Learning Single-Index Models with Shallow Neural Networks

Alberto Bietti (NYU)

joint with Joan Bruna (NYU), Clayton Sanford (Columbia), Min Jae Song (NYU)

"Youth in High Dimensions" workshop. ICTP, Trieste, June 30, 2022.

Structure for Neural Networks

Properties of usual deep learning problems (*e.g.*, images, text, graphs, proteins)

- **High-dimensional** data, representation learning
- **Optimization** with gradient descent works
- **Expressive** models (*e.g.*, zero training error)

Structure for Neural Networks

Properties of usual deep learning problems (*e.g.*, images, text, graphs, proteins)

- **High-dimensional** data, representation learning
- **Optimization** with gradient descent works
- **Expressive** models (*e.g.*, zero training error)

Data structure: consider regression problems with

$$y = F^*(x) + \text{noise}$$

- What are good structural assumptions on F^* for common problems?
- How can neural networks learn efficiently with such structure?

Prior Work

$$y = F^*(x) + \text{noise}$$

- **Non-parametric** classes: F^* is Lipschitz, β-smooth, etc.
  - Efficient learning even in kernel regimes (Caponnetto and De Vito, 2007)
  - **but**: curse of dimensionality

Prior Work

$$y = F^*(x) + \text{noise}$$

- **Non-parametric** classes: F^* is Lipschitz, β-smooth, etc.
  - Efficient learning even in kernel regimes (Caponnetto and De Vito, 2007)
  - **but**: curse of dimensionality
- **Teacher-student/planted** models: $F^*(x) = \phi(\langle \theta^*, x \rangle)$
  - Efficient optimization and recovery (Ben Arous et al., 2021; Soltanolkotabi, 2017)
  - **but**: not expressive, need to know the right activation ϕ

Prior Work

$$y = F^*(x) + \text{noise}$$

- **Non-parametric** classes: F^* is Lipschitz, β-smooth, etc.
  - Efficient learning even in kernel regimes (Caponnetto and De Vito, 2007)
  - **but**: curse of dimensionality
- **Teacher-student/planted** models: $F^*(x) = \phi(\langle \theta^*, x \rangle)$
  - Efficient optimization and recovery (Ben Arous et al., 2021; Soltanolkotabi, 2017)
  - **but**: not expressive, need to know the right activation ϕ
- **Single-index** models: $F^*(x) = f_*(\langle \theta^*, x \rangle)$, with f_* in non-parametric class
  - Break the curse of dimensionality with convex NNs/mean field regime (Bach, 2017a; Chizat and Bach, 2018; Mei et al., 2019)
  - **but**: intractable (mean-field), or complex dedicated algorithms (Dudeja and Hsu, 2018)

Prior Work

$$y = F^*(x) + \text{noise}$$

- **Non-parametric** classes: F^* is Lipschitz, β-smooth, etc.
  - Efficient learning even in kernel regimes (Caponnetto and De Vito, 2007)
  - **but**: curse of dimensionality
- **Teacher-student/planted** models: $F^*(x) = \phi(\langle \theta^*, x \rangle)$
  - Efficient optimization and recovery (Ben Arous et al., 2021; Soltanolkotabi, 2017)
  - **but**: not expressive, need to know the right activation ϕ
- **Single-index** models: $F^*(x) = f_*(\langle \theta^*, x \rangle)$, with f_* in non-parametric class
  - Break the curse of dimensionality with convex NNs/mean field regime (Bach, 2017a; Chizat and Bach, 2018; Mei et al., 2019)
  - **but**: intractable (mean-field), or complex dedicated algorithms (Dudeja and Hsu, 2018)
- **Others**: multi-index models, symmetries/invariances, hierarchy, ...

Prior Work

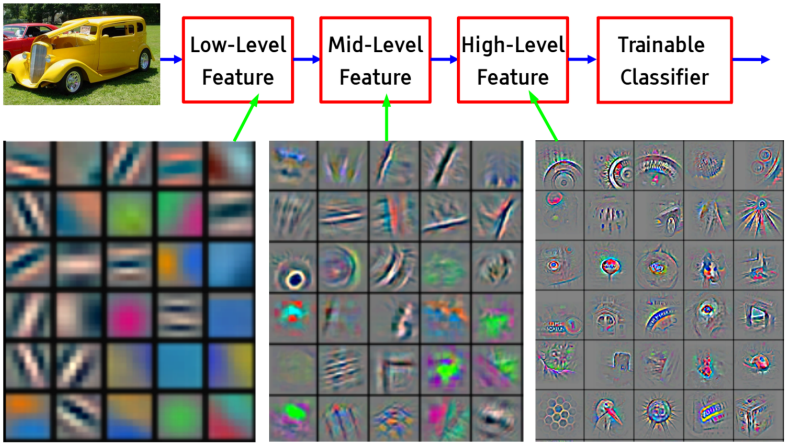
$$y = F^*(x) + \text{noise}$$

- **Non-parametric** classes: F^* is Lipschitz, β-smooth, etc.
  - Efficient learning even in kernel regimes (Caponnetto and De Vito, 2007)
  - **but**: curse of dimensionality
- **Teacher-student/planted** models: $F^*(x) = \phi(\langle \theta^*, x \rangle)$
  - Efficient optimization and recovery (Ben Arous et al., 2021; Soltanolkotabi, 2017)
  - **but**: not expressive, need to know the right activation ϕ
- **Single-index** models: $F^*(x) = f_*(\langle \theta^*, x \rangle)$, with f_* in non-parametric class
  - Break the curse of dimensionality with convex NNs/mean field regime (Bach, 2017a; Chizat and Bach, 2018; Mei et al., 2019)
  - **but**: intractable (mean-field), or complex dedicated algorithms (Dudeja and Hsu, 2018)
- **Others**: multi-index models, symmetries/invariances, hierarchy, ...

This work: efficient learning of single-index models with shallow networks

Example motivation: CNN filters

Multi-index model: $F^*(x) = f_*(\langle \theta_1^*, x \rangle, \dots, \langle \theta_k^*, x \rangle)$, where θ_j^* are well-chosen filters



Feature visualization of convolutional net trained on ImageNet from [Zeiler & Fergus 2013]

Problem Setting

Data model

- Gaussian inputs: $x \sim \mathcal{N}(0, I_d)$
- **Single-index** target model:

$$y = f_*(\langle \theta^*, x \rangle) + \xi, \qquad \text{with } \xi \sim \mathcal{N}(0, \sigma^2), \quad \|\theta^*\| = 1$$

Problem Setting

Data model

- Gaussian inputs: $x \sim \mathcal{N}(0, I_d)$
- **Single-index** target model:

$$y = f_*(\langle \theta^*, x \rangle) + \xi, \qquad \text{with } \xi \sim \mathcal{N}(0, \sigma^2), \quad \|\theta^*\| = 1$$

Network architecture

$$f_{c,\theta}(x) = c^\top \Phi(\langle \theta, x \rangle) = \frac{1}{\sqrt{N}} \sum_{i=1}^{N} c_i \phi(\langle \theta, x \rangle - b_i), \qquad \|\theta\| = 1$$

- $\phi(u) = \max(0, u)$: ReLU activation
- $b_i \sim \mathcal{N}(0, \tau^2)$: fixed, random biases

Problem Setting

Data model

- Gaussian inputs: $x \sim \mathcal{N}(0, I_d)$
- **Single-index** target model:

$$y = f_*(\langle \theta^*, x \rangle) + \xi, \qquad \text{with } \xi \sim \mathcal{N}(0, \sigma^2), \quad \|\theta^*\| = 1$$

Network architecture

$$f_{c,\theta}(x) = c^\top \Phi(\langle \theta, x \rangle) = \frac{1}{\sqrt{N}} \sum_{i=1}^{N} c_i \phi(\langle \theta, x \rangle - b_i), \qquad \|\theta\| = 1$$

- $\phi(u) = \max(0, u)$: ReLU activation
- $b_i \sim \mathcal{N}(0, \tau^2)$: fixed, random biases

Training algorithm: (projected) gradient descent on empirical loss

$$L_n(c, \theta) = \frac{1}{n} \sum_{i=1}^{n} (y_i - f_{c,\theta}(x_i))^2 + \lambda \|c\|^2$$

Warmup: landscape for teacher-student (Ben Arous et al., 2021)

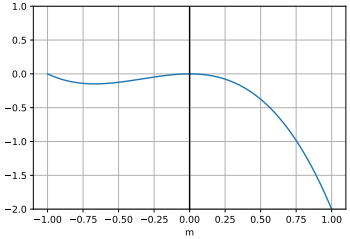
- Consider $f_* = \phi = \sum_j \alpha_j h_j$ known
- h_j: Hermite polynomials, with $\langle h_j, h_{j'} \rangle_\gamma = \delta_{jj'}$

Warmup: landscape for teacher-student (Ben Arous et al., 2021)

- Consider $f_* = \phi = \sum_j \alpha_j h_j$ known
- h_j: Hermite polynomials, with $\langle h_j, h_{j'} \rangle_\gamma = \delta_{jj'}$

Population loss:

$$\begin{aligned} L(\theta) &= \mathbb{E}_x[(f_*(\langle \theta, x \rangle) - f_*(\langle \theta^*, x \rangle))^2] \\ &= \text{cst} - 2\,\mathbb{E}_x[f_*(\langle \theta, x \rangle) f_*(\langle \theta^*, x \rangle)] \\ &= \text{cst} - 2 \sum_j {\alpha_j}^2 m^j, \qquad \text{with } m := \langle \theta, \theta^* \rangle \end{aligned}$$

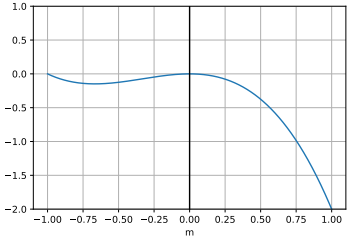


Warmup: landscape for teacher-student (Ben Arous et al., 2021)

- Consider $f_* = \phi = \sum_j \alpha_j h_j$ known
- h_j: Hermite polynomials, with $\langle h_j, h_{j'} \rangle_\gamma = \delta_{jj'}$

Population loss:

$$\begin{aligned} L(\theta) &= \mathbb{E}_x[(f_*(\langle \theta, x \rangle) - f_*(\langle \theta^*, x \rangle))^2] \\ &= \text{cst} - 2\,\mathbb{E}_x[f_*(\langle \theta, x \rangle) f_*(\langle \theta^*, x \rangle)] \\ &= \text{cst} - 2 \sum_j {\alpha_j}^2 m^j, \qquad \text{with } m := \langle \theta, \theta^* \rangle \end{aligned}$$



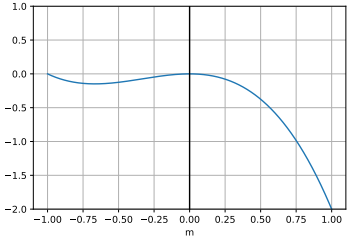
- **Information exponent** s: first non-zero j such that $\alpha_j \neq 0$

Warmup: landscape for teacher-student (Ben Arous et al., 2021)

- Consider $f_* = \phi = \sum_j \alpha_j h_j$ known
- h_j: Hermite polynomials, with $\langle h_j, h_{j'} \rangle_\gamma = \delta_{jj'}$

Population loss:

$$\begin{aligned} L(\theta) &= \mathbb{E}_x[(f_*(\langle \theta, x \rangle) - f_*(\langle \theta^*, x \rangle))^2] \\ &= \text{cst} - 2\,\mathbb{E}_x[f_*(\langle \theta, x \rangle) f_*(\langle \theta^*, x \rangle)] \\ &= \text{cst} - 2 \sum_j \alpha_j{}^2 m^j, \qquad \text{with } m := \langle \theta, \theta^* \rangle \end{aligned}$$



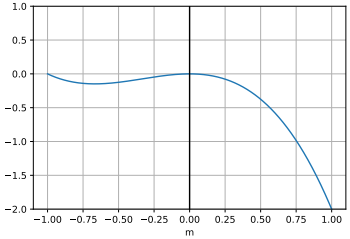
- **Information exponent** s: first non-zero j such that $\alpha_j \neq 0$
- Initialization near the "equator" ($m = 0$), $m \sim 1/\sqrt{d}$, $m > 0$ w.p. $1/2$
- The initial saddle m^s can be escaped with $n \gtrsim d^s$ samples

Warmup: landscape for teacher-student (Ben Arous et al., 2021)

- Consider $f_* = \phi = \sum_j \alpha_j h_j$ known
- h_j: Hermite polynomials, with $\langle h_j, h_{j'} \rangle_\gamma = \delta_{jj'}$

Population loss:

$$\begin{aligned} L(\theta) &= \mathbb{E}_x[(f_*(\langle \theta, x \rangle) - f_*(\langle \theta^*, x \rangle))^2] \\ &= \text{cst} - 2\,\mathbb{E}_x[f_*(\langle \theta, x \rangle) f_*(\langle \theta^*, x \rangle)] \\ &= \text{cst} - 2 \sum_j {\alpha_j}^2 m^j, \qquad \text{with } m := \langle \theta, \theta^* \rangle \end{aligned}$$



- **Information exponent** s: first non-zero j such that $\alpha_j \neq 0$
- Initialization near the "equator" ($m = 0$), $m \sim 1/\sqrt{d}$, $m > 0$ w.p. $1/2$
- The initial saddle m^s can be escaped with $n \gtrsim d^s$ samples
- Recovery $m \to 1$ is easy after that

What if we don't know f_*?

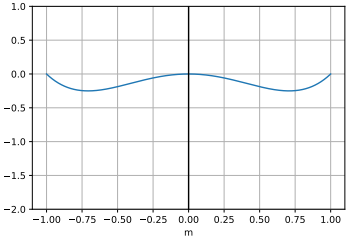
- $f_* = \sum_j \alpha_j h_j$ unknown
- $\phi = \sum_j \beta_j h_j$ known

What if we don't know f_*?

- $f_* = \sum_j \alpha_j h_j$ unknown
- $\phi = \sum_j \beta_j h_j$ known

Population loss

$$L(\theta) = \mathbb{E}_x[(g(\langle\theta, x\rangle) - f_*(\langle\theta^*, x\rangle))^2]$$
$$= \text{cst} - 2\sum_j \alpha_j \beta_j m^j, \qquad \text{with } m := \langle\theta, \theta^*\rangle$$

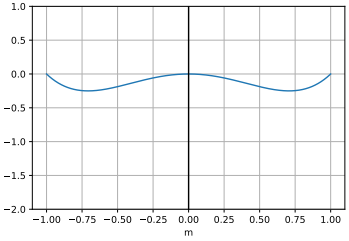


What if we don't know f_*?

- $f_* = \sum_j \alpha_j h_j$ unknown
- $\phi = \sum_j \beta_j h_j$ known

Population loss

$$L(\theta) = \mathbb{E}_x[(g(\langle\theta, x\rangle) - f_*(\langle\theta^*, x\rangle))^2]$$
$$= \text{cst} - 2\sum_j \alpha_j\beta_j m^j, \qquad \text{with } m := \langle\theta, \theta^*\rangle$$



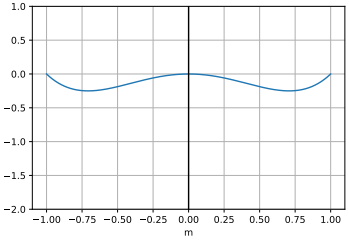
- If $\alpha_j\beta_j < 0$ for some j, may not recover $m \to 1$
- If $\alpha_s\beta_s > 0$, we may still reach $m \to \gamma \in (0, 1)$

What if we don't know f_*?

- $f_* = \sum_j \alpha_j h_j$ unknown
- $\phi = \sum_j \beta_j h_j$ known

Population loss

$$L(\theta) = \mathbb{E}_x[(g(\langle\theta, x\rangle) - f_*(\langle\theta^*, x\rangle))^2]$$
$$= \text{cst} - 2\sum_j \alpha_j\beta_j m^j, \qquad \text{with } m := \langle\theta, \theta^*\rangle$$



- If $\alpha_j\beta_j < 0$ for some j, may not recover $m \to 1$
- If $\alpha_s\beta_s > 0$, we may still reach $m \to \gamma \in (0, 1)$

This work: learn the β_j using random features! Hopefully $\beta_j \to \alpha_j$

Population Landscape with Random Features

- $f_* = \sum_j \alpha_j h_j$
- $f_{c,\theta}(x) = c^\top \Phi(\langle \theta, x \rangle) = \frac{1}{\sqrt{N}} \sum_{i=1}^N c_i \phi(\langle \theta, x \rangle - b_i)$

Population Landscape with Random Features

- $f_* = \sum_j \alpha_j h_j$
- $f_{c,\theta}(x) = c^\top \Phi(\langle \theta, x \rangle) = \frac{1}{\sqrt{N}} \sum_{i=1}^N c_i \phi(\langle \theta, x \rangle - b_i)$

Population loss

$$\begin{aligned} L(c, \theta) &= \mathbb{E}_x[(f_{c,\theta}(x) - f_*(\langle \theta^*, x \rangle))^2] + \lambda \|c\|^2 \\ &= \text{cst} + c^\top (Q + \lambda I) c - 2 \sum_j \alpha_j c^\top \mathcal{T}_j m^j \end{aligned}$$

Population Landscape with Random Features

- $f_* = \sum_j \alpha_j h_j$
- $f_{c,\theta}(x) = c^\top \Phi(\langle \theta, x \rangle) = \frac{1}{\sqrt{N}} \sum_{i=1}^N c_i \phi(\langle \theta, x \rangle - b_i)$

Population loss

$$\begin{aligned} L(c, \theta) &= \mathbb{E}_x[(f_{c,\theta}(x) - f_*(\langle \theta^*, x \rangle))^2] + \lambda \|c\|^2 \\ &= \text{cst} + c^\top (Q + \lambda I) c - 2 \sum_j \alpha_j c^\top \mathcal{T}_j m^j \end{aligned}$$

- $m = \langle \theta, \theta^* \rangle$

Population Landscape with Random Features

- $f_* = \sum_j \alpha_j h_j$
- $f_{c,\theta}(x) = c^\top \Phi(\langle \theta, x \rangle) = \frac{1}{\sqrt{N}} \sum_{i=1}^N c_i \phi(\langle \theta, x \rangle - b_i)$

Population loss

$$\begin{aligned} L(c, \theta) &= \mathbb{E}_x[(f_{c,\theta}(x) - f_*(\langle \theta^*, x \rangle))^2] + \lambda \|c\|^2 \\ &= \text{cst} + c^\top (Q + \lambda I) c - 2 \sum_j \alpha_j c^\top \mathcal{T}_j m^j \end{aligned}$$

- $m = \langle \theta, \theta^* \rangle$
- $\mathcal{T}_j = \mathcal{T} h_j$, where the operator $\mathcal{T} : L^2(\gamma) \to \mathbb{R}^N$ is given by

$$\mathcal{T} g = \frac{1}{\sqrt{N}} [\langle \phi(\cdot - b_i), g \rangle_\gamma]_i \in \mathbb{R}^N$$

Population Landscape with Random Features

- $f_* = \sum_j \alpha_j h_j$
- $f_{c,\theta}(x) = c^\top \Phi(\langle \theta, x \rangle) = \frac{1}{\sqrt{N}} \sum_{i=1}^N c_i \phi(\langle \theta, x \rangle - b_i)$

Population loss

$$\begin{aligned} L(c, \theta) &= \mathbb{E}_x[(f_{c,\theta}(x) - f_*(\langle \theta^*, x \rangle))^2] + \lambda \|c\|^2 \\ &= \text{cst} + c^\top (Q + \lambda I) c - 2 \sum_j \alpha_j c^\top \mathcal{T}_j m^j \end{aligned}$$

- $m = \langle \theta, \theta^* \rangle$
- $\mathcal{T}_j = \mathcal{T} h_j$, where the operator $\mathcal{T} : L^2(\gamma) \to \mathbb{R}^N$ is given by

$$\mathcal{T} g = \frac{1}{\sqrt{N}} [\langle \phi(\cdot - b_i), g \rangle_\gamma]_i \in \mathbb{R}^N$$

- $Q = \mathcal{T}\mathcal{T}^* \in \mathbb{R}^{N \times N}$: **covariance** matrix

Population landscape: Critical points

Theorem (Critical points)

Assume $\lambda < (s\alpha_s^2/C_{f_*})^{2/\beta}$ *and* $N \gtrsim C/\lambda$. *The first-order critical points of* $L(c, \theta)$ *satisfy one of the following:*

- $m = 0$, *i.e.,* $\langle \theta, \theta^* \rangle = 0$, *and* $c = 0$
- $m \in \{\pm 1\}$, *i.e.,* $\theta \in \{\pm \theta^*\}$, *and* $c = \arg\min_c L(c, \theta)$

Population landscape: Critical points

Theorem (Critical points)

Assume $\lambda < (s\alpha_s^2/C_{f_*})^{2/\beta}$ and $N \gtrsim C/\lambda$. The first-order critical points of $L(c, \theta)$ satisfy one of the following:

- $m = 0$, *i.e.*, $\langle \theta, \theta^* \rangle = 0$, *and* $c = 0$
- $m \in \{\pm 1\}$, *i.e.*, $\theta \in \{\pm \theta^*\}$, *and* $c = \arg\min_c L(c, \theta)$

- Uses approximation properties of the kernel $k(x, x') = \mathbb{E}_b[\phi(x - b)\phi(x' - b)]$:

$$\|(I - \hat{P}_\lambda) f\|_\gamma^2 \leq \lambda^\beta \|f''\|_\gamma^2 \quad \text{if } N \gtrsim C/\lambda$$

  - $\beta := \frac{\tau^2}{\tau^2 + 1}$
  - $\hat{P}_\lambda = \mathcal{T}^* \mathcal{T} (\mathcal{T}^* \mathcal{T} + \lambda I)^{-1}$ is a regularized projection operator on the RKHS of the random feature kernel (extends Bach, 2017b)

Population landscape: Critical points

Theorem (Critical points)

Assume $\lambda < (s\alpha_s^2/C_{f_*})^{2/\beta}$ and $N \gtrsim C/\lambda$. The first-order critical points of $L(c, \theta)$ satisfy one of the following:

- $m = 0$, *i.e.*, $\langle \theta, \theta^* \rangle = 0$, *and* $c = 0$
- $m \in \{\pm 1\}$, *i.e.*, $\theta \in \{\pm \theta^*\}$, *and* $c = \arg\min_c L(c, \theta)$

- Uses approximation properties of the kernel $k(x, x') = \mathbb{E}_b[\phi(x - b)\phi(x' - b)]$:

$$\|(I - \hat{P}_\lambda) f\|_\gamma^2 \leq \lambda^\beta \|f''\|_\gamma^2 \quad \text{if } N \gtrsim C/\lambda$$

  - $\beta := \frac{\tau^2}{\tau^2 + 1}$
  - $\hat{P}_\lambda = \mathcal{T}^*\mathcal{T}(\mathcal{T}^*\mathcal{T} + \lambda I)^{-1}$ is a regularized projection operator on the RKHS of the random feature kernel (extends Bach, 2017b)
- When λ is small enough, we may have $\alpha_j c^\top \mathcal{T}_j > 0$ for all j.

Gradient descent algorithm: intuition on population loss

$$L(c, \theta) = \text{cst} + c^\top (Q + \lambda I) c - 2 \sum_j \alpha_j c^\top \mathcal{T}_j m^j$$

Gradient descent algorithm: intuition on population loss

$$L(c, \theta) = \text{cst} + c^\top (Q + \lambda I) c - 2 \sum_j \alpha_j c^\top \mathcal{T}_j m^j$$

Random initialization $\theta \in \mathbb{S}^{d-1}$ and $c \in \rho \mathbb{S}^{N-1}$

- Anti-concentration at initialization: $|m| \gtrsim 1/\sqrt{d}, \quad |c^\top \mathcal{T}_s| \gtrsim \|c\| \|\mathcal{T}_s\| / \sqrt{N}.$

Gradient descent algorithm: intuition on population loss

$$L(c, \theta) = \text{cst} + c^\top (Q + \lambda I) c - 2 \sum_j \alpha_j c^\top \mathcal{T}_j m^j$$

Random initialization $\theta \in \mathbb{S}^{d-1}$ and $c \in \rho \mathbb{S}^{N-1}$

- Anti-concentration at initialization: $|m| \gtrsim 1/\sqrt{d}, \quad |c^\top \mathcal{T}_s| \gtrsim \|c\| \|\mathcal{T}_s\| / \sqrt{N}.$
- Assume $\alpha_s c^\top \mathcal{T}_s m^s > 0$ (prob. $1/2$ event)

Gradient descent algorithm: intuition on population loss

$$L(c, \theta) = \text{cst} + c^\top (Q + \lambda I) c - 2 \sum_j \alpha_j c^\top \mathcal{T}_j m^j$$

Random initialization $\theta \in \mathbb{S}^{d-1}$ and $c \in \rho \mathbb{S}^{N-1}$

- Anti-concentration at initialization: $|m| \gtrsim 1/\sqrt{d}, \quad |c^\top \mathcal{T}_s| \gtrsim \|c\| \|\mathcal{T}_s\| / \sqrt{N}.$
- Assume $\alpha_s c^\top \mathcal{T}_s m^s > 0$ (prob. $1/2$ event)

First phase: train only $\theta \implies m \to \gamma \in (0, 1]$

$$L(c, \theta) \approx \text{cst} - O(\alpha_s c^\top \mathcal{T}_s m^s)$$

Gradient descent algorithm: intuition on population loss

$$L(c, \theta) = \text{cst} + c^\top (Q + \lambda I) c - 2 \sum_j \alpha_j c^\top \mathcal{T}_j m^j$$

Random initialization $\theta \in \mathbb{S}^{d-1}$ and $c \in \rho \mathbb{S}^{N-1}$

- Anti-concentration at initialization: $|m| \gtrsim 1/\sqrt{d}, \quad |c^\top \mathcal{T}_s| \gtrsim \|c\| \|\mathcal{T}_s\| / \sqrt{N}.$
- Assume $\alpha_s c^\top \mathcal{T}_s m^s > 0$ (prob. $1/2$ event)

First phase: train only $\theta \implies m \to \gamma \in (0, 1]$

$$L(c, \theta) \approx \text{cst} - O(\alpha_s c^\top \mathcal{T}_s m^s)$$

- Initialization norm $\rho = \|c\|$ chosen to escape the level set of bad critical points

Gradient descent algorithm: intuition on population loss

$$L(c, \theta) = \text{cst} + c^\top (Q + \lambda I) c - 2 \sum_j \alpha_j c^\top \mathcal{T}_j m^j$$

Random initialization $\theta \in \mathbb{S}^{d-1}$ and $c \in \rho \mathbb{S}^{N-1}$

- Anti-concentration at initialization: $|m| \gtrsim 1/\sqrt{d}, \quad |c^\top \mathcal{T}_s| \gtrsim \|c\| \|\mathcal{T}_s\| / \sqrt{N}.$
- Assume $\alpha_s c^\top \mathcal{T}_s m^s > 0$ (prob. $1/2$ event)

First phase: train only $\theta \implies m \to \gamma \in (0, 1]$

$$L(c, \theta) \approx \text{cst} - O(\alpha_s c^\top \mathcal{T}_s m^s)$$

- Initialization norm $\rho = \|c\|$ chosen to escape the level set of bad critical points

Second phase: joint training of θ and c to a stationary point

Gradient descent algorithm: intuition on population loss

$$L(c, \theta) = \text{cst} + c^\top (Q + \lambda I) c - 2 \sum_j \alpha_j c^\top \mathcal{T}_j m^j$$

Random initialization $\theta \in \mathbb{S}^{d-1}$ and $c \in \rho \mathbb{S}^{N-1}$

- Anti-concentration at initialization: $|m| \gtrsim 1/\sqrt{d}, \quad |c^\top \mathcal{T}_s| \gtrsim \|c\| \|\mathcal{T}_s\| / \sqrt{N}.$
- Assume $\alpha_s c^\top \mathcal{T}_s m^s > 0$ (prob. $1/2$ event)

First phase: train only $\theta \implies m \to \gamma \in (0, 1]$

$$L(c, \theta) \approx \text{cst} - O(\alpha_s c^\top \mathcal{T}_s m^s)$$

- Initialization norm $\rho = \|c\|$ chosen to escape the level set of bad critical points

Second phase: joint training of θ and c to a stationary point

$\implies$ **must reach** $|m| \approx 1$ by previous theorem!

Gradient descent algorithm: intuition on population loss

$$L(c, \theta) = \text{cst} + c^\top (Q + \lambda I) c - 2 \sum_j \alpha_j c^\top \mathcal{T}_j m^j$$

Random initialization $\theta \in \mathbb{S}^{d-1}$ and $c \in \rho \mathbb{S}^{N-1}$

- Anti-concentration at initialization: $|m| \gtrsim 1/\sqrt{d}, \quad |c^\top \mathcal{T}_s| \gtrsim \|c\| \|\mathcal{T}_s\| / \sqrt{N}.$
- Assume $\alpha_s c^\top \mathcal{T}_s m^s > 0$ (prob. $1/2$ event)

First phase: train only $\theta \implies m \to \gamma \in (0, 1]$

$$L(c, \theta) \approx \text{cst} - O(\alpha_s c^\top \mathcal{T}_s m^s)$$

- Initialization norm $\rho = \|c\|$ chosen to escape the level set of bad critical points

Second phase: joint training of θ and c to a stationary point

$\implies$ **must reach** $|m| \approx 1$ by previous theorem!

Final fine-tuning phase: re-train second layer c on n' fresh samples with suitable $\lambda_{n'}$

- optional, but needed for better rates

Generalization Guarantees

Theorem (Excess risk bound (informal))

First/second phase: n *samples, assume* $\lambda \approx (s\alpha_s^2/C_{f_*})^{2/\beta}$, $n \gtrsim d^s/\lambda$.

Fine-tuning phase: n' *samples, assume* $\lambda_{n'} \leq (1/n')^{\frac{1}{\beta+1}}$, *and set* $N \gtrsim C \max\{1/\lambda, 1/\lambda_{n'}\}$.

With probability close to $1/2$, *the final* $\hat{F} = f_{\hat{c},\hat{\theta}}$ *satisfies*

$$\mathbb{E}_x[(\hat{F}(x) - F^*(x))^2] \lesssim \left(\frac{d}{n}\right)^2 + \left(\frac{1}{n'}\right)^{\frac{\beta}{\beta+1}}$$

Generalization Guarantees

Theorem (Excess risk bound (informal))

First/second phase: n *samples, assume* $\lambda \approx (s\alpha_s^2/C_{f_*})^{2/\beta}$, $n \gtrsim d^s/\lambda$.

Fine-tuning phase: n' *samples, assume* $\lambda_{n'} \leq (1/n')^{\frac{1}{\beta+1}}$, *and set* $N \gtrsim C\max\{1/\lambda, 1/\lambda_{n'}\}$.

With probability close to $1/2$, *the final* $\hat{F} = f_{\hat{c},\hat{\theta}}$ *satisfies*

$$\mathbb{E}_x[(\hat{F}(x) - F^*(x))^2] \lesssim \left(\frac{d}{n}\right)^2 + \left(\frac{1}{n'}\right)^{\frac{\beta}{\beta+1}}$$

- Dynamics on empirical loss rely on landscape concentration results (Mei et al., 2016).

Generalization Guarantees

Theorem (Excess risk bound (informal))

First/second phase: n *samples, assume* $\lambda \approx (s\alpha_s^2/C_{f_*})^{2/\beta}$, $n \gtrsim d^s/\lambda$.

Fine-tuning phase: n' *samples, assume* $\lambda_{n'} \leq (1/n')^{\frac{1}{\beta+1}}$, *and set* $N \gtrsim C\max\{1/\lambda, 1/\lambda_{n'}\}$.

With probability close to $1/2$, *the final* $\hat{F} = f_{\hat{c},\hat{\theta}}$ *satisfies*

$$\mathbb{E}_x[(\hat{F}(x) - F^*(x))^2] \lesssim \left(\frac{d}{n}\right)^2 + \left(\frac{1}{n'}\right)^{\frac{\beta}{\beta+1}}$$

- Dynamics on empirical loss rely on landscape concentration results (Mei et al., 2016).
- Recovery of θ^* is near-optimal: $n \gtrsim d^s$ almost matches (Ben Arous et al., 2021).

Generalization Guarantees

Theorem (Excess risk bound (informal))

First/second phase: n *samples, assume* $\lambda \approx (s\alpha_s^2/C_{f_*})^{2/\beta}$, $n \gtrsim d^s/\lambda$.

Fine-tuning phase: n' *samples, assume* $\lambda_{n'} \leq (1/n')^{\frac{1}{\beta+1}}$, *and set* $N \gtrsim C\max\{1/\lambda, 1/\lambda_{n'}\}$.

With probability close to $1/2$, *the final* $\hat{F} = f_{\hat{c},\hat{\theta}}$ *satisfies*

$$\mathbb{E}_x[(\hat{F}(x) - F^*(x))^2] \lesssim \left(\frac{d}{n}\right)^2 + \left(\frac{1}{n'}\right)^{\frac{\beta}{\beta+1}}$$

- Dynamics on empirical loss rely on landscape concentration results (Mei et al., 2016).
- Recovery of θ^* is near-optimal: $n \gtrsim d^s$ almost matches (Ben Arous et al., 2021).
- Fine-tuning recovers 1D non-parametric rates from kernel methods for fitting f_*.

Generalization Guarantees

Theorem (Excess risk bound (informal))

First/second phase: n *samples, assume* $\lambda \approx (s\alpha_s^2/C_{f_*})^{2/\beta}$, $n \gtrsim d^s/\lambda$.

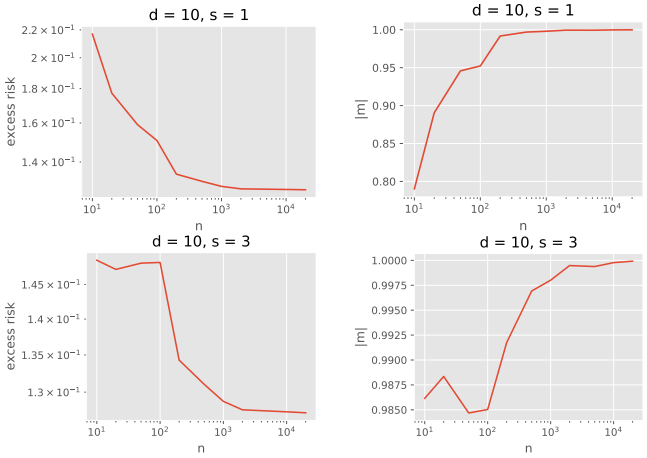
Fine-tuning phase: n' *samples, assume* $\lambda_{n'} \leq (1/n')^{\frac{1}{\beta+1}}$, *and set* $N \gtrsim C\max\{1/\lambda, 1/\lambda_{n'}\}$.

With probability close to $1/2$, *the final* $\hat{F} = f_{\hat{c},\hat{\theta}}$ *satisfies*

$$\mathbb{E}_x[(\hat{F}(x) - F^*(x))^2] \lesssim \left(\frac{d}{n}\right)^2 + \left(\frac{1}{n'}\right)^{\frac{\beta}{\beta+1}}$$

- Dynamics on empirical loss rely on landscape concentration results (Mei et al., 2016).
- Recovery of θ^* is near-optimal: $n \gtrsim d^s$ almost matches (Ben Arous et al., 2021).
- Fine-tuning recovers 1D non-parametric rates from kernel methods for fitting f_*.
- Without fine-tuning, we can still obtain vanishing excess risk, but at slower rate.

Preliminary Experiments



First/second phase with piece-wise linear teacher f_*

Conclusions and Perspectives

Efficient learning of single-index models

- Shallow networks with tied neuron directions and random biases
- Combines **feature learning** of θ^* with **non-parametric** 1D estimation of f_*

Conclusions and Perspectives

Efficient learning of single-index models

- Shallow networks with tied neuron directions and random biases
- Combines **feature learning** of θ^* with **non-parametric** 1D estimation of f_*

Further questions

- What if we train (c, θ) jointly from the start?
- SGD on the population loss?
- Untied neuron directions?
- Training the biases?
- Multi-index models?
- Is fine-tuning necessary for good rates?

Conclusions and Perspectives

Efficient learning of single-index models

- Shallow networks with tied neuron directions and random biases
- Combines **feature learning** of θ^* with **non-parametric** 1D estimation of f_*

Further questions

- What if we train (c, θ) jointly from the start?
- SGD on the population loss?
- Untied neuron directions?
- Training the biases?
- Multi-index models?
- Is fine-tuning necessary for good rates?

Thank you!

References I

F. Bach. Breaking the curse of dimensionality with convex neural networks. *Journal of Machine Learning Research (JMLR)*, 18(19):1–53, 2017a.

F. Bach. On the equivalence between kernel quadrature rules and random feature expansions. *Journal of Machine Learning Research (JMLR)*, 18(21):1–38, 2017b.

G. Ben Arous, R. Gheissari, and A. Jagannath. Online stochastic gradient descent on non-convex losses from high-dimensional inference. *Journal of Machine Learning Research (JMLR)*, 2021.

A. Caponnetto and E. De Vito. Optimal rates for the regularized least-squares algorithm. *Foundations of Computational Mathematics*, 7(3):331–368, 2007.

L. Chizat and F. Bach. On the global convergence of gradient descent for over-parameterized models using optimal transport. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

R. Dudeja and D. Hsu. Learning single-index models in gaussian space. In S. Bubeck, V. Perchet, and P. Rigollet, editors, *Proceedings of the 31st Conference On Learning Theory*, volume 75 of *Proceedings of Machine Learning Research*, pages 1887–1930. PMLR, 06–09 Jul 2018. URL `https://proceedings.mlr.press/v75/dudeja18a.html`.

S. Mei, Y. Bai, and A. Montanari. The landscape of empirical risk for non-convex losses. *arXiv preprint arXiv:1607.06534*, 2016.

References II

S. Mei, T. Misiakiewicz, and A. Montanari. Mean-field theory of two-layers neural networks: dimension-free bounds and kernel limit. In *Conference on Learning Theory (COLT)*, 2019.

M. Soltanolkotabi. Learning relus via gradient descent. *Advances in neural information processing systems*, 30, 2017.