

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

---

Fanghui Liu

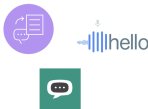
[fanghui.liu@warwick.ac.uk](mailto:fanghui.liu@warwick.ac.uk)

*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 Department of Computer Science, University of Wisconsin-Madison



## In the era of deep learning



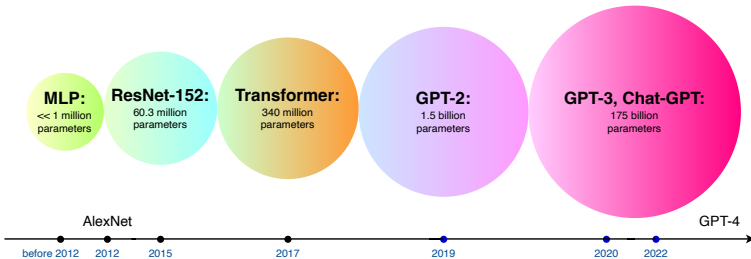
```

class Node {
public:
    // this node is not working like I expect - how do I fix it?

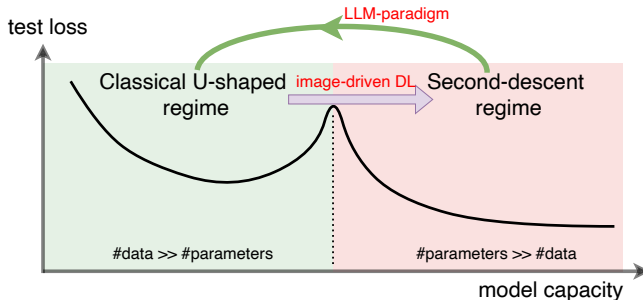
    Node(int data) : data(data), next(nullptr) {}
    Node(int data, Node* next) : data(data), next(next) {}

    int data;
    Node* next;
};

// test it out
int main() {
    Node n1(1);
    Node n2(2, &n1);
    Node n3(3, &n2);
    Node n4(4, &n3);
    Node n5(5, &n4);
    Node n6(6, &n5);
    Node n7(7, &n6);
    Node n8(8, &n7);
    Node n9(9, &n8);
    Node n10(10, &n9);
    Node n11(11, &n10);
    Node n12(12, &n11);
    Node n13(13, &n12);
    Node n14(14, &n13);
    Node n15(15, &n14);
    Node n16(16, &n15);
    Node n17(17, &n16);
    Node n18(18, &n17);
    Node n19(19, &n18);
    Node n20(20, &n19);
    Node n21(21, &n20);
    Node n22(22, &n21);
    Node n23(23, &n22);
    Node n24(24, &n23);
    Node n25(25, &n24);
    Node n26(26, &n25);
    Node n27(27, &n26);
    Node n28(28, &n27);
    Node n29(29, &n28);
    Node n30(30, &n29);
    Node n31(31, &n30);
    Node n32(32, &n31);
    Node n33(33, &n32);
    Node n34(34, &n33);
    Node n35(35, &n34);
    Node n36(36, &n35);
    Node n37(37, &n36);
    Node n38(38, &n37);
    Node n39(39, &n38);
    Node n40(40, &n39);
    Node n41(41, &n40);
    Node n42(42, &n41);
    Node n43(43, &n42);
    Node n44(44, &n43);
    Node n45(45, &n44);
    Node n46(46, &n45);
    Node n47(47, &n46);
    Node n48(48, &n47);
    Node n49(49, &n48);
    Node n50(50, &n49);
    Node n51(51, &n50);
    Node n52(52, &n51);
    Node n53(53, &n52);
    Node n54(54, &n53);
    Node n55(55, &n54);
    Node n56(56, &n55);
    Node n57(57, &n56);
    Node n58(58, &n57);
    Node n59(59, &n58);
    Node n60(60, &n59);
    Node n61(61, &n60);
    Node n62(62, &n61);
    Node n63(63, &n62);
    Node n64(64, &n63);
    Node n65(65, &n64);
    Node n66(66, &n65);
    Node n67(67, &n66);
    Node n68(68, &n67);
    Node n69(69, &n68);
    Node n70(70, &n69);
    Node n71(71, &n70);
    Node n72(72, &n71);
    Node n73(73, &n72);
    Node n74(74, &n73);
    Node n75(75, &n74);
    Node n76(76, &n75);
    Node n77(77, &n76);
    Node n78(78, &n77);
    Node n79(79, &n78);
    Node n80(80, &n79);
    Node n81(81, &n80);
    Node n82(82, &n81);
    Node n83(83, &n82);
    Node n84(84, &n83);
    Node n85(85, &n84);
    Node n86(86, &n85);
    Node n87(87, &n86);
    Node n88(88, &n87);
    Node n89(89, &n88);
    Node n90(90, &n89);
    Node n91(91, &n90);
    Node n92(92, &n91);
    Node n93(93, &n92);
    Node n94(94, &n93);
    Node n95(95, &n94);
    Node n96(96, &n95);
    Node n97(97, &n96);
    Node n98(98, &n97);
    Node n99(99, &n98);
    Node n100(100, &n99);
    Node n101(101, &n100);
    Node n102(102, &n101);
    Node n103(103, &n102);
    Node n104(104, &n103);
    Node n105(105, &n104);
    Node n106(106, &n105);
    Node n107(107, &n106);
    Node n108(108, &n107);
    Node n109(109, &n108);
    Node n110(110, &n109);
    Node n111(111, &n110);
    Node n112(112, &n111);
    Node n113(113, &n112);
    Node n114(114, &n113);
    Node n115(115, &n114);
    Node n116(116, &n115);
    Node n117(117, &n116);
    Node n118(118, &n117);
    Node n119(119, &n118);
    Node n120(120, &n119);
    Node n121(121, &n120);
    Node n122(122, &n121);
    Node n123(123, &n122);
    Node n124(124, &n123);
    Node n125(125, &n124);
    Node n126(126, &n125);
    Node n127(127, &n126);
    Node n128(128, &n127);
    Node n129(129, &n128);
    Node n130(130, &n129);
    Node n131(131, &n130);
    Node n132(132, &n131);
    Node n133(133, &n132);
    Node n134(134, &n133);
    Node n135(135, &n134);
    Node n136(136, &n135);
    Node n137(137, &n136);
    Node n138(138, &n137);
    Node n139(139, &n138);
    Node n140(140, &n139);
    Node n141(141, &n140);
    Node n142(142, &n141);
    Node n143(143, &n142);
    Node n144(144, &n143);
    Node n145(145, &n144);
    Node n146(146, &n145);
    Node n147(147, &n146);
    Node n148(148, &n147);
    Node n149(149, &n148);
    Node n150(150, &n149);
    Node n151(151, &n150);
    Node n152(152, &n151);
    Node n153(153, &n152);
    Node n154(154, &n153);
    Node n155(155, &n154);
    Node n156(156, &n155);
    Node n157(157, &n156);
    Node n158(158, &n157);
    Node n159(159, &n158);
    Node n160(160, &n159);
    Node n161(161, &n160);
    Node n162(162, &n161);
    Node n163(163, &n162);
    Node n164(164, &n163);
    Node n165(165, &n164);
    Node n166(166, &n165);
    Node n167(167, &n166);
    Node n168(168, &n167);
    Node n169(169, &n168);
    Node n170(170, &n169);
    Node n171(171, &n170);
    Node n172(172, &n171);
    Node n173(173, &n172);
    Node n174(174, &n173);
    Node n175(175, &n174);
    Node n176(176, &n175);
    Node n177(177, &n176);
    Node n178(178, &n177);
    Node n179(179, &n178);
    Node n180(180, &n179);
    Node n181(181, &n180);
    Node n182(182, &n181);
    Node n183(183, &n182);
    Node n184(184, &n183);
    Node n185(185, &n184);
    Node n186(186, &n185);
    Node n187(187, &n186);
    Node n188(188, &n187);
    Node n189(189, &n188);
    Node n190(190, &n189);
    Node n191(191, &n190);
    Node n192(192, &n191);
    Node n193(193, &n192);
    Node n194(194, &n193);
    Node n195(195, &n194);
    Node n196(196, &n195);
    Node n197(197, &n196);
    Node n198(198, &n197);
    Node n199(199, &n198);
    Node n200(200, &n199);
    Node n201(201, &n200);
    Node n202(202, &n201);
    Node n203(203, &n202);
    Node n204(204, &n203);
    Node n205(205, &n204);
    Node n206(206, &n205);
    Node n207(207, &n206);
    Node n208(208, &n207);
    Node n209(209, &n208);
    Node n210(210, &n209);
    Node n211(211, &n210);
    Node n212(212, &n211);
    Node n213(213, &n212);
    Node n214(214, &n213);
    Node n215(215, &n214);
    Node n216(216, &n215);
    Node n217(217, &n216);
    Node n218(218, &n217);
    Node n219(219, &n218);
    Node n220(220, &n219);
    Node n221(221, &n220);
    Node n222(222, &n221);
    Node n223(223, &n222);
    Node n224(224, &n223);
    Node n225(225, &n224);
    Node n226(226, &n225);
    Node n227(227, &n226);
    Node n228(228, &n227);
    Node n229(229, &
```



# Learning paradigm transition in the past twenty years



**Figure 1:** Paradigm among test loss, data, and model capacity.

- double descent [5] (Belkin, Hsu, Ma, Mandal, 2019)

scaling law [15]

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

# Learning paradigm transition in the past twenty years

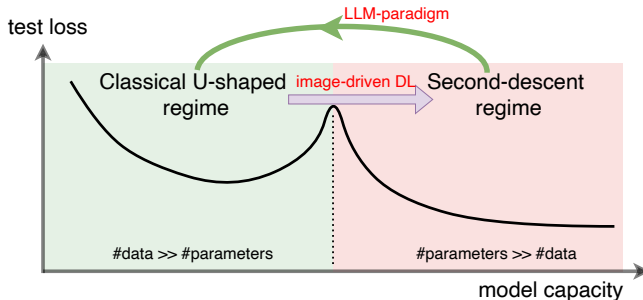


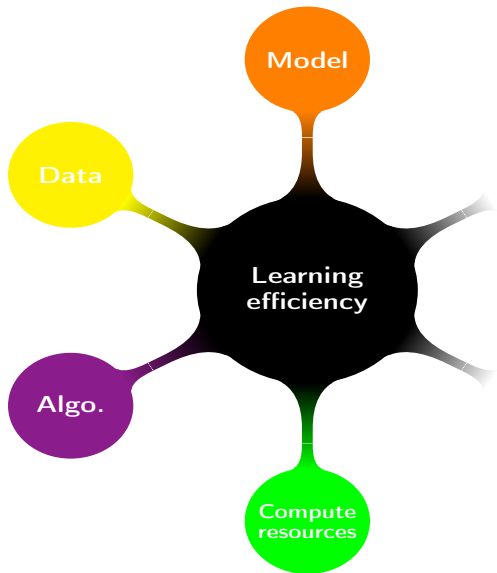
Figure 1: Paradigm among test loss, data, and model capacity.

- double descent [5] (Belkin, Hsu, Ma, Mandal, 2019)

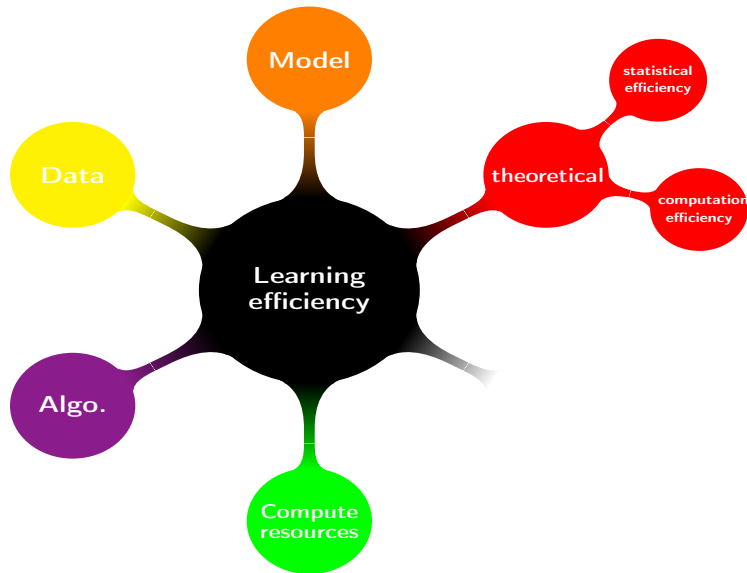
## scaling law [15]

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

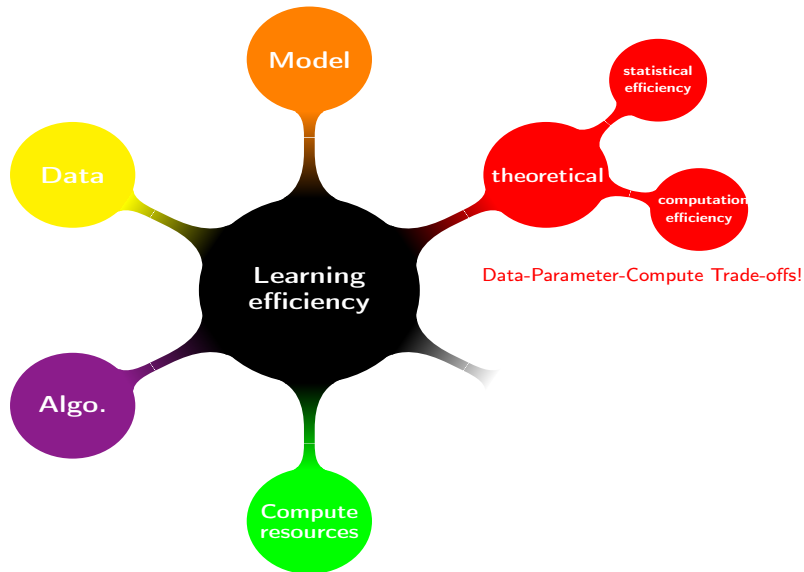
## Pipeline: Learning efficiency - curse of dimensionality



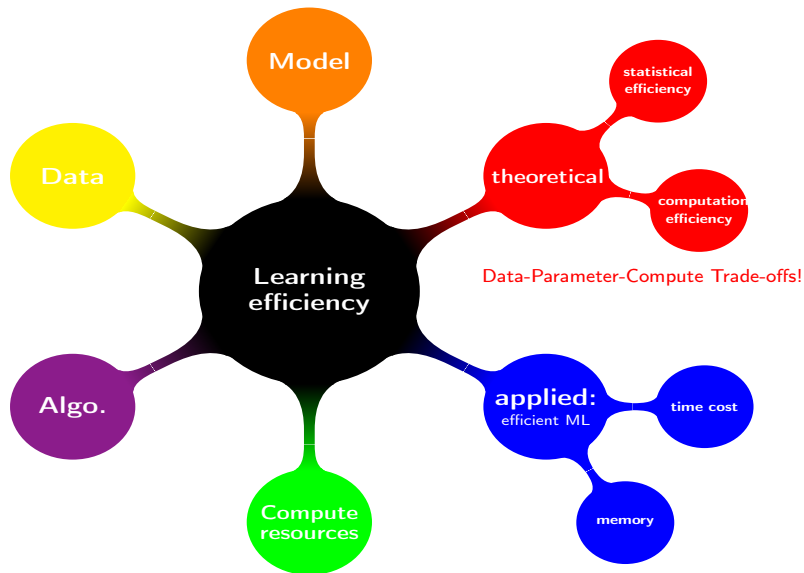
# Pipeline: Learning efficiency - Theory



# Pipeline: Learning efficiency - Theory

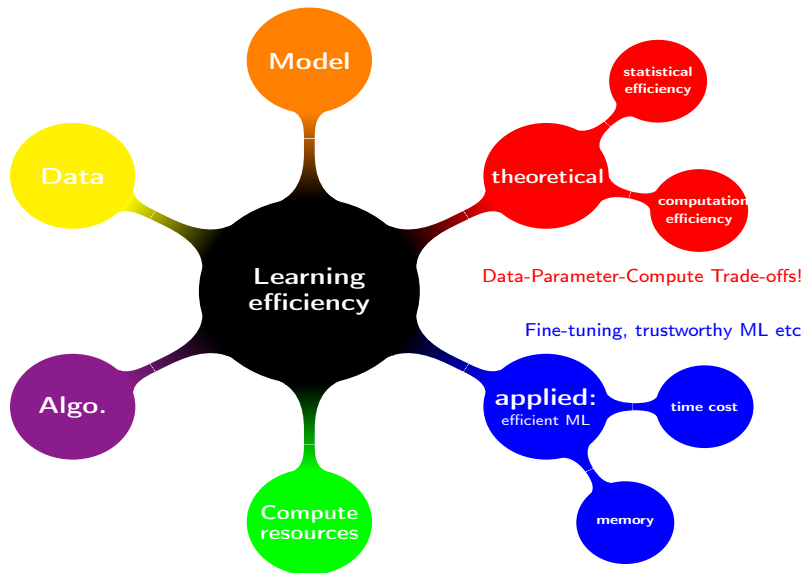


# Pipeline: Learning efficiency - Application





# Pipeline: Learning efficiency - Application



- model size is a good metric?
- Learning with path-norm: the perspective of **model capacity**
- Learning with Barron spaces: the perspective of **function space**
- statistical/computational learning efficiency

## Target

Characterize the “right” model capacity as well as the induced “right” *function spaces* for ML models and track statistical/computational learning efficiency.

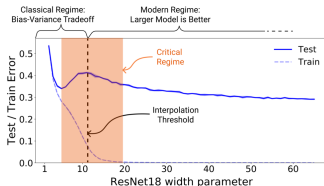
- model size is a good metric?
- Learning with path-norm: the perspective of **model capacity**
- Learning with Barron spaces: the perspective of **function space**
- statistical/computational learning efficiency

## Target

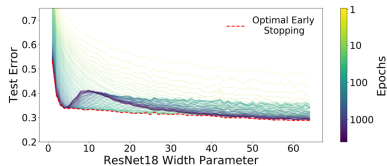
Characterize the “right” model capacity as well as the induced “right” *function spaces* for ML models and track statistical/computational learning efficiency.

# Model size is a “right” complexity?

- double descent can disappear!



(a) Results on ResNet18 [20]

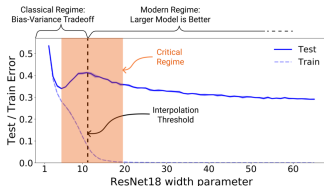


(b) Optimal early stopping [20].

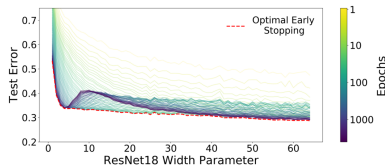
- Empirically: neural network pruning [18], lottery ticket hypothesis [13], fine-tuning with large dropout [31]
- Theoretically: how much over-parameterization is sufficient [8, 28]

# Model size is a “right” complexity?

- double descent can disappear!



(a) Results on ResNet18 [20]



(b) Optimal early stopping [20].

- Empirically: neural network pruning [18], lottery ticket hypothesis [13], fine-tuning with large dropout [31]
- Theoretically: how much over-parameterization is sufficient [8, 28]

# 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:[21, 25, 22, 10]

name	definition	rank correlation
Parameter Frobenius norm	$\sum_{i=1}^L \ W_i\ _F^2$	0.073
Frobenius distance to initialization [19]	$\sum_{i=1}^L \ W_i - W_i^0\ _F^2$	-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 [16]	$\frac{(L+1)^2}{n} \sum_{i=1}^n \langle W, \nabla_W \ell(h_W(x_i), y_i) \rangle$	0.078
Path-norm [21]	$\sum_{(i_0, \dots, i_L)} \prod_{j=1}^L (W_{i_j, i_{j-1}})^2$	0.373

Table 1: Complexity measures compared in the empirical study [14], 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: [21, 25, 22, 10]

name	definition	rank correlation
Parameter Frobenius norm	$\sum_{i=1}^L \ W_i\ _F^2$	0.073
Frobenius distance to initialization [19]	$\sum_{i=1}^L \ W_i - W_i^0\ _F^2$	-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 [16]	$\frac{(L+1)^2}{n} \sum_{i=1}^n \langle W, \nabla_W \ell(h_W(x_i), y_i) \rangle$	0.078
Path-norm [21]	$\sum_{(i_0, \dots, i_L)} \prod_{j=1}^L (W_{i_j, i_{j-1}})^2$	0.373

Table 1: Complexity measures compared in the empirical study [14], 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: [21, 25, 22, 10]

name	definition	rank correlation
Parameter Frobenius norm	$\sum_{i=1}^L \ W_i\ _F^2$	0.073
Frobenius distance to initialization [19]	$\sum_{i=1}^L \ W_i - W_i^0\ _F^2$	-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 [16]	$\frac{(L+1)^2}{n} \sum_{i=1}^n \langle W, \nabla_W \ell(h_W(x_i), y_i) \rangle$	0.078
Path-norm [21]	$\sum_{(i_0, \dots, i_L)} \prod_{j=1}^L (W_{i_j, i_{j-1}})^2$	0.373

Table 1: Complexity measures compared in the empirical study [14], 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](#))

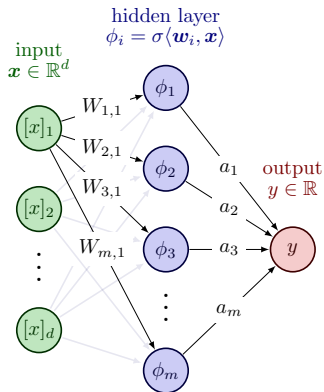
The size of the weights is more important than the size of the network!

Norm-based capacity: [21, 25, 22, 10]

name	definition	rank correlation
Parameter Frobenius norm	$\sum_{i=1}^L \ \mathbf{W}_i\ _F^2$	0.073
Frobenius distance to initialization [19]	$\sum_{i=1}^L \ \mathbf{W}_i - \mathbf{W}_i^0\ _F^2$	-0.263
Spectral complexity [4]	$\prod_{i=1}^L \ \mathbf{W}_i\  \left( \sum_{i=1}^L \frac{\ \mathbf{W}_i\ _{2,1}^{3/2}}{\ \mathbf{W}_i\ ^{3/2}} \right)^{2/3}$	-0.537
Fisher-Rao [16]	$\frac{(L+1)^2}{n} \sum_{i=1}^n \langle \mathbf{W}, \nabla_{\mathbf{W}} \ell(h_{\mathbf{W}}(\mathbf{x}_i), y_i) \rangle$	0.078
Path-norm [21]	$\sum_{(i_0, \dots, i_L)} \prod_{j=1}^L (\mathbf{W}_{i_j, i_{j-1}})^2$	0.373

**Table 1:** Complexity measures compared in the empirical study [14], and their correlation with generalization

# Two-layer neural networks, path norm



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

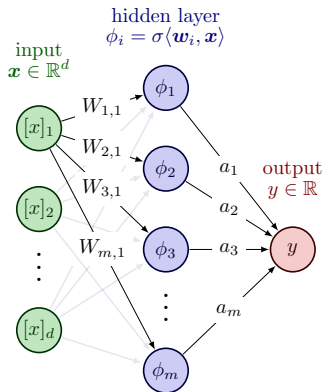
$\ell_1$ -path norm

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

- semi-norm
- relations to Barron spaces  $\mathcal{B}$  [2, 12]

$$\|f\|_{\mathcal{B}} \leq \|\theta\|_{\mathcal{P}} \leq 2\|f\|_{\mathcal{B}}$$

# Two-layer neural networks, path norm



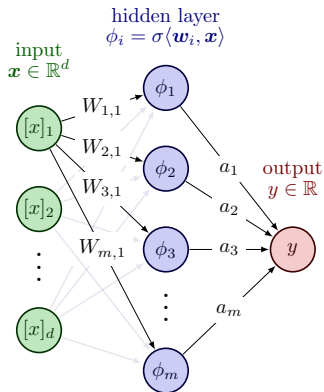
$$\mathcal{P}_m = \{f_{\theta}(\cdot) := \frac{1}{m} \sum_{k=1}^m a_k \phi(\langle \mathbf{w}_k, \cdot \rangle)\}$$

**$\ell_1$ -path norm**

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

- semi-norm
- relations to Barron spaces  $\mathcal{B}$  [2, 12]  
 $\|f\|_{\mathcal{B}} \leq \|\theta\|_{\mathcal{P}} \leq 2\|f\|_{\mathcal{B}}$

# Two-layer neural networks, path norm



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

**$\ell_1$ -path norm**

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

- semi-norm
- relations to Barron spaces  $\mathcal{B}$  [2, 12]

$$\|f\|_{\mathcal{B}} \leq \|\boldsymbol{\theta}\|_{\mathcal{P}} \leq 2\|f\|_{\mathcal{B}}$$

## \*Path norm, Barron spaces, RKHS [11, 6]

Consider a random features model (RFM) [23, 17]

- first layer:  $\mathbf{w} \stackrel{iid}{\sim} \mu \in \mathcal{P}(\mathcal{W})$ ; only train the second layer

infinite many features  $f_a(\mathbf{x}) = \int_{\mathcal{W}} a(\mathbf{w}) \phi(\mathbf{x}, \mathbf{w}) d\mu(\mathbf{w})$

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

- RFMs  $\equiv$  kernel methods by taking  $p = 2$  using Representer theorem [24]
- 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  $p \geq q$

For any  $1 \leq p \leq \infty$ , define

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

- largest
- data-adaptive

## \*Path norm, Barron spaces, RKHS [11, 6]

Consider a random features model (RFM) [23, 17]

- first layer:  $\mathbf{w} \stackrel{iid}{\sim} \mu \in \mathcal{P}(\mathcal{W})$ ; only train the second layer

infinite many features  $f_a(\mathbf{x}) = \int_{\mathcal{W}} a(\mathbf{w}) \phi(\mathbf{x}, \mathbf{w}) d\mu(\mathbf{w})$

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

- RFMs  $\equiv$  kernel methods by taking  $p = 2$  using Representer theorem [24]
- 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  $p \geq q$

For any  $1 \leq p \leq \infty$ , define

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

- largest
- data-adaptive

## \*Path norm, Barron spaces, RKHS [11, 6]

Consider a random features model (RFM) [23, 17]

- first layer:  $\mathbf{w} \stackrel{iid}{\sim} \mu \in \mathcal{P}(\mathcal{W})$ ; only train the second layer

infinite many features  $f_a(\mathbf{x}) = \int_{\mathcal{W}} a(\mathbf{w}) \phi(\mathbf{x}, \mathbf{w}) d\mu(\mathbf{w})$

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

- RFMs  $\equiv$  kernel methods by taking  $p = 2$  using Representer theorem [24]
- 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  $p \geq q$

For any  $1 \leq p \leq \infty$ , define

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

- largest
- data-adaptive

## \*Path norm, Barron spaces, RKHS [11, 6]

Consider a random features model (RFM) [23, 17]

- first layer:  $\mathbf{w} \stackrel{iid}{\sim} \mu \in \mathcal{P}(\mathcal{W})$ ; only train the second layer

infinite many features  $f_a(\mathbf{x}) = \int_{\mathcal{W}} a(\mathbf{w}) \phi(\mathbf{x}, \mathbf{w}) d\mu(\mathbf{w})$

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

- RFMs  $\equiv$  kernel methods by taking  $p = 2$  using Representer theorem [24]
- 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  $p \geq q$

For any  $1 \leq p \leq \infty$ , define

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

- largest
- data-adaptive



## \*Path norm, Barron spaces, RKHS [11, 6]

Consider a random features model (RFM) [23, 17]

- first layer:  $\mathbf{w} \stackrel{iid}{\sim} \mu \in \mathcal{P}(\mathcal{W})$ ; only train the second layer

infinite many features  $f_a(\mathbf{x}) = \int_{\mathcal{W}} a(\mathbf{w}) \phi(\mathbf{x}, \mathbf{w}) d\mu(\mathbf{w})$

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

- RFMs  $\equiv$  kernel methods by taking  $p = 2$  using Representer theorem [24]
- 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  $p \geq q$

For any  $1 \leq p \leq \infty$ , define

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

- largest
- data-adaptive

## \*Path norm, Barron spaces, RKHS [11, 6]

Consider a random features model (RFM) [23, 17]

- first layer:  $\mathbf{w} \stackrel{iid}{\sim} \mu \in \mathcal{P}(\mathcal{W})$ ; only train the second layer

infinite many features  $f_a(\mathbf{x}) = \int_{\mathcal{W}} a(\mathbf{w}) \phi(\mathbf{x}, \mathbf{w}) d\mu(\mathbf{w})$

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

- RFMs  $\equiv$  kernel methods by taking  $p = 2$  using Representer theorem [24]
- 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  $p \geq q$

For any  $1 \leq p \leq \infty$ , define

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

- largest
- data-adaptive

# Our results: statistical guarantees

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

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

**Theorem (Liu, Dadi, Cevher, JMLR 2024)**

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

$$\|\hat{f}_{\theta} - f^*\|_{L^2_{\rho_X}}^2 \lesssim \frac{R^2}{m} + R^2 d^{\frac{1}{3}} n^{-\frac{d+2}{2d+2}} \quad w.h.p.$$

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

# Our results: statistical guarantees

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

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

## Theorem (Liu, Dadi, Cevher, JMLR 2024)

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

$$\|\hat{f}_{\theta} - f^*\|_{L^2_{\rho_X}}^2 \lesssim \frac{R^2}{m} + R^2 d^{\frac{1}{3}} n^{-\frac{d+2}{2d+2}} \quad w.h.p.$$

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

# Our results: statistical guarantees

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

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

## Theorem (Liu, Dadi, Cevher, JMLR 2024)

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

$$\|\hat{f}_{\theta} - f^*\|_{L^2_{\rho_X}}^2 \lesssim \frac{R^2}{m} + R^2 d^{\frac{1}{3}} n^{-\frac{d+2}{2d+2}} \quad w.h.p.$$

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

## Metric entropy: minimax?

### Proposition (metric entropy)

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

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

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

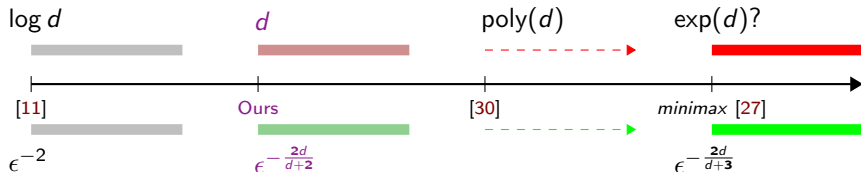
# Metric entropy: minimax?

## Proposition (metric entropy)

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

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

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



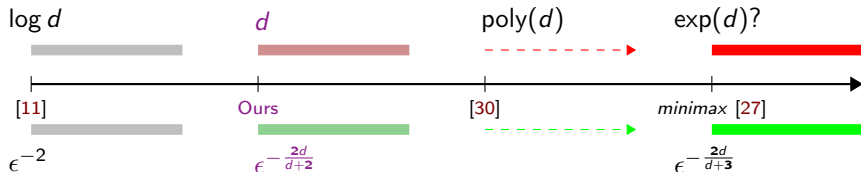
# Metric entropy: minimax?

## Proposition (metric entropy)

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

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

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



The “best” trade-off between  $\epsilon$  and  $d$ .



# Proof sketch: convex hull technique and its constant!

- Consider the following function space

$$\mathcal{F} = \{\sigma(\langle \tilde{\mathbf{w}}, \cdot \rangle) : \tilde{\mathbf{w}} \in \mathcal{W}\} \cup \{0\} \cup \{-\sigma(\langle \tilde{\mathbf{w}}, \cdot \rangle) : \tilde{\mathbf{w}} \in \mathbb{S}_1^{d-1} \text{ with the } \ell_1 \text{ ball}\}$$

- the convex hull of  $\mathcal{F}$  is

$$\overline{\text{conv}} \mathcal{F} = \left\{ \sum_{i=1}^m \alpha_i f_i \mid f_i \in \mathcal{F}, \sum_{i=1}^m \alpha_i = 1, \alpha_i \geq 0, m \in \mathbb{N} \right\}.$$

- convex hull technique [29, Theorem 2.6.9]

$$\log \mathcal{N}_2(\mathcal{G}_1, \epsilon) \leq \log \mathcal{N}_2(\overline{\text{conv}} \mathcal{F}, \epsilon, \mu) \leq C \left( \frac{1}{\epsilon} \right)^{\frac{2d}{d+2}}.$$

- control the constant  $C$

$$C := \underbrace{D_k}_{=\Theta(d)} \left[ \underbrace{C_k}_{=\Theta(1)} (2^{d+1} + 1)^{\frac{1}{d}} \right]^{\frac{2d}{d+2}} \leq 10^7 d \quad \text{if } d > 5$$

# Proof sketch: convex hull technique and its constant!

- Consider the following function space

$$\mathcal{F} = \{\sigma(\langle \tilde{\mathbf{w}}, \cdot \rangle) : \tilde{\mathbf{w}} \in \mathcal{W}\} \cup \{0\} \cup \{-\sigma(\langle \tilde{\mathbf{w}}, \cdot \rangle) : \tilde{\mathbf{w}} \in \mathbb{S}_1^{d-1} \text{ with the } \ell_1 \text{ ball}\}$$

- the convex hull of  $\mathcal{F}$  is

$$\overline{\text{conv}}\mathcal{F} = \left\{ \sum_{i=1}^m \alpha_i f_i \mid f_i \in \mathcal{F}, \sum_{i=1}^m \alpha_i = 1, \alpha_i \geq 0, m \in \mathbb{N} \right\}.$$

- convex hull technique [29, Theorem 2.6.9]

$$\log \mathcal{N}_2(\mathcal{G}_1, \epsilon) \leq \log \mathcal{N}_2(\overline{\text{conv}}\mathcal{F}, \epsilon, \mu) \leq C \left( \frac{1}{\epsilon} \right)^{\frac{2d}{d+2}}.$$

- control the constant  $C$

$$C := \underbrace{D_k}_{=\Theta(d)} \left[ \underbrace{C_k}_{=\Theta(1)} (2^{d+1} + 1)^{\frac{1}{d}} \right]^{\frac{2d}{d+2}} \leq 10^7 d \quad \text{if } d > 5$$

# Proof sketch: convex hull technique and its constant!

- Consider the following function space

$$\mathcal{F} = \{\sigma(\langle \tilde{\mathbf{w}}, \cdot \rangle) : \tilde{\mathbf{w}} \in \mathcal{W}\} \cup \{0\} \cup \{-\sigma(\langle \tilde{\mathbf{w}}, \cdot \rangle) : \tilde{\mathbf{w}} \in \mathbb{S}_1^{d-1} \text{ with the } \ell_1 \text{ ball}\}$$

- the convex hull of  $\mathcal{F}$  is

$$\overline{\text{conv}} \mathcal{F} = \left\{ \sum_{i=1}^m \alpha_i f_i \mid f_i \in \mathcal{F}, \sum_{i=1}^m \alpha_i = 1, \alpha_i \geq 0, m \in \mathbb{N} \right\}.$$

- convex hull technique [29, Theorem 2.6.9]

$$\log \mathcal{N}_2(\mathcal{G}_1, \epsilon) \leq \log \mathcal{N}_2(\overline{\text{conv}} \mathcal{F}, \epsilon, \mu) \leq \mathbf{C} \left( \frac{1}{\epsilon} \right)^{\frac{2d}{d+2}}.$$

- control the constant  $\mathbf{C}$

$$\mathbf{C} := \underbrace{D_k}_{=\Theta(d)} \left[ \underbrace{C_k}_{=\Theta(1)} (2^{d+1} + 1)^{\frac{1}{d}} \right]^{\frac{2d}{d+2}} \leq 10^7 d \quad \text{if } d > 5$$

# Proof sketch: convex hull technique and its constant!

- Consider the following function space

$$\mathcal{F} = \{\sigma(\langle \tilde{\mathbf{w}}, \cdot \rangle) : \tilde{\mathbf{w}} \in \mathcal{W}\} \cup \{0\} \cup \{-\sigma(\langle \tilde{\mathbf{w}}, \cdot \rangle) : \tilde{\mathbf{w}} \in \mathbb{S}_1^{d-1} \text{ with the } \ell_1 \text{ ball}\}$$

- the convex hull of  $\mathcal{F}$  is

$$\overline{\text{conv}} \mathcal{F} = \left\{ \sum_{i=1}^m \alpha_i f_i \mid f_i \in \mathcal{F}, \sum_{i=1}^m \alpha_i = 1, \alpha_i \geq 0, m \in \mathbb{N} \right\}.$$

- convex hull technique [29, Theorem 2.6.9]

$$\log \mathcal{N}_2(\mathcal{G}_1, \epsilon) \leq \log \mathcal{N}_2(\overline{\text{conv}} \mathcal{F}, \epsilon, \mu) \leq \mathcal{C} \left( \frac{1}{\epsilon} \right)^{\frac{2d}{d+2}}.$$

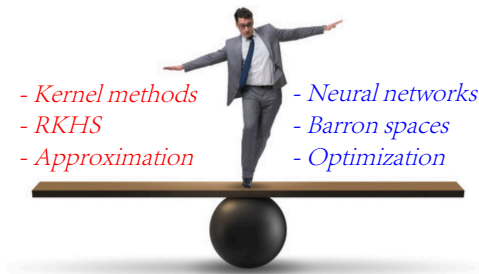
- control the constant  $\mathcal{C}$

$$\mathcal{C} := \underbrace{D_k}_{=\Theta(d)} \left[ \underbrace{C_k}_{=\Theta(1)} (2^{d+1} + 1)^{\frac{1}{d}} \right]^{\frac{2d}{d+2}} \leq 10^7 d \quad \text{if } d > 5$$

Optimization in Barron spaces is NP hard: curse of dimensionality! [1]

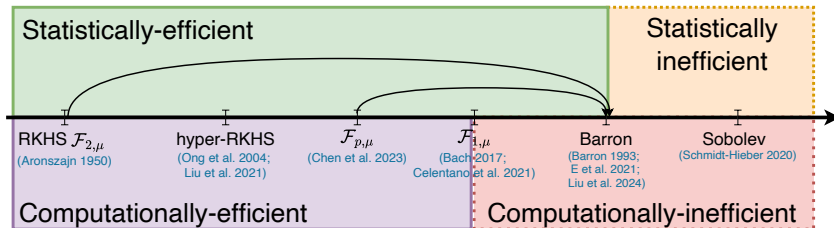
# Computational-statistical gaps

Optimization in Barron spaces is NP hard: curse of dimensionality! [1]



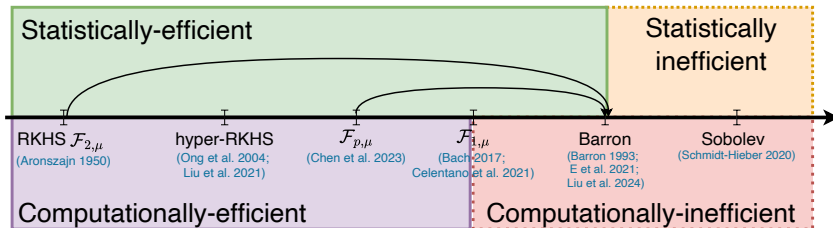
# Computational-statistical gaps

Optimization in Barron spaces is NP hard: curse of dimensionality! [1]



# Computational-statistical gaps

Optimization in Barron spaces is NP hard: curse of dimensionality! [1]



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



# Learning with multiple ReLU neurons

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

$$f^*(\mathbf{x}) = \sum_{j=1}^k a_j \sigma(\langle \mathbf{v}_j, \mathbf{x} \rangle), k = \mathcal{O}(1)$$

$$\|\hat{f} - f^*\|_{L^2(d\mu)} \leq \epsilon \text{ from } \{\mathbf{x}_i, f^*(\mathbf{x}_i)\}_{i=1}^n \text{ with } \mathbf{x}_i \sim \mathcal{N}(0, I_d)$$

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

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

- correlational statistical query (CSQ):  $|\tilde{q} - \mathbb{E}_{\mathbf{x}, y}[\psi(\mathbf{x})y]| \leq \tau$
- the lower bound for any CSQ algorithm is  $d^{\Omega(k)}$  [9]

# Learning with multiple ReLU neurons

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

$$f^*(\mathbf{x}) = \sum_{j=1}^k a_j \sigma(\langle \mathbf{v}_j, \mathbf{x} \rangle), k = \mathcal{O}(1)$$

$$\|\hat{f} - f^*\|_{L^2(d\mu)} \leq \epsilon \text{ from } \{\mathbf{x}_i, f^*(\mathbf{x}_i)\}_{i=1}^n \text{ with } \mathbf{x}_i \sim \mathcal{N}(0, I_d)$$

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

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

- correlational statistical query (CSQ):  $|\tilde{q} - \mathbb{E}_{\mathbf{x}, y}[\psi(\mathbf{x})y]| \leq \tau$
- the lower bound for any CSQ algorithm is  $d^{\Omega(k)}$  [9]

# Learning with multiple ReLU neurons

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

$$f^*(\mathbf{x}) = \sum_{j=1}^k a_j \sigma(\langle \mathbf{v}_j, \mathbf{x} \rangle), k = \mathcal{O}(1)$$

$$\|\hat{f} - f^*\|_{L^2(d\mu)} \leq \epsilon \text{ from } \{\mathbf{x}_i, f^*(\mathbf{x}_i)\}_{i=1}^n \text{ with } \mathbf{x}_i \sim \mathcal{N}(0, I_d)$$

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

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

- correlational statistical query (CSQ):  $|\tilde{q} - \mathbb{E}_{\mathbf{x}, y}[\psi(\mathbf{x})y]| \leq \tau$
- the lower bound for any CSQ algorithm is  $d^{\Omega(k)}$  [9]

# Learning with multiple ReLU neurons

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

$$f^*(\mathbf{x}) = \sum_{j=1}^k a_j \sigma(\langle \mathbf{v}_j, \mathbf{x} \rangle), k = \mathcal{O}(1)$$

$$\|\hat{f} - f^*\|_{L^2(d\mu)} \leq \epsilon \text{ from } \{\mathbf{x}_i, f^*(\mathbf{x}_i)\}_{i=1}^n \text{ with } \mathbf{x}_i \sim \mathcal{N}(0, I_d)$$

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

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

- correlational statistical query (CSQ):  $|\tilde{q} - \mathbb{E}_{\mathbf{x}, y}[\psi(\mathbf{x})y]| \leq \tau$
- the lower bound for any CSQ algorithm is  $d^{\Omega(k)}$  [9]

# Learning with multiple ReLU neurons

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

$$f^*(\mathbf{x}) = \sum_{j=1}^k a_j \sigma(\langle \mathbf{v}_j, \mathbf{x} \rangle), k = \mathcal{O}(1)$$

$$\|\hat{f} - f^*\|_{L^2(d\mu)} \leq \epsilon \text{ from } \{\mathbf{x}_i, f^*(\mathbf{x}_i)\}_{i=1}^n \text{ with } \mathbf{x}_i \sim \mathcal{N}(0, I_d)$$

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

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

- correlational statistical query (CSQ):  $|\tilde{q} - \mathbb{E}_{\mathbf{x}, y}[\psi(\mathbf{x})y]| \leq \tau$
- the lower bound for any CSQ algorithm is  $d^{\Omega(k)}$  [9]

# How to do it by gradient descent

## Learning multi ReLU neurons by two-layer NN via GD

$$L(\mathbf{W}) = \frac{1}{2} \mathbb{E}_{\mathbf{x} \sim \mathcal{N}(0, I_d)} \left( \sum_{i=1}^m \sigma(\langle \mathbf{w}_i, \mathbf{x} \rangle) - f^*(\mathbf{x}) \right)^2$$

- Gaussian initialization for student neurons:  $\mathbf{w}_i \sim \mathcal{N}(0, \sigma^2 I_d)$
- angle:  $\theta_{ij} \triangleq \angle(\mathbf{w}_i, \mathbf{v}_j)$ ,  $\varphi_{ij} \triangleq \angle(\mathbf{w}_i, \mathbf{w}_j)$

## Assumption

- *diverse teacher neurons*:  $\{\mathbf{v}_j\}_{j=1}^k$  are *orthogonal* and  $\|\mathbf{v}_j\|_2 = \text{const}$
- *warm start*: the *smallest* angle not close to orthogonal
  - *weak recovery*:  $\langle \mathbf{w}_i, \mathbf{v}_{i^*} \rangle \gg \langle \mathbf{w}_i, \mathbf{v}_j \rangle$

# How to do it by gradient descent

## Learning multi ReLU neurons by two-layer NN via GD

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

- Gaussian initialization for student neurons:  $\mathbf{w}_i \sim \mathcal{N}(0, \sigma^2 \mathbf{I}_d)$
- angle:  $\theta_{ij} \triangleq \angle(\mathbf{w}_i, \mathbf{v}_j)$ ,  $\varphi_{ij} \triangleq \angle(\mathbf{w}_i, \mathbf{w}_j)$

## Assumption

- *diverse teacher neurons*:  $\{\mathbf{v}_j\}_{j=1}^k$  are *orthogonal* and  $\|\mathbf{v}_j\|_2 = \text{const}$
- *warm start*: the *smallest* angle not close to orthogonal
  - *weak recovery*:  $\langle \mathbf{w}_i, \mathbf{v}_{i^*} \rangle \gg \langle \mathbf{w}_i, \mathbf{v}_j \rangle$

# How to do it by gradient descent

## Learning multi ReLU neurons by two-layer NN via GD

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

- Gaussian initialization for student neurons:  $\mathbf{w}_i \sim \mathcal{N}(0, \sigma^2 \mathbf{I}_d)$
- angle:  $\theta_{ij} \triangleq \angle(\mathbf{w}_i, \mathbf{v}_j)$ ,  $\varphi_{ij} \triangleq \angle(\mathbf{w}_i, \mathbf{w}_j)$

## Assumption

- *diverse teacher neurons*:  $\{\mathbf{v}_j\}_{j=1}^k$  are **orthogonal** and  $\|\mathbf{v}_j\|_2 = \text{const}$
- *warm start*: the **smallest** angle not close to orthogonal
  - *weak recovery*:  $\langle \mathbf{w}_i, \mathbf{v}_{i^*} \rangle \gg \langle \mathbf{w}_i, \mathbf{v}_j \rangle$



# How to do it by gradient descent

## Learning multi ReLU neurons by two-layer NN via GD

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

- Gaussian initialization for student neurons:  $\mathbf{w}_i \sim \mathcal{N}(0, \sigma^2 \mathbf{I}_d)$
- angle:  $\theta_{ij} \triangleq \angle(\mathbf{w}_i, \mathbf{v}_j)$ ,  $\varphi_{ij} \triangleq \angle(\mathbf{w}_i, \mathbf{w}_j)$

## Assumption

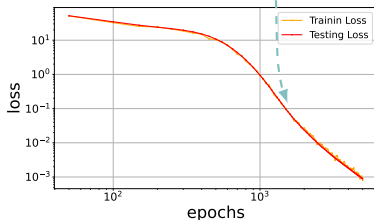
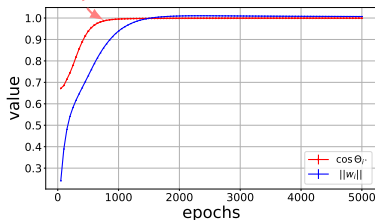
- *diverse teacher neurons*:  $\{\mathbf{v}_j\}_{j=1}^k$  are **orthogonal** and  $\|\mathbf{v}_j\|_2 = \text{const}$
- *warm start*: the **smallest** angle not close to orthogonal
  - *weak recovery*:  $\langle \mathbf{w}_i, \mathbf{v}_{i^*} \rangle \gg \langle \mathbf{w}_i, \mathbf{v}_j \rangle$

# How does GD learn features and recover teacher neurons?

- align  $\theta_{i*} \rightarrow 0$

norm converge

then fit



Theorem (Zhu, Liu, Cevher, 2024)

For sufficiently small initialization and step-size  $\sigma, \eta = o(m^{-k^2})$ , then after weak recovery, there exists a time  $T_0 = \frac{1}{\eta}$  such that  $\forall T \in \mathbb{N}$  and  $i \in [m]$ ,

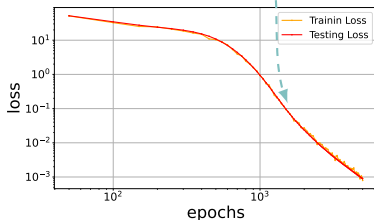
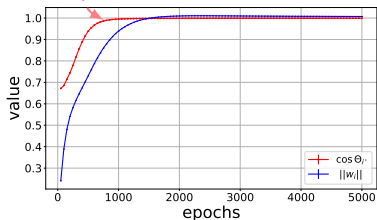
$$L(\mathcal{W}(T + T_0)) \leq \mathcal{O}\left(\frac{1}{T^3}\right), \|\mathbf{w}_i(T + T_0)\|_2 = \Theta\left(\frac{k\|\mathbf{v}\|_2}{m}\right) \text{ w.h.p.}$$

# How does GD learn features and recover teacher neurons?

- align  $\theta_{i*} \rightarrow 0$

norm converge

then fit



## Theorem (Zhu, Liu, Cevher, 2024)

For sufficiently small initialization and step-size  $\sigma, \eta = o(m^{-k^2})$ , then after weak recovery, there exists a time  $T_0 = \frac{1}{\eta}$  such that  $\forall T \in \mathbb{N}$  and  $i \in [m]$ ,

$$L(\mathbf{W}(T + T_0)) \leq \mathcal{O}\left(\frac{1}{T^3}\right), \|\mathbf{w}_i(T + T_0)\|_2 = \Theta\left(\frac{k\|\mathbf{v}\|_2}{m}\right) \text{ w.h.p.}$$

# Proof Sketch

- gradient expression  $\nabla_i(t)$ : a function of  $\|\mathbf{w}_i\|, \|\mathbf{v}_i\|, \theta_{il}, \varphi_{ij}$
- Phase 1 (amplify alignment)
  - $\langle \mathbf{w}_i(t), \nabla_i(t) \rangle \leq 0 \Rightarrow \|\mathbf{w}_i(t)\|$  increasing
  - track  $\theta_{i^*}(t)$ : linear convergence of  $\sin \theta_{i^*} \Rightarrow \theta_{i^*}(t) \leq \epsilon_1$

- Phase 2 (norm convergence)

- norm-balanced

$$\frac{\|\mathbf{w}_i(t)\|}{\|\mathbf{w}_j(t)\|} = \Theta(1), \forall i, j \in [m], \quad T_1 \leq t \leq T_2.$$

- track  $\theta_{i^*}(t)$  and obtain  $L(\mathbf{W}(T_2)) \leq \epsilon_2$

- Phase 3 (local convergence)

gradient lower bound + local smoothness of the loss  $L$

$\Rightarrow$  classical optimization results

# Proof Sketch

- gradient expression  $\nabla_i(t)$ : a function of  $\|\mathbf{w}_i\|, \|\mathbf{v}_i\|, \theta_{il}, \varphi_{ij}$
- Phase 1 (amplify alignment)
  - $\langle \mathbf{w}_i(t), \nabla_i(t) \rangle \leq 0 \Rightarrow \|\mathbf{w}_i(t)\|$  increasing
  - track  $\theta_{i^*}(t)$ : linear convergence of  $\sin \theta_{i^*} \Rightarrow \theta_{i^*}(t) \leq \epsilon_1$

- Phase 2 (norm convergence)

- norm-balanced

$$\frac{\|\mathbf{w}_i(t)\|}{\|\mathbf{w}_j(t)\|} = \Theta(1), \forall i, j \in [m], \quad T_1 \leq t \leq T_2.$$

- track  $\theta_{i^*}(t)$  and obtain  $L(\mathbf{W}(T_2)) \leq \epsilon_2$

- Phase 3 (local convergence)

gradient lower bound + local smoothness of the loss  $L$

$\Rightarrow$  classical optimization results

# Proof Sketch

- gradient expression  $\nabla_i(t)$ : a function of  $\|\mathbf{w}_i\|, \|\mathbf{v}_i\|, \theta_{il}, \varphi_{ij}$
- Phase 1 (amplify alignment)
  - $\langle \mathbf{w}_i(t), \nabla_i(t) \rangle \leq 0 \Rightarrow \|\mathbf{w}_i(t)\|$  increasing
  - track  $\theta_{i^*}(t)$ : linear convergence of  $\sin \theta_{i^*} \Rightarrow \theta_{i^*}(t) \leq \epsilon_1$

- Phase 2 (norm convergence)

- norm-balanced

$$\frac{\|\mathbf{w}_i(t)\|}{\|\mathbf{w}_j(t)\|} = \Theta(1), \forall i, j \in [m], \quad T_1 \leq t \leq T_2.$$

- track  $\theta_{i^*}(t)$  and obtain  $L(\mathbf{W}(T_2)) \leq \epsilon_2$

- Phase 3 (local convergence)

gradient lower bound + local smoothness of the loss  $L$   
 $\Rightarrow$  classical optimization results

# Proof Sketch

- gradient expression  $\nabla_i(t)$ : a function of  $\|\mathbf{w}_i\|, \|\mathbf{v}_i\|, \theta_{il}, \varphi_{ij}$
- Phase 1 (amplify alignment)
  - $\langle \mathbf{w}_i(t), \nabla_i(t) \rangle \leq 0 \Rightarrow \|\mathbf{w}_i(t)\|$  increasing
  - track  $\theta_{i^*}(t)$ : linear convergence of  $\sin \theta_{i^*} \Rightarrow \theta_{i^*}(t) \leq \epsilon_1$

- Phase 2 (norm convergence)

- norm-balanced

$$\frac{\|\mathbf{w}_i(t)\|}{\|\mathbf{w}_j(t)\|} = \Theta(1), \forall i, j \in [m], T_1 \leq t \leq T_2.$$

- track  $\theta_{i^*}(t)$  and obtain  $L(\mathbf{W}(T_2)) \leq \epsilon_2$

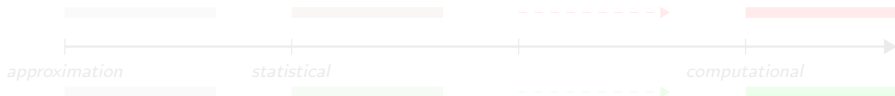
- Phase 3 (local convergence)

gradient lower bound + local smoothness of the loss  $L$

$\Rightarrow$  classical optimization results

# Take-away messages

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

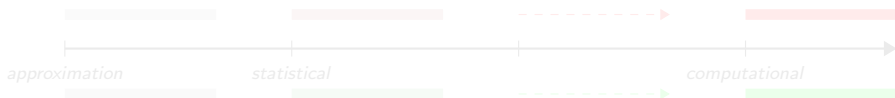


- robust overfitting from the perspective of **approximation** [26]  
well-separated data + target function is smooth enough + perturbation is small enough  
 $\Rightarrow$  **Avoid robust overfitting!**
- LoRA with low-rank dynamics: from **statistical** to **computational**



# Take-away messages

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



- robust overfitting from the perspective of **approximation** [26]  
well-separated data + target function is smooth enough + perturbation is small enough  
 $\Rightarrow$  **Avoid robust overfitting!**
- LoRA with low-rank dynamics: from **statistical** to **computational**

# Take-away messages

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



- robust overfitting from the perspective of **approximation** [26]  
well-separated data + target function is smooth enough + perturbation is small enough  
 $\Rightarrow$  **Avoid robust overfitting!**
- LoRA with low-rank dynamics: from **statistical** to **computational**

# Take-away messages

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



- robust overfitting from the perspective of **approximation** [26]  
well-separated data + target function is smooth enough + perturbation is small enough

$\Rightarrow$  **Avoid robust overfitting!**

- LoRA with low-rank dynamics: from statistical to computational

# Take-away messages

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



- robust overfitting from the perspective of **approximation** [26]  
well-separated data + target function is smooth enough + perturbation is small enough  
 $\Rightarrow$  **Avoid robust overfitting!**
- LoRA with low-rank dynamics: from **statistical** to **computational**

# We're organizing Fine-tuning workshop at NeurIPS 2024!

Fine-Tuning in Modern Machine Learning: Principles and Scalability

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

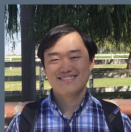
## Invited Speakers



Azalia Mirhoseini  
(Stanford/DeepMind)



Jason Lee (Princeton)



Yuandong Tian (Meta AI)



Quanquan Gu (UCLA)

## Panelist



Danqi Chen  
(Princeton)



Tri Dao  
(Princeton)



Anna Goodie  
(Stanford/DeepMind)



Quanquan Gu  
(UCLA)



Taiji Suzuki  
(UTokyo/RIKEN)



Yuandong Tian  
(Meta)



Leena C. Vankadara  
(Amazon Research)



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.



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.



Hongrui Chen, Jihao Long, and Lei Wu.

**A duality framework for generalization analysis of random feature models and two-layer neural networks.**

*arXiv preprint arXiv:2305.05642*, 2023.



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.



Ilias Diakonikolas, Daniel M. Kane, Vasilis Kontonis, and Nikos Zarifis.

**Algorithms and SQ lower bounds for PAC learning one-hidden-layer ReLU networks.**

In *Thirty Third Conference on Learning Theory*, volume 125, pages 1514–1539. PMLR, 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.



Fanghui Liu, Xiaolin Huang, Yudong Chen, and Johan AK Suykens.  
**Random features for kernel approximation: A survey on algorithms,  
theory, and beyond.**  
*IEEE Transactions on Pattern Analysis and Machine Intelligence*,  
44(10):7128–7148, 2021.



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.



Ali Rahimi and Benjamin Recht.

**Random features for large-scale kernel machines.**

In *Advances in Neural Information Processing Systems*, pages 1177–1184, 2007.



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.



Zhongjie Shi, Fanghui Liu, Yuan Cao, and Johan AK Suykens.

**Can overfitted deep neural networks in adversarial training generalize?—an approximation viewpoint.**

*arXiv preprint arXiv:2401.13624*, 2024.



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.



Aad W Van Der Vaart, Adrianus Willem van der Vaart, Aad van der Vaart, and Jon Wellner.

***Weak convergence and empirical processes: with applications to statistics.***

Springer Science & Business Media, 1996.



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.



## \*Separation between robust and clean generalization

	#parameters	Upper bound
standard generalization	$\mathcal{O}(nd)$	$\tilde{\mathcal{O}}\left(n^{-\frac{2\alpha}{2\alpha+d}}\right)$
robust generalization	$\mathcal{O}\left(nd + \delta^{-\frac{d}{2\alpha-2}} \log \frac{1}{\delta}\right)$	$\mathcal{O}(\sqrt{d}\delta)$

- more smooth, less #params
- Examples:  $\delta < n^{-\frac{1}{d-1}}$ 
  - $\delta = \frac{1}{n}$  : robust overfitting?

well-separated data + target function is smooth enough + perturbation is small enough

⇒ Avoid robust overfitting!

## \*Separation between robust and clean generalization

	#parameters	Upper bound
standard generalization	$\mathcal{O}(nd)$	$\tilde{\mathcal{O}}\left(n^{-\frac{2\alpha}{2\alpha+d}}\right)$
robust generalization	$\mathcal{O}\left(nd + \delta^{-\frac{d}{2\alpha-2}} \log \frac{1}{\delta}\right)$	$\mathcal{O}(\sqrt{d}\delta)$

- more smooth, less #params
- Examples:  $\delta < n^{-\frac{1}{d-1}}$ 
  - $\delta = \frac{1}{n}$  : robust overfitting?

well-separated data + target function is smooth enough + perturbation is small enough

⇒ **Avoid robust overfitting!**