Learning with norm-based neural networks: model capacity, function spaces, and computational-statistical gaps

Fanghui Liu

fanghui.liu@warwick.ac.uk

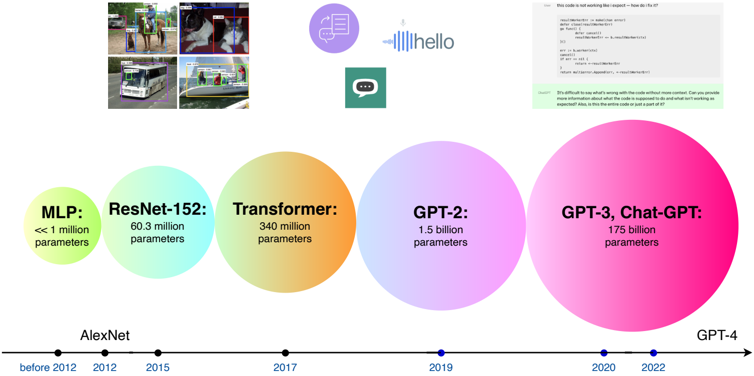
Department of Computer Science, University of Warwick, UK

Centre for Discrete Mathematics and its Applications (DIMAP), Warwick

[joint work with Leello Dadi, Zhenyu Zhu, Volkan Cevher (EPFL)]

at Shanghai Jiao Tong University 2024

Over-parameterization: more parameters than training data



Scaling law: under compute budget

scaling law [14]

test loss = A $\times$ Model Size^{-a} + B $\times$ Data Size^{-b} + C

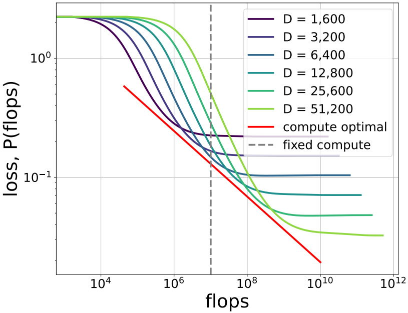
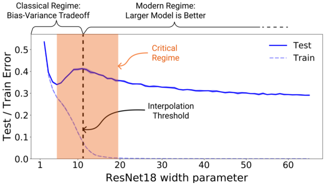


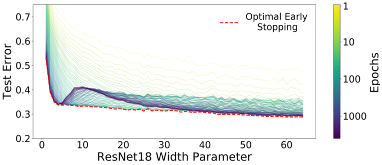
Figure 1: Scaling law under compute-optimal configuration [21].

Model size is a “right” complexity?

- double descent [6] (Belkin, Hsu, Ma, Mandal, 2019)



(a) Results on ResNet18 [18]

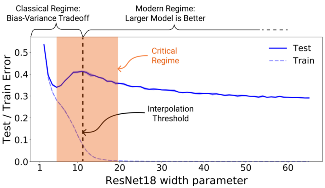


(b) Optimal early stopping [18].

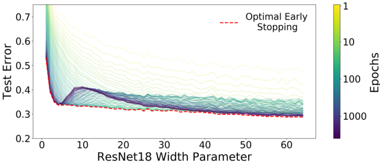
- Empirically: neural network pruning [16], lottery ticket hypothesis [12], fine-tuning with large dropout [27]
- Theoretically: how much over-parameterization is sufficient? [8, 25]

Model size is a "right" complexity?

- double descent [6] (Belkin, Hsu, Ma, Mandal, 2019)



(a) Results on ResNet18 [18]



(b) Optimal early stopping [18].

- Empirically: neural network pruning [16], lottery ticket hypothesis [12], fine-tuning with large dropout [27]
- Theoretically: how much over-parameterization is sufficient? [8, 25]

What is the “right” model complexity?

◦ Complexity of a prediction rule, e.g.,

- number of parameters
- norm of parameters

[3] (Bartlett, 1998)

The size of the weights is more important than the size of the network!

Norm-based capacity:[19, 23, 20, 9]

name	definition	rank correlation
Parameter Frobenius norm	$\sum_{i=1}^{L} \|W_i\|_F^2$	0.073
Frobenius distance to initialization [17]	$\sum_{i=1}^{L} \|W_i - W_i^0\|_F^2$	−0.263
Spectral complexity [4]	$\prod_{i=1}^{L} \|W_i\| \left(\sum_{i=1}^{L} \frac{\|W_i\|_{2,1}^{3/2}}{\|W_i\|^{3/2}}\right)^{2/3}$	−0.537
Fisher-Rao [15]	$\frac{(L+1)^2}{n} \sum_{i=1}^{n} \langle W, \nabla_W \ell(h_W(x_i), y_i) \rangle$	0.078
Path-norm [19]	$\sum_{(i_0,\dots,i_L)} \prod_{j=1}^{L} \left(W_{i_j, i_{j-1}}\right)^2$	0.373

Table 1: Complexity measures compared in the empirical study [13], and their correlation with generalization

What is the “right” model complexity?

∘ Complexity of a prediction rule, e.g.,

- number of parameters
- norm of parameters

[3] (Bartlett, 1998)

The size of the weights is more important than the size of the network!

Norm-based capacity:[19, 23, 20, 9]

name	definition	rank correlation
Parameter Frobenius norm	$\sum_{i=1}^{L} \|W_i\|_F^2$	0.073
Frobenius distance to initialization [17]	$\sum_{i=1}^{L} \|W_i - W_i^0\|_F^2$	−0.263
Spectral complexity [4]	$\prod_{i=1}^{L} \|W_i\| \left(\sum_{i=1}^{L} \frac{\|W_i\|_{2,1}^{3/2}}{\|W_i\|^{3/2}} \right)^{2/3}$	−0.537
Fisher-Rao [15]	$\frac{(L+1)^2}{n} \sum_{i=1}^{n} \langle W, \nabla_W \ell(h_W(x_i), y_i) \rangle$	0.078
Path-norm [19]	$\sum_{(i_0,\dots,i_L)} \prod_{j=1}^{L} \left(W_{i_j, i_{j-1}} \right)^2$	0.373

Table 1: Complexity measures compared in the empirical study [13], and their correlation with generalization

What is the “right” model complexity?

◦ Complexity of a prediction rule, e.g.,

- number of parameters
- norm of parameters

[3] (Bartlett, 1998)

The size of the weights is more important than the size of the network!

Norm-based capacity:[19, 23, 20, 9]

name	definition	rank correlation
Parameter Frobenius norm	$\sum_{i=1}^{L} \|W_i\|_F^2$	0.073
Frobenius distance to initialization [17]	$\sum_{i=1}^{L} \|W_i - W_i^0\|_F^2$	−0.263
Spectral complexity [4]	$\prod_{i=1}^{L} \|W_i\| \left(\sum_{i=1}^{L} \frac{\|W_i\|_{2,1}^{3/2}}{\|W_i\|^{3/2}}\right)^{2/3}$	−0.537
Fisher-Rao [15]	$\frac{(L+1)^2}{n} \sum_{i=1}^{n} \langle W, \nabla_W \ell(h_W(x_i), y_i) \rangle$	0.078
Path-norm [19]	$\sum_{(i_0,\dots,i_L)} \prod_{j=1}^{L} \left(W_{i_j, i_{j-1}}\right)^2$	0.373

Table 1: Complexity measures compared in the empirical study [13], and their correlation with generalization

What is the "right" model complexity?

◦ Complexity of a prediction rule, e.g.,

- number of parameters
- norm of parameters

[3] (Bartlett, 1998)

The size of the weights is more important than the size of the network!

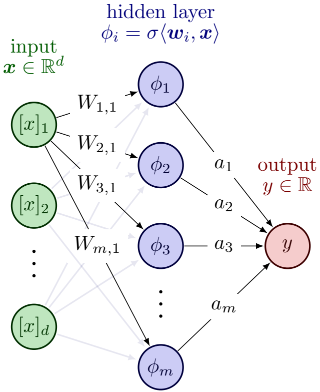
Norm-based capacity:[19, 23, 20, 9]

name	definition	rank correlation
Parameter Frobenius norm	$\sum_{i=1}^{L} \|\boldsymbol{W}_i\|_F^2$	0.073
Frobenius distance to initialization [17]	$\sum_{i=1}^{L} \|\boldsymbol{W}_i - \boldsymbol{W}_i^0\|_\mathrm{F}^2$	−0.263
Spectral complexity [4]	$\prod_{i=1}^{L} \|\boldsymbol{W}_i\| \left(\sum_{i=1}^{L} \frac{\|\boldsymbol{W}_i\|_{2,1}^{3/2}}{\|\boldsymbol{W}_i\|^{3/2}} \right)^{2/3}$	−0.537
Fisher-Rao [15]	$\frac{(L+1)^2}{n} \sum_{i=1}^{n} \langle \boldsymbol{W}, \nabla_{\boldsymbol{W}} \ell(h_{\boldsymbol{W}}(\boldsymbol{x}_i), y_i) \rangle$	0.078
Path-norm [19]	$\sum_{(i_0,\dots,i_L)} \prod_{j=1}^{L} \left(\boldsymbol{W}_{i_j, i_{j-1}} \right)^2$	0.373

Table 1: Complexity measures compared in the empirical study [13], and their correlation with generalization

Two-layer neural networks, path norm

$$\mathcal{P}_m = \left\{ f_{\boldsymbol{\theta}}(\cdot) := \frac{1}{m} \sum_{k=1}^{m} a_k \phi\big(\langle \boldsymbol{w}_k, \cdot \rangle\big) \right\}$$



$$f_a(\boldsymbol{x}) = \int_{\mathcal{W}} a(\boldsymbol{w}) \phi(\boldsymbol{x}, \boldsymbol{w}) \mathrm{d}\mu(\boldsymbol{w})$$

ℓ_1-path norm

$$\|\boldsymbol{\theta}\|_{\mathcal{P}} := \frac{1}{m} \sum_{k=1}^{m} |a_k| \|\boldsymbol{w}_k\|_1$$

- equivalent to Barron spaces [2, 11]

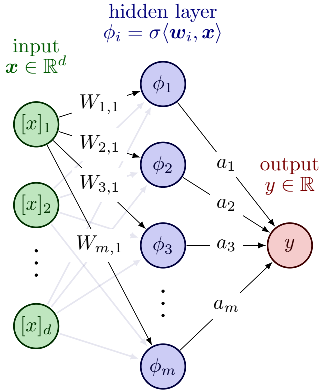
$$\mathcal{B} := \cup_{\mu \in \mathcal{P}(\mathcal{W})} \{ f_a : \|\boldsymbol{a}\|_{L^2(\mu)} < \infty \}$$

$$\|f_a\|_{\mathcal{B}} := \inf_{\mu \in \mathcal{P}(\mathcal{W})} \|\boldsymbol{a}\|_{L^2(\mu)}$$

- largest function space for two-layer neural networks
- No CoD for approximation

Two-layer neural networks, path norm

$$\mathcal{P}_m = \left\{ f_{\boldsymbol{\theta}}(\cdot) := \frac{1}{m} \sum_{k=1}^{m} a_k \phi\big(\langle \boldsymbol{w}_k, \cdot \rangle\big) \right\}$$



$$f_a(\boldsymbol{x}) = \int_{\mathcal{W}} a(\boldsymbol{w}) \phi(\boldsymbol{x}, \boldsymbol{w}) \mathrm{d}\mu(\boldsymbol{w})$$

ℓ_1-path norm

$$\|\boldsymbol{\theta}\|_{\mathcal{P}} := \frac{1}{m} \sum_{k=1}^{m} |a_k| \|\boldsymbol{w}_k\|_1$$

- equivalent to Barron spaces [2, 11]

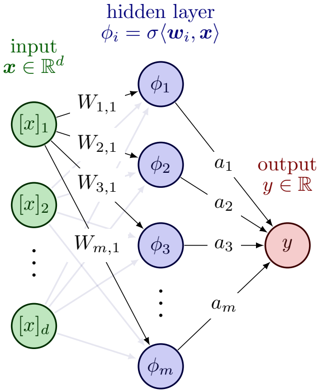
$$\mathcal{B} := \cup_{\mu \in \mathcal{P}(\mathcal{W})} \{ f_a : \|\boldsymbol{a}\|_{L^2(\mu)} < \infty \}$$

$$\|f_a\|_{\mathcal{B}} := \inf_{\mu \in \mathcal{P}(\mathcal{W})} \|\boldsymbol{a}\|_{L^2(\mu)}$$

- largest function space for two-layer neural networks
- No CoD for approximation

Two-layer neural networks, path norm

$$\mathcal{P}_m = \left\{ f_{\boldsymbol{\theta}}(\cdot) := \frac{1}{m}\sum_{k=1}^m a_k \phi\big(\langle \boldsymbol{w}_k, \cdot \rangle\big) \right\}$$



$$f_a(\boldsymbol{x}) = \int_{\mathcal{W}} a(\boldsymbol{w})\phi(\boldsymbol{x}, \boldsymbol{w})\mathrm{d}\mu(\boldsymbol{w})$$

ℓ_1-path norm

$$\|\boldsymbol{\theta}\|_{\mathcal{P}} := \frac{1}{m}\sum_{k=1}^m |a_k|\|\boldsymbol{w}_k\|_1$$

- equivalent to Barron spaces [2, 11]

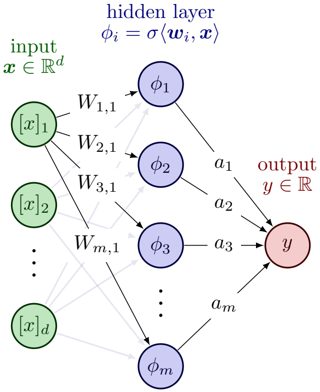
$$\mathcal{B} := \cup_{\mu \in \mathcal{P}(\mathcal{W})}\{f_a : \|\boldsymbol{a}\|_{L^2(\mu)} < \infty\}$$

$$\|f_a\|_{\mathcal{B}} := \inf_{\mu \in \mathcal{P}(\mathcal{W})} \|\boldsymbol{a}\|_{L^2(\mu)}$$

- largest function space for two-layer neural networks
- No CoD for approximation

Two-layer neural networks, path norm

$$\mathcal{P}_m = \left\{ f_{\boldsymbol{\theta}}(\cdot) := \frac{1}{m} \sum_{k=1}^m a_k \phi\big(\langle \boldsymbol{w}_k, \cdot \rangle\big) \right\}$$



$$f_a(\boldsymbol{x}) = \int_{\mathcal{W}} a(\boldsymbol{w}) \phi(\boldsymbol{x}, \boldsymbol{w}) \mathrm{d}\mu(\boldsymbol{w})$$

ℓ_1-path norm

$$\|\boldsymbol{\theta}\|_{\mathcal{P}} := \frac{1}{m} \sum_{k=1}^m |a_k| \|\boldsymbol{w}_k\|_1$$

- equivalent to Barron spaces [2, 11]

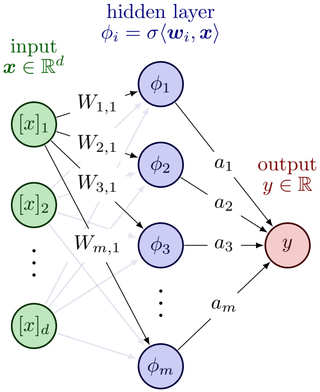
$$\mathcal{B} := \cup_{\mu \in \mathcal{P}(\mathcal{W})} \{ f_a : \|\boldsymbol{a}\|_{L^2(\mu)} < \infty \}$$

$$\|f_a\|_{\mathcal{B}} := \inf_{\mu \in \mathcal{P}(\mathcal{W})} \|\boldsymbol{a}\|_{L^2(\mu)}$$

- largest function space for two-layer neural networks
- No CoD for approximation

Two-layer neural networks, path norm

$$\mathcal{P}_m = \left\{ f_{\boldsymbol{\theta}}(\cdot) := \frac{1}{m} \sum_{k=1}^m a_k \phi\big(\langle \boldsymbol{w}_k, \cdot \rangle\big) \right\}$$



$$f_a(\boldsymbol{x}) = \int_{\mathcal{W}} a(\boldsymbol{w}) \phi(\boldsymbol{x}, \boldsymbol{w}) \mathrm{d}\mu(\boldsymbol{w})$$

ℓ_1-path norm

$$\|\boldsymbol{\theta}\|_{\mathcal{P}} := \frac{1}{m} \sum_{k=1}^m |a_k| \|\boldsymbol{w}_k\|_1$$

- equivalent to Barron spaces [2, 11]

$$\mathcal{B} := \cup_{\mu \in \mathcal{P}(\mathcal{W})} \{ f_a : \|\boldsymbol{a}\|_{L^2(\mu)} < \infty \}$$

$$\|f_a\|_{\mathcal{B}} := \inf_{\mu \in \mathcal{P}(\mathcal{W})} \|\boldsymbol{a}\|_{L^2(\mu)}$$

- largest function space for two-layer neural networks
- No CoD for approximation

Our results: statistical guarantees

For the class of two-layer neural networks $\mathcal{G}_R = \{f_{\boldsymbol{\theta}} \in \mathcal{P}_m : \|\boldsymbol{\theta}\|_{\mathcal{P}} \leqslant R\}$

$$\widehat{f_{\boldsymbol{\theta}}} := \operatorname*{argmin}_{f_{\boldsymbol{\theta}} \in \mathcal{G}_R} \frac{1}{n} \sum_{i=1}^{n} (y_i - f_{\boldsymbol{\theta}}(\boldsymbol{x}_i))^2 \,.$$

Theorem (Liu, Dadi, Cevher, JMLR 2024)

Under standard assumptions (bounded data, $f^\star \in \mathcal{B}$), for two-layer over-parameterized neural networks, we have

$$\left\|\widehat{f_{\theta}} - f^\star\right\|^2_{L^2_{\rho_X}} \lesssim \frac{R^2}{m} + R^2 d^{\frac{1}{3}} n^{-\frac{d+2}{2d+2}} \qquad w.h.p.$$

$n^{-\frac{d+2}{2d+2}}$ is always faster than $n^{-\frac{1}{2}}$: No curse of dimensionality!

Our results: statistical guarantees

For the class of two-layer neural networks $\mathcal{G}_R = \{f_{\boldsymbol{\theta}} \in \mathcal{P}_m : \|\boldsymbol{\theta}\|_{\mathcal{P}} \leqslant R\}$

$$\widehat{f_{\boldsymbol{\theta}}} := \operatorname*{argmin}_{f_{\boldsymbol{\theta}} \in \mathcal{G}_R} \frac{1}{n} \sum_{i=1}^{n} (y_i - f_{\boldsymbol{\theta}}(\boldsymbol{x}_i))^2 \,.$$

Theorem (Liu, Dadi, Cevher, JMLR 2024)

Under standard assumptions (bounded data, $f^\star \in \mathcal{B}$), for two-layer over-parameterized neural networks, we have

$$\left\|\widehat{f_{\boldsymbol{\theta}}} - f^\star\right\|^2_{L^2_{\rho_X}} \lesssim \frac{R^2}{m} + R^2 d^{\frac{1}{3}} n^{-\frac{d+2}{2d+2}} \qquad w.h.p.$$

$n^{-\frac{d+2}{2d+2}}$ is always faster than $n^{-\frac{1}{2}}$: No curse of dimensionality!

Our results: statistical guarantees

For the class of two-layer neural networks $\mathcal{G}_R = \{f_{\boldsymbol{\theta}} \in \mathcal{P}_m : \|\boldsymbol{\theta}\|_{\mathcal{P}} \leqslant R\}$

$$\widehat{f_{\boldsymbol{\theta}}} := \operatorname*{argmin}_{f_{\boldsymbol{\theta}} \in \mathcal{G}_R} \frac{1}{n} \sum_{i=1}^{n} (y_i - f_{\boldsymbol{\theta}}(\boldsymbol{x}_i))^2 \,.$$

Theorem (Liu, Dadi, Cevher, JMLR 2024)

Under standard assumptions (bounded data, $f^\star \in \mathcal{B}$), for two-layer over-parameterized neural networks, we have

$$\left\|\widehat{f_{\boldsymbol{\theta}}} - f^\star\right\|^2_{L^2_{\rho_X}} \lesssim \frac{R^2}{m} + R^2 d^{\frac{1}{3}} n^{-\frac{d+2}{2d+2}} \qquad w.h.p.$$

$n^{-\frac{d+2}{2d+2}}$ is always faster than $n^{-\frac{1}{2}}$: No curse of dimensionality!

Sample complexity

Proposition (metric entropy)

For bounded data $\|\boldsymbol{x}\|_\infty \leq 1$*, denote* $\mathcal{G}_R = \{f_{\boldsymbol{\theta}} \in \mathcal{P}_m : \|\boldsymbol{\theta}\|_{\mathcal{P}} \leqslant R\}$*, the metric entropy of* $\mathcal{G}_1$ *can be bounded by*

$$\log \mathcal{N}_2(\mathcal{G}_1, \epsilon) \leqslant Cd\epsilon^{-\frac{2d}{d+2}}\,, \quad \forall \epsilon > 0 \quad \textit{and} \quad d \geq 5\,,$$

with some universal constant C *independent of* d*.*

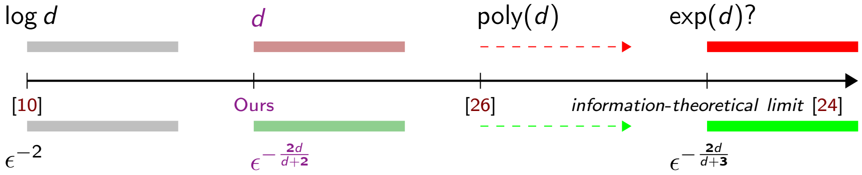
Sample complexity

Proposition (metric entropy)

For bounded data $\|\boldsymbol{x}\|_\infty \le 1$, *denote* $\mathcal{G}_R = \{f_{\boldsymbol{\theta}} \in \mathcal{P}_m : \|\boldsymbol{\theta}\|_{\mathcal{P}} \leqslant R\}$, *the metric entropy of* $\mathcal{G}_1$ *can be bounded by*

$$\log \mathcal{N}_2(\mathcal{G}_1, \epsilon) \leqslant C d \epsilon^{-\frac{2d}{d+2}}\,, \quad \forall \epsilon > 0 \quad \text{and} \quad d \ge 5\,,$$

with some universal constant C *independent of* d.



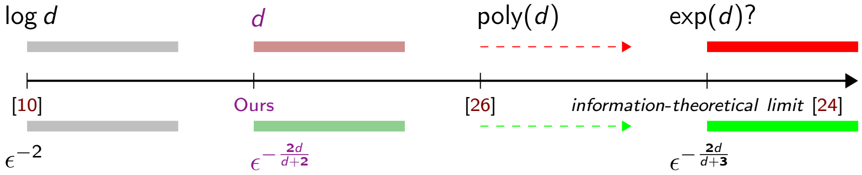
Sample complexity

Proposition (metric entropy)

For bounded data $\|\boldsymbol{x}\|_\infty \leq 1$, denote $\mathcal{G}_R = \{f_{\boldsymbol{\theta}} \in \mathcal{P}_m : \|\boldsymbol{\theta}\|_{\mathcal{P}} \leqslant R\}$, the metric entropy of $\mathcal{G}_1$ can be bounded by

$$\log \mathcal{N}_2(\mathcal{G}_1, \epsilon) \leqslant C d \epsilon^{-\frac{2d}{d+2}}\,, \quad \forall \epsilon > 0 \quad \text{and} \quad d \geq 5\,,$$

with some universal constant C independent of d.



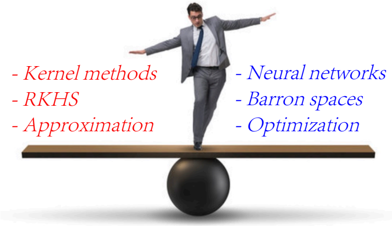
The "best" trade-off between ϵ and d.

Computational-to-statistical gaps

Optimization in Barron spaces is NP hard: curse of dimensionality!

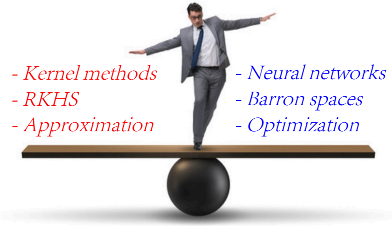
Computational-to-statistical gaps

Optimization in Barron spaces is NP hard: curse of dimensionality!



Computational-to-statistical gaps

Optimization in Barron spaces is NP hard: curse of dimensionality!



Do some Barron functions can be learned by two-layer NNs, both statistically and computationally efficient?

Learning with multiple ReLU neurons under GD training

Can we learn multiple ReLU neurons by two-layer NNs, both statistically and computationally efficient?

$$f^\star(\boldsymbol{x}) = \sum_{l=1}^{k} \sigma(\langle \boldsymbol{v}_l, \boldsymbol{x} \rangle), k = \mathcal{O}(1)$$

$\|\hat{f} - f^\star\|_{L^2(\mathrm{d}\mu)} \leq \epsilon$ from $\{\boldsymbol{x}_i, f^\star(\boldsymbol{x}_i)\}_{i=1}^n$ with $\boldsymbol{x}_i \sim \mathcal{N}(0, \boldsymbol{I}_d)$

Theorem ([7] PAC learning $f^\star$ under Gaussian measure)

There exists an algorithm that requires time/samples at $(d/\epsilon)^{\mathcal{O}(k^2)}$

- correlational statistical query (CSQ): $|\tilde{q} - \mathbb{E}_{\boldsymbol{x},y}[\psi(\boldsymbol{x})y]| \leq \tau$

Learning with multiple ReLU neurons under GD training

Can we learn multiple ReLU neurons by two-layer NNs, both statistically and computationally efficient?

$$f^\star(\boldsymbol{x}) = \sum_{l=1}^{k} \sigma(\langle \boldsymbol{v}_l, \boldsymbol{x} \rangle), k = \mathcal{O}(1)$$

$\|\hat{f} - f^\star\|_{L^2(\mathrm{d}\mu)} \leq \epsilon$ from $\{\boldsymbol{x}_i, f^\star(\boldsymbol{x}_i)\}_{i=1}^n$ with $\boldsymbol{x}_i \sim \mathcal{N}(0, \boldsymbol{I}_d)$

Theorem ([7] PAC learning $f^\star$ under Gaussian measure)

There exists an algorithm that requires time/samples at $(d/\epsilon)^{\mathcal{O}(k^2)}$

- correlational statistical query (CSQ): $|\tilde{q} - \mathbb{E}_{\boldsymbol{x},y}[\psi(\boldsymbol{x})y]| \leq \tau$

Learning with multiple ReLU neurons under GD training

Can we learn multiple ReLU neurons by two-layer NNs, both statistically and computationally efficient?

$$f^\star(\boldsymbol{x}) = \sum_{l=1}^{k} \sigma(\langle \boldsymbol{v}_l, \boldsymbol{x} \rangle), k = \mathcal{O}(1)$$

$\|\hat{f} - f^\star\|_{L^2(\mathrm{d}\mu)} \leq \epsilon$ from $\{\boldsymbol{x}_i, f^\star(\boldsymbol{x}_i)\}_{i=1}^n$ with $\boldsymbol{x}_i \sim \mathcal{N}(0, \boldsymbol{I}_d)$

Theorem ([7] PAC learning $f^\star$ under Gaussian measure)

There exists an algorithm that requires time/samples at $(d/\epsilon)^{\mathcal{O}(k^2)}$

- correlational statistical query (CSQ): $|\tilde{q} - \mathbb{E}_{\boldsymbol{x},y}[\psi(\boldsymbol{x})y]| \leq \tau$

Learning with multiple ReLU neurons under GD training

Can we learn multiple ReLU neurons by two-layer NNs, both statistically and computationally efficient?

$$f^\star(\boldsymbol{x}) = \sum_{l=1}^{k} \sigma(\langle \boldsymbol{v}_l, \boldsymbol{x} \rangle), k = \mathcal{O}(1)$$

$\|\hat{f} - f^\star\|_{L^2(\mathrm{d}\mu)} \leq \epsilon$ from $\{\boldsymbol{x}_i, f^\star(\boldsymbol{x}_i)\}_{i=1}^n$ with $\boldsymbol{x}_i \sim \mathcal{N}(0, \boldsymbol{I}_d)$

Theorem ([7] PAC learning $f^\star$ under Gaussian measure)

There exists an algorithm that requires time/samples at $(d/\epsilon)^{\mathcal{O}(k^2)}$

- correlational statistical query (CSQ): $|\tilde{q} - \mathbb{E}_{\boldsymbol{x},y}[\psi(\boldsymbol{x})y]| \leq \tau$

Learning with multiple ReLU neurons under GD training

Can we learn multiple ReLU neurons by two-layer NNs, both statistically and computationally efficient?

$$f^\star(\boldsymbol{x}) = \sum_{l=1}^{k} \sigma(\langle \boldsymbol{v}_l, \boldsymbol{x} \rangle), k = \mathcal{O}(1)$$

$\|\hat{f} - f^\star\|_{L^2(\mathrm{d}\mu)} \leq \epsilon$ from $\{\boldsymbol{x}_i, f^\star(\boldsymbol{x}_i)\}_{i=1}^n$ with $\boldsymbol{x}_i \sim \mathcal{N}(0, \boldsymbol{I}_d)$

Theorem ([7] PAC learning $f^\star$ under Gaussian measure)

There exists an algorithm that requires time/samples at $(d/\epsilon)^{\mathcal{O}(k^2)}$

- correlational statistical query (CSQ): $|\tilde{q} - \mathbb{E}_{\boldsymbol{x},y}[\psi(\boldsymbol{x})y]| \leq \tau$

How does student(s) become teacher(s) under GD training?

Learning multi ReLU neurons by two-layer NN via online SGD

$$L(\boldsymbol{W}) = \frac{1}{2}\mathbb{E}_{\boldsymbol{x}\sim\mathcal{N}(0,\boldsymbol{I}_d)}\left(\sum_{i=1}^{m}\sigma(\langle \boldsymbol{w}_i, \boldsymbol{x}\rangle) - \sum_{l=1}^{k}\sigma(\langle \boldsymbol{v}_l, \boldsymbol{x}\rangle)\right)^2$$

- Gaussian initialization $\boldsymbol{w}_i \sim \mathcal{N}(0, \sigma^2 \boldsymbol{I}_d)$
- angle: $\theta_{il} \triangleq \angle(\boldsymbol{w}_i, \boldsymbol{v}_l)$

Assumption

- *diverse teacher neurons:* $\{\boldsymbol{v}_l\}_{l=1}^{d}$ *are (nearly) orthogonal and* $\|\boldsymbol{v}_l\|_2 = \texttt{const}$
- *warm start: the smallest angle not close to orthogonal*
  - *hold w.p.* $\exp(-\mathcal{O}(1))$ *for fixed dimension*

How does student(s) become teacher(s) under GD training?

Learning multi ReLU neurons by two-layer NN via online SGD

$$L(\boldsymbol{W}) = \frac{1}{2}\mathbb{E}_{\boldsymbol{x}\sim\mathcal{N}(0,\boldsymbol{I}_d)}\left(\sum_{i=1}^{m}\sigma(\langle \boldsymbol{w}_i, \boldsymbol{x}\rangle) - \sum_{l=1}^{k}\sigma(\langle \boldsymbol{v}_l, \boldsymbol{x}\rangle)\right)^2$$

- Gaussian initialization $\boldsymbol{w}_i \sim \mathcal{N}(0, \sigma^2 \boldsymbol{I}_d)$
- angle: $\theta_{il} \triangleq \angle(\boldsymbol{w}_i, \boldsymbol{v}_l)$

Assumption

- *diverse teacher neurons:* $\{\boldsymbol{v}_l\}_{l=1}^{d}$ *are (nearly) orthogonal and* $\|\boldsymbol{v}_l\|_2 = \texttt{const}$
- *warm start: the smallest angle not close to orthogonal*
    - *hold w.p.* $\exp(-\mathcal{O}(1))$ *for fixed dimension*

How does student(s) become teacher(s) under GD training?

Learning multi ReLU neurons by two-layer NN via online SGD

$$L(\boldsymbol{W}) = \frac{1}{2}\mathbb{E}_{\boldsymbol{x}\sim\mathcal{N}(0,\boldsymbol{I}_d)}\left(\sum_{i=1}^{m}\sigma(\langle \boldsymbol{w}_i, \boldsymbol{x}\rangle) - \sum_{l=1}^{k}\sigma(\langle \boldsymbol{v}_l, \boldsymbol{x}\rangle)\right)^2$$

- Gaussian initialization $\boldsymbol{w}_i \sim \mathcal{N}(0, \sigma^2 \boldsymbol{I}_d)$
- angle: $\theta_{il} \triangleq \angle(\boldsymbol{w}_i, \boldsymbol{v}_l)$

Assumption

- *diverse teacher neurons:* $\{\boldsymbol{v}_l\}_{l=1}^d$ *are (nearly) orthogonal and* $\|\boldsymbol{v}_l\|_2 = \texttt{const}$
- *warm start: the smallest angle not close to orthogonal*
  - *hold w.p.* $\exp(-\mathcal{O}(1))$ *for fixed dimension*

How does student(s) become teacher(s) under GD training?

Learning multi ReLU neurons by two-layer NN via online SGD

$$L(\boldsymbol{W}) = \frac{1}{2}\mathbb{E}_{\boldsymbol{x}\sim\mathcal{N}(0,\boldsymbol{I}_d)}\left(\sum_{i=1}^{m}\sigma(\langle \boldsymbol{w}_i, \boldsymbol{x}\rangle) - \sum_{l=1}^{k}\sigma(\langle \boldsymbol{v}_l, \boldsymbol{x}\rangle)\right)^2$$

- Gaussian initialization $\boldsymbol{w}_i \sim \mathcal{N}(0, \sigma^2 \boldsymbol{I}_d)$
- angle: $\theta_{il} \triangleq \angle(\boldsymbol{w}_i, \boldsymbol{v}_l)$

Assumption

- *diverse teacher neurons:* $\{\boldsymbol{v}_l\}_{l=1}^{d}$ *are (nearly) orthogonal and* $\|\boldsymbol{v}_l\|_2 = \texttt{const}$
- *warm start: the smallest angle not close to orthogonal*
  - *hold w.p.* $\exp(-\mathcal{O}(1))$ *for fixed dimension*

How does student(s) become teacher(s) under GD training?

Theorem (Zhu, Liu, Cevher, 2024)

For sufficiently small initialization and step-size $\sigma, \eta = o(m^{-k^2})$, *then there exists a time* $T_2 = \frac{1}{\eta}$ *such that* $\forall T \in \mathbb{N}$ *and* $i \in [m]$,

$$L(\boldsymbol{W}(T + T_2)) \leq \mathcal{O}\left(\frac{1}{T^3}\right), \|\boldsymbol{w}_i(T + T_2)\|_2 = \Theta\left(\frac{k\|\boldsymbol{v}\|_2}{m}\right) \; w.h.p.$$

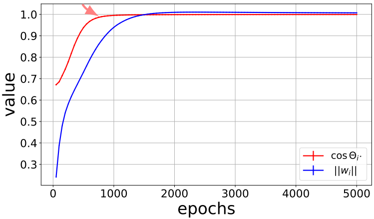
How does student(s) become teacher(s) under GD training?

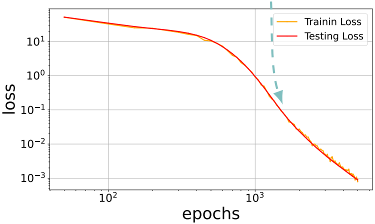
Theorem (Zhu, Liu, Cevher, 2024)

For sufficiently small initialization and step-size $\sigma, \eta = o(m^{-k^2})$, then there exists a time $T_2 = \frac{1}{\eta}$ such that $\forall T \in \mathbb{N}$ and $i \in [m]$,

$$L(\boldsymbol{W}(T + T_2)) \leq \mathcal{O}\left(\frac{1}{T^3}\right), \|\boldsymbol{w}_i(T + T_2)\|_2 = \Theta\left(\frac{k\|\boldsymbol{v}\|_2}{m}\right) \; w.h.p.$$

- align $\theta_{i^*} \to 0$ norm converge then fit





Take-away messages

- model size -> size of weights -> path norm -> Barron spaces
- statistical guarantees with improved sample complexity
- computational-statistical gap -> learning with multiple ReLU neurons

We're organizing one workshop at NeurIPS 2024!

Fine-Tuning in Modern Machine Learning: Principles and Scalability

https://sites.google.com/view/neurips2024-ftw/home

Thanks for your attention!

Q & A

my homepage www.lfhsgre.org for more information!

Take-away messages

- model size -> size of weights -> path norm -> Barron spaces
- statistical guarantees with improved sample complexity
- computational-statistical gap -> learning with multiple ReLU neurons

We're organizing one workshop at NeurIPS 2024!

Fine-Tuning in Modern Machine Learning: Principles and Scalability

`https://sites.google.com/view/neurips2024-ftw/home`

Thanks for your attention!

Q & A

my homepage `www.lfhsgre.org` for more information!

Take-away messages

- model size -> size of weights -> path norm -> Barron spaces
- statistical guarantees with improved sample complexity
- computational-statistical gap -> learning with multiple ReLU neurons

We're organizing one workshop at NeurIPS 2024!

Fine-Tuning in Modern Machine Learning: Principles and Scalability

https://sites.google.com/view/neurips2024-ftw/home

Thanks for your attention!

Q & A

my homepage www.lfhsgre.org for more information!

Take-away messages

- model size -> size of weights -> path norm -> Barron spaces
- statistical guarantees with improved sample complexity
- computational-statistical gap -> learning with multiple ReLU neurons

We're organizing one workshop at NeurIPS 2024!

Fine-Tuning in Modern Machine Learning: Principles and Scalability

https://sites.google.com/view/neurips2024-ftw/home

Thanks for your attention!

Q & A

my homepage www.lfhsgre.org for more information!

Take-away messages

- model size -> size of weights -> path norm -> Barron spaces
- statistical guarantees with improved sample complexity
- computational-statistical gap -> learning with multiple ReLU neurons

We're organizing one workshop at NeurIPS 2024!

Fine-Tuning in Modern Machine Learning: Principles and Scalability

https://sites.google.com/view/neurips2024-ftw/home

Thanks for your attention!

Q & A

my homepage www.lfhsgre.org for more information!

Francis Bach.
Breaking the curse of dimensionality with convex neural networks.
Journal of Machine Learning Research, 18(1):629–681, 2017.

Andrew R Barron.
Universal approximation bounds for superpositions of a sigmoidal function.
IEEE Transactions on Information theory, 39(3):930–945, 1993.

Peter Bartlett.
The sample complexity of pattern classification with neural networks: the size of the weights is more important than the size of the network.
IEEE Transactions on Information Theory, 44(2):525–536, 1998.

Peter Bartlett, Dylan Foster, and Matus Telgarsky.
Spectrally-normalized margin bounds for neural networks.
In *Advances in Neural Information Processing Systems*, pages 6241–6250, 2017.

Francesca Bartolucci, Ernesto De Vito, Lorenzo Rosasco, and Stefano Vigogna.
Understanding neural networks with reproducing kernel Banach spaces.
Applied and Computational Harmonic Analysis, 2022.

Mikhail Belkin, Daniel Hsu, Siyuan Ma, and Soumik Mandal.
Reconciling modern machine-learning practice and the classical bias–variance trade-off.
the National Academy of Sciences, 116(32):15849–15854, 2019.

Sitan Chen and Shyam Narayanan.
A faster and simpler algorithm for learning shallow networks.
arXiv preprint arXiv:2307.12496, 2023.

Zixiang Chen, Yuan Cao, Difan Zou, and Quanquan Gu.
How much over-parameterization is sufficient to learn deep relu networks?
In *International Conference on Learning Representations*, 2020.

Carles Domingo-Enrich and Youssef Mroueh.
Tighter sparse approximation bounds for relu neural networks.
In *International Conference on Learning Representations*, 2022.

Weinan E, Chao Ma, and Lei Wu.
A priori estimates of the population risk for two-layer neural networks.
Communications in Mathematical Sciences, 17(5):1407–1425, 2019.

Weinan E, Chao Ma, and Lei Wu.
The barron space and the flow-induced function spaces for neural network models.
Constructive Approximation, pages 1–38, 2021.

Jonathan Frankle and Michael Carbin.
The lottery ticket hypothesis: Finding sparse, trainable neural networks.
In *International Conference on Learning Representations*, 2019.

Yiding Jiang, Behnam Neyshabur, Hossein Mobahi, Dilip Krishnan, and Samy Bengio.
Fantastic generalization measures and where to find them.
In *International Conference on Learning Representations*, 2020.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei.
Scaling laws for neural language models.
arXiv preprint arXiv:2001.08361, 2020.

Tengyuan Liang, Tomaso Poggio, Alexander Rakhlin, and James Stokes.
Fisher-rao metric, geometry, and complexity of neural networks.
In *International conference on Artificial Intelligence and Statistics*, pages 888–896, 2019.

Pavlo Molchanov, Stephen Tyree, Tero Karras, Timo Aila, and Jan Kautz.
Pruning convolutional neural networks for resource efficient inference.
In *International Conference on Learning Representations*, 2017.

Vaishnavh Nagarajan and J Zico Kolter.
Generalization in deep networks: The role of distance from initialization.
arXiv preprint arXiv:1901.01672, 2019.

Preetum Nakkiran, Gal Kaplun, Yamini Bansal, Tristan Yang, Boaz Barak, and Ilya Sutskever.
Deep double descent: Where bigger models and more data hurt.
In *International Conference on Learning Representations*, 2019.

Behnam Neyshabur, Ryota Tomioka, and Nathan Srebro.
Norm-based capacity control in neural networks.
In *Conference on Learning Theory*, pages 1376–1401. PMLR, 2015.

Greg Ongie, Rebecca Willett, Daniel Soudry, and Nathan Srebro.
A function space view of bounded norm infinite width relu nets: The multivariate case.
In *International Conference on Learning Representations*, 2020.

Elliot Paquette, Courtney Paquette, Lechao Xiao, and Jeffrey Pennington.
4+3 phases of compute-optimal neural scaling laws.
arXiv preprint arXiv:2405.15074, 2024.

Ali Rahimi and Benjamin Recht.
Uniform approximation of functions with random bases.
In *Annual Allerton Conference on Communication, Control, and Computing*, pages 555–561. IEEE, 2008.

Pedro Savarese, Itay Evron, Daniel Soudry, and Nathan Srebro.
How do infinite width bounded norm networks look in function space?
In *Conference on Learning Theory*, pages 2667–2690. PMLR, 2019.

Jonathan W Siegel and Jinchao Xu.
Sharp bounds on the approximation rates, metric entropy, and n**-widths of shallow neural networks.**
arXiv preprint arXiv:2101.12365, 2021.

Taiji Suzuki, Denny Wu, Kazusato Oko, and Atsushi Nitanda.
Feature learning via mean-field langevin dynamics: classifying sparse parities and beyond.
In *Advances in Neural Information Processing Systems*, 2023.

Lei Wu and Jihao Long.
A spectral-based analysis of the separation between two-layer neural networks and linear methods.
Journal of Machine Learning Research, 119:1–34, 2022.

Jianyu Zhang and Léon Bottou.
Fine-tuning with very large dropout.
arXiv preprint arXiv:2403.00946, 2024.

Background: RFMs and kernel methods

Consider a RFM with infinite many features $f_a(\boldsymbol{x}) = \int_{\mathcal{W}} a(\boldsymbol{w})\phi(\boldsymbol{x}, \boldsymbol{w})\mathrm{d}\mu(\boldsymbol{w})$, define

$$\mathcal{F}_{p,\mu} := \{f_a : \|\boldsymbol{a}\|_{L^p(\mu)} < \infty\}, \quad \|f\|_{\mathcal{F}_{p,\mu}} := \inf_{f_a = f} \|\boldsymbol{a}\|_{L^p(\mu)}$$

- RFMs $\equiv$ kernel methods by taking $p = 2$ using Representer theorem [22]
  - function space: reproducing kernel Hilbert space $\mathcal{H}_{k_\mu} = \mathcal{F}_{2,\mu}$

$$\hat{k}_m(\boldsymbol{x}, \boldsymbol{x}') = \frac{1}{m}\sum_{i=1}^{m} \phi(\boldsymbol{x}, \boldsymbol{w}_i)\phi(\boldsymbol{x}', \boldsymbol{w}_i) \rightarrow k_\mu(\boldsymbol{x}, \boldsymbol{x}') = \int_{\mathcal{W}} \phi(\boldsymbol{x}, \boldsymbol{w})\phi(\boldsymbol{x}', \boldsymbol{w})\mathrm{d}\mu(\boldsymbol{w})$$

- RFMs $\not\equiv$ kernel methods if $p < 2$

  function space: $\mathcal{F}_{\infty,\mu} \subseteq \mathcal{F}_{p,\mu} \subseteq \mathcal{F}_{q,\mu} \subseteq \mathcal{F}_{1,\mu}$ if $p \geq q$

Background: RFMs and kernel methods

Consider a RFM with infinite many features $f_a(\boldsymbol{x}) = \int_{\mathcal{W}} a(\boldsymbol{w})\phi(\boldsymbol{x}, \boldsymbol{w})\mathrm{d}\mu(\boldsymbol{w})$, define

$$\mathcal{F}_{p,\mu} := \{f_a : \|\boldsymbol{a}\|_{L^p(\mu)} < \infty\}, \quad \|f\|_{\mathcal{F}_{p,\mu}} := \inf_{f_a = f} \|\boldsymbol{a}\|_{L^p(\mu)}$$

- RFMs $\equiv$ kernel methods by taking $p = 2$ using Representer theorem [22]
  - function space: reproducing kernel Hilbert space $\mathcal{H}_{k_\mu} = \mathcal{F}_{2,\mu}$

$$\hat{k}_m(\boldsymbol{x}, \boldsymbol{x}') = \frac{1}{m}\sum_{i=1}^{m} \phi(\boldsymbol{x}, \boldsymbol{w}_i)\phi(\boldsymbol{x}', \boldsymbol{w}_i) \;\to\; k_\mu(\boldsymbol{x}, \boldsymbol{x}') = \int_{\mathcal{W}} \phi(\boldsymbol{x}, \boldsymbol{w})\phi(\boldsymbol{x}', \boldsymbol{w})\mathrm{d}\mu(\boldsymbol{w})$$

- RFMs $\not\equiv$ kernel methods if $p < 2$

  function space: $\mathcal{F}_{\infty,\mu} \subseteq \mathcal{F}_{p,\mu} \subseteq \mathcal{F}_{q,\mu} \subseteq \mathcal{F}_{1,\mu}$ if $p \geq q$

Background: RFMs and kernel methods

Consider a RFM with infinite many features $f_a(\boldsymbol{x}) = \int_{\mathcal{W}} a(\boldsymbol{w})\phi(\boldsymbol{x}, \boldsymbol{w})\mathrm{d}\mu(\boldsymbol{w})$, define

$$\mathcal{F}_{p,\mu} := \{f_a : \|\boldsymbol{a}\|_{L^p(\mu)} < \infty\}, \quad \|f\|_{\mathcal{F}_{p,\mu}} := \inf_{f_a = f} \|\boldsymbol{a}\|_{L^p(\mu)}$$

- RFMs $\equiv$ kernel methods by taking $p = 2$ using Representer theorem [22]
  - function space: reproducing kernel Hilbert space $\mathcal{H}_{k_\mu} = \mathcal{F}_{2,\mu}$

$$\hat{k}_m(\boldsymbol{x}, \boldsymbol{x}') = \frac{1}{m}\sum_{i=1}^{m} \phi(\boldsymbol{x}, \boldsymbol{w}_i)\phi(\boldsymbol{x}', \boldsymbol{w}_i) \ \rightarrow \ k_\mu(\boldsymbol{x}, \boldsymbol{x}') = \int_{\mathcal{W}} \phi(\boldsymbol{x}, \boldsymbol{w})\phi(\boldsymbol{x}', \boldsymbol{w})\mathrm{d}\mu(\boldsymbol{w})$$

- RFMs $\not\equiv$ kernel methods if $p < 2$

  function space: $\mathcal{F}_{\infty,\mu} \subseteq \mathcal{F}_{p,\mu} \subseteq \mathcal{F}_{q,\mu} \subseteq \mathcal{F}_{1,\mu}$ if $p \geq q$

From RKHS to Barron space

Definition (Barron space [11] (E, Ma, Wu, 2021))

For any $1 \leq p \leq \infty$, we have

$$\mathcal{B} = \cup_{\mu \in \mathcal{P}(\mathcal{W})} \mathcal{F}_{p,\mu}\,, \quad \|f\|_{\mathcal{B}} = \inf_{\mu \in \mathcal{P}(\mathcal{W})} \|f\|_{\mathcal{F}_{p,\mu}}$$

Remark: ◦ Two-layer neural networks: data-adaptive kernel $\mathcal{B} = \cup_{\mu \in \mathcal{P}(\mathcal{W})} \mathcal{H}_{k_\mu}$

◦ equivalent to path norm $\|\Theta\|_{\mathcal{P}} := \frac{1}{m}\sum_{k=1}^{m} |a_k| \|\boldsymbol{w}_k\|_1$

◦ parameter space vs. measure space

e.g., [1] (Bach, 2017), [5] (Bartolucci, Vito, Rosasco, Vigogna, 2022).

Optimization in Barron spaces is difficult: curse of dimensionality!

From RKHS to Barron space

Definition (Barron space [11] (E, Ma, Wu, 2021))

For any $1 \leq p \leq \infty$, we have

$$\mathcal{B} = \cup_{\mu \in \mathcal{P}(\mathcal{W})} \mathcal{F}_{p,\mu}\,, \quad \|f\|_{\mathcal{B}} = \inf_{\mu \in \mathcal{P}(\mathcal{W})} \|f\|_{\mathcal{F}_{p,\mu}}$$

Remark: ◦ Two-layer neural networks: data-adaptive kernel $\mathcal{B} = \cup_{\mu \in \mathcal{P}(\mathcal{W})} \mathcal{H}_{k_\mu}$

◦ equivalent to path norm $\|\boldsymbol{\Theta}\|_{\mathcal{P}} := \frac{1}{m} \sum_{k=1}^m |a_k| \|\boldsymbol{w}_k\|_1$

◦ parameter space vs. measure space

e.g., [1] (Bach, 2017), [5] (Bartolucci, Vito, Rosasco, Vigogna, 2022).

Optimization in Barron spaces is difficult: curse of dimensionality!

From RKHS to Barron space

Definition (Barron space [11] (E, Ma, Wu, 2021))

For any $1 \leq p \leq \infty$, we have

$$\mathcal{B} = \cup_{\mu \in \mathcal{P}(\mathcal{W})} \mathcal{F}_{p,\mu}\,, \quad \|f\|_{\mathcal{B}} = \inf_{\mu \in \mathcal{P}(\mathcal{W})} \|f\|_{\mathcal{F}_{p,\mu}}$$

Remark: ∘ Two-layer neural networks: data-adaptive kernel $\mathcal{B} = \cup_{\mu \in \mathcal{P}(\mathcal{W})} \mathcal{H}_{k_\mu}$

∘ equivalent to path norm $\|\boldsymbol{\Theta}\|_{\mathcal{P}} := \frac{1}{m} \sum_{k=1}^{m} |a_k| \|\boldsymbol{w}_k\|_1$

∘ parameter space vs. measure space

e.g., [1] (Bach, 2017), [5] (Bartolucci, Vito, Rosasco, Vigogna, 2022).

Optimization in Barron spaces is difficult: curse of dimensionality!