# Memorization With Neural Nets: Going Beyond the Worst Case

Patrick Finke (Utrecht University)

based on joint work with

Sjoerd Dirksen (Utrecht University)
Martin Genzel (Helmholtz-Zentrum Berlin)

AIM networking event
December 1, 2022

# Motivation

Fully-connected neural net:

$F = \Phi_L \circ \cdots \circ \Phi_1$ composition of layers $\Phi_\ell(\boldsymbol{x}) = \sigma(\boldsymbol{W}_\ell \boldsymbol{x} + \boldsymbol{b}_\ell)$

# Motivation

## Fully-connected neural net:

$F = \Phi_L \circ \cdots \circ \Phi_1$ composition of layers $\Phi_\ell(\boldsymbol{x}) = \sigma(\boldsymbol{W}_\ell \boldsymbol{x} + \boldsymbol{b}_\ell)$

## Memorization capacity:

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

$$\forall \boldsymbol{x}_i \in \mathbb{R}^d, y_i \in \{\pm 1\} \ \exists \text{parameters}: F(\boldsymbol{x}_i) = y_i, \ i \in [N].$$

# Motivation

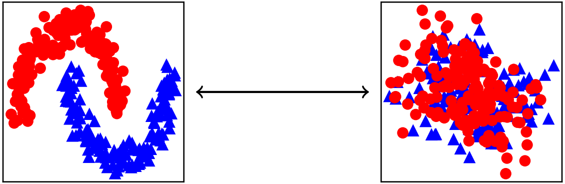
## Fully-connected neural net:

$F = \Phi_L \circ \cdots \circ \Phi_1$ composition of layers $\Phi_\ell(\boldsymbol{x}) = \sigma(\boldsymbol{W}_\ell \boldsymbol{x} + \boldsymbol{b}_\ell)$

## Memorization capacity:

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

$$\forall \boldsymbol{x}_i \in \mathbb{R}^d, y_i \in \{\pm 1\} \; \exists \text{parameters}\colon F(\boldsymbol{x}_i) = y_i, \; i \in [N].$$



structured — unstructured

Memorization capacity amounts to a *worst-case analysis*.

# Motivation

## Instance-specific viewpoint:

How big does a neural net need to be to memorize a *specific dataset*?

# Motivation

## Instance-specific viewpoint:

How big does a neural net need to be to memorize a *specific dataset*?

## Our approach:

1. *randomized algorithm* that produces a memorizing neural net
2. link size of net to *geometric properties* of the classes and their *mutual arrangement*

# Motivation

## Instance-specific viewpoint:

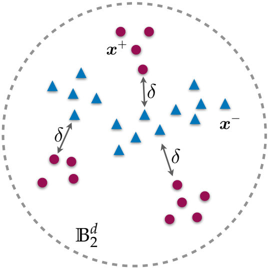
How big does a neural net need to be to memorize a *specific dataset*?

## Our approach:

1. *randomized algorithm* that produces a memorizing neural net
2. link size of net to *geometric properties* of the classes and their *mutual arrangement*

## Setup:

- $\mathcal{X}^-, \mathcal{X}^+ \subset \mathbb{B}_2^d$
- finite
- $\delta$-separated



# Algorithm

Assume $\sigma(t) = \mathrm{Thres}(t) = \mathbb{1}[t \geq 0]$ $(\mathrm{ReLU}$ possible)

Algorithm:

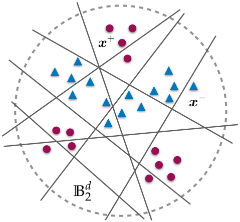
# Algorithm

Assume $\sigma(t) = \mathrm{Thres}(t) = \mathbb{1}[t \geq 0]$ $(\mathrm{ReLU}$ possible$)$

## Algorithm:

1. Randomly sample $\Phi\colon \mathbb{R}^d \to \mathbb{R}^n$

$$\forall \boldsymbol{x}^-, \boldsymbol{x}^+ \ \exists i \in [n]\colon$$
$$[\Phi(\boldsymbol{x}^-)]_i = 0, \ [\Phi(\boldsymbol{x}^+)]_i > 0$$



# Algorithm

Assume $\sigma(t) = \mathrm{Thres}(t) = \mathbb{1}[t \geq 0]$ (ReLU possible)

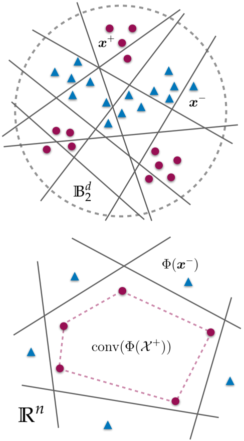
Algorithm:

1. Randomly sample $\Phi \colon \mathbb{R}^d \to \mathbb{R}^n$

$$\forall \boldsymbol{x}^-, \boldsymbol{x}^+ \ \exists i \in [n] \colon$$
$$[\Phi(\boldsymbol{x}^-)]_i = 0, \ [\Phi(\boldsymbol{x}^+)]_i > 0$$

2. Construct $\hat{\Phi} \colon \mathbb{R}^n \to \mathbb{R}^{|\mathcal{X}^-|}$

$$\forall \boldsymbol{x}^- \quad \colon [\hat{\Phi}(\Phi(\boldsymbol{x}^-))]_{\boldsymbol{x}^-} > 0$$
$$\forall \boldsymbol{x}^-, \boldsymbol{x}^+ \colon [\hat{\Phi}(\Phi(\boldsymbol{x}^+))]_{\boldsymbol{x}^-} = 0$$



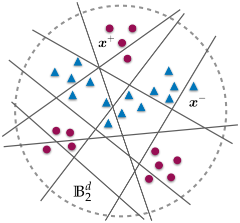
# Algorithm

Assume $\sigma(t) = \mathrm{Thres}(t) = \mathbb{1}[t \geq 0]$ (ReLU possible)

## Algorithm:

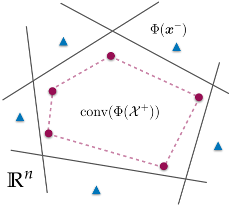
1. Randomly sample $\Phi\colon \mathbb{R}^d \to \mathbb{R}^n$

$$\forall \boldsymbol{x}^-, \boldsymbol{x}^+ \ \exists i \in [n]:$$
$$[\Phi(\boldsymbol{x}^-)]_i = 0, \ [\Phi(\boldsymbol{x}^+)]_i > 0$$



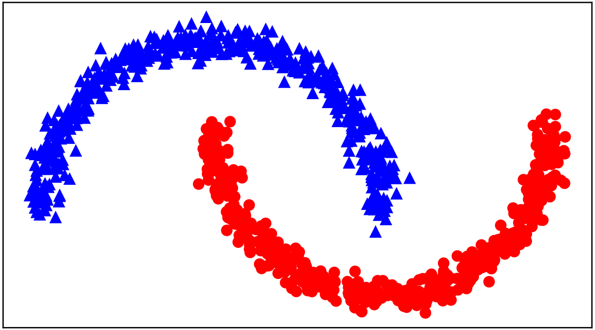

2. Construct $\hat{\Phi}\colon \mathbb{R}^n \to \mathbb{R}^{|\mathcal{X}^-|}$

$$\forall \boldsymbol{x}^- \quad : [\hat{\Phi}(\Phi(\boldsymbol{x}^-))]_{\boldsymbol{x}^-} > 0$$
$$\forall \boldsymbol{x}^-, \boldsymbol{x}^+ : [\hat{\Phi}(\Phi(\boldsymbol{x}^+))]_{\boldsymbol{x}^-} = 0$$

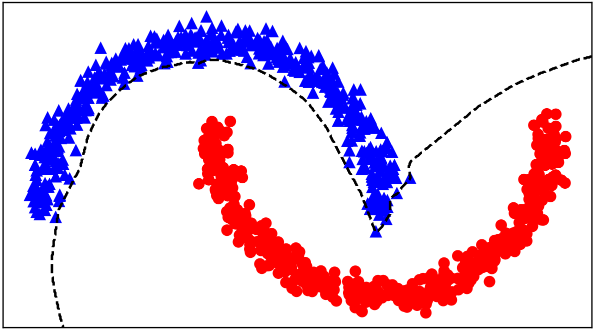



3. Return $F(\boldsymbol{x}) = \mathrm{sign}(-\langle \mathbf{1}, \hat{\Phi}(\Phi(\boldsymbol{x}))\rangle)$.

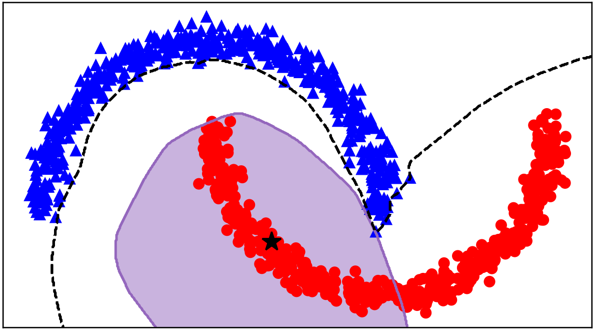
# Algorithm



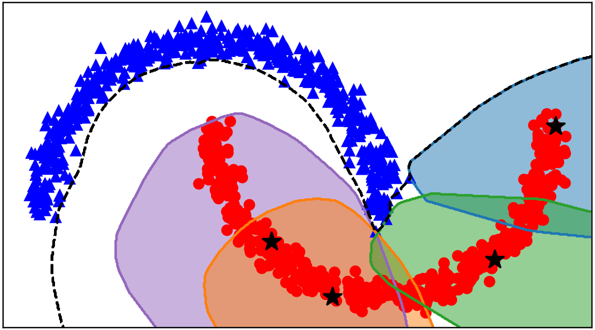
# Algorithm



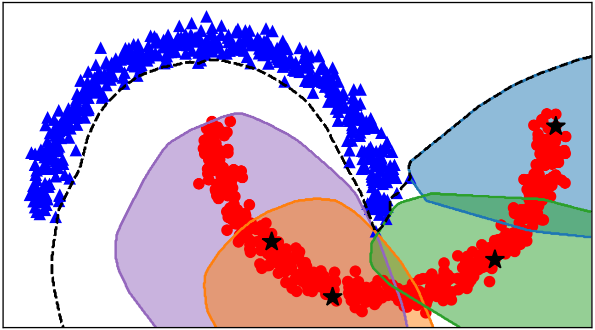
# Algorithm



# Algorithm



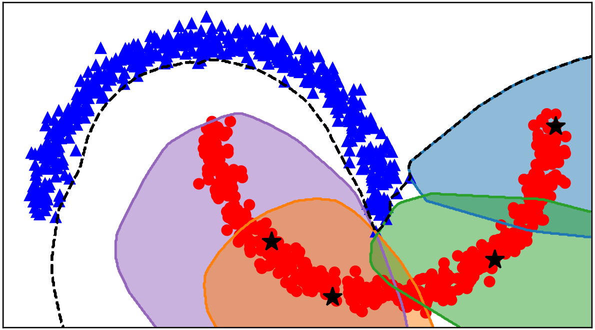
# Algorithm



## Pruning:

- prune neurons from $\hat{\Phi}$ by solving a *set cover problem*

# Algorithm



## Pruning:

- prune neurons from $\hat{\Phi}$ by solving a *set cover problem*
- NP-hard but poly-time approximation algorithms exist

# Main result

Assume $\sigma(t) = \mathrm{Thres}(t) = \mathbb{1}[t \geq 0]$ ($\mathrm{ReLU}$ possible)

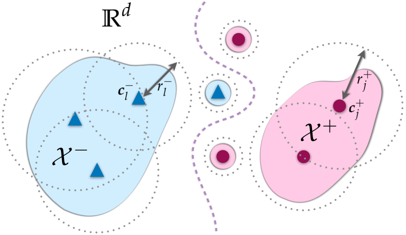
## Theorem:

Let $\mathcal{X}^-, \mathcal{X}^+ \subset \mathbb{B}_2^d$ be finite and $\delta$-separated. Assume, that

$$n \gtrsim \delta^{-1} \log(M^- M^+ / \eta),$$

$$n \gtrsim \max_{\ell \in [M^-]} (r_\ell^-)^{-3} w^2(\mathcal{X}_\ell^- - \boldsymbol{c}_\ell^-),$$

$$n \gtrsim \max_{j \in [M^+]} (r_j^+)^{-3} w^2(\mathcal{X}_j^+ - \boldsymbol{c}_j^+).$$




$$r_\ell^- \lesssim \mathrm{d}(\boldsymbol{c}_\ell^-, \mathcal{C}^+), \ \ell \in [M^-]$$

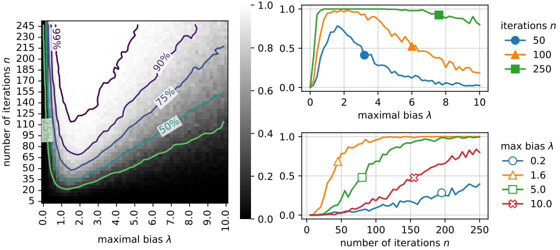
$$r_j^+ \lesssim \mathrm{d}(\boldsymbol{c}_j^+, \mathcal{C}^-), \ j \in [M^+]$$

Then, $F$ memorizes $\mathcal{X}^-$ and $\mathcal{X}^+$ with probability $\geq 1 - \eta$.
Moreover, $\hat{\Phi}$ has at most $M^-$ neurons.

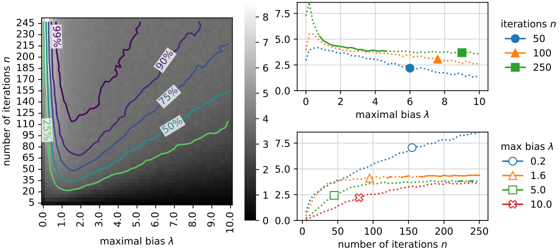
Any Questions?

# Numerical Results – Two Moons

## Interpolation probability:




## Width of $\hat{\Phi}$:



# Algorithm

Assume $\sigma(t) = 0$ $(t < 0)$ and $\sigma(t) > 0$ $(t > 0)$.

## Algorithm

1. Randomly sample $\Phi\colon \mathbb{R}^d \to \mathbb{R}^n, \boldsymbol{x} \mapsto \sigma(\boldsymbol{W}\boldsymbol{x} + \boldsymbol{b})$

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

2. Construct $\hat{\Phi}\colon \mathbb{R}^n \to \mathbb{R}^{|\mathcal{X}^-|}, \boldsymbol{z} \mapsto \sigma(-\boldsymbol{U}\boldsymbol{z} + \boldsymbol{m}/2)$

$$\boldsymbol{U}_{\boldsymbol{x}^-} = \mathbb{1}[\Phi(\boldsymbol{x}^-) = \boldsymbol{0}] \quad \text{and} \quad m_{\boldsymbol{x}^-} = \min_{\boldsymbol{x}^+} \langle \boldsymbol{U}_{\boldsymbol{x}^-}, \Phi(\boldsymbol{x}^+) \rangle$$

3. Return $F(\boldsymbol{x}) = \operatorname{sign}(-\langle \mathbf{1}, \hat{\Phi}(\Phi(\boldsymbol{x})) \rangle)$.