Learning Single-Index Models with Shallow Neural Networks

Alberto Bietti[a,b], Joan Bruna[a], Clayton Sanford[c], Min Jae Song[a]

[a]New York University, [b]Meta AI, [c]Columbia University

NEURAL INFORMATION PROCESSING SYSTEMS

Motivation

Motivating question: How do neural networks (NNs) provably outperform linear models (e.g., kernel methods)?

- **Universal approximation** properties of NNs are well-known, but these say little about what is learnable by *gradient descent* (GD).
- Under certain scalings, very wide NNs trained with GD converge to the kernel ridge regression (KRR) predictor with respect to the **Neural Tangent Kernel (NTK)**. However, first-layer weights are nearly fixed in the NTK regime.
- In practice, gradient descent learns "features" and adapts to low-dimensional structure present in the data.
- Understanding how NNs outperform linear models requires understanding feature learning and going beyond NTK.

We prove the adaptability of certain two-layer NNs to low-dimensional structure via feature learning.

Problem Setting

Data model

- **Gaussian covariates.** d-dimensional samples $x \sim \gamma_d$, where $\gamma_d = \mathcal{N}(0, I_d)$.
- **Single-index model.** $y = f_*(\langle \theta^*, x \rangle) + \xi$, where $\|\theta^*\| = 1$ and $\xi \sim \mathcal{N}(0, \sigma^2)$ is independent label noise.
- **Information exponent** of f_*, which we denote by $s \in \mathbb{N}$, is the index of the smallest non-zero Hermite coefficient of f_*.
- **Training data.** n i.i.d. samples $(x_i, y_i)_{i \in [n]}$.

Network architecture

- Depth-two width-N ReLU network with tied first-layer weights $\theta \in \mathbb{S}^{d-1}$:

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

- Rectified Linear Unit (ReLU) activation: $\varphi : z \mapsto \max\{0, z\}$.
- **Frozen** i.i.d. random biases and signs: $b_i \sim \mathcal{N}(0, \tau^2)$, $\varepsilon_i \sim \mathrm{Unif}(\{\pm 1\})$. ($\theta$ and c are randomly initialized and trained.)

Training algorithm

- **Projected gradient flow (PGF)** on **regularized empirical loss**:

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

- For $t \in [0, T_0]$, only optimize θ. Train (c, θ) jointly afterwards.

$$\dot{c}(t) = -\mathbf{1}\{t > T_0\} \nabla_c L_n(c, \theta),$$
$$\dot{\theta}(t) = -\nabla_\theta^{\mathbb{S}^{d-1}} L_n(c, \theta).$$

- $\nabla_\theta^{\mathbb{S}^{d-1}}$ is a **spherical gradient** that ensures that the (shared) first-layer weight θ remains on the unit sphere $\mathbb{S}^{d-1}$.

Population Loss Landscape

Population loss: $L(c, \theta) = \mathbb{E}_{(x,y)}[(y - f_{c,\theta}(x))^2] + \lambda \|c\|^2$, where $y = f_*(\langle \theta^*, x \rangle) + \xi$.

Theorem (Critical points of $L(c, \theta)$)

Under regularity assumptions on f_*, for sufficiently small $\lambda > 0$ and $N \gtrsim \frac{1}{\lambda}$, if $\nabla_c L(c, \theta) = 0$ and $\nabla_\theta^{\mathbb{S}^{d-1}} L(c, \theta) = 0$, then (c, θ) is either

1. ***Bad:*** *$m := \langle \theta, \theta^* \rangle = 0$ and $c = 0$; or*
2. ***Good:*** *$m \in \{\pm 1\}$ and $c = \arg\min_c L(c, \theta)$.*

Key proof idea: Show that (projected) gradients depend only on m, and the Hermite coefficients of activation φ and target link f_*.

Recovery of θ^*

Can we recover the direction θ^* in the first layer weights of our NN?

Theorem (Projected gradient flow recovers θ^*)

If $\lambda = \Theta(1)$, $N = \Theta(\frac{1}{\lambda})$, and $n = \Omega(\max\{d^s, d^{\frac{s+3}{2}}\})$, then the following holds with probability at least 0.49.

$$|\langle \theta(T), \theta^* \rangle| \geq 1 - \tilde{O}\left(\max\left\{\frac{d}{n}, \frac{d^4}{n^2}\right\}\right).$$

Proof intuition:

- Uniform convergence of the empirical loss landscape to its population counterpart.
- Topological properties of L are inherited by L_n. Critical points of L_n split into "bad" ones on the equator and "good" ones at the poles.
- With large sample size n, gradient flow escapes the equatorial region ($m \approx 0$) and converges to stable critical points at poles ($|m| \approx 1$). Larger information exponent s requires larger n to escape.

Fine-tuning for Improved Rates

Fine-tuning: After PGF terminates, draw n' new samples and use KRR to obtain $\hat{c}$ with new regularization $\lambda_{n'}$: $\hat{c} = \arg\min_c L_{n'}(c, \theta(T))$.

Theorem (PGF with fine-tuning converges to F_*)

After completing PGF as above, fine-tuning with n' additional samples, appropriate $\lambda_{n'}$ and width N' produces $\hat{c} \in \mathbb{R}^{N'}$ satisfying

$$\mathbb{E}_{n'}[\|f_{\hat{c},\theta(T)} - F_*\|_{\gamma^{\otimes d}}^2] \leq \tilde{O}\left(\max\left\{\frac{d}{n}, \frac{d^4}{n^2}\right\} + (n')^{-\frac{\beta}{\beta+1}}\right),$$

for $\beta = \frac{1 - 1/\tau^2}{3 + 1/\tau^2}$.

Experimental Validation

- Validated theoretical results on synthetic f_* with information exponent $s \in \{1, 2, 3\}$, width $N = 100$, and varying n and d.
- Larger s requires larger n to escape equator and achieve $|\langle \theta^*, \theta(T) \rangle| \approx 1$.

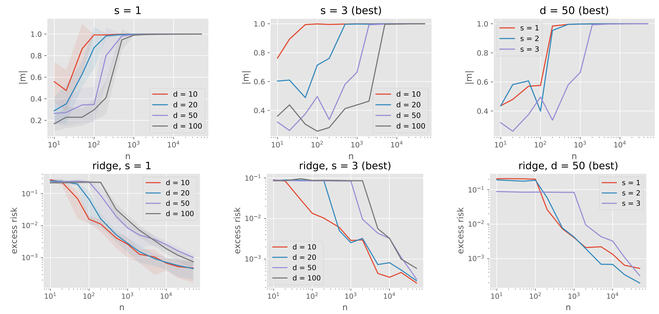

Figure: Correlation $|m|$ (top row) and excess risk $\|\hat{F} - F^*\|_{\gamma_d}^2$ with final ridge/fine-tuning step (bottom row) as a function of sample size n.

Conclusion and Future Work

- We analyze *joint* training of first and second layer weights (c, θ) via uniform convergence of the empirical landscape.
- Because our NN learns the "feature" θ^* of the single-index model, it requires much smaller width N compared to methods that do not perform feature learning (e.g., random features).
- Our sample complexity for recovering θ^* is near-optimal since $n \gtrsim d^s$ is necessary for target link f_* with information exponent s using SGD (Ben-Arous et al., 2021).
- **Q1.** If first layer weights are *not* shared, will they all converge to either the poles ($|m| \approx 1$) or the equator ($|m| \approx 0$)?
- **Q2.** Can we learn multi-index models ($F_*(x) = f_*(\langle \theta_1^*, x \rangle, \ldots, \langle \theta_r^*, x \rangle)$) with shallow neural networks?
- **Q3.** Differences between multi-pass GD and online SGD?

Full Version

- For more details, **check out**:

https://arxiv.org/abs/2210.15651