Convergence of Gradient Descent for the training of Deep Residual Neural Networks

Raphaël Barboni *
raphael.barboni@ens.fr

September 28, 2022

Abstract

The following work is a small introductory note on the work I began during my Master Thesis and will continue during my PhD under supervision of G.Peyré et F-X.Vialard.

We study properties of Gradient Descent for the training of overparameterized machine learning models. Indeed, such simple first order optimization methods have been observed to efficiently train predictive models whose loss-landscape is a highly-nonconvex function. Explaining this phenomenon in generality thus became an active research subject. We propose here an introduction to the topic, exhibiting known results and sketching promising ideas.

Keywords: Neural Networks, Residual Networks, Neural ODE, Supervised learning, Nonconvex optimization, Overparametrized models

Contents

*Département Mathématiques et Applications - ENS Ulm

Figure 1: Images automatically generated on prompted text by DALLE-2
(try by yourself here: `https://openai.com/blog/dall-e/`)

1 Introduction: ResNets and Neural ODEs

The term *Neural Networks* (sometimes called *Artificial Neural Networks (ANN)*) refers to a class of algorithm whose implementation is inspired by the structure of actual biological neural systems. As is the case in the human brain, ANNs process the information sequentially through a network of interconnected processing units, usually organized in successive layers, each of them implementing a simple linear or non-linear transformation. Encouraged by the parallel development of adapted hard-ware infrastructure, the use of ANNs is now ubiquitous to a lot of regression, classification or generation tasks with impressive results (e.g. fig. 1). However, these performances remain widely unexplained. In particular, ANNs are often considered as "black-box" computing systems and, apart from some heuristics, there lacks of a mathematical understanding of their performances. Developing a general theory to understand and predict the performances of ANNs is now an important research topic in the field of *Machine Learning*.

1.1 Residual Network and Neural ODEs

Residual Networks (ResNet) [16] are ANNs architectures composed of successive computational layers and for which each layer consists in the the addition of a *skip connection*, adding the result of the previous layer, and a *residual term*, which is a novel transformation. An example of such architecture, with L layers, is the following algorithm:

$$\begin{aligned} x^{(0)} &= x_{in}, \\ x^{(l+1)} &= x^{(l)} + f_l(W^{(l)}, x^{(l)}), \quad \forall l \in \{0, ..., L-1\}, \\ F(W, x_{in}) &\overset{\text{def.}}{=} f_L(W^{(L)}, x^{(L)}), \end{aligned} \tag{1}$$

where $x_{in} \in \mathbb{R}^d$ is the input data and $F(W, x_{in})$ is the algorithm's predicted output for input x_{in} and parameter $W = (W^{(l)})_{0 \leq l \leq L}$. For each layer $l \in \{0, ..., L\}$. The residual term is $f_l : \mathbb{R}^m \times \mathbb{R}^d \to \mathbb{R}^d$, a function parameterized by $W_l \in \mathbb{R}^m$. Those are typically smaller ANNs such as the following *2-layer perceptron*:

$$f_l : x \in \mathbb{R}^d \mapsto A_l \sigma(B_l x)$$

where the parameter $W_l = (A_l, B_l) \in \mathbb{R}^{d \times m_l} \times \mathbb{R}^{m_l \times d}$ is a couple of matrices and $\sigma : \mathbb{R} \to \mathbb{R}$ is a fixed non-linearity applied componentwise. We woud refer to m_l as the *width* of layer l.

Remark 1. *Note that the architecture in eq.* (1) *is far from the architectures one would use for real world applications. In practice, successful ANN architectures alternate between various kind of layer, each with different roles in processing the data:* downsampling, upsampling, normalization *... Nonetheless, this kind of models are complex enough to be relevant yet simple enough to be well-suited for the purpose of theoretical study.*

Similarities between ResNet architectures and discrete numerical schemes motivated the introduction of *Neural Ordinary Differential Equations (NODE)*, thought to be limiting models of eq. (1) when the depth L tends to infinity with a proper re-normalization [10]. A typical example is:

$$F(W, x_{in}) \overset{\text{def.}}{=} x_1, \quad \text{with} \begin{cases} \frac{\mathrm{d}x_s}{\mathrm{d}s} &= f_s(W_s, x_s), \quad \forall s \in [0,1] \\ x_0 &= x_{in}, \end{cases} \tag{2}$$

where $x_{in} \in \mathbb{R}^d$ is the input data and the residual term is $f_s : \mathbb{R}^m \to \mathbb{R}^d$, a time-varying vector-field controlling the ODE and parameterized by $W_s \in \mathbb{R}^m$. The obtained algorithm outputs the prediction function F parameterized by the *infinite-dimensional* parameter $W = (W_s)_{s\in[0,1]}$.

On the level of theoretical analysis, Neural ODEs (eq. (2)) have several advantages compared to discrete ResNets (eq. (1)) as their formulation is somewhat more concise, leading to simpler calculations. Moreover, their analysis can be casted into the framework of the already well-established mathematical theory of optimal control [29, 13, 28]. For these reasons, the introduction of Neural ODEs brings the hope of understanding the behavior of very deep ANN architectures.

1.2 Supervised learning

As they are predictive models, we consider ResNets in a *supervised learning* setting. Formally, this means one is provided with a training set including both input data $(x^i)_{1\leq i\leq N} \in (\mathbb{R}^d)^N$ and associated target data $(y^i)_{1\leq i\leq N} \in (\mathbb{R}^{d'})^N$, where $N \geq 1$ is the size of the dataset. Given a *loss* or *distance* function $\ell : \mathbb{R}^{d'} \times \mathbb{R}^d \to \mathbb{R}_+$, we define the (regularized) *Empirical Risk* (or *Training Loss*) associated to the predictive model F with parameter W as:

$$\mathcal{L}(W) := \sum_{i=1}^{N} \ell(F(W, x^i), y^i) + \lambda \mathcal{R}(W), \tag{3}$$

where $\mathcal{R}$ is a regularization term and $\lambda \geq 0$ is a positive constant. Training F then amounts to solve the following *Empirical Risk Minimization (ERM)* problem

$$\text{Find} \quad W^* \in \operatorname{argmin} \mathcal{L}(W). \tag{ERM}$$

Although several optimization procedure are available to solve eq. (ERM), the most popular among practitioners are first-order methods, such as "vanilla" *Gradient Descent (GD)*. For initialization W^0 the training dynamic reads:

$$W^{k+1} = W^k - \eta \nabla \mathcal{L}(W^k), \quad \forall k \geq 0 \tag{4}$$

for some *step size* $\eta > 0$. This comes along with the continuous counterpart, the limiting dynamic when $\eta \to 0$ which we refer to as *Gradient Flow*, sometimes considered for theoretical purposes:

$$\frac{\mathrm{d}}{\mathrm{d}t} W^t = -\nabla \mathcal{L}(W^t), \quad \forall t \geq 0. \tag{5}$$

Remark 2. *We use the lower indices and the discrete (resp. continuous) variable $l \in \{0, ..., L\}$ (resp. $s \in [0,1]$)) to refer to the layers of our model. On the other hand we use the upper indices and the discrete (resp. continuous) variable $k \in \mathbb{N}$ (resp. $t \in \mathbb{R}_+$) to evolution of the Gradient Descent (resp. Gradient Flow). For example W_l^k is the parameter of layer l after k step of GD.*

While convergence properties of GD are well understood for the minimization of convex functions [23], this is usually not the case in practice where people consider the *unregularized* risk, setting $\lambda = 0$ in eq. (3). In this setting the risk $\mathcal{L}$ associated to some ANN is usually a highly non-convex function of an high-dimensional parameter W. However, one still observes convergence of GD towards an optimum of the risk. Moreover, this training comes along with good generalization performances on (unseen) test data. Explaining both of these phenomena is today an active area of research.

1.3 Organization of the presentation

The purpose of this note is to expose several results concerning the convergence of Gradient Descent for the training of deep Neural Network. We are more precisely concerned with the study of deep Residual Network and Neural ODEs.

In section 2 we detail general properties of overparameterized model and propose a common proof scheme to prove convergence of GD. In section 3 we focus on the analysis of linear Neural Networks for whom global convergence results can be stated. Section 4 gives an overview of our own work: we propose a new model and interpret the associated supervised learning problem as an optimisation problem over a group of diffeomorphisms.

2 Overparametrized models: Neural Tangent Kernel and Implicit bias

The past few years saw the emergence of increasingly complex ANNs having millions, billions or even trillions of trainable parameters, which is way more than the dataset dimension. We refer to these models as *overparameterized*. ResNets appeared a few years ago as one of the first examples overparameterized architectures. Indeed, the presence of skip connections in ResNet architectures allows to deal with the problem of vanishing gradient encountered in other "deep" architectures [26]: whereas ANNs with more than 10 layers are very hard to train, ResNets with an (almost) arbitrary number of layers can be efficiently trained.

2.1 Neural Tangent Kernel analysis

A classical argument to prove convergence of GD in the training of an overparameterized model is to rely on the fact that such model can be well approximated by a linear model corresponding to its first order expansion around the initialization [19]:

$$F(W^1, x) \simeq F(W^0) + D_W F(W^0, x)(W^1 - W^0), \quad \forall W^1, W^0$$

This phenomenon, called "linear" or "kernel regime" is related to the constancy of the *Neural Tangent Kernel (NTK)* [17]. More precisely, for some prediction algorithm F parameterized by $W \in \mathbb{R}^m$, note that the gradient of the empriccal risk can be expressed as:

$$\nabla \mathcal{L}(W) = \frac{1}{N} \sum_{i=1}^{N} D_W F(W, x^i)^\top \nabla \ell(F(W, x^i), y^i).$$

Therefore, after one step of training $W^1 = W^0 - \eta \nabla \mathcal{L}(W_0)$, the output of F on input x^j becomes:

$$\begin{aligned} F(W^1, x^j) &\simeq F(W^0, x^j) - \eta \frac{1}{N} \sum_{i=1}^{N} \left[D_W F(W^0, x^j) D_W F(W^0, x^i)^\top \right] \nabla \ell(F(W^0, x^i), y^i) \\ &\simeq F(W^0, x^j) - \eta \frac{1}{N} \sum_{i=1}^{N} K(x^j, x^i) \nabla \ell(F(W_0, x^i), y^i) \end{aligned}$$

where we set $K(x^i, x^j) = D_W F(W^0, x^j) D_W F(W^0, x^i)^\top$. One can see that, along GD, the model's output evolves accoring to the gradient of the loss ℓ but for the metric defined by the kernel K, which we refer to as *Neural Tangent Kernel (NTK)*.

Definition 1 (Neural Tangent Kernel [17])**.** *For the model F parameterized by W the* Neural Tangent Kernel *of F at some W is defined by:*

$$K(x^i, x^j) = D_W F(W, x^i) D_W F(W, x^j)^\top$$

Note in particular that the kernel K depends on the parameterization W. This is not longer true for ANNs of infinite width: in [17] it is shown that K tends towards a deterministic kernel when the network width tends to infinity and W^0 is initialized randomly. Moreover, in this asymptotic limit, the kernel K stays constant when W evolves under gradient descent. In other words, training F amounts to perform a kernel regression for the kernel K which is a (strongly) convex optimization problem when K is a positive (definite) kernel. Constancy of the NTK is in fact a specific property of linear models:

Proposition 1 (Constant NTK = Linear model [20])**.** *The Neural Tangent Kernel of a differentiable function $F(W, x)$ is constant if and only if F is linear w.r.t. W.*

Therefore, convergence of GD in the "linear" / "constant NTK" regime is not necessarily good as it restricts the class of functions computed by the model. This regime can for example be obtained as of inappropriate scaling of the parameters [11]. The trained rescaled models then have poor generalization property even if they achieve a zero training error.

2.2 Polyak-Lojasiewicz inequalities

When dealing with overparameterized models, one cannot expect the loss $\mathcal{L}$ to be a convex function of the parameters. In this setting, a more careful analysis of the loss landscape is required in order to prove the convergence of GD towards a minimizer of the risk. A classical result from Lojasiewicz shows that if the gradient dynamic is bounded it converges towards a first-order critical point:

Theorem 1 (Lojasiewicz [22])**.** *If $\mathcal{L} : \mathbb{R}^m \to \mathbb{R}_+$ is analytic and the curve $W : t \geq 0 \mapsto W^t \in \mathbb{R}^m$ is bounded and a solution of the gradient flow eq.* (5)*, then it converges towards a (first-order) critical point of $\mathcal{L}$.*

However this result is not entirely satisfying: because of their large number of parameter, overparameterized models trained with GD should be able to fit the data exactly, that is to reach a global minimizer of risk 0. Indeed, for N target data points $y^i \in \mathbb{R}^{d'}$ the condition $\mathcal{L} = 0$ amounts to Nd' equations. Thus one should typically expect the set of global minimizers of the risk to be a sub-variety of co-dimension Nd' in the parameter space. In fact the loss landscape of overparameterized models typically possesses an infinite continuum of global minima and, althought not convex, is well-behaved in the neighbourhood of these minima [21]. One can show the loss satisfies a set of functional inequalities allowing to control its decrease rate along GD.

Definition 2 ((local) Polyak-Lojasiewicz property)**.** *Let $\mathcal{L} : \mathcal{H} \to \mathbb{R}_+$ be a differentiable function on a real Hilbert space $\mathcal{H}$. We say that $\mathcal{L}$ satisfies a (local) Polyak-Lojasiewicz (PL) property if there exist positive continuous functions $m, M : \mathbb{R}_+ \to \mathbb{R}_+^*$ s.t. for every $W \in \mathcal{H}$*

$$2m(\|W\|)L(v) \leq \|\nabla L(W)\|^2 \leq 2M(\|W\|)L(W). \tag{6}$$

A direct consequence of Definition 2 is that $\mathcal{L}$ does not admit any spurious local minima but only global minima. Combined with Theorem 1, this ensures convergence of GD towards a global minimizer if the dynamic is bounded. Indeed, if m and M are uniformly lower- and upper-bounded along the dynamic, then $\mathcal{L}$ decreases at a linear rate. Considering the gradient flow $\frac{\mathrm{d}}{\mathrm{d}t}W = -\nabla\mathcal{L}(W)$, eq. (6) implies:

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathcal{L}(W) = -\|\nabla\mathcal{L}\|^2 \leq -m\mathcal{L}(W).$$

However, in most cases, m and M are degenerate when $\|v\| \to +\infty$. Hence, if the dynamic is not bounded, $\mathcal{L}$ can decrease slower than at a linear rate or converge towards a strictly positive limit. It is thus in general not possible to conclude to an unconditional convergence of GD towards

a global minimizer of the risk. Nonetheless, PL inequalities are sufficient to prove convergence of GD towards a global minimizer when the problem is not too hard, i.e. when the loss at initialization is not too high.

Theorem 2 (Theorem 6 of [21]). *Let $\mathcal{L} : \mathcal{H} \to \mathbb{R}_+$ be a loss function satisfying a local PL property with local constants m and M. Let $W^0 \in \mathcal{H}$ and $R \geq 0$ be such that*

$$\frac{8M(\|W^0\| + R)}{m(\|W^0\| + R)^2} L(W^0) \leq R^2. \tag{7}$$

Then, if $\mathcal{L}$ is "smooth enough" within the ball $B(W^0, R)$, for a sufficiently small step size η, GD with initialization W^0 converges towards a global minimizer of $\mathcal{L}$ within a ball of radius R. More precisely, for every $k \geq 0$:

$$\mathcal{L}(W^k) \leq (1 - m(\|W^0\| + R)\eta)^k \mathcal{L}(W^0) \quad \textit{and} \quad \|W^k - W^0\| \leq R, \ \forall k \geq 0. \tag{8}$$

Theorem 2 is a local convergence result in which the condition in eq. (7) expresses a quantitative threshold between two kinds of behaviors:

(i) If $\mathcal{L}(v^0)$ is sufficiently small, the training dynamic converges towards a global minimizer. The limiting behaviour is when the l.h.s. of eq. (7) tends to 0. Because of a regularizing effect of GD (i.e. that $\|W^k - W^0\| \leq R$), the parameter stays in a ball of arbitrary small radius R all along the training dynamic. In this limit, we recover the aforementioned linear regime where the model is well approximated by its linearization around W^0. Moreover, the constants M and m play an important role in the result as the rate M/m^2 controls the size of the region where convergence of GD is ensured. This rate can in fact be shown to depend on the good conditioning of the NTK [20].

(ii) If $\mathcal{L}(W^0)$ is too large, the result says nothing about the convergence of the GD. However, it is still observed in practice that the training dynamic often converges towards a global minimizer. Explaining this phenomenon in a general setting remains a challenging open question !

3 Linear Networks

Linear Neural Networks are Neural Networks whose layers consists only of matrix vector products. Even though the class of functions that can be represented by such a model is restricted, it provides insight on how to deal with more general models.

Given a data matrix $X = (x^1|...|x^N) \in \mathbb{R}^{d \times N}$ and a set of matrix parameter $W = (W_l)_{1 \leq l \leq L}$ respectively of size $n_l \times n_{l-1}$, $n_0 = d$, the output of a Linear Neural Network with L layers parameterized by W is given by:

$$F(W) = W_L W_{L-1} ... W_1 X. \tag{9}$$

Analogously, a Linear Residual Network is defined as a ResNet whose residual terms consist of linear transformations. It can be considered as a Linear Neural Network whose parameters are defined around the identity matrix Id:

$$F(W) = \left(\mathrm{Id} + \frac{1}{L} W_L\right) ... \left(\mathrm{Id} + \frac{1}{L} W_1\right) X, \tag{10}$$

where the compatibility condition for intermediary dimensions is $n_0 = ... = n_L = d$. Note that we introduces $1/L$ as a rescaling factor in order to ensure the consistency with continuous models when $L \to \infty$. Indeed, passing to the limit of infinite depth, we can consider the following Linear-NODE model:

Definition 3 (Linear-NODE)**.** *Given a data matrix $X \in \mathbb{R}^{d\times N}$ we define for any $W \in L^2([0,1], \mathbb{R}^{d\times d})$ the output of the Linear-NODE model by:*

$$F(W) := U_1 X$$

with U the unique absolutely continuous solution to the forward problem*:*

$$\begin{cases} U_0 &= \mathrm{Id} \\ \frac{\mathrm{d}U_s}{\mathrm{d}s} &= W_s U_s. \end{cases} \tag{11}$$

W will therefore be called control parameter*.*

Remark 3. *On the space of matrices we consider the Frobenius norm, noted $\|.\|$. On the parameter space $L^2([0,1], \mathbb{R}^{d\times d})$ we consider the induced norm $\|W\| \overset{\text{def.}}{=} \int_0^1 \|W_s\| \mathrm{d}s$*

For the Linear-NODE model of Definition 3, it is natural to consider the square distance between the output $F(W)$ and the target data $Y = (y^1|...|y^N)$. The associated risk is defined as:

$$\mathcal{L}(W) = \frac{1}{2}\|F(W) - Y\|. \tag{12}$$

Moreover, solving problem eq. (ERM) amounts to perform a simple linear regression on the data matrices (X, Y). In this framework, the risk $\mathcal{L}(W)$ can be decomposed by orthogonal projection of Y onto $\mathrm{Span}(X)$:

$$\mathcal{L}(W) = \mathcal{L}^* + \frac{1}{2}\|F(W) - U^* X\|^2$$

where $\mathcal{L}^* = \inf_U \|UX - Y\|^2 = \|U^* X - Y\|^2$ is achieved at $U^* = YX^\dagger$ with $X^\dagger = X^\top (XX^\top)^{-1}$ the Moore-Penrose generalized inverse of X.

3.1 Local convergence and counter-example to global convergence

We consider training the Linear-NODE model with Gradient Descent on the risk $\mathcal{L}$ defined above. For this problem, one can show that the loss landscape is well-behaved as it satisfies a (local) Polyak-Lojasiewicz inequality. In particular, there are no spurious local minimizer of the risk.

Proposition 2 (PL inequality for Linear-NODEs)**.** *Consider the loss $\mathcal{L} : L^2([0,1], \mathbb{R}^{d\times d}) \to \mathbb{R}_+$ defined in eq.* (12)*. Then there exists strictly positive continuous functions $m, M : \mathbb{R}_+ \to \mathbb{R}_+$ such that for every control parameter W:*

$$m(\|W\|_{L^2}) \left[\mathcal{L}(W) - \mathcal{L}^*\right] \;\leq\; \|\nabla \mathcal{L}(W)\|^2 \;\leq\; M(\|W\|_{L^2}) \left[\mathcal{L}(W) - \mathcal{L}^*\right]. \tag{13}$$

As a consequence, GD is proved to converge towards a global minimizer W^* when initialized with a sufficiently low risk. This establishes the local convergent behaviour of GD. However, the possible degeneracy of the functions m and M prevents from controlling the dynamic when $W \to \infty$.

In order to illustrate the limitations of the above analysis, we provide a setting where the gradient dynamic can be calculated in closed form and exhibit a counter-example to convergence. More precisely we consider the setting where $X = \mathrm{Id}$, that is the data consist on the canonical basis of $\mathbb{R}^d$. In particular $\mathcal{L}^* = 0$.

Proposition 3. *Assume Y is a normal matrix and write $Y = A\,\mathrm{diag}((\lambda_i))A^*$ with $(\lambda_i)_{1\leq i\leq d}$ its (complex) eigenvalues and A some unitary matrix. For every time $t \geq 0$, we note $W^t \in L^2([0,1], \mathbb{R}^d)$ the gradient flow of the risk $\mathcal{L}$ initialiaed at $W^0 = 0$. Then at every time $t \geq 0$:*

$$W^t : s \in [0,1] \mapsto A\mathit{diag}(z_i^\tau)A^*,$$

with $(z_i)_i$ *solutions of:*

$$\begin{cases} \frac{d}{dt} z_i^t &= -(e^{z_i^t} - \lambda_i) e^{\bar{z}_i^t} \\ z_i^0 &= 0. \end{cases} \tag{14}$$

In particular W^t *is independent of* s*.*

As a consequence, it is not possible to reach the target $Y = -I_d$ starting from initialization $W^0 = 0$. Indeed, setting $\lambda_i = -1$, eq. (14) becomes:

$$\begin{cases} \frac{\mathrm{d}z}{\mathrm{d}t} &= -e^{\bar{z}}(e^z + 1) \\ z(t=0) &= 0. \end{cases}$$

In particular z stays real and the variable $x = e^{-z}$ satisfies the ODE $\dot{x} = 1 + 1/x \geq 1$, so that $x \to +\infty$ and $z = -\log(x) \to -\infty$. Taking the exponential, the map computed by F shrinks to 0: $F(W) = e^z I_d \to 0$. In fact, the dynamic on the eigenvalue $z \in \mathbb{C}$ corresponds to the gradient flow for the loss $\mathcal{L} : z \mapsto \frac{1}{2}|e^z - \lambda|^2$ for which a local PL property is verified:

$$|\nabla \mathcal{L}(z)|^2 = \left|e^{\bar{z}}(e^z - \lambda)\right|^2 = 2e^{2\Re(z)}\mathcal{L}(z).$$

Nonetheless, if $\lambda = re^{\imath\theta}$ and we write $z = -\log(\alpha) + \imath(\beta - \theta)$, the gradient flow dynamic reads:

$$\begin{cases} \frac{\mathrm{d}\alpha}{\mathrm{d}t} &= 1/\alpha - r\cos(\beta) \\ \frac{\mathrm{d}\beta}{\mathrm{d}t} &= -\frac{r}{\alpha}\sin\beta. \end{cases}$$

Thus the only critical points is $\beta = 0, \alpha = 1/r$, corresponding to $F(W) = Y$ and $L(z) = 0$, but if at initialization $\beta^0 = \pi$ then $\alpha \to +\infty$ and $L(z) \to r > 0$, corresponding to $F(W) \to 0$. However, the latter behavior is unstable. Indeed one can see the dynamic converges towards the first critical point as soon as $\beta^0 < \pi$ at initialization. This behavior is illustrated in fig. 2.

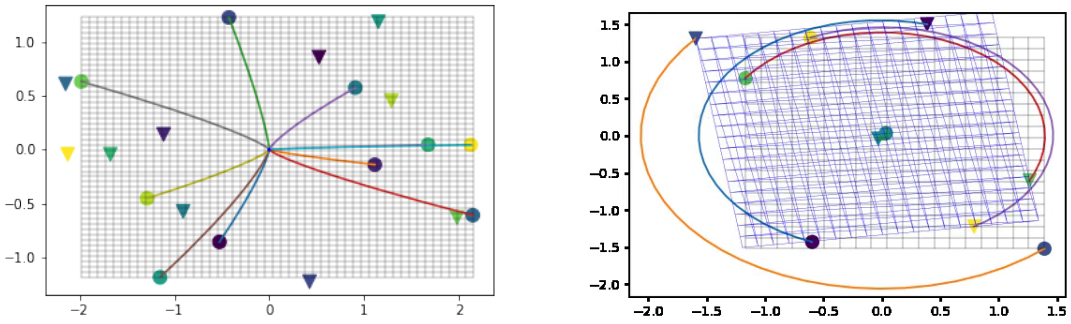

Figure 2: Linear-NODE trained to recover −Id in dimension 2. Circles are input data, triangles are corresponding targets and lines represent the flow implemented by the model. Left: $W^0 = 0$ and GD does not converge, the computed map shrinks onto zero. Right: W^0 is perturbed and GD converges, the computed map is −Id.

3.2 Global Convergence

The above discussion shed the light on the fact that even in the absence of spurious local minima, GD might fail to converge towards a global minimizer of the risk. This is due to the degeneracy of the Polyak-Lojasiewicz inequality when the parameter tends to infinity. As a consequence, GD might be attracted towards a direction in which $\nabla\mathcal{L}$ vanishes whereas $\mathcal{L} \geq c > 0$. However,

we observed this kind of behavior to be unstable to small perturbation of the initialization. We detail here a (almost sure) convergence result confirming this intuition.

Consider as model F a classical linear neural network with L layers given by:

$$F(W, X) = W_L W_{L-1} ... W_1 X,$$

where X is the data matrix and $W = (W_1, ..., W_L)$ are learned matrix parameters. Note that this model implements a linear mapping $\mathbb{R}^d \to \mathbb{R}^d$ corresponding to the composition of the W_i's. We note $W_{comp} \overset{\text{def.}}{=} W_L ... W_1$ this mapping. For a target matrix Y, consider training F on the square Frobenius distance with associated risk:

$$\mathcal{L}((W_1, ..., W_L)) = \frac{1}{2}\|F(W, X) - Y\|^2.$$

In particular this loss also corresponds to a loss on the mapping W_{comp} defined by:

$$\mathcal{L}_{comp}(W_{comp}) \overset{\text{def.}}{=} \frac{1}{2}\|W_{comp} X - Y\|^2$$

Then the gradient flow of $\mathcal{L}$ almost always converges towards a global minimizer of $\mathcal{L}_{comp}$:

Theorem 3 (Global convergence [4])**.** *Assume $XX^\top$ is a full rank. For $t \geq 0$, let $(W_1^t, ..., W_L^t)$ be the gradient flow of $\mathcal{L}$ and note W_{comp}^t the associated composition. Then for almost every initialization $(W_1^0, ..., W_L^0)$, W_{comp} converges towards a global minimizer of $\mathcal{L}_{comp}$ on the manifold of matrices of fixed rank k, with $k = \text{rank}(W_{comp}^0)$.*

Note that, in contrast with Theorem 2, no convergence rate can her be provided. This is consistent with the analysis of the previous section: there exist sets (of measure zero) where the gradient flow does not converge and convergence can be made arbitrarily slow by considering initializations arbitrarily close to these sets.

4 A promising setup: the RKHS-NODE model

Most often in the literature studying the training properties of ResNets (e.g. [12, 1, 15]), the considered residual transformations are of the form:

$$f : ((W, U), x) \mapsto W\sigma(Ux), \tag{15}$$

where $U \in \mathbb{R}^{q \times d}$ and $W \in \mathbb{R}^{d \times q}$ are trained matrix parameters and $\sigma : \mathbb{R} \to \mathbb{R}$ is a fixed non-linearity applied component-wise. Such residual transformation can approximate any continuous function of the variable x (see [5]). However, due to the composition with the non-linearity σ, such models are hard to analyse.

In contrast, we consider a simpler setting where the parameter U (called *feature matrix*) is not trained. Given a fixed *feature map* $\varphi : \mathbb{R}^d \to \mathbb{R}^q$, we consider residual functions of the form:

$$f : (W, x) \mapsto W\varphi(x), \tag{16}$$

where the matrices $W \in \mathbb{R}^{d \times q}$ are the trained parameters. As one can see, the main difference between eq. (15) and eq. (16) is linearity w.r.t. the parameters. This is however not a big restriction as, due to successive composition of the residuals the ResNet's output will still be highly non-linear w.r.t. parameters and input.

4.1 RKHS-NODE for regression

Here we show that the residuals of the form eq. (16) benefits from a particular metric structure. More precisely, the space of residual transformation is a *Reproducing Kernel Hilbert Space* of vector-valued function, that is a space a functions for which the evaluation is a continuous linear map [3]. This appens for example for Sobolev spaces of sufficient regularity.

Considering residuals of the form eq. (16), the space of residual transformations is:

$$V \overset{\text{def.}}{=} \{v : x \mapsto W\varphi(x) | W \in \mathbb{R}^{d\times q}.\}, \tag{17}$$

As is standard when implementing ANNs, the gradient with respect to the parameter W is computed in the L^2 sense of Frobenius metric on the set of matrices. Such an L^2 penalization induces a metric structure on V through the identification $v \in V \leftrightarrow W \in \mathbb{R}^{d\times q}$. For every $v, v' \in V$ on can define:

$$\langle v, v' \rangle_V \overset{\text{def.}}{=} \langle W, W' \rangle. \tag{18}$$

Provided with this metric structure, the evaluation is a continuous linear map on V as:

$$\forall x, p \in \mathbb{R}^d,\ \forall v \in V, \quad |\langle p, v(x)\rangle| = |\langle p, W\varphi(x)\rangle| \leq \|v\|_V \|p\| \|\varphi(x)\|.$$

Thus, V is a RKHS whose associated feature map is φ and whose associated kernel is $K : (x, x') \mapsto \langle \varphi(x), \varphi(x')\rangle \mathrm{Id}$. This identification motivates us to define a new type of ResNet / Neural ODE model whose residuals are not parameterized but directly instantiated as object in a RKHS. In contrast with the parametric setting, such a RKHS could be of infinite dimension.

Definition 4 (RKHS Neural ODE (RKHS-NODE))**.** *Let V be a RKHS of vector-fields over $\mathbb{R}^d$. Then for $v \in L^2([0,1], V)$ and a data input $x \in \mathbb{R}^d$, the RKHS-NODE model is defined as $F : (v, x) \mapsto x_1$ where $(x_s)_{s\in[0,1]}$ is the solution to the* forward problem*:*

$$\frac{\mathrm{d}x_s}{\mathrm{d}s} = v_s(x_s) \quad \textit{and} \quad x_0 = x. \tag{19}$$

Given some RKHS V we consider RKHS-NODE models in a regression task where we are provided with N data points $(x^i, y^i) \in \mathbb{R}^d \times \mathbb{R}^{d'}$ and consider training the model with the square loss on the output space. The associated risk writes for every parameter $v \in L^2([0,1], V)$:

$$\mathcal{L}(v) = \frac{1}{2} \sum_{i=1}^{N} \|F(v, x^i) - y^i\|^2.$$

Similarly to the linear case, the associated supervised learning problem fits into our framework of Polyak-Lojasiewicz analysis. In particular, local convergence of GD is ensured as soon as it is initialized sufficiently close from a global minimizer.

Proposition 4 (RKHS-NODE satisfy PL (informal))**.** *Let V be some "nice enough" RKHS. Let $\mathcal{L} : L^2([0,1], V) \to \mathbb{R}_+$ be the empirical risk associated to the RKHS-NODE model with the square loss. Then $\mathcal{L}$ satisfies the PL inequality of Definition 2.*

4.2 Groups of Diffeomorphims and right-invariant metric

Using a parametrization of the residuals terms with an abstract RKHS opens an interesting connection with the study of groups of diffeomorphisms, groups of deformation of the space that where also extensively studied in the framework of medical Image Registration [30, 6].

The solution $(x_t)_{t\in[0,1]}$ of the forward problem eq. (19) can be interpreted as the flow of the time-varying vector field v. General theory for the well-posedness of such transport equations has been broadly studied (see [2]). In our case, assuming a continuous embedding $V \hookrightarrow W^{1,\infty}$ is sufficient to ensure the forward problem is well-posed:

Proposition 5. *Let $v \in L^2([0,1], V)$. Then there exists a unique flow map $\Phi^v : [0,1] \times \mathbb{R}^d \to \mathbb{R}^d$ so that for any $x \in \mathbb{R}^d$, $\Phi_t^v(x)$ is the unique solution of the ODE:*

$$\begin{cases} \frac{\mathrm{d}}{\mathrm{d}s}\Phi_s^v(x) &= v_s(\Phi_s^v(x)) \\ \Phi_0^v(x) &= x. \end{cases} \tag{20}$$

Therefore, taking the point of view of the global deformation of the space, we can restate our model in a new way. Setting the control parameter $v \in L^2([0,1], V)$ the output of the RKHS-FlowResNet model F of Definition 4 is given, for any input $x \in \mathbb{R}^d$, by:

$$F(v, x) = \Phi_1^v(x),$$

with Φ^v provided by Proposition 5.

Considering the set of time one flows Φ_1^v for $v \in V$:

$$\mathcal{G}_V = \{\Phi_1^v \mid \partial_t \Phi_t^v = v_t \circ \Phi_t^v,\ \Phi_0^v = Id,\ v \in L^2([0,1], V)\},$$

A. Trouvé showed this is a group of $\mathcal{C}^1$-diffeomorphisms for the left-action by composition, V being its set of infinitesimal generators [27]. Furthermore, the scalar structure of V provides $\mathcal{G}_V$ with a local metric for which it is invariant by the right-action by composition:

$$\forall v, v' \in V,\ \forall \Phi \in \mathcal{G}_V, \quad \langle v \circ \Phi, v' \circ \Phi \rangle_{T_\Phi \mathcal{G}_V} = \langle v, v' \rangle_{T_{Id}\mathcal{G}_V} = \langle v, v' \rangle_V.$$

In particular, $\mathcal{G}_V$ is proven to be complete for this metric [27, 30]. In the specific case where $V = H^s$ is a Sobolev space, assuming $s > d/2 + 1$, the flow map belongs to the space of Sobolev diffeomorphisms $\mathcal{D}^s(\mathbb{R}^d)$, defined as:

$$\mathcal{D}^s(\mathbb{R}^d) = \{\Phi \in Id + H^s \mid \Phi \text{ bijective},\ \Phi^{-1} \in Id + H^s\} \tag{21}$$

Moreover $\mathcal{D}^s$ can be provided with a structure of Hilbert manifold, with $T_{\mathrm{Id}}\mathcal{D}^s = H^s$, and is geodesically when provided with the right-invariant metric [8]. That is, there always exists a minimizing geodesic between two diffeomorphisms in the same connected component. This property is non-trivial as $\mathcal{D}^s$ is of infinite dimension and the Hopf-Rinow theorem does not apply.

4.3 RKHS-NODE acting on measures

We considered in the previous sections a finite dimensional supervised learning setting where the action of V is defined on a finite number of data points N. Here we take interest in the limiting model when $N \to \infty$ and V acts on positive measures through a transport equation. The associated supervised learning problem is a density fitting problem which has applications for automatic synthetic data generation [18].

For what follows we consider $V = H^s$ for some $s > d/2 + 1$. Given as input data some probability measure μ on $\mathbb{R}^d$ and as control parameter $v \in L^2([0,1], V)$, we consider the model whose output is given by the push-forward action of the flow map Φ_1^v on μ_0:

$$F(v, \mu) \overset{\text{def.}}{=} (\Phi_1^v)_{\#}\mu \tag{22}$$

Anagolously to eq. (19), $F(v, \mu_0)$ can be equivalently defined as the solution at $s = 1$ to the *continuity equation*:

$$\begin{cases} \partial_s \mu_s + \mathrm{div}(\mu_s v_s) &= 0 \\ \mu_0 &= \mu. \end{cases} \tag{23}$$

Given some loss functional ℓ on the set of measures and some target probability measure μ^*, the risk associated to a parameter $v \in L^2([0,1], V)$ can then be defined as:

$$\mathcal{L}(v) = \ell(F(v, \mu), \mu^*) = \ell((\Phi_1^v)_{\#}\mu, \mu^*) \tag{24}$$

Note that here the training problem is of different nature as our model acts on infinite dimensional objects. As a consequence, we will in particular observe a deterioration on the convergence performances of GD.

Example on the flat torus For convenience, we choose to consider densities defined on the d-dimensional flat torus $\mathbb{T}^d = \mathbb{R}^d/\mathbb{Z}^d$. On the space of probability measures, we consider as loss functionals either the L^2 distance defines as:

$$\ell(\mu_1, \mu^*) \overset{\text{def.}}{=} \|\mu_1 - \mu^*\|_{L^2(\mathbb{T}^d)}.$$

Moreover, we assume that μ and μ^* both admit a density with respect to the Lebesgue measure on $\mathbb{T}^d$, noted ρ and ρ^* respectively, such that $\rho, \rho^* \in H^s(\mathbb{T}^d)$. We also assume that ρ is uniformly upper- ad lower- bounded by strictly positive constants. One can then show the existence of functional inequalities similar to PL for the model's risk:

Proposition 6. *Let $\mathcal{L}$ be the risk associated to ℓ on $\mathbb{T}^d$. Then there exists positive continuous functions $m, M : \mathbb{R}_+ \to \mathbb{R}_+^*$ such that for every control parameter $v \in L^2([0,1], H^s)$:*

$$m(\|v\|)\mathcal{L}(v)^{1-\nu} \leq \|\nabla\mathcal{L}(v)\| \leq M(\|v\|)\mathcal{L}(v)^{1+\lambda}, \tag{25}$$

for positive constants ν and λ. Moreover one can take $\lambda = 1 + 2/(s-2)$.

This last result ensures that the gradient dynamics converges towards a global minimizer when the dynamic is bounded, i.e. when the constants m and M can be uniformly bounded along GD. However note that the result in Proposition 6 is weaker than its counterpart in Definition 2. In particular it can no longer ensures the local convergence behavior of GD.

4.4 Long term behavior of Gradient Descent and diffusion phenomenons

In this last section we make a connection with the problem of convergence of GD for RKHS-NODE acting on measures and the long term behavior of some diffusion PDEs. Here we note $K*$ the convolution operator associated to the RKHS V with kernel K.

Consider, as previously, the model whose output is given by the pushforward action $(\Phi 6_1)_{\#}\mu$ for input measure μ and parameter $v \in L^2([0,1], V)$. Then given a target measure μ^* and some loss functional ℓ, the gradient with respect to v of the risk $\mathcal{L}$ is $\nabla\mathcal{L}(v) = -K * (\mu\nabla I)$ where the solution of the following *backward problem*:

$$\begin{cases} \partial_s I_s + \langle v_s, \nabla I_s \rangle &= 0 \\ I_1 &= -\partial\ell(\mu_1, \mu^*), \end{cases} \tag{26}$$

where (μ_s) is the solution to the forward problem eq. (23). Moreover taking the infinitesimal variation $v' = v + \varepsilon\delta v$ indices a variation $\mu' = \mu + \varepsilon\delta\mu$ where $\delta\mu$ is the solution to the transport equation with source term:

$$\begin{cases} \partial_s \delta\mu + \operatorname{div}(\delta\mu v) + \operatorname{div}(\mu\delta v) &= 0 \\ \delta\mu_0 &= 0. \end{cases} \tag{27}$$

Finally, choosing $v = 0$ and $\delta v = -\nabla\mathcal{L}(v)$ gives the variation at time $s = 1$:

$$\delta\mu_1 = -\operatorname{div}(\mu_1 \nabla\mathcal{L}(0)) = \operatorname{div}(\mu_1 K * (\mu_1 \nabla\ell(\mu_1, \mu^*))) \tag{28}$$

If eq. (28) might look unfamiliar at first sight, it might be useful to rewrite it in the simpler case where measures are defined on the flat torus $\mathbb{T}^d$ with target measures $\mu^* = \mathrm{d}x$ the Lebesgue measure and with the trivial kernel $K = \delta_0$. Then, assuming μ as density ρ and considering the L^2 loss, eq. (28) writes:

$$\partial_t \rho = \operatorname{div}(\rho^2 \nabla\rho) = \frac{1}{3}\Delta(\rho^3).$$

This last equation is known as the *Porous Medium Equation* for which convergence as been analyzed extensively [24]. The convergence argument relies on functional inequalities which can

be interpreted in terms of convexity of the energy in the Wasserstein space [25, 14, 9], as well as in terms of Polyak-Lojasiewicz inequalities [7].

Therefore we have showed that, at least formally, when initialized at $v^0 = 0$ the gradient flow acts at time $t = 0$ on the measure μ through the evolution PDE eq. (28). If some instances of such PDEs, such as the porous medium equation, are well understood it is not the case for the general form. Understanding the convergence and regularity properties of such PDEs could therefore be a first step towards a better understanding of the convergence properties of GD.

5 Conclusion

As we exposed in this note, the convergence of GD for the training ofoverparameterized neural network models is nowadays an active area of research, borrowing tools from various areas of mathematics. Although some common heuristics exist, such as Neural Tangent Kernel and local Polyak-Lojasiewicz analysis, they only manage to describe a regime of linear convergence when initialization is already to global minimizer (Theorem 2). On the other hand, long term convergence behaviors can be proven for simple models such as linear networks (Theorem 3). In an attempt to generalized these results to more expressive non-linear models we proposed ResNets models with a linear parameterization of the residuals. For this model the supervised learning problem has a nice interpretation as an optimization problem on groups of diffeomorphisms. Moreover we exhibited connections between the convergence properties of GD and the long term behavior of some evolution PDEs.

References

[1] Z. Allen-Zhu, Y. Li, and Z. Song, *A convergence theory for deep learning via over-parameterization*, in International Conference on Machine Learning, PMLR, 2019, pp. 242–252.

[2] L. Ambrosio, *Transport equation and cauchy problem for non-smooth vector fields*, in Calculus of variations and nonlinear partial differential equations, Springer, 2008, pp. 1–41.

[3] N. Aronszajn, *Theory of reproducing kernels*, Transactions of the American mathematical society, 68 (1950), pp. 337–404.

[4] B. Bah, H. Rauhut, U. Terstiege, and M. Westdickenberg, *Learning deep linear neural networks: Riemannian gradient flows and convergence to global minimizers*, Information and Inference: A Journal of the IMA, 11 (2022), pp. 307–353.

[5] A. Barron, *Universal approximation bounds for superpositions of a sigmoidal function*, IEEE Transactions on Information Theory, 39 (1993), pp. 930–945.

[6] M. F. Beg, M. I. Miller, A. Trouvé, and L. Younes, *Computing large deformation metric mappings via geodesic flows of diffeomorphisms*, International journal of computer vision, 61 (2005), pp. 139–157.

[7] A. Blanchet and J. Bolte, *A family of functional inequalities: Łojasiewicz inequalities and displacement convex functions*, Journal of Functional Analysis, 275 (2018), pp. 1650–1673.

[8] M. Bruveris and F.-X. Vialard, *On completeness of groups of diffeomorphisms*, Journal of the European Mathematical Society, 19 (2017), pp. 1507–1544.

[9] J. A. Carrillo, R. J. McCann, and C. Villani, *Kinetic equilibration rates for granular media and related equations: entropy dissipation and mass transportation estimates*, Revista Matematica Iberoamericana, 19 (2003), pp. 971–1018.

[10] R. T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. Duvenaud, *Neural ordinary differential equations*, Advances in Neural Information Processing Systems, (2018).

[11] L. Chizat, E. Oyallon, and F. Bach, *On lazy training in differentiable programming*, in NeurIPS 2019-33rd Conference on Neural Information Processing Systems, 2019, pp. 2937–2947.

[12] S. Du, J. Lee, H. Li, L. Wang, and X. Zhai, *Gradient descent finds global minima of deep neural networks*, in International Conference on Machine Learning, PMLR, 2019, pp. 1675–1685.

[13] W. E, J. Han, and Q. Li, *A mean-field optimal control formulation of deep learning*, Research in the Mathematical Sciences, 6 (2019), p. 10.

[14] W. Gangbo and R. J. McCann, *The geometry of optimal transportation*, Acta Mathematica, 177 (1996), pp. 113–161.

[15] M. Hardt and T. Ma, *Identity Matters in Deep Learning*, arXiv:1611.04231 [cs, stat], (2018). arXiv: 1611.04231.

[16] K. He, X. Zhang, S. Ren, and J. Sun, *Deep residual learning for image recognition*, in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.

[17] A. Jacot, F. Gabriel, and C. Hongler, *Neural tangent kernel: convergence and generalization in neural networks (invited paper)*, in Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing, Virtual Italy, June 2021, ACM, pp. 6–6.

[18] I. Kobyzev, S. Prince, and M. Brubaker, *Normalizing Flows: An Introduction and Review of Current Methods*, IEEE Transactions on Pattern Analysis and Machine Intelligence, (2020), pp. 1–1.

[19] J. Lee, L. Xiao, S. Schoenholz, Y. Bahri, R. Novak, J. Sohl-Dickstein, and J. Pennington, *Wide neural networks of any depth evolve as linear models under gradient descent*, Advances in neural information processing systems, 32 (2019), pp. 8572–8583.

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

[21] ——, *Loss landscapes and optimization in over-parameterized non-linear systems and neural networks*, arXiv:2003.00307 [cs, math, stat], (2021). arXiv: 2003.00307.

[22] S. Lojasiewicz, *Sur les trajectoires du gradient d'une fonction analytique*, Seminari di geometria, 1983 (1982), pp. 115–117.

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

[24] F. OTTO, *The geometry of dissipative evolution equations: the porous medium equation*, Comm. Partial Differential Equations, 26 (2001), pp. 101–174.

[25] F. Otto and C. Villani, *Generalization of an inequality by talagrand and links with the logarithmic sobolev inequality*, Journal of Functional Analysis, 173 (2000), pp. 361–400.

[26] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, *Going deeper with convolutions*, in Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 1–9.

[27] A. Trouvé, *Diffeomorphisms groups and pattern matching in image analysis*, International journal of computer vision, 28 (1998), pp. 213–221.

[28] F.-X. Vialard, R. Kwitt, S. Wei, and M. Niethammer, *A shooting formulation of deep learning*, Advances in Neural Information Processing Systems, 33 (2020).

[29] E. Weinan, *A proposal on machine learning via dynamical systems*, Communications in Mathematics and Statistics, 5 (2017), pp. 1–11.

[30] L. Younes, *Shapes and Diffeomorphisms*, vol. 171 of Applied Mathematical Sciences, Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.