Theories of Neural Networks Training

Challenges and Recent Results

Lénaïc Chizat*, joint work with Francis Bach+

Feb. 19th 2019 - Dep. CSA of IISc - Bangalore

*CNRS and Université Paris-Sud +INRIA and ENS Paris

Introduction

Setting

Supervised machine learning

- given input/output training data $(x^{(1)}, y^{(1)}), \dots, (x^{(n)}, y^{(n)})$
- build a function f such that $f(x) \approx y$ for unseen data (x, y)

Gradient-based learning

- choose a parametric class of functions $f(w, \cdot) : x \mapsto f(w, x)$
- a loss ℓ to compare outputs: squared, logistic, cross-entropy...
- starting from some w_0, update parameters using gradients

Example: Stochastic Gradient Descent with step-sizes $(\eta^{(k)})_{k \geq 1}$

$$w^{(k)} = w^{(k-1)} - \eta^{(k)} \nabla_w [\ell(f(w^{(k-1)}, x^{(k)}), y^{(k)})]$$

[Refs]:
Robbins, Monroe (1951). *A Stochastic Approximation Method.*
LeCun, Bottou, Bengio, Haffner (1998). *Gradient-Based Learning Applied to Document Recognition.*

Models

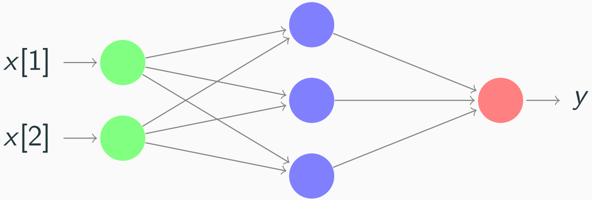
Linear: linear regression, ad-hoc features, kernel methods:

$$f(w, x) = w \cdot \phi(x)$$

Non-linear: neural networks (NNs). Example of a vanilla NN:

$$f(w, x) = W_L^T \sigma(W_{L-1}^T \sigma(\dots \sigma(W_1^T x + b_1) \dots) + b_{L-1}) + b_L$$

with activation σ and parameters $w = (W_1, b_1), \dots, (W_L, b_L)$.



Challenges for Theory

Need for new theoretical approaches

- optimization: non-convex, many interacting parameters
- statistics: over-parameterized, implicit regularization

Why should we care?

- effects of hyper-parameters
- insights on individual tools in a pipeline
- more robust, more efficient, more accessible models

Today's program

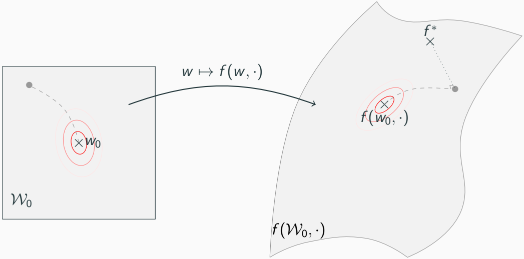
- lazy training
- global convergence for over-parameterized two-layers NNs

[Refs]:
Zhang, Bengio, Hardt, Recht, Vinyals (2016). *Understanding Deep Learning Requires Rethinking Generalization*.

Lazy Training

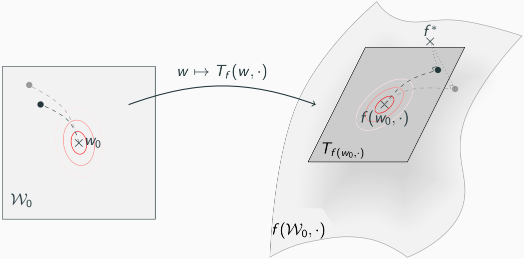
Tangent Model

Let $f(w, x)$ be a differentiable model and w_0 an initialization.



Tangent Model

Let $f(w, x)$ be a differentiable model and w_0 an initialization.



Tangent model

$$T_f(w, x) = f(w_0, x) + (w - w_0) \cdot \nabla_w f(w_0, x)$$

Scaling the output by α makes the linearization more accurate.

Lazy Training Theorem

Theorem (Lazy training through rescaling)

Assume that $f(w_0, \cdot) = 0$ and that the loss is quadratic. In the limit of a small step-size and a large scale α, gradient-based methods on the non-linear model αf and on the tangent model T_f learn the same model, up to a $O(1/\alpha)$ remainder.

- properties of linear models very well understood
- *lazy* because parameters hardly move

[Refs]:

Jacot, Gabriel, Hongler (2018). *Neural Tangent Kernel: Convergence and Generalization in Neural Networks*.
Du, Lee, Li, Wang, Zhai (2018). *Gradient Descent Finds Global Minima of Deep Neural Networks*.
Allen-Zhu, Li, Liang (2018). *Learning and Generalization in Overparameterized Neural Networks [...]*.
Chizat, Bach (2018). *A Note on Lazy Training in Supervised Differentiable Programming*.

Range of Lazy Training

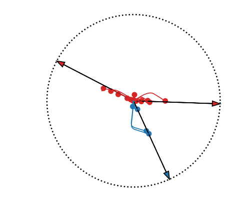
Criteria for lazy training (informal)

$$\underbrace{\| T_f(w^*, \cdot) - f(w_0, \cdot) \|}_{\text{Distance to best linear model}} \qquad \ll \qquad \underbrace{\frac{\|\nabla f(w_0, \cdot)\|^2}{\|\nabla^2 f(w_0, \cdot)\|}}_{\text{“Flatness” around initialization}}$$

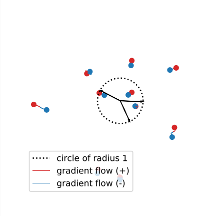
Examples

- *Homogeneous models.*
  If for $\lambda > 0$, $f(\lambda w, x) = \lambda^p f(w, x)$ then flatness $\sim \|w_0\|^p$
- *NNs with large layers.*
  Occurs if initialized with scale $O(1/\sqrt{\text{fan\_in}})$

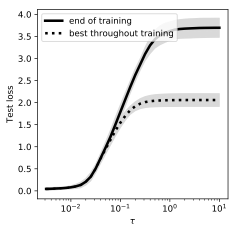
Numerical Illustrations



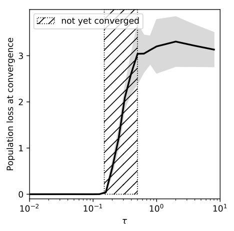
(a) Active



(b) Lazy



(c) Over-param.



(d) Under-param.

Training a 2-layers ReLU NN in the teacher-student setting
(a-b) trajectories (c-d) generalization in 100-d vs init. scale τ

Lessons to be drawn

For practice

- our guess: instead, feature selection is why NNs work
- ongoing investigation on hard tasks

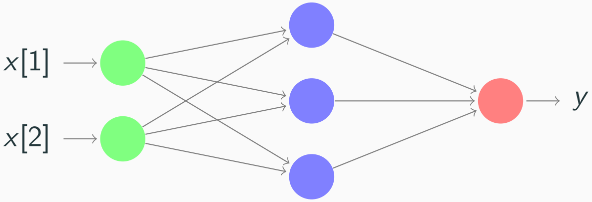
For theory

- in depth analysis sometimes possible
- not just one theory for NNs training

[Refs]:
Zhang, Bengio, Singer (2019). *Are all layers created equal?*
Lee, Bahri, Novak, Schoenholz, Pennington, Sohl-Dickstein (2018). *Deep neural networks as gaussian processes*

Global convergence for 2-layers NNs

Two Layers NNs



With activation σ, define $\phi(w_i, x) = c_i \sigma(a_i \cdot x + b_i)$ and

$$f(w, x) = \frac{1}{m} \sum_{i=1}^{m} \phi(w_i, x)$$

Statistical setting: minimize population loss $\mathbb{E}_{(x,y)}[\ell(f(w, x), y)]$.

Hard problem: existence of spurious minima even with slight over-parameterization and good initialization

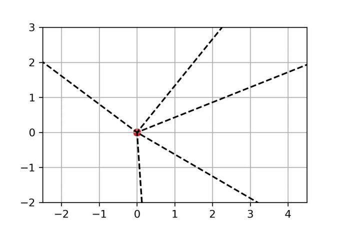
[Refs]:
Livni, Shalev-Shwartz, Shamir (2014). *On the Computational Efficiency of Training Neural Networks*.
Safran, Shamir (2018). *Spurious Local Minima are Common in Two-layer ReLU Neural Networks*.

Mean-Field Analysis

Many-particle limit

Training dynamics in the small step-size and infinite width limit:

$$\mu_{t,m} = \frac{1}{m}\sum_{i=1}^{m} \delta_{w_i(t)} \underset{m\to\infty}{\to} \mu_{t,\infty}$$



[Refs]:
Mei, Montanari, Nguyen (2018). *A Mean Field View of the Landscape of Two-Layers Neural Networks.*
Rotskoff, Vanden-Eijndem (2018). *Parameters as Interacting Particles [...].*
Sirignano, Spiliopoulos (2018). *Mean Field Analysis of Neural Networks.*
Chizat, Bach (2018) *On the Global Convergence of Gradient Descent for Over-parameterized Models [...]*

Global Convergence

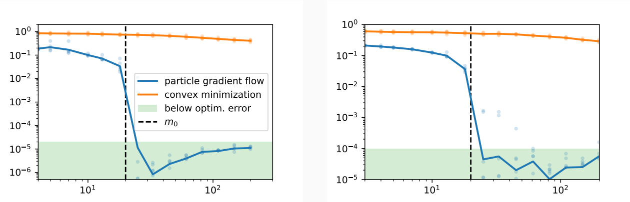
Theorem (Global convergence, informal)

In the limit of a small step-size, a large data set and large hidden layer, NNs trained with gradient-based methods initialized with "sufficient diversity" converge globally.

- diversity at initialization is key for success of training
- non-standard scaling of initialization and step-size
- highly non-linear dynamics and regularization allowed

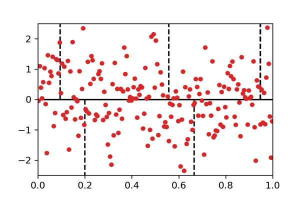
[Refs]:
Chizat, Bach (2018). *On the Global Convergence of Gradient Descent for Over-parameterized Models [...].*

Numerical Illustrations



(a) ReLU **(b)** Sigmoid

Population loss at convergence vs m for training a 2-layers NN in the teacher-student setting in 100-d.



This principle is general: e.g. sparse deconvolution.

Wasserstein Gradient Flow

- parameterize the model with a probability measure μ:

$$f(\mu, x) = \int \phi(w, x) \mathrm{d}\mu(w), \quad \mu \in \mathcal{P}(\mathbb{R}^d)$$

Wasserstein Gradient Flow

- parameterize the model with a probability measure μ:

$$f(\mu, x) = \int \phi(w, x) \mathrm{d}\mu(w), \quad \mu \in \mathcal{P}(\mathbb{R}^d)$$

- consider the population loss over $\mathcal{P}(\mathbb{R}^d)$:

$$F(\mu) := \mathbb{E}_{(x,y)} \left[\ell \left(\int \phi(w, x) \mathrm{d}\mu(w), y \right) \right].$$

$\rightsquigarrow$ convex in linear geometry but non-convex in Wasserstein

Wasserstein Gradient Flow

- parameterize the model with a probability measure μ:

$$f(\mu, x) = \int \phi(w, x)\mathrm{d}\mu(w), \quad \mu \in \mathcal{P}(\mathbb{R}^d)$$

- consider the population loss over $\mathcal{P}(\mathbb{R}^d)$:

$$F(\mu) := \mathbb{E}_{(x,y)}\left[\ell\left(\int \phi(w, x)\mathrm{d}\mu(w), y\right)\right].$$

$\rightsquigarrow$ convex in linear geometry but non-convex in Wasserstein

- define the gradient flow (small step-size gradient descent):

$$\mu_0 \in \mathcal{P}(\mathbb{R}^d), \qquad \frac{d}{dt}\mu_t = -\mathrm{div}(\mu_t v_t)$$

where $v_t(w) = -\nabla F'(\mu_t)$ is the Wasserstein gradient of F.

[Refs]:

Bach (2017). *Breaking the Curse of Dimensionality with Convex Neural Networks.*

Ambrosio, Gigli, Savaré (2008). *Gradient Flows in Metric Spaces and in the Space of Probability Measures.*

Mean-Field Limit for SGD

Now consider the actual training trajectory ((x_k, y_k) i.i.d):

$$\begin{cases} w^{(k)} = w^{(k-1)} - \eta m \nabla_w [\ell(f(w^{(k-1)}, x^{(k)}), y^{(k)})] \\ \hat{\mu}_m^{(k)} = \dfrac{1}{m} \displaystyle\sum_{i=1}^{m} \delta_{w_i^{(k)}} \end{cases}$$

Theorem (Mei, Montanari, Nguyen '18)

Under regularity assumptions, if $w_1(0), w_2(0), \ldots$ are drawn independently accordingly to μ_0 then with probability $1 - e^{-z}$,

$$\|\hat{\mu}_m^{(\lfloor t/\eta \rfloor)} - \mu_t\|_{BL}^2 \lesssim e^{Ct} \max\left\{\eta, \frac{1}{m}\right\} \left(z + d + \log \frac{m}{\eta}\right)$$

[Refs]:

Mei, Montanari, Nguyen (2018). *A Mean Field View of the Landscape of Two-Layers Neural Networks.*

Global Convergence (more formal)

Theorem (Homogeneous case)

Assume that μ_0 is supported on a centered sphere or ball, that ϕ is 2-homogeneous in the weights and some regularity. If μ_t converges in Wasserstein distance to μ_∞ then μ_∞ is a global minimizer of F. In particular, if $w_1(0), w_2(0), \ldots$ are drawn accordingly to μ_0 then

$$\lim_{t\to\infty, m\to\infty} F(\mu_{t,m}) = \min F.$$

- applies to 2-layers ReLU NNs
- other statement for sigmoid NNs

[Refs]:
Chizat, Bach (2018). *On the Global Convergence of Gradient Descent for Over-parameterized Models [...].*

Remark on the scaling

Change of parameterization/initialization $\Rightarrow$ change of behavior.

		Mean field	Lazy
model	$f(w,x)$	$\frac{1}{m}\sum\phi(w_i,x)$	$\frac{1}{\sqrt{m}}\sum\phi(w_i,x)$
step-size	η	$O(m)$	$O(1)$
init. predictor	$\Vert f(w_0,\cdot)\Vert$	$O(1/\sqrt{m})$	$O(1)$
"flatness"	$\Vert\nabla f\Vert^2/\Vert\nabla^2 f\Vert$	$O(1)$	$O(\sqrt{m})$
displacement	$\Vert w_\infty - w_0\Vert$	$O(1)$	$O(1/\sqrt{m})$

Generalization : implicit or explicit

Through single-pass SGD

Single-pass SGD acts like gradient flow

Through regularization

In regression tasks, adaptivity to subspace when minimizing

$$\min_{\mu \in \mathcal{P}(\mathbb{R}^d)} \frac{1}{n} \sum_{i=1}^{n} \left| \int \phi(w, x_i) \mathrm{d}\mu(w) - y_i \right|^2 + \int V(w) \mathrm{d}\mu(w)$$

where ϕ is ReLU activation and V a ℓ_1-type regularizer.

$\rightsquigarrow$ explicit sample complexity bounds (but differentiability issues)

$\rightsquigarrow$ also some bounds under separability assumptions (same issues)

[Refs]:

Bach (2017). *Breaking the Curse of Dimensionality with Convex Neural Networks.*

Wei, Lee, Liu, Ma (2018). *On the Margin Theory of Feedforward Neural Networks.*

Lessons to be drawn

For practice

- over-parameterization/random init. yields global convergence
- choice of scalings is fundamental

For theory

- strong generalization guaranties need neurons that move
- non-quantitative technics still leads to insights

$\rightsquigarrow$ ongoing research: quantitative statements, multi-layers

What I did not talk about

Focus was on gradient-based training in “realistic” settings.

Wide range of other approaches

- loss landscape analysis
- linear neural networks
- phase transition/computational barriers
- tensor decomposition
- ...

[Refs]:
Arora, Cohen, Golowich, Hu (2018). *Convergence Analysis of Gradient Descent for Deep Linear Neural Networks*
Aubin, Maillard, Barbier, Krzakala, Macris, Zdeborová (2018). *The Committee Machine: Computational to Statistical Gaps in Learning a Two-layers Neural Network.*
Zhang, Yu, Wang, Gu (2018). *Learning One-hidden-layer ReLU Networks via Gradient Descent.*

Conclusion

- **several regimes, several theories**
- **calls for new tools, new math models**

Perspectives

- **performance of tangent model?**
- **how do NNs efficiently perform high dimensional feature selection?**

[Papers with F. Bach:]
- On the Global Convergence of Over-parameterized Models using Optimal Transport. (NeurIPS 2018).
- A Note on Lazy Training in Differentiable Programming.