On the Convergence of Gradient Descent for Wide Two-Layer Neural Networks

Francis Bach
INRIA - Ecole Normale Supérieure, Paris, France

Joint work with Lénaïc Chizat
Parametric supervised machine learning

- **Data**: $n$ observations $(x_i, y_i) \in \mathcal{X} \times \mathcal{Y}, i = 1, \ldots, n$

- **Prediction function** $h(x, \theta) \in \mathbb{R}$ parameterized by $\theta \in \mathbb{R}^d$
Parametric supervised machine learning

- **Data:** $n$ observations $(x_i, y_i) \in X \times Y$, $i = 1, \ldots, n$

- **Prediction function** $h(x, \theta) \in \mathbb{R}$ parameterized by $\theta \in \mathbb{R}^d$

- **Linear predictions**
  
  - $h(x, \theta) = \theta^\top \Phi(x)$

- E.g., **advertising**: $n > 10^9$
  
  - $\Phi(x) \in \{0, 1\}^d$, $d > 10^9$
  
  - Navigation history + ad
Parametric supervised machine learning

- **Data**: $n$ observations $(x_i, y_i) \in \mathcal{X} \times \mathcal{Y}$, $i = 1, \ldots, n$

- **Prediction function** $h(x, \theta) \in \mathbb{R}$ parameterized by $\theta \in \mathbb{R}^d$

\[
\begin{align*}
x_1 & \quad x_2 & \quad x_3 & \quad x_4 & \quad x_5 & \quad x_6 \\
\end{align*}
\]

\[
\begin{align*}
y_1 & = 1 & y_2 & = 1 & y_3 & = 1 & y_4 & = -1 & y_5 & = -1 & y_6 & = -1 \\
\end{align*}
\]

- **Neural networks** ($n, d > 10^6$): $h(x, \theta) = \theta_r^\top \sigma(\theta_{r-1}^\top \sigma(\cdots \theta_2^\top \sigma(\theta_1^\top x)))$
Parametric supervised machine learning

**Data:** $n$ observations $(x_i, y_i) \in X \times Y$, $i = 1, \ldots, n$

**Prediction function** $h(x, \theta) \in \mathbb{R}$ parameterized by $\theta \in \mathbb{R}^d$

**(regularized) empirical risk minimization:**

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

data fitting term + regularizer
Parametric supervised machine learning

• **Data:** \( n \) observations \((x_i, y_i) \in \mathcal{X} \times \mathcal{Y}, i = 1, \ldots, n\)

• **Prediction function** \( h(x, \theta) \in \mathbb{R} \) parameterized by \( \theta \in \mathbb{R}^d \)

• **(regularized) empirical risk minimization:**

\[
\min_{\theta \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, h(x_i, \theta)) + \lambda \Omega(\theta)
\]

\[
\text{data fitting term} + \text{regularizer}
\]

• **Actual goal:** minimize test error \( \mathbb{E}_{p(x,y)} \ell(y, h(x, \theta)) \)
Convex optimization problems

\[
\min_{\theta \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, h(x_i, \theta)) + \lambda \Omega(\theta)
\]

- **Conditions**: Convex loss and linear predictions \( h(x, \theta) = \theta^T \Phi(x) \)

- **Consequences**
  - Efficient algorithms (typically gradient-based)
  - Quantitative runtime and prediction performance guarantees
Convex optimization problems

\[
\min_{\theta \in \mathbb{R}^d} \; \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, h(x_i, \theta)) + \lambda \Omega(\theta)
\]

- **Conditions**: Convex loss and linear predictions \( h(x, \theta) = \theta^\top \Phi(x) \)

- **Consequences**
  - Efficient algorithms (typically gradient-based)
  - Quantitative runtime and prediction performance guarantees

- **Golden years of convexity in machine learning** (1995 to 2020)
  - Support vector machines and kernel methods
  - Sparsity / low-rank models with first-order methods
  - Optimal transport
  - Stochastic methods for large-scale learning and online learning
  - etc.
Theoretical analysis of deep learning

- Multi-layer neural network $h(x, \theta) = \theta_r^\top \sigma(\theta_{r-1}^\top \sigma(\cdots \theta_2^\top \sigma(\theta_1^\top x)))$

- NB: already a simplification
Theoretical analysis of deep learning

- **Multi-layer neural network** \( h(x, \theta) = \theta_r^\top \sigma(\theta_{r-1}^\top \sigma(\cdots \theta_2^\top \sigma(\theta_1^\top x))) \)

  - NB: already a simplification

- **Main difficulties**

  1. Non-convex optimization problems
  2. Generalization guarantees in the overparameterized regime
Optimization for multi-layer neural networks

- What can go wrong with non-convex optimization problems?
  - Local minima
  - Stationary points
  - Plateaux
  - Bad initialization
  - etc...
Optimization for multi-layer neural networks

• **What can go wrong with non-convex optimization problems?**
  
  – Local minima
  – Stationary points
  – Plateaux
  – Bad initialization
  – etc...

• **Generic local theoretical guarantees**
  
  – Convergence to stationary points or local minima
  – See, e.g., Lee et al. (2016); Jin et al. (2017)
Optimization for multi-layer neural networks

• What can go wrong with non-convex optimization problems?

  – Local minima
  – Stationary points
  – Plateaux
  – Bad initialization
  – etc...

• General **global** performance guarantees impossible to obtain
Optimization for multi-layer neural networks

• What can go wrong with non-convex optimization problems?
  – Local minima
  – Stationary points
  – Plateaux
  – Bad initialization
  – etc...

• General global performance guarantees impossible to obtain

• Special case of (deep) neural networks
  – Most local minima are equivalent (Choromanska et al., 2015)
  – No spurrious local minima (Soltanolkotabi et al., 2018)
Gradient descent for a single hidden layer

- **Predictor**: \( h(x) = \frac{1}{m} \theta_2^\top \sigma(\theta_1^\top x) = \frac{1}{m} \sum_{j=1}^{m} \theta_2(j) \cdot \sigma[\theta_1(\cdot, j)^\top x] \)

- **Goal**: minimize \( R(h) = \mathbb{E}_{p(x,y)} \ell(y, h(x)), \) with \( R \) convex
Gradient descent for a single hidden layer

- **Predictor:** \( h(x) = \frac{1}{m} \theta_2^\top \sigma(\theta_1^\top x) = \frac{1}{m} \sum_{j=1}^{m} \theta_2(j) \cdot \sigma[\theta_1(\cdot, j)^\top x] \)

- Family: \( h = \frac{1}{m} \sum_{j=1}^{m} \Psi(w_j) \) with \( \Psi(w_j)(x) = \theta_2(j) \cdot \sigma[\theta_1(\cdot, j)^\top x] \)

- **Goal:** minimize \( R(h) = \mathbb{E}_{p(x,y)} \ell(y, h(x)) \), with \( R \) convex
Gradient descent for a single hidden layer

- **Predictor**: \( h(x) = \frac{1}{m} \theta_2^\top \sigma(\theta_1^\top x) = \frac{1}{m} \sum_{j=1}^{m} \theta_2(j) \cdot \sigma[\theta_1(\cdot, j)^\top x] \)

- **Family**: \( h = \frac{1}{m} \sum_{j=1}^{m} \Psi(w_j) \) with \( \Psi(w_j)(x) = \theta_2(j) \cdot \sigma[\theta_1(\cdot, j)^\top x] \)

- **Goal**: minimize \( R(h) = \mathbb{E}_{p(x,y)} \ell(y, h(x)) \), with \( R \) convex

- **Main insight**

  \[ h = \frac{1}{m} \sum_{j=1}^{m} \Psi(w_j) = \int_{\mathcal{W}} \Psi(w) d\mu(w) \text{ with } d\mu(w) = \frac{1}{m} \sum_{j=1}^{m} \delta_{w_j} \]
Gradient descent for a single hidden layer

• **Predictor:**  
  \[ h(x) = \frac{1}{m} \theta_2^\top \sigma(\theta_1^\top x) = \frac{1}{m} \sum_{j=1}^{m} \theta_2(j) \cdot \sigma[\theta_1(\cdot, j)^\top x] \]

  – **Family:**  
  \[ h = \frac{1}{m} \sum_{j=1}^{m} \Psi(w_j) \quad \text{with} \quad \Psi(w_j)(x) = \theta_2(j) \cdot \sigma[\theta_1(\cdot, j)^\top x] \]

• **Goal:** minimize  
  \[ R(h) = \mathbb{E}_{p(x,y)} \ell(y, h(x)), \text{ with } R \text{ convex} \]

• **Main insight**
  
  –  
  \[ h = \frac{1}{m} \sum_{j=1}^{m} \Psi(w_j) = \int_{\mathcal{W}} \Psi(w) d\mu(w) \quad \text{with} \quad d\mu(w) = \frac{1}{m} \sum_{j=1}^{m} \delta_{w_j} \]

  – Overparameterized models with \( m \) large \( \approx \) measure \( \mu \) with densities
  
  – Barron (1993); Kurkova and Sanguineti (2001); Bengio et al. (2006); Rosset et al. (2007); Bach (2017)
Optimization on measures

• Minimize with respect to measure $\mu$: $R\left(\int_{W} \Psi(w) d\mu(w)\right)$
  
  – Convex optimization problem on measures
  – Frank-Wolfe techniques for incremental learning
  – Non-tractable (Bach, 2017), not what is used in practice
Optimization on measures

- Minimize with respect to measure $\mu$: $R\left(\int_{\mathcal{W}} \Psi(w) d\mu(w)\right)$
  - Convex optimization problem on measures
  - Frank-Wolfe techniques for incremental learning
  - Non-tractable (Bach, 2017), not what is used in practice

- Represent $\mu$ by a finite set of “particles” $\mu = \frac{1}{m} \sum_{j=1}^{m} \delta_{w_j}$
  - Backpropagation = gradient descent on $(w_1, \ldots, w_m)$

- Three questions:
  - Algorithm limit when number of particles $m$ gets large
  - Global convergence to a global minimizer
  - Prediction performance
Many particle limit and global convergence (Chizat and Bach, 2018)

- **General framework:** minimize $F(\mu) = R\left( \int_{\mathcal{W}} \Psi(w) d\mu(w) \right)$

- **Algorithm:** minimizing $F_m(w_1, \ldots, w_m) = R\left( \frac{1}{m} \sum_{j=1}^{m} \Psi(w_j) \right)$
Many particle limit and global convergence  
(Chizat and Bach, 2018)

- **General framework:** minimize \( F(\mu) = R\left( \int_{\mathcal{W}} \Psi(w) d\mu(w) \right) \)

  - Algorithm: minimizing \( F_m(w_1, \ldots, w_m) = R\left( \frac{1}{m} \sum_{j=1}^{m} \Psi(w_j) \right) \)

  - Gradient flow \( \dot{W} = -m \nabla F_m(W) \), with \( W = (w_1, \ldots, w_m) \)

  - Idealization of (stochastic) gradient descent
Many particle limit and global convergence (Chizat and Bach, 2018)

- General framework: minimize $F(\mu) = R\left( \int_\mathcal{W} \Psi(w) d\mu(w) \right)$

- Algorithm: minimizing $F_m(w_1, \ldots, w_m) = R\left( \frac{1}{m} \sum_{j=1}^{m} \Psi(w_j) \right)$

- Gradient flow $\dot{W} = -m \nabla F_m(W)$, with $W = (w_1, \ldots, w_m)$

- Idealization of (stochastic) gradient descent
  1. Single pass SGD on the unobserved expected risk
  2. Multiple pass SGD or full GD on the empirical risk
Many particle limit and global convergence
(Chizat and Bach, 2018)

• **General framework:** minimize $F(\mu) = R\left(\int_{\mathcal{W}} \Psi(w) d\mu(w)\right)$

  – Algorithm: minimizing $F_m(w_1, \ldots, w_m) = R\left(\frac{1}{m} \sum_{j=1}^{m} \Psi(w_j)\right)$

  – Gradient flow $\dot{W} = -m \nabla F_m(W)$, with $W = (w_1, \ldots, w_m)$

  – Idealization of (stochastic) gradient descent

• **Limit when $m$ tends to infinity**

  – Wasserstein gradient flow (Nitanda and Suzuki, 2017; Chizat and Bach, 2018; Mei, Montanari, and Nguyen, 2018; Sirignano and Spiliopoulos, 2018; Rotskoff and Vanden-Eijnden, 2018)

• NB: for more details on gradient flows, see Ambrosio et al. (2008)
Many particle limit and global convergence
(Chizat and Bach, 2018)

• (informal) theorem: when the number of particles tends to infinity, the gradient flow converges to the global optimum
Many particle limit and global convergence
(Chizat and Bach, 2018)

- **(informal) theorem**: when the number of particles tends to infinity, the gradient flow converges to the global optimum
  - See precise definitions and statement in paper
  - Two key ingredients: homogeneity and initialization
Many particle limit and global convergence  
(Chizat and Bach, 2018)

- **(informal) theorem**: when the number of particles tends to infinity, the gradient flow converges to the global optimum
  - See precise definitions and statement in paper
  - Two key ingredients: homogeneity and initialization

- **Homogeneity** (see, e.g., Haeffele and Vidal, 2017; Bach et al., 2008)
  - Full or partial, e.g., \( \Psi(w_j)(x) = m\theta_2(j) \cdot \sigma[\theta_1(\cdot, j)^\top x] \)
  - Applies to rectified linear units (but also to sigmoid activations)

- **Sufficiently spread initial measure**
  - Needs to cover the entire sphere of directions
Many particle limit and global convergence
(Chizat and Bach, 2018)

- **(informal) theorem**: when the number of particles tends to infinity, the gradient flow converges to the global optimum
  - See precise definitions and statement in paper
  - Two key ingredients: homogeneity and initialization

- **Homogeneity** (see, e.g., Haeffele and Vidal, 2017; Bach et al., 2008)
  - Full or partial, e.g., \( \Psi(w_j)(x) = m\theta_2(j) \cdot \sigma[\theta_1(\cdot, j)^\top x] \)
  - Applies to rectified linear units (but also to sigmoid activations)

- **Sufficiently spread initial measure**
  - Needs to cover the entire sphere of directions

- **Only qualitative!**
Simple simulations with neural networks

- ReLU units with \( d = 2 \) (optimal predictor has 5 neurons)

\[
h(x) = \frac{1}{m} \sum_{j=1}^{m} \Psi(w_j)(x) = \frac{1}{m} \sum_{j=1}^{m} \theta_2(j)\left(\theta_1(\cdot, j)^\top x\right)_+ \\
(\text{plotting } |\theta_2(j)|\theta_1(\cdot, j) \text{ for each hidden neuron } j)
\]

NB: also applies to spike deconvolution
Simple simulations with neural networks

- ReLU units with $d = 2$ (optimal predictor has 5 neurons)

NB: also applies to spike deconvolution
From optimization to statistics

- **Summary**: with $h(x) = \frac{1}{m} \sum_{j=1}^{m} \Psi(w_j)(x) = \frac{1}{m} \sum_{j=1}^{m} \theta_2(j)(\theta_1(\cdot,j)^\top x) +$

  - If $m$ tends to infinity, the gradient flow converges to a global minimizer of the risk $R(h) = \mathbb{E}_{p(x,y)}\ell(y, h(x))$
  - Requires well-spread initialization, no quantitative results
From optimization to statistics

• **Summary:** with \( h(x) = \frac{1}{m} \sum_{j=1}^{m} \Psi(w_j)(x) = \frac{1}{m} \sum_{j=1}^{m} \theta_2(j)(\theta_1(\cdot, j)^\top x) \)

  – If \( m \) tends to infinity, the gradient flow converges to a global minimizer of the risk \( R(h) = E_{p(x,y)} \ell(y, h(x)) \)
  – Requires well-spread initialization, no quantitative results

• **Single-pass SGD** with \( R \) the (unobserved) expected risk

  – Converges to an optimal predictor on the testing distribution
  – Tends to underfit
From optimization to statistics

**Summary:** with $h(x) = \frac{1}{m} \sum_{j=1}^{m} \Psi(w_j)(x) = \frac{1}{m} \sum_{j=1}^{m} \theta_2(j)(\theta_1(\cdot, j)^\top x)$,

- If $m$ tends to infinity, the gradient flow converges to a global minimizer of the risk $R(h) = \mathbb{E}_{p(x,y)} \ell(y, h(x))$
- Requires well-spread initialization, no quantitative results

**Single-pass SGD** with $R$ the (unobserved) **expected** risk

- Converges to an optimal predictor on the testing distribution
- Tends to underfit

**Multiple-pass SGD or full GD** with $R$ the **empirical** risk

- Converges to an optimal predictor on the training distribution
- Should overfit?
Interpolation regime

- Minimizing $R(h) = \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, h(x_i))$ for $h(x) = \frac{1}{m} \sum_{j=1}^{m} \theta_2(j) (\theta_1(\cdot, j) ^\top x)$ +

  - When $m(d + 1) > n$, typically there exist many $h$ such that

    $\forall i \in \{1, \ldots, n\}, \ h(x_i) = y_i \quad \text{(or } \ell(y_i, h(x_i)) = 0)\)

  - See Belkin et al. (2018); Ma et al. (2018); Vaswani et al. (2019)
Interpolation regime

• Minimizing $R(h) = \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, h(x_i))$ for $h(x) = \frac{1}{m} \sum_{j=1}^{m} \theta_2(j)(\theta_1(\cdot, j)^\top x)$

  – When $m(d + 1) > n$, typically there exist many $h$ such that

    $\forall i \in \{1, \ldots, n\}, \ h(x_i) = y_i \quad \text{(or } \ell(y_i, h(x_i)) = 0\text{)}$

  – See Belkin et al. (2018); Ma et al. (2018); Vaswani et al. (2019)

• Which $h$ is the gradient flow converging to?

  – Implicit bias of (stochastic) gradient descent
Interpolation regime

- Minimizing $R(h) = \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, h(x_i))$ for $h(x) = \frac{1}{m} \sum_{j=1}^{m} \theta_2(j) (\theta_1(\cdot, j)^\top x)$.
- When $m(d + 1) > n$, typically there exist many $h$ such that
  \[ \forall i \in \{1, \ldots, n\}, \ h(x_i) = y_i \quad \text{(or } \ell(y_i, h(x_i)) = 0) \]
- See Belkin et al. (2018); Ma et al. (2018); Vaswani et al. (2019)

- **Which $h$ is the gradient flow converging to?**
  - Implicit bias of (stochastic) gradient descent
  - Typically **minimum Euclidean norm solution** (Gunasekar et al., 2017; Soudry et al., 2018; Gunasekar et al., 2018)
Interpolation regime

• Minimizing $R(h) = \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, h(x_i))$ for $h(x) = \frac{1}{m} \sum_{j=1}^{m} \theta_2(j)(\theta_1(\cdot, j)^\top x)$

  – When $m(d+1) > n$, typically there exist many $h$ such that

    $$\forall i \in \{1, \ldots, n\}, \ h(x_i) = y_i \quad (\text{or } \ell(y_i, h(x_i)) = 0)$$

  – See Belkin et al. (2018); Ma et al. (2018); Vaswani et al. (2019)

• Which $h$ is the gradient flow converging to?

  – Implicit bias of (stochastic) gradient descent
  – Typically minimum Euclidean norm solution (Gunasekar et al., 2017; Soudry et al., 2018; Gunasekar et al., 2018)
  – Surprisingly difficult for the square loss
  – Surprisingly easy for the logistic loss
Maximum margin and logistic regression

- **Logistic regression:**\[ \min_{\theta \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} \log(1 + \exp(-y_i \theta^\top x_i)) \]

- Separable data: \exists \theta \in \mathbb{R}^d, \forall i \in \{1, \ldots, n\}, y_i \theta^\top x_i > 0
Maximum margin and logistic regression

- **Logistic regression:** \[ \min_{\theta \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} \log(1 + \exp(-y_i \theta^\top x_i)) \]

- Separable data: \( \exists \theta \in \mathbb{R}^d, \forall i \in \{1, \ldots, n\}, y_i \theta^\top x_i > 1 \)
Maximum margin and logistic regression

- **Logistic regression:**
  
  \[
  \min_{\theta \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} \log(1 + \exp(-y_i \theta^\top x_i))
  \]

  - Separable data: \( \exists \theta \in \mathbb{R}^d, \forall i \in \{1, \ldots, n\}, y_i \theta^\top x_i > 1 \)
  - \( 0 = \text{infimum of the risk, attained for infinitely large } \|\theta\|_2 \)

(with \( u = y_i \theta^\top x_i \))
Maximum margin and logistic regression

• **Logistic regression:** \( \min_{\theta \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} \log(1 + \exp(-y_i \theta^\top x_i)) \)

  – Separable data: \( \exists \theta \in \mathbb{R}^d, \ \forall i \in \{1, \ldots, n\}, \ y_i \theta^\top x_i > 1 \)
  – \( 0 = \inf \)imum of the risk, attained for infinitely large \( \|\theta\|_2 \)

• **Implicit bias of gradient descent** (Soudry et al., 2018)

  – GD diverges but \( \frac{1}{\|\theta_t\|_2} \theta_t \) converges to maximum margin separator

    \[ \max_{\|\eta\|_2=1} \min_{i \in \{1, \ldots, n\}} y_i \eta^\top x_i \]

    – often written as

    \[ \min \|\theta\|_2^2 \text{ such that } \forall i, \ y_i \theta^\top x_i > 1 \]

    – Separable support vector machine (Vapnik and Chervonenkis, 1964)
Logistic regression for two-layer neural networks

\[ h(x) = \frac{1}{m} \sum_{j=1}^{m} \theta_2(j) (\theta_1(\cdot, j)^\top x)_+ \]

- **Overparameterized regime** \( m \to +\infty \)
  - Will converge to well-defined “maximum margin” separator
Logistic regression for two-layer neural networks

\[ h(x) = \frac{1}{m} \sum_{j=1}^{m} \theta_2(j) (\theta_1(\cdot, j) ^\top x)_+ \]

- Overparameterized regime \( m \to +\infty \)
  - Will converge to well-defined “maximum margin” separator

- Two different regimes (Chizat and Bach, 2020)
  1. Optimizing over output layer only \( \theta_2 \): random feature kernel
  2. Optimizing over all layers \( \theta_1, \theta_2 \): feature learning
Random feature kernel regime - I

- **Prediction function** \( h(x) = \frac{1}{m} \sum_{j=1}^{m} \theta_2(j) (\theta_1(\cdot, j)^\top x) \) +
  
  - Input weights \( \theta_1(\cdot, j), j = 1, \ldots, m \), random and fixed
  - Optimize over output weights \( \theta_2 \in \mathbb{R}^m \)
  - Corresponds to linear predictor with \( \Phi(x)_j = \frac{1}{\sqrt{m}}(\theta_1(\cdot, j)^\top x) \) +
Random feature kernel regime - I

- **Prediction function** \( h(x) = \frac{1}{m} \sum_{j=1}^{m} \theta_2(j)(\theta_1(\cdot, j)^\top x)_+ \)
  - Input weights \( \theta_1(\cdot, j), j = 1, \ldots, m, \) random and fixed
  - Optimize over output weights \( \theta_2 \in \mathbb{R}^m \)
  - Corresponds to linear predictor with \( \Phi(x)_j = \frac{1}{\sqrt{m}}(\theta_1(\cdot, j)^\top x)_+ \)

- **Converges to separator with minimum norm** \( \|\theta_2\|_2^2 \)
  - Direct application of results from Soudry et al. (2018)
  - Limit when \( m \) tends to infinity?
Random feature kernel regime - I

- **Prediction function** 
  \[ h(x) = \frac{1}{m} \sum_{j=1}^{m} \theta_2(j) (\theta_1(\cdot, j)^\top x)_+ \]
  - Input weights \( \theta_1(\cdot, j), j = 1, \ldots, m \), random and fixed
  - Optimize over output weights \( \theta_2 \in \mathbb{R}^m \)
  - Corresponds to linear predictor with \( \Phi(x)_j = \frac{1}{\sqrt{m}} (\theta_1(\cdot, j)^\top x)_+ \)

- **Converges to separator with minimum norm** \( \|\theta_2\|_2^2 \)
  - Direct application of results from Soudry et al. (2018)
  - Limit when \( m \) tends to infinity?

- **Kernel** 
  \[ \Phi(x)^\top \Phi(x') = \frac{1}{m} \sum_{j=1}^{m} (\theta_1(\cdot, j)^\top x)_+ (\theta_1(\cdot, j)^\top x')_+ \]
  - Converges to \( \mathbb{E}_{\eta}(\eta^\top x)_+ (\eta^\top x')_+ \)
  - “Random features” (Neal, 1995; Rahimi and Recht, 2007)
• **Limiting kernel** \( \mathbb{E}_\eta(\eta^\top x)_+ (\eta^\top x')_+ \)
  
  - Reproducing kernel Hilbert spaces (RKHS) (see, e.g., Schölkopf and Smola, 2001)
  - Space of (very) **smooth** functions (Bach, 2017)
Random feature kernel regime - II

- **Limiting kernel** $\mathbb{E}_\eta(\eta^\top x)_+ + (\eta^\top x')_+$
  - Reproducing kernel Hilbert spaces (RKHS) (see, e.g., Schölkopf and Smola, 2001)
  - Space of (very) smooth functions (Bach, 2017)

- **(informal) theorem** (Chizat and Bach, 2020): when $m \to +\infty$, the gradient flow converges to the function in the RKHS that separates the data with minimum RKHS norm
  - Quantitative analysis available
  - Letting $m \to +\infty$ is useless in practice
  - See Montanari et al. (2019) for related work in the context of “double descent”
From RKHS norm to variation norm

- Alternative definition of the RKHS norm

\[ \|f\|^2 = \inf_{a(\cdot)} \int_{S^d} |a(\eta)|^2 d\tau(\eta) \text{ such that } f(x) = \int_{S^d} (\eta^\top x) + a(\eta) d\tau(\eta) \]

- Input weights uniformly distributed on the sphere (Bach, 2017)
- Smooth functions (does not allow single hidden neuron)
From RKHS norm to variation norm

• Alternative definition of the RKHS norm

\[ \|f\|^2 = \inf_{a(\cdot)} \int_{S^d} |a(\eta)|^2 d\tau(\eta) \text{ such that } f(x) = \int_{S^d} (\eta^\top x) + a(\eta) d\tau(\eta) \]

– Input weights uniformly distributed on the sphere (Bach, 2017)
– Smooth functions (does not allow single hidden neuron)

• Variation norm (Kurkova and Sanguineti, 2001)

\[ \Omega(f) = \inf_{a(\cdot)} \int_{S^d} |a(\eta)| d\tau(\eta) \text{ such that } f(x) = \int_{S^d} (\eta^\top x) + a(\eta) d\tau(\eta) \]

– Larger space including non-smooth functions
– Allows single hidden neuron
– Adaptivity to linear structures (Bach, 2017)
Feature learning regime

- **Prediction function** \( h(x) = \frac{1}{m} \sum_{j=1}^{m} \theta_2(j)(\theta_1(\cdot, j)^\top x) \) +

- Optimize over all weights \( \theta_1, \theta_2 \)
Feature learning regime

- **Prediction function** $h(x) = \frac{1}{m} \sum_{j=1}^{m} \theta_2(j)(\theta_1(\cdot, j)^\top x)_+$
  - Optimize over all weights $\theta_1, \theta_2$

- **(informal) theorem** (Chizat and Bach, 2020): when $m \to +\infty$, the gradient flow converges to the function that separates the data with minimum variation norm
  - Actual learning of representations
  - Adaptivity to linear structures (see Chizat and Bach, 2020)
  - No known convex optimization algorithms in polynomial time
  - End of the curve of double descent (Belkin et al., 2018)
Optimizing over two layers

- Two-dimensional classification with “bias” term

**Space of parameters**
- Plot of $|\theta_2(j)|\theta_1(\cdot, j)$
- Color depends on sign of $\theta_2(j)$
- “tanh” radial scale

**Space of predictors**
- $(+/-)$ training set
- One color per class
- Line shows 0 level set of $h$
Comparison of kernel and feature learning regimes

- $l_2$ (left: kernel) vs. $l_1$ (right: feature learning and variation norm)
Comparison of kernel and feature learning regimes

- Adaptivity to linear structures

- **Two-class classification in dimension** $d = 15$
  - Two first coordinates as shown below
  - All other coordinates uniformly at random
Conclusion

• Summary
  – Qualitative analysis of gradient descent for 2-layer neural networks
  – Global convergence with infinitely many neurons
  – Convergence to maximum margin separators in well-defined function spaces
  – Only qualitative
Conclusion

• Summary
  – Qualitative analysis of gradient descent for 2-layer neural networks
  – Global convergence with infinitely many neurons
  – Convergence to maximum margin separators in well-defined function spaces
  – Only qualitative

• Open problems
  – Quantitative analysis in terms of number of neurons $m$ and time $t$
  – Extension to convolutional neural networks
  – Extension to deep neural networks
References


Andrea Montanari, Feng Ruan, Youngtak Sohn, and Jun Yan. The generalization error of max-margin linear classifiers: High-dimensional asymptotics in the overparametrized regime. arXiv preprint


