On the Generalization Power of Overfitted Two-Layer Neural Tangent Kernel Models

June 3, 2022

Peizhong Ju

Postdoc, ECE

This is a joint work with Prof. Xiaojun Lin (Purdue) and Prof. Ness Shroff (OSU)

Outline

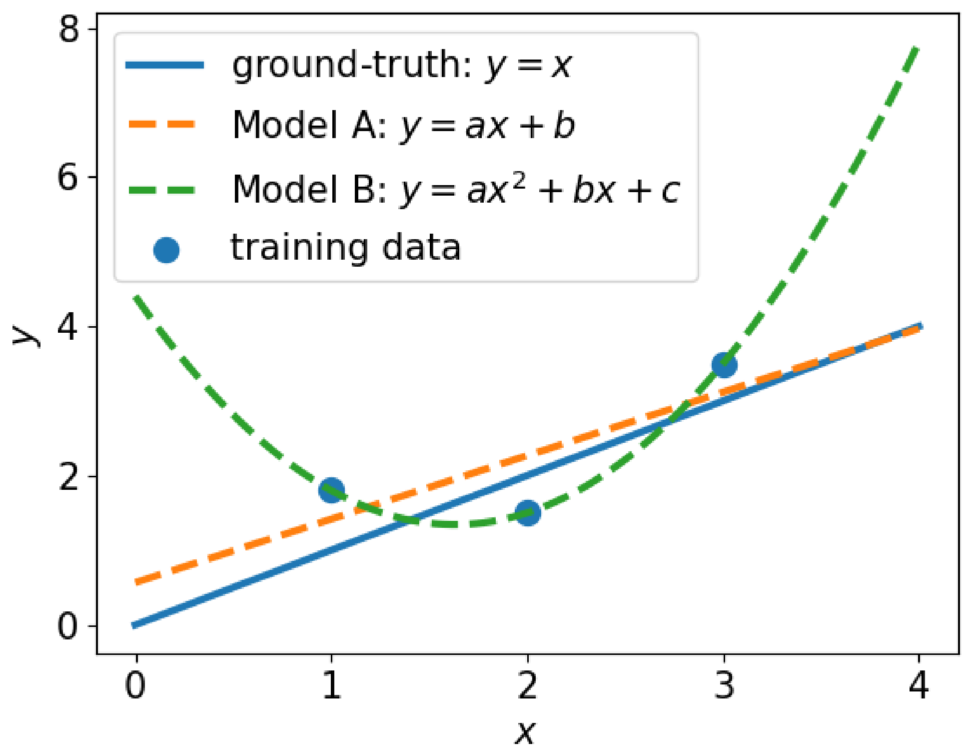
- Background knowledge
- Motivation
- Neural tangent kernel (NTK) models
- Related work
- Problem setup
- Main results
- Conclusion

Background knowledge

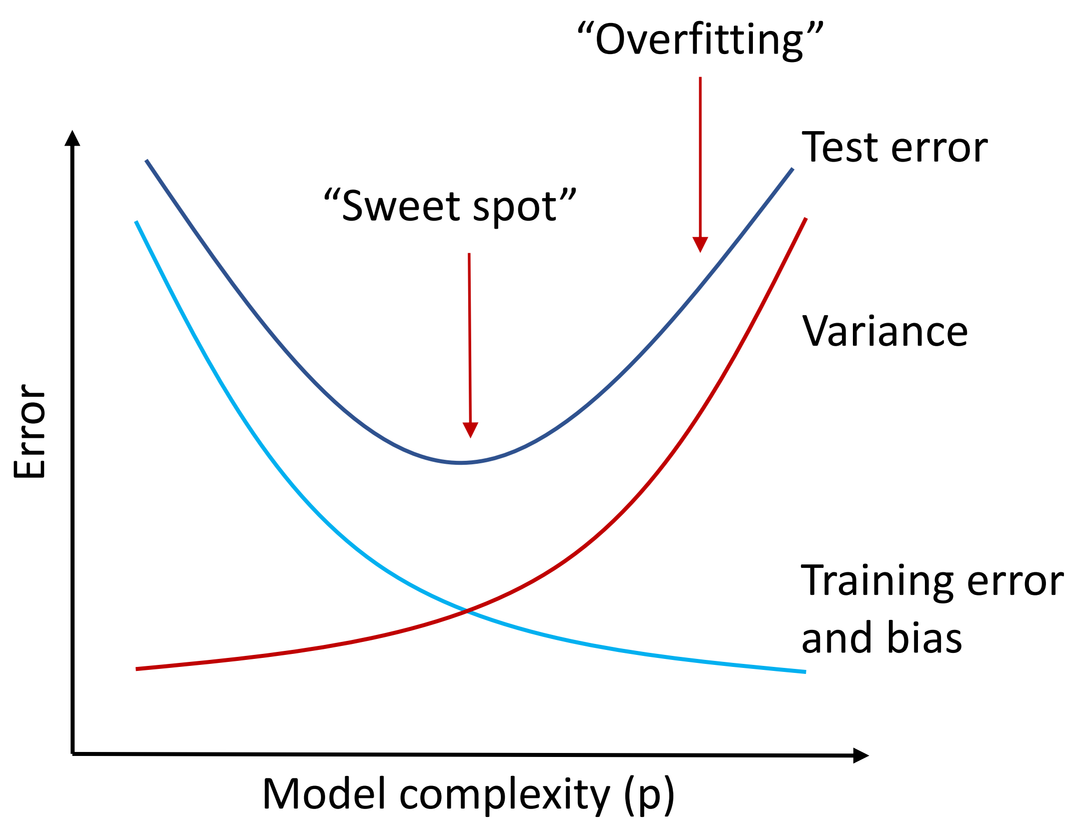
- Supervised learning
  - Given *n* training data/samples
    - Generated by a ground-truth function and noise
  - Determine values of parameters of a model to fit those training data (e.g, training by gradient descent)
    - Training error: how well the model fit all training data
  - Test error: evaluate the performance on unseen test data
- Overfitting & overparameterization
  - When a model has enough many parameters, it can completely fit all training data, i.e., training error is zero
- Overfitted linear regression models $y = \boldsymbol{x}^T \widehat{\boldsymbol{\beta}}$
  - Matrix equation with *n* samples: $\mathbf{Y} = \mathbf{X}^T \widehat{\boldsymbol{\beta}} \in \mathbf{R}^n$
  - Every element of $\widehat{\boldsymbol{\beta}}$ is a parameter (determined by training)
  - Every element of $\boldsymbol{x}$ is a feature: $\boldsymbol{x} \in \mathbf{R}^p$ (*p* is the num of features)
    - Gaussian feature: every element of $\boldsymbol{x}$ follows *i.i.d.* Gaussian distribution
    - Fourier feature: $\boldsymbol{x}^T = [1 \quad \cos\theta \quad \sin\theta \quad \cos 2\theta \quad \sin 2\theta \quad \cdots]$
  - Min l2-norm solution: $\min \|\hat{\beta}\|_2 \quad \text{subject to } \mathbf{Y} = \mathbf{X}^T \hat{\beta}$

A simple example of overfitting

- A ground-truth function: $f(x) = x$
- Three training data (with noise): (1, 1.8), (2, 1.5), (3, 3.5)
- Model A with 2 parameters: $\hat{f}(x) = ax + b$
- Model B with 3 parameters: $\hat{f}(x) = ax^2 + bx + c$
- Training by minimize mean-square-error (MSE)



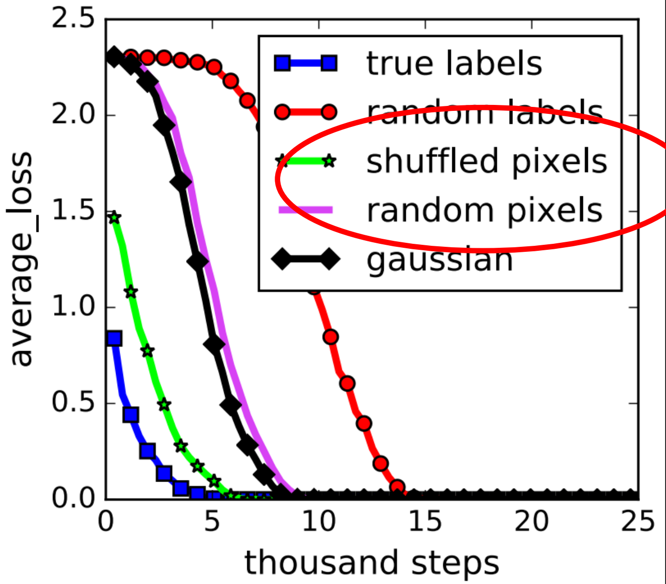
The bias-variance tradeoff



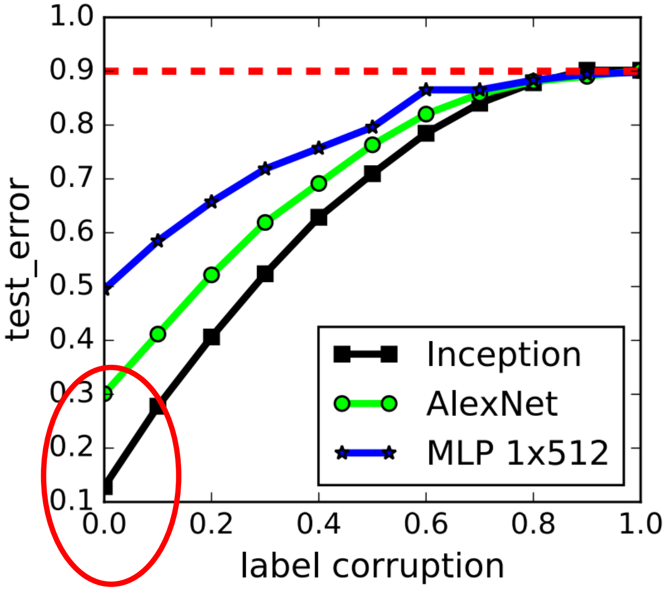
- "Sweet spot" is usually achieved by regularization (e.g. LASSO and ridge regression) to ensure that overfitting does not occur

Overfitting is usually harmful, however…

- Deep neural networks (DNNs) can generalize well even when they overfit the training data [Zhang et al, 2017]



Training error with perturbed/noisy data

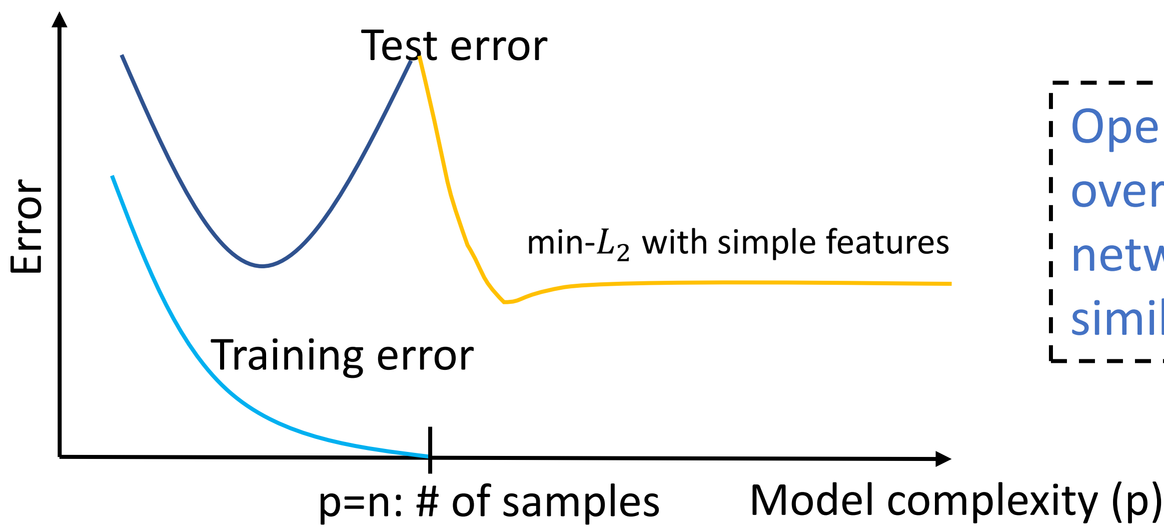


Test error

Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. (2017). "Understanding deep learning requires rethinking generalization." ICLR 2017.

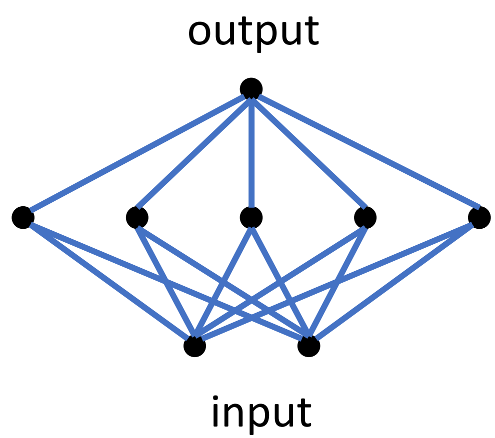
Motivation

- Why can DNNs generalize well when heavily overparameterized [Zhang et al, 2017]?
- Recent attempts: "double descent" for linear regression with simple features (e.g., Gaussian and Fourier features) [See, e.g., Belkin et al., 2018, 2019; Bartlett et al., 2019; Hastie et al., 2019; Mei and Montanari, 2019; Muthukumar et al., 2019]
  - double descent: test error descends again in the overparameterized regime



Open question: do overparameterized neural networks also experience a similar descent behavior?

Neural Tangent Kernel (NTK) model (Jacot et al., 2018)



Fix top-layer weights

ReLU (Rectifier Linear Unit): $\max(0, x)$

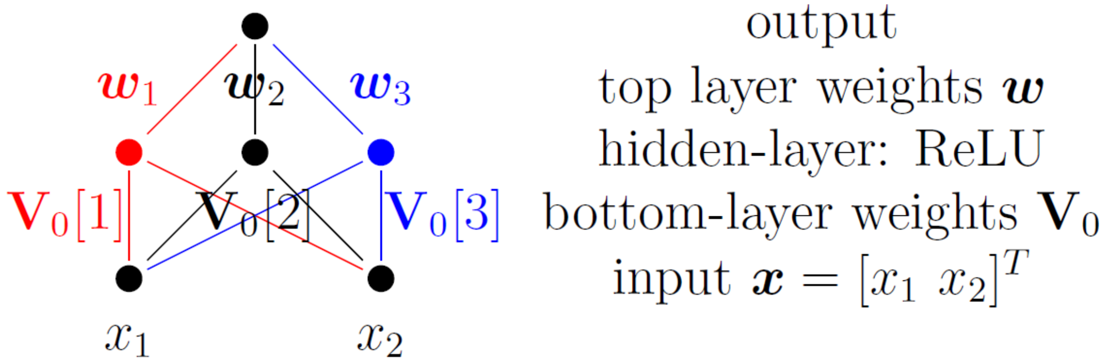
Only train bottom-layer weights

- NTK: a linear approximation of neural network
  - For such a wide and fully-connected two-layer neural network, both the weights and activation patterns do not change much after gradient descent training with a small step size [Li & Liang, 2018; Du et al., 2018]
  - Features of NTK model are formed by the nonlinear activation function.
- We study the descent behavior of min-l_2-norm solutions.
- More details will be discussed later in the problem setup.

Related work

- Generalization error of "random feature" (RF) model where *p*, *n*, and *d* grow proportionally to infinity [e.g., Mei & Montanari, 2019; d'Ascoli et al., 2020]
  - The RF model only trains top layer weights.
  - Only linear functions can be learned.
  - We are interested in the situation that *p>>n.*
- Expressiveness of NTK and RF model [e.g., Ghorbani et at., 2019]
  - Can approximate highly non-linear ground-truth functions with sufficiently many neurons.
  - Cannot characterize the generalization performance.
- Overfitted generalization error of NTK (similar to our setting) [e.g., Arora et al., 2019; Satpathi & Srikant, 2021; Fiat et at., 2019]
  - Provide an upper bound when *p* is larger than a threshold
  - Does not contain *p* in the expression of their upper bound, thus cannot characterize the descent behavior
- Other related settings: NTK without overfitting [e.g., Allen-Zhu et at., 2019], classification by NTK [e.g., Ji & Telgarsky, 2019]
  - Different from our setting where we consider overfitted solutions for regression.

Problem setup



An example of 2-layer NN where num of neurons *p*=3, input dim *d*=2

Change of the output after training:

similar when $\overline{\Delta \mathbf{V}}$ is small.
NTK assumes they are the same.

$$\sum_{j=1}^{p} \boldsymbol{w}_j \mathbf{1}_{\{\boldsymbol{x}^T(\mathbf{V}_0[j]+\overline{\Delta\mathbf{V}}[j])>0\}} \cdot (\mathbf{V}_0[j] + \overline{\Delta\mathbf{V}}[j])^T \boldsymbol{x} - \sum_{j=1}^{p} \boldsymbol{w}_j \mathbf{1}_{\{\boldsymbol{x}^T\mathbf{V}_0[j]>0\}} \cdot \mathbf{V}_0[j]^T \boldsymbol{x}$$

p neurons

top layer weights

j-th neuron

activation pattern by ReLU

initial bottom layer weights

change of bottom layer weights

output of after training

initial output

Problem setup (Cont'd)

- After training, the change of the output can be approximated by a linear model

top layer weights — change of bottom layer weights

$$\sum_{j=1}^{p} \boldsymbol{w}_j \mathbf{1}_{\{\boldsymbol{x}^T \mathbf{V}_0[j]>0\}} \cdot \overline{\Delta \mathbf{V}[j]}^T \boldsymbol{x}$$

activation pattern of initial state

Rewrite it as $$\hat{f}_{\Delta\mathbf{V},\mathbf{V}_0}(\boldsymbol{x}) := \boldsymbol{h}_{\mathbf{V}_0,\boldsymbol{x}} \Delta\mathbf{V}.$$

feature vector

The feature vector is very different from *i.i.d.* Gaussian or Fourier features.

design matrix formed by n samples: $\mathbf{H} \in \mathbb{R}^{n\times(dp)}$

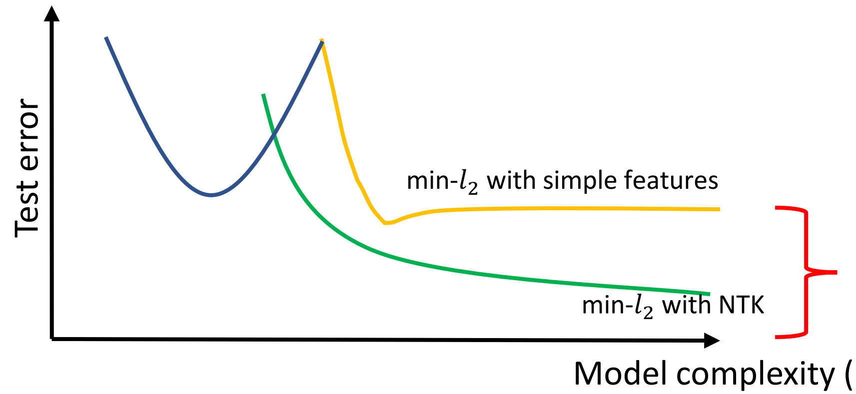
Min-l_2-norm solution: $$\Delta\mathbf{V}^{\ell_2} := \arg\min_{\boldsymbol{v}} \|\boldsymbol{v}\|_2, \text{ subject to } \mathbf{H}\boldsymbol{v} = \boldsymbol{y}.$$

$$\Delta\mathbf{V}^{\ell_2} = \mathbf{H}^T(\mathbf{H}\mathbf{H}^T)^{-1}\boldsymbol{y}$$

Trained model: $$\hat{f}^{\ell_2}(\boldsymbol{x}) := \boldsymbol{h}_{\mathbf{V}_0,\boldsymbol{x}} \Delta\mathbf{V}^{\ell_2}.$$

Ground-truth data model
$y = f(\boldsymbol{x}) + \epsilon, \;\; \boldsymbol{x} \in \mathbb{R}^d$

Main Results



Considering the following class of ground-truth functions:

$$\mathcal{F}^{\ell_2} := \left\{ f_g(\boldsymbol{x}) = \int_{\mathcal{S}^{d-1}} \boldsymbol{x}^T \boldsymbol{z} \frac{\pi - \arccos(\boldsymbol{x}^T \boldsymbol{z})}{2\pi} g(\boldsymbol{z}) d\mu(\boldsymbol{z}) \right\}$$

$\mu(\cdot)$ is the uniform distribution on the unit sphere S^{d-1}
$g(\cdot)$ is any function whose norm is bounded

We provide an upper bound on the test error for finite *p* (num of neurons)

$$|\hat{f}^{\ell_2}(\boldsymbol{x}) - f(\boldsymbol{x})| = O\left(\frac{1}{\sqrt{n}}\right) + \mathrm{poly}_1(n,d) \cdot O\left(\frac{1}{\sqrt{p}}\right) + \mathrm{poly}_2(n,d) \cdot \text{noise level}$$

num of training data — num of neurons

When n is larger and noise level is low, the test error decreases to a very small value as *p* increases num of training data — num of neurons — noise level

$$\Pr_{\mathbf{V}_0,\mathbf{X}}\left\{|\hat{f}^{\ell_2}(\boldsymbol{x})-f(\boldsymbol{x})|\geq n^{-\frac{1}{2}(1-\frac{1}{q})}+\left(1+\sqrt{J_m(n,d)n}\right)p^{-\frac{1}{2}(1-\frac{1}{q})}+\sqrt{J_m(n,d)n}\|\boldsymbol{\epsilon}\|_2 \text{ for all } \boldsymbol{\epsilon}\in\mathbb{R}^n\right\}$$

$$\leq 2e^2\left(\exp\left(-\frac{\sqrt[q]{n}}{8\|g\|_\infty^2}\right)+\exp\left(-\frac{\sqrt[q]{p}}{8\|g\|_1^2}\right)+\exp\left(-\frac{\sqrt[q]{p}}{8n\|g\|_1^2}\right)\right)+\frac{4}{\sqrt[m]{n}}.$$

$q \geq 1$ can be any large number, $J_m(n,d)$ is a function of n and d.

- The first result in the literature showing a descent curve as a function of p for the NTK model
- Also the first result characterizing the speed of descent
- Prior results in the literature [e.g., Arora et al.,2019] only show an upper bound only for p above a threshold, and thus are independent on p

For comparison [Arora et al.,2019]:

$$\Pr\left\{\mathop{\mathrm{E}}_{\boldsymbol{x}}|\hat{f}(\boldsymbol{x})-f(\boldsymbol{x})|\leq\sqrt{\frac{2\boldsymbol{y}^T(\mathbf{H}^\infty)^{-1}\boldsymbol{y}}{n}}+O\left(\sqrt{\frac{\log\frac{n}{\zeta\cdot\min\mathrm{eig}(\mathbf{H}^\infty)}}{n}}\right)\right\}\geq 1-\zeta$$

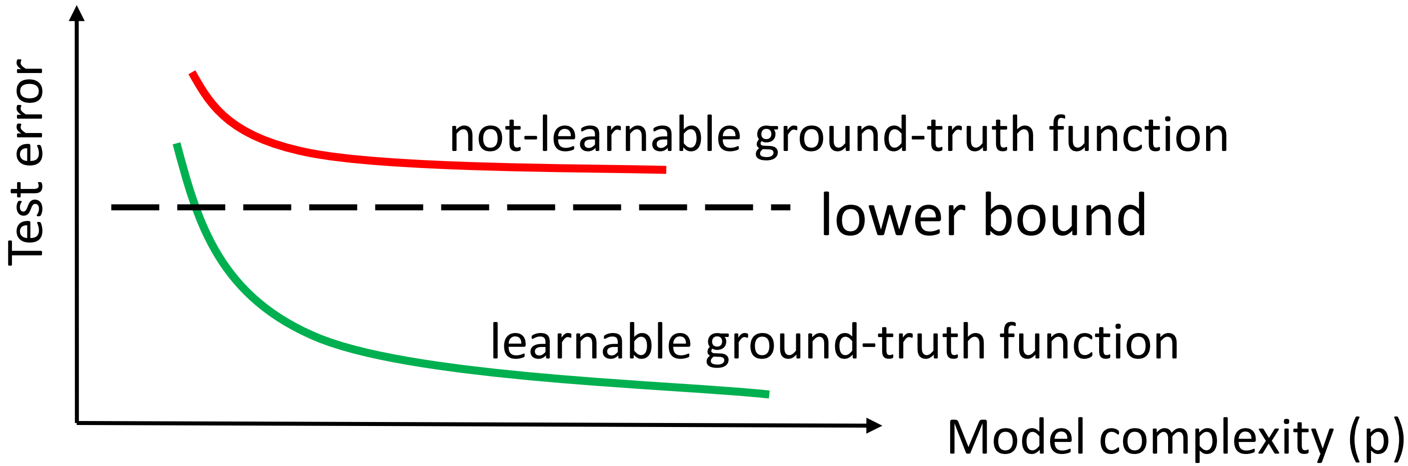
- The descent of NTK is very different from that of linear models with simple features

[Belkin et al.,2019]:

$$\mathrm{MSE}=\|f\|_2^2\left(1-\frac{n}{p}\right)+\frac{\sigma^2 n}{p-n-1}, \quad \text{for } p\geq n+2$$

increase as p (num of feature) increases

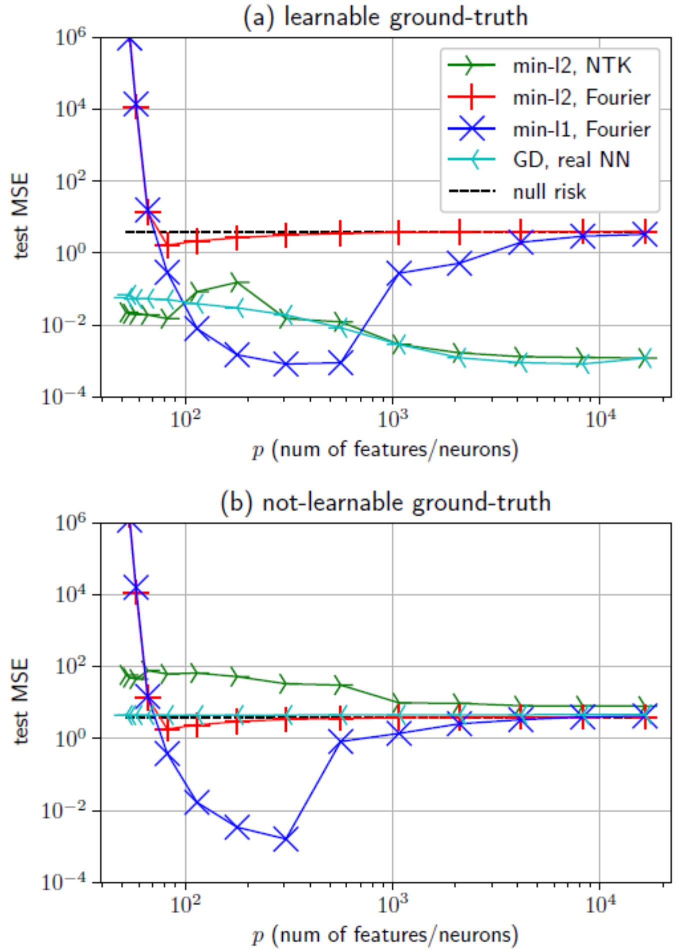
Ground-Truth Functions: Learnable or Not



- The descent of the NTK model critically depends on the ground-truth function.
- For the specific NTK model in our work (ReLU without bias), we provide a lower bound on the generalization error for not-learnable functions

If the ground-truth function $f \notin \overline{\mathcal{F}^{\ell_2}}$ (or equivalently, $D(f, \mathcal{F}^{\ell_2}) > 0$), then the MSE of $\hat{f}_\infty^{\ell_2}$ (with respect to the ground-truth function f) is at least $D(f, \mathcal{F}^{\ell_2})$.

Simulation result



$$f(\theta) = \sum_{k \in \{0,1,2,4\}} (\sin(k\theta) + \cos(k\theta))$$

Learnable ground-truth function: corresponds to linear and even power polynomials

$$f(\theta) = \sum_{k \in \{3,5,7,9\}} (\sin(k\theta) + \cos(k\theta))$$

Not-learnable ground-truth function: corresponds to odd polynomials

What exactly are the functions in $\mathcal{F}^{\ell_2}$?

$$\mathcal{F}^{\ell_2} := \left\{ f_g(\boldsymbol{x}) = \int_{\mathcal{S}^{d-1}} \boldsymbol{x}^T \boldsymbol{z} \frac{\pi - \arccos(\boldsymbol{x}^T \boldsymbol{z})}{2\pi} g(\boldsymbol{z}) d\mu(\boldsymbol{z}) \right\}$$

Rewrite $f_g \in \mathcal{F}^{\ell_2}$ as a convolution

$$f_g(\boldsymbol{x}) = g \circledast h(\boldsymbol{x}) := \int_{\mathsf{SO}(d)} g(\mathbf{S}\boldsymbol{e}) h(\mathbf{S}^{-1}\boldsymbol{x}) d\mathbf{S}$$

$$h(\boldsymbol{x}) := \boldsymbol{x}^T \boldsymbol{e} \frac{\pi - \arccos(\boldsymbol{x}^T \boldsymbol{e})}{2\pi}$$

Important property: convolution corresponds to multiplication in the frequency domain

$$c_{f_g}(l, \mathbf{K}) = \Lambda \cdot c_g(l, \mathbf{K}) c_h(l, \mathbf{0})$$

[Dokmanic & Petrinovic, 2009]

(similar to Fourier coefficients)

h as a "filter" or "channel"

We prove that:

$c_h(l, \mathbf{0})$ is zero for $l = 3, 5, 7, \cdots$ and is non-zero for $l = 0, 1, 2, 4, 6, \cdots$.

linear and even-power polynomials (e.g., $f(\boldsymbol{x}) = (\boldsymbol{a}^T \boldsymbol{x})^4$) are learnable (consistent with [Arora et al.,2019]), while other odd-power polynomials (e.g., $f(\boldsymbol{x}) = (\boldsymbol{a}^T \boldsymbol{x})^3$) are not.

Caution: the set of learnable functions depends on the model structure

- [Satpathi & Srikant, 2021] shows that if bias is considered in ReLU, then both even-power and odd-power polynomials are learnable.

ReLU without bias: $\max(0, x)$
ReLU with bias: $\max(\text{bias}, x)$

- Although our setting does not include bias in ReLU, we can easily derive similar conclusions when bias in ReLU is considered.
- Adding a bias is equivalent to fixing the last element of input $\boldsymbol{x}$ by a constant.
- Even though only a subset of functions are learnable in the *d*-dim space, when projected into a *(d-1)*-dim subspace, they may already span all functions.

Conclusion

- We give an upper bound of the generalization error of min-l_2-norm overfitting solutions for 2-layer NTK, which is the first known result to characterize the descent curve as a function of p for the NTK model.
- We show that the descent behavior of NTK is different from that of linear models with simple features (such as Gaussian and Fourier features).
- We show that the descent behavior critically depends on the ground-truth functions and the model structure.