Overview of Recent Progresses in Theoretical Deep Learning

Lei Wu

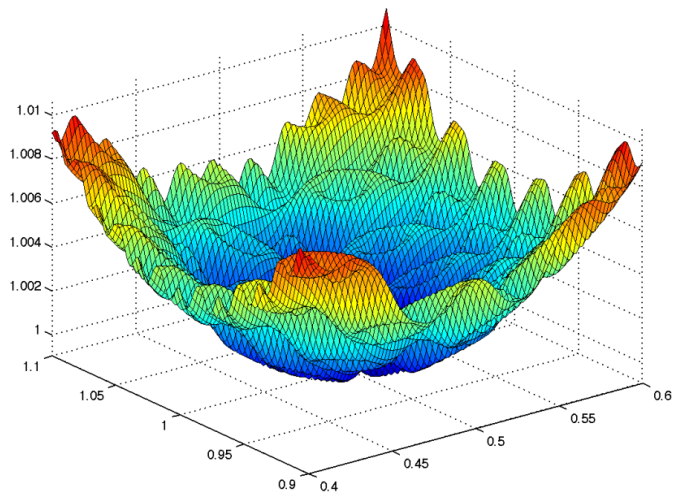
PACM, Princeton University

July 23, 2019

Outline

The mysteries in deep learning

- **Optimization:** The landscape $\hat{R}_n(\cdot)$ is highly non-convex. Why can gradient descent can find the global minima?
  1. Saddle point
  2. Local minima
  3. Convergence rate



- **Approximation:** What kind of functions can be efficiently approximated by neural network?
  1. High-dimensioanlity.

The mysteries in deep learning (cont'd)

- **Generalization:** Why the solution can generalize?
  1. Over-parameterization
  2. High-dimensioanlity. Traditional error rate is $O(\frac{1}{n^{s/d}})$.
  3. Implicit Regularizations.

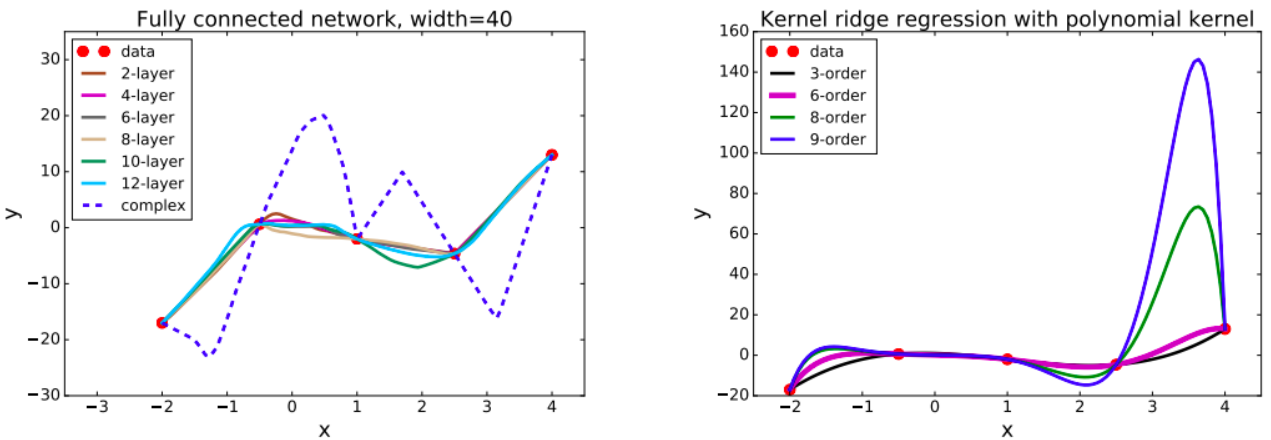


Figure 1: **(Left),** fitting results for 5 data points using FNNs with different number of layers; the overfitting solution with a high complexity (in dashed line) is intentionally constructed. **(Right),** fitting results by kernel regression with different orders of polynomial kernels.

Landscape analysis: Linear net [1]

We can then write the output of a feedforward deep linear model, $\overline{Y}(W, X) \in \mathbb{R}^{d_y \times m}$, as

$$\overline{Y}(W, X) = W_{H+1} W_H W_{H-1} \cdots W_2 W_1 X.$$

We consider one of the most widely used loss functions, squared error loss:

$$\bar{\mathcal{L}}(W) = \frac{1}{2} \sum_{i=1}^{m} \|\overline{Y}(W, X)_{\cdot,i} - Y_{\cdot,i}\|_2^2 = \frac{1}{2} \|\overline{Y}(W, X) - Y\|_F^2,$$

Theorem 2.3 (Loss surface of *deep* linear networks) *Assume that XX^T and XY^T are of full rank with $d_y \leq d_x$ and Σ has d_y distinct eigenvalues. Then, for any depth $H \geq 1$ and for any layer widths and any input-output dimensions $d_y, d_H, d_{H-1}, \ldots, d_1, d_x \geq 1$ (the widths can arbitrarily differ from each other and from d_y and d_x), the loss function $\bar{\mathcal{L}}(W)$ has the following properties:*

(i) It is non-convex and non-concave.

(ii) Every local minimum is a global minimum.

(iii) Every critical point that is not a global minimum is a saddle point.

(iv) If $\operatorname{rank}(W_H \cdots W_2) = p$, then the Hessian at any saddle point has at least one (strictly) negative eigenvalue.[1]

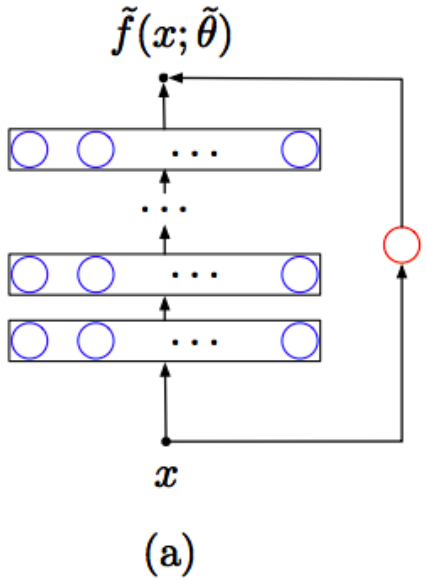
[1] Kenji Kawaguchi, Deep Learning without Poor Local Minima

Landscape analysis: Nonlinear net [2]

Assumption 1 (Loss function) *Assume that the loss function $\ell : \mathbb{R} \to \mathbb{R}$ is monotonically non-decreasing and twice differentiable, i.e., $\ell \in C^2$. Assume that every critical point of the loss function $\ell(z)$ is also a global minimum and every global minimum z satisfies $z < 0$.*

Assumption 2 (Realizability) *Assume that there exists a set of parameters $\boldsymbol{\theta}$ such that the neural network $f(\cdot; \boldsymbol{\theta})$ is able to correctly classify all samples in the dataset $\mathcal{D}$.*

$$\tilde{L}_n(\tilde{\boldsymbol{\theta}}) = \sum_{i=1}^{n} \ell\left(-y_i(f(x_i;\boldsymbol{\theta}) + a\exp(\boldsymbol{w}^\top x_i + b))\right) + \frac{\lambda a^2}{2}.$$



(a)

[2] Shiyu Liang, et al, Adding One Neuron Can Eliminate All Bad Local Minima

Landscape analysis: Nonlinear net

Theorem 1 *Suppose that Assumption 1 and 2 hold. Assume that* $\tilde{\boldsymbol{\theta}}^* = (\boldsymbol{\theta}^*, a^*, \boldsymbol{w}^*, b^*)$ *is a local minimum of the empirical loss function* $\tilde{L}_n(\tilde{\boldsymbol{\theta}})$*, then* $\tilde{\boldsymbol{\theta}}^*$ *is a global minimum of* $\tilde{L}_n(\tilde{\boldsymbol{\theta}})$*. Furthermore,* $\boldsymbol{\theta}^*$ *achieves the minimum loss value and the minimum misclassification rate on the dataset* $\mathcal{D}$*, i.e.,* $\boldsymbol{\theta}^* \in \arg\min_{\boldsymbol{\theta}} L_n(\boldsymbol{\theta})$ *and* $\boldsymbol{\theta}^* \in \arg\min_{\boldsymbol{\theta}} R_n(\boldsymbol{\theta}; f)$.

Landscape analysis: nonlinear net [3]

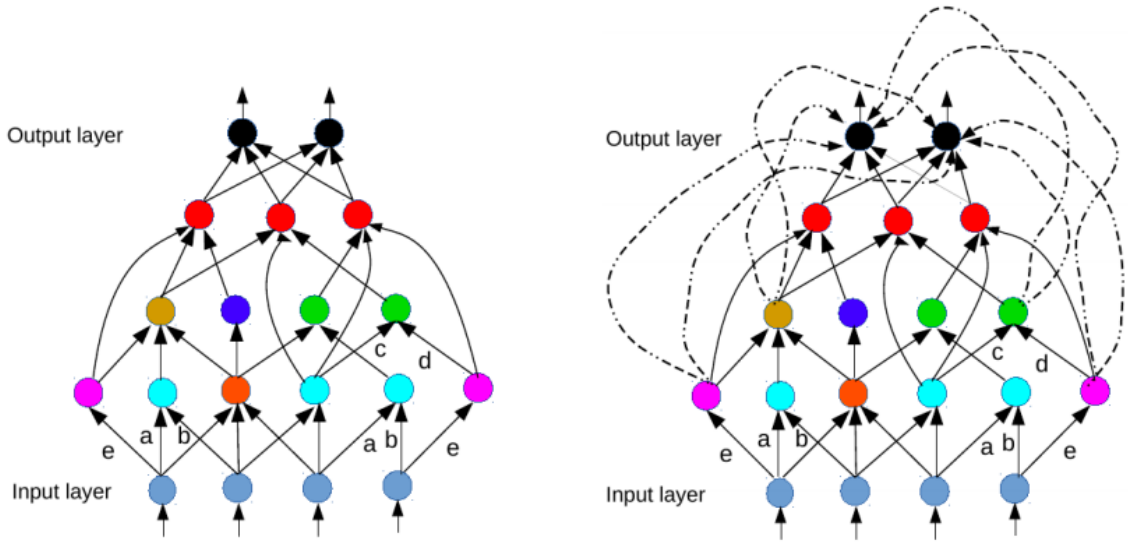


Figure 2: **Left**: An example neural network represented as directed acyclic graph. **Right**: The same network with skip connections added from a subset of hidden neurons to the output layer. All neurons with the same color can have shared or non-shared weights.

Theorem 3.4 *The following holds under Assumption 3.1:*

1. *There exist uncountably many solutions with zero training error.*
2. *The loss landscape of Φ does not have any bad local valley.*
3. *There exists no suboptimal strict local minimum.*
4. *There exists no local maximum.*

[3] Quynh Nguyen, et al, On the loss landscape of a class of deep neural networks with no bad local valleys

Convergence: linear network [4]

$$\ell(W_1,\dots,W_L) = \frac{1}{2}\sum_{p=1}^{n}\left\|\frac{1}{\sqrt{m^{L-1}d_{\text{out}}}}W_L\cdots W_1 x_p - y_p\right\|^2 = \frac{1}{2}\left\|\frac{1}{\sqrt{m^{L-1}d_{\text{out}}}}W_L\cdots W_1 X - Y\right\|_F^2, \tag{1}$$

where $W_1 \in \mathbb{R}^{m\times d_{\text{in}}}, W_2,\dots,W_{L-1}\in\mathbb{R}^{m\times m}$ and $W_L\in\mathbb{R}^{m\times d_{\text{out}}}$ are weight matrices to be learned. Here $\frac{1}{\sqrt{m^{L-1}d_{\text{out}}}}$ is a scaling factor corresponding to Xavier initialization[4] (Glorot & Bengio, 2010), for which we provide a justification in Section 3.3.

- We initialize all the entries of $W_1,\dots,W_L$ independently from $\mathcal{N}(0,1)$. Let $W_1(0),\dots,W_L(0)$ be the weight matrices at initialization.
- Then we update the weights using GD: for $t=0,1,2,\dots$ and $i\in[L]$,

$$W_i(t+1) = W_i(t) - \eta\frac{\partial \ell}{\partial W_i}(W_1(t),\dots,W_L(t)), \tag{2}$$

where $\eta>0$ is the learning rate.

Theorem 4.1. *Suppose*

$$m \geq C\cdot L\cdot \max\left\{r\kappa^3 d_{\text{out}}(1+\|\Phi\|^2), r\kappa^3\log\frac{r}{\delta}, \log L\right\} \tag{4}$$

for some $\delta\in(0,1)$ and a sufficiently large universal constant $C>0$ and we set $\eta\leq\frac{d_{\text{out}}}{3L\|X^\top X\|}$. Then with probability at least $1-\delta$ over the random initialization, we have

$$\ell(0) - \mathsf{OPT} \leq O\left(\max\left\{1, \frac{\log(r/\delta)}{d_{\text{out}}}, \|\Phi\|^2\right\}\right)\|X\|_F^2,$$

$$\ell(t) - \mathsf{OPT} \leq \left(1 - \frac{\eta L\cdot\lambda_r(X^\top X)}{4d_{\text{out}}}\right)^t(\ell(0)-\mathsf{OPT}).$$

[4] Width Provably Matters in Optimization for Deep Linear Neural Networks

Convergence: linear network [5]

$$\mathcal{R}(W_1, \dots, W_L) = \frac{1}{2}\|W_L \cdots W_1 - \Phi\|_F^2.$$

$$W_l(0) = I,\ l = 1, \dots, L-1, \quad \text{and} \quad W_L(0) = 0.$$

Theorem 4.3 (Discrete gradient descent). *For deep linear network (2.3) with zero-asymmetric initialization (3.1) and discrete-time gradient descent (2.4), if the learning rate satisfies*

$$\eta \le \min\left\{\frac{1}{4L^3\phi^6}, \frac{1}{144L^2\phi^4}\right\}$$

where $\phi = \max\left\{2\|\Phi\|_F, 3L^{-1/2}, 1\right\}$, *then we have linear convergence*

$$\mathcal{R}(t) \le \left(1 - \frac{\eta}{2}\right)^t \mathcal{R}(0), \quad t = 0, 1, 2, \dots \tag{4.5}$$

[5] On the Importance of Initialization in Optimization for Deep Linear Neural Networks

Approximation of deep ReLU network: [6]

$$\|f\|_{W^{n,\infty}([0,1]^d)} = \max_{\mathbf{n}:|\mathbf{n}|\le n} \operatorname{ess\,sup}_{\mathbf{x}\in[0,1]^d} |D^{\mathbf{n}} f(\mathbf{x})|,$$

$$F_{n,d} = \{f \in W^{n,\infty}([0,1]^d) : \|f\|_{W^{n,\infty}([0,1]^d)} \le 1\}.$$

Theorem 1. *For any d, n and $\epsilon \in (0,1)$, there is a ReLU network architecture that*

1. *is capable of expressing any function from $F_{d,n}$ with error ϵ;*
2. *has the depth at most $c(\ln(1/\epsilon)+1)$ and at most $c\epsilon^{-d/n}(\ln(1/\epsilon)+1)$ weights and computation units, with some constant $c = c(d,n)$.*

[6] Dmitry Yarotsky, Error bounds for approximations with deep ReLU networks

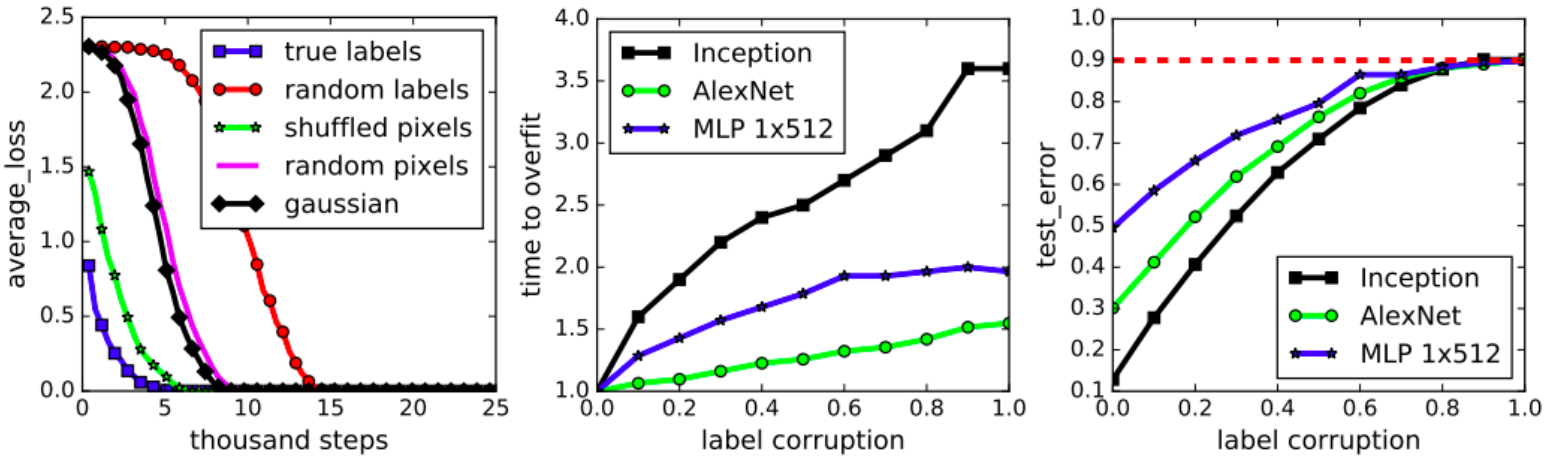
Depth separation: [7]

Abstract

Let $f : \mathbb{S}^{d-1} \times \mathbb{S}^{d-1} \to \mathbb{R}$ be a function of the form $f(\mathbf{x}, \mathbf{x}') = g(\langle \mathbf{x}, \mathbf{x}' \rangle)$ for $g : [-1, 1] \to \mathbb{R}$. We give a simple proof that shows that poly-size depth two neural networks with (exponentially) bounded weights cannot approximate f whenever g cannot be approximated by a low degree polynomial. Moreover, for many g's, such as $g(x) = \sin(\pi d^3 x)$, the number of neurons must be $2^{\Omega(d \log(d))}$. Furthermore, the result holds w.r.t. the uniform distribution on $\mathbb{S}^{d-1} \times \mathbb{S}^{d-1}$. As many functions of the above form can be well approximated by poly-size depth three networks with poly-bounded weights, this establishes a separation between depth two and depth three networks w.r.t. the uniform distribution on $\mathbb{S}^{d-1} \times \mathbb{S}^{d-1}$.

[7] Amit Daniely, Depth Separation for Neural Networks

UNDERSTANDING DEEP LEARNING REQUIRES RETHINKING GENERALIZATION



(a) learning curves (b) convergence slowdown (c) generalization error growth

Figure 1: Fitting random labels and random pixels on CIFAR10. (a) shows the training loss of various experiment settings decaying with the training steps. (b) shows the relative convergence time with different label corruption ratio. (c) shows the test error (also the generalization error since training error is 0) under different label corruptions.

Theorem 1. *There exists a two-layer neural network with ReLU activations and $2n + d$ weights that can represent any function on a sample of size n in d dimensions.*

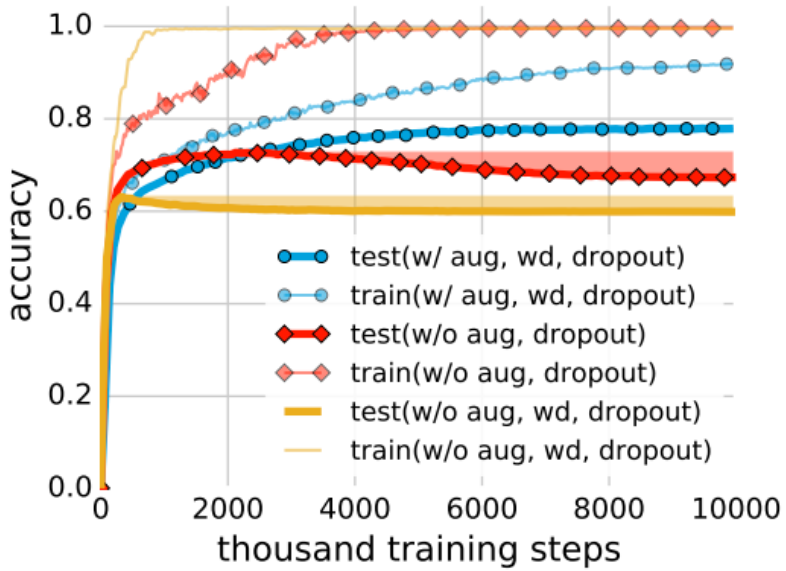
UNDERSTANDING DEEP LEARNING REQUIRES RETHINKING GENERALIZATION

Rademacher complexity and VC-dimension. Rademacher complexity is commonly used and flexible complexity measure of a hypothesis class. The empirical Rademacher complexity of a hypothesis class $\mathcal{H}$ on a dataset $\{x_1, \ldots, x_n\}$ is defined as

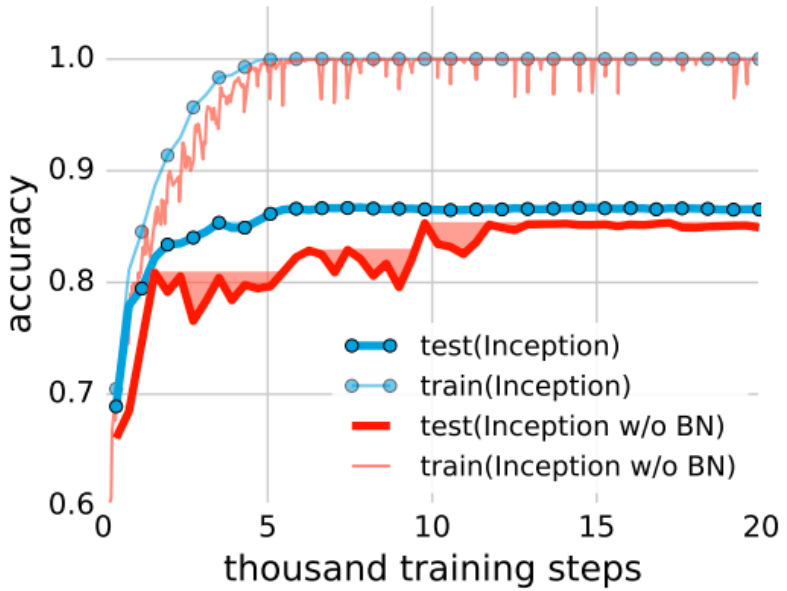
$$\hat{\mathfrak{R}}_n(\mathcal{H}) = \mathbb{E}_\sigma \left[\sup_{h \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} \sigma_i h(x_i) \right] \tag{1}$$

where $\sigma_1, \ldots, \sigma_n \in \{\pm 1\}$ are i.i.d. uniform random variables. This definition closely resembles our randomization test. Specifically, $\hat{\mathfrak{R}}_n(\mathcal{H})$ measures ability of $\mathcal{H}$ to fit random ± 1 binary label assignments. While we consider multiclass problems, it is straightforward to consider related binary classification problems for which the same experimental observations hold. Since our randomization tests suggest that many neural networks fit the training set with random labels perfectly, we expect that $\hat{\mathfrak{R}}_n(\mathcal{H}) \approx 1$ for the corresponding model class $\mathcal{H}$. This is, of course, a trivial upper bound on the Rademacher complexity that does not lead to useful generalization bounds in realistic settings.

UNDERSTANDING DEEP LEARNING REQUIRES RETHINKING GENERALIZATION



(a) Inception on ImageNet



(b) Inception on CIFAR10

UNDERSTANDING DEEP LEARNING REQUIRES RETHINKING GENERALIZATION

In this work we presented a simple experimental framework for defining and understanding a notion of *effective capacity* of machine learning models. The experiments we conducted emphasize that the effective capacity of several successful neural network architectures is large enough to shatter the training data. Consequently, these models are in principle rich enough to memorize the training data. This situation poses a conceptual challenge to statistical learning theory as traditional measures of model complexity struggle to explain the generalization ability of large artificial neural networks. We argue that we have yet to discover a precise formal measure under which these enormous models are simple. Another insight resulting from our experiments is that optimization continues to be empirically easy even if the resulting model does not generalize. This shows that the reasons for why optimization is empirically easy must be different from the true cause of generalization.

Neural Tangent Kernel [8]

$$
\begin{aligned}
f(\boldsymbol{\theta}, \boldsymbol{x}) &= f^{(L+1)}(\boldsymbol{x}) = \boldsymbol{W}^{(L+1)} \cdot \boldsymbol{g}^{(L)}(\boldsymbol{x}) \\
&= \boldsymbol{W}^{(L+1)} \cdot \sqrt{\frac{c_\sigma}{d_L}} \sigma\left(\boldsymbol{W}^{(L)} \cdot \sqrt{\frac{c_\sigma}{d_{L-1}}} \sigma\left(\boldsymbol{W}^{(L-1)} \cdots \sqrt{\frac{c_\sigma}{d_1}} \sigma\left(\boldsymbol{W}^{(1)} \boldsymbol{x}\right)\right)\right),
\end{aligned}
$$

$$
\ker\left(\boldsymbol{x}, \boldsymbol{x}'\right) = \mathop{\mathbb{E}}_{\boldsymbol{\theta} \sim \mathcal{W}} \left\langle \frac{\partial f(\boldsymbol{\theta}, \boldsymbol{x})}{\partial \boldsymbol{\theta}}, \frac{\partial f(\boldsymbol{\theta}, \boldsymbol{x}')}{\partial \boldsymbol{\theta}} \right\rangle.
$$

$$
f_{ntk}\left(\boldsymbol{x}_{te}\right) = \left(\ker_{ntk}\left(\boldsymbol{x}_{te}, \boldsymbol{X}\right)\right)^{\top} \left(\boldsymbol{H}^{*}\right)^{-1} \boldsymbol{y}.
$$

Theorem 3.2 (Main theorem). *Suppose* $\sigma(z) = \max(0, z)$ $(z \in \mathbb{R})$, $1/\kappa = \mathrm{poly}(1/\epsilon, \log(n/\delta))$ *and* $d_1 = d_2 = \cdots = d_L = m$ *with* $m \geq \mathrm{poly}(1/\kappa, L, 1/\lambda_0, n, \log(1/\delta))$. *Then for any* $\boldsymbol{x}_{te} \in \mathbb{R}^d$ *with* $\|\boldsymbol{x}_{te}\| = 1$, *with probability at least* $1 - \delta$ *over the random initialization, we have*

$$
|f_{nn}(\boldsymbol{x}_{te}) - f_{ntk}(\boldsymbol{x}_{te})| \leq \epsilon.
$$

[8] On Exact Computation with an Infinitely Wide Neural Net

NTK series

- Neural Tangent Kernel: Convergence and Generalization in Neural Networks
- Fine-grained analysis of optimization and generalization for overparameterized two-layer neural networks
- Gradient descent finds global minima of deep neural networks
- A convergence theory for deep learning via over-parameterization
- Learning and generalization in overparameterized neural networks, going beyond two layers
- Stochastic gradient descent optimizes over-parameterized deep relu networks
- A generalization theory of gradient descent for learning over-parameterized deep relu networks
- Generalization Bounds of Stochastic Gradient Descent for Wide and Deep Neural Networks
- ...

Mean field view

$$f_n(\boldsymbol{x}) = \frac{1}{n}\sum_{i=1}^{n} c_i \varphi(\boldsymbol{x}, \boldsymbol{y}_i)$$

$$\ell(f, f_n) = \tfrac{1}{2}\int_\Omega \left|f(\boldsymbol{x}) - f_n(\boldsymbol{x})\right|^2 d\mu(\boldsymbol{x})$$

$$\ell(f, f_n) = C_f - \frac{1}{n}\sum_{i=1}^{n} c_i F(\boldsymbol{y}_i) + \frac{1}{2n^2}\sum_{i,j=1}^{n} c_i c_j K(\boldsymbol{y}_i, \boldsymbol{y}_j) \tag{1.3}$$

where $C_f = \frac{1}{2}\int_\Omega \left|f(\boldsymbol{x})\right|^2 d\mu(\boldsymbol{x})$ and we defined

$$F(\boldsymbol{y}) = \int_\Omega f(\boldsymbol{x})\varphi(\boldsymbol{x}, \boldsymbol{y}) d\mu(\boldsymbol{x}), \qquad K(\boldsymbol{y}, \boldsymbol{z}) = \int_\Omega \varphi(\boldsymbol{x}, \boldsymbol{y})\varphi(\boldsymbol{x}, \boldsymbol{z}) d\mu(\boldsymbol{x}) \equiv K(\boldsymbol{z}, \boldsymbol{y}). \tag{1.4}$$

$$\ell(f, \tilde{f}) = C_f - \int_D F(\boldsymbol{y})G(\boldsymbol{y}) d\boldsymbol{y} + \tfrac{1}{2}\int_{D\times D} K(\boldsymbol{y}, \boldsymbol{z})G(\boldsymbol{y})G(\boldsymbol{z}) d\boldsymbol{y} d\boldsymbol{z}$$

Mean field view

- A Mean Field View of the Landscape of Two-Layer Neural Networks
- Neural Networks as Interacting Particle Systems: Asymptotic Convexity of the Loss Landscape and Universal Scaling of the Approximation Error
- Mean Field Analysis of Neural Networks.

Thank You