The Role of Over-parameterization in Machine Learning

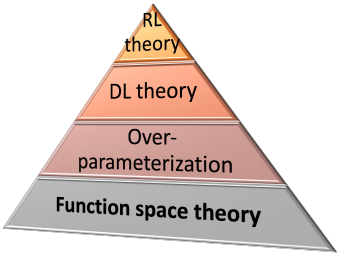
- a function space perspective

Fanghui Liu

Laboratory for Information and Inference Systems (LIONS)
École Polytechnique Fédérale de Lausanne (EPFL)
Switzerland

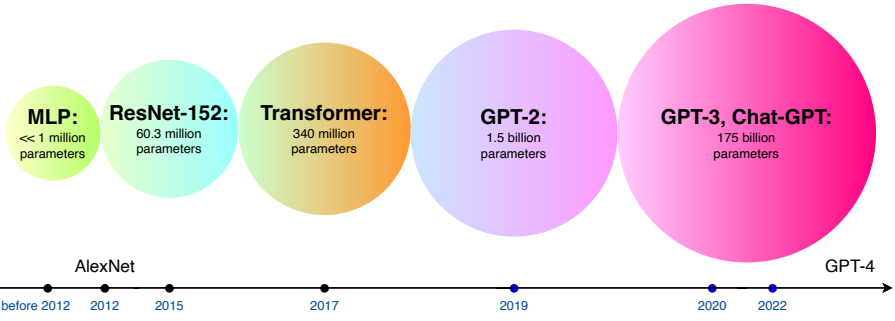
lions@epfl EPFL

Today's Over-parameterization journey



- Motivation
- Robustness in deep learning theory
- Function approximation in reinforcement learning theory

Over-parameterization: more parameters than training data



Over-parameterization: more parameters than training data

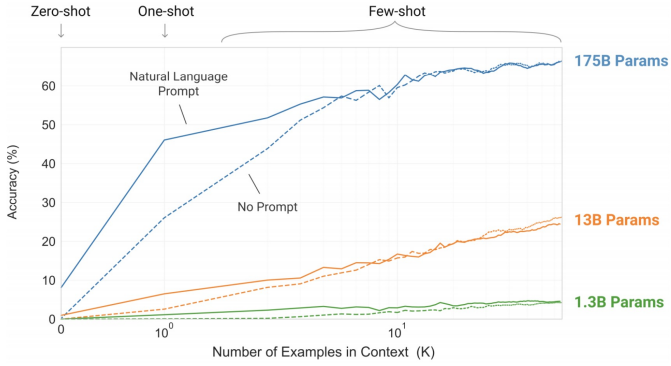


Figure: Larger models make increasingly efficient use of in-context information: source from Open AI.

A grand challenge in deep learning applications: Robustness

Robust, Secure, Trustworthy Machine Learning

(a) Turtle classified as rifle [1]

(b) Stop sign classified as 45 mph sign [2]

A grand challenge in deep learning applications: Robustness

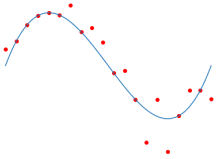
Robust, Secure, Trustworthy Machine Learning

(a) Turtle classified as rifle [1]

(b) Stop sign classified as 45 mph sign [2]

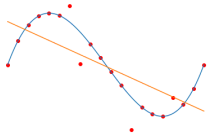
"Understanding deep learning (still) requires rethinking generalization" [3]

A toy example: curve fitting



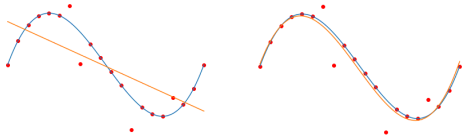
(a) $y_i = f_\rho(\boldsymbol{x}_i) + \epsilon$

A toy example: curve fitting



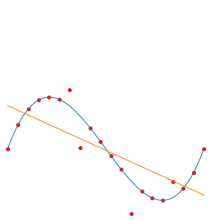
(a) under-fitting

A toy example: curve fitting

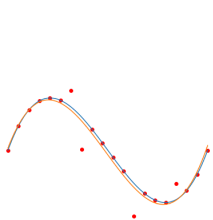


(a) under-fitting (b) sweet spot

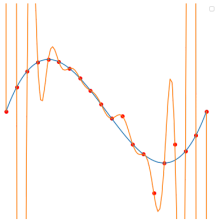
A toy example: curve fitting



(a) under-fitting

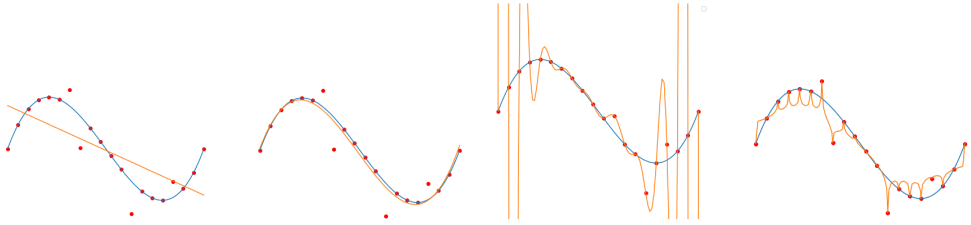


(b) sweet spot



(c) overfitting

A toy example: curve fitting



(a) under-fitting (b) sweet spot (c) overfitting (d) benign overfitting

Figure: Test performance on curve fitting: source from Open AI.

Benign overfitting and double descent

benign overfitting [4, 5, 6]:

- model is very complex
- perfectly fit noisy data and generalize well

Benign overfitting and double descent

benign overfitting [4, 5, 6]:

- model is very complex
- perfectly fit noisy data and generalize well

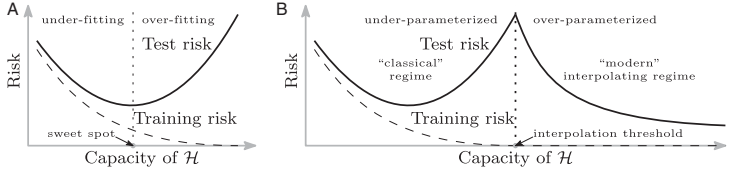
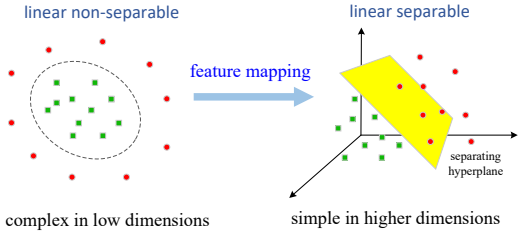
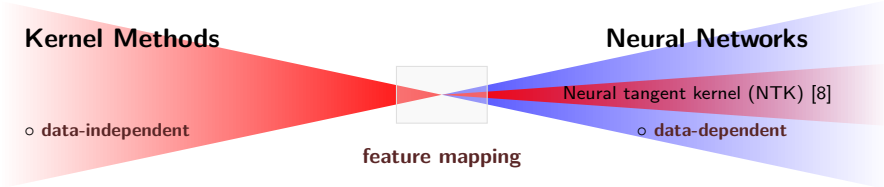


Figure: classical learning theory vs. double descent: source from [7].

Feature mapping: from kernel methods to neural networks



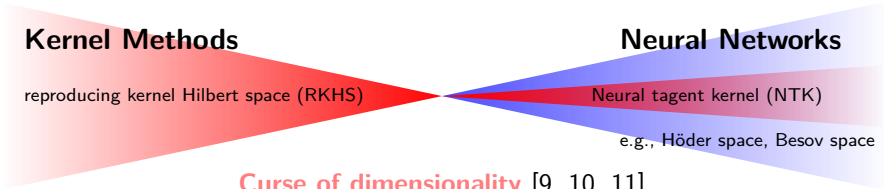
Feature mapping: from kernel methods to neural networks



$$k(\boldsymbol{x}, \boldsymbol{x}') = \langle \phi(\boldsymbol{x}), \phi(\boldsymbol{x}') \rangle_{\mathcal{H}}$$

Function space: from kernel methods to neural networks

efficiently approximate non-smooth functions?



Curse of dimensionality [9, 10, 11]

$$\|f_{\text{Lip}} - f_{\text{kernel}}\|^2 \lesssim \mathcal{O}(n^{-\frac{1}{d}})$$

Why function space theory is needed? (lazy training regime)

$$\mathcal{F}_{\mathrm{NN},m} = \left\{ f_m(\boldsymbol{x};\boldsymbol{\Theta}) = \sum_{i=1}^{m} a_i \max\left(\langle \boldsymbol{w}_i, \boldsymbol{x}\rangle, 0\right) : a_i \in \mathbb{R}, \boldsymbol{w}_i \in \mathbb{R}^d \right\}$$

- Gaussian initialization: $\boldsymbol{w}_i, a_i \sim \mathcal{N}(0, \mathtt{var})$

Why function space theory is needed? (lazy training regime)

$$\mathcal{F}_{\mathrm{NN},m} = \left\{ f_m(\boldsymbol{x};\boldsymbol{\Theta}) = \sum_{i=1}^{m} a_i \max\left(\langle \boldsymbol{w}_i, \boldsymbol{x}\rangle, 0\right) : a_i \in \mathbb{R}, \boldsymbol{w}_i \in \mathbb{R}^d \right\}$$

- Gaussian initialization: $\boldsymbol{w}_i, a_i \sim \mathcal{N}(0, \mathtt{var})$

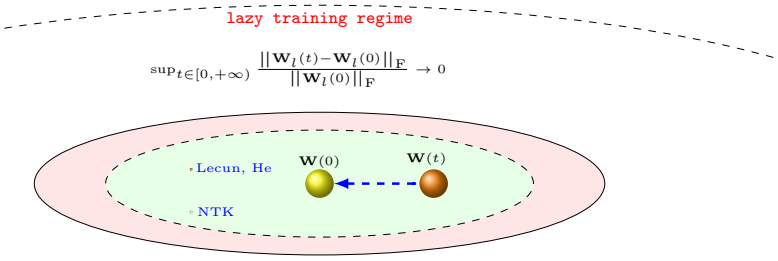


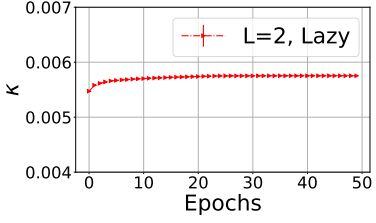
Figure: Training dynamics of two-layer ReLU NNs under different initializations [8, 12, 13].

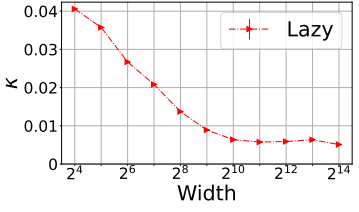
Why function space theory is needed? (lazy training regime)

$$\mathcal{F}_{\mathrm{NN},m} = \left\{ f_m(\boldsymbol{x};\boldsymbol{\Theta}) = \sum_{i=1}^{m} a_i \max\left(\langle \boldsymbol{w}_i, \boldsymbol{x}\rangle, 0\right) : a_i \in \mathbb{R}, \boldsymbol{w}_i \in \mathbb{R}^d \right\}$$

- Gaussian initialization: $\boldsymbol{w}_i, a_i \sim \mathcal{N}(0, \mathtt{var})$

$$\text{lazy training ratio } \kappa := \frac{\sum_{l=1}^{L} \|\boldsymbol{W}_l(t) - \boldsymbol{W}_l(0)\|_{\mathrm{F}}}{\sum_{l=1}^{L} \|\boldsymbol{W}_l(0)\|_{\mathrm{F}}}$$





Why function space theory is needed? (non-lazy training regime)

$$\mathcal{F}_{\mathrm{NN},m} = \left\{ f_m(\boldsymbol{x};\boldsymbol{\Theta}) = \sum_{i=1}^{m} a_i \max\left(\langle \boldsymbol{w}_i, \boldsymbol{x}\rangle, 0\right) : \frac{1}{m}\sum_{i=1}^{m} |a_i| \|\boldsymbol{w}_i\|_1 < \infty \right\} \quad \text{(Barron space)}$$

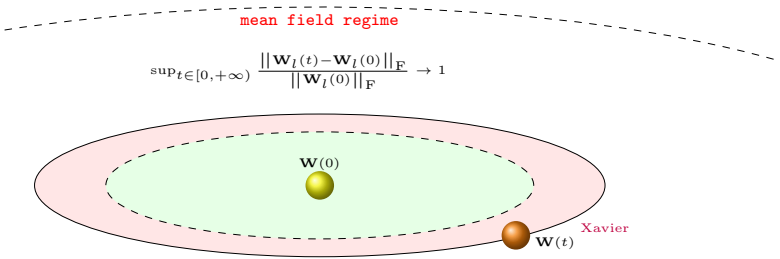


Figure: Training dynamics of two-layer ReLU NNs under different initializations [8, 12, 13].

Why function space theory is needed? (non-lazy training regime)

$$\mathcal{F}_{\mathrm{NN},m} = \left\{ f_m(\boldsymbol{x};\boldsymbol{\Theta}) = \sum_{i=1}^{m} a_i \max\left(\langle \boldsymbol{w}_i, \boldsymbol{x}\rangle, 0\right) : \frac{1}{m}\sum_{i=1}^{m} |a_i| \|\boldsymbol{w}_i\|_1 < \infty \right\} \quad \text{(Barron space)}$$

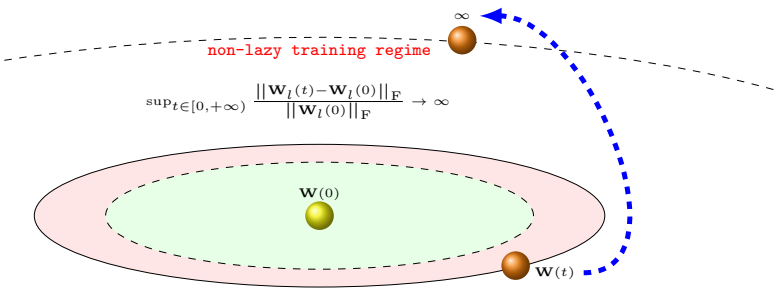
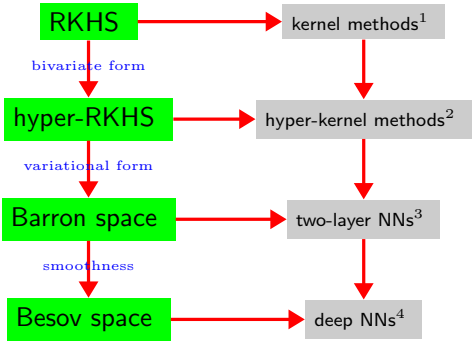
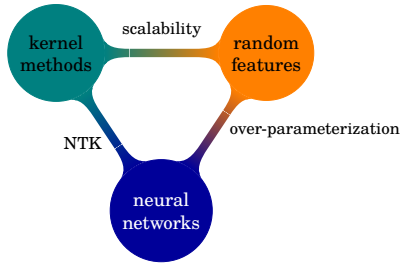


Figure: Training dynamics of two-layer ReLU NNs under different initializations [8, 12, 13].

Research interests

Understanding from a function space perspective!





[1] [LHGYL, JMLR20; LHCS, TPAMI21; LLS, AISTATS21]

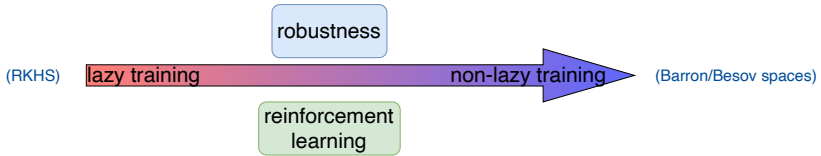
[2] [LSHYS, JMLR21]

[3] [LSC, NeurIPS22; LHCS, TPAMI22; LHCS, AISTATS21]

[4] [LVC, NeurIPS22; ZLCC, NeurIPS22; WZLCC, NeurIPS22]

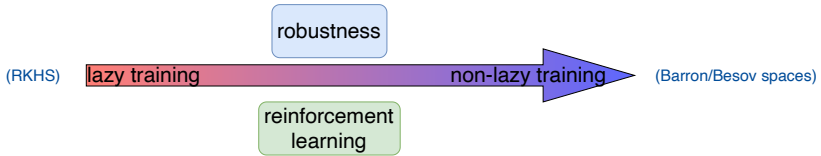
Research Overview: Today's talk

What is the role of over-parameterization in DNNs from the function space perspective?



Research Overview: Today's talk

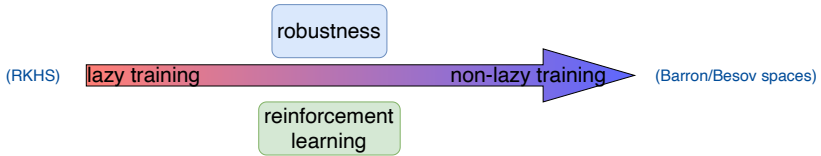
What is the role of over-parameterization in DNNs from the function space perspective?



- **robustness** of NNs
  - [ZLCC, NeurIPS22] **Over-parameterization** helps or hurts robustness?

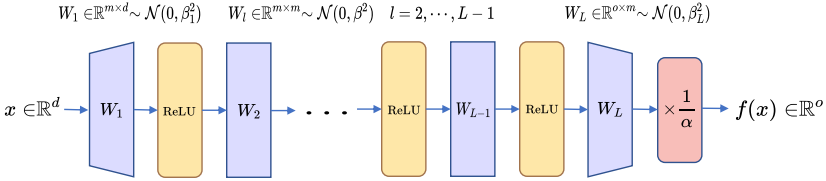
Research Overview: Today's talk

What is the role of over-parameterization in DNNs from the function space perspective?



- **robustness** of NNs
  - [ZLCC, NeurIPS22] **Over-parameterization** helps or hurts robustness?
- **reinforcement learning** via value/Q function approximation
  - [LVC, NeurIPS22] Deep RL beyond the **lazy-training** regime

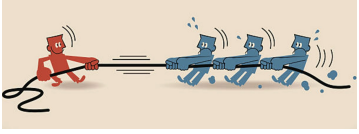
Architecture of DNNs



Initialization	Formulation
LeCun initialization	$\beta_1 = \sqrt{\frac{1}{d}}$, $\beta = \beta_L = \sqrt{\frac{1}{m}}$
He initialization	$\beta_1 = \sqrt{\frac{2}{d}}$, $\beta = \beta_L = \sqrt{\frac{2}{m}}$
NTK initialization	$\beta = \beta_1 = \sqrt{\frac{2}{m}}$, $\beta_L = 1$

Over-parameterization helps or hurts robustness?[1]

Helps! [14]

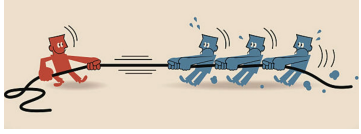


Hurts! [15, 16, 17]

[1] Zhenyu Zhu, **Fanghui Liu**, Grigorios Chrysos, Volkan Cevher, *Robustness in deep learning: The good (width), the bad (depth), and the ugly (initialization)*. NeurIPS 2022.

Over-parameterization helps or hurts robustness?[1]

Helps! [14]



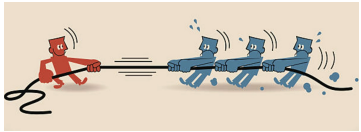
Hurts! [15, 16, 17]

- initialization (e.g., lazy training, non-lazy training)
- architecture (e.g., width, depth)

[1] Zhenyu Zhu, **Fanghui Liu**, Grigorios Chrysos, Volkan Cevher, *Robustness in deep learning: The good (width), the bad (depth), and the ugly (initialization)*. NeurIPS 2022.

Over-parameterization helps or hurts robustness?[1]

Helps! [14]



Hurts! [15, 16, 17]

- initialization (e.g., lazy training, non-lazy training)
- architecture (e.g., width, depth)

Definition (perturbation stability)

The perturbation stability of a ReLU DNN $f(\boldsymbol{x}; \boldsymbol{W})$ is

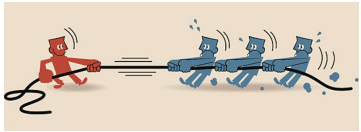
$$\mathscr{P}(f, \epsilon) = \mathbb{E}_{\boldsymbol{x}, \hat{\boldsymbol{x}}, \boldsymbol{W}} \left\| \nabla_{\boldsymbol{x}} f(\boldsymbol{x}; \boldsymbol{W})^\top (\boldsymbol{x} - \hat{\boldsymbol{x}}) \right\|_2, \quad \hat{\boldsymbol{x}} \sim \mathsf{Unif}(\mathbb{B}(\epsilon, \boldsymbol{x})),$$

where ϵ is the perturbation radius.

[1] Zhenyu Zhu, **Fanghui Liu**, Grigorios Chrysos, Volkan Cevher, *Robustness in deep learning: The good (width), the bad (depth), and the ugly (initialization)*. NeurIPS 2022.

Over-parameterization helps or hurts robustness?[1]

Helps! [14]



Hurts! [15, 16, 17]

- initialization (e.g., lazy training, non-lazy training)
- architecture (e.g., width, depth)

Definition (perturbation stability: lazy training regime)

The perturbation stability of a ReLU DNN $f(\boldsymbol{x}; \boldsymbol{W})$ is

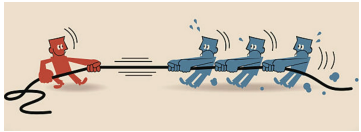
$$\mathscr{P}(f, \epsilon) = \mathbb{E}_{\boldsymbol{x}, \hat{\boldsymbol{x}}, \boldsymbol{W}(0)} \left\| \nabla_{\boldsymbol{x}} f(\boldsymbol{x}; \boldsymbol{W})^{\top} (\boldsymbol{x} - \hat{\boldsymbol{x}}) \right\|_2, \quad \hat{\boldsymbol{x}} \sim \mathsf{Unif}(\mathbb{B}(\epsilon, \boldsymbol{x})),$$

where ϵ is the perturbation radius.

[1] Zhenyu Zhu, **Fanghui Liu**, Grigorios Chrysos, Volkan Cevher, *Robustness in deep learning: The good (width), the bad (depth), and the ugly (initialization)*. NeurIPS 2022.

Over-parameterization helps or hurts robustness?[1]

Helps! [14]



Hurts! [15, 16, 17]

- initialization (e.g., lazy training, non-lazy training)
- architecture (e.g., width, depth)

Definition (perturbation stability: non-lazy training regime)

The perturbation stability of a ReLU DNN $f(\boldsymbol{x}; \boldsymbol{W})$ is

$$\mathscr{P}(f, \epsilon) = \mathbb{E}_{\boldsymbol{x}, \hat{\boldsymbol{x}}} \left\| \nabla_{\boldsymbol{x}} f(\boldsymbol{x}; \boldsymbol{W})^{\top} (\boldsymbol{x} - \hat{\boldsymbol{x}}) \right\|_2, \quad \hat{\boldsymbol{x}} \sim \mathsf{Unif}(\mathbb{B}(\epsilon, \boldsymbol{x})),$$

where ϵ is the perturbation radius.

[1] Zhenyu Zhu, **Fanghui Liu**, Grigorios Chrysos, Volkan Cevher, *Robustness in deep learning: The good (width), the bad (depth), and the ugly (initialization)*. NeurIPS 2022.

Main results (Lazy-training regime)

Theorem: perturbation stability $\lesssim \mathrm{Func}(m, L, \beta)$

Assumption	Initialization	Our bound for $\mathscr{P}(f,\epsilon)/\epsilon$	Trend of width m [1]	Trend of depth L [1]
$\|\boldsymbol{x}\|_2 = 1$	LeCun initialization	$\left(\sqrt{\frac{L^3 m}{d}} e^{-m/L^3} + \sqrt{\frac{1}{d}}\right)(\frac{\sqrt{2}}{2})^{L-2}$	↗ ↘	↘
	He initialization	$\sqrt{\frac{L^3 m}{d}} e^{-m/L^3} + \sqrt{\frac{1}{d}}$	↗ ↘	↗
	NTK initialization	$\sqrt{\frac{L^3 m}{d}} e^{-m/L^3} + 1$	↗ ↘	↗

[1] The larger perturbation stability means worse average robustness.

Takeaway messages: **the good (width), the bad (depth), the ugly (initialization)**

Main results (Lazy-training regime)

Theorem: perturbation stability $\lesssim \mathrm{Func}(m, L, \beta)$

Assumption	Initialization	Our bound for $\mathscr{P}(f,\epsilon)/\epsilon$	Trend of width m [1]	Trend of depth L [1]
$\|\boldsymbol{x}\|_2 = 1$	LeCun initialization	$\left(\sqrt{\frac{L^3 m}{d}} e^{-m/L^3} + \sqrt{\frac{1}{d}}\right)(\frac{\sqrt{2}}{2})^{L-2}$	↗ ↘	↘
	He initialization	$\sqrt{\frac{L^3 m}{d}} e^{-m/L^3} + \sqrt{\frac{1}{d}}$	↗ ↘	↗
	NTK initialization	$\sqrt{\frac{L^3 m}{d}} e^{-m/L^3} + 1$	↗ ↘	↗

[1] The larger perturbation stability means worse average robustness.

Takeaway messages: **the good (width), the bad (depth), the ugly (initialization)**

- width **helps** robustness in the over-parameterized regime

Main results (Lazy-training regime)

Theorem: perturbation stability $\lesssim \mathrm{Func}(m, L, \beta)$

Assumption	Initialization	Our bound for $\mathscr{P}(f,\epsilon)/\epsilon$	Trend of width m [1]	Trend of depth L [1]
$\|\boldsymbol{x}\|_2 = 1$	LeCun initialization	$\left(\sqrt{\frac{L^3 m}{d}} e^{-m/L^3} + \sqrt{\frac{1}{d}}\right)(\frac{\sqrt{2}}{2})^{L-2}$	↗ ↘	↘
	He initialization	$\sqrt{\frac{L^3 m}{d}} e^{-m/L^3} + \sqrt{\frac{1}{d}}$	↗ ↘	↗
	NTK initialization	$\sqrt{\frac{L^3 m}{d}} e^{-m/L^3} + 1$	↗ ↘	↗

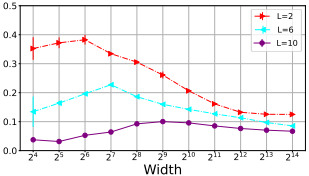
[1] The larger perturbation stability means worse average robustness.

Takeaway messages: **the good (width), the bad (depth), the ugly (initialization)**

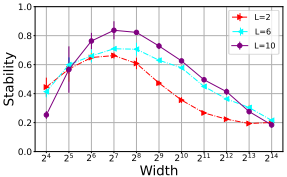
- width **helps** robustness in the over-parameterized regime
- depth **helps** robustness in LeCun initialization but **hurts** robustness in He/NTK initialization

Experiments: robustness under lazy-training regime

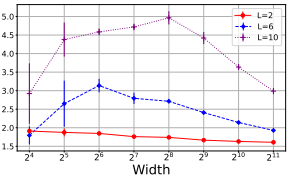
Metrics	Ours (NTK initialization)	[16]	[17]
$\mathscr{P}(f,\epsilon)/\epsilon$	$\sqrt{\frac{L^3 m}{d}} e^{-m/L^3} + 1$	$L^2 m^{1/3} \sqrt{\log m} + \sqrt{mL}$	$2^{\frac{3L-5}{2}} \sqrt{L}$



(a) LeCun initialization

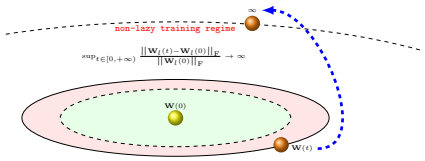


(b) He initialization



(c) NTK initialization

Main results (Non-lazy training regime)



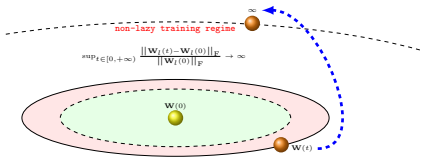
sufficient condition for DNNs

for large enough m and $m \gg d$, w.h.p, DNNs fall into non-lazy training regime if

$$\alpha \gg (m^{3/2} \sum_{i=1}^{L} \beta_i)^L .$$

e.g., $L = 2$, $\alpha = 1$, $\beta_1 = \beta_2 = \beta \sim \frac{1}{m^c}$ with $c > 1.5$

Main results (Non-lazy training regime)



sufficient condition for DNNs

for large enough m and $m \gg d$, w.h.p, DNNs fall into non-lazy training regime if

$$\alpha \gg (m^{3/2} \sum_{i=1}^{L} \beta_i)^L .$$

e.g., $L = 2$, $\alpha = 1$, $\beta_1 = \beta_2 = \beta \sim \frac{1}{m^c}$ with $c > 1.5$

Theorem (non-lazy training regime for two-layer NNs)

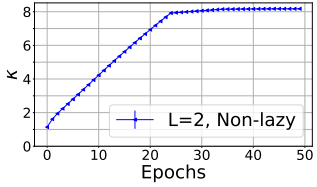
Under this setting with $m \gg n^2$ and standard assumptions, then

$$\frac{\mathscr{P}(\boldsymbol{f}_t, \epsilon)}{\epsilon} \leq \widetilde{\mathcal{O}}\Big(\frac{n}{m^{c+1.5}}\Big), \ w.h.p$$

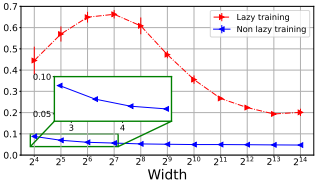
- width helps robustness in the over-parameterized regime in both lazy/non-lazy training regime

Experiment: Non-lazy training regime

$$\text{lazy training ratio } \kappa := \frac{\sum_{l=1}^{L} \|\boldsymbol{W}_l(t) - \boldsymbol{W}_l(0)\|_{\mathrm{F}}}{\sum_{l=1}^{L} \|\boldsymbol{W}_l(0)\|_{\mathrm{F}}}$$



(a) lazy training ratio *vs.* epochs



(b) perturbation stability

Future direction

- robustness improvement
- flateness, robustness, and generalization

Current and future directions in generalization guarantees

Figure: Double descent [7].

Regime of interest

n, d, m are comparably large

- high dimensional setting [18, 19]: $c \leqslant \{d/n, m/n\} \leqslant C$

Current and future directions in generalization guarantees

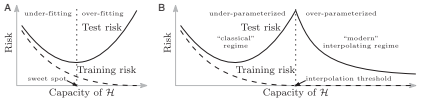


Figure: Double descent [7].

Regime of interest

n, d, m are comparably large

- high dimensional setting [18, 19]: $c \leqslant \{d/n, m/n\} \leqslant C$

ridge regression: $\hat{\boldsymbol{\beta}} = (\boldsymbol{X}^\top \boldsymbol{X} + \lambda \boldsymbol{I})^{-1} \boldsymbol{X}^\top \boldsymbol{y} \xrightarrow{?} \boldsymbol{\beta}^\star := \arg\min_{\boldsymbol{\beta}} \mathbb{E}[(\boldsymbol{y} - \boldsymbol{\beta}^\top \boldsymbol{x})^2]$

Current and future directions in generalization guarantees

Figure: Double descent [7].

Regime of interest

n, d, m are comparably large

- high dimensional setting [18, 19]: $c \leqslant \{d/n, m/n\} \leqslant C$

ridge regression: $\hat{\boldsymbol{\beta}} = (\boldsymbol{X}^\top \boldsymbol{X} + \lambda \boldsymbol{I})^{-1} \boldsymbol{X}^\top \boldsymbol{y} \xrightarrow{?} \boldsymbol{\beta}^\star := \arg\min_{\boldsymbol{\beta}} \mathbb{E}[(\boldsymbol{y} - \boldsymbol{\beta}^\top \boldsymbol{x})^2]$

Consistency of estimator?

- classical low-dimensional setting (d fixed and $n \to \infty$): ✓
- classical high-dimensional setting ($d \gg n$) if sparsity [20]: ✓

Current and future directions in generalization guarantees

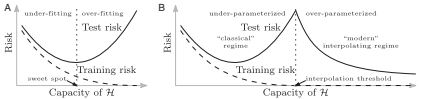


Figure: Double descent [7].

Regime of interest

n, d, m are comparably large

- high dimensional setting [18, 19]: $c \leqslant \{d/n, m/n\} \leqslant C$

ridge regression: $\hat{\boldsymbol{\beta}} = (\boldsymbol{X}^\top \boldsymbol{X} + \lambda \boldsymbol{I})^{-1} \boldsymbol{X}^\top \boldsymbol{y} \xrightarrow{?} \boldsymbol{\beta}^\star := \arg\min_{\boldsymbol{\beta}} \mathbb{E}[(\boldsymbol{y} - \boldsymbol{\beta}^\top \boldsymbol{x})^2]$

Consistency of estimator?

- classical low-dimensional setting (d fixed and $n \to \infty$): ✓
- classical high-dimensional setting ($d \gg n$) if sparsity [20]: ✓
- n, d, m are comparably large: ✗

Current and future directions in generalization guarantees

Figure: Double descent [7].

Regime of interest

n, d, m are comparably large

- high dimensional setting [18, 19]: $c \leqslant \{d/n, m/n\} \leqslant C$

ridge regression: $\hat{\boldsymbol{\beta}} = (\boldsymbol{X}^\top \boldsymbol{X} + \lambda \boldsymbol{I})^{-1} \boldsymbol{X}^\top \boldsymbol{y} \xrightarrow{?} \boldsymbol{\beta}^\star := \arg\min_{\boldsymbol{\beta}} \mathbb{E}[(\boldsymbol{y} - \boldsymbol{\beta}^\top \boldsymbol{x})^2]$

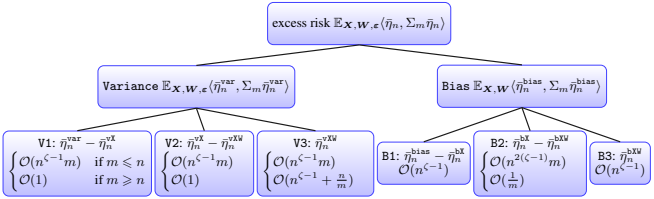
Consistency of estimator?

- classical low-dimensional setting (d fixed and $n \to \infty$): ✓
- classical high-dimensional setting ($d \gg n$) if sparsity [20]: ✓
- n, d, m are comparably large: ✗

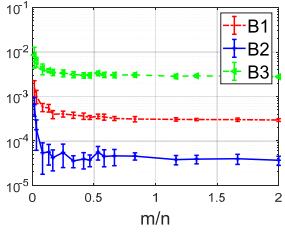
for what sample size, and what data distributions, the estimator can generalize well?

Results: Why over-parameterized model can generalize well under SGD?

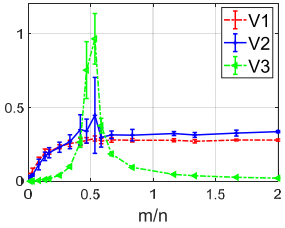
Theorem [LSC, NeurIPS22]: Under sub-Gaussian data, label noise with bounded variance, $f_\rho \in \mathcal{H}$, we have



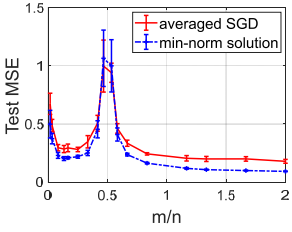
- (partially) decouple multiple randomness sources
- converge to $\mathcal{O}(1)$ order (noise variance)
- the minimax rate: constant step-size SGD vs. minimum-norm solution



(a) bias



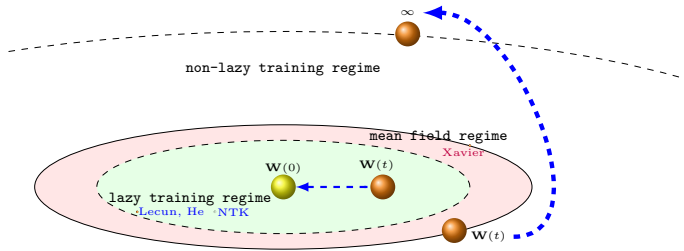
(b) variance



(c) excess risk

Future direction in generalization guarantees

- benign overfitting/double descent in deep neural networks [6]



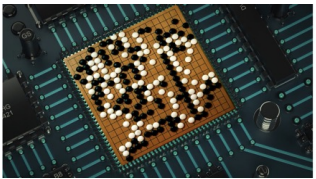
- high dimensional kernel methods: only learn linear function!
  - function space for neural networks, e.g., Barron space [21, 22]
  - feature learning beyond RKHS: hyper-RKHS[2], mean field regime [23, 24]
- non-i.i.d. setting [25, 26]

[2] **Fanghui Liu**, **Lei Shi**, Xiaolin Huang, Jie Yang, Johan Suykens. Generalization properties of hyper-RKHS and its applications. JMLR 2021.

Background: Deep reinforcement learning and function approximation

Deep RL, e.g., deep Q network [27]

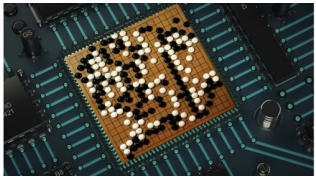
Representation + Decision making + **value/Q function approximation** = **efficient** Human-level AI



Background: Deep reinforcement learning and function approximation

Deep RL, e.g., deep Q network [27]

Representation + Decision making + **value/Q function approximation** = **efficient** Human-level AI



Why function approximation?

- RL in large state spaces: Go has more than 10^{172} states

Background: Episodic Markov decision process in the online setting

Episodic Markov decision process: MDP$(\mathcal{S}, \mathcal{A}, H, \mathbb{P}, r)$

- $\mathcal{S}$: infinite state space; $\mathcal{A}$: finite action space; $r : \mathcal{S} \times \mathcal{A} \to [0, 1]$: unknown reward function
- $\mathbb{P}_h(\cdot|s_h, a_h)$: unknown transition kernel; H: finite horizon - terminate when $h = H$

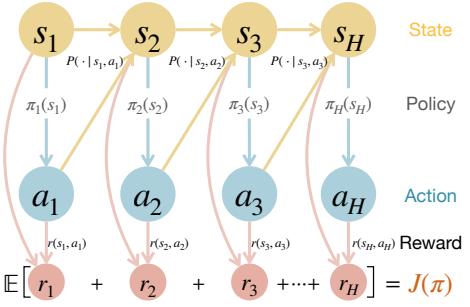
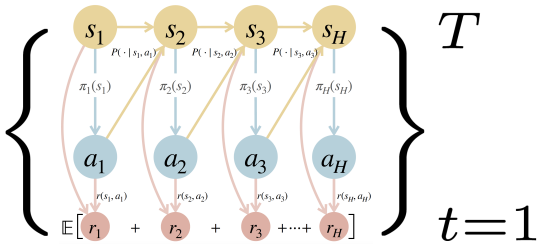


Figure: Episodic MDP [28], where $\pi := \{\pi_h\}_{h=1}^H$.

Background: Episodic Markov decision process in the online setting

Episodic Markov decision process: MDP$(\mathcal{S}, \mathcal{A}, H, \mathbb{P}, r)$

- $\mathcal{S}$: infinite state space; $\mathcal{A}$: finite action space; $r : \mathcal{S} \times \mathcal{A} \to [0, 1]$: unknown reward function
- $\mathbb{P}_h(\cdot|s_h, a_h)$: unknown transition kernel; H: finite horizon - terminate when $h = H$



$$\texttt{Regret}(T) := \sum_{t=1}^{T} \left[J(\pi^\star) - J(\pi^t) \right]$$

Understanding deep RL beyond the lazy training regime[3]

Discrepancy between deep RL and previous theoretical work

- Neural Tangent Kernel (NTK): "linear" regime
- Eluder dimension: exponential order for two-layer neural networks [29, 30]

[3] **Fanghui Liu**, Luca Viano, Volkan Cevher, *Understanding deep neural function approximation in reinforcement learning via ε-greedy exploration*, NeurIPS 2022.

Understanding deep RL beyond the lazy training regime[3]

Discrepancy between deep RL and previous theoretical work

- Neural Tangent Kernel (NTK): "linear" regime
- Eluder dimension: exponential order for two-layer neural networks [29, 30]

Understanding deep RL from function space theory!

[3] **Fanghui Liu**, Luca Viano, Volkan Cevher, *Understanding deep neural function approximation in reinforcement learning via ε-greedy exploration*, NeurIPS 2022.

Understanding deep RL beyond the lazy training regime[3]

Discrepancy between deep RL and previous theoretical work

- Neural Tangent Kernel (NTK): "linear" regime
- Eluder dimension: exponential order for two-layer neural networks [29, 30]

Understanding deep RL from function space theory!

beyond RKHS with non-smooth functions

- Barron spaces for two-layer NNs
- Besov spaces for DNNs

[3] **Fanghui Liu**, Luca Viano, Volkan Cevher, *Understanding deep neural function approximation in reinforcement learning via ε-greedy exploration*, NeurIPS 2022.

Algorithm

Algorithm 1: Value iteration via DNNs under ϵ-greedy exploration with experience replay

Input: Function class of DNNs $\mathcal{F}$, mini-batch ratio $\varrho \in (0,1)$.

Initialize replay memory $\mathcal{D}$;
for *episode* $t = 1, \ldots, T$ **do**
  Receive the initial state s_1^t and $V_{H+1}^t = 0$; Set the minibatch size $\tilde{t} := \lceil \varrho t \rceil$ for experience replay;
  for *step* $h = H, \ldots, 1$ **do**
    Obtain $\widehat{Q}_h^t := \arg\min_{f \in \mathcal{F}}$ least squares value iteration via DNNs $\sum_{j=1}^{\tilde{t}} \left[f(s_h^{\tau_j}, a_h^{\tau_j}) - r_h(s_h^{\tau_j}, a_h^{\tau_j}) - V_{h+1}^t(s_{h+1}^{\tau_j}) \right]^2$;
    Obtain $Q_h^t := \widehat{Q}_h^t$ and $V_h^t(\cdot) = \max_{a \in \mathcal{A}} Q_h^t(\cdot, a)$;
  end
  ϵ-greedy for exploration: take $\{\tilde{\pi}_h^t\}_{h=1}^H$ to be greedy w.p $1-\epsilon$ or any policy w.p ϵ;
  for *step* $h = 1, \ldots, H$ **do**
    Take $a_h^t \sim \tilde{\pi}_h^t(\cdot | s_h^t)$; Observe the reward $r_h(s_h^t, a_h^t)$ and obtain the next state s_{h+1}^t;
  end
  experience replay:
  Store transition $\{(s_h^t, a_h^t, r_h, s_{h+1}^t)\}_{h=1}^H$ in $\mathcal{D}$;
  Sample random mini-batch of transitions from $\mathcal{D}$ with $\tilde{t}$ pairs $\{(s_h^{\tau_j}, a_h^{\tau_j}, s_{h+1}^{\tau_j})\}_{(j,h) \in [\tilde{t}] \times [H]}$;
end

Results: sublinear regret under DNNs in Besov spaces

Theorem

Assume $\mathbb{T}_h^\star Q \in$ `Besov space`, *if the target Q function is* α*-smooth*,

$$\textit{choosing the depth } L = \mathcal{O}(\log T) \textit{ and the width } m = \widetilde{\mathcal{O}}\left(T^{\frac{d}{2\alpha+d}}\right), \textit{then}$$

$$\texttt{Regret}(T) \lesssim o(T) \quad \textit{w.h.p.}$$

Results: sublinear regret under DNNs in Besov spaces

Theorem

Assume $\mathbb{T}_h^\star Q \in$ Besov space, *if the target Q function is* α*-smooth,*

$$\textit{choosing the depth } L = \mathcal{O}(\log T) \textit{ and the width } m = \widetilde{\mathcal{O}}\left(T^{\frac{d}{2\alpha+d}}\right), \textit{then}$$

$$\texttt{Regret}(T) \lesssim o(T) \quad \textit{w.h.p.}$$

new proof framework:

$$\texttt{Regret}(T) \lesssim \underbrace{\text{generalization error}}_{\text{over non-iid state-action pairs}} + \text{approximation error} + \widetilde{\mathcal{O}}(\sqrt{H^3 T}) + \epsilon H T\,.$$

Guidelines for deep RL

more difficult task in deep RL:

- smoothness: $\searrow$
- exploration: $\nwarrow$
  slower rate on regret

Guidelines for deep RL

more difficult task in deep RL:

- smoothness: $\searrow$
- exploration: $\nwarrow$
  slower rate on regret

Width-depth guidelines:

- sublinear regret: $\mathcal{O}(\log T)$ depth and $m = \widetilde{\mathcal{O}}\left(T^{\frac{d}{2\alpha+d}}\right)$ width

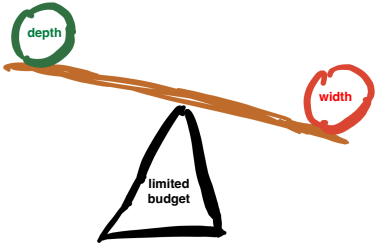
Guidelines for deep RL

more difficult task in deep RL:

- smoothness: $\searrow$
- exploration: $\nwarrow$
  slower rate on regret

Width-depth guidelines:

- sublinear regret: $\mathcal{O}(\log T)$ depth and $m = \widetilde{\mathcal{O}}\left(T^{\frac{d}{2\alpha+d}}\right)$ width



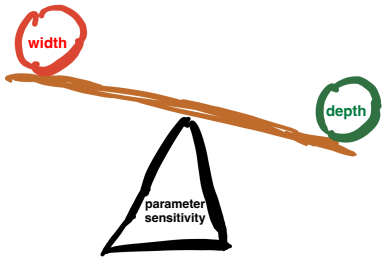
Guidelines for deep RL

more difficult task in deep RL:

- smoothness: $\searrow$
- exploration: $\nwarrow$
  slower rate on regret

Width-depth guidelines:

- sublinear regret: $\mathcal{O}(\log T)$ depth and $m = \widetilde{\mathcal{O}}\left(T^{\frac{d}{2\alpha+d}}\right)$ width



Future direction: Offline data/RL for online RL

- data coverage condition [31, 32]

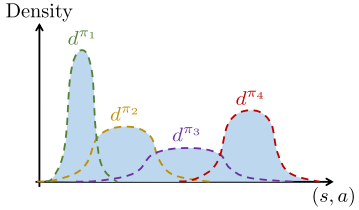
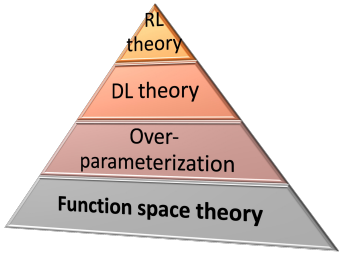


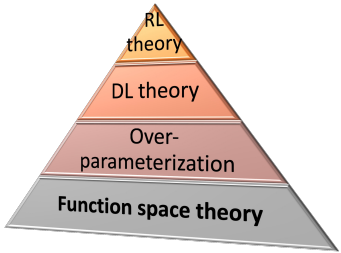
Figure: Good data coverage implies sample-efficient online RL [33].

- relationship between structural assumption and coverage condition?
- Partial coverage + offline data = sample-efficient online RL?

Today's Over-parameterization journey



Today's Over-parameterization journey



Take away messages: trustworthy decision making system

- initialization, function spaces
- the good (width), the bad (depth), the ugly (initialization)
- sample-efficient deep RL beyond lazy training

Future direction in application

- Trustworthy decision making system: robustness, safe RL, privacy, federated learning
- Neural architecture search (ZLCC, NeurIPS22)

Future direction in application

- Trustworthy decision making system: robustness, safe RL, privacy, federated learning
- Neural architecture search (ZLCC, NeurIPS22)
- large scale kernel approximation (LHCS, TPAMI21,22)

$$\text{Attention}(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V}) = \underbrace{\text{softmax}(\boldsymbol{Q}\boldsymbol{K}^\top)}_{:=\boldsymbol{A}}\boldsymbol{V} \approx \boldsymbol{Q}'\boldsymbol{K}'^\top\boldsymbol{V}\,,$$

where $A_{ij} = k(\boldsymbol{q}_i,\boldsymbol{k}_j) = \mathbb{E}[\sigma(\boldsymbol{q}_i)^\top\sigma(\boldsymbol{k}_j)]$

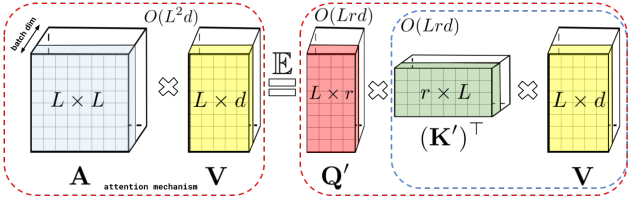


Figure: Approximation of self-attention. source: [34].

Future direction in application

- Trustworthy decision making system: robustness, safe RL, privacy, federated learning
- Neural architecture search (ZLCC, NeurIPS22)
- large scale kernel approximation (LHCS, TPAMI21,22)

$$\mathrm{Attention}(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V}) = \underbrace{\mathrm{softmax}(\boldsymbol{Q}\boldsymbol{K}^\top)}_{:=\boldsymbol{A}}\boldsymbol{V} \approx \boldsymbol{Q}'\boldsymbol{K}'^\top\boldsymbol{V}\,,$$

where $A_{ij} = k(\boldsymbol{q}_i,\boldsymbol{k}_j) = \mathbb{E}[\sigma(\boldsymbol{q}_i)^\top\sigma(\boldsymbol{k}_j)]$

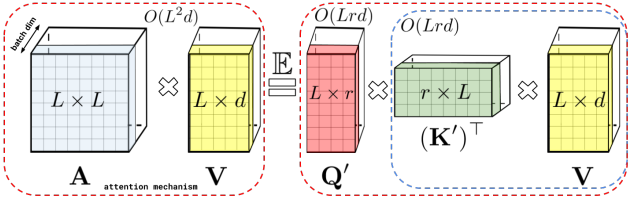


Figure: Approximation of self-attention. source: [34].

Tutorials

- ICASSP 2023 - Neural networks: the good, the bad, and the ugly
- CVPR 2023 - Deep learning theory for computer vision

Thanks for your attention!

Q & A

my homepage www.lfhsgre.org for more information!

References I

[1] Anish Athalye, Logan Engstrom, Andrew Ilyas, and Kevin Kwok.
Synthesizing robust adversarial examples.
In *International Conference on Machine Learning*, pages 284–293. PMLR, 2018.
(Cited on pages 5 and 6.)

[2] Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Chaowei Xiao, Atul Prakash, Tadayoshi Kohno, and Dawn Song.
Robust physical-world attacks on deep learning visual classification.
In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1625–1634, 2018.
(Cited on pages 5 and 6.)

[3] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals.
Understanding deep learning (still) requires rethinking generalization.
Communications of the ACM, 64(3):107–115, 2021.
(Cited on pages 5 and 6.)

[4] Peter L. Bartlett, Philip M. Long, Gábor Lugosi, and Alexander Tsigler.
Benign overfitting in linear regression.
the National Academy of Sciences, 2020.
(Cited on pages 12 and 13.)

References II

[5] Niladri S Chatterji and Philip M Long.
Foolish crowds support benign overfitting.
Journal of Machine Learning Research, 23(125):1–12, 2022.
(Cited on pages 12 and 13.)

[6] Spencer Frei, Niladri S Chatterji, and Peter Bartlett.
Benign overfitting without linearity: Neural network classifiers trained by gradient descent for noisy linear data.
In *Conference on Learning Theory*, pages 2668–2703. PMLR, 2022.
(Cited on pages 12, 13, and 45.)

[7] 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.
(Cited on pages 12, 13, 39, 40, 41, 42, and 43.)

[8] Arthur Jacot, Franck Gabriel, and Clément Hongler.
Neural tangent kernel: Convergence and generalization in neural networks.
In *Advances in Neural Information Processing Systems*, pages 8571–8580, 2018.
(Cited on pages 15, 17, 18, 20, and 21.)

References III

[9] Francis Bach.
Breaking the curse of dimensionality with convex neural networks.
Journal of Machine Learning Research, 18(1):629–681, 2017.
(Cited on page 16.)

[10] Gilad Yehudai and Ohad Shamir.
On the power and limitations of random features for understanding neural networks.
In *Advances in Neural Information Processing Systems*, pages 6594–6604, 2019.
(Cited on page 16.)

[11] Michael Celentano, Theodor Misiakiewicz, and Andrea Montanari.
Minimum complexity interpolation in random features models.
arXiv preprint arXiv:2103.15996, 2021.
(Cited on page 16.)

[12] Lenaic Chizat, Edouard Oyallon, and Francis Bach.
On lazy training in differentiable programming.
In *Advances in Neural Information Processing Systems*, pages 2933–2943, 2019.
(Cited on pages 17, 18, 20, and 21.)

References IV

[13] Tao Luo, Zhi-Qin John Xu, Zheng Ma, and Yaoyu Zhang.
Phase diagram for two-layer relu neural networks at infinite-width limit.
Journal of Machine Learning Research, 22(71):1–47, 2021.
(Cited on pages 17, 18, 20, and 21.)

[14] Sébastien Bubeck and Mark Sellke.
A universal law of robustness via isoperimetry.
In *Advances in Neural Information Processing Systems*, pages 28811–28822, 2021.
(Cited on pages 27, 28, 29, 30, and 31.)

[15] Hamed Hassani and Adel Javanmard.
The curse of overparametrization in adversarial training: Precise analysis of robust generalization for random features regression.
arXiv preprint arXiv:2201.05149, 2022.
(Cited on pages 27, 28, 29, 30, and 31.)

[16] Boxi Wu, Jinghui Chen, Deng Cai, Xiaofei He, and Quanquan Gu.
Do wider neural networks really help adversarial robustness?
In *Advances in Neural Information Processing Systems*, pages 7054–7067, 2021.
(Cited on pages 27, 28, 29, 30, 31, and 35.)

References V

[17] Hanxun Huang, Yisen Wang, Sarah Erfani, Quanquan Gu, James Bailey, and Xingjun Ma.
Exploring architectural ingredients of adversarially robust deep neural networks.
In *Advances in Neural Information Processing Systems*, pages 5545–5559, 2021.
(Cited on pages 27, 28, 29, 30, 31, and 35.)

[18] Noureddine El Karoui.
The spectrum of kernel random matrices.
Annals of Statistics, 38(1):1–50, 2010.
(Cited on pages 39, 40, 41, 42, and 43.)

[19] Trevor Hastie, Andrea Montanari, Saharon Rosset, and Ryan J Tibshirani.
Surprises in high-dimensional ridgeless least squares interpolation.
Annals of Statistics, 50(2):949–986, 2022.
(Cited on pages 39, 40, 41, 42, and 43.)

[20] Emmanuel J Candes and Terence Tao.
Decoding by linear programming.
IEEE Transactions on Information Theory, 51(12):4203–4215, 2005.
(Cited on pages 39, 40, 41, 42, and 43.)

References VI

[21] 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.
(Cited on page 45.)

[22] Rahul Parhi and Robert D Nowak.
Near-minimax optimal estimation with shallow ReLU neural networks.
IEEE Transactions on Information Theory, 2022.
(Cited on page 45.)

[23] Lénaïc Chizat.
Convergence rates of gradient methods for convex optimization in the space of measures.
arXiv preprint arXiv:2105.08368, 2021.
(Cited on page 45.)

[24] Song Mei, Theodor Misiakiewicz, and Andrea Montanari.
Mean-field theory of two-layers neural networks: dimension-free bounds and kernel limit.
In *Conference on Learning Theory*, pages 2388–2464. PMLR, 2019.
(Cited on page 45.)

References VII

[25] Dheeraj Nagaraj, Xian Wu, Guy Bresler, Prateek Jain, and Praneeth Netrapalli.
Least squares regression with markovian data: Fundamental limits and algorithms.
In *Advances in Neural Information Processing Systems*, pages 16666–16676, 2020.
(Cited on page 45.)

[26] Senthil Purushwalkam, Pedro Morgado, and Abhinav Gupta.
The challenges of continuous self-supervised learning.
arXiv preprint arXiv:2203.12710, 2022.
(Cited on page 45.)

[27] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al.
Human-level control through deep reinforcement learning.
Nature, 518(7540):529–533, 2015.
(Cited on pages 46 and 47.)

[28] Zhuoran Yang, Chi Jin, Zhaoran Wang, Mengdi Wang, and Michael I Jordan.
On function approximation in reinforcement learning: Optimism in the face of large state spaces.
In *Advances in Neural Information Processing Systems*, 2020.
(Cited on page 48.)

References VIII

[29] Gene Li, Pritish Kamath, Dylan J Foster, and Nathan Srebro.
Eluder dimension and generalized rank.
arXiv preprint arXiv:2104.06970, 2021.
(Cited on pages 50, 51, and 52.)

[30] Kefan Dong, Jiaqi Yang, and Tengyu Ma.
Provable model-based nonlinear bandit and reinforcement learning: Shelve optimism, embrace virtual curvature.
In *Advances in Neural Information Processing Systems*, volume 34, 2021.
(Cited on pages 50, 51, and 52.)

[31] Rémi Munos and Csaba Szepesvári.
Finite-time bounds for fitted value iteration.
Journal of Machine Learning Research, 9(5), 2008.
(Cited on page 60.)

[32] Jinglin Chen and Nan Jiang.
Information-theoretic considerations in batch reinforcement learning.
In *International Conference on Machine Learning*, pages 1042–1051. PMLR, 2019.
(Cited on page 60.)

References IX

[33] Tengyang Xie, Dylan J Foster, Yu Bai, Nan Jiang, and Sham M Kakade.
The role of coverage in online reinforcement learning.
In *International Conference on Learning Representations*, 2023.
(Cited on page 60.)

[34] Krzysztof Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Lukasz Kaiser, and Weller Adrian.
Rethinking attention with performers.
In *International Conference on Learning Representations*, pages 1–14, 2021.
(Cited on pages 63, 64, and 65.)