Memorization With Neural Nets: Going Beyond the Worst Case

Sjoerd Dirksen[1], **Patrick Finke**[1], Martin Genzel[2]

[1]Utrecht University [2]Merantix Momentum (work done while at Utrecht University)

Motivation and Problem Setup

Memorization Capacity as a Worst-Case

How big does a neural network F_θ need to be to **memorize** N points, i.e.,

$$\forall\{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_N, y_N)\} \subset \mathbb{R}^d \times \{\pm 1\}\ \exists \text{parameters } \theta\colon F_\theta(\mathbf{x}_i) = y_i\ \forall i \in [N].$$

- This requires interpolation of unstructured data, including random noise.
- The network size must depend on the number of samples.

An Instance-Specific Approach

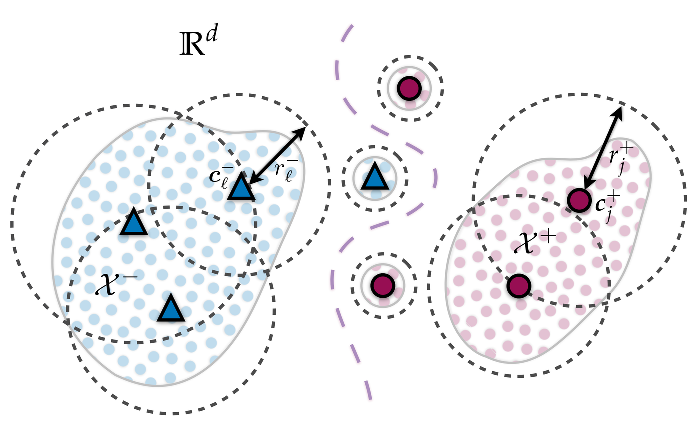
Let $\mathcal{X}^-, \mathcal{X}^+ \subset \mathbb{R}^d$ be disjoint and finite sets, representing two classes of objects.

> A classification function $F\colon \mathbb{R}^d \to \{\pm 1\}$ **interpolates** $\mathcal{X}^-$ and $\mathcal{X}^+$ if
> $$F(\mathbf{x}^-) = -1 \quad \text{and} \quad F(\mathbf{x}^+) = +1$$
> for all $\mathbf{x}^- \in \mathcal{X}^-$ and $\mathbf{x}^+ \in \mathcal{X}^+$.

- How big does a neural network need to be to interpolate a specific dataset?
- How does the size relate to the mutual geometric complexity of the classes?
- Is there an algorithm to obtain an interpolating network for given data?

Mutual Complexity

> A **mutual covering** of $\mathcal{X}^-$ and $\mathcal{X}^+$ consists of two sets of *components*
> $\mathcal{X}_\ell^- := \mathcal{X}^- \cap \mathbb{B}_2^d(\mathbf{c}_\ell^-, r_\ell^-)$, $\ell \in [M^-]$,
> $\mathcal{X}_j^+ := \mathcal{X}^+ \cap \mathbb{B}_2^d(\mathbf{c}_j^+, r_j^+)$, $j \in [M^+]$,
> so that each covers its respective class and satisfies (a) and (b) below.



(a) δ-separated centers, i.e., $\left\|\mathbf{c}_\ell^- - \mathbf{c}_j^+\right\|_2 \geq \delta$ for all $\ell \in [M^-], j \in [M^+]$.

(b) The component radii adapt to the mutual arrangement of the classes, i.e.,

$$r_\ell^- \lesssim \frac{d(\mathbf{c}_\ell^-, \mathcal{C}^+)}{\log^{1/2}(e\lambda / d(\mathbf{c}_\ell^-, \mathcal{C}^+))} \quad \text{and} \quad r_j^+ \lesssim \frac{d(\mathbf{c}_j^+, \mathcal{C}^-)}{\log^{1/2}(e\lambda / d(\mathbf{c}_j^+, \mathcal{C}^-))}.$$

- Covering numbers M^- and M^+ capture global complexity,
- Maximal component 'size' $\omega := \{\omega^-, \omega^+\}$ captures local complexity.

$$\omega^- := \max_{\ell \in [M^-]} \frac{w^2(\mathcal{X}_\ell^- - \mathbf{c}_\ell^-)}{d^3(\mathbf{c}_\ell^-, \mathcal{C}^+)} \quad \text{and} \quad \omega^+ := \max_{j \in [M^+]} \frac{w^2(\mathcal{X}_j^+ - \mathbf{c}_j^+)}{d^3(\mathbf{c}_j^+, \mathcal{C}^-)}.$$

Gaussian mean width: $w(\mathcal{A}) := \mathbb{E}_{\mathbf{g} \sim N(\mathbf{0}, \mathbf{I}_d)}\left[\sup_{\mathbf{x} \in \mathcal{A}} |\langle \mathbf{g}, \mathbf{x} \rangle|\right]$

Main Result

> **Theorem (Informal).** Let $\mathcal{X}^-, \mathcal{X}^+ \subset R\mathbb{B}_2^d$ be finite and disjoint. Suppose that there is a mutual covering. Then, w.h.p., our algorithm outputs a three-layer fully-connected neural network with threshold activations and
> - $\mathcal{O}\left(M^- + R\delta^{-1}\log(2M^-M^+) + R\omega\right)$ neurons,
> - $\mathcal{O}\left(R(d + M^-)(\delta^{-1}\log(2M^-M^+) + \omega)\right)$ parameters,
>
> that interpolates $\mathcal{X}^-$ and $\mathcal{X}^+$.

- Our result is 'problem-dependent' but independent of the number of samples.
- Linear dependence on M^- but only logarithmic dependence on M^+.
- General activations, e.g., ReLU ($\sigma(t) = 0, t \leq 0$ and $\sigma(t) > 0, t > 0$).

Algorithm

Input: $\mathcal{X}^-, \mathcal{X}^+ \subset \mathbb{R}^d$ disjoint and finite, width $n \geq 1$, maximal bias $\lambda \geq 0$.
Output: Three-layer fully-connected neural network $F\colon \mathbb{R}^d \to \{\pm 1\}$.

A.1: Randomly sample $\mathbf{W} \in \mathbb{R}^{n \times d}$ and $\mathbf{b} \in \mathbb{R}^n$ where

$$\mathbf{W}_i \sim N(\mathbf{0}, \mathbf{I}_d) \quad \text{and} \quad b_i \sim \mathrm{Unif}([-\lambda, \lambda])$$

are all independent and define $\Phi(\mathbf{x}) = \mathit{Thres}(\mathbf{W}\mathbf{x} + \mathbf{b})$.

B.1: Initialize $\mathcal{C} := \mathcal{X}^-$, $\mathcal{U} := \mathcal{X}^-$ and $\mathcal{A} := \emptyset$.

B.2: While $\mathcal{C} \neq \emptyset$ and $\mathcal{U} \neq \emptyset$ do

B.3: Select $\mathbf{x}_*^- \in \mathcal{C}$ at random and update $\mathcal{C} := \mathcal{C} \setminus \{\mathbf{x}_*^-\}$.

B.4: Calculate $\mathbf{u}_{\mathbf{x}_*^-} \in \{0,1\}^n$ and $m_{\mathbf{x}_*^-} \geq 0$ according to

$$\mathbf{u}_{\mathbf{x}_*^-} = 1[\Phi(\mathbf{x}_*^-) = \mathbf{0}] \quad \text{and} \quad m_{\mathbf{x}_*^-} = \min_{\mathbf{x}^+ \in \mathcal{X}^+} \langle \mathbf{u}_{\mathbf{x}_*^-}, \Phi(\mathbf{x}^+) \rangle$$

and set $\mathcal{T} := \{\mathbf{x}^- \in \mathcal{U} : \langle \mathbf{u}_{\mathbf{x}_*^-}, \Phi(\mathbf{x}^-) \rangle < m_{\mathbf{x}_*^-}\}$.

B.5: If $|\mathcal{T}| > 0$, update $\mathcal{C} := \mathcal{C} \setminus \mathcal{T}$, $\mathcal{U} := \mathcal{U} \setminus \mathcal{T}$, and $\mathcal{A} := \mathcal{A} \cup \{\mathbf{x}_*^-\}$.

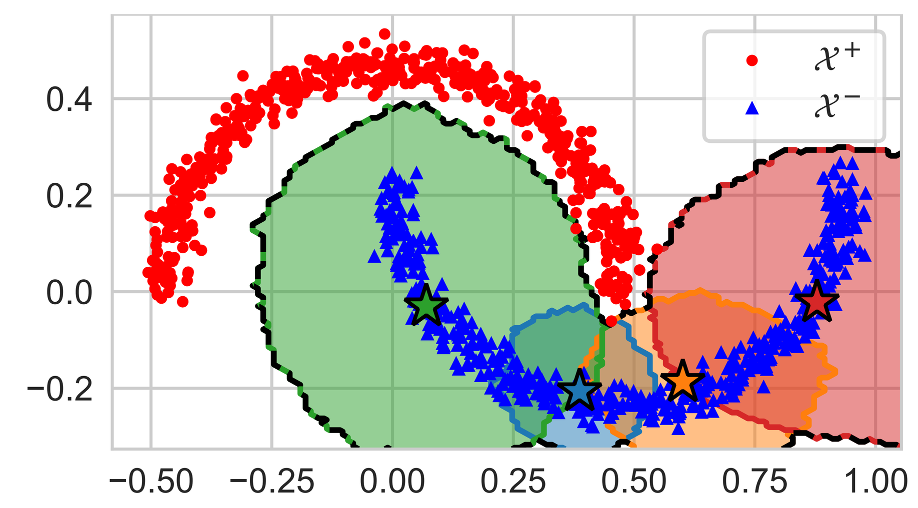
B.6: Define $\hat{\Phi}(\mathbf{z}) = \mathit{Thres}(-\mathbf{U}\mathbf{z} + \mathbf{m})$ where

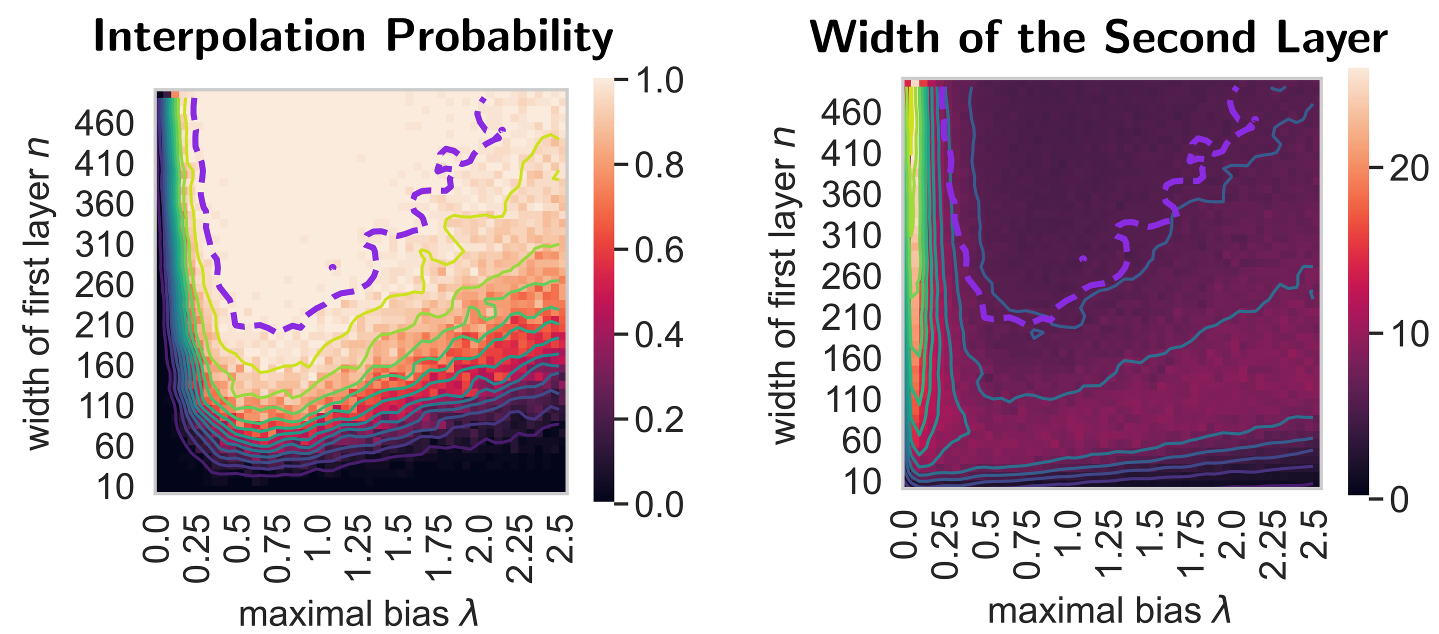
$$\mathbf{U} \leftarrow \left[\mathbf{u}_{\mathbf{x}_*^-}^\top\right]_{\mathbf{x}_*^- \in \mathcal{A}} \quad \text{and} \quad \mathbf{m} \leftarrow \left[m_{\mathbf{x}_*^-}\right]_{\mathbf{x}_*^- \in \mathcal{A}}.$$

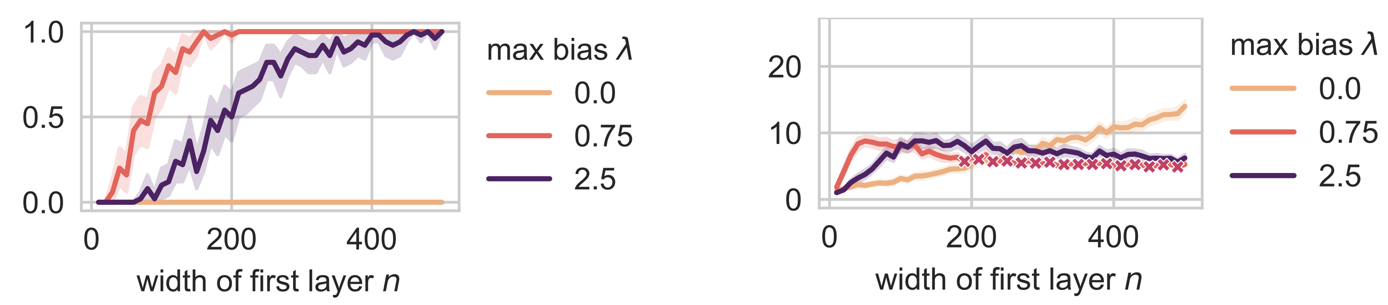
C.1: Return $F(\mathbf{x}) = \mathit{sign}(-\langle \mathbf{1}, \hat{\Phi}(\Phi(\mathbf{x})) \rangle)$.

> **Theorem.** Let $\mathcal{X}^-, \mathcal{X}^+ \subset R\mathbb{B}_2^d$ be finite and disjoint. Suppose that there is a mutual covering. Assume that
> $$\lambda \gtrsim R \quad \text{and} \quad n \gtrsim \lambda\delta^{-1}\log(2M^-M^+/\eta) + \lambda\omega.$$
> Then, with probability at least $1 - \eta$, our algorithm outputs a three-layer fully-connected neural network with threshold activations that interpolates $\mathcal{X}^-$ and $\mathcal{X}^+$ and its second layer has width at most M^-.

Numerical Verification







Proof Idea

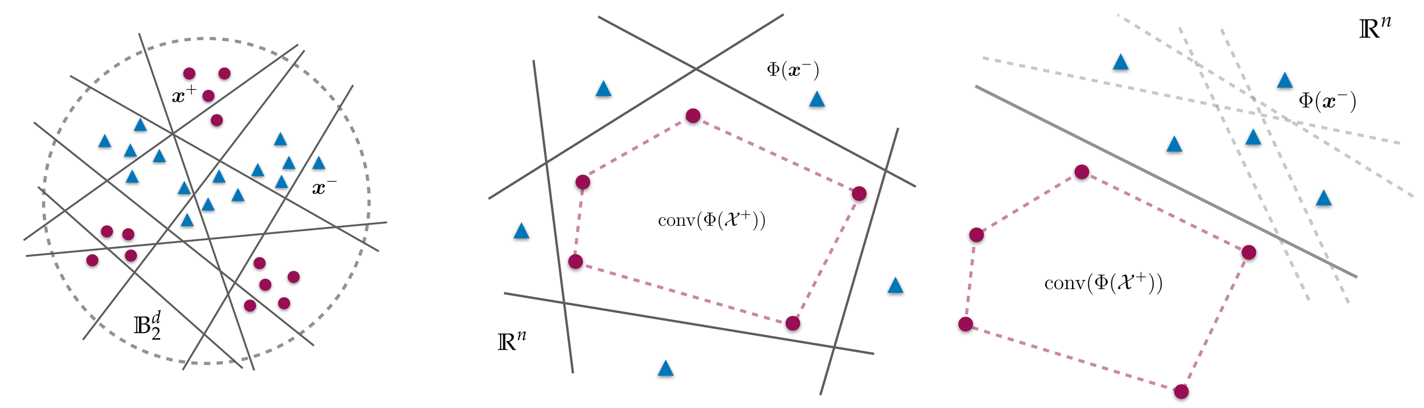
- Interplay between separation and distance-preserving properties of random NN layers.
- Related to the separation capacity of random neural networks. [Dirksen et al. '22]

Step 1: tessellation of the intput space with random hyperplanes

Step 2: separate 'satellites' from 'planet' via dedicated deterministic hyperplanes

Step 3: one hyperplane is enough to separate a whole component

Step 4: greedy forward selection of hyperplanes until all points are separated



Find the preprint on arXiv: