Gradient Descent on Infinitely Wide Neural Networks: Global Convergence and Generalization

Francis Bach, Lénaïc Chizat

Gradient Descent on Infinitely Wide Neural Networks: Global Convergence and Generalization

Francis Bach
Inria & Ecole Normale Supérieure
PSL Research University
francis.bach@inria.fr

Lénaïc Chizat
Ecole Polytechnique Fédérale de Lausanne
lenaic.chizat@epfl.ch

October 15, 2021

Abstract

Many supervised machine learning methods are naturally cast as optimization problems. For prediction models which are linear in their parameters, this often leads to convex problems for which many mathematical guarantees exist. Models which are non-linear in their parameters such as neural networks lead to non-convex optimization problems for which guarantees are harder to obtain. In this review paper, we consider two-layer neural networks with homogeneous activation functions where the number of hidden neurons tends to infinity, and show how qualitative convergence guarantees may be derived.

1 Introduction

In the past twenty years, data in all their forms have played an increasing role: in personal lives, with various forms of multimedia and social networks, in the economic sector where most industries monitor all of their processes and aim at making data-driven decisions, and in sciences, where data-based research is having more and more impact, both in fields which are traditionally data-driven such as medecine and biology, but also in humanities.

This proliferation of data leads to a need for automatic processing, with striking recent progress in some perception tasks where humans excel, such as image recognition, or natural language processing. These advances in artificial intelligence were fueled by the combination of three factors: (1) massive data to learn from, such as millions of labeled images, (2) increased computing resources to treat this data, and (3) continued scientific progress in algorithms.

Machine learning is one of the scientific disciplines that have made this progress possible, by blending statistics and optimization to design algorithms with theoretical generalization guarantees. The goal of this review paper that will be published in the Proceedings of the 2022 International Congress of Mathematicians is to highlight our recent progress from already published works [8, 9], and to present a few open mathematical problems.

2 Supervised learning

In this paper, we will focus on the supervised machine learning problem, where we are being given n pairs of observations $(x_i, y_i) \in \mathcal{X} \times \mathcal{Y}$, $i = 1, \dots, n$, for example images ($\mathcal{X}$ is then the set of all possible images), with a set of labels ($\mathcal{Y}$ is then a finite set, which we will assume to be a subset of $\mathbb{R}$ for simplicity). The goal is to be able to predict a new output $y \in \mathcal{Y}$, given a previously unobserved input $x \in \mathcal{X}$.

Following the traditional statistical *M-estimation* framework [45], this can be performed by considering prediction functions $x \mapsto h(x, \theta) \in \mathbb{R}$, parameterized by $\theta \in \mathbb{R}^d$. The vector θ is then estimated through regularized empirical risk minimization, that is by solving

$$\min_{\theta \in \mathbb{R}^d} \quad \frac{1}{n} \sum_{i=1}^{n} \ell\big(y_i, h(x_i, \theta)\big) + \lambda \Omega(\theta), \tag{2.1}$$

where $\ell : \mathcal{Y} \times \mathbb{R} \to \mathbb{R}$ is a loss function, and $\Omega : \mathbb{R}^d \to \mathbb{R}$ is a regularization term that avoids overfitting (that is learning a carbon copy of the observed data that does not generalize well to unseen data).

Typical loss functions are the square loss $\ell\big(y_i, h(x_i, \theta)\big) = \frac{1}{2}\big(y_i - h(x_i, \theta)\big)^2$ for regression problems, and the logistic loss $\ell\big(y_i, h(x_i, \theta)\big) = \log\big(1 + \exp(-y_i h(x_i, \theta))\big)$ for binary classification where $\mathcal{Y} = \{-1, 1\}$. In this paper, we will always assume that the loss function is continuously twice differentiable and convex with respect to the second variable. This applies to a wide variety of output spaces beyond regression and binary classification (see [36] and references therein).

When the predictor depends linearly in the parameters, typical regularizers are the squared Euclidean norm $\Omega(\theta) = \frac{1}{2}\|\theta\|_2^2$ or the ℓ_1-norm $\Omega(\theta) = \|\theta\|_1$, that both lead to improved generalization performance, with the ℓ_1-norm providing additional variable selection benefits [14].

2.1 Statistics and optimization

The optimization problem in Eq. (2.1) leads naturally to two sets of questions, which are often treated separately. Given that some minimizer $\hat{\theta}$ is obtained (no matter how), how does the corresponding prediction function generalize to unseen data? This is a statistical question that requires assumptions on the link between the observed data (usually called the "training data"), and the unseen data (usually called the "testing data"). It is typical to assume that the training and testing data are sampled independently and identically from the same fixed distribution. Then a series of theoretical guarantees applies, based on various probabilistic concentration inequalities (see, e.g., [31]).

The second question is how to obtain an approximate minimizer $\hat{\theta}$, which is an optimization problem, regardless on the relevance of $\hat{\theta}$ on unseen data (see, e.g., [6]). For high-dimensional problems where d is large (up to millions or billions), classical gradient-based algorithms are preferred because of their simplicity, efficiency, robustness and favorable convergence properties. The most classical one is gradient descent, which is an iterative algorithm with iteration:

$$\theta_k = \theta_{k-1} - \gamma \nabla \mathcal{R}(\theta_{k-1}),$$

where $\mathcal{R}(\theta) = \frac{1}{n}\sum_{i=1}^{n} \ell\big(y_i, h(x_i, \theta)\big) + \lambda\Omega(\theta)$ is the objective function in Eq. (2.1), and $\gamma > 0$ the step-size.

In this paper, where we aim at tackling high-dimensional problems, we will often consider the two problems of optimization and statistical estimation jointly.

2.2 Linear predictors and convex optimization

In many applications, a prediction function which is linear in the parameter θ is sufficient for good predictive performance, that is, we can write

$$h(x, \theta) = \theta^\top \Phi(x)$$

for some function $\Phi : \mathcal{X} \to \mathbb{R}^d$, which is often called a "feature function". For simplicity we have assumed finite-dimensional features, but infinite-dimensional features can also be considered, with a specific computational argument to allow finite-dimensional computations through reproducing kernel Hilbert spaces (see, e.g., [40] and references therein).

Given a convex loss function, the optimization problem is convex and gradient descent on the objective function, together with its stochastic extensions, has led to a number of efficient algorithms with strong generalization guarantees of convergence towards the *global* optimum of the objective function [6]. For example, for the square loss or the logistic loss, if the feature function is bounded in ℓ_2-norm by R for all observations, and for the squared Euclidean norm $\Omega(\theta) = \frac{1}{2}\|\theta\|_2^2$, bounds on the number of iterations to reach a certain precision ε (difference between the candidate function value $\mathcal{R}(\theta)$ and the minimal value) can be obtained:

- For gradient descent, $\frac{R^2}{\lambda} \log \frac{1}{\varepsilon}$ iterations are needed, but each iteration has a running time complexity of $O(nd)$, because the d-dimensional gradients of the n functions $\theta \mapsto \ell\big(y_i, h(x_i, \theta)\big)$, $i = 1, \dots, n$, are needed.
- For stochastic gradient descent, with iteration $\theta_k = \theta_{k-1} - \gamma \nabla \ell\big(y_{i(k)}, h(x_{i(k)}, \theta_{k-1})\big)$, with $i(k) \in \{1, \dots, n\}$ taken uniformly at random, the number of iterations is at most $\frac{R^2}{\lambda}\frac{1}{\varepsilon}$. We lose the logarithmic dependence, but each iteration has complexity $O(d)$, which can be a substantial gain when n is large.
- More recent algorithms based on variance reduction can achieve an overall complexity proportional to $\big(n + \frac{R^2}{\lambda}\big) \log \frac{1}{\varepsilon}$, thus with an exponential convergence rate at low iteration cost (see [16] and references therein).

In summary, for linear models, algorithms come with strong performance guarantees that reasonably match their empirical behavior. As shown below, non-linear models exhibit more difficulties.

2.3 Neural networks and non-convex optimization

In many other application areas, in particular in multimedia processing, linear predictors have been superseded by non-linear predictors, with neural networks being the most classical example (see [15]). A vanilla neural network is a prediction function of the form

$$h(x, \theta) = \theta_s^\top \sigma(\theta_{s-1}^\top \sigma(\cdots \theta_2^\top \sigma(\theta_1^\top x)),$$

where the function $\sigma : \mathbb{R} \to \mathbb{R}$ is taken component-wise, with the classical examples being the sigmoid function $\sigma(t) = (1 + \exp(-t))^{-1}$ and the "rectified linear unit" (ReLU), $\sigma(t) = t_+ =$

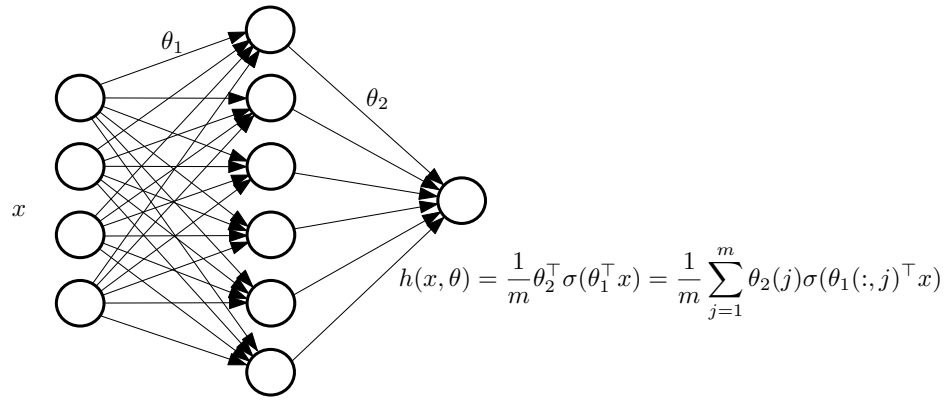


Figure 1: Neural network with a single hidden layer, with an input weight matrix $\theta_1 \in \mathbb{R}^{d\times m}$ and a output weight vector $\theta_2 \in \mathbb{R}^m$.

$\max\{t, 0\}$. The matrices $\theta_1, \dots, \theta_s$ are called weight matrices. The simplest non-linear predictor is for $s = 2$, and will be the main subject of study in this paper. See Figure 1 for an illustration.

The main difficulty is that now the optimization problem in Eq. (2.1) is not convex anymore, and gradient descent can converge to stationary points that are not global minima. Theoretical guarantees can be obtained regarding the decay of the norm of the gradient of the objective function, or convergence to a local minimizer may be ensured [25, 21], but this does not exclude bad local minima, and global quantitative convergence guarantees can only be obtained with exponential dependence in dimension for the class of (potentially non-convex) functions of a given regularity [33].

An extra difficulty is related to the number of hidden neurons, also referred to as the *width* of the network (equal to the size of θ_2 when $s = 2$), which is often very large in practice, which poses both statistical and optimization issues. We will see that this is precisely this overparameterization that allows to obtain qualitative global convergence guarantees.

3 Mean field limit of overparameterized one-hidden layer neural networks

We now tackle the study of neural networks with one infinitely wide hidden layer. They are also referred to as (wide) two-layer neural networks, because they have two layers of weights. We first rescale the prediction function by $1/m$ (which can be obtained by rescaling θ_2 by $1/m$), and express it explicitly as en empirical average, as

$$h(x,\theta) = \frac{1}{m}\theta_2^\top \sigma(x^\top \theta_1) = \frac{1}{m}\sum_{j=1}^{m} \theta_2(j) \cdot \sigma\big[x^\top \theta_1(\cdot, j)\big],$$

where $\theta_2(j) \in \mathbb{R}$ is the output weight associated to neuron j, and $\theta_1(\cdot, j) \in \mathbb{R}^d$ the corresponding vector of input weights. The key observation is that the prediction function $x \mapsto h(x,\theta)$ is the average of m prediction functions $x \mapsto \theta_2(j) \cdot \sigma\big[\theta_1(\cdot, j)^\top x\big]$, for $j = 1, \dots, m$, with *no sharing of the parameters* (which is not true if extra layers of hidden neurons are added).

In order to highlight this parameter separability, we define

$$w_j = \big[\theta_2(j), \theta_1(\cdot, j)\big] \in \mathbb{R}^{d+1}$$

the set of weights associated to the hidden neuron j, and we define

$$\Psi(w) : x \mapsto w(1) \cdot \sigma\big[x^\top w(2, \dots, d+1)\big],$$

so that the prediction function $x \mapsto h(\cdot, w_1, \dots, w_m)$, parameterized by $w_1, \dots, w_m$, is now

$$h(\cdot, w_1, \dots, w_m) = \frac{1}{m} \sum_{j=1}^{m} \Psi(w_j). \tag{3.1}$$

The empirical risk is of the form

$$R(h) = \mathbb{E}\big[\ell(y, h(x))\big],$$

which is convex in h for convex loss functions (even for neural networks), but typically non convex in w. Note that the resulting problem of minimizing a convex function $R(h)$ for $h = \frac{1}{m}\sum_{j=1}^{m} \Psi(w_j)$ applies beyond neural networks, for example, for sparse deconvolution [7].

3.1 Reformulation with probability measures

We now define by $\mathcal{P}(\mathcal{W})$ the set of probability measures on $\mathcal{W} = \mathbb{R}^{d+1}$. We can rewrite Eq. (3.1) as

$$h = \int_{\mathcal{W}} \Psi(w) d\mu(w),$$

with $\mu = \frac{1}{m}\sum_{j=1}^{m} \delta_{w_j}$ an average of Dirac measures at each $w_1, \dots, w_m$. Following a physics analogy, we will refer to each w_j as a *particle*. When the number m of particles grow, then the empirical measure $\frac{1}{m}\sum_{j=1}^{m} \delta_{w_j}$ may converge in distribution to a probability measure with a density, often referred to as a *mean field* limit. Our main reformulation will thus be to consider an optimization problem over probability measures.

The optimization problem we are faced with is equivalent to

$$\inf_{\mu \in \mathcal{P}(\mathcal{W})} R\Big(\int_{\mathcal{W}} \Psi(w) d\mu(w) \Big), \tag{3.2}$$

with the constraint that μ is an average of m Dirac measures. In this paper, following a long line of work in statistics and signal processing [5, 23], we consider the optimization problem *without this constraint*, and relate optimization algorithms for finite but large m (thus acting on $W = (w_1, \dots, w_m)$ in $\mathcal{W}^m$) to a well-defined algorithm in $\mathcal{P}(\mathcal{W})$.

Note that we now have a convex optimization problem, with a convex objective in μ over a convex set (all probability measures). However, it is still an infinite-dimensional space that requires dedicated finite-dimensional algorithms. In this paper we focus on gradient descent on w, which corresponds to standard practice in neural networks (e.g., back-propagation). For algorithms based on classical convex optimization algorithms such as the Frank-Wolfe algorithm, see [4].

3.2 From gradient descent to gradient flow

Our general goal is to study the gradient descent recursion on $W = (w_1, \dots, w_m) \in \mathcal{W}^m$, defined as

$$W_k = W_{k-1} - \gamma m \nabla G(W_{k-1}), \tag{3.3}$$

with

$$G(W) = R\big(h(\cdot, w_1, \dots, w_m)\big) = R\Big(\frac{1}{m}\sum_{j=1}^{m}\Psi(w_j)\Big).$$

In the context of neural networks, this is exactly the back-propagation algorithm. We include the factor m in the step-size to obtain a well-defined limit when m tends to infinity (see Section 3.3).

For convenience in the analysis, we look at the limit when the step-size γ goes to zero. If we consider a function $V : \mathbb{R} \to \mathcal{W}^m$, with values $V(k\gamma) = W_k$ at $t = k\gamma$, and we interpolate linearly between these points, then, we obtain exactly the standard Euler discretization of the ordinary differential equation (ODE) [44]:

$$\dot{V} = -m\nabla G(V). \tag{3.4}$$

This gradient flow will be our main focus in this paper. As highlighted above, and with extra regularity assumptions, it is the limit of the gradient recursion in Eq. (3.3) for vanishing step-sizes γ. Moreover, under appropriate conditions, stochastic gradient descent, where we only observe an unbiased noisy version of the gradient, also leads in the limit $\gamma \to 0$ to the same ODE [24]. This allows to apply our results to probability distributions of the data (x, y) which are not the observed empirical distribution, but the unseen test distribution, where the stochastic gradients come from the gradient of the loss from a single observation.

Three questions now emerge:

1. What is the limit (if any) of the gradient flow in Eq. (3.4) when the number of particles m gets large?
2. Where can the gradient flow converge to?
3. Can we ensure a good generalization performance when the number of parameters grows unbounded?

In this paper, we will focus primarily in the next sections on the first two questions, and tackle the third question in Section 5.

3.3 Wasserstein gradient flow

Above, we have described a general framework where we want to minimize a function F defined on probability measures:

$$F(\mu) = R\Big(\int_{\mathcal{W}}\Psi(w)d\mu(w)\Big), \tag{3.5}$$

with an algorithm minimizing $G(w_1, \dots, w_m) = R\Big(\frac{1}{m}\sum_{j=1}^{m}\Psi(w_j)\Big)$ through the gradient flow $\dot{W} = -m\nabla G(W)$, with $W = (w_1, \dots, w_m)$.

As shown in a series of works concerned with the infinite width limit of two-layer neural networks [35, 8, 30, 41, 38], this converges to a well-defined mathematical object called a Wasserstein gradient flow [2]. This is a gradient flow derived from the Wasserstein metric on the set of probability measures, which is defined as [39],

$$W_2(\mu, \nu)^2 = \inf_{\gamma \in \Pi(\mu,\nu)} \int \|v - w\|_2^2 d\gamma(v, w),$$

where $\Pi(\mu, \nu)$ is the set of probability measures on $\mathcal{W} \times \mathcal{W}$ with marginals μ and ν. In a nutshell, the gradient flow is defined as the limit when γ tends to zero of the extension of the following discrete time dynamics:

$$\mu(t+\gamma) = \inf_{\nu \in \mathcal{P}(\mathcal{W})} F(\nu) + \frac{1}{2\gamma} W_2(\mu(t), \nu)^2.$$

When applying such a definition in a Euclidean space with the Euclidean metric, we recover the usual gradient flow $\dot{\mu} = -\nabla F(\mu)$, but here with the Wasserstein metric, this defines a specific flow on the set of measures. When the initial measure is a weighted sum of Diracs, this is exactly asymptotically (when $\gamma \to 0$) equivalent to backpropagation. When initialized with an arbitrary probability measure, we obtain a partial differential equation (PDE), satisfied in the sense of distributions. Moreover, when the sum of Diracs converges in distribution to some measure, the flow converges to the solution of the PDE. More precisely, assuming $\Psi : \mathbb{R}^{d+1} \to \mathcal{F}$ a Hilbert space, and $\nabla R(h) \in \mathcal{F}$ the gradient of R, we consider the *mean potential*

$$J(w|\mu) = \Big\langle \Psi(w), \nabla R\Big(\int_{\mathcal{W}} \Psi(v) d\mu(v) \Big) \Big\rangle. \tag{3.6}$$

The PDE is then the classical continuity equation:

$$\partial_t \mu_t(w) = \operatorname{div}(\mu_t(w) \nabla J(w|\mu_t)), \tag{3.7}$$

which is understood in the sense of distributions. The following result formalizes this behavior (see [8] for details and a more general statement).

Theorem 1 *Assume that $R : \mathcal{F} \to [0, +\infty[$ and $\Psi : \mathcal{W} = \mathbb{R}^{d+1} \to \mathcal{F}$ are (Fréchet) differentiable with Lipschitz differentials, and that R is Lipschitz on its sublevel sets. Consider a sequence of initial weights $(w_j(0))_{j \geq 1}$ contained in a compact subset of $\mathcal{W}$ and let $\mu_{t,m} \coloneqq \frac{1}{m} \sum_{j=1}^{m} w_j(t)$ where $(w_1(t), \dots, w_m(t))$ solves the ODE (3.4). If $\mu_{0,m}$ weakly converges to some $\mu_0 \in \mathcal{P}(\mathcal{W})$ then $\mu_{t,m}$ weakly converges to μ_t where $(\mu_t)_{t \geq 0}$ is the unique weakly continuous solution to (3.7) initialized with μ_0.*

In the following section, we will study the solution of this PDE (i.e., the Wasserstein gradient flow), interpreting it as the limit of the gradient flow in Eq. (3.4), when the number of particles m tend to infinity.

4 Global convergence

We consider the Wasserstein gradient flow defined above, which leads to the PDE in Eq. (3.7). Our goal is to understand when we can expect that when $t \to \infty$, μ_t converges to a global minimum of F defined in Eq. (3.5). Obtaining a global convergence result is not out of the question because F is a convex functional defined on the convex set of probability measures. However, it is non trivial because with our choice of the Wasserstein geometry on measures, which allows an approximation through particles, the flow has some stationary points which are not the global optimum (see examples in Section 4.4).

We start with an informal general result without technical assumptions before stating a formal simplified result.

4.1 Informal result

In order to avoid too many technicalities, we first consider an informal theorem in this paper and refer to [8] for a detailed set of technical assumptions (in particular smoothness assumptions). This leads to the informal theorem:

Theorem 2 (Informal) *If the support of the initial distribution includes all directions in $\mathbb{R}^{d+1}$, and if the function Ψ is positively 2-homogeneous then if the Wasserstein gradient flow weakly converges to a distribution, it can only be to a global optimum of F.*

In [8] another version of this result that allows for *partial* homogeneity (e.g., with respect to a subset of variables) of degree 1 is proven, at the cost of a more technical assumption on the initialization. For neural networks, we have $\Psi(w_j)(x) = m\theta_2(j) \cdot \sigma\big[\theta_1(\cdot, j)^\top x\big]$, and this more general version applies. For the classical ReLU activation function $u \mapsto \max\{0, u\}$, we get a positively 2-homogeneous function, as required in the previous statement. A simple way to spread all directions is to initialize neural network weights from Gaussian distributions, which is standard in applications [15].

From qualitative to quantitative results? Our result states that for infinitely many particles, we can only converge to a global optimum (note that we cannot show that the flow always converges). However, it is only a qualitative result in comparison with what is known for convex optimization problems in Section 2.2:

- This is only for $m = +\infty$, and we cannot provide an estimation of the number of particles needed to approximate the mean field regime that is not exponential in t (see such results e.g. in [28]).
- We cannot provide an estimation of the performance as the function of time, that would provide an upper bound on the running time complexity.

Moreover, our result does not apply beyond a single hidden layer, and understanding the non-linear infinite width limits for deeper networks is an important research area [34, 3, 13, 19, 42, 12, 48].

From informal to formal results. Beyond the lack of quantitative guarantees, obtaining a formal result requires regularity and compactness assumptions which are not satisfied for the classical ReLU activation function $u \mapsto \max\{0, u\}$, which is not differentiable at zero (a similar result can be obtained in this case but under stronger assumptions on the data distribution and the initialization [47, 9]). In the next section, we will consider a simplified formal result, with a detailed proof.

4.2 Simplified formal result

In order to state a precise result, we will cast the flow on probability measures on $\mathcal{W} = \mathbb{R}^{d+1}$ to a flow on measures on the unit sphere

$$\mathcal{S}^d = \{w \in \mathbb{R}^{d+1},\ \|w\|_2 = 1\}.$$

This is possible when the function Ψ is positively 2-homogeneous on $\mathcal{W} = \mathbb{R}^{d+1}$, that is, such that $\Psi(\lambda w) = \lambda^2 \Psi(w)$ for $\lambda > 0$. We can use homogeneity by reparameterizing each particle w_j in polar coordinates as

$$w_j = r_j \eta_j, \text{ with } r_j \in \mathbb{R} \text{ and } \eta_j \in \mathcal{S}^d.$$

Using homogenetity, we have a prediction function:

$$h = \frac{1}{m} \sum_{j=1}^{m} \Psi(w_j) = \frac{1}{m} \sum_{j=1}^{m} r_j^2 \Psi(\eta_j).$$

Moreover the function J defined in Eq. (3.6) is also 2-homogeneous, and its gradient then 1-homogeneous. The flow from Eq. (3.4), can be written

$$\dot{w}_j = -\nabla J(w_j|\mu) \text{ with } \mu = \frac{1}{m} \sum_{i=1}^{m} \delta_{w_j}.$$

A short calculation shows that the flow

$$\left\{ \begin{array}{ll} \dot{r}_j & = -2 r_j J(\eta_j|\nu) \\ \dot{\eta}_j & = -(I - \eta_j \eta_j^\top) \nabla J(\eta_j|\nu) \end{array} \right. \quad \text{with } \nu = \frac{1}{m} \sum_{i=1}^{m} r_j^2 \delta_{\eta_j}, \tag{4.1}$$

leads to exactly the same dynamics. Indeed, by homogeneity of Ψ, the two definitions of μ and ν (through the w_j's, or the η_j's and r_j's) lead to the same functions $J(\cdot|\mu)$ and $J(\cdot|\nu)$, and we get

$$\begin{array}{rcl} \dot{w}_j & = & \dot{r}_j \eta_j + r_j \dot{\eta}_j = -2 r_j J(\eta_j|\nu) \eta_j - r_j (I - \eta_j \eta_j^\top) \nabla J(\eta_j|\nu) \\ & = & -r_j \nabla J(\eta_j|\nu) - r_j \big[2 J(\eta_j|\nu) - \eta_j^\top \nabla J(\eta_j|\nu) \big] \eta_j \\ & = & -\nabla J(w_j|\mu), \end{array}$$

because $w \mapsto \nabla J(w|\nu)$ is 1-homogeneous and by the Euler identity for the 2-homogeneous function $w \mapsto J(w|\nu) = J(w|\mu)$.

Moreover, the flow defined in Eq. (4.1) is such that η_j remains on the sphere $\mathcal{S}^d$. We will study this flow under the assumption that the function Ψ is sufficient regular, which excludes ReLU neural networks, but makes the proof easier (see more details in [7]).

We first derive a PDE analogous to Eq. (3.7). We consider a smooth test function $f : \mathcal{S}^d \to \mathbb{R}$, and the quantity

$$a = \int_{\mathcal{S}^d} f(\eta) d\nu(\eta) = \frac{1}{m} \sum_{j=1}^{m} r_j^2 f(\eta_j).$$

We have

$$\begin{array}{rcl} \dot{a} & = & \displaystyle \frac{1}{m} \sum_{j=1}^{m} 2 r_j \dot{r}_j f(\eta_j) + \frac{1}{m} \sum_{j=1}^{m} r_j^2 \nabla f(\eta_j)^\top \dot{\eta}_j \\ & = & \displaystyle -\frac{1}{m} \sum_{j=1}^{m} 4 r_j^2 J(\eta_j|\nu) f(\eta_j) - \frac{1}{m} \sum_{j=1}^{m} r_j^2 \nabla f(\eta_j)^\top (I - \eta_j \eta_j^\top) \nabla J(\eta_j|\nu) \\ & = & \displaystyle -4 \int_{\mathcal{S}^d} f(\eta) J(\eta|\nu) d\nu(\eta) - \int_{\mathcal{S}^d} \nabla f(\eta|\nu)^\top (I - \eta\eta^\top) \nabla J(\eta|\nu) d\nu(\eta). \end{array} \tag{4.2}$$

This exactly shows that we have the PDE for the density ν_t at time t

$$\partial_t \nu_t(\eta) = -4J(\eta|\nu_t) + \mathrm{div}(\nu_t(\eta)\nabla J(\eta|\nu_t)) \tag{4.3}$$

satisfied in the sense of distributions. We can now state our main result.

Theorem 3 *Assume the function $\Psi : \mathcal{S}^d \to \mathcal{F}$ is d-times continuously differentiable. Assume ν_0 is a nonnegative measure on the sphere $\mathcal{S}^d$ with finite mass and full support. Then the flow defined in Eq. (4.3) is well defined for all $t \geqslant 0$. Moreover, if ν_t converges weakly to some limit ν_∞, then ν_∞ is a global minimum of the function $\nu \mapsto F(\nu) = R\Big(\int_{\mathcal{S}^d} \Psi(\eta) d\nu(\eta)\Big)$ over the set of nonnegative measures.*

4.3 Proof of Theorem 3

The global optimality conditions for minimizing the convex functional F is that on the support of ν_∞ then $J(\eta|\nu_\infty) = 0$, while on the entire sphere $J(\eta|\nu_\infty) \geqslant 0$. The proof, adapted from [7], then goes a follows:

- The existence and uniqueness of the flow $(\nu_t)_{t\geq 0}$ can be proved by using the equivalence with a Wasserstein gradient flow $(\mu_t)_{t\geq 0}$ in $\mathcal{P}(\mathbb{R}^{d+1})$ and the theory of Wasserstein gradient flows [2]. As a matter of fact, $(\nu_t)_{t\geq 0}$ is itself a gradient flow for a certain metric between nonnegative measures that is, in a certain sense, the inf-convolution between the Wasserstein and the Hellinger metric, see the discussion in [7].

- The flow ν_t has a full support at all time t. This can be deduced from the representation of the solutions to Eq. (4.3) as

  $$\nu_t = X(t,\cdot)_\#\Big(\nu_0 \exp\Big(-4\int_0^t J(X(s,\cdot)|\nu_s)ds\Big)\Big),$$

  where $X : [0,+\infty[\times\mathcal{S}^d \to \mathcal{S}^d$ is the flow associated to the time-dependent vector field $-\nabla J(\cdot\,|\,\nu_t)$, i.e., it satisfies $X(0,\eta) = \eta$ and $\frac{d}{dt}X(t,\eta) = -\nabla J(X(t,\eta)|\nu_t)$ for all $\eta \in \mathcal{S}^d$, see, e.g., [27]. Under our regularity assumptions, standard stability results for ODEs guarantee that at all time t, $X(t,\cdot)$ is a diffeomorphism of the sphere. Thus ν_t is the image measure (this is what the "sharp" notation stands for) by a diffeomorphism of a measure of the form $\nu_0 \exp(\dots)$ which has full support and thus ν_t has full support.

- We assume that the flow converges to some measure ν_∞ (which could be singular). From Eq. (4.1), this imposes by stationarity of ν_∞ that $J(\eta|\nu_\infty) = 0$ on the support of ν_∞, but nothing is imposed beyond the support of ν_∞ (and we need non-negativity of $J(\eta|\nu_\infty)$ for all $\eta \in \mathcal{S}^d$).

  In order to show that $\min_{\eta\in\mathcal{S}^d} J(\eta|\nu_\infty) \geqslant 0$, we assume that it is strictly negative and will obtain a contradiction. We first need a $v < 0$ such that $v > \min_{\eta\in\mathcal{S}^d} J(\eta|\nu_\infty)$, and the gradient $\nabla J(\eta|\nu_\infty)$ does not vanish on the v-level-set $\{\eta \in \mathcal{S}^d, J(\eta|\nu_\infty) = v\}$ of $J(\cdot|\nu_\infty)$. Such a v exists because of Morse-Sard lemma which applies because under our assumptions, $J(\cdot|\nu)$ is d-times continuously differentiable for any finite nonnegative measure ν.

We then consider the set $K = \big\{\eta \in \mathcal{S}^d,\ J(\eta|\nu_\infty) \leqslant v\big\}$, which has some boundary ∂K, such that the gradient $\nabla J(\eta|\nu_\infty)$ has strictly positive dot-product with an outward normal vector to the level set at $\eta \in \partial K$.

Since ν_t converges weakly to ν_∞, there exists $t_0 > 0$ such that for all $t \geqslant t_0$, $\sup_{\eta\in K} J(\eta|\nu_t) < v/2$, while on the boundary $\nabla J(\eta|\nu_\infty)$ has non-negative dot-product with an outward normal vector. This means that for all $t > t_0$, applying Eq. (4.2) to the indicator function of K, if $a_t = \nu_t(K)$,

$$a'(t) \geqslant -4 \sup_{\eta\in K} J(\eta|\nu_t) a(t).$$

By the previous point, $a(t_0) > 0$ and thus, by Grönwall's lemma, $a(t)$ diverges, which is a contradiction with the convergence of ν_t to ν_∞.

4.4 Experiments

In order to illustrate[1] the global convergence result from earlier sections, we consider a supervised learning problem on $\mathbb{R}^2$, with Gaussian input data x, and output data given by a "teacher" neural network

$$y = \sum_{j=1}^{m_0} \theta_2(j) \max\{\theta_1(:,j)^\top x, 0\}$$

for some finite m_0 and weights θ_1 and θ_2. We consider $R(h)$ the expected square loss and stochastic gradient descent with fresh new samples (x_i, y_i) and a small step-size.

We consider several number m of hidden neurons, to assess when the original neurons can be recovered. In Figure 2, for large m (e.g., $m = 100$ or $m = 1000$), all learned neurons converge to the neurons that generated the function which is in accordance with our main global convergence result (note that in general, recovering the neurons of the teacher is not a necessary condition for optimality, but it is always sufficient), while for $m = 5 > m_0$, where the global optimum will lead to perfect estimation, we may not recover the global optimum with a gradient flow. An interesting open question is to characterize mathematically the case $m = 20$, where we obtain the global optimum with moderate m.

In Figure 3, we consider several random initializations and random "teacher" networks and compute the generalization performance of the neural network after optimization. We see that for large m, good performance is achieved, while when m is too small, local minima remain problematic. This experiment suggests that the probability of global convergence quickly tends to 1 as m increases beyond m_0 in this setting, even in moderately high dimension.

5 Generalization guarantees and implicit bias for overparameterized models

A shown above, overparameterization – which takes the form of large number of hidden neurons in our context – is a blessing for optimization, as it allows to ensure convergence to a global minimizer. When stochastic gradient descent with fresh observations at each iteration is used, then the predictor will converge to the optimal predictor (that is, it will minimize the performance on

[1] The code to reproduce Figures 2 and 3 is available on this webpage `https://github.com/lchizat/2021-exp-ICM`.

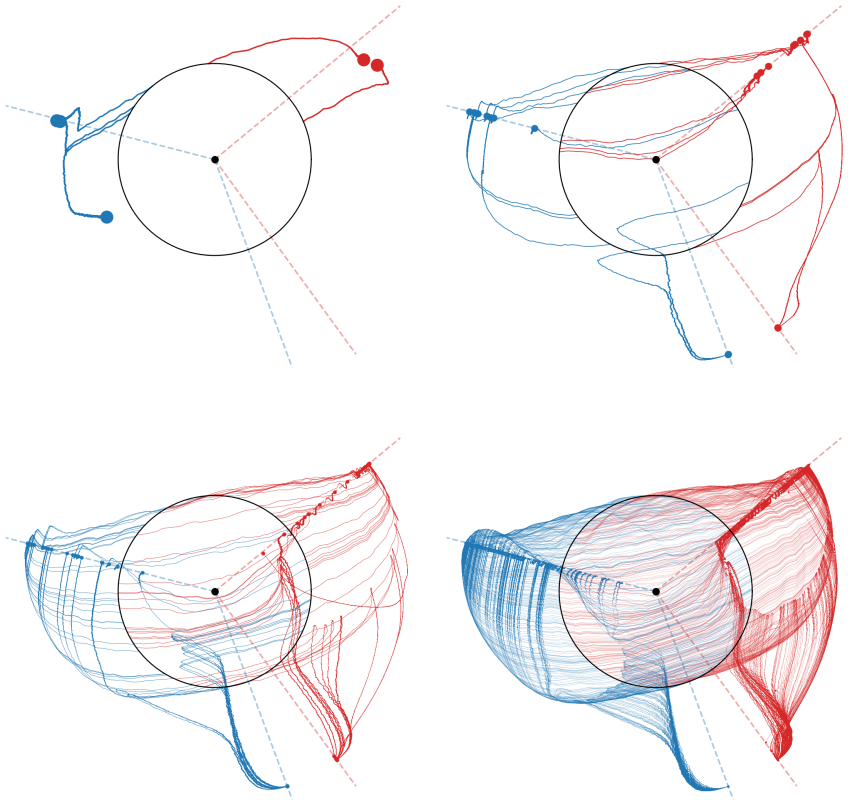


Figure 2: Gradient flow on a two-layer ReLU neural network with respectively $m = 5$, $m = 20$, $m = 100$ and $m = 1000$. The position of the particles is given by $|\theta_2(j)| \cdot \theta_1(\cdot, j)$ and the color depends on the sign of $\theta_2(j)$. The dashed directions represent the neurons of the network that generates the data distribution (with $m_0 = 4$). The unit circle, where the particles are initialized, is plotted in black and the radial axis is scaled by tanh to improve lisibility.

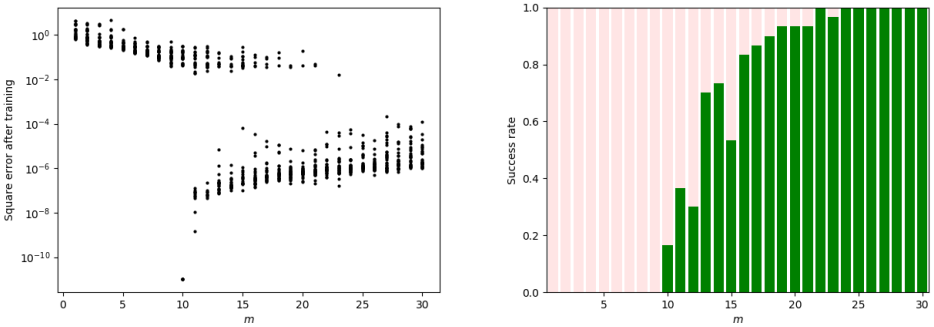


Figure 3: SGD on the square loss in the "teacher-student" setting (10^4 iterations, batch size 100, learning rate 0.005, $d = 100$, the teacher has $m_0 = 10$ neurons). (left) Risk (expected square loss) after training as a function of m over 30 random repetitions; (right) Success rate as a function of m over 30 repetitions (success means that the risk after training is below 10^{-3}).

unseen data), but will do so potentially at a slow speed, and with the need for many observations. In this context, overparameterization does not lead to overfitting, but may rather underfit.

In practice, several passes over a finite amount of data (n observations) are used, and then overparameterization can in principle lead to overfitting. Indeed, among all predictors that will perfectly predict the training data, some will generalize, some will not. In this section, we show that the predictor obtained after convergence of the gradient flow can in certain cases be characterized precisely.

To obtain the simplest result, following [18, 43, 17] this will be done for binary classification problems with the logistic loss. We will first review the implicit bias for linear models before considering neural networks.

5.1 Implicit bias for linear logistic regression

In this section, we consider a linear model $h(x, \theta) = \theta^\top \Phi(x)$ and we consider the minimization of the unregularized empirical risk with the logistic loss, that is,

$$\mathcal{R}(\theta) = \frac{1}{n} \sum_{i=1}^{n} \log \big(1 + \exp(-y_i \theta^\top \Phi(x_i)\big). \tag{5.1}$$

We consider a *separable* problem where there exists a linear function in $\Phi(x)$, $\theta^\top \Phi(x)$ such that $y_i \theta^\top \Phi(x_i) > 0$ for all $i \in \{1, \dots, n\}$. By rescaling, we may equivalently assume that there exists $\theta \in \mathbb{R}^d$ such that

$$\forall i \in \{1, \dots, n\},\ y_i \theta^\top \Phi(x_i) \geqslant 1.$$

This means that the objective function in Eq. (5.1) has an infimal value of zero, which is not attained for any θ, since it is strictly positive. However, taking any θ that separates the data as above, it holds that $\mathcal{R}(t\theta)$ converges towards 0 as t tends to infinity. There are thus in general an infinite number of directions towards which θ can tend to reach zero risk.

It turns out that gradient descent selects a particular one: the iterate of gradient descent will diverge, but its direction (that is the element of the sphere it is proportional to) will converge [43] to the direction of a *maximum margin classifier* defined as [46] a solution to

$$\min_{\theta \in \mathbb{R}^d} \ \|\theta\|_2^2 \quad \text{subject to} \quad \forall i \in \{1, \dots, n\},\ y_i \theta^\top \Phi(x_i) \geqslant 1. \tag{5.2}$$

The optimization problem above has a nice geometric interpretation (see Figure 4). These classifiers with a large margin has been shown to have favorable generalization guarantees in a wide range of contexts [22].

5.2 Extension to two-layer neural networks

We will now extend this convergence of gradient descent to a minimum norm classifier beyond linear models. We consider the minimization of the logistic loss

$$\frac{1}{n} \sum_{i=1}^{n} \log \big(1 + \exp(-y_i h(x_i)\big),$$

where $h(x) = \frac{1}{m} \sum_{j=1}^{m} \theta_2(j) \max\{\theta_1(:, j)^\top x, 0\}$ is a two-layer neural network. We will consider two regimes: (1) where only the output weights $\theta_2(j)$, $j = 1, \dots, m$ are optimized, and (2) where all

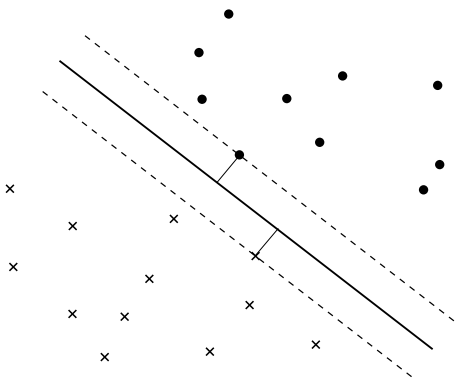


Figure 4: Geometric interpretation of Eq. (5.2) with a linearly separable binary classification problem in two dimensions (with each observation represented by one of the two labels $\times$ or $\bullet$): among all separating hyperplanes going through zero, the one with the largest minimal distance from observations to the hyperplane will be selected.

weights are optimized. In these two situations, we will let the width m go to infinity and consider the infinite-dimensional resulting flows. As shown in the previous section, when they converge, these flows converge to the global optimum of the objective function. But in the separable classification setting, the functions h should diverge. We essentially characterize towards which directions they diverge, by identifying the norms that are implicitly minimized [9].

5.3 Kernel regime

In this section, we consider random input weights $\theta_1(:, j)$, sampled from the uniform distribution on the sphere, and kept fixed throughout the optimization procedure. In other words, we only run the gradient flow with respect to the output weights $\theta_2 \in \mathbb{R}^m$.

Since the model is a linear model with feature vectors in m dimensions with components

$$\Phi(x)_j = \frac{1}{\sqrt{m}} \max\{\theta_1(:, j)^\top x, 0\},$$

we can apply directly the result above from [43], and the resulting classifier will minimize implicitly $\|\theta_2\|_2^2$, that is the direction of θ_2 will tend to a maximum margin direction.

In order to study the situation when the number of features m tends to infinity, it is classical within statistics and machine learning to consider the kernel function $\hat{k} : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ defined as

$$\hat{k}(x, x') = \Phi(x)^\top \Phi(x') = \frac{1}{m} \sum_{j=1}^{m} \max\{\theta_1(:, j)^\top x, 0\} \max\{\theta_1(:, j)^\top x', 0\}.$$

When m tends to infinity, the law of large number implies that $\hat{k}(x, x')$ tends to

$$k(x, x') = \mathbb{E}\Big[\max\{\eta^\top x, 0\} \max\{\eta^\top x', 0\} \Big],$$

for η uniformly distributed on the sphere.

Thus, we should expect that in the overparameterized regime, the predictor behaves like predictors associated with the limiting kernel function [32, 37]. It turns out that the kernel k can be computed in closed form [11], and that the reproducing kernel Hilbert space (RKHS) functional norm $\|\cdot\|$ associated to the kernel k is well understood (see below for a formula that defines it). In particular, this norm is infinite unless the function is at least $d/2$-times differentiable [4], and thus very smooth in high dimension (this is to be contrasted with the fact that each individual neuron leads to a non-smooth function). We thus expect smooth decision boundaries at convergence (see experiments below). This leads to the following result (see details in [9]):

Theorem 4 (informal) *When $m, t \to +\infty$ (limits can be interchanged), the predictor associated to the gradient flow converges (up to normalization) to the function in the RKHS that separates the data with minimum* RKHS norm $\|\cdot\|$*, that is the solution to*

$$\min_f \ \|f\|^2 \quad \textit{subject to} \quad \forall i \in \{1, \dots, n\}, \ y_i f(x_i) \geqslant 1.$$

Note that the minimum RKHS norm function can also be found by using the finite-dimensional representation $f(x) = \sum_{i=1}^n \alpha_i k(x, x_i)$ and minimizing $\sum_{i,j=1}^n \alpha_i \alpha_j k(x_i, x_j)$ under the margin constraint, which is a finite-dimensional convex optimization problem.

A striking phenomenon is the absence of catastrophic overfitting, where the observed data are perfectly classified but with a very irregular function that would potentially not generalize well. Despite the strong overparameterization, the classifier selected by gradient descent can be shown to generalize through classical results from maximum margin estimation. See [29] for a related result where the performance as a function of m, and not only for infinite m, is considered in special settings. We will see a similar behavior when optimizing the two layers, but with a different functional norm.

5.4 Feature learning regime

We now consider the minimization with respect to both input and output weights. This will correspond to another functional norm that will not anymore be an RKHS norm, and will allow for more adaptivity, where the learned function can exhibit finer behaviors.

We first provide an alternative formulation of the RKHS norm as [4]

$$\|f\|^2 = \inf_{a(\cdot)} \int_{\mathcal{S}^{d-1}} |a(\eta)|^2 d\tau(\eta) \ \text{ such that } \ f(x) = \int_{\mathcal{S}^{d-1}} (\eta^\top x)_+ a(\eta) d\tau(\eta),$$

where the infimum is taken over all square-integrable functions on the sphere $\mathcal{S}^{d-1}$, and τ is the uniform probability measure on the sphere. This formulation highlights that functions in the RKHS combine infinitely many neurons.

We can then define the alternative *variation norm* [23] as

$$\Omega(f) = \inf_{a(\cdot)} \int_{\mathcal{S}^{d-1}} |a(\eta)| d\tau(\eta) \ \text{ such that } \ f(x) = \int_{\mathcal{S}^{d-1}} (\eta^\top x)_+ a(\eta) d\tau(\eta),$$

where the infimum is now taken over all integrable functions on $\mathcal{S}^{d-1}$. Going from squared L_2-norms to L_1-norms enlarges the space by adding non-smooth functions. For example, a single neuron corresponds to $a(\cdot)d\tau(\cdot)$ tending to a Dirac measure at a certain point, and thus has a finite variation norm.

This leads to the following result (see details and full set of assumptions in [9]).

Theorem 5 (informal) *When $m, t \to +\infty$, if the predictor associated to the gradient flow converges (up to normalization), then the limit is the function that separates the data with minimum* ***variation norm*** *$\Omega(f)$, that is the solution to*

$$\min_f \ \Omega(f) \quad \textit{subject to} \quad \forall i \in \{1, \dots, n\}, \ y_i f(x_i) \geqslant 1.$$

Compared to the RKHS norm result, there is no known finite-dimensional convex optimization algorithms to efficiently obtain the minimum variation norm algorithm. Moreover, the choice of an L_1-norm has a sparsity-inducing effect, where the optimal $a(\cdot)d\tau(\cdot)$ will often corresponds to singular measure supported by a finite number of elements of the sphere. These elements can be seen as features learned by the algorithm: neural networks are considered as methods that learn representations of the data, and we provide here a justification with a single hidden layer. Such feature learning can be shown to lead to improved prediction performance in a series of classical situations, such as when the optimal function only depends on a few of the d original variables [4, 9].

5.5 Experiments

In this section, we consider a large ReLU network with $m = 1000$ hidden units, and compare the implicit bias and statistical performances of training both layers – which leads to a max margin classifier with the variation norm – versus the output layer – which leads to max margin classifier in the RKHS norm. These experiments are reproduced from [9].

Setting. Our data distribution is supported on $[-1/2, 1/2]^d$ and is generated as follows. In dimension $d = 2$, the distribution of input variables is a mixture of k^2 uniform distributions on disks of radius $1/(3k-1)$ on a uniform 2-dimensional grid with step $3/(3k-1)$, see Figure 6(a) for an illustration with $k = 3$. In dimension larger than 2, all other coordinates follow a uniform distribution on $[-1/2, 1/2]$. Each cluster is then randomly assigned a class in $\{-1, +1\}$.

Low dimensional illustrations. Figure 5 illustrates the differences in the implicit biases when $d = 2$. It represents a sampled training set and the resulting decision boundary between the two classes for 4 examples. The variation norm max-margin classifier is non-smooth and piecewise affine, which comes from the fact that the L_1-norm favors sparse solutions. In contrast, the max-margin classifier for the RKHS norm has a smooth decision boundary, which is typical of learning in a RKHS.

Performance. In higher dimensions, we observe the superiority of training both layers by plotting the test error versus m or d on Figure 6(b) and 6(c). We ran 20 independent experiments with $k = 3$ and show with a thick line the average of the test error $\mathbb{P}(yf(x) < 0)$ after training. For each m, we ran 30 experiments using fresh random samples from the same data distribution.

6 Discussion

In this paper, we have presented qualitative convergence guarantees for infinitely-wide two layer neural networks. These were obtained with a precise scaling – in the number of neurons – of the

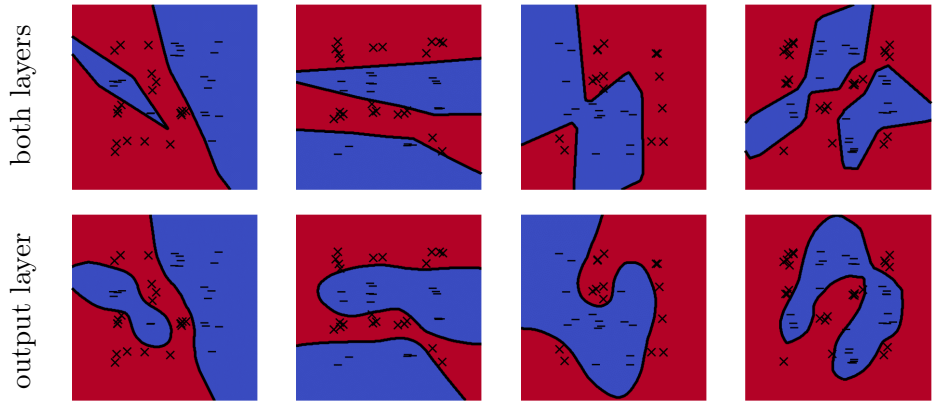
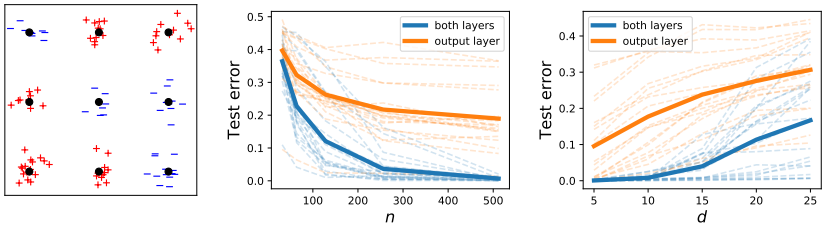


Figure 5: Comparison of the implicit bias of training (top) both layers versus (bottom) only the output layer for wide two-layer ReLU networks with $d = 2$ and for 4 different random training sets.



(a) Projected distribution

(b) Test error vs. n

(c) Test error vs. d

Figure 6: (a) Projection of the data distribution on the two first dimensions, (b) test error as a function of n with $d = 15$, (c) test error as a function of d with $n = 256$.

prediction function, the initialization and the step-size used in the gradient flow. With those scalings, the mean-field limit exhibits feature learning capabilities, as illustrated in binary classification where precise functional spaces could be used to analyse where optimization converges to. However, this limit currently does not lead to quantitative guarantees regarding the number of neurons or the convergence time, and obtaining such guarantees remains an open problem. This is an active area of research with, in particular, recent results concerning the local convergence [49, 1, 7] or global convergence under strong assumption on the data [26]. Moreover, extending this analysis to more than a single hidden layer or convolutional networks remains difficult.

Different scalings lead to different behaviors [10]. In particular, there is a scaling for which the limit behaves as a kernel method (even though all layers are trained, and not just the output layer) leading to another RKHS norm with a larger space than the one from Section 5.3, see [20]. While not leading to representation learning, extensions to deeper networks are possible with this scaling and provide one of few optimization and statistical guarantees for these models. Some recent progress has been made in the categorization of the various possible scalings for deep networks [48], and this emerging general picture calls for a large theoretical effort to understand the asymptotic behaviors of wide neural networks.

Acknowledgements. This work was funded in part by the French government under management of Agence Nationale de la Recherche as part of the "Investissements d'avenir" program, reference ANR-19-P3IA-0001 (PRAIRIE 3IA Institute). We also acknowledge support the European Research Council (grant SEQUOIA 724063).

References

[1] S. Akiyama and T. Suzuki, On learnability via gradient method for two-layer relu neural networks in teacher-student setting. Tech. Rep. 2106.06251, arXiv, 2021.

[2] L. Ambrosio, N. Gigli, and G. Savaré, *Gradient flows: in metric spaces and in the space of probability measures*. Springer Science & Business Media, 2008.

[3] D. Araújo, R. I. Oliveira, and D. Yukimura, A mean-field limit for certain deep neural networks. *arXiv preprint arXiv:1906.00193* (2019).

[4] F. Bach, Breaking the curse of dimensionality with convex neural networks. *Journal of Machine Learning Research* **18** (2017), no. 1, 629–681.

[5] A. R. Barron, Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Transactions on Information Theory* **39** (1993), no. 3, 930–945.

[6] S. Bubeck, Convex optimization: Algorithms and complexity. *Foundations and Trends in Machine Learning* **8** (2015), no. 3-4, 231–357.

[7] L. Chizat, Sparse optimization on measures with over-parameterized gradient descent. *Mathematical Programming* (2021), 1–46.

[8] L. Chizat and F. Bach, On the global convergence of gradient descent for over-parameterized models using optimal transport. *Advances in Neural Information Processing Systems* **31** (2018), 3036–3046.

[9] L. Chizat and F. Bach, Implicit bias of gradient descent for wide two-layer neural networks trained with the logistic loss. In *Conference on learning theory*, pp. 1305–1338, PMLR, 2020.

[10] L. Chizat, E. Oyallon, and F. Bach, On lazy training in differentiable programming. In *Advances in neural information processing systems*, 2019.

[11] Y. Cho and L. K. Saul, Kernel methods for deep learning. In *Advances in neural information processing systems*, pp. 342–350, 2009.

[12] W. E and S. Wojtowytsch, On the Banach spaces associated with multi-layer relu networks: Function representation, approximation theory and gradient descent dynamics. *arXiv preprint arXiv:2007.15623* (2020).

[13] C. Fang, J. Lee, P. Yang, and T. Zhang, Modeling from features: a mean-field framework for over-parameterized deep neural networks. In *Conference on learning theory*, pp. 1887–1936, PMLR, 2021.

[14] C. Giraud, *Introduction to high-dimensional statistics*. Chapman and Hall/CRC, 2021.

[15] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT Press, 2016.

[16] R. M. Gower, M. Schmidt, F. Bach, and P. Richtárik, Variance-reduced methods for machine learning. *Proceedings of the IEEE* **108** (2020), no. 11, 1968–1983.

[17] S. Gunasekar, J. Lee, D. Soudry, and N. Srebro, Characterizing implicit bias in terms of optimization geometry. In *International conference on machine learning*, pp. 1832–1841, 2018.

[18] S. Gunasekar, B. E. Woodworth, S. Bhojanapalli, B. Neyshabur, and N. Srebro, Implicit regularization in matrix factorization. In *Advances in neural information processing systems*, pp. 6151–6159, 2017.

[19] B. Hanin and M. Nica, Finite depth and width corrections to the neural tangent kernel. In *International conference on learning representations*, 2019.

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

[21] C. Jin, R. Ge, P. Netrapalli, S. M. Kakade, and M. I. Jordan, How to escape saddle points efficiently. In *International conference on machine learning*, pp. 1724–1732, PMLR, 2017.

[22] V. Koltchinskii and D. Panchenko, Empirical margin distributions and bounding the generalization error of combined classifiers. *The Annals of Statistics* **30** (2002), no. 1, 1–50.

[23] V. Kurkova and M. Sanguineti, Bounds on rates of variable-basis and neural-network approximation. *IEEE Transactions on Information Theory* **47** (2001), no. 6, 2659–2665.

[24] H. J. Kushner and G. G. Yin, *Stochastic approximation and recursive algorithms and applications*. Second edn., Springer-Verlag, 2003.

[25] J. D. Lee, M. Simchowitz, M. I. Jordan, and B. Recht, Gradient descent only converges to minimizers. In *Conference on learning theory*, pp. 1246–1257, 2016.

[26] Y. Li, T. Ma, and H. R. Zhang, Learning over-parametrized two-layer neural networks beyond NTK. In *Conference on learning theory*, pp. 2613–2682, PMLR, 2020.

[27] S. Maniglia, Probabilistic representation and uniqueness results for measure-valued solutions of transport equations. *Journal de mathématiques pures et appliquées* **87** (2007), no. 6, 601–626.

[28] S. Mei, T. Misiakiewicz, and A. Montanari, Mean-field theory of two-layers neural networks: dimension-free bounds and kernel limit. In *Conference on learning theory*, pp. 2388–2464, PMLR, 2019.

[29] S. Mei and A. Montanari, The generalization error of random features regression: Precise asymptotics and the double descent curve. *Communications on Pure and Applied Mathematics* (2019).

[30] S. Mei, A. Montanari, and P.-M. Nguyen, A mean field view of the landscape of two-layer neural networks. *Proceedings of the National Academy of Sciences* **115** (2018), no. 33, E7665–E7671.

[31] M. Mohri, A. Rostamizadeh, and A. Talwalkar, *Foundations of machine learning*. MIT Press, 2018.

[32] R. M. Neal, *Bayesian learning for neural networks*. Ph.D. thesis, University of Toronto, 1995.

[33] Y. Nesterov, *Lectures on convex optimization*. 137, Springer, 2018.

[34] P.-M. Nguyen and H. T. Pham, A rigorous framework for the mean field limit of multilayer neural networks. Tech. Rep. 2001.11443, arXiv, 2020.

[35] A. Nitanda and T. Suzuki, Stochastic particle gradient descent for infinite ensembles. Tech. Rep. 1712.05438, arXiv, 2017.

[36] A. Nowak-Vila, F. Bach, and A. Rudi, A general theory for structured prediction with smooth convex surrogates. Tech. Rep. 1902.01958, arXiv, 2019.

[37] A. Rahimi and B. Recht, Random features for large-scale kernel machines. *Advances in neural information processing systems* **20** (2007), 1177–1184.

[38] G. M. Rotskoff and E. Vanden-Eijnden, Parameters as interacting particles: long time convergence and asymptotic error scaling of neural networks. In *Advances in neural information processing systems*, pp. 7146–7155, 31, 2018.

[39] F. Santambrogio, *Optimal transport for applied mathematicians*. Springer, 2015.

[40] B. Schölkopf and A. J. Smola, *Learning with kernels*. MIT Press, 2001.

[41] J. Sirignano and K. Spiliopoulos, Mean field analysis of neural networks: A law of large numbers. *SIAM Journal on Applied Mathematics* **80** (2020), no. 2, 725–752.

[42] J. Sirignano and K. Spiliopoulos, Mean field analysis of deep neural networks. *Mathematics of Operations Research* (2021).

[43] D. Soudry, E. Hoffer, M. S. Nacson, S. Gunasekar, and N. Srebro, The implicit bias of gradient descent on separable data. *The Journal of Machine Learning Research* **19** (2018), no. 1, 2822–2878.

[44] E. Suli and D. F. Mayers, *An introduction to numerical analysis*. Cambridge University Press, 2003, 2003.

[45] A. W. Van der Vaart, *Asymptotic statistics*. 3, Cambridge University Press, 2000.

[46] V. N. Vapnik and A. Y. Chervonenkis, On a perceptron class. *Avtomat. i Telemekh.* **25** (1964), no. 1, 112–120.

[47] S. Wojtowytsch, On the convergence of gradient descent training for two-layer ReLU-networks in the mean field regime. Tech. Rep. 2005.13530, arXiv, 2020.

[48] G. Yang and E. J. Hu, Feature learning in infinite-width neural networks. Tech. Rep. 2011.14522, arXiv, 2020.

[49] M. Zhou, R. Ge, and C. Jin, A local convergence theory for mildly over-parameterized two-layer neural network. In *Conference on learning theory*, PMLR, 2021.