Mean-field Analysis of Neural Network: An Overview

Hongkang Yang HY1194@NYU.EDU
Courant Institute of Mathematical Sciences
New York University
New York, NY 10009, USA

Abstract

Neural networks have exhibited outstanding performance in terms of convergence and generalization, which attribute to neural networks' growing popularity. Contrary to traditional ideas on trainability and overfitting, empirical findings demonstrate that neural networks attain better performance when they are overparametrized, having much more neurons than required by the solution and greater than the amount of training data. In this paper, we offer a survey of the mean-field model of neural networks, an emerging theory that has elucidated part of the mystery of overparametrization. We will cover both the static and dynamic aspects of the theory: the static theory concerns function spaces defined by neural networks and explains their generalization ability, while the dynamic theory concerns the training behavior of overparametrized networks and leads to optimal convergence guarantees.

Keywords: Continuous neural network, mean-field analysis, expectation and particle method, neural functional analysis, optimal convergence.

1. Introduction

The conventional wisdom of machine learning is that we need to balance expressivity, trainability, and generalization. Apparently, the more complex a parametrized model, the more difficult it becomes to train the model to optimality and generalize well on unseen data instead of overfitting, so perhaps we always have to make a compromise. Yet, neural network and its success story seem to be an overt contradiction to this constraint. It has been widely observed that *overparametrized* neural networks, with parameters or neurons far exceeding the amount of training data, can be trained to zero loss and achieve very small generalization error [ZBH$^+$16]. This empirical surprise suggests two possibilities:

1. The non-convex loss landscape of neural networks is simplified, instead of becoming more convoluted, as the number of neurons N approaches infinity, and consequently enables convergence to global optimum.

2. The correct complexity measure (i.e. functional norm) of the target function (of a regression problem etc.) is intrinsically linked to neural networks while being independent of N, and thus enables small generalization error.

The phenomenon that large N or overparametrization facilitates learning implies that, in order to properly address the training and generalization of neural networks, we need to go beyond the "parameter space" perspective of neural nets and construct a new formalism more amenable to analysis. In this paper, we study such a continuous formulation of neural nets, which is based on the limit $N \to \infty$ and is known as the mean-field perspective.

There are two motivations for the infinite limit, one practical and one theoretical.

1.1 Practical motivation

Pioneering works from the last century have demonstrated empirically that solving regression with more neurons than necessary can help to accelerate training and avoid bad local minima [RHW85]. As the classical generalization bounds from statistical learning theory such as VC dimension grow proportionally to the number N of neurons [SSBD14] and thus are not applicable to analyzing overparametrization, researchers turned to look for bounds that do not depend on N. For instance, [Bar98]

derived the following type of bounds: with probability $1-\delta$,

$$\text{test error} \le \text{training error} + \frac{f(A,\delta)}{\sqrt{\text{sample size}}}$$

where A is an L^1-norm upper bound for the weights of two-layer networks, which is independent of N. A similar PAC-type bound is proposed in [LBW96], with sample complexity depending only on the L^1-norm bound.

More recent studies from this century such as [LL18, DZPS18, AZLL19] also confirmed that the training and generalization of neural networks become easier to analyze as $N \to \infty$. Even though the limiting rules of these studies differ from that of the mean-field analysis, they all indicate that it could be highly rewarding to develop a theory of a generalized neural network, which includes finite neural networks as a special case.

1.2 Theoretical motivation

One common technique in mathematical problem solving is "extension and embedding", such that if we are looking for a solution in some domain $\mathcal{X}$, then instead of solving directly in $\mathcal{X}$, we find a solution $x^* \in \mathcal{Y}$ for some larger space $\mathcal{Y} \supseteq \mathcal{X}$, and then prove that x^* actually lies in or can be approximated by $\mathcal{X}$. The heuristics is that the extension $\mathcal{Y}$ has more "regularity", such as linearity, completeness, compactness or algebraic closure, which enables us to apply more mathematical machinery to obtain a solution, while the regularity of the problem allows us to embed x^* back to $\mathcal{X}$.

One elementary example is solving a linear system of ODEs, $\dot{x} = Ax$, such that we extend from $\mathcal{X} =$ real solutions to $\mathcal{Y} =$ complex solutions and then construct $x^*(t)$ via eigenfunctions. We can easily come up with more well-known results: In PDE, for instance, we extend from classical C^k solutions to weak solutions and go back via Sobolev embedding [Eva10]. In graphon theory, we extend from finite graphs to continuous graphons and go back via limit sequences [Lov12]. In optimal transport, we extend from Monge solutions to Kantorovich solutions and go back via the regularity of the transported measure [Vil08].

Hence, one could expect that the finite neural network can be extended to some more general, continuous counterpart, and the analysis of training and generalization can be simplified by using more advanced tools from functional analysis and PDE.

Regarding what will or what should become the future developments of the theory of neural network, it is natural to look into history to identify which discipline started as empirical and unsophisticated as our present understanding of neural network and to study how a transparent theoretical foundation eventually emerged from this discipline. For instance, [EMW19c] offered an analogy between the present study of neural networks and the field of numerical analysis in 1950's, in particular, the stability analysis of the finite element and finite different schemes for solving PDE. Just as these schemes are discretizations of continuous PDE solutions, we can also consider neural networks as discretizations of some continuous object. We have seen that taking the "temporal limit" (i.e. gradient flow) can facilitate the analysis of training and convergence [B$^+$15], it is a reasonable next step to explore the "spatial limit" in the neurons.

The rest of this paper is structured as follows. In Section 2, we study the simple case of 2-layer neural networks and identify reasonable ways to define continuous limits, and the subsections will introduce useful ideas on functional analysis and Wasserstein gradient flow. Next, in Section 3, we discuss results on the mean-field analysis of 2-layer networks, in particular, on their generalization error bound and optimal convergence guarantee. Finally, Section 4 summarizes recent advances in the mean-field analysis of multilayer networks and residual networks.

2. Continuous neural network

We begin with the finite 2-layer networks, $f : \mathbb{R}^d \to \mathbb{R}$, temporarily defined by

$$f(x) = \sum_{i=1}^{N} \sigma(x; \theta_i) = \sum_{i=1}^{N} a_i \ \phi(w_i \cdot x + b_i) \tag{1}$$

where σ is the activation function defined by some nonlinearity ϕ. To take the limit $N \to \infty$, it is important to specify a limiting scheme or approximation scheme. There are two common schemes, one based on the central limit theorem (CLT) and the other on the law of large numbers (LLN),

- To inspire the CLT scheme, note that in practice we often initialize a neural network by drawing the weights a_i, w_i, b_i from some distributions such as Gaussian. By CLT, the function (1) grows at the rate of $O(\sqrt{N})$, and thus we can define the continuous limit as the following random function

  $$f(x) = \lim_{N\to\infty} \frac{1}{\sqrt{N}} \sum_{i=1}^{N} a_i \ \phi(w_i \cdot x + b_i) \tag{2}$$

  with i.i.d. a_i, w_i, b_i. This is effectively a Gaussian process and leads to the theory of neural tangent kernels (NTK) [JGH18].

- For the LLN scheme, we simply take the average of all neurons:

  $$f(x) = \frac{1}{N} \sum_{i=1}^{N} \sigma(x; \theta_i) = \frac{1}{N} \sum_{i=1}^{N} a_i \ \phi(w_i \cdot x + b_i)$$

  This formula is effectively an expectation

  $$f(x) = \mathbb{E}_{\rho(\theta)}\big[\sigma(x; \theta)\big] = \mathbb{E}_{\rho(a,w,b)}\big[a \ \phi(w \cdot x + b)\big] \tag{3}$$

  with the probability measure ρ given by $1/N \sum_{i=1}^{N} \delta_{\theta_i}$ [CB18]. Thus, for the continuous limit, we can define a (generalized) 2-layer network to be any function of the form (3) for any probability measure ρ. The LLN scheme gives rise to the mean-field analysis.

The heuristics of taking limits is that two factors matter—the approximating basis and the limiting rule. In our case, the approximating basis is fixed to be the finite 2-layer nets (1). The choice of the limiting rule is consequential and it can often be formalized as a functional norm: for instance, with C_C^∞ functions as our basis, if we take the limit in the L^p norm, C^k norm or Sobolev norm, then we obtain L^p, C^k or Sobolev functions. In fact, the CLT and LLN schemes correspond respectively to the RKHS norm (of the NTK) and the Barron norm [EMW19b], which will be covered in Section 2.1.

It might appear unnecessary to specify the limiting rules, considering that the various Universal Approximating Theorems have already shown that neural networks are dense in an array of spaces like C^0, L^p and Sobolev spaces [HSW90]. Nevertheless, different limiting rules lead to different function spaces, and this distinction will manifest in two highly non-trivial ways:

1. First, the limiting rule influences the training behavior. For the CLT scheme (2), the gradient updates are diluted among the neurons and vanish asymptotically as $O(1/\sqrt{N})$. It follows that the neurons are close to their initialization and asymptotically the learning reduces to kernel regression by NTK [COB19, DZPS18]. In contrast, for the LLN scheme, we will see in Section 2.2 below that the training becomes a gradient flow on the measure ρ in (3) and the updates on the neurons are non-trivial.

2. More importantly, the target function f^* we are learning could belong to different function spaces. If f^* has finite Barron norm but infinite RKHS norm (or vice versa), then it might happen that the gradient flow of the training process only converges under the LLN scheme instead of the CLT scheme. In practice, this difference could imply that the generalization error bound goes to zero for one scheme but not for the other.

From now on, we will focus on the LLN scheme (3) and mean-field analysis. Section 2.1 will discuss the static aspect of the theory that concerns the function space, and then Section 2.2 will cover the dynamic aspect that concerns the training behavior and gradient flow.

2.1 The Static Theory

Now, we seek to construct a suitable function space, or equivalently a functional norm $||\cdot||$, based on the LLN scheme (3). For neural networks, it is desirable that a functional norm have the following properties:

1. Approximation property: For any function f in this space (i.e. $||f|| < \infty$), we can approximate f arbitrarily well using (finite) neural networks f_N ($N = 1, 2, \dots$) with bounded norm, $\sup_N ||f_N|| \lesssim ||f||$. The approximation means that f is representable by neural nets, while the uniform norm bound means that f is learnable such that the training might converge. Eventually, we will be able to construct *a priori* generalization error bounds using $||f||$ (See Section 3.1).

2. Inverse approximation: If a function f can be approximated using (finite) neural networks with bounded norms, then f belongs to this space. The significance is that we are able to judge by the training process whether our target function is representable and learnable.

3. Escape curse of dimensionality: The norm $||f||$ should be "intrinsic" to the function f and do not depend on the ambient dimension. Moreover, the approximation rate of f by (finite) networks should also be independent of dimension. The removal of ambient dimension is crucial for machine learning applications, considering that many tasks such as image processing involve dimensions that are practically infinite. As remarked in [E19], high dimensionality is the key feature that separates machine learning from other mathematical problems.

One formulation that satisfies these properties is the Barron space [EMW19b]. The Barron norm $||\cdot||_{\mathcal{B}_p}$ (for $p \in [1, \infty]$) is defined by

$$||f||_{\mathcal{B}_p} = \inf_{\rho} \left(\mathbb{E}_{\rho} \big[|a|^p (||w|| + |b|)^p \big] \right)^{1/p} \tag{4}$$

where the inf ranges over all ρ that represents f, $f(x) = \mathbb{E}_{\rho(\theta)}[\sigma(x; \theta)]$. The technical detail is that we are using ReLU for the activation, but it seems probable that all results below can be applied to other activations.

The Barron space $\mathcal{B}$ consists of all functions with finite Barron norm. The exponent p has no influence on the norm, that is, $||f||_{B_p}$ is equal for all $p \in [1, \infty]$ [EMW19b] and we can write $||f||_{\mathcal{B}}$ for convenience. As an assurance, the space $\mathcal{B}$ contains most of the functions of our interest, and in particular, if the Fourier transform of f has finite second moment:

$$\int ||\xi||^2 |\hat{f}(\xi)| d\xi < \infty$$

then f belongs to the Barron space [Bre93].

Before we discuss the specific theorems, let us mention one common advantage of the LLN scheme and the Barron norm. Note that both (3) and (4) are formulated using expectations $\mathbb{E}_{\rho}$, so it is natural to approximate the expectation using Monte-Carlo estimates [EMW19c]. In general, the approximation error should scale as

$$\left| \mathbb{E}_{\rho} \big[\text{quantity}(\theta) \big] - \frac{1}{N} \sum_{i=1}^{N} \text{quantity}(\theta_i) \right| \lesssim \frac{\text{standard deviation}(\rho)}{\sqrt{N}}$$

where θ_i are i.i.d. samples from the distribution ρ. This estimate suggests that a generalized network (3) can be approximated by a finite network, and that the approximation error does not depend on dimension.

Back to the aforementioned three properties, we have that the Barron functions (functions of $\mathcal{B}$) can be approximated by finite 2-layer neural networks.

Theorem 2.1 (Theorem 4 of [EMW19b]). For any Barron function $f \in \mathcal{B}$ and any N, there exists a 2-layer network $f_N(x;\Theta)$ with N neurons and parameters $\Theta = (\theta_1, \dots \theta_N)$ such that

$$||f - f_N(\cdot;\Theta)||^2_{L^2(\mu)} \leq \frac{3||f||^2_{\mathcal{B}}}{N}$$

(where $L^2(\mu)$ is the L^2 norm with respect to any fixed data distribution $\mu(x)$). Furthermore, the "path norm" of Θ can be uniformly bounded,

$$||\Theta||_P := \frac{1}{N}\sum_{i=1}^{N} |a_i|\big(||w_i|| + |b_i|\big) \leq 2||f||_{\mathcal{B}}$$

This theorem implies that the Barron norm satisfies the Property 1 above, and furthermore the bounds are dimension-independent, so it aligns with Property 3. The proof technique is to draw the parameters θ_i from a distribution ρ that represents the Barron function f.

Regarding Property 2, we have the following result.

Theorem 2.2 (Theorem 5 of [EMW19b]). Given a continuous function f, if a sequence of (finite) neural networks $f_N(\cdot;\Theta_N)$ converges pointwise to f, and if the path norms $||\Theta_N||_P$ are uniformly bounded, then f is a Barron function and

$$||f||_{\mathcal{B}} \leq \sup_N ||\Theta_N||_P$$

Finally, let us remark that, just as the CLT scheme leads to the neural tangent kernels, the Barron space can also be formulated using RKHS [EMW19b]. The Barron space $\mathcal{B} = \mathcal{B}_2$ (by the equivalence of the exponent p) consists of all functions of the form

$$f(x) = \mathbb{E}_\rho\big[a\ \phi(\tilde{w}\cdot\tilde{x})\big],\ \mathbb{E}_\rho\big[|a|^2||\tilde{w}||^2\big] < \infty$$

where $\tilde{w} = (w, b)$ and $\tilde{x} = (x, 1)$. We can scale $\tilde{w}$ to be on the sphere $\mathbb{S}^d$ (by the homogeneity of ReLU), and collect the conditional distributions $\rho(a|\tilde{w})$ into a measureable function $a(\tilde{w})$. Then, the above formula becomes

$$\begin{aligned} f(x) &= \mathbb{E}_{\rho(\tilde{w})}\big[a(\tilde{w})\phi(\tilde{w}\cdot\tilde{x})\big] \\ \mathbb{E}_\rho\big[|a|^2||\tilde{w}||^2\big] &= ||a(\tilde{w})||^2_{L^2(\rho(\tilde{w}))} < \infty \end{aligned}$$

This condition means exactly that f belongs to the RKHS H_ρ defined by the kernel

$$k_\rho(x, x') = \mathbb{E}_{\rho(\tilde{w})}\big[\phi(\tilde{w}\cdot\tilde{x})\phi(\tilde{w}\cdot\tilde{x}')\big]$$

It follows that the Barron space is a union of RKHS,

$$\mathcal{B} = \cup_\rho H_\rho$$

where the union ranges over all distributions $\rho(\tilde{w})$ on the sphere $\mathbb{S}^d$. In a sense, it suggests that the LLN scheme could be more flexible than the CLT scheme, because the neural tangent kernel given by the CLT scheme is always fixed, while for the LLN scheme, the distribution $\rho(\tilde{w})$ evolves during training and thus the corresponding kernel k_ρ is adaptive.

2.2 The Dynamic Theory

Next, we move on to study the training of the continuous neural network (3), and we need to establish the infinite limit analogue of the gradient descent or gradient flow. The evolution of each neuron parameter $\theta(t)$ is characterized by its velocity $v(t)$, and for a finite neural network, the evolution of its empirical measure

$$\rho_t^N = \frac{1}{N}\sum_{i=1}^{N} \delta_{\theta_i(t)} \tag{5}$$

is characterized by the vector field

$$V_t(\theta) = \sum_i v_i(t) \mathbf{1}_{\theta=\theta_i(t)}$$

For the general network (3), its parameter distribution ρ can be seen as a nondeterministic cloud of parameters Θ (a random variable $\sim \rho$) and its corresponding velocity is a random vector located at the random position Θ. So the velocity of the evolution ρ_t can be characterized as a velocity field $V_t(\theta)$ on the support of ρ_t (or more precisely, the vector-valued measure $V_t\rho_t$).

The gradient descent training rule imposes strong physical constraints on how the probability measure $\rho(t)$ could evolve. The gradient flow of each parameter $\theta(t)$ is continuous—it cannot be teleported instantaneously to remote places—and thus the evolution $\rho(t)$ satisfies the *conservation of local mass*. This conservation law states that given any region Ω, the total mass of $\rho(t)$ inside Ω can only change through neurons moving in and out through the boundary $\partial\Omega$:

$$\frac{d}{dt}\int_\Omega d\rho_t + \int_{\partial\Omega} \vec{n} \cdot dV_t\rho_t = 0$$

where $\vec{n}$ is the outward pointing normal vector field on $\partial\Omega$. Then, by the divergence theorem, the parameter distribution ρ_t satisfies the continuity equation (or linear transport equation):

$$\partial\rho_t + \nabla \cdot (V_t\rho_t) = 0 \tag{6}$$

This equation holds in the weak sense (with respect to test functions), and is known in literature as Liouville's equation [RVE19] or distributional dynamics [MMN18].

Abstractly, a law of motion such as (6) determines the structure or topology of the space. The space of probability distributions $\mathcal{P}(\mathbb{R}^{d+2})$ ($d+2$ is the dimension of a, w, c) can be seen as a Riemannian manifold whose tangent space at each point ρ consists of velocity fields V, and the "exponential map" that transport ρ by V is given by equation (6) (A rigorous discussion can be found in [JKO98]). Given a path ρ_t ($t \in [0,1]$) with velocity V_t, we can calculate the length of the path by $\int_0^1 ||V_t|| dt$ for some norm $||V||$ on the tangent space. Using the natural choice of $L^2(d\rho_t)$ norm, the length l can be defined by

$$l^2 = \int_0^1 ||V_t||^2_{L^2(d\rho_t)} dt$$

It follows that we can define a geodesic distance on $\mathcal{P}(\mathbb{R}^{d+2})$ by

$$d(\rho_0, \rho_1)^2 := \inf_{\rho_t, V_t} \int_0^1 ||V_t||^2_{L^2(d\rho_t)} dt$$

where (ρ_t, V_t) ranges over all pairs that connect ρ_0, ρ_1 and satisfy the linear transport equation (6). Now, a very useful result is that this geodesic distance is exactly the Wasserstein metric W_2 on the space $\mathcal{P}_2(\mathbb{R}^{d+2})$ of probabilities with finite second moments. This result is known as the Benamou-Brenier theorem [BB00] and further discussions can be found in [Vil03, San15].

The significance of this line of reasoning is that the evolution of the parameter distribution ρ_t satisfies the conservation of local mass, and thus can be characterized as a flow in the Wasserstein space $(\mathcal{P}_2(\mathbb{R}^{d+2}), W_2)$ and be analyzed using powerful tools from the theory of Wasserstein gradient flows [AGS08].

Now we move on to study optimization problems on probability measures. Given an objective function $L(\rho)$ on $\rho \in \mathcal{P}(\mathbb{R}^{d+2})$, how do we find a minimizer under the constraint of the Wasserstein geometry? As a trivial example, let $L(\rho)$ be linear:

$$L(\rho) = \int E(\theta) d\rho(\theta)$$

where E is some potential function. Then, it is straightforward to show that ρ is a minimizer if and only if ρ is concentrated on the minimizers of E. Nevertheless, we cannot simply teleport each neuron

to the minimizer set. Instead, they descend continuously on E along the velocity field $V = -\nabla E$, and the continuity equation (6) leads to the evolution equation:

$$\partial_t \rho_t = \nabla \cdot (\rho_t \nabla E)$$

In general, $L(\rho)$ is not linear, so we look at its first-order approximation. Classically, the probability spaces $\mathcal{P}$ was not studied with the Wasserstein topology, but as a convex subspace of the linear space of (signed) Randon measures. The linearity allows us to speak of convexity and derivative, and in particular, let $L(\rho)$ be a convex function, we can define its subdifferential (or first variation) at ρ as any function F that satisfies, for any $\rho' \in \mathcal{P}$,

$$\liminf_{t \to 0^+} \frac{L\big(\rho + t(\rho' - \rho)\big) - L(\rho)}{t} \geq \int F \, d(\rho' - \rho) \tag{7}$$

Often, the subdifferential F is unique (up to an additive constant, which is irrelevant) and we denote it by $\delta L / \delta \rho$. Then, we can perform gradient flow along the velocity field $V_t = -\nabla \frac{\delta L}{\delta \rho}(\theta; \rho_t)$, and the continuity equation in this general case is given by

$$\partial \rho_t = \nabla \cdot \big(\rho_t \nabla \frac{\delta L}{\delta \rho}(\theta; \rho_t)\big) \tag{8}$$

It is straightforward to show that $L(\rho_t)$ is monotonically decreasing (or dissipates energy):

$$\frac{dL(\rho_t)}{dt} = \int \frac{\delta L}{\delta \rho}(\theta; \rho_t) \; \partial_t \rho_t = -\int \big|\big|\nabla \frac{\delta L}{\delta \rho}\big|\big|^2 d\rho_t \leq 0$$

A rigorous discussion of the Wasserstein gradient flow can be found in [AGS08, San15].

Note that in practice, our objective function $L(\rho)$ is often convex (in the linear space view): L could be defined as a convex function of the neural network f (e.g. the mean-square loss $\mathbb{E}||f - f^*||^2$), while the neural network $f = f(x; \rho)$ by definition (3) is linear in ρ, so the objective L is linear in ρ and the condition (7) holds for all $t \in [0, 1]$. Nevertheless, as we will see in Section 3.2, it is not apparent whether this convexity can benefit neural network training, because the latter is constrained by the "dissipation mechanism" (8). Even though the gradient flow dissipates the energy L along the way, the dissipation mechanism sets barriers for the flow to achieve global minimum [Pel14].

To see how equation (8) is equivalent to our ordinary gradient flow on each parameter θ_i, we can plug in the empirical distribution (5) and then it is straightforward to recover the classical gradient flow [RVE19]:

$$\dot{\theta}_i(t) = -\nabla \frac{\delta L}{\delta \rho}\big(\theta_i(t); \rho_t^N\big) = -N \cdot \frac{\partial L}{\partial \theta_i} \tag{9}$$

A technical detail is that the gradient is multiplied by N; otherwise, the gradient update becomes diluted among the N neurons.

Just as we simulate fluid flow using Lagrangian markers, the N neurons θ_i could be seen as a finite particle approximation of some continuous flow. Specifically, assume that the neurons' initializations $\theta_i(0)$ are i.i.d. random variables of some distribution ρ_0 such as Gaussian. Let ρ_t be the solution of the Wasserstein gradient flow (8) with initialization ρ_0, and let ρ_t^N be the empirical measure of $\theta_i(t)$ under the classical gradient flow (9). Then, it is reasonable to expect that ρ_t^N, as a "discretization" of ρ_t, should converge to ρ_t as $N \to \infty$. Indeed, this intuition is true:

Theorem 2.3 (Theorem 2.6 of [CB18])**.** Under technical assumptions, the flow ρ_t^N converges weakly to ρ_t.

A similar result is established by Theorem 3 of [MMN18] for discrete time gradient descent.

This type of results is an instance of the phenomenon of *propagation of chaos* [Szn91]. The idea is that the initializations $\theta_i(0)$ are independent, but in the classical gradient flow (9), the term $L = L(\rho_t^N)$ involves all particles $\theta_i(t)$, and thus they influence each other along the way and breaks their independence. Yet, their mutual influences are of a special kind—each particle only "feels" the others'

presence via their average, the empirical measure ρ_t^N—and this type of systems are known as mean-field particle flow. As $N \to \infty$, the randomness of ρ_t^N reduces to the determinacy of ρ_t, and thus the evolution of each particle can be approximated by the McKean-Vlasov equation:

$$\dot{\theta}_i(t) = -\nabla \frac{\delta L}{\delta \rho}\big(\theta_i(t); \rho_t\big)$$

It follows that the particles remain independent in the infinite limit, and thus the name propagation of independence/chaos.

Remark 1. In this section, we derived the Wasserstein geometry from neural network training's conservation of local mass. Let us mention an alternative argument to show that the Wasserstein geometry is the natural way to look at neural network's parameter space. For finite 2-layer networks, modelled by $(\theta_1, \dots \theta_N)$, [MMN18] observed that the labeling $\{1, \dots N\}$ is nonessential for the neurons—the network only depends on the empirical measure $1/N \sum \delta_{\theta_i}$, which is invariant under permutations of the neurons. Thus, if we are going to impose a metric topology on the parameter space, then a reasonable metric should be invariant under permutations. The canonical way to define an invariant metric is through the quotient: Given any metric d on the parameter space and two sets of parameters, $\Theta = (\theta_1, \dots \theta_N)$ and $\Xi = (\xi_1, \dots \xi_N)$, the quotient (pseudo)metric is given by

$$\tilde{d}(\Theta, \Xi) = \inf_{\tau \in S_N} d\big(\tau(\Theta), \Xi\big)$$

where S_N is the permutation group on N elements and $\tau(\Theta) = (\theta_{\tau(1)}, \dots \theta_{\tau(N)})$. This new metric $\tilde{d}$ is the optimal matching cost derived from d, and if d is the Euclidean distance, then $\tilde{d}$ is exactly the W_2 metric between the empirical measures of Θ and Ξ. Now, given general distributions ρ and μ, let Θ and Ξ be their i.i.d. samples, then standard LLN arguments (e.g. Corollary 2.14 of [Lac18]) show that

$$\tilde{d}(\Theta, \Xi) \to W_2(\rho, \mu)$$

almost surely as $N \to \infty$. Hence, solely from the property that 2-layer networks are permutation-invariant, we recover the Wasserstein metric.

3. Results for 2-layer nets

Having laid the theoretical foundation, we now apply this framework to the study of the training and generalization of 2-layer neural networks. We begin with the static results concerning the function space analysis and generalization errors. Then, we discuss the dynamic results involving training and optimality guarantees.

3.1 Static: Barron space

We have seen in Section 2.1 that the formulation of a Barron function as an expectation is the key to establishing approximation with dimension-free bounds. By similar arguments, one can show that Barron functions have low complexity and thus can be learned with small generalization error. Intuitively, given a target function f in the Barron space, represented by a probability measure ρ, we can approximate f arbitrarily well by sampling from ρ. Then, f is approximately an average of finitely many activation functions $\sigma(x; \theta_i)$ and should be simple. Formally, we have the follow result.

Theorem 3.1 (Theorem 6 of [EMW19b])**.** Let $\mathcal{B}_R$ be the set of functions f with bounded Barron norm: $||f||_{\mathcal{B}} \leq R$, then the Rademacher complexity Rad_n of $\mathcal{B}_R$ is bounded by

$$Rad_n(\mathcal{B}_R) \leq 2R\sqrt{\frac{2\ln(2d)}{n}}$$

where the test points $(x_1, \dots x_n)$ of Rad_n are taken from $[0, 1]^d$.

Note that this complexity is almost as small as that of families of linear functions [SSBD14]. Then, this complexity bound can be translated to an *a priori* generalization error bound.

Theorem 3.2 (Theorem 3.1 of [EMW18])**.** Fix a regularization constant $\lambda \geq 4\sqrt{\ln(2d)/n}$ and some data distribution $\mu(x)$ on $[0,1]^d$. Given any target function f^* in the Barron space, let Θ be the parameter of a width-N 2-layer ReLU neural network that minimizes the regularized objective:

$$L_n(\Theta) = \frac{1}{2n}\sum_{j=1}^{n}\big(f^*(x_j) - f(x_j;\Theta)\big)^2 + \lambda \cdot ||\Theta||_P$$

where $\{x_j\}_{i=1}^n$ is a sample of μ and $||\Theta||_P = \sum_{i=1}^N |a_i| \cdot ||\tilde{w}_i||$ is the path norm. Then, with probability $1-\delta$ over the i.i.d. sampling of $\{x_j\}$, the generalization error can be bounded by

$$\mathbb{E}_\mu\big|f^* - f(\cdot;\Theta)\big|^2 \lesssim \frac{||f^*||_B^2}{N} + \lambda \max\big(||f^*||_{\mathcal{B}}, 1\big) + \frac{\max\big(||f^*||_{\mathcal{B}}, 1\big) + \sqrt{\ln(n/\delta)}}{\sqrt{n}}$$

In short, the generalization error of every regularized minimizer is $O\big(\frac{1}{N} + \frac{1}{\sqrt{n}}\big)$ and this bound is almost independent of the ambient dimension d. Moreover, it is *a priori* in the sense that it only depends on the norm of the target $||f^*||_{\mathcal{B}}$ but not on the solution $f(\cdot;\Theta)$.

3.2 Dynamic: Optimal convergence guarantee

Next, we move on to study the training dynamics of the continuous 2-layer neural network. In particular, we will be concerned with optimality guarantees, that is, whether gradient flow can arrive at global minima or be trapped at local minima.

As remarked in Section 2.2, our objective function $L(\rho)$ is usually convex in ρ. Intuitively, if the subgradient $\nabla \delta L/\delta\rho$ is zero at ρ, then ρ is a global minimizer. This intuition can be formalized by:

Lemma 3.3 (Proposition 7.20 of [San15] and Proposition 3.1 of [CB18])**.** Under regularity conditions, given a convex objective function $L(\rho)$, ρ is a global minimizer if and only if ρ is supported in argmin $\delta L/\delta\rho$.

The latter condition means that no matter where we teleport the neurons of ρ, we cannot decrease $L(\rho)$ any more. Yet, the issue is that $\delta L/\delta\rho$ might have a convoluted landscape while the neurons can only move continuously along $-\nabla\delta L/\delta\rho$, so even if the flow converges and we have $\nabla\delta L/\delta\rho = 0$ ρ-almost everywhere, it does not imply in general that ρ has arrived at the global minimizers of $\delta L/\delta\rho$.

Fortunately, in the particular setting of neural network training, we can overcome this problem. Let us illustrate one useful technique from [CB18, RVE19], in the simple case of the mean-square loss: Let $f^*(x)$ be the target function,

$$\begin{aligned} L(\rho) &= \frac{1}{2}\mathbb{E}_x\big(f(x) - f^*(x)\big)^2 = \frac{1}{2}\mathbb{E}_x\big(\mathbb{E}_{\rho(\theta)}[\sigma(x;\theta)] - f^*\big)^2 \\ &= \frac{1}{2}\mathbb{E}_x\big(f^*(x)\big)^2 - \mathbb{E}_{\rho(\theta)}\mathbb{E}_x\big[f^*(x)\sigma(x;\theta)\big]^2 + \frac{1}{2}\mathbb{E}_{\rho^{\otimes 2}(\theta,\theta')}\mathbb{E}_x\big[\sigma(x;\theta)\sigma(x;\theta')\big] \end{aligned}$$

Then, we are effectively performing gradient flow on the following landscape

$$L(\rho) = \mathbb{E}_\rho\big[E(\theta)\big] + \frac{1}{2}\mathbb{E}_{\rho^{\otimes 2}}\big[K(\theta,\theta')\big]$$

where $E(\theta) = -\mathbb{E}_x[f^*(x)\sigma(x;\theta)]$ can be seen as a potential energy and $K(\theta,\theta') = \mathbb{E}_x[\sigma(x;\theta)\sigma(x;\theta')]$ is an interaction energy [MMN18]. The gradient flow is characterized by (8) with the velocity field given by

$$V_t(\theta) = \nabla\frac{\delta L}{\delta\rho}(\theta;\rho_t) = \nabla E(\theta) + \mathbb{E}_{\rho_t(\theta')}\nabla K(\theta,\theta') = \mathbb{E}_x\big[(f-f^*)\nabla_\theta\sigma(x;\theta)\big]$$

and the objective function $L(\rho_t)$ evolves by

$$\frac{dL}{dt} = -\mathbb{E}_{\rho_t(\theta)}||V_t(\theta)||^2 \tag{10}$$

Recall from (1) that the activation function has the form $\sigma(x;\theta) = a\ \phi(w \cdot x + c)$, so the derivative becomes

$$\frac{dL}{dt} = -\mathbb{E}_{\rho_t(w,c)}\left|\mathbb{E}_x(f - f^*)\phi(w \cdot x + c)\right|^2 - \mathbb{E}_{\rho_t(a,w,c)}\left|\mathbb{E}_x(f - f^*)a\nabla\phi(w \cdot x + c)(x,1)^T\right|^2$$

Now assume that ρ_t converges to some ρ^* as $t \to \infty$, then the derivative, in particular its first term, vanishes:

$$\mathbb{E}_{\rho^*(w,c)}\left|\mathbb{E}_x(f - f^*)\phi(w \cdot x + c)\right|^2 = 0$$

If the marginal distribution $\rho^*(w,c)$ has full support over, say, the sphere $(w,c) \in \mathbb{S}^d$, then

$$\mathbb{E}_x(f - f^*)\phi(w \cdot x + c) = 0$$

for all directions (w,c). By universal approximation theorem, we must have $f = f^*$ everywhere. Equivalently, since $\delta L/\delta\rho = 0$ everywhere, Lemma 3.3 implies that ρ^* is the global minimizer.

In summary, we have the following theorem.

Theorem 3.4. (Theorem 3.3 of [CB18] and Proposition 3.4 of [RVE19]) Under technical assumptions, given an objective function $L(f)$ that is convex in f, and we model f by a continuous 2-layer neural network, if the gradient flow ρ_t defined by (8) converges in Wasserstein metric to some ρ^*, then ρ^* is a global minimizer of L.

Note that there are two difficulties. First, the above argument needs the final ρ^* to cover all directions in (w,c). This condition can be relaxed by assuming that the initialization ρ_0 contains some submanifold that covers all directions, and that the training is smooth enough so that this condition continue to hold for all finite time t.

Second, these results assume that ρ_t converges, but this condition does not hold in general. We will discuss this issue in Section 3.3 below.

Recall that Theorem 2.3 established that the training of finite 2-layer network ρ_t^N converges to the gradient flow of the continuous network ρ_t. Combining Theorem 3.4, we have an optimal convergence guarantee for sufficiently overparametrized network.

Theorem 3.5 (Theorem 3.3 of [CB18] and Proposition 3.5 of [RVE19])**.** In the setting of Theorem 3.4, let ρ_t^N be the classical gradient flow of a finite 2-layer network with N neurons and let ρ_t be the Wasserstein gradient flow of a continuous network, if the initialization ρ_0^N converges weakly to ρ_0 (e.g. ρ_0^N is an i.i.d. sampled from ρ_0), then

$$\lim_{N\to\infty}\lim_{t\to\infty} L(\rho_t^N) = \lim_{t\to\infty}\lim_{N\to\infty} L(\rho_t^N) = \lim_{t\to\infty} L(\rho_t) = \inf L$$

In particular, for the mean-square regression problem, let $f_t^N = f(\rho_t^N)$ and $f_t = f(\rho_t)$ be the finite and continuous networks, then

$$\lim_{N\to\infty}\lim_{t\to\infty} f_t^N = \lim_{t\to\infty} f_t = f^*$$

The analysis of the more general case with online stochastic gradient descent can be found in [RVE19, MMN18].

Hence, we have verified the intuition from Section 1.2 that overparametrization can be seen as a finite particle approximation of some continuous flow with desirable convergence property, and increasing the number of flow particles contributes to the accuracy in the modeling of this continuous flow. In particular, the speed of convergence does not depend on the number N of neurons.

Naturally, the next question is to explore the finite data setting analogous to Theorem 3.2, and show that overparametrized networks can dynamically achieve such a generalization error bound $O(1/N + 1/\sqrt{n})$. Unlike the explicit regularization term in Theorem 3.2, we can hope that training with gradient descent could induce some implicit regularization that encourages small path norm.

Finally, let us mention some further developments along this line of works. Note that even though inequality (10) shows that the objective L is decreasing along the gradient flow ρ_t, it does not provide

an explicit rate. [RJBVE19] proposed one modification of the training process, called the "neuron birth-death process", that achieves an $O(1/t)$ rate of convergence. Specifically, the training rule (8) is revised to the following non-local transport equation,

$$\partial \rho_t = \nabla \cdot \Big(\rho_t \nabla \frac{\delta L}{\delta \rho}(\theta; \rho_t) \Big) - \alpha \Big(\frac{\delta L}{\delta \rho} - \overline{\frac{\delta L}{\delta \rho}} \Big) \rho_t \tag{11}$$

where $\alpha > 0$ is a constant and $\overline{\delta L/\delta \rho} = \mathbb{E}_{\rho_t}[\delta L/\delta \rho]$ is the average. Recall from Section 2.2 that during gradient flow the neurons descend on the landscape of $\delta L/\delta \rho$. To accelerate the descent, we can teleport those neurons inside the region where $\delta L/\delta \rho$ is higher than average to the region below average. This modified dynamics satisfies the conservation of total mass, and can be implemented in practice by adding and deleting neurons. Then, the dissipation inequality (10) can be strengthened to

$$\frac{dL}{dt} = -\mathbb{E}_{\rho_t} \Big|\Big| \nabla \frac{\delta L}{\delta \rho} \Big|\Big|^2 - \alpha \ \mathbb{E}_{\rho_t} \Big(\frac{\delta L}{\delta \rho} - \overline{\frac{\delta L}{\delta \rho}} \Big)^2$$

which leads to the following logarithmic rate of convergence.

Theorem 3.6 (Theorem 4.4 of [RJBVE19])**.** Under technical assumptions, if the initialization ρ_0 has full support, and if the gradient flow ρ_t under (11) converges weakly to some ρ^*, then ρ^* is a global minimizer and

$$L(\rho_t) - L(\rho^*) = O(t^{-1})$$

3.3 Open question: Establish convergence

Recall that in the above argument, we showed that if ρ_t ever converges, we necessarily have $f = f^*$; in other words, the target f^* lies in the Barron space $\mathcal{B}$. Therefore, if the target does not have a finite Barron norm, then ρ_t necessarily diverges (given the theorem's setting). It suggests that in order to establish convergence, we might need to view the training process from the Barron norm perspective. For instance, one possible approach would be to define a Lyapunov function involving the term $||f - f^*||_{\mathcal{B}}^2$.

The Lyapunov approach works in the simpler setting of the random feature model (more details can be found in Section 6 of [EMW19c]). By random feature model, we mean that the first layer is fixed and we only train the second layer, e.g. $\rho(w, c)$ can be set to the uniform distribution over $\mathbb{S}^d$. Consider again the mean-square loss $L(\rho)$ with a target function $f^* = f(\rho^*)$ that belongs to the Barron space. As the scaling factor $\rho(a|w, c)$ can be summed into a function $a(w, c)$, we can define the following Lyapunov function [EMW19c]:

$$F(a_t) = t \cdot L(\rho_t) + \frac{1}{2} ||a_t - a^*||_{L^2(\rho(w,c))}^2$$

where $\rho_t = \delta_{a(w,c)} \rho(w, c)$ and $\rho^* = \delta_{a^*} \rho(w, c)$. Since f^* is assumed to be a Barron function, $||a^*||_{L^2(\rho)}^2$ is finite. It is straightforward to show that $\partial_t F \le 0$, so we obtain the following convergence rate:

$$L(\rho_t) - L(\rho^*) \le \frac{||a_0 - a^*||_{L^2(\rho(w,c))}^2}{2t}$$

Yet, it is not clear how to generalize this argument when we train both layers. One reason the random feature model is easier to work with is that the objective L becomes a convex function on $a \in L^2(\rho(w, c))$, so that a_t can always be joined to the minimizer a^* via a straight line segment in the sublevel set. For 2-layer networks, L is defined on the Wasserstein space. One of the natural ways to define convexity in this setting is via the geodesics that we discussed in Section 2.2: L is called displacement convex if it is convex along the minimizing geodesics in the Wasserstein space [Vil03]. Yet, it seems that in general L is not displacement convex.

There is, however, one theoretical method to circumvent the problem of nonconvexity and local minima, though its practical value may be limited. [MMN18] proposed to inject noise (or diffusion) to

the gradient flow (8) so that intuitively the neurons under stochastic motion can eventually overcome the landscape barriers and discover global minima. Specifically, they considered the following modified objective function (or free energy):

$$L_{\beta,\lambda}(\rho) = L(\rho) + \beta^{-1} \int \rho(\theta) \log\big(\rho(\theta)\big) d\theta + \frac{\lambda}{2} \mathbb{E}_{\rho(\theta)} ||\theta||^2$$

where β^{-1} is the inverse temperature, the second term is the entropy, which encourages ρ to be diffuse, and the third term is a regularization, which prevents the neurons from moving too far away. From the neuron's perspective, this modified gradient flow becomes

$$d\theta_t = -\nabla \frac{\delta L}{\delta \rho}\big(\theta_t; \rho_t\big) + \sqrt{2\beta^{-1}} dW_t - \lambda \theta_t$$

where W_t is the Brownian motion. Then, we have the following result.

Theorem 3.7. (Theorem 4 of [MMN18]) Under technical assumptions, the modified gradient flow ρ_t always converges weakly to the unique minimizer of $L_{\beta,\lambda}$.

[MMN18] remarked that the convergence can take an exponential amount of time in the problem's dimension.

4. Deep Networks

Having discussed 2-layer networks in the preceding sections, let us briefly mention some recent progress in the mean-field analysis of deeper neural networks. The following sections will focus on full-connected multilayer networks and residual networks.

4.1 Multilayer nets

Several studies [Ngu19, SS20, AOY19] have examined the mean-field limit of the classical fully-connected multilayer neural networks, and established results that are parallel to the dynamic theory in Section 2.2. Similar to the 2-layer network with 1 hidden layer, assume that there are L hidden layers: the input x is mapped to each of the N_1 neurons at the first layer, and from each neuron to each of the N_2 neurons at the second layer, and so on, until eventually, all the N_L neurons of the L-th hidden layer are mapped to the output. Just as we have considered each hidden neuron of a 2-layer network as a random flow particle, [AOY19] redefined each particle as a path of weights

$$\vec{\theta} = (\theta_{1,n_1}, \theta_{n_1,n_2}, \dots \theta_{n_{L-1},n_L}, \theta_{n_L,1})$$

that connects the input to the output, and demonstrated that these particles satisfy the propagation of chaos property discussed in Section 2.2. Furthermore, the hidden layers are shown to be independent, such that if we initialize the paths $\vec{\theta}$ independently over the layers:

$$\rho_{t=0}(\vec{\theta}) = \rho_{t=0}^{(0,1)}(\theta_{1,n_1}, \theta_{n_1,n_2}) \cdot \prod_{l=2}^{L-2} \rho_{t=0}^{(i)}(\theta_{n_i,n_{i+1}}) \cdot \rho_{t=0}^{(L-1,L)}(\theta_{n_{L-1},n_L}, \theta_{n_L,1})$$

then this independence holds for later times t. The intuition is that neurons at layer $2 \le l \le L-2$ are influenced by many upstream and downstream neurons and thus the influence becomes deterministic in the infinite limit as a corollary of LLN.

Meanwhile, [SS20] established a global optimality guarantee, similar to Theorem 3.4 above, for 3-layer neural networks (with 2 hidden layers).

Theorem 4.1. Under technical assumptions, given the mean-square regression problem, if the initialization ρ_0 has full support and if ρ_t converges weakly to some ρ^*, then ρ^* is the global minimizer.

Besides these results that concern the dynamic aspect, we do not seem to have results on the static aspect, e.g. functional norms and generalization error. [EMW19c] remarked that very deep (or infinite depth) fully-connected multilayer networks might not have a reasonable continuum limit, which may explain why they suffer from numerical instabilities like exploding and vanishing gradients.

4.2 Residual nets

It seems that residual networks are more amenable to analysis, because their functional form resembles the forward Euler scheme and thus can be examined using tools from ODE and optimal control theory [LML$^+$20], and recent studies [EMW19a, EMW19b, LML$^+$20] have established both static and dynamic results analogous to those from Sections 3.1 and 3.2.

One way to define a continuous limit of the residual network is to take the infinite limit in the number of layers L. Specifically, each layer $1 \le l \le L$ of the finite residual network is given by

$$z_0 = x,\ z_l = z_{l-1} + \frac{1}{L} U_l \cdot \sigma\big(W_l \cdot z_{l-1}\big)$$

As usual, each layer is replaced by an expectation $\mathbb{E}_{\rho_l}$ and the layers l are replaced by continuous time $t \in [0, 1]$:

$$\dot{z}_t = \mathbb{E}_{\rho_t(\theta)}\big[\sigma(z_t; \theta)\big] \tag{12}$$

Regarding the static aspect, [EMW19a, EMW19b] proposed the compositional function norm $||f||_{D_p}$. Similar to the Barron norm, $||f||_{D_p}$ is an infimum over all $\{\rho_t \mid t \in [0, 1]\}$ that generate f by (12). Correspondingly, the compositional function space has its own approximation theorem, inverse approximation theorem, Rademacher complexity bound, and generalization error bound (similar to Theorems 2.1, 2.2, 3.1, 3.2).

Regarding the dynamic aspect, [LML$^+$20] established an optimal convergence guarantee similar to Theorem 3.4. The heuristic is that a residual network can be approximated by flattening out the residual blocks into a 2-layer network, so that the arguments for 2-layer network can be adapted to residual network.

5. Conclusion

The mean-field analysis of neural networks is a relatively nascent field, but it has been proven highly useful and inspirational in our understanding of the training behavior and generalization ability of overparametrized neural networks. One could expect that it is a promising beginning that will eventually lead to a mature theory of neural networks and high-dimensional numerical analysis. This survey covered only a small fraction of the growing literature of the mean-field theory, but still we managed to introduce several useful concepts—mainly, the Barron norm and Wasserstein gradient flow—which lead to satisfactory results on the static and dynamic aspects of overparametrized networks.

Besides the open problem on the convergence of gradient flow (Section 3.3), it could be worthwhile the explore the following topics:

1. Dynamics in the finite data regime: Recall that Theorem 3.2 offers a generalization bound for regularized global minimizers. It would be useful to extend this result to trained neural networks, and in particular, [EMW19c] remarked that we can look for the following form of generalization error bound:
$$R(f_{m,n,t}) \le e(1/m, 1/n, t, ||f||)$$
where $f_{m,n,t}$ is a network with m neurons, trained with n sample points for finite time t.

2. Landscape and convergence: One possible way to study convergence is to understand the landscape of $L(\rho)$ over the Wasserstein space. The analysis could be benefited by helpful ideas from [VBB18, SJL18].

3. Advanced architectures: Having explored the mean-field formulation of simple feedforward networks and residual networks, we can try to adapt the same ideas to CNN, RNN, GAN and Deep Q-learning. As the CLT regime (neural tangent kernels) has been successfully applied to CNN [ADH$^+$19] and Q-learning [SS19], it is promising that these architectures correspond to well-defined continuous objects in the mean-field regime.

4. Design principles: We have seen that taking the continuous limit of neural network reveals which parts of the training rules and architecture designs are essential for good performance, e.g. initializing the parameter distribution with full support, plus overparametrization, guarantees optimal convergence (Theorem 3.5). Similarly, the Wasserstein gradient flow has inspired the accelerated training rule of the neuronal birth-death process (Section 3.2), and the flow-based mean-field formulation of residual networks motivated [LML+20] to train ResNet by permuting its blocks. One could follow this line of reasoning to discover better training schemes and architectures informed by the mean-field theory.

References

[ADH+19] Sanjeev Arora, Simon S Du, Wei Hu, Zhiyuan Li, Russ R Salakhutdinov, and Ruosong Wang. On exact computation with an infinitely wide neural net. In *Advances in Neural Information Processing Systems*, pages 8139–8148, 2019.

[AGS08] Luigi Ambrosio, Nicola Gigli, and Giuseppe Savaré. *Gradient flows: in metric spaces and in the space of probability measures*. Springer Science & Business Media, 2008.

[AOY19] Dyego Araújo, Roberto I Oliveira, and Daniel Yukimura. A mean-field limit for certain deep neural networks. *arXiv preprint arXiv:1906.00193*, 2019.

[AZLL19] Zeyuan Allen-Zhu, Yuanzhi Li, and Yingyu Liang. Learning and generalization in overparameterized neural networks, going beyond two layers. In *Advances in neural information processing systems*, pages 6155–6166, 2019.

[B+15] Sébastien Bubeck et al. Convex optimization: Algorithms and complexity. *Foundations and Trends® in Machine Learning*, 8(3-4):231–357, 2015.

[Bar98] Peter L Bartlett. The sample complexity of pattern classification with neural networks: the size of the weights is more important than the size of the network. *IEEE transactions on Information Theory*, 44(2):525–536, 1998.

[BB00] Jean-David Benamou and Yann Brenier. A computational fluid mechanics solution to the monge-kantorovich mass transfer problem. *Numerische Mathematik*, 84(3):375–393, 2000.

[Bre93] Leo Breiman. Hinging hyperplanes for regression, classification, and function approximation. *IEEE Transactions on Information Theory*, 39(3):999–1013, 1993.

[CB18] Lenaic Chizat and Francis Bach. On the global convergence of gradient descent for over-parameterized models using optimal transport. In *Advances in neural information processing systems*, pages 3036–3046, 2018.

[COB19] Lenaic Chizat, Edouard Oyallon, and Francis Bach. On lazy training in differentiable programming. In *Advances in Neural Information Processing Systems*, pages 2933–2943, 2019.

[DZPS18] Simon S Du, Xiyu Zhai, Barnabas Poczos, and Aarti Singh. Gradient descent provably optimizes over-parameterized neural networks. *arXiv preprint arXiv:1810.02054*, 2018.

[E19] Weinan E. Machine learning: Mathematical theory and scientific applications. *Notices of the American Mathematical Society*, 66(11), 2019.

[EMW18] Weinan E, Chao Ma, and Lei Wu. A priori estimates for two-layer neural networks. *arXiv preprint arXiv:1810.06397*, 2018.

[EMW19a] Weinan E, Chao Ma, and Qingcan Wang. A priori estimates of the population risk for residual networks. *arXiv preprint arXiv:1903.02154*, 1(7), 2019.

[EMW19b] Weinan E, Chao Ma, and Lei Wu. Barron spaces and the compositional function spaces for neural network models. *arXiv preprint arXiv:1906.08039*, 2019.

[EMW19c] Weinan E, Chao Ma, and Lei Wu. Machine learning from a continuous viewpoint. *arXiv preprint arXiv:1912.12777*, 2019.

[Eva10] Lawrence C Evans. *Partial differential equations*, volume 19. American Mathematical Soc., 2010.

[HSW90] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks. *Neural networks*, 3(5):551–560, 1990.

[JGH18] Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in neural information processing systems*, pages 8571–8580, 2018.

[JKO98] Richard Jordan, David Kinderlehrer, and Felix Otto. The variational formulation of the fokker–planck equation. *SIAM journal on mathematical analysis*, 29(1):1–17, 1998.

[Lac18] Daniel Lacker. Mean field games and interacting particle systems. *Preprint*, 2018.

[LBW96] Wee Sun Lee, Peter L Bartlett, and Robert C Williamson. Efficient agnostic learning of neural networks with bounded fan-in. *IEEE Transactions on Information Theory*, 42(6):2118–2132, 1996.

[LL18] Yuanzhi Li and Yingyu Liang. Learning overparameterized neural networks via stochastic gradient descent on structured data. In *Advances in Neural Information Processing Systems*, pages 8157–8166, 2018.

[LML+20] Yiping Lu, Chao Ma, Yulong Lu, Jianfeng Lu, and Lexing Ying. A mean-field analysis of deep resnet and beyond: Towards provable optimization via overparameterization from depth. *arXiv preprint arXiv:2003.05508*, 2020.

[Lov12] László Lovász. *Large networks and graph limits*, volume 60. American Mathematical Soc., 2012.

[MMN18] Song Mei, Andrea Montanari, and Phan-Minh Nguyen. A mean field view of the landscape of two-layer neural networks. *Proceedings of the National Academy of Sciences*, 115(33):E7665–E7671, 2018.

[Ngu19] Phan-Minh Nguyen. Mean field limit of the learning dynamics of multilayer neural networks. *arXiv preprint arXiv:1902.02880*, 2019.

[Pel14] Mark A Peletier. Variational modelling: Energies, gradient flows, and large deviations. *arXiv preprint arXiv:1402.1990*, 2014.

[RHW85] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning internal representations by error propagation. Technical report, California Univ San Diego La Jolla Inst for Cognitive Science, 1985.

[RJBVE19] Grant Rotskoff, Samy Jelassi, Joan Bruna, and Eric Vanden-Eijnden. Global convergence of neuron birth-death dynamics. *arXiv preprint arXiv:1902.01843*, 2019.

[RVE19] Grant M Rotskoff and Eric Vanden-Eijden. Trainability and accuracy of neural networks: An interacting particle system approach. *stat*, 1050:30, 2019.

[San15] Filippo Santambrogio. Optimal transport for applied mathematicians. *Birkäuser, NY*, 55(58-63):94, 2015.

[SJL18] Mahdi Soltanolkotabi, Adel Javanmard, and Jason D Lee. Theoretical insights into the optimization landscape of over-parameterized shallow neural networks. *IEEE Transactions on Information Theory*, 65(2):742–769, 2018.

[SS19] Justin Sirignano and Konstantinos Spiliopoulos. Asymptotics of reinforcement learning with neural networks. *arXiv preprint arXiv:1911.07304*, 2019.

[SS20] Justin Sirignano and Konstantinos Spiliopoulos. Mean field analysis of deep neural networks. *arXiv preprint arXiv:1903.04440*, 2020.

[SSBD14] Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

[Szn91] Alain-Sol Sznitman. Topics in propagation of chaos. In *Ecole d'été de probabilités de Saint-Flour XIX—1989*, pages 165–251. Springer, 1991.

[VBB18] Luca Venturi, Afonso S Bandeira, and Joan Bruna. Spurious valleys in two-layer neural network optimization landscapes. *arXiv preprint arXiv:1802.06384*, 2018.

[Vil03] Cédric Villani. *Topics in optimal transportation*. Number 58. American Mathematical Soc., 2003.

[Vil08] Cédric Villani. *Optimal transport: old and new*, volume 338. Springer Science & Business Media, 2008.

[ZBH+16] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530*, 2016.