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep Residual Learning for Image Recognition". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 770–778. DOI: 10.1109/CVPR.2016.90.

[24] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. "Multilayer feedforward networks are universal approximators". In: *Neural Networks* 2.5 (1989), pp. 359–366. ISSN: 0893-6080. DOI: https://doi.org/10.1016/0893-6080(89)90020-8. URL: https://www.sciencedirect.com/science/article/pii/0893608089900208.

[25] Martin Hutzenthaler et al. *Convergence proof for stochastic gradient descent in the training of deep neural networks with ReLU activation for constant target functions*. 2023. arXiv: 2112.07369 [cs.LG].

[26] Sergey Ioffe and Christian Szegedy. "Batch normalization: accelerating deep network training by reducing internal covariate shift". In: *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*. ICML'15. Lille, France: JMLR.org, 2015, pp. 448–456.

[27] Kenji Kawaguchi. "Deep learning without poor local minima". In: *Proceedings of the 30th International Conference on Neural Information Processing Systems*. NIPS'16. Barcelona, Spain: Curran Associates Inc., 2016, pp. 586–594. ISBN: 9781510838819.

[28] Alex Krizhevsky. "Learning Multiple Layers of Features from Tiny Images". In: 2009. URL: https://api.semanticscholar.org/CorpusID:18268744.

[29] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. "ImageNet classification with deep convolutional neural networks". In: *Commun. ACM* 60.6 (May 2017), pp. 84–90. ISSN: 0001-0782. DOI: 10.1145/3065386. URL: https://doi.org/10.1145/3065386.

[30] Alina Kuznetsova et al. “The Open Images Dataset V4: Unified image classification, object detection, and visual relationship detection at scale”. In: *IJCV* (2020).

[31] Y. LeCun et al. “Backpropagation Applied to Handwritten Zip Code Recognition”. In: *Neural Computation* 1.4 (1989), pp. 541–551. DOI: 10.1162/neco.1989.1.4.541.

[32] Yann A. LeCun, Léon Bottou, Genevieve B. Orr, and Klaus-Robert Müller. “Efficient BackProp”. In: *Neural Networks: Tricks of the Trade: Second Edition*. Ed. by Grégoire Montavon, Geneviève B. Orr, and Klaus-Robert Müller. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 9–48. ISBN: 978-3-642-35289-8. DOI: 10.1007/978-3-642-35289-8_3. URL: https://doi.org/10.1007/978-3-642-35289-8_3.

[33] D. J. Lewis, B. R. McDonald, M. F. Atiyah, and I. G. MacDonald. “Introduction to Commutative Algebra.” In: *The American Mathematical Monthly* 77.7 (Aug. 1970), p. 783. ISSN: 0002-9890. DOI: 10.2307/2316241. URL: http://dx.doi.org/10.2307/2316241.

[34] Hao Li et al. *Visualizing the Loss Landscape of Neural Nets*. 2018. arXiv: 1712.09913 [cs.LG].

[35] Marvin Minsky and Seymour Papert. *Perceptrons: An introduction to computational geometry*. MIT Press, 1969.

[36] Jorge Nocedal and Stephen J. Wright. *Numerical optimization*. English. 2nd ed. Springer Ser. Oper. Res. Financ. Eng. New York, NY: Springer, 2006. ISBN: 0-387-30303-0.

[37] Adam Paszke et al. “Automatic differentiation in PyTorch”. In: (2017).

[38] B.T. Polyak. “Some methods of speeding up the convergence of iteration methods”. In: *USSR Computational Mathematics and Mathematical Physics* 4.5 (Jan. 1964), pp. 1–17. ISSN: 0041-5553. DOI: 10.1016/0041-5553(64)90137-5. URL: http://dx.doi.org/10.1016/0041-5553(64)90137-5.

[39] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. “Learning representations by back-propagating errors”. In: *Nature* 323.6088 (Oct. 1986), pp. 533–536. ISSN: 1476-4687. DOI: 10.1038/323533a0. URL: http://dx.doi.org/10.1038/323533a0.

[40] Nitish Srivastava et al. “Dropout: A Simple Way to Prevent Neural Networks from Overfitting”. In: *Journal of Machine Learning Research* 15.56 (2014), pp. 1929–1958. URL: http://jmlr.org/papers/v15/srivastava14a.html.

[41] Pauli Virtanen et al. "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python". In: *Nature Methods* 17 (2020), pp. 261–272. DOI: 10.1038/s41592-019-0686-2.

[42] Greg Yang. *Tensor Programs I: Wide Feedforward or Recurrent Neural Networks of Any Architecture are Gaussian Processes*. 2021. arXiv: 1910.12478 [cs.NE].