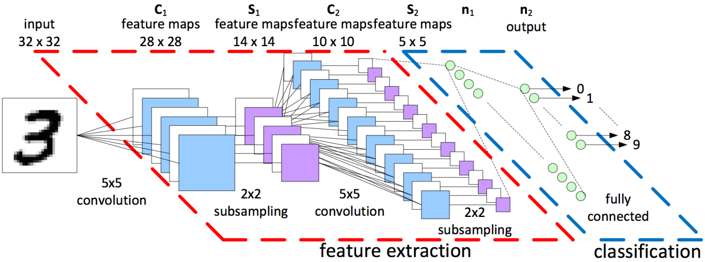
Linearized two-layers neural network in high dimensions

Song Mei

Stanford University

May 26, 2019

Joint work with Andrea Montanari, Theodor Misiakiewicz, Behrooz Ghorbani


C1
S1
C2
S2
n1
n2
input
feature maps
feature mapsfeature mapsfeature maps
output
32 x 32
28 x 28
14 x 14
10 x 10
5 x 5
0
1
8
9
5x5
convolution
2x2
subsampling
5x5
convolution
2x2
subsampling
fully
connected
feature extraction
classification

$$R(\Theta) = \min_{\Theta} \mathbb{E}[\ell(y, \boldsymbol{W}_1 \boldsymbol{\sigma} \circ \boldsymbol{W}_2 \circ \boldsymbol{\sigma} \circ \cdots \circ \boldsymbol{W}_k \circ \boldsymbol{x})].$$

Empirical surprise of neural network [Zhang *et al.*, 2016]

- Over-parameterized regime.
- Optimization surprise: efficiently fit all the data.
- Generalization surprise: generalize well.

Two-layers neural network

$$\hat{f}_N(x;\Theta) = \sum_{i=1}^{N} a_i \sigma(\langle w_i, x \rangle), \quad \Theta = (a_1, w_1, \ldots, a_N, w_N).$$

- Feature $x \in \mathbb{R}^d$.
- Bottom layer weights $w_i \in \mathbb{R}^d$, $i = 1, 2, \ldots, N$.
- Top layer weights $a_i \in \mathbb{R}$, $i = 1, 2, \ldots, N$.
- Over-parametrization: N large.

Two-layers neural network

$$\hat{f}_N(\boldsymbol{x};\boldsymbol{\Theta}) = \sum_{i=1}^{N} a_i \sigma(\langle \boldsymbol{w}_i, \boldsymbol{x} \rangle), \quad \boldsymbol{\Theta} = (a_1, \boldsymbol{w}_1, \ldots, a_N, \boldsymbol{w}_N).$$

- Feature $\boldsymbol{x} \in \mathbb{R}^d$.
- Bottom layer weights $\boldsymbol{w}_i \in \mathbb{R}^d$, $i = 1, 2, \ldots, N$.
- Top layer weights $a_i \in \mathbb{R}$, $i = 1, 2, \ldots, N$.
- Over-parametrization: N large.

Two-layers neural network

$$\hat{f}_N(x; \Theta) = \sum_{i=1}^{N} a_i \sigma(\langle w_i, x \rangle), \quad \Theta = (a_1, w_1, \ldots, a_N, w_N).$$

- Feature $x \in \mathbb{R}^d$.
- Bottom layer weights $w_i \in \mathbb{R}^d$, $i = 1, 2, \ldots, N$.
- Top layer weights $a_i \in \mathbb{R}$, $i = 1, 2, \ldots, N$.
- Over-parametrization: N large.

Two-layers neural network

$$\hat{f}_N(\boldsymbol{x}; \boldsymbol{\Theta}) = \sum_{i=1}^{N} a_i \sigma(\langle \boldsymbol{w}_i, \boldsymbol{x} \rangle), \quad \boldsymbol{\Theta} = (a_1, \boldsymbol{w}_1, \ldots, a_N, \boldsymbol{w}_N).$$

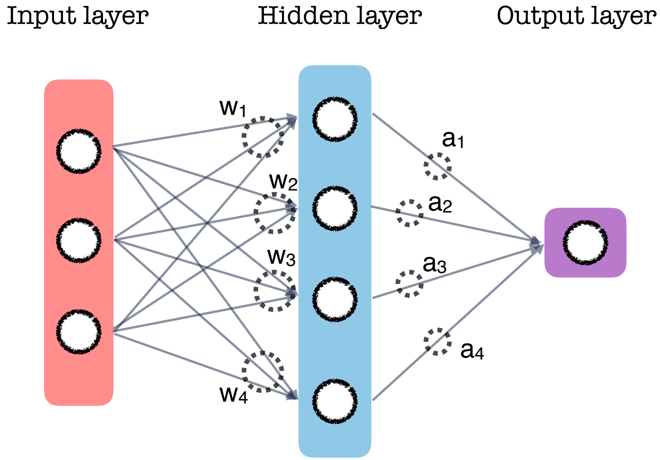
- Feature $\boldsymbol{x} \in \mathbb{R}^d$.
- Bottom layer weights $\boldsymbol{w}_i \in \mathbb{R}^d$, $i = 1, 2, \ldots, N$.
- Top layer weights $a_i \in \mathbb{R}$, $i = 1, 2, \ldots, N$.
- Over-parametrization: N large.

Two-layers neural network

$$\hat{f}_{N}(\boldsymbol{x};\boldsymbol{\Theta}) = \sum_{i=1}^{N} a_i \sigma(\langle \boldsymbol{w}_i, \boldsymbol{x} \rangle), \quad \boldsymbol{\Theta} = (a_1, \boldsymbol{w}_1, \ldots, a_N, \boldsymbol{w}_N).$$

- Feature $\boldsymbol{x} \in \mathbb{R}^d$.
- Bottom layer weights $\boldsymbol{w}_i \in \mathbb{R}^d$, $i = 1, 2, \ldots, N$.
- Top layer weights $a_i \in \mathbb{R}$, $i = 1, 2, \ldots, N$.
- Over-parametrization: N large.

Two-layers neural network



Gradient flow with random initialization

Empirical risk: (n: # data; N: # neuron)

$$R_{n,N}(\boldsymbol{\Theta}) = \hat{\mathbb{E}}_{\boldsymbol{x},n}[(y - \hat{f}_N(\boldsymbol{x}; \boldsymbol{\Theta}))^2]$$

Gradient flow, on empirical risk, with random initialization:

$$\dot{\boldsymbol{\Theta}}(t) = -\nabla R_{n,N}(\boldsymbol{\Theta}(t)),$$
$$(a_i(0), \boldsymbol{w}_i(0)) \sim_{i.i.d.} \mathbb{P}_{a,\boldsymbol{w}}.$$

Convergence guarantees

Lemma (Global min. Not surprise.)

For $N > n$, we have

$$\inf_{\boldsymbol{\Theta}} R_{n,N}(\boldsymbol{\Theta}) = 0.$$

There are many global minimizer with empirical risk 0.

Convergence guarantees

Lemma (Global min. Not surprise.)

For $N > n$, we have

$$\inf_{\boldsymbol{\Theta}} R_{n,N}(\boldsymbol{\Theta}) = 0.$$

There are many global minimizer with empirical risk 0.

But there are also local minimizers with non-zero risk.

Convergence guarantees

Lemma (Global min. Not surprise.)

For $N > n$, *we have*

$$\inf_{\boldsymbol{\Theta}} R_{n,N}(\boldsymbol{\Theta}) = 0.$$

There are many global minimizer with empirical risk 0.

But there are also local minimizers with non-zero risk.

Theorem (The optimization surprise.)

For $N \gg n^{1+c}$, *we have*

$$\lim_{t \to \infty} R_{n,N}(\boldsymbol{\Theta}(t)) = 0,$$

i.e., training loss converges to 0.

Convergence guarantees

Lemma (Global min. Not surprise.)

For $N > n$, we have

$$\inf_{\boldsymbol{\Theta}} R_{n,N}(\boldsymbol{\Theta}) = 0.$$

There are many global minimizer with empirical risk 0.

But there are also local minimizers with non-zero risk.

Theorem (The optimization surprise.)

For $N \gg n^{1+c}$, we have

$$\lim_{t \to \infty} R_{n,N}(\boldsymbol{\Theta}(t)) = 0,$$

i.e., training loss converges to 0.

Under what assumptions?

Three variants of the convergence theorem

Gradient flow (n: # data; N: # neuron):

$$\dot{\boldsymbol{\Theta}}(t) = -\nabla \hat{\mathbb{E}}_{\boldsymbol{x},n}[(y - \hat{f}_N(\boldsymbol{x}; \boldsymbol{\Theta}(t)))^2],$$
$$\boldsymbol{w}_i(0) \sim_{i.i.d.} \mathcal{N}(\mathbf{0}, \mathbf{I}_d/d).$$

Theorem: for N large enough, we have

$$\lim_{t\to\infty} R_{n,N}(\boldsymbol{\Theta}(t)) = 0.$$

Random feature (RF) regime

$$\hat{f}_N(\boldsymbol{x}; \boldsymbol{\Theta}) = \sum_{i=1}^{N} a_i \sigma(\langle \boldsymbol{w}_i, \boldsymbol{x} \rangle), \quad a_i(0) \sim_{i.i.d.} \mathcal{N}(0, 1/N^2).$$

[Andoni *et al.*, 2014], [Danialy, 2017], [Yehudai and Shamir, 2019] ...

Three variants of the convergence theorem

Gradient flow (n: # data; N: # neuron):

$$\dot{\Theta}(t) = -\nabla \hat{\mathbb{E}}_{x,n}[(y - \hat{f}_N(x; \Theta(t)))^2],$$
$$w_i(0) \sim_{i.i.d.} \mathcal{N}(\mathbf{0}, \mathbf{I}_d/d).$$

Theorem: for N large enough, we have

$$\lim_{t \to \infty} R_{n,N}(\Theta(t)) = 0.$$

Neural tangent (NT) regime

$$\hat{f}_N(x; \Theta) = \frac{1}{\sqrt{N}} \sum_{i=1}^{N} a_i \sigma(\langle w_i, x \rangle), \quad a_i(0) \sim_{i.i.d.} \mathcal{N}(0, 1).$$

[Jacot *et al.*, 2018], [Du *et al.*, 2018], [Du *et al.*, 2018], [Allen-Zhu *el al.*, 2018], [Zou *et al.*, 2018]...

Three variants of the convergence theorem

Gradient flow (n: # data; N: # neuron):

$$\dot{\Theta}(t) = -\nabla \hat{\mathbb{E}}_{x,n}[(y - \hat{f}_N(x; \Theta(t)))^2],$$
$$w_i(0) \sim_{i.i.d.} \mathcal{N}(\mathbf{0}, \mathbf{I}_d/d).$$

Theorem: for N large enough, we have *

$$\lim_{t \to \infty} R_{n,N}(\Theta(t)) = 0.$$

Mean field (MF) regime

$$\hat{f}_N(x; \Theta) = \frac{1}{N} \sum_{i=1}^{N} a_i \sigma(\langle w_i, x \rangle), \quad a_i(0) \sim_{i.i.d.} \mathcal{N}(0, 1).$$

[Mei *et al.*, 2018], [Rotskoff and Vanden-Eijden, 2018], [Chizat and Bach, 2018]...

Three variants of the convergence theorem

Random feature (RF) regime

$$\hat{f}_N(x;\Theta) = \sum_{i=1}^{N} a_i \sigma(\langle w_i, x\rangle), \quad a_i \sim \mathcal{N}(0, 1/N^2).$$

Neural tangent (NT) regime

$$\hat{f}_N(x;\Theta) = \frac{1}{\sqrt{N}} \sum_{i=1}^{N} a_i \sigma(\langle w_i, x\rangle), \quad a_i \sim \mathcal{N}(0, 1).$$

Mean field (MF) regime

$$\hat{f}_N(x;\Theta) = \frac{1}{N} \sum_{i=1}^{N} a_i \sigma(\langle w_i, x\rangle), \quad a_i \sim \mathcal{N}(0, 1).$$

… but different behavior of dynamics

Random feature (RF) regime

$$\hat{f}_N(\boldsymbol{x};\boldsymbol{\Theta}) = \sum_{i=1}^{N} a_i \sigma(\langle \boldsymbol{w}_i, \boldsymbol{x}\rangle), \quad a_i \sim \mathcal{N}(0, 1/N^2).$$

- The limiting dynamics is linear (effectively only $\boldsymbol{a}$ is updated).
- Prediction function: kernel ridge regression with kernel

$$k_{\mathsf{RF}}(\boldsymbol{x},\boldsymbol{z}) = \hat{\mathbb{E}}_{\boldsymbol{w},N}[\sigma(\langle \boldsymbol{w},\boldsymbol{z}\rangle)\sigma(\langle \boldsymbol{w},\boldsymbol{z}\rangle)].$$

[Andoni *et al.*, 2014], [Danialy, 2017], [Yehudai and Shamir, 2019] …

... but different behavior of dynamics

Neural tangent (NT) regime

$$\hat{f}_N(x;\Theta) = \frac{1}{\sqrt{N}}\sum_{i=1}^{N} a_i\sigma(\langle w_i, x\rangle), \quad a_i \sim \mathcal{N}(0,1).$$

- The limiting dynamics is linear (the change of Θ is small).
- Prediction function: kernel ridge regression with kernel

$$k_{\mathsf{NT}}(x,z) = \hat{\mathbb{E}}_{w,N}[\sigma'(\langle w, x\rangle)\sigma'(\langle w, z\rangle)]\langle x, z\rangle + k_{\mathsf{RF}}(x,z).$$

[Jacot *et al.*, 2018], [Du *et al.*, 2018], [Du *et al.*, 2018], [Allen-Zhu *el al.*, 2018], [Zou *et al.*, 2018]...

… but different behavior of dynamics

Mean field (MF) regime

$$\hat{f}_N(x;\Theta) = \frac{1}{N}\sum_{i=1}^{N} a_i \sigma(\langle w_i, x\rangle), \quad a_i \sim \mathcal{N}(0,1).$$

- The limiting dynamics is non-linear (both $\boldsymbol{a}$ and $\boldsymbol{W}$ are updated).
- Distributional dynamics:
  $\partial_t \rho_t(a,w) = \nabla \cdot (\rho \nabla \Psi(a,w;\rho_t)) + \beta^{-1}\Delta\rho_t$.
- Prediction function: $\hat{f}(x;\rho_\infty) = \int a\sigma(\langle w, x\rangle)\rho_\infty(\mathrm{d}a\mathrm{d}w)$.

[Mei *et al.*, 2018], [Rotskoff and Vanden-Eijnden, 2018], [Chizat and Bach, 2018], [Sirignano and Spiliopoulos, 2018]...

Optimization: 0 training loss.

Test risk = training loss + generalization risk.

Today: generalization.

Optimization: 0 training loss.

Test risk = training loss + generalization risk.

Today: generalization.

Generalization theory for kernel methods

- Traditional theory: assume $f_\star \in$ RKHS, then kernel ridge regression generalize well.
- Problem: in high dimension, RKHS is a very small space.

Today: in high dimension, kernel methods (RF and NT) don't generalize well.

Generalization theory for kernel methods

- Traditional theory: assume $f_\star \in$ RKHS, then kernel ridge regression generalize well.
- Problem: in high dimension, RKHS is a very small space.

Today: in high dimension, kernel methods (RF and NT) don't generalize well.

Generalization theory for kernel methods

- Traditional theory: assume $f_\star \in$ RKHS, then kernel ridge regression generalize well.
- Problem: in high dimension, RKHS is a very small space.

Today: in high dimension, kernel methods (RF and NT) don't generalize well.

Setting 1: N finite, n infinite

Distribution:

$$x \in \mathrm{Unif}(\mathbb{S}^{d-1}(\sqrt{d})), \quad y = f_\star(x), \quad f_\star \in L^2(\mathbb{S}^d(\sqrt{d})).$$

Two classes of linearized neural network: $(\boldsymbol{w}_i \sim \mathrm{Unif}(\mathbb{S}^d))$

$$\mathcal{F}_{\mathsf{RF},N}(\boldsymbol{W}) = \Big\{ f = \sum_{i=1}^{N} a_i \sigma(\langle \boldsymbol{w}_i, \boldsymbol{x}\rangle) : a_i \in \mathbb{R}, i \in [N] \Big\},$$

$$\mathcal{F}_{\mathsf{NT},N}(\boldsymbol{W}) = \Big\{ f = \sum_{i=1}^{N} \sigma'(\langle \boldsymbol{w}_i, \boldsymbol{x}\rangle)\langle \boldsymbol{a}_i, \boldsymbol{x}\rangle : \boldsymbol{a}_i \in \mathbb{R}^d, i \in [N] \Big\}.$$

Mild assumptions on σ (universal approximation, growth not too fast).

Lower bound: N finite, n infinite

$$\mathcal{F}_{\mathrm{RF},N}(\boldsymbol{W}) = \Big\{ f = \sum_{i=1}^{N} a_i \sigma(\langle \boldsymbol{w}_i, \boldsymbol{x} \rangle) : a_i \in \mathbb{R}, i \in [N] \Big\}.$$

Theorem (Ghorbani, Mei, Misiakiwics, Montanari, 2019)

Assume $N = O_d(d^{\ell-\delta})$, *and* $(\boldsymbol{w}_i)_{i \in [N]} \sim \mathrm{Unif}(\mathbb{S}^d)$, *we have*

$$\inf_{f \in \mathcal{F}_{\mathrm{RF},N}(\boldsymbol{W})} \mathbb{E}_{\boldsymbol{x}}[(f_\star(\boldsymbol{x}) - f(\boldsymbol{x}))^2] \ge \|\mathsf{P}_{>\ell} f_\star\|_{L^2}^2 + o_{d,\mathbb{P}}(\|f_\star\|_2^2),$$

where $\mathsf{P}_{>\ell}$ *is the projection operator orthogonal to the space of degree-*ℓ *polynomials.*

Example: for $f_\star(x) = x_1^2 - 1$, we have $\mathsf{P}_{>2} f_\star \approx f_\star$. Then random feature regression with $N = O_d(d^{2-\delta})$ neuron achieves trivial risk, which is $\|f_\star\|_{L^2}^2$.

Lower bound: N finite, n infinite

$$\mathcal{F}_{\mathrm{RF},N}(\boldsymbol{W}) = \Big\{ f = \sum_{i=1}^{N} a_i \sigma(\langle \boldsymbol{w}_i, \boldsymbol{x} \rangle) : a_i \in \mathbb{R}, i \in [N] \Big\}.$$

Theorem (Ghorbani, Mei, Misiakiwics, Montanari, 2019)

Assume $N = O_d(d^{\ell - \delta})$, *and* $(\boldsymbol{w}_i)_{i \in [N]} \sim \mathrm{Unif}(\mathbb{S}^d)$, *we have*

$$\inf_{f \in \mathcal{F}_{\mathrm{RF},N}(\boldsymbol{W})} \mathbb{E}_{\boldsymbol{x}}[(f_\star(\boldsymbol{x}) - f(\boldsymbol{x}))^2] \geq \|\mathsf{P}_{>\ell} f_\star\|_{L^2}^2 + o_{d,\mathbb{P}}(\|f_\star\|_2^2),$$

where $\mathsf{P}_{>\ell}$ *is the projection operator orthogonal to the space of degree-*ℓ *polynomials.*

Example: for $f_\star(\boldsymbol{x}) = x_1^2 - 1$, we have $\mathsf{P}_{>2} f_\star \approx f_\star$. Then random feature regression with $N = O_d(d^{2-\delta})$ neuron achieves trivial risk, which is $\|f_\star\|_{L^2}^2$.

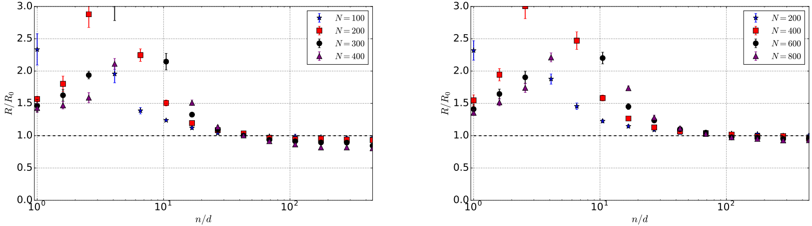


Figure: Test risk for learning $f(x) = x_1^2 - 1$, d = 50 and d = 100.

Similar result for NT

$$\mathcal{F}_{\mathsf{NT},N}(\boldsymbol{W}) = \Big\{ f = \sum_{i=1}^{N} \sigma'(\langle \boldsymbol{w}_i, \boldsymbol{x}\rangle)\langle \boldsymbol{a}_i, \boldsymbol{x}\rangle : \boldsymbol{a}_i \in \mathbb{R}^d, i \in [N] \Big\}.$$

Theorem (Ghorbani, Mei, Misiakiwics, Montanari, 2019)

Assume $N = O_d(d^{\ell-\delta})$, *and* $(\boldsymbol{w}_i)_{i\in[N]} \sim \mathrm{Unif}(\mathbb{S}^d)$, *we have*

$$\inf_{f\in\mathcal{F}_{\mathsf{RF},N}(\boldsymbol{W})} \mathbb{E}_{\boldsymbol{x}}[(f_\star(\boldsymbol{x}) - f(\boldsymbol{x}))^2] \ge \|\mathsf{P}_{>\ell+1} f_\star\|_{L^2}^2 + o_{d,\mathbb{P}}(\|f_\star\|_2^2),$$

where $\mathsf{P}_{>\ell+1}$ *is the projection operator orthogonal to the space of degree-*$(\ell+1)$ *polynomials.*

Example: for $f_\star(x) = x_1^3 - x_1$, we have $\mathsf{P}_{>3} f_\star \approx f_\star$. Then random feature regression with $N = O_d(d^{2-\delta})$ neuron achieves trivial risk, which is $\|f_\star\|_{L^2}^2$.

Similar result for NT

$$\mathcal{F}_{\mathsf{NT},N}(\boldsymbol{W}) = \Big\{ f = \sum_{i=1}^{N} \sigma'(\langle \boldsymbol{w}_i, \boldsymbol{x} \rangle)\langle \boldsymbol{a}_i, \boldsymbol{x} \rangle : \boldsymbol{a}_i \in \mathbb{R}^d, i \in [N] \Big\}.$$

Theorem (Ghorbani, Mei, Misiakiwics, Montanari, 2019)

Assume $N = O_d(d^{\ell-\delta})$, *and* $(\boldsymbol{w}_i)_{i\in[N]} \sim \mathrm{Unif}(\mathbb{S}^d)$, *we have*

$$\inf_{f \in \mathcal{F}_{\mathsf{RF},N}(\boldsymbol{W})} \mathbb{E}_{\boldsymbol{x}}[(f_\star(\boldsymbol{x}) - f(\boldsymbol{x}))^2] \ge \|\mathsf{P}_{>\ell+1} f_\star\|_{L^2}^2 + o_{d,\mathbb{P}}(\|f_\star\|_2^2),$$

where $\mathsf{P}_{>\ell+1}$ *is the projection operator orthogonal to the space of degree-*$(\ell+1)$ *polynomials.*

Example: for $f_\star(\boldsymbol{x}) = x_1^3 - x_1$, we have $\mathsf{P}_{>3} f_\star \approx f_\star$. Then random feature regression with $N = O_d(d^{2-\delta})$ neuron achieves trivial risk, which is $\|f_\star\|_{L^2}^2$.

Setting 2: N infinite, n finite

Distribution:

$$\boldsymbol{x}_i \in \mathrm{Unif}(\mathbb{S}^{d-1}), \quad y_i = f_\star(\boldsymbol{x}_i), \quad f_\star \in L^2(\mathbb{S}^d(\sqrt{d})).$$

Predicting using regularized kernel ridge regression:

$$\hat{f}_\lambda(\boldsymbol{x}) = \boldsymbol{k}(\boldsymbol{x}, \mathcal{X})(\boldsymbol{k}(\mathcal{X}, \mathcal{X}) + \lambda \mathbf{I})^{-1} f_\star(\boldsymbol{x}),$$

where

$$\boldsymbol{k}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \mathbb{E}_{\boldsymbol{w} \sim \mathbb{S}^{d-1}}[\sigma(\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle)\sigma(\langle \boldsymbol{w}, \boldsymbol{x}_j \rangle)].$$

Lower bound: N infinite, n finite

Theorem (Ghorbani, Mei, Misiakiwics, Montanari, 2019)

Assume $n = O_d(d^{\ell-\delta})$, *we have*

$$\inf_{\lambda} \mathbb{E}_{\boldsymbol{x}}[(f_\star(\boldsymbol{x}) - \hat{f}_\lambda(\boldsymbol{x}))^2] \geq \|\mathsf{P}_{>\ell} f_\star\|_{L^2}^2 + o_{d,\mathbb{P}}(\|f_\star\|_2^2),$$

where $\mathsf{P}_{>\ell}$ *is the projection operator orthogonal to the space of degree-*ℓ *polynomials.*

Intuition behind these results

In high dimension, the correlation between a degree-k Hermite polynomial and a random feature is very small

$$\mathbb{E}_{\boldsymbol{w}}[\mathrm{He}_k(x_1)\sigma(\langle \boldsymbol{w}, \boldsymbol{x} \rangle)] = O_d(1/d^k).$$

Also observed in [Danialy, 2016], [Bach, 2017].

Implications & Conclusions

- In high dimension, even for simple function $f(x) = x_1^k$, it takes n, $N = O_d(d^k)$ to learn it well using linearized neural network (kernel methods);
- ... while a neural network can learn it (conjecture to be efficiently) using n, $N = O_d(1)$.
- Neural network is more powerful than kernel methods.
- Future work: what class of functions neural network can learn efficiently.

Implications & Conclusions

- In high dimension, even for simple function $f(x) = x_1^k$, it takes n, $N = O_d(d^k)$ to learn it well using linearized neural network (kernel methods);
- ... while a neural network can learn it (conjecture to be efficiently) using n, $N = O_d(1)$.
- Neural network is more powerful than kernel methods.
- Future work: what class of functions neural network can learn efficiently.

Implications & Conclusions

- In high dimension, even for simple function $f(x) = x_1^k$, it takes n, $N = O_d(d^k)$ to learn it well using linearized neural network (kernel methods);
- ... while a neural network can learn it (conjecture to be efficiently) using n, $N = O_d(1)$.
- Neural network is more powerful than kernel methods.
- Future work: what class of functions neural network can learn efficiently.

Implications & Conclusions

- In high dimension, even for simple function $f(x) = x_1^k$, it takes n, $N = O_d(d^k)$ to learn it well using linearized neural network (kernel methods);
- ... while a neural network can learn it (conjecture to be efficiently) using n, $N = O_d(1)$.
- Neural network is more powerful than kernel methods.
- Future work: what class of functions neural network can learn efficiently.