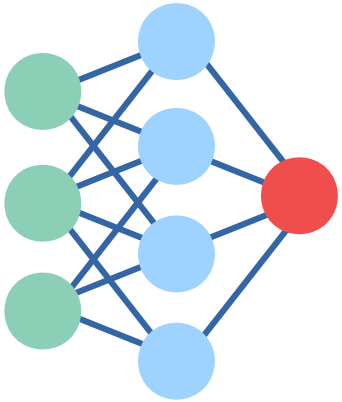
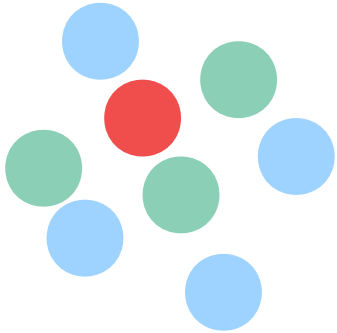
Neural networks as interacting particle systems



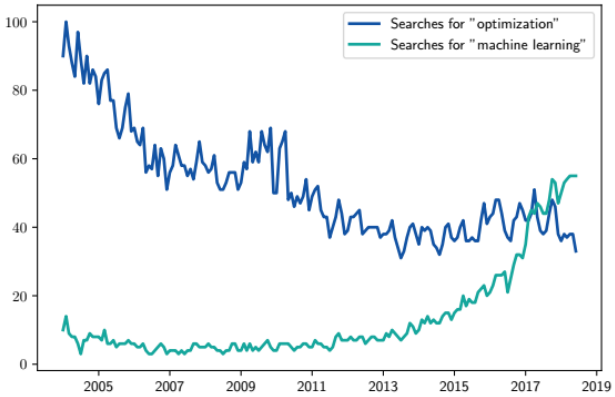
Machine Learning Seminar
Stellenbosch University
19 Oct 2018



Grant M. Rotskoff
Courant Institute of Mathematical Sciences

Work with Eric Vanden-Eijnden
arXiv:1805.00915
NIPS Montreal, 2018

The new "optimization"



[via google trends]

Machine learning is replacing classical optimization in *many* domains:

- Pattern recognition (images)
- Finance
- Physics and Chemistry

Fundamental questions about neural networks remain unanswered...

I will focus on two "fundamental" questions.

Question #1: Can we learn an optimal solution?

- More mathematically... Is there some limit in which we gradient descent based algorithms **converge** to a representation that **minimizes** the loss function.
- Key assumptions
  - Not data limited (e.g., try to approximate a given function)
  - Train as long as necessary (we're interested in the asymptotics)

Question #2: Is an optimal solution unique?

- Empirical observation: Reinitialize neural net parameters randomly, retrain *with identical samples*, training converges to same loss / training error, *but the parameters are different.*
  - Is the optimal representation unique at the level of the parameters?
  - If not, is there any explanation for why?

Are image recognition problems special?

MNIST digits

$f_n(\boldsymbol{x}) : \{0,1\}^{784} \rightarrow \{0,\ldots,9\}$

- Problems assumed to be intractable a decade ago are now routine. Are image / speech recognition inherently low dimensional?
- Alternatively, can neural networks accurately represent truly high-dimensional data?
- Consider representing a *known* function traditional methods (e.g., finite elements) *cannot* be scaled to high-dimensional problems. Grid sizes quickly become intractable.
- Nonlinear functions, like neural networks, avoid this issue, but must be optimized.

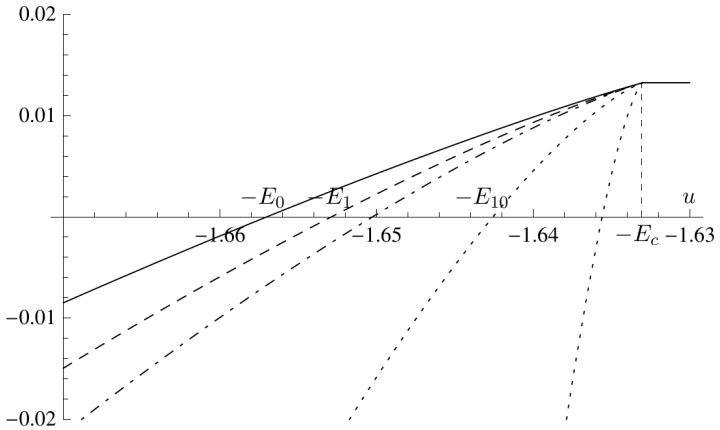
No: Neural networks are expressive

- Universal approximation theorems (Cybenko, Barron)
  - *Any* sufficiently smooth function can be represented arbitrarily well by a very wide single layer neural network!
  - Proof: Abstract functional analysis: neural networks are "dense" in the space of square integrable functions.
  - These theorems **do not** answer Q1 or Q2:
    - How should we get to the optimal parameters?
    - Do typical algorithms (gradient descent, SGD) converge?
    - How bad is the error when the number of neurons is finite?

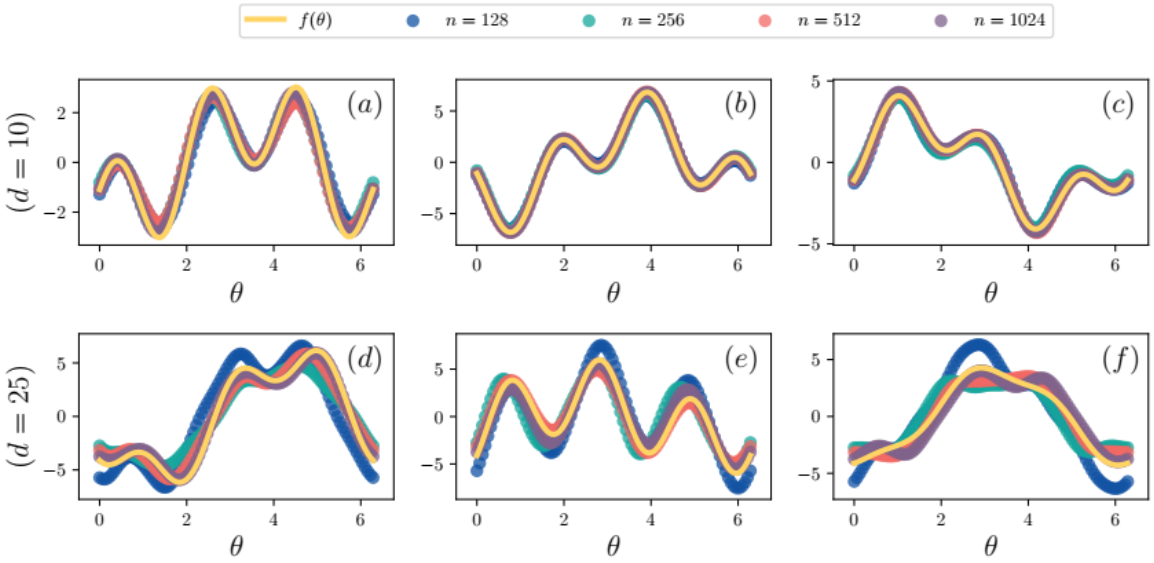
A simple test case (with a lot of "complexity")

$$f(\boldsymbol{x}) = \frac{1}{d} \sum_{i,j,k=1}^{d} a_{ijk} x_i x_j x_k \qquad \boldsymbol{x} \in S^{d-1}(\sqrt{d})$$

$$a_{ijk} \in \mathbb{R} \sim \mathcal{N}(0,1)$$



Spherical 3-spin model
Cf. Ben Arous, others



$$\boldsymbol{x} = (\sin\theta, \cos\theta, 0, \ldots, 0)$$

Training a neural network

1) Initialize "network" f_n : data $\to$ output

2) Compute "loss" by sampling a training set

3) Iterate until "convergence" with stochastic gradient descent

Single layer "sigmoid" neural net

$$f_n(\boldsymbol{x}) = \sum_{i=1}^{n} h(\boldsymbol{a}_i \cdot x + b_i) + c$$

$$h(x) = 1/(1 + e^{-x})$$

Mean squared loss function

$$\ell(f, f_n) = \tfrac{1}{2} \int_\Omega |f(\boldsymbol{x}) - f_n(\boldsymbol{x})|^2 \, d\mu(\boldsymbol{x})$$

$$\frac{1}{2}\mathbb{E}_{x \sim \mathbb{P}_{\text{data}}} \left(f(\boldsymbol{x}) - f_n(\boldsymbol{x})\right)^2$$

Stochastic gradient descent

$$\boldsymbol{z}(t + \Delta t) = \boldsymbol{z}(t) + \Delta t \nabla_{\boldsymbol{z}} \ell_P(\boldsymbol{z})$$

$$\ell_P(\boldsymbol{z}) = \frac{1}{2P} \sum_{i=1}^{P} \left(f(\boldsymbol{x}_i) - f_n(\boldsymbol{x}_i)\right)^2$$

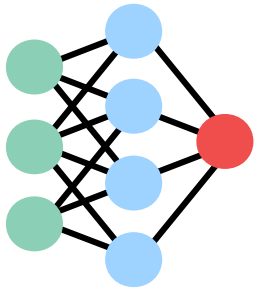
Stochastic gradient descent, step-by-step

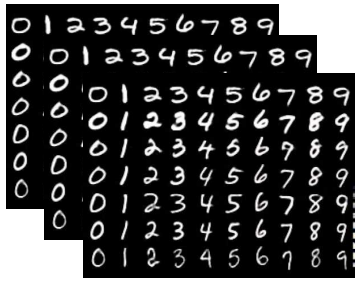
1. Get set of labeled data (e.g., images with their classification) or "training set". Initialize network with random parameters.

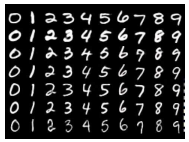
2. Randomly sample a "batch" of data, a small number of points.

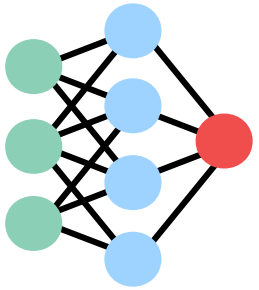
3. Compute the gradient of the objective function on this batch of points (the estimate is noisy, but unbiased).

4. Iterate this batch optimization procedure for many steps, passing over the data multiple times, until convergence.









Compute discrepancy or "loss"

Compute gradients

Formalizing the optimization procedure

Write the representation as $f_n(\boldsymbol{x}) = \frac{1}{n}\sum_{i=1}^{n} c_i \varphi(\boldsymbol{x}, \boldsymbol{y}_i)$

- **Radial basis function networks.** $D \subset \Omega$ and $\varphi(\boldsymbol{x}, \boldsymbol{y}) \equiv \phi(\boldsymbol{x} - \boldsymbol{y})$ where ϕ is a radial function

$$\phi(\boldsymbol{x}) = \exp\left(-\tfrac{1}{2}\kappa|\boldsymbol{x}|^2\right)$$

  where $\kappa > 0$ is a fixed constant.

- **Single-layer neural networks.** $D \subset \mathbb{R}^{d+1}$ and $\varphi(\boldsymbol{x}, \boldsymbol{y}) = \varphi(\boldsymbol{x}, \boldsymbol{a}, b)$ with $\boldsymbol{a} \in \mathbb{R}^d$, $b \in \mathbb{R}$, and

$$\varphi(\boldsymbol{x}, \boldsymbol{a}, b) = h(\boldsymbol{a} \cdot \boldsymbol{x} + b)$$

  where $h : \mathbb{R} \to \mathbb{R}$ is e.g. a sigmoid function $h(z) = 1/(1 + e^{-z})$.

Formalizing the optimization procedure

The loss function is $\ell(f, f_n) = \frac{1}{2} \int_\Omega |f(\boldsymbol{x}) - f_n(\boldsymbol{x})|^2 \, d\mu(\boldsymbol{x})$

Which we can expand as $\ell(f, f_n) = C_f - \frac{1}{n} \sum_{i=1}^{n} c_i F(\boldsymbol{y}_i) + \frac{1}{2n^2} \sum_{i,j=1}^{n} c_i c_j K(\boldsymbol{y}_i, \boldsymbol{y}_j)$

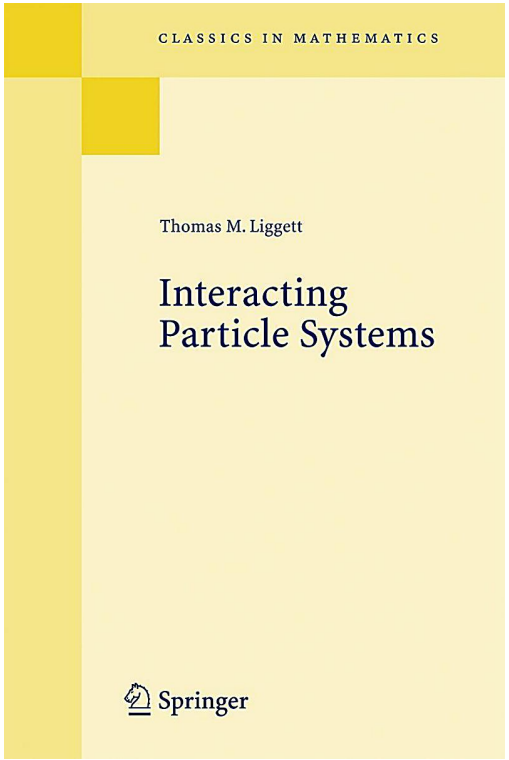
Using $f_n(\boldsymbol{x}) = \frac{1}{n} \sum_{i=1}^{n} c_i \varphi(\boldsymbol{x}, \boldsymbol{y}_i)$, we define,

$$F(\boldsymbol{y}) = \int_\Omega f(\boldsymbol{x}) \varphi(\boldsymbol{x}, \boldsymbol{y}) d\mu(\boldsymbol{x}) \qquad K(\boldsymbol{y}, \boldsymbol{z}) = \int_\Omega \varphi(\boldsymbol{x}, \boldsymbol{y}) \varphi(\boldsymbol{x}, \boldsymbol{z}) d\mu(\boldsymbol{x})$$

This a complicated system...

... with complicated interactions

Luckily, systems of this type have been studied extensively in mathematical physics



Interpretation as an interacting particle system

The loss function is $\ell(f, f_n) = \frac{1}{2} \int_\Omega |f(\boldsymbol{x}) - f_n(\boldsymbol{x})|^2 \, d\mu(\boldsymbol{x})$

Energy function

Which we can expand as $\ell(f, f_n) = C_f - \frac{1}{n} \sum_{i=1}^{n} c_i F(\boldsymbol{y}_i) + \frac{1}{2n^2} \sum_{i,j=1}^{n} c_i c_j K(\boldsymbol{y}_i, \boldsymbol{y}_j)$

charges particles

Using $f_n(\boldsymbol{x}) = \frac{1}{n} \sum_{i=1}^{n} c_i \varphi(\boldsymbol{x}, \boldsymbol{y}_i)$, we define,

$$F(\boldsymbol{y}) = \int_\Omega f(\boldsymbol{x}) \varphi(\boldsymbol{x}, \boldsymbol{y}) d\mu(\boldsymbol{x}) \qquad K(\boldsymbol{y}, \boldsymbol{z}) = \int_\Omega \varphi(\boldsymbol{x}, \boldsymbol{y}) \varphi(\boldsymbol{x}, \boldsymbol{z}) d\mu(\boldsymbol{x})$$

Single body potential Interaction potential

Nonequilibrium dynamics of optimization

The neural net representation

$$f_n(t, \boldsymbol{x}) = \frac{1}{n} \sum_{i=1}^{n} C_i(t) \varphi(\boldsymbol{x}, \boldsymbol{Y}_i(t))$$

Evolves according to gradient descent on a complex landscape,

$$\begin{cases} \dot{\boldsymbol{Y}}_i = C_i \nabla F(\boldsymbol{Y}_i) - \dfrac{1}{n} \displaystyle\sum_{j=1}^{n} C_i C_j \nabla K(\boldsymbol{Y}_i, \boldsymbol{Y}_j), \\ \dot{C}_i = F(\boldsymbol{Y}_i) - \dfrac{1}{n} \displaystyle\sum_{j=1}^{n} C_j K(\boldsymbol{Y}_i, \boldsymbol{Y}_j) \end{cases}$$

We can extend to both Langevin dynamics, stochastic gradient descent

Hydrodynamic approach: parameter density

Work with the particle density:

$$\rho_n(t, \boldsymbol{y}, c) = \frac{1}{n} \sum_{i=1}^{n} \delta(c - C_i(t)) \delta(\boldsymbol{y} - \boldsymbol{Y}_i(t))$$

So that the representation satisfies

$$f_n(t, \boldsymbol{x}) = \frac{1}{n} \sum_{i=1}^{n} C_i(t) \varphi(\boldsymbol{x}, \boldsymbol{Y}_i(t)) = \int_{D \times R} c \varphi(\boldsymbol{x}, \boldsymbol{y}) \rho_n(t, \boldsymbol{y}, c) d\boldsymbol{y} dc$$

To study the **training dynamics** we can
study the **time evolution of the density**

A partial differential equation for the density

We know the time derivatives of $\boldsymbol{Y}$ and C

To compute $\partial_t \rho$ we just use the chain rule, and we get a PDE

$$\begin{aligned}\partial_t \rho_n = \nabla \cdot \left(-c \nabla F \rho_n + \int_{D \times \mathbb{R}} c c' \nabla K(\boldsymbol{y}, \boldsymbol{y}') \rho'_n \rho_n d\boldsymbol{y}' dc' \right) \\ + \partial_c \left(-F \rho_n + \int_{D \times \mathbb{R}} c' K(\boldsymbol{y}, \boldsymbol{y}') \rho'_n \rho_n d\boldsymbol{y}' dc' \right)\end{aligned}$$

A partial differential equation for the density

We know the time derivatives of $\boldsymbol{Y}$ and C

To compute $\partial_t \rho$ we just use the chain rule, and we get a PDE

$$\partial_t \rho_n = \nabla \cdot \left(-c \nabla \boxed{Q2:\surd}(\boldsymbol{y}, \boldsymbol{y}') \rho'_n \rho_n d\boldsymbol{y}' dc' \right)$$
$$+ \partial_c \left(-F \rho_n + \int_{D \times \mathbb{R}} c' K(\boldsymbol{y}, \boldsymbol{y}') \rho'_n \rho_n d\boldsymbol{y}' dc' \right)$$

(if we can show the PDE converges to the optimum)

The mean-field limit (i.e., O(1) term)

In the limit $n \to \infty$

$$\partial_t \rho_0 = \nabla \cdot \left(-c \nabla F \rho_0 + \int_{D \times \mathbb{R}} c c' \nabla K(\boldsymbol{y}, \boldsymbol{y}') \rho_0' \rho_0 d\boldsymbol{y}' dc' \right) + \partial_c \left(-F \rho_0 + \int_{D \times \mathbb{R}} c' K(\boldsymbol{y}, \boldsymbol{y}') \rho_0' \rho_0 d\boldsymbol{y}' dc' \right)$$

now with smooth initial conditions

Which can be written,

$$\partial_t \rho_0 = \nabla \cdot \left(\rho_0 \nabla \frac{\delta \mathcal{E}_0}{\delta \rho_0} \right) + \partial_c \left(\rho_0 \partial_c \frac{\delta \mathcal{E}_0}{\delta \rho_0} \right)$$

Asymptotic convexity $\overset{?}{\iff}$ Asymptotic convergence

$$\partial_t \rho_0 = \nabla \cdot \left(\rho_0 \nabla \frac{\delta \mathcal{E}_0}{\delta \rho_0} \right) + \partial_c \left(\rho_0 \partial_c \frac{\delta \mathcal{E}_0}{\delta \rho_0} \right)$$

$$\mathcal{E}_0[\rho_0] = C_f - \int_{D \times \mathbb{R}} cF\rho_0 d\boldsymbol{y} dc + \frac{1}{2} \int_{(D \times \mathbb{R})^2} cc' K(\boldsymbol{y}, \boldsymbol{y}') \rho_0 \rho_0' d\boldsymbol{y} dc d\boldsymbol{y}' dc'$$

$$= \frac{1}{2} \int_\Omega \left(f(\boldsymbol{x}) - \int_{D \times \mathbb{R}} c\varphi(\boldsymbol{x}, \boldsymbol{y}) \rho_0 d\boldsymbol{y} dc \right)^2 d\mu(\boldsymbol{x}) \geq 0$$

- Fixed points of this equation, written as above, can occur if the density singular
- Not all fixed points are energy minimizers
- Continuous dynamics in the charges changes this story

Theorem (Law of Large Numbers) for convergence

Let $f_n(t) = f_n(t, \boldsymbol{x})$ with $\{\boldsymbol{Y}_i(t), C_i(t)\}_{i=1}^n$ evolve according to gradient descent with initial condition drawn from some $\mathbb{P}_{\text{in}}$. Then

$$\lim_{n\to\infty} f_n(t) = f_0(t) \qquad \mathbb{P}_{\text{in}}\text{-almost surely} \tag{1}$$

where $f_0(t)$ solves the differential equation below and satisfies

$$\lim_{t\to\infty} f_0(t) = f \quad \text{a.e. in} \quad \Omega \tag{2}$$

See also: Mei, Montanari, Pham and Sirignano, Spiliopolous

Evolution equation for the representation is inherited from evolution of density,

$$\partial_t f_0(t, \boldsymbol{x}) = \int_{D\times\mathbb{R}} c\varphi(\boldsymbol{x}, \boldsymbol{y})\partial_t \rho_0(t, \boldsymbol{y}, c) d\boldsymbol{y} dc$$

Theorem (Law of Large Numbers) for convergence

Let $f_n(t) = f_n(t, \boldsymbol{x})$ with $\{\boldsymbol{Y}_i(t), C_i(t)\}_{i=1}^n$ evolve according to gradient descent with initial condition drawn from some $\mathbb{P}_{\text{in}}$. Then

$$\lim_{n\to\infty} f_n(t) \text{ [illegible] st surely} \tag{1}$$

Q1: √

where $f_0(t)$ solves the differen[illegible]nd satisfies

$$\lim_{t\to\infty} f_0(t) = f \quad \text{a.e. in} \quad \Omega \tag{2}$$

See also: Mei, Montanari, Pham and Sirignano, Spiliopolous

Evolution equation for the representation is inherited from evolution of density,

$$\partial_t f_0(t, \boldsymbol{x}) = \int_{D\times\mathbb{R}} c\varphi(\boldsymbol{x}, \boldsymbol{y})\partial_t \rho_0(t, \boldsymbol{y}, c) d\boldsymbol{y} dc$$

Theorem: the error term is at order $1/n$

Let $f_n(t) = f_n(t, \boldsymbol{x})$ and with $\{\boldsymbol{Y}_i(t), C_i(t)\}_{i=1}^n$ evolve according to gradient descent with initial condition drawn from $\mathbb{P}_{\text{in}}$. Then for any $\bar{\xi} < 1$ and any $a_n > 0$ such that $a_n / \log n \to \infty$ as $n \to \infty$, we have

$$\lim_{n \to \infty} n^{\bar{\xi}} \left(f_n(a_n) - f_0(a_n) \right) = 0 \qquad \text{almost surely}$$

where satisfies $f_0(t) \to f$ as $t \to \infty$.

Initial scaling from CLT $\xi = 1/2$ ⟶ Asymptotic scaling after optimization $\xi = 1$

Typically, we don't know F or K

Solution: Stochastic samples of the potential / interaction term:

$$F_P(t, \boldsymbol{y}) = \frac{1}{P}\sum_{p=1}^{P} f(\boldsymbol{X}_p(t))\varphi(\boldsymbol{X}_p(t), \boldsymbol{y}), \qquad K_P(t, \boldsymbol{y}, \boldsymbol{y}') = \frac{1}{P}\sum_{p=1}^{P} \varphi(\boldsymbol{X}_p(t), \boldsymbol{y})\varphi(\boldsymbol{X}_p(t), \boldsymbol{y}')$$

Leading to an SDE,

$$d\boldsymbol{Z} = \nabla_{\boldsymbol{z}} \ell(f, f_n(\boldsymbol{z}))dt + \sqrt{\theta} d\boldsymbol{B}$$

Where the quadratic variation of the noise is

$$\mathbb{E}\left(\nabla_{\boldsymbol{z}}(L_P(\boldsymbol{z}) - n\ell(f, f_n(\boldsymbol{z})))\right) \otimes \left(\nabla_{\boldsymbol{z}}(L_P(\boldsymbol{z}') - n\ell(f, f_n(\boldsymbol{z}'))\right) = \frac{1}{P} R(\boldsymbol{z})$$

$$\theta = \Delta t / P = \text{“timestep/batch size”}$$

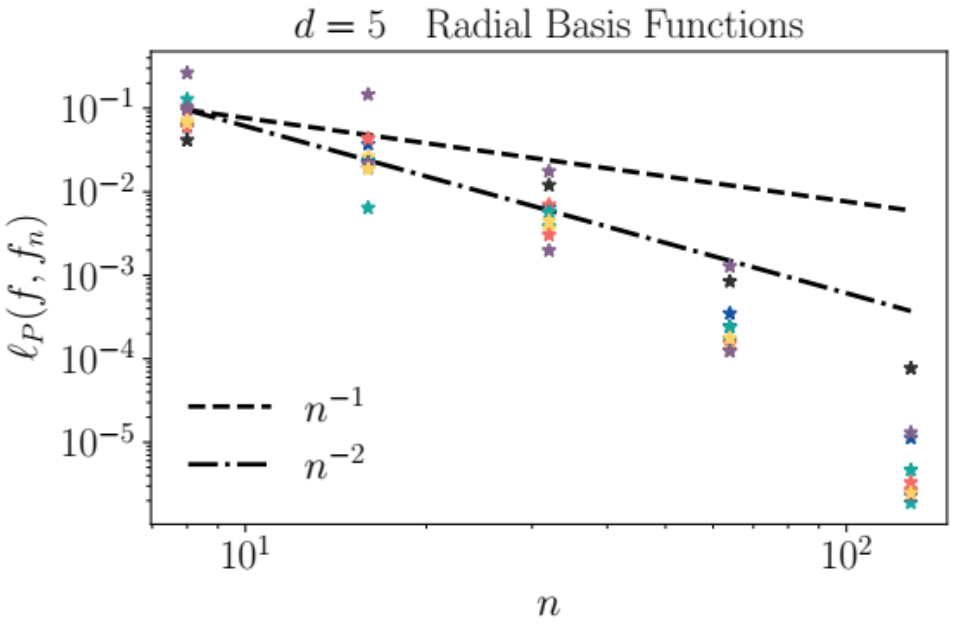
Dean's equation for stochastic gradient descent

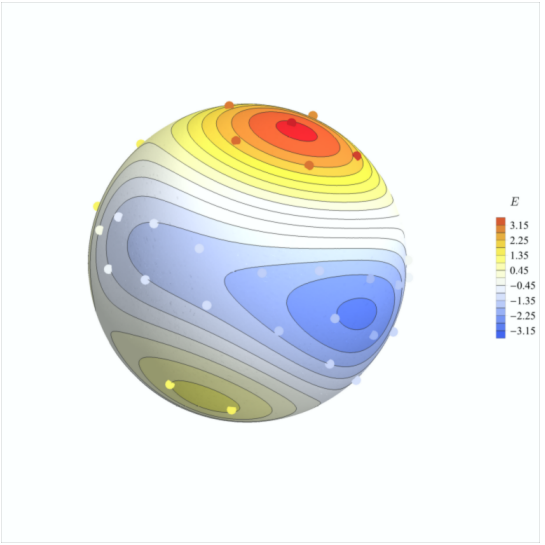
$$\partial_t \rho_n = \nabla \cdot \left(-c \nabla F \rho_n + \int_{D \times \mathbb{R}} c c' \nabla K(\boldsymbol{y}, \boldsymbol{y}') \rho_n' \rho_n d\boldsymbol{y}' dc' \right)$$
$$+ \partial_c \left(-F \rho_n + \int_{D \times \mathbb{R}} c' K(\boldsymbol{y}, \boldsymbol{y}') \rho_n' \rho_n d\boldsymbol{y}' dc' \right)$$
$$+ \tfrac{1}{2}\theta \nabla\nabla : \left(\rho_n c^2 A_2([f_n(t) - f], \boldsymbol{y}, \boldsymbol{y}) \right) + \tfrac{1}{2}\theta \partial_c^2 \left(\rho_n A_0([f_n(t) - f], \boldsymbol{y}, \boldsymbol{y}) \right)$$
$$+ \theta \partial_c \nabla \cdot \left(\rho_n c A_1([f_n(t) - f], \boldsymbol{y}, \boldsymbol{y}) \right)$$
$$+ \sqrt{\theta}\, \dot{\eta}_n(t, \boldsymbol{y}, c)$$

$P = n^2 \implies$

- Same first order term as gradient descent
- Guarantee of convergence
- Recover the error scaling

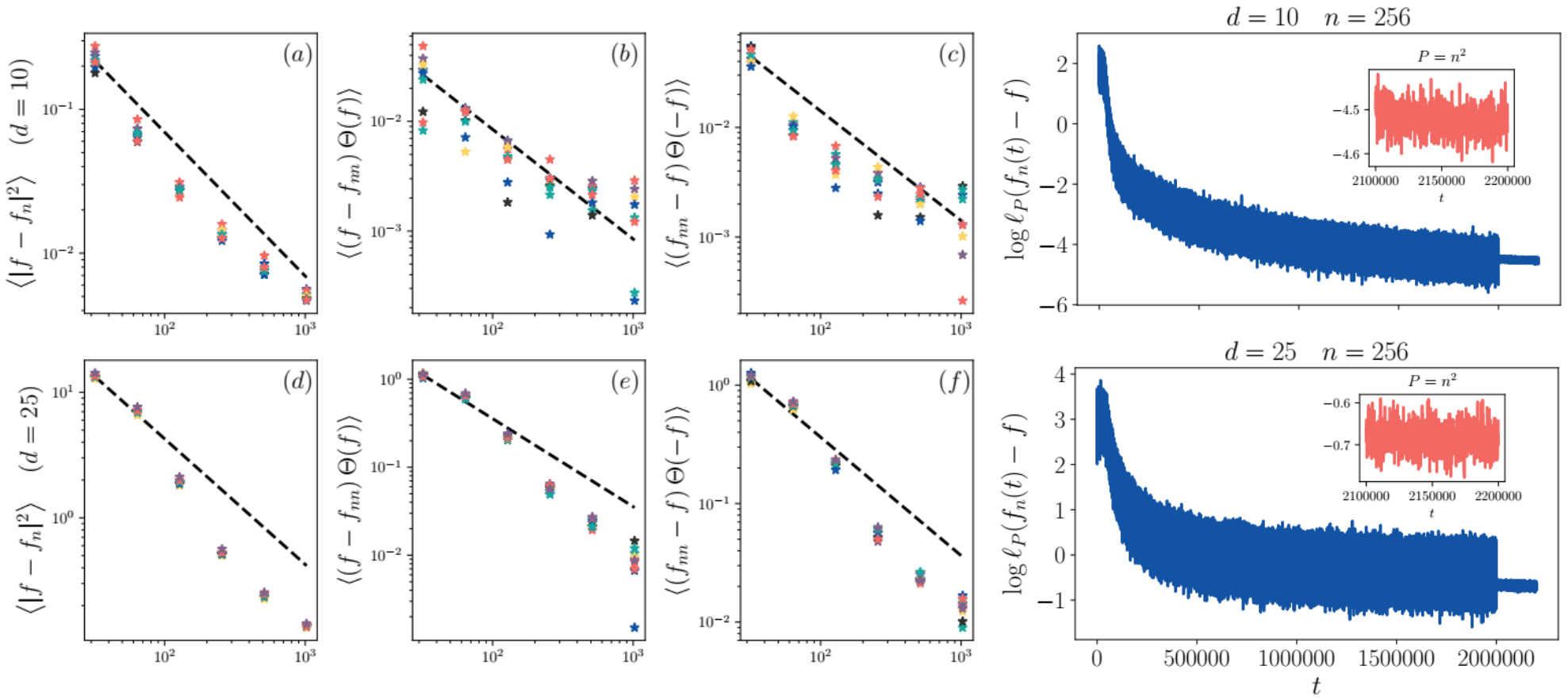
Experiments illuminate the role of charges





$$\phi(\boldsymbol{x}) = \exp\left(-\tfrac{1}{2}\kappa|\boldsymbol{x}|^2\right)$$

Confirming the $1/n$ error scaling



Conclusions

- Neural networks are a potentially powerful tool for computational physics and applied mathematics. They can massively reduce the cost of representing functions in high dimensional spaces.

- By interpreting the parameters as interacting particles, we can demonstrate the asymptotic convexity of the loss landscape, in the process showing that stochastic gradient descent converges to an energy minimizer, with an appropriate quench.

- The error and its scaling can be identified up to a constant, which shows that errors can be controlled precisely in the limit.

- Applications to free energy methods, quantum variational energy calculations, and PDEs are only beginning to be explored.

Statistics > Machine Learning

Neural Networks as Interacting Particle Systems: Asymptotic Convexity of the Loss Landscape and Universal Scaling of the Approximation Error

Grant M. Rotskoff, Eric Vanden-Eijnden

(Submitted on 2 May 2018 (v1), last revised 22 May 2018 (this version, v2))

Neural networks, a central tool in machine learning, have demonstrated remarkable, high fidelity performance on image recognition and classification tasks. These successes evince an ability to accurately represent high dimensional functions, potentially of great use in computational and applied mathematics. That said, there are few rigorous results about the representation error and trainability of neural networks. Here we characterize both the error and the scaling of the error with the size of the network by reinterpreting the standard optimization algorithm used in machine learning applications, stochastic gradient descent, as the evolution of a particle system with interactions governed by a potential related to the objective or "loss" function used to train the network. We show that, when the number n of parameters is large, the empirical distribution of the particles descends on a convex landscape towards a minimizer at a rate independent of n. We establish a Law of Large Numbers and a Central Limit Theorem for the empirical distribution, which together show that the approximation error of the network universally scales as $O(n^{-1})$. Remarkably, these properties do not depend on the dimensionality of the domain of the function that we seek to represent. Our analysis also quantifies the scale and nature of the noise introduced by stochastic gradient descent and provides guidelines for the step size and batch size to use when training a neural network. We illustrate our findings on examples in which we train neural network to learn the energy function of the continuous 3-spin model on the sphere. The approximation error scales as our analysis predicts in as high a dimension as $d = 25$.

Subjects: **Machine Learning (stat.ML)**; Statistical Mechanics (cond-mat.stat-mech); Learning (cs.LG)
Cite as: **arXiv:1805.00915 [stat.ML]**
(or **arXiv:1805.00915v2 [stat.ML]** for this version)

Submission history

From: Grant Rotskoff [view email]
[v1] Wed, 2 May 2018 17:23:42 GMT (1115kb,D)
[v2] Tue, 22 May 2018 15:03:44 GMT (1129kb,D)

Statistics > Machine Learning

A Mean Field View of the Landscape of Two-Layers Neural Networks

Song Mei, Andrea Montanari, Phan-Minh Nguyen

(Submitted on 18 Apr 2018)

Multi-layer neural networks are among the most powerful models in machine learning, yet the fundamental reasons for this success defy mathematical understanding. Learning a neural network requires to optimize a non-convex high-dimensional objective (risk function), a problem which is usually attacked using stochastic gradient descent (SGD). Does SGD converge to a global optimum of the risk or only to a local optimum? In the first case, does this happen because local minima are absent, or because SGD somehow avoids them? In the second, why do local minima reached by SGD have good generalization properties?

In this paper we consider a simple case, namely two-layers neural networks, and prove that -in a suitable scaling limit- SGD dynamics is captured by a certain non-linear partial differential equation (PDE) that we call distributional dynamics (DD). We then consider several specific examples, and show how DD can be used to prove convergence of SGD to networks with nearlyideal generalization error. This description allows to 'average-out' some of the complexities of the landscape of neural networks, and can be used to prove a general convergence result for noisy SGD.

Comments: 101 pages
Subjects: **Machine Learning (stat.ML)**; Statistical Mechanics (cond-mat.stat-mech); Learning (cs.LG); Statistics Theory (math.ST)
Cite as: **arXiv:1804.06561 [stat.ML]**
(or **arXiv:1804.06561v1 [stat.ML]** for this version)

Submission history

From: Song Mei [view email]
[v1] Wed, 18 Apr 2018 05:31:45 GMT (1533kb)

Mathematics > Probability

Mean Field Analysis of Neural Networks

Justin Sirignano, Konstantinos Spiliopoulos

(Submitted on 2 May 2018)

Machine learning has revolutionized fields such as image, text, and speech recognition. There's also growing interest in applying machine and deep learning ideas in engineering, robotics, biotechnology, and finance. Despite their immense success in practice, there is limited mathematical understanding of neural networks. We mathematically study neural networks in the asymptotic regime of simultaneously (A) large network sizes and (B) large numbers of stochastic gradient descent training iterations. We rigorously prove that the empirical distribution of the neural network parameters converges to the solution of a nonlinear partial differential equation. This result can be considered a law of large numbers for neural networks. In addition, a consequence of our analysis is that the trained parameters of the neural network asymptotically become independent, a property which is commonly called "propagation of chaos".

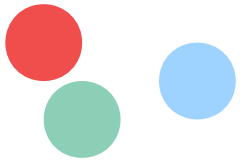
Subjects: **Probability (math.PR)**
Cite as: **arXiv:1805.01053 [math.PR]**
(or **arXiv:1805.01053v1 [math.PR]** for this version)

Submission history

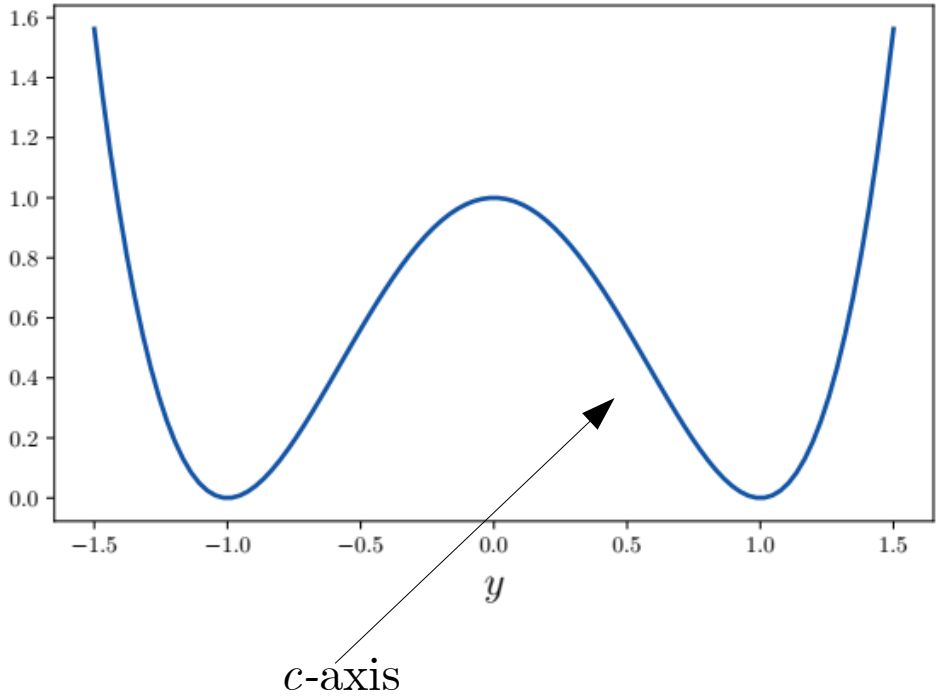
From: Justin Sirignano [view email]
[v1] Wed, 2 May 2018 23:28:59 GMT (598kb,D)

Avoiding the non-minimizing fixed points

We can always lower the energy by increasing the support of the density, the charges are a crucial piece of this consideration.



Charges flip the sign of the potential F



Continuous formulation (taking n large)

We rewrite the representation as $f_n \to \tilde{f} = \int_D \varphi(\cdot, \boldsymbol{y}) G(\boldsymbol{y}) d\boldsymbol{y}$ as $n \to \infty$

With a charge weighted density,

$$G(\boldsymbol{y})d\boldsymbol{y} \to \frac{1}{n}\sum_{i=1}^{n} c_i \delta(\boldsymbol{y} - \boldsymbol{y}_i) d\boldsymbol{y}$$

We can now view the kernel as an operator on functions

$$\varphi G = \int_D \varphi(\cdot, \boldsymbol{y}) G(\boldsymbol{y}) d\boldsymbol{y}, \qquad \varphi^\dagger g = \int_\Omega g(\boldsymbol{x}) \varphi(\boldsymbol{x}, \cdot) d\mu(\boldsymbol{x})$$

So that we want to solve,

$$f(\boldsymbol{x}) = \int_D \varphi(\cdot, \boldsymbol{y}) G(\boldsymbol{y}) d\boldsymbol{y} \implies F = KG \quad \text{(Euler-Lagrange Eqn for the loss)}$$

Beyond abstract guarantees?

Universal Approximation Theorems (Barron, Cybenko, Park, others)

For a suitable kernel φ, given any $f \in L^2(\Omega, \mu)$ and $\epsilon > 0$, there exists $f^* \in \operatorname{ran} \varphi \cup \operatorname{ran} \varphi^\dagger$ which is such that

$$\|f - f^*\|_{L^2(\Omega,\mu)} \leq \epsilon \tag{1}$$

and admits the representation

$$\varphi G^* = f^* \quad \text{a.e. in } \Omega \tag{2}$$

where G^* solves

$$F^* = KG^* \qquad \text{with } F^* = \varphi^\dagger f^* \tag{3}$$