Learning Over-parameterized Neural Networks: From Neural Tangent Kernel to Mean-field Analysis

Quanquan Gu

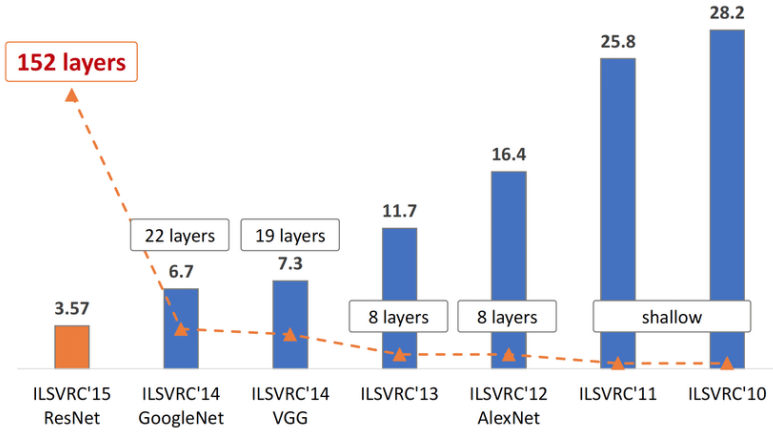
Computer Science Department
UCLA

IPAM Workshop on PDE and Inverse Problem Methods in Machine Learning

Joint work with Zixiang Chen, Yuan Cao and Tong Zhang

The Rise of Deep Learning

The evolution of the winning entries on the ImageNet

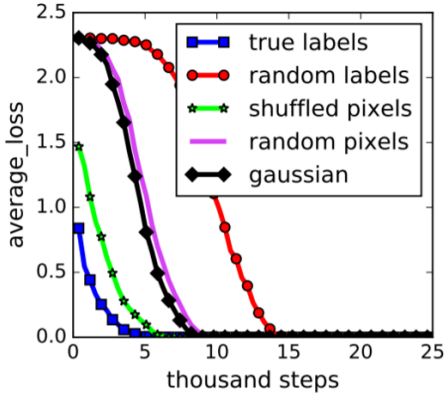


Alex Krizhevsky et al. 2012. "Imagenet classification with deep convolutional neural networks". In *Advances in neural information processing systems*, 1097–1105

Kien Nguyen et al. 2017. "Iris recognition with off-the-shelf CNN features: A deep learning perspective". *IEEE Access* 6:18848–18855

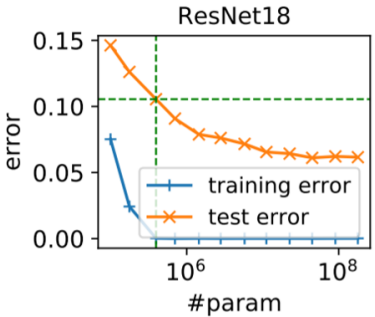
Deep Learning Can Fit Random Labels and Random Data

Wide deep neural networks can be trained to a global minimum despite the highly non-convex optimization landscape



Chiyuan Zhang et al. 2017. "Understanding deep learning requires rethinking generalization". In *International Conference on Learning Representations*

Challenges in Deep Learning Theory



Optimization

The optimization landscape is highly non-convex. Why doesn't (S)GD get stuck at a bad local minimum?

Generalization

There are much more network parameters than training examples. Why don't deep neural networks overfit?

Behnam Neyshabur et al. 2018b. "Towards Understanding the Role of Over-Parametrization in Generalization of Neural Networks". *arXiv preprint arXiv:1805.12076*

Numerous Lines of Research on Optimization and Generalization Theory for Deep learning

- Information bottleneck/invariance (Tishby and Zaslavsky 2015; Achille and Soatto 2018)
- Rademacher complexity bounds (Bartlett et al. 2017; Golowich et al. 2018), etc.
- PAC-Bayes bounds (Dziugaite and Roy 2017; Neyshabur et al. 2018a), etc.
- Neural tangent kernel/Lazy training (Jacot et al. 2018; Du et al. 2019a; Allen-Zhu et al. 2019b; Du et al. 2019b; Zou et al. 2018; Chizat et al. 2019) etc.
- Mean-field analysis (Chizat and Bach 2018; Mei et al. 2018; Mei et al. 2019)
- Double descent (Belkin et al. 2019a; Belkin et al. 2019b), etc.
- Entropy SGD (Chaudhari et al. 2019)
- ...

Numerous Lines of Research on Optimization and Generalization Theory for Deep learning

- Information bottleneck/invariance (Tishby and Zaslavsky 2015; Achille and Soatto 2018)
- Rademacher complexity bounds (Bartlett et al. 2017; Golowich et al. 2018), etc.
- PAC-Bayes bounds (Dziugaite and Roy 2017; Neyshabur et al. 2018a), etc.
- **Neural tangent kernel/Lazy training** (Jacot et al. 2018; Du et al. 2019a; Allen-Zhu et al. 2019b; Du et al. 2019b; Zou et al. 2018; Chizat et al. 2019) etc.
- **Mean-field analysis** (Chizat and Bach 2018; Mei et al. 2018; Mei et al. 2019)
- Double descent (Belkin et al. 2019a; Belkin et al. 2019b), etc.
- Entropy SGD (Chaudhari et al. 2019)
- ...

What is Neural Tangent Kernel?

Training of Deep Neural Networks

- An L-layer ReLU network takes the form

$$f_{\boldsymbol{\theta}}(\mathbf{x}) = \sqrt{m} \cdot \mathbf{W}_L h(\mathbf{W}_{L-1} h(\mathbf{W}_{L-2} \cdots h(\mathbf{W}_1 \mathbf{x}) \cdots)),$$

  $\mathbf{W}_1 \in \mathbb{R}^{m \times d}$, $\mathbf{W}_l \in \mathbb{R}^{m \times m}$ for $l = 2, \ldots, L-1$, $\mathbf{W}_L \in \mathbb{R}^{1 \times m}$, m: width of hidden layers, $h(\cdot)$: activation function, $\boldsymbol{\theta} = \{\mathbf{W}_1, \mathbf{W}_2, \ldots, \mathbf{W}_L\}$: collection of all parameters.

- Given a training set $S = \{\mathbf{x}_j, y_j\}_{j=1}^n$, define the training loss

$$L(\boldsymbol{\theta}) = (2n)^{-1} \sum_{j=1}^{n} [f_{\boldsymbol{\theta}}(\mathbf{x}_j) - y_j]^2$$

- He initialization (He et al. 2015): initialize each parameter from $N(0, 2/m)$

Gradient Descent Dynamics

- Gradient descent

$$\begin{aligned}\boldsymbol{\theta}^{(t+1)} &= \boldsymbol{\theta}^{(t)} - \eta \nabla_{\boldsymbol{\theta}} L(\boldsymbol{\theta}^{(t)}) \\ &= \boldsymbol{\theta}^{(t)} - \frac{\eta}{n} \sum_{j=1}^{n} [f_{\boldsymbol{\theta}^{(t)}}(\mathbf{x}_j) - y_j] \nabla_{\boldsymbol{\theta}} f_{\boldsymbol{\theta}^{(t)}}(\mathbf{x}_j).\end{aligned}$$

  where $\eta > 0$ is step size.

- Local linearization of the neural network along the gradient descent trajectory:

$$\begin{aligned}f_{\boldsymbol{\theta}^{(t+1)}}(\mathbf{x}_i) &\approx f_{\boldsymbol{\theta}^{(t)}}(\mathbf{x}_i) + \langle \nabla f_{\boldsymbol{\theta}^{(t)}}(\mathbf{x}_i), \boldsymbol{\theta}^{(t+1)} - \boldsymbol{\theta}^{(t)} \rangle \\ &= f_{\boldsymbol{\theta}^{(t)}}(\mathbf{x}_i) - \left\langle \nabla f_{\boldsymbol{\theta}^{(t)}}(\mathbf{x}_i), \frac{\eta}{n} \sum_{j=1}^{n} [f_{\boldsymbol{\theta}^{(t+1)}}(\mathbf{x}_j) - y_j] \nabla f_{\boldsymbol{\theta}^{(t)}}(\mathbf{x}_j) \right\rangle\end{aligned}$$

Neural Network Residual Dynamics

- Neural network residual dynamic:

$$\begin{pmatrix} f_{\boldsymbol{\theta}(t+1)}(\mathbf{x}_1) - y_1 \\ \vdots \\ f_{\boldsymbol{\theta}(t+1)}(\mathbf{x}_n) - y_n \end{pmatrix} \approx \begin{pmatrix} f_{\boldsymbol{\theta}(t)}(\mathbf{x}_1) - y_1 \\ \vdots \\ f_{\boldsymbol{\theta}(t)}(\mathbf{x}_n) - y_n \end{pmatrix}$$
$$- (\eta/n) \begin{pmatrix} \langle \nabla f_{\boldsymbol{\theta}(t)}(\mathbf{x}_1), \nabla f_{\boldsymbol{\theta}(t)}(\mathbf{x}_1) \rangle & \cdots & \langle \nabla f_{\boldsymbol{\theta}(t)}(\mathbf{x}_1), \nabla f_{\boldsymbol{\theta}(t)}(\mathbf{x}_n) \rangle \\ \vdots & \vdots & \vdots \\ \langle \nabla f_{\boldsymbol{\theta}(t)}(\mathbf{x}_n), \nabla f_{\boldsymbol{\theta}(t)}(\mathbf{x}_1) \rangle & \cdots & \langle \nabla f_{\boldsymbol{\theta}(t)}(\mathbf{x}_n), \nabla f_{\boldsymbol{\theta}(t)}(\mathbf{x}_n) \rangle \end{pmatrix} \begin{pmatrix} f_{\boldsymbol{\theta}(t)}(\mathbf{x}_1) - y_1 \\ \vdots \\ f_{\boldsymbol{\theta}(t)}(\mathbf{x}_n) - y_n \end{pmatrix}$$

- Matrix form of neural network residual dynamic:

$$\mathbf{f}_{\boldsymbol{\theta}^{(t+1)}} - \mathbf{y} \approx [\mathbf{I} - (m\eta/n)\mathbf{H}_t](\mathbf{f}_{\boldsymbol{\theta}^{(t)}} - \mathbf{y}),$$

where $\mathbf{f}_{\boldsymbol{\theta}} = [f_{\boldsymbol{\theta}}(\mathbf{x}_1), \ldots, f_{\boldsymbol{\theta}}(\mathbf{x}_n)]^\top$, $\mathbf{y} = [y_1, \ldots, y_n]^\top$,
$\mathbf{H}_t = (m^{-1}\langle \nabla f_{\boldsymbol{\theta}^{(t)}}(\mathbf{x}_i), \nabla f_{\boldsymbol{\theta}^{(t)}}(\mathbf{x}_j) \rangle)_{n \times n}$.

Neural Tangent Kernel (Jacot et al. 2018)

Definition of NTK

$$\mathbf{H} = \lim_{m\to\infty} \mathbf{H}_0 := \lim_{m\to\infty} (m^{-1}\langle \nabla f_{\boldsymbol{\theta}^{(0)}}(\mathbf{x}_i), \nabla f_{\boldsymbol{\theta}^{(0)}}(\mathbf{x}_j)\rangle)_{n\times n}$$

is the Gram matrix of NTK.

Neural Tangent Kernel Based Analysis

- Using residual dynamic governed by neural tangent kernel to approximate neural network residual dynamic

$$\mathbf{f}_{\boldsymbol{\theta}^{(t+1)}} - \mathbf{y} \overset{\text{when } \boldsymbol{\theta}^{(t+1)} \text{ is close to } \boldsymbol{\theta}^{(t)}}{\approx} [\mathbf{I} - (m\eta/n)\mathbf{H}_t](\mathbf{f}_{\boldsymbol{\theta}^{(t)}} - \mathbf{y}) \quad \text{(NN dynamic)}$$

$$\overset{\text{when } \boldsymbol{\theta}^{(t)} \text{ is close to } \boldsymbol{\theta}^{(0)}}{\approx} [\mathbf{I} - (m\eta/n)\mathbf{H}_0](\mathbf{f}_{\boldsymbol{\theta}^{(t)}} - \mathbf{y})$$

$$\overset{\text{when } m \text{ is large}}{\approx} [\mathbf{I} - (m\eta/n)\mathbf{H}](\mathbf{f}_{\boldsymbol{\theta}^{(t)}} - \mathbf{y}), \quad \text{(NTK dynamic)}$$

Main Results by Neural Tangent Kernel Analysis

Theorem (NTK type result (informal))

Under certain data assumptions, if

- *the network width* $m \geq \mathrm{poly}(L, n, \epsilon^{-1})$,
- *the step size* $\eta = O(m^{-1})$,

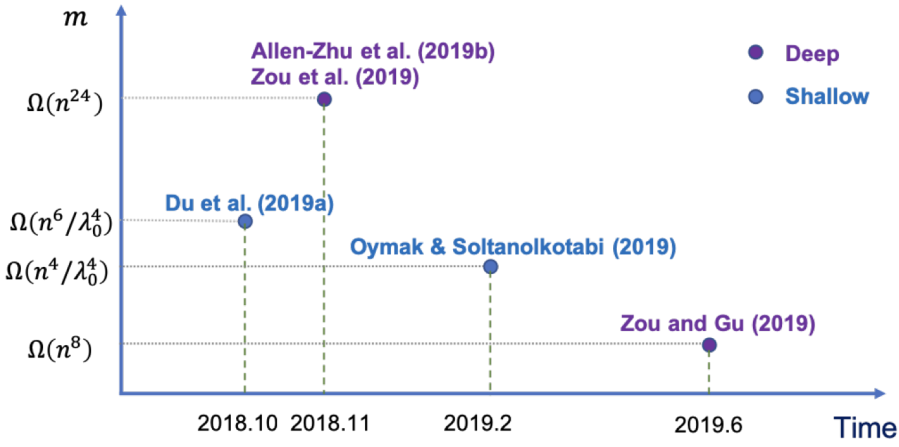
then (stochastic) gradient descent can achieve

- ϵ *training error*
- $\epsilon + \widetilde{O}(n^{-1/2})$ *test error.*

n is the training sample size, L is the depth of neural networks

Reference: Jacot et al. (2018), Du et al. (2019a), Allen-Zhu et al. (2019b), Du et al. (2019b), Zou et al. (2018), Arora et al. (2019a), Arora et al. (2019b), Zou and Gu (2019), Cao and Gu (2019), Ji and Telgarsky (2019), and Chen et al. (2019)...

Comparison of Optimization Results Based on NTK Analysis



n is the training sample size, λ_0 is the smallest eigenvalue of the NTK gram matrix $\mathbf{H}$.

Comparison of Generalization Results Based on NTK Analysis

	Over-para. Condition	Iter. Complexity	Sample Complexity	Network
Allen-Zhu et al. (2019a)	$\mathrm{poly}(n, \epsilon^{-1})$	$\widetilde{O}(\epsilon^{-2})$	$\widetilde{O}(\epsilon^{-4})$	3-Layer
Arora et al. (2019a)	$\mathrm{poly}(n, \epsilon^{-1}, \lambda_0^{-1})$	$\widetilde{O}(n^2\lambda_0^{-2})$	$\widetilde{O}(\epsilon^{-2})$	2-Layer
Cao and Gu (2019)	$\widetilde{\Omega}(\epsilon^{-14}) \cdot \mathrm{poly}(L)$	$\widetilde{O}(\epsilon^{-2}) \cdot \mathrm{poly}(L)$	$\widetilde{O}(\epsilon^{-2}) \cdot \mathrm{poly}(L)$	L-Layer
Ji and Telgarsky (2019)	$\mathrm{polylog}(n, \epsilon^{-1})$	$O(\epsilon^{-1})$	$\widetilde{O}(\epsilon^{-1})$	2-Layer
Chen et al. (2019)	$\mathrm{polylog}(n, \epsilon^{-1}) \cdot \mathrm{poly}(L)$	$O(\epsilon^{-1}) \cdot \mathrm{poly}(L)$	$\widetilde{O}(\epsilon^{-1}) \cdot \mathrm{poly}(L)$	L-Layer

n is the training sample size, L is the depth of neural networks, λ_0 is the smallest eigenvalue of NTK gram matrix.

Best-Known Results by Neural Tangent Kernel Analysis

Theorem (Chen et al. (2019))

Under certain assumptions on the data distribution and using logistic loss, for any $\epsilon > 0$, if $m \geq \Omega(\mathrm{poly}(L) \cdot \mathrm{polylog}(n, \epsilon^{-1}))$, and step size $\eta = \widetilde{O}(m^{-1})$, then with high probability, an L-Layer ReLU networks trained by SGD satisfies

- $L_S(\boldsymbol{\theta}_{\mathrm{SGD}}) \leq \epsilon$.
- $L_{\mathcal{D}}^{0-1}(\boldsymbol{\theta}_{\mathrm{SGD}}) \leq \widetilde{\mathcal{O}}(L^2/n) + \epsilon$

A similar result for two-layer networks is established in Ji and Telgarsky (2019).

Limitation of Neural Tangent Kernel Based Analysis

- Recall that

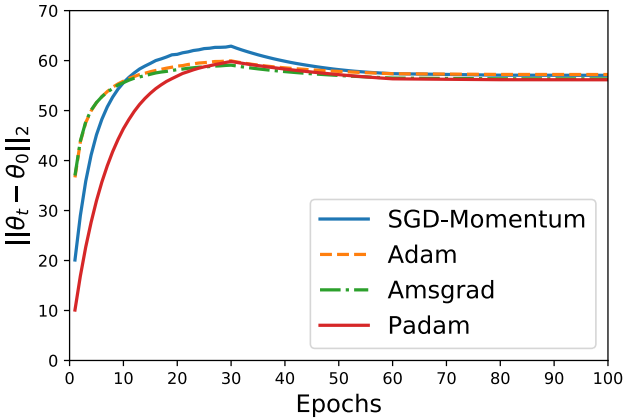
$$\mathbf{f}_{\boldsymbol{\theta}^{(t+1)}} - \mathbf{y} \overset{\text{when } \boldsymbol{\theta}^{(t+1)} \text{ is close to } \boldsymbol{\theta}^{(t)}}{\approx} [\mathbf{I} - (m\eta/n)\mathbf{H}_t](\mathbf{f}_{\boldsymbol{\theta}^{(t)}} - \mathbf{y}) \quad \text{(NN dynamic)}$$

$$\overset{\text{when } \boldsymbol{\theta}^{(t)} \text{ is close to } \boldsymbol{\theta}^{(0)}}{\approx} [\mathbf{I} - (m\eta/n)\mathbf{H}_0](\mathbf{f}_{\boldsymbol{\theta}^{(t)}} - \mathbf{y})$$

$$\overset{\text{when } m \text{ is large}}{\approx} [\mathbf{I} - (m\eta/n)\mathbf{H}](\mathbf{f}_{\boldsymbol{\theta}^{(t)}} - \mathbf{y}), \quad \text{(NTK dynamic)}$$

- Require $\boldsymbol{\theta}^{(t)}$ to be close to $\boldsymbol{\theta}^{(0)}$!!
- How close? $\|\boldsymbol{\theta}^{(t)} - \boldsymbol{\theta}^{(0)}\|_2 = O(m^{-1/2})$.
- This is contradictory to deep learning practice

Limitation of Neural Tangent Kernel Based Analysis



Jinghui Chen and Quanquan Gu. 2018. "Closing the generalization gap of adaptive gradient methods in training deep neural networks". *arXiv preprint arXiv:1806.06763*

How to Overcome the Limitation of Neural Tangent Kernel Based Analysis?

A Mean-field View of Neural Networks Learning

- A two-layer neural network can be seen as an approximation to an infinitely wide neural network, defined based on the distribution of parameters (Mei et al. 2018; Chizat and Bach 2018)

$$f_m(\boldsymbol{\theta}, \mathbf{x}) = \frac{1}{m} \sum_{j=1}^{m} u_j h(\mathbf{w}_j, \mathbf{x})$$

$$\approx f(p, \mathbf{x}) = \int u h(\mathbf{w}, \mathbf{x}) p(\mathbf{w}, u) d\mathbf{w} du$$

  where $\boldsymbol{\theta} = \{\mathbf{w}_j, u_j\}_{j=1}^m$
- Given a training set $S = \{\mathbf{x}_j, y_j\}_{j=1}^n$, define the training loss for infinitely wide neural networks

$$L(p) = \frac{1}{n} \sum_{i=1}^{n} \phi\big(f(p, \mathbf{x}_i), y_i\big) = \widehat{\mathbb{E}}_S[\phi\big((f(p, \mathbf{x}), y\big)].$$

Optimization in Parameter Space v.s. Distribution Space

Distribution space

Energy functional:

$$Q(p) = \widehat{\mathbb{E}}_S[\phi\big(f(p, \mathbf{x}), y\big)] + \underbrace{\lambda D_{\mathrm{KL}}(p \| p_0)}_{\text{KL regularization}}$$

where p_0 is the distribution of $\boldsymbol{\theta}^{(0)}$
Training method: Wasserstein gradient flow

Parameter space

Objective function:

$$\widehat{Q}(\boldsymbol{\theta}) = \widehat{\mathbb{E}}_S[\phi(f_m(\boldsymbol{\theta}, \mathbf{x}), y)] + \underbrace{\frac{\lambda}{2m} \sum_{j=1}^{m} (u_j^2 + \|\mathbf{w}_j\|_2^2)}_{\text{weight decay regularization}}$$

Training method: noisy gradient descent

Main Results in Mean-Field Analysis

Theorem (Mean-field type result (Informal))

Then under certain regularity assumptions, for any $t \geq 0$ and any $\mathbf{x}$,

$$\lim_{m\to\infty} \lim_{\eta\to 0} f_m(\boldsymbol{\theta}_{\lfloor t/\eta \rfloor}, \mathbf{x}) = f(p_t, \mathbf{x}).$$

where η is the step size of noisy gradient descent. In addition, let p^* be the minimizer of $Q(p)$. Then p_t weakly converges to p^*.

Reference: (Mei et al. 2018; Mei et al. 2019; Chizat and Bach 2018)

Neural Tangent Kernel v.s. Mean-field Analysis

Neural tangent kernel scaling

$$f_{\boldsymbol{\theta}}(\mathbf{x}) = \frac{1}{\sqrt{m}} \sum_{j=1}^{m} u_j h(\mathbf{w}_j, \mathbf{x}).$$

- Pros
  - Same scaling as in practice
  - Finite-time convergence rate
  - Generalization error bounds
- Cons
  - Require the parameter $\boldsymbol{\theta}^{(t)}$ to be close to $\boldsymbol{\theta}^{(0)}$.

Mean-field analysis scaling

$$f_m(\boldsymbol{\theta}, \mathbf{x}) = \frac{1}{m} \sum_{j=1}^{m} u_j h(\mathbf{w}_j, \mathbf{x})$$

- Pros
  - Does not require the parameter $\boldsymbol{\theta}^{(t)}$ to be close to $\boldsymbol{\theta}^{(0)}$
  - Can potentially learn a larger class of functions
- Cons
  - Not the same scaling as in practice
  - No finite-time convergence rate
  - No generalization error bounds

Leverage the Advantages of Both NTK and Mean-field Analysis

Consider a scaling factor α (Chizat et al. 2019; Mei et al. 2019; Woodworth et al. 2019):

$$f_m(\boldsymbol{\theta}, \mathbf{x}) = \frac{\alpha}{m} \sum_{j=1}^{m} u_j h(\mathbf{w}_j, \mathbf{x}).$$

NTK regime: $\alpha = \sqrt{m}$; Mean-field regime: $\alpha = 1$.

The corresponding infinitely wide neural network:

$$f(p, \mathbf{x}) = \alpha \int_{\mathbb{R}^{d+1}} u h(\mathbf{w}, \mathbf{x}) p(\mathbf{w}, u) d\mathbf{w} du.$$

Overview of Our Results

With a large scaling parameter α, we

- establish NTK-type optimization/generalization results without requiring the closeness between $\boldsymbol{\theta}^{(t)}$ and $\boldsymbol{\theta}^{(0)}$

With a small scaling parameter α, we

- establish generalization error bounds

	$\alpha = O(1)$	$\alpha \gg 1$
Optimization	Mei et al. (2018), Chizat and Bach (2018) Fang et al. (2019)	Mei et al. (2019), Chizat et al. (2019) **This work**
Generalization	**This work**	**This work**

Distribution Dynamics

- We consider the partial differential equation

$$\frac{dp_t(\mathbf{w},u)}{dt} = -\nabla_u[p_t(\mathbf{w},u)g_1(t,\mathbf{w},u)] - \nabla_{\mathbf{w}}\cdot[p_t(\mathbf{w},u)g_2(t,\mathbf{w},u)] + \lambda\Delta[p_t(\mathbf{w},u)], \qquad (1)$$

  where

$$g_1(t,\mathbf{w},u) = -\alpha\widehat{\mathbb{E}}_S[\nabla_{y'}\phi(f(p_t,\mathbf{x}),y)h(\mathbf{w},\mathbf{x})] - \lambda u/\sigma_u^2,$$
$$g_2(t,\mathbf{w},u) = -\alpha\widehat{\mathbb{E}}_S[\nabla_{y'}\phi(f(p_t,\mathbf{x}),y)u\nabla_{\mathbf{w}}h(\mathbf{w},\mathbf{x})] - \lambda\mathbf{w}/\sigma_{\mathbf{w}}^2.$$

- The solution to PDE (1) minimizes the energy functional as t goes to infinity

Energy functional:

$$Q(p) = \widehat{\mathbb{E}}_S[\phi\big(f(p,\mathbf{x}),y\big)] + \underbrace{\lambda D_{\mathrm{KL}}(p\|p_0)}_{\text{KL regularization}}$$

Optimization result when $\alpha \gg 1$

Theorem

Assume that the activation function $h(\mathbf{w}, \mathbf{x})$ satisfies certain smoothness properties. Let $\lambda_0 = \lambda_{\min}(\mathbf{H})/n$. If $\alpha \geq \mathrm{poly}(\lambda_0^{-1})$, then for all $t \in [0, +\infty)$, the following results hold:

$$L(p_t) \leq 2\exp(-2\alpha^2\lambda_0 t)L(p_0) + O(\lambda^2\alpha^{-2}\lambda_0^{-2})$$
$$D_{\mathrm{KL}}(p_t||p_0) \leq O(\alpha^{-2}\lambda_0^{-2}),$$

$\mathbf{H} = \mathbf{H}(p_0)$ is the Gram matrix of neural tangent kernel.

Linear Convergence Up to $O(\alpha^{-2})$. Convergence rate depends on the smallest eigenvalue of NTK Gram matrix!

Revisiting NTK Approximation

Define $\mathbf{f}(t) = (f(p_t, \mathbf{x}_1), \ldots, f(p_t, \mathbf{x}_n))^\top$, and let $\mathbf{f}_{\mathsf{NTK}}(t) \in \mathbb{R}^n$ be the solution of ODEs

$$\frac{d[\mathbf{f}_{\mathsf{NTK}}(t) - \mathbf{y}]}{dt} = -\frac{2\alpha^2}{n}\mathbf{H}[\mathbf{f}_{\mathsf{NTK}}(t) - \mathbf{y}], \quad \mathbf{f}_{\mathsf{NTK}}(0) = \mathbf{0}.$$

Corollary

For any time $t \in [0, +\infty)$,

$$\|\mathbf{H}(p_t) - \mathbf{H}\|_{\infty,\infty} \leq \widetilde{O}(\alpha^{-1}\lambda_0^{-1}),$$
$$\frac{1}{n}\|\mathbf{f}(t) - \mathbf{f}_{NTK}(t)\|_2^2 \leq \widetilde{O}(\alpha^{-2}\lambda_0^{-4}).$$

Neural network training with weight decay regularization and gradient noises can still have similar dynamics to neural tangent kernel!

Generalization Bounds when $\alpha \gg 1$

Theorem

Suppose that the target function has the form

$$y = \int uh(\mathbf{w}, \mathbf{x}) p_{\textit{true}}(\mathbf{w}, u) d\mathbf{w} du.$$

Then as long as α is large enough, with high probability,

$$\mathbb{E}_{\mathcal{D}}[\ell^{0\text{-}1}(f(p^*, \mathbf{x}), y)] \leq \widetilde{O}\left(\sqrt{\frac{D_{\chi^2}(p_{\text{true}} || p_0)}{n}}\right).$$

When α is large, two-layer neural networks trained by NGD can learn

$$\mathcal{F}_{\chi^2} = \left\{ \int uh(\mathbf{w}, \mathbf{x}) p(\mathbf{w}, u) d\mathbf{w} du : D_{\chi^2}(p || p_0) < \infty \right\}$$

Generalization Bounds when $\alpha = O(1)$

Theorem

Suppose that $h(\mathbf{w}, \mathbf{x})$ is bounded, and the target function has the form

$$y = \int u h(\mathbf{w}, \mathbf{x}) p_{true}(\mathbf{w}, u) d\mathbf{w} du$$

Then for appropriately chosen λ, with high probability,

$$\mathbb{E}_{\mathcal{D}}[\ell^{0\text{-}1}(f(p^*, \mathbf{x}), y)] \leq \widetilde{O}\left(\sqrt{\frac{D_{KL}(p_{true}||p_0)}{n}}\right).$$

When $\alpha = O(1)$, two-layer neural networks trained by NGD can learn

$$\mathcal{F}_{\mathsf{KL}} = \left\{ \int u h(\mathbf{w}, \mathbf{x}) p(\mathbf{w}, u) d\mathbf{w} du : D_{\mathsf{KL}}(p||p_0) < \infty \right\}$$

Generalization Bounds: $\alpha = O(1)$ v.s. $\alpha \gg 1$

When α is large, two-layer neural networks trained by NGD can learn

$$\mathcal{F}_{\chi^2} = \left\{ \int uh(\mathbf{w}, \mathbf{x}) p(\mathbf{w}, u) d\mathbf{w} du : D_{\chi^2}(p||p_0) < \infty \right\};$$

When $\alpha = O(1)$, two-layer neural networks trained by NGD can learn

$$\mathcal{F}_{\mathsf{KL}} = \left\{ \int uh(\mathbf{w}, \mathbf{x}) p(\mathbf{w}, u) d\mathbf{w} du : D_{\mathsf{KL}}(p||p_0) < \infty \right\}.$$

One can check $\mathcal{F}_{\chi^2} \subsetneq \mathcal{F}_{\mathsf{KL}}$. The regime with small α may outperform the one with large α?

Summary

- Poly-logarithmic width is sufficient to learn a deep neural work in the NTK setting
- Closeness in distribution instead of parameter is sufficient to derive NTK-type results.

Thank you!