Theories of Neural Networks Training

Lazy and Mean Field Regimes

Lénaïc Chizat*, joint work with Francis Bach+
April 10th 2019 - University of Basel

*CNRS and Université Paris-Sud  +INRIA and ENS Paris
Introduction
Supervised machine learning

- given input/output training data \((x^{(1)}, y^{(1)}), \ldots, (x^{(n)}, y^{(n)})\)
- 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 \(\ell\) to compare outputs: squared, logistic, cross-entropy...
- starting from some \(w_0\), update parameters using gradients

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

\[
\begin{align*}
    w^{(k)} &= w^{(k-1)} - \eta^{(k)} \nabla_w [\ell(f(w^{(k-1)}, x^{(k)}), y^{(k)})]
\end{align*}
\]

[Refs]:
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:

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

with activation \( \sigma \) and parameters \( w = (W_1, b_1), ..., (W_L, b_L) \).
### Challenges for Theory

#### Need for new theoretical approaches
- optimization: non-convex, compositional structure
- statistics: over-parameterized, works without 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]:
Lazy Training
Let $f(w, x)$ be a differentiable model and $w_0$ an initialization.
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 \( \alpha \) 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 \( \alpha \), gradient-based methods on the non-linear model \( \alpha f \) and on the tangent model \( T_f \) learn the same model, up to a \( O(1/\alpha) \) remainder.

- lazy because parameters hardly move
- optimization of linear models is rather well understood
- recovers kernel ridgeless regression with offset \( f(w_0, \cdot) \) and

\[
K(x, x') = \langle \nabla_w f(w_0, x), \nabla_w f(w_0, x') \rangle
\]

[Refs]:
Allen-Zhu, Li, Liang (2018). *Learning and Generalization in Overparameterized Neural Networks [...].*

5/20
Criteria for lazy training (informal)

\[ \| T_f(w^*, \cdot) - f(w_0, \cdot) \| \ll \frac{\| \nabla f(w_0, \cdot) \|^2}{\| \nabla^2 f(w_0, \cdot) \|} \]

Distance to best linear model \quad \text{“Flatness” around initialization}

\[ \nRightarrow \text{difficult to estimate in general} \]

Examples

- **Homogeneous models.**
  If for \( \lambda > 0, \ f(\lambda w, x) = \lambda^L f(w, x) \) then flatness \( \sim \| w_0 \|^L \)

- **NNs with large layers.**
  Occurs if initialized with scale \( O(1/\sqrt{\text{fan}_{\text{in}}}) \)
Vanilla NN with $W_{i,j} \sim \mathcal{N}(0, \tau_w^2 / \text{fan}_\text{in})$ and $b_i \sim \mathcal{N}(0, \tau_b^2)$.

**Model at initialization**

As widths of layers diverge, $f(w_0, \cdot) \sim \mathcal{GP}(0, \Sigma^L)$ where

$$
\Sigma^{l+1}(x, x') = \tau_b^2 + \tau_w^2 \cdot \mathbb{E}_{z^l \sim \mathcal{GP}(0, \Sigma^l)}[\sigma(z^l(x)) \cdot \sigma(z^l(x'))].
$$

**Limit tangent kernel**

In the same limit, $\langle \nabla_w f(w_0, x), \nabla_w f(w_0, x') \rangle \rightarrow K^L(x, x')$ where

$$
K^{l+1}(x, x') = K^l(x, x') \dot{\Sigma}^{l+1}(x, x') + \Sigma^{l+1}(x, x')
$$

and $\dot{\Sigma}^{l+1}(x, x') = \mathbb{E}_{z^l \sim \mathcal{GP}(0, \Sigma^l)}[\dot{\sigma}(z^l(x)) \cdot \dot{\sigma}(z^l(x'))].$

$\rightarrow$ cf. A. Jacot’s talk of last week

[Refs]:
Numerical Illustrations

(a) Not lazy
(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 $\tau$
Lessons to be drawn

For practice
- our guess: instead, feature selection is why NNs work
- investigation needed 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?*
Global convergence for 2-layers NNs
Two Layers NNs

With activation $\sigma$, 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:

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

[Refs]:
Rotskoff, Vanden-Eijndem (2018). *Parameters as Interacting Particles [...].*
Sirignano, Spiliopoulos (2018). *Mean Field Analysis of Neural Networks.*
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
- highly non-linear dynamics and regularization allowed

[Refs]:
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.
Idealized Dynamic

- parameterize the model with a probability measure $\mu$:

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

[Refs]:
• parameterize the model with a probability measure $\mu$:

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

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

$$ F(\mu) := \mathbb{E}_{(x, y)} [\ell (f(\mu, x), y)]. $$

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

- parameterize the model with a probability measure $\mu$:
  \[ f(\mu, x) = \int \phi(w, x) d\mu(w), \quad \mu \in \mathcal{P}(\mathbb{R}^d) \]

- consider the population loss over $\mathcal{P}(\mathbb{R}^d)$:
  \[ F(\mu) := \mathbb{E}_{(x, y)} [\ell(f(\mu, x), y)]. \]
  $\sim$ convex in linear geometry but non-convex in Wasserstein

- define the Wasserstein Gradient Flow:
  \[ \mu_0 \in \mathcal{P}(\mathbb{R}^d), \quad \frac{d}{dt} \mu_t = -\text{div}(\mu_t v_t) \]
  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.*
Mean-Field Limit for SGD

Now consider the actual training trajectory \((x_k, y_k) \text{ i.i.d.}):\n\begin{align*}
\begin{cases}
  w^{(k)} &= w^{(k-1)} - \eta m \nabla_w [\ell (f (w^{(k-1)}, x^{(k)}), y^{(k)})] \\
  \hat{\mu}_m^{(k)} &= \frac{1}{m} \sum_{i=1}^{m} \delta_{w_i^{(k)}}
\end{cases}
\end{align*}

**Theorem (Mei, Montanari, Nguyen ’18)**

*Under regularity assumptions, if \(w_1(0), w_2(0), \ldots\) are drawn independently accordingly to \(\mu_0\) then with probability \(1 - e^{-z}\),*
\[
\left\| \hat{\mu}_m^{(\lfloor t/\eta \rfloor)} - \mu_t \right\|_{BL}^2 \lesssim e^{Ct} \max \left\{ \eta, \frac{1}{m} \right\} \left( z + d + \log \frac{m}{\eta} \right)
\]

[Refs]:
Theorem (Homogeneous case)

Assume that $\mu_0$ is supported on a centered sphere or ball, that $\phi$ is $2$-homogeneous in the weights and some regularity. If $\mu_t$ converges in Wasserstein distance to $\mu_\infty$ then $\mu_\infty$ is a global minimizer of $F$. In particular, if $w_1(0), w_2(0), \ldots$ are drawn accordingly to $\mu_0$ then

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

- applies to $2$-layers ReLU NNs (different statement for sigmoid)
- general consistency principle for optimization over measures
- see paper for precise conditions

[Refs]:
Remark on the scaling

Change of init. scaling \(\Rightarrow\) change of asymptotic behavior.

<table>
<thead>
<tr>
<th></th>
<th>Mean field</th>
<th>Lazy</th>
</tr>
</thead>
<tbody>
<tr>
<td>model</td>
<td>(f(w, x))</td>
<td>(\frac{1}{m} \sum \phi(w_i, x))</td>
</tr>
<tr>
<td>init. predictor</td>
<td>(|f(w_0, \cdot)|)</td>
<td>(O(1/\sqrt{m}))</td>
</tr>
<tr>
<td>“flatness”</td>
<td>(|\nabla f|^2/|\nabla^2 f|)</td>
<td>(O(1))</td>
</tr>
<tr>
<td>displacement</td>
<td>(|w_\infty - w_0|)</td>
<td>(O(1))</td>
</tr>
</tbody>
</table>

- deep NNs need initialization in \(O(\sqrt{2/fan_{in}})\)
- yet, linearization doesn’t seem to explain state of the art perf
Generalization: implicit or explicit

**Through single-pass SGD**

Single-pass SGD acts like gradient flow of *population* loss.

\[ \text{but needs convergence rate} \]

**Through regularization**

In regression tasks, adaptivity to subspace when minimizing

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

where \( \phi \) is ReLU activation and \( V \) a \( \ell_1 \)-type regularizer.

\[ \text{both explicit sample complexity bounds (but differentiability issues)} \]

\[ \text{also some bounds under separability assumptions (same issues)} \]

[Refs]:
Bach (2017). *Breaking the Curse of Dimensionality with Convex Neural Networks.*
# Lessons to be drawn

## For practice
- over-parameterization/random init. yields global convergence
- changing variance of initialization impacts behavior

## For theory
- strong generalization guaranties need neurons that move
- non-quantitative technics still lead to insights
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*
## Conclusion
- several regimes, several theories
- calls for new tools, new math models

## Perspectives
**How do NNs efficiently perform high dimensional feature selection?**

[Papers with F. Bach:]
- A Note on Lazy Training in Differentiable Programming.