Introduction:

Theoretical Understanding of Deep Neural Networks

presented by Shao-Qun Zhang

Emails :	zhangsq@lamda.nju.edu.cn	Websites:	DBLP
	zhangsqhndn@gmail.com		Google Scholar
Personal:	HomePage		ORCID
	CV		OpenReview

Date: April 1, 2022

Research Interests

My current research interests mainly include *Machine Learning* and *Data Mining*. More specifically, I am interested in the following topics:

- **Neural Computation & Deep Neural Networks**
  - **with Spiking Neural Networks**. Spiking neural networks (SNNs) take into account the time of spike firing rather than simply relying on the accumulated signal strength in conventional neural networks, and thus offering the possibility for modeling time-dependent data. Here, we provide a theoretical framework for investigating spiking neural models from the perspective of dynamical systems.
  - **with Neuroscience**. Recently, we proposed a novel bio-plausible neuron model, the *Flexible Transmitter* (FT) model. The FT model is inspired by the one-way communication neurotransmitter mechanism in nervous systems, and has the formation of a two-variable two-valued function, which takes the commonly-used MP neuron model as its special case. We empirically show its potential with handling spatio-temporal data and present theoretical understandings on the advantages of FT model.
  - **with Complex-valued Neural Networks**. Recent years have witnessed an increasing interest on complex-valued neural networks. Here, we formulate a practical formation of complex-valued neural networks and provide theoretical understandings on the merits of complex-valued neural networks in comparison with real-valued ones, especially in terms of approximation and optimization dynamics.

- **Deep Learning Theory**

  I am focusing on the theoretical understanding of deep learning in terms of approximation, optimization and generalization. Especially, I care more about the following issues:
  - **about Representation Learning**. Recently, I am working on theoretically investigate *Feature Space Transformation*, *Spatio-Temporal Representation*, *Approximate Capability*, and *Computational Complexity*, which might be a key to understand the mysteries behind the success of deep neural networks.
  - **about Geometry and Dynamics within Optimization**.

- **Time Series Analysis**
  - **about Forecasting Algorithm**. I am interested in time series forecasting, including *Accurate Forecasting*, *Quantitative Analysis*, *Uncertainty Estimation*, etc.
  - **about Forecasting Theory**. I also make some efforts on the forecasting theory, including *Predictable PAC Learning Theory* and *Long/Short-term Causal System*.

Outline

Deep Learning & Deep Neural Networks

Neural Network Learning = Neural Network Model + Learning Algorithm

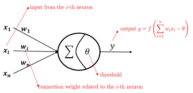
Learning Algorithm ⇩ e.g., BP algorithm

Neural Network Model ⇩ Neuron Model + Network Architecture

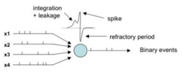
Neuron Model ⇩ e.g., MP model

Spiking Neural Model

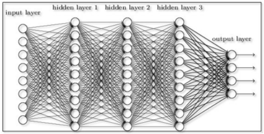
Network Architecture ⇩ e.g., feed-forward


input from the i-th neuron
x_1
w_1
x_i
w_i
w_n
x_n
$\sum$
θ
y
output $y = f\left(\sum_{i=1}^{n} w_i x_i - \theta\right)$
threshold
connection weight related to the i-th neuron

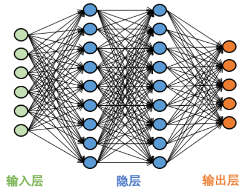
The 2nd generation of NNs



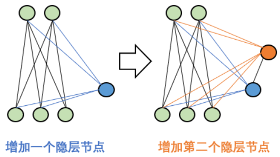
The 3rd generation of NNs



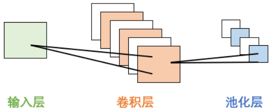
Architectures



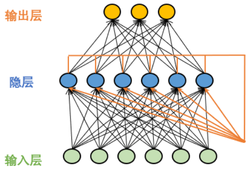
(a) FNN



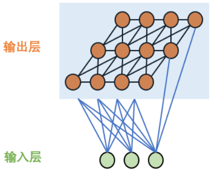
(b) CCN



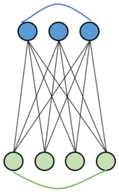
(c) CNN

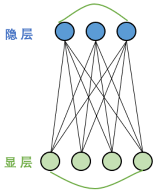


(d) RNN



(e) SOM


(f) Boltzmann 机



(g) RBM

Concerns

1. Approximation
   - Universal Approximation
   - Approximation Complexity (paras, time)
   - Representation Theory
2. Optimization
   - Training Algorithms (GD, SGD)
   - Local Minima (landscape)
3. Generalization (OOD)

Journals & Conferences

Journals:

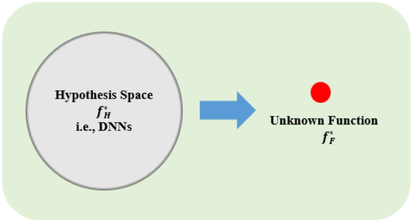
- AIJ, JMLR
- Neural Computation (NCJ), Neural Networks, TNNLS
- Neurocomputing
- TPAMI, TKDE, Machine Learning (MLJ)

Conferences:

- COLT, ITCS
- NeurIPS, ICML
- ICLR, AISTATS
- AAAI, IJCAI

Outline - Approximation

Deep Neural Networks can handle what function or task?



1. Universal Approximation
2. Approximation Complexity
3. Representation Theory

Universal Approximation

Universal Approximation Theorem

Let $K \subset \mathbb{R}^m$ be a compact set. If the activation function σ is well-posed (e.g., l-finite), then for all $r \in [l]$, the set of scalar functions $f : K \to \mathbb{R}$ of the form

$$f(\boldsymbol{x}) = \sum_{i \in [n]} v_i \sigma(\boldsymbol{w}_i^\top \boldsymbol{x} - b_i),$$

is dense in $\mathcal{C}^r(K, \mathbb{R})$.

Importance:

activations, connection weights (paras), architectures (FNN, ResNet, RNN)

Significance:

provides a legitimate guarantee for neural networks, corresponding to PAC identification for statistical learning.

Key Refs

- $\propto$ K.-I. Funahashi. On the Approximate Realization of Continuous Mappings by Neural Networks. Neural Networks. 1989.
- $\propto$ K. Hornik. Approximation Capabilities of Multilayer Feedforward Networks. Neural Networks. 1991.
- $\propto$ M. Leshno, V. Lin, A. Pinkus, S. Schocken. Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. Neural Networks. 1993.
- $\propto$ A. Barron. Approximation and Estimation Bounds for Artificial Neural Networks. Machine Learning. 1994.
- $\propto$ P. Kidger, T. Lyons. Universal Approximation with Deep Narrow Networks. In COLT'2016.

Approximation Complexity

1. Computational Complexity
   - FLOPS: floating point operations
   - flops: Floating-point Operations Per Second
   - MACCs: multiply-accumulate operations

   as indicators for running time, energy consumption, hardware, etc.

   such as how fast is my model?

   supported by TCS

2. Parameter Capacity/Capability
   The number of learnable parameters, i.e., $\sum_{l \in [L]} (|\text{weights}^l|_{\#} + |\boldsymbol{bias}^l|_{\#})$.

   show which architecture and what size are sufficient for the concerned tasks.

   upper bound

   only can be answered by theory!

Examples for Computational Complexity

Convolution Kernel

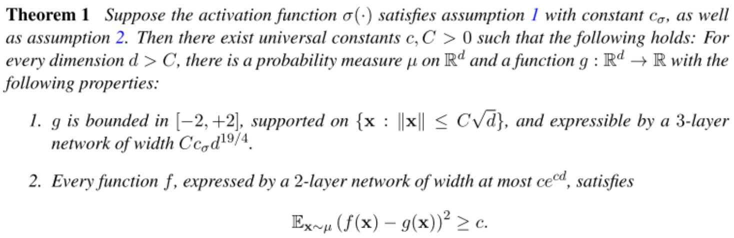
Kernel Size: $K_{height} \times K_{width}$
Output Size: $L_{height} \times L_{width}$

Number of Input Channel: C_{input}
Number of Output Channel: C_{output}

Then we have

- FLOPS: $2 \times (C_{input} \times K_{height} \times K_{width} - 1) \times L_{height} \times L_{width} \times C_{output}$,
- flops: $2 \times C_{input} \times K_{height} \times K_{width} \times L_{height} \times L_{width} \times C_{output}$,
- MACCs: $C_{input} \times K_{height} \times K_{width} \times L_{height} \times L_{width} \times C_{output}$.

Example for Parameter Capability



Theorem 1 *Suppose the activation function $\sigma(\cdot)$ satisfies assumption 1 with constant c_σ, as well as assumption 2. Then there exist universal constants $c, C > 0$ such that the following holds: For every dimension $d > C$, there is a probability measure μ on $\mathbb{R}^d$ and a function $g : \mathbb{R}^d \to \mathbb{R}$ with the following properties:*

1. *g is bounded in $[-2, +2]$, supported on $\{\mathbf{x} : \|\mathbf{x}\| \leq C\sqrt{d}\}$, and expressible by a 3-layer network of width $Cc_\sigma d^{19/4}$.*

2. *Every function f, expressed by a 2-layer network of width at most ce^{cd}, satisfies*

$$\mathbb{E}_{\mathbf{x}\sim\mu}\left(f(\mathbf{x}) - g(\mathbf{x})\right)^2 \geq c.$$

Highlight: There exists some radial function that can be approximated by a 3-layer FNN of width polynomial neurons, whereas approximated by a 2-layer FNN of width at least exponential neurons.

∝ R. Eldan and O. Shamir. The power of depth for feedforward neural networks. In COLT'2016.

Representation Theory

- & Kernel Learning,
  such as deep multiple kernels, neural network Gaussian process (NNGP), neural tangent kernel (NTK), etc.

- from Dynamical Systems, such as Neural ODE
  - $\propto$ R. TQ. Chen, Y. Rubanova, J. Bettencourt, D. K. Duvenaud. Neural Ordinary Differential Equations. In NIPS'2018 (best paper).

- from Functional Analysis, such as transforming features layer by layer.

DNN & Kernel

Kernel: input space → higher-dimensional feature space, such that the feature space is linearly separable.

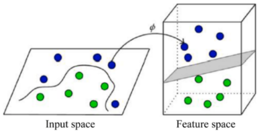


Fig. 1. Kernel mapping from input space to feature space.

- Cascading multiple kernels layer by layer
- Diversity (heuristic)

$$\begin{cases} \phi : \boldsymbol{x} \mapsto \phi(\boldsymbol{x}), \\ f(\boldsymbol{x}) = \sum_{i \in [n]} \alpha_i \langle \phi(\boldsymbol{x}), \phi(\boldsymbol{x}_i) \rangle . \end{cases}$$

feature mapping + (linear) prediction

NTK

a kernel mapping built on the tangent space of deep neural networks.

- A. Jacot, G. Franck, H. Clément. Neural Tangent Kernel: Convergence and Generalization in Neural Networks. In NIPS'2018.
- M. Belkin, M. Siyuan, M. Soumik. To Understand Deep Learning We Need to Understand Kernel Learning. In ICLR'2018.

Feed-forward Procedure:

$$\begin{cases} \boldsymbol{z}^{(\mathbf{0})}(\boldsymbol{x};\theta) = \boldsymbol{x} \\ \tilde{\boldsymbol{z}}^{(\ell+\mathbf{1})}(\boldsymbol{x};\theta) = \dfrac{1}{\sqrt{n_\ell}} W^{(\ell)} \boldsymbol{z}^{(\ell)}(\boldsymbol{x};\theta) + \beta b^{(\ell)} \\ \boldsymbol{z}^{(\ell)}(\boldsymbol{x};\theta) = \sigma\left(\tilde{\boldsymbol{z}}^{(\ell)}(\boldsymbol{x};\theta)\right) \end{cases}$$

$$\boldsymbol{y} = f^L(\boldsymbol{x};\theta) \quad \text{with} \quad \boldsymbol{z}^{(\ell)}(\boldsymbol{x};\theta) = f^\ell(\boldsymbol{x};\theta)$$

Tangent Approximation:

$$\begin{cases} \text{Primary:} \quad \boldsymbol{x} \mapsto f(\boldsymbol{x};\theta) \qquad \text{nonlinear} \\ \text{Tangent:} \quad \theta \mapsto \nabla_\theta f(\boldsymbol{x};\theta) \qquad \text{linear} \end{cases}$$

gradient-based kernel

$$\mathcal{K}^\ell_{ntk}(\boldsymbol{x},\boldsymbol{x}') = \left\langle \nabla_\theta f^\ell(\boldsymbol{x};\theta), \nabla_\theta f^\ell(\boldsymbol{x}';\theta) \right\rangle,$$

and as $n_\ell \to \infty$, the following holds from CLT

$$\lim_{n_\ell\to\infty} \mathbb{E}_{\theta\sim\mathcal{N}(0,\Sigma^\ell)} \left[\mathcal{K}^\ell_{ntk}\right] = \mathcal{K}^*_{ntk}.$$

NTK

Strengths:

- regarding one layer of neurons as a kernel, contributing to representation learning
- a well-formulated cornerstone for theory. For example, bounding errors by calculating the eigenvalues of this kernel

Weaknesses:

- CLT holds as $n \to \infty$, that is, infinite width, beyond computation
- all layers and architectures (e.g., RNN, ResNet, FTNet, etc.) are Gaussian kernels with the same mean and covariance.

NNGP

regard the primary DNN as a cascade compound of Gaussian kernels.

$\propto$ J. Lee, Y. Bahri, R. Novak, S. S. Schoenholz, J. Pennington, J. Sohl-dickstein. Deep Neural Networks as Gaussian Processes. In ICLR'2018.

Feed-forward Procedure:

$$\begin{cases} \boldsymbol{z}^{(\mathbf{0})}(\boldsymbol{x};\theta) = \boldsymbol{x} \\ \tilde{\boldsymbol{z}}^{(\ell+\mathbf{1})}(\boldsymbol{x};\theta) = \dfrac{1}{\sqrt{n_\ell}} W^{(\ell)} \boldsymbol{z}^{(\ell)}(\boldsymbol{x};\theta) + \beta b^{(\ell)} \\ \boldsymbol{z}^{(\ell)}(\boldsymbol{x};\theta) = \sigma\left(\tilde{\boldsymbol{z}}^{(\ell)}(\boldsymbol{x};\theta)\right) \end{cases}$$

$$\boldsymbol{y} = f^L(\boldsymbol{x};\theta) \quad \text{with} \quad \boldsymbol{z}^{(\ell)}(\boldsymbol{x};\theta) = f^\ell(\boldsymbol{x};\theta)$$

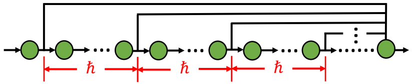
NNGP:

$$\begin{cases} \text{Primary:} \quad \boldsymbol{x} \mapsto f(\boldsymbol{x};\theta) \quad \textcolor{red}{\text{nonlinear}} \\ NNGP: \quad \mathcal{K}^\ell_{nngp}(\boldsymbol{x},\boldsymbol{x}') = \left\langle f^\ell(\boldsymbol{x};\theta), f^\ell\left(\boldsymbol{x}';\theta\right)\right\rangle, \end{cases}$$

as $n_\ell \to \infty$, the following holds from CLT

$$\lim_{n_\ell \to \infty} \mathbb{E}_{\theta\sim\mathcal{N}(0,\Sigma^\ell)}\left[\mathcal{K}^\ell_{nngp}\right] = \mathcal{K}^*_{nngp}.$$

Our Work: Deep NNGP

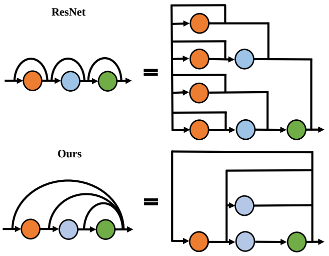


Feed-forward Procedure:

$$\begin{cases} \boldsymbol{z}^{\mathbf{0}} = \boldsymbol{x} \\ \boldsymbol{z}^{l} = \sigma(\mathrm{W}^{l}\boldsymbol{z}^{l-\mathbf{1}} + \boldsymbol{b}^{l}) \\ f(\boldsymbol{x};\boldsymbol{\theta}) = \frac{\mathbf{1}}{\sqrt{M_{\boldsymbol{z}}}} \sum_{\kappa=\mathbf{0}}^{K} \mathbf{1}^{l_{\mathbf{1}}+\kappa\hbar} \boldsymbol{z}^{l_{\mathbf{1}}+\kappa\hbar}. \end{cases}$$

Conclusion: with mild assumptions on connection parameters and $\hbar$, this network induces a kernel via generalized CLT.

Comparison:



Strengths: 1. This kernel is uniformly tight;

2. We can bound its smallest eigenvalue $\lambda_{\min}\left(\mathrm{K}_{\mathcal{D},\mathcal{D}}\right) = \Theta(d)$.

Our Work: Deep NNGP

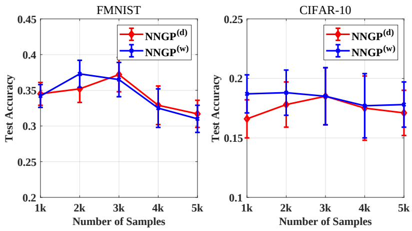


Figure 1: The performance of the NNGP$^{(d)}$ and NNGP$^{(w)}$ kernels constructed by different number of samples on FMNIST and CIFAR-10.

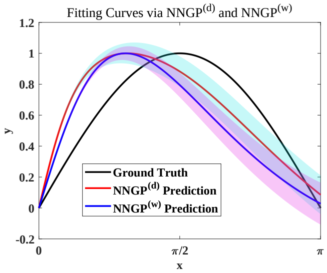


Figure 2: The fitting curves of the NNGP$^{(d)}$ and NNGP$^{(w)}$ kernels for a sine function over $[0, \pi]$.

$\propto$ S.-Q. Zhang, F.-L. Fan. Neural Network Gaussian Processes by Increasing Depth. arXiv:2108.12862. 2021. (submitted to TNNLS)

Future Issues

- whether the depth is more important than width?
- approximation complexity
- white-box representation learning (for science)

Optimization

In contrast to the Approximation theory that concerns about

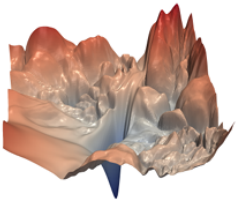
- the expressive power led by architectures and learnable parameters
- $\exists$ some points (including architecture and a group of paras), such that the concerned neural network is apposite.

Optimization theory in neural networks focuses on

- can we (algorithms) find this point or these points?
- how find? (algorithms, mechanisms, etc.)
- how fast?

Observations

- extremely non-convex landscape



- local minima usually is sufficient.

$$\exists\ \theta \text{ s.t. } |f(\boldsymbol{x};\theta) - f(\boldsymbol{x};\theta^*)| < \epsilon$$

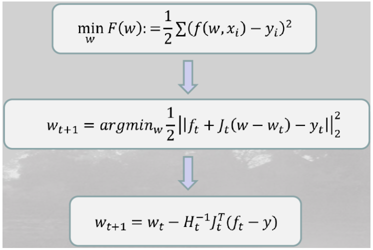
where θ^* is the optimal point.

This is a very tricky problem.

- S. Ruder. An Overview of Gradient Descent Optimization Algorithms. arXiv:1609.04747. 2016.
- H. Li, Z. Xu, G. Taylor, C. Studer, and T. Goldstein. Visualizing the loss landscape of neural nets. NIPS'2018.

Future Issues

[Q1:] developing (fast and stable) algorithms for DNNs



$$\min_{w} F(w) := \frac{1}{2}\sum(f(w,x_i)-y_i)^2$$

$$w_{t+1} = argmin_w \frac{1}{2}||f_t + J_t(w-w_t)-y_t||_2^2$$

$$w_{t+1} = w_t - H_t^{-1}J_t^T(f_t - y)$$

- Tensor Decomposition
- Half Space
- Sparse Connection
- Kernel-based Method
- Implicit Bias
- Margin Maximization

Future Issues

[Q2:] investigating SGD from dynamical systems
It is observed that

$$\text{GD:} \quad \theta(t+1) = \theta(t) - \eta \nabla_\theta L(\theta(t)) \quad \text{with} \quad L(\theta) = \frac{1}{|\mathcal{S}|} \sum_{\boldsymbol{x} \in \mathcal{S}} L(\boldsymbol{x}; \theta)$$

can be regarded as

$$\text{ODE:} \quad \mathrm{d}\theta = -\nabla_\theta L(\theta(t))\, \mathrm{d}t.$$

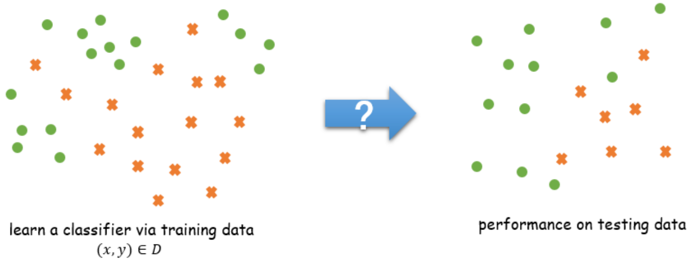
Further, we have

$$\text{SGD:} \quad \theta(t+1) = \theta(t) - \eta \nabla_\theta L_r(\theta(t)) \quad \text{with} \quad L_r(\theta) = \frac{1}{|\mathcal{S}_r|} \sum_{\boldsymbol{x} \in \mathcal{S}_r \subseteq \mathcal{S}} L(\boldsymbol{x}; \theta)$$

can be regarded as

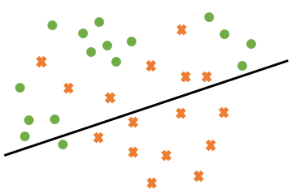
$$\text{SDE:} \quad \mathrm{d}\theta = -\nabla_\theta L(\theta(t))\, \mathrm{d}t + \sigma\, \mathrm{d}\xi(t), \quad \text{where} \quad \xi(t) \sim \mathcal{N}(0, \sigma^2).$$

Generalization

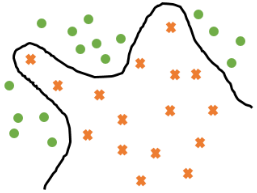

learn a classifier via training data
$(x, y) \in D$
?
performance on testing data

$$\mathbb{L}_{testing}(\theta) \leqslant \widehat{\mathbb{L}}_{training}(\theta) + \text{something} \ .$$

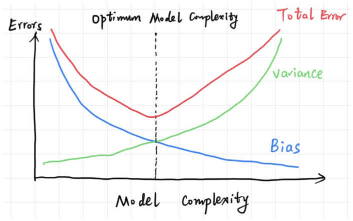
Underfitting VS Overfitting



under-fitting



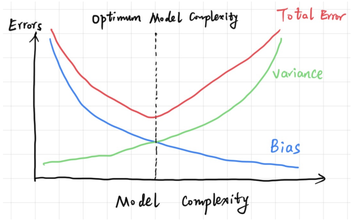
over-fitting

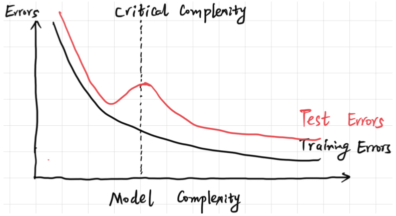


classical machine learning:

$$\text{Errors} = \text{Bias} + \text{Variance} + \epsilon$$

Generalization





classical machine learning:

$$\text{Errors} = \text{Bias} + \text{Variance} + \epsilon$$

deep neural networks:

Double Descents against Overfitting

∝ M. Loog, T. Viering, A. Mey, J. H. Krijthe, D. MJ. Tax. A brief prehistory of double descent. In Proceedings of the National Academy of Sciences 117(20): 10625-10626. 2020.

Generalization

Common form of generalization bound

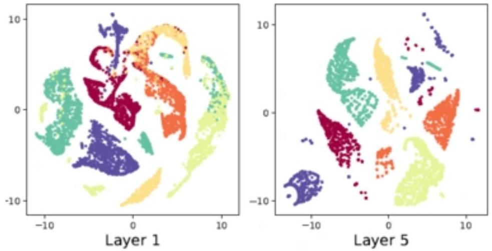
$$\mathbb{L}_{testing}(\theta) \leqslant \widehat{\mathbb{L}}_{training}(\theta) + \sqrt{|\mathcal{H}|/n}\ .$$

In depth, $|\mathcal{H}|$ can be bounded by

VC	$\mathcal{O}(\lvert\text{edges}\rvert \log(\lvert\text{edges}\rvert))$ or $\propto$ depth $\times$ width
ϵ-covering number	$\mathcal{O}((AL)^{L(L+1)}/\epsilon^{2L})$
Rademacher Average	$\mathcal{R}_m(\mathcal{H}) \leqslant \mathcal{O}(\mu^L)$

Generalization

Margin theory and norms (nearly no dependence on the parameter capacity)



- P. L. Bartlett, D. J. Foster, M. J. Telgarsky. Spectrally-normalized Margin Bounds for Neural Networks. In NIPS'2017.
- S.-H. Lyu, L. Wang, Z.-H. Zhou. Improving Generalization of Neural Networks by Leveraging Margin Distribution. Neural Networks, in press. 2022.

Future Issues

- ♠ Double Descents against Overfitting.

  ∝ Z.-H. Zhou. Why over-parameterization of deep neural networks does not overfit? Science China Information Sciences, 64(1):1-3, 2021.

- ♠ Lacking a formal description, there are some efforts from NTK.

- ♠ Impact Regularization

理论的作用是什么

大致来说，学习理论至少有这样几个方面的作用：

- 学习理论可以告诉我们：某件事能做吗？
  - 例如，若学习理论告诉我们某个问题是“不可学”或者“不可预测的”，那么我们就不必徒劳地去设计机器学习算法了。
  - Universal Approximation 对于 DNNs 的作用在此

- 学习理论可以告诉我们：什么因素重要？
  - 例如，我们发现某个问题的泛化误差界为 $\sqrt{d/m}$，则我们知道在设计机器学习模型或者算法的时候，需要尽量地去减小 d 而增大 m。
  - Approximation Complexity, Generalization Bound 等理论都在考虑这个问题

- 学习理论可以告诉我们：能够做得更好？
  - 例如，若我们发现某个优化算法的收敛率为 $O(1/T)$，而学习理论告诉我们这个问题的最优速率为 $O(1/T^2)$，那么我们还可以努力去改进或者寻找更快的机器学习算法。
  - Optimization，Generalization Bound 等理论都在考虑这个问题

Thank you!

Q & A and Continuing...