When do neural networks outperform kernel methods?

Song Mei

Stanford University

June 28, 2020

Joint work with Behrooz Ghorbani, Theodor Misiakiewicz, and Andrea Montanari

Neural tangent model

- Multi-layers NN: $f_N(x;\theta)$, $x \in \mathbb{R}^d$, $\theta \in \mathbb{R}^N$
- Expanding around θ_0:

$$f_N(x;\theta) = f_N(x;\theta_0) + \underline{\langle \theta - \theta_0, \nabla_\theta f_N(x;\theta_0)\rangle} + o(\|\theta - \theta_0\|_2).$$

- Neural tangent model:

$$f_{\mathrm{NT},N}(x;\beta,\theta_0) = \langle \beta, \nabla_\theta f_N(x;\theta_0)\rangle.$$

- Coupled gradient flow:

$$\frac{\mathrm{d}}{\mathrm{d}t}\theta^t = -\nabla_\theta \hat{\mathbb{E}}[(y - f_N(x;\theta^t))^2], \qquad \theta^0 = \theta_0,$$
$$\frac{\mathrm{d}}{\mathrm{d}t}\beta^t = -\nabla_\beta \hat{\mathbb{E}}[(y - f_{\mathrm{NT},N}(x;\beta^t,\theta_0))^2], \qquad \beta^0 = \mathbf{0}.$$

- Under proper initialization and over-parameterization:

$$\lim_{N\to\infty} |f_N(x;\theta^t) - f_{\mathrm{NT},N}(x;\beta^t)| = 0.$$

[Jacot, Gabriel, Hongler, 2018], [Du, Zhai, Poczos, Singh, 2018], [Chizat, Bach, 2018b], ...

Neural tangent model

- Multi-layers NN: $f_N(\boldsymbol{x}; \boldsymbol{\theta})$, $\boldsymbol{x} \in \mathbb{R}^d$, $\boldsymbol{\theta} \in \mathbb{R}^N$
- Expanding around $\boldsymbol{\theta}_0$:

$$f_N(\boldsymbol{x}; \boldsymbol{\theta}) = f_N(\boldsymbol{x}; \boldsymbol{\theta}_0) + \underline{\langle \boldsymbol{\theta} - \boldsymbol{\theta}_0, \nabla_{\boldsymbol{\theta}} f_N(\boldsymbol{x}; \boldsymbol{\theta}_0) \rangle} + o(\|\boldsymbol{\theta} - \boldsymbol{\theta}_0\|_2).$$

- Neural tangent model:

$$f_{\mathrm{NT},N}(\boldsymbol{x}; \boldsymbol{\beta}, \boldsymbol{\theta}_0) = \langle \boldsymbol{\beta}, \nabla_{\boldsymbol{\theta}} f_N(\boldsymbol{x}; \boldsymbol{\theta}_0) \rangle.$$

- Coupled gradient flow:

$$\frac{\mathrm{d}}{\mathrm{d}t} \boldsymbol{\theta}^t = -\nabla_{\boldsymbol{\theta}} \hat{\mathbb{E}}[(y - f_N(\boldsymbol{x}; \boldsymbol{\theta}^t))^2], \qquad \boldsymbol{\theta}^0 = \boldsymbol{\theta}_0,$$
$$\frac{\mathrm{d}}{\mathrm{d}t} \boldsymbol{\beta}^t = -\nabla_{\boldsymbol{\beta}} \hat{\mathbb{E}}[(y - f_{\mathrm{NT},N}(\boldsymbol{x}; \boldsymbol{\beta}^t, \boldsymbol{\theta}_0))^2], \qquad \boldsymbol{\beta}^0 = \mathbf{0}.$$

- Under proper initialization and over-parameterization:

$$\lim_{N \to \infty} |f_N(\boldsymbol{x}; \boldsymbol{\theta}^t) - f_{\mathrm{NT},N}(\boldsymbol{x}; \boldsymbol{\beta}^t)| = 0.$$

[Jacot, Gabriel, Hongler, 2018], [Du, Zhai, Poczos, Singh, 2018], [Chizat, Bach, 2018b], ...

Neural tangent model

- Multi-layers NN: $f_N(\boldsymbol{x}; \boldsymbol{\theta})$, $\boldsymbol{x} \in \mathbb{R}^d$, $\boldsymbol{\theta} \in \mathbb{R}^N$
- Expanding around $\boldsymbol{\theta}_0$:

$$f_N(\boldsymbol{x}; \boldsymbol{\theta}) = f_N(\boldsymbol{x}; \boldsymbol{\theta}_0) + \underline{\langle \boldsymbol{\theta} - \boldsymbol{\theta}_0, \nabla_{\boldsymbol{\theta}} f_N(\boldsymbol{x}; \boldsymbol{\theta}_0) \rangle} + o(\|\boldsymbol{\theta} - \boldsymbol{\theta}_0\|_2).$$

- Neural tangent model:

$$f_{\mathsf{NT},N}(\boldsymbol{x}; \boldsymbol{\beta}, \boldsymbol{\theta}_0) = \langle \boldsymbol{\beta}, \nabla_{\boldsymbol{\theta}} f_N(\boldsymbol{x}; \boldsymbol{\theta}_0) \rangle.$$

- Coupled gradient flow:

$$\frac{\mathrm{d}}{\mathrm{d}t}\theta^t = -\nabla_\theta \hat{\mathbb{E}}[(y - f_N(x; \theta^t))^2], \qquad \theta^0 = \theta_0,$$
$$\frac{\mathrm{d}}{\mathrm{d}t}\beta^t = -\nabla_\beta \hat{\mathbb{E}}[(y - f_{\mathsf{NT},N}(x; \beta^t, \theta_0))^2], \qquad \beta^0 = \mathbf{0}.$$

- Under proper initialization and over-parameterization:

$$\lim_{N \to \infty} |f_N(x; \theta^t) - f_{\mathsf{NT},N}(x; \beta^t)| = 0.$$

[Jacot, Gabriel, Hongler, 2018], [Du, Zhai, Poczos, Singh, 2018], [Chizat, Bach, 2018b], ...

Neural tangent model

- Multi-layers NN: $f_N(x;\theta)$, $x \in \mathbb{R}^d$, $\theta \in \mathbb{R}^N$
- Expanding around θ_0:

$$f_N(x;\theta) = f_N(x;\theta_0) + \underline{\langle \theta - \theta_0, \nabla_\theta f_N(x;\theta_0)\rangle} + o(\|\theta - \theta_0\|_2).$$

- Neural tangent model:

$$f_{\mathrm{NT},N}(x;\beta,\theta_0) = \langle \beta, \nabla_\theta f_N(x;\theta_0)\rangle.$$

- Coupled gradient flow:

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t}\theta^t &= -\nabla_\theta \hat{\mathbb{E}}[(y - f_N(x;\theta^t))^2], && \theta^0 = \theta_0,\\
\frac{\mathrm{d}}{\mathrm{d}t}\beta^t &= -\nabla_\beta \hat{\mathbb{E}}[(y - f_{\mathrm{NT},N}(x;\beta^t,\theta_0))^2], && \beta^0 = \mathbf{0}.
\end{aligned}$$

- Under proper initialization and over-parameterization:

$$\lim_{N\to\infty} |f_N(x;\theta^t) - f_{\mathrm{NT},N}(x;\beta^t)| = 0.$$

[Jacot, Gabriel, Hongler, 2018], [Du, Zhai, Poczos, Singh, 2018], [Chizat, Bach, 2018b], ...

Neural tangent model

- Multi-layers NN: $f_N(\boldsymbol{x}; \boldsymbol{\theta})$, $\boldsymbol{x} \in \mathbb{R}^d$, $\boldsymbol{\theta} \in \mathbb{R}^N$
- Expanding around $\boldsymbol{\theta}_0$:

$$f_N(\boldsymbol{x}; \boldsymbol{\theta}) = f_N(\boldsymbol{x}; \boldsymbol{\theta}_0) + \underline{\langle \boldsymbol{\theta} - \boldsymbol{\theta}_0, \nabla_{\boldsymbol{\theta}} f_N(\boldsymbol{x}; \boldsymbol{\theta}_0) \rangle} + o(\|\boldsymbol{\theta} - \boldsymbol{\theta}_0\|_2).$$

- Neural tangent model:

$$f_{\mathrm{NT},N}(\boldsymbol{x}; \boldsymbol{\beta}, \boldsymbol{\theta}_0) = \langle \boldsymbol{\beta}, \nabla_{\boldsymbol{\theta}} f_N(\boldsymbol{x}; \boldsymbol{\theta}_0) \rangle.$$

- Coupled gradient flow:

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t} \boldsymbol{\theta}^t &= -\nabla_{\boldsymbol{\theta}} \hat{\mathbb{E}}[(y - f_N(\boldsymbol{x}; \boldsymbol{\theta}^t))^2], & \boldsymbol{\theta}^0 &= \boldsymbol{\theta}_0, \\ \frac{\mathrm{d}}{\mathrm{d}t} \boldsymbol{\beta}^t &= -\nabla_{\boldsymbol{\beta}} \hat{\mathbb{E}}[(y - f_{\mathrm{NT},N}(\boldsymbol{x}; \boldsymbol{\beta}^t, \boldsymbol{\theta}_0))^2], & \boldsymbol{\beta}^0 &= \mathbf{0}. \end{aligned}$$

- Under proper initialization and over-parameterization:

$$\lim_{N \to \infty} |f_N(\boldsymbol{x}; \boldsymbol{\theta}^t) - f_{\mathrm{NT},N}(\boldsymbol{x}; \boldsymbol{\beta}^t)| = 0.$$

[Jacot, Gabriel, Hongler, 2018], [Du, Zhai, Poczos, Singh, 2018], [Chizat, Bach, 2018b], ...

How about generalization?

- [Arora, Du, Hu, Li, Salakhutdinov, Wang, 2019]:
  Cifar10 experiments. NT: 23% test error. NN: less than 5% test error.
- [Arora, Du, Li, Salakhutdinov, Wang, Yu, 2019]:
  Small dataset, NT sometimes generalize better than NN.
- [Shankar, Fang, Guo, Fridovich-Keil, Schmidt, Ragan-Kelley, Recht, 2020]
  [Li, Wang, Yu, Du, Hu, Salakhutdinov, Arora, 2019]:
  Smaller gap between NT and NN on Cifar10 (10% for NT).

Sometimes there is a large gap, while sometimes the gap is small.

How about generalization?

- [Arora, Du, Hu, Li, Salakhutdinov, Wang, 2019]:
  Cifar10 experiments. NT: 23% test error. NN: less than 5% test error.
- [Arora, Du, Li, Salakhutdinov, Wang, Yu, 2019]:
  Small dataset, NT sometimes generalize better than NN.
- [Shankar, Fang, Guo, Fridovich-Keil, Schmidt, Ragan-Kelley, Recht, 2020]
  [Li, Wang, Yu, Du, Hu, Salakhutdinov, Arora, 2019]:
  Smaller gap between NT and NN on Cifar10 (10% for NT).

Sometimes there is a large gap, while sometimes the gap is small.

Focus of this talk

When is there a large performance gap between NN and NT?

Two-layers neural networks

Neural networks:

$$\mathcal{F}_{\mathsf{NN},N} = \left\{ f_N(x;\Theta) = \sum_{i=1}^{N} a_i \sigma(\langle w_i, x\rangle) : a_i \in \mathbb{R}, w_i \in \mathbb{R}^d \right\}.$$

Linearization:

$$f_N(x;\Theta) = f_N(x;\Theta^0) + \underbrace{\sum_{i=1}^{N} \Delta a_i \sigma(\langle w_i^0, x\rangle)}_{\text{Top layer linearization}} + \underbrace{\sum_{i=1}^{N} a_i^0 \sigma'(\langle w_i^0, x\rangle)\langle \Delta w_i, x\rangle}_{\text{Bottom layer linearization}} + o(\cdot).$$

Linearized neural networks ($W = (w_i)_{i\in[N]} \sim_{iid} \mathrm{Unif}(\mathbb{S}^{d-1})$):

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

$$\mathcal{F}_{\mathsf{NT},N}(W) = \left\{ f = \sum_{i=1}^{N} \sigma'(\langle w_i, x\rangle)\langle b_i, x\rangle : b_i \in \mathbb{R}^d, i \in [N] \right\}.$$

Two-layers neural networks

Neural networks:

$$\mathcal{F}_{\mathsf{NN},N} = \Big\{ f_N(\boldsymbol{x}; \boldsymbol{\Theta}) = \sum_{i=1}^{N} a_i \sigma(\langle \boldsymbol{w}_i, \boldsymbol{x} \rangle) : a_i \in \mathbb{R}, \boldsymbol{w}_i \in \mathbb{R}^d \Big\}.$$

Linearization:

$$f_N(\boldsymbol{x}; \boldsymbol{\Theta}) = f_N(\boldsymbol{x}; \boldsymbol{\Theta}^0) + \underbrace{\sum_{i=1}^{N} \Delta a_i \sigma(\langle \boldsymbol{w}_i^0, \boldsymbol{x} \rangle)}_{\text{Top layer linearization}} + \underbrace{\sum_{i=1}^{N} a_i^0 \sigma'(\langle \boldsymbol{w}_i^0, \boldsymbol{x} \rangle) \langle \Delta \boldsymbol{w}_i, \boldsymbol{x} \rangle}_{\text{Bottom layer linearization}} + o(\cdot).$$

Linearized neural networks ($\boldsymbol{W} = (\boldsymbol{w}_i)_{i \in [N]} \sim_{iid} \mathrm{Unif}(\mathbb{S}^{d-1})$):

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

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

Spiked features model

- Signal features and junk features

$$\boldsymbol{x} = (\boldsymbol{x}_1, \boldsymbol{x}_2) \in \mathbb{R}^d, \quad \boldsymbol{x}_1 \in \mathbb{R}^{d_{\mathrm{s}}}, \quad \boldsymbol{x}_2 \in \mathbb{R}^{d-d_{\mathrm{s}}},$$
$$d_{\mathrm{s}} = d^{\eta}, \quad 0 \le \eta \le 1,$$
$$\mathrm{Cov}(\boldsymbol{x}_1) = \mathrm{snr}_{\mathrm{f}} \cdot \mathbf{I}_{d_{\mathrm{s}}}, \quad \mathrm{Cov}(\boldsymbol{x}_2) = \mathbf{I}_{d-d_{\mathrm{s}}},$$
$$\mathrm{snr}_{\mathrm{f}} = d^{\kappa}, \quad 0 \le \kappa < \infty \quad \text{(feature SNR)}.$$

- Response depend on signal features

$$y = f_\star(\boldsymbol{x}) + \varepsilon, \quad f_\star(\boldsymbol{x}) = \varphi(\boldsymbol{x}_1).$$

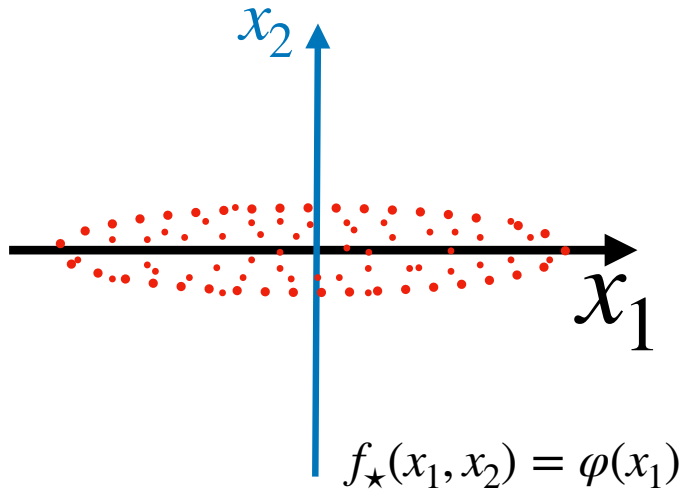


Figure: Anisotropic features: $\kappa > 0$, $\mathrm{snr}_{\mathrm{f}} > 1$

- Feature SNR: $\mathrm{snr}_{\mathrm{f}} = d^{\kappa} \ge 1$.
- Effective dimension: $d_{\mathrm{eff}} = d_{\mathrm{s}} \vee (d/\mathrm{snr}_{\mathrm{f}})$. We have $d_{\mathrm{s}} \le d_{\mathrm{eff}} \le d$.
- Larger $\mathrm{snr}_{\mathrm{f}}$ induces smaller d_{eff}.

More precisely: $\boldsymbol{x} \sim \mathrm{Unif}(\mathbb{S}^{d_{\mathrm{s}}}(r\sqrt{d_{\mathrm{s}}})) \times \mathrm{Unif}(\mathbb{S}^{d-d_{\mathrm{s}}}(\sqrt{d}))$. Generalizable to multi-spheres.

Spiked features model

- Signal features and junk features

$$
\begin{aligned}
&\boldsymbol{x} = (\boldsymbol{x}_1, \boldsymbol{x}_2) \in \mathbb{R}^d, \quad \boldsymbol{x}_1 \in \mathbb{R}^{d_{\mathrm{s}}}, \quad \boldsymbol{x}_2 \in \mathbb{R}^{d-d_{\mathrm{s}}}, \\
&d_{\mathrm{s}} = d^{\eta}, \quad 0 \le \eta \le 1, \\
&\mathrm{Cov}(\boldsymbol{x}_1) = \mathrm{snr}_{\mathrm{f}} \cdot \mathbf{I}_{d_{\mathrm{s}}}, \quad \mathrm{Cov}(\boldsymbol{x}_2) = \mathbf{I}_{d-d_{\mathrm{s}}}, \\
&\mathrm{snr}_{\mathrm{f}} = d^{\kappa}, \quad 0 \le \kappa < \infty \quad \text{(feature SNR)}.
\end{aligned}
$$

- Response depend on signal features

$$
y = f_\star(\boldsymbol{x}) + \varepsilon, \quad f_\star(\boldsymbol{x}) = \varphi(\boldsymbol{x}_1).
$$

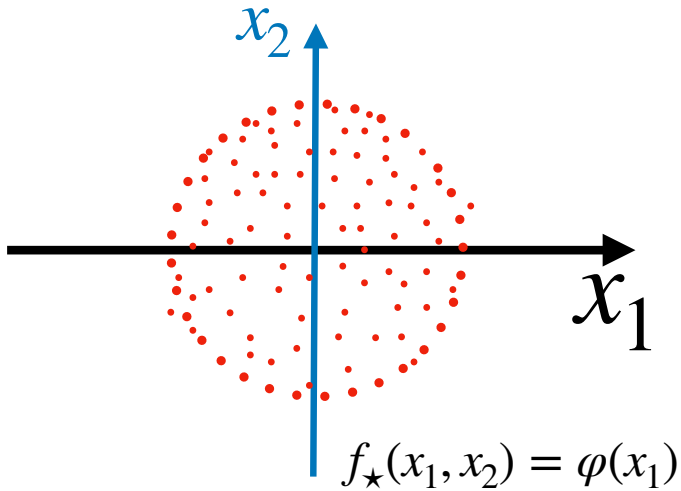


Figure: Isotropic features: $\kappa = 0$, $\mathrm{snr}_{\mathrm{f}} = 1$

- Feature SNR: $\mathrm{snr}_{\mathrm{f}} = d^{\kappa} \ge 1$.
- Effective dimension: $d_{\mathrm{eff}} = d_{\mathrm{s}} \vee (d/\mathrm{snr}_{\mathrm{f}})$. We have $d_{\mathrm{s}} \le d_{\mathrm{eff}} \le d$.
- Larger $\mathrm{snr}_{\mathrm{f}}$ induces smaller d_{eff}.

More precisely: $\boldsymbol{x} \sim \mathrm{Unif}(\mathbb{S}^{d_{\mathrm{s}}}(r\sqrt{d_{\mathrm{s}}})) \times \mathrm{Unif}(\mathbb{S}^{d-d_{\mathrm{s}}}(\sqrt{d}))$. Generalizable to multi-spheres.

Spiked features model

- Signal features and junk features

$$\boldsymbol{x} = (\boldsymbol{x}_1, \boldsymbol{x}_2) \in \mathbb{R}^d, \quad \boldsymbol{x}_1 \in \mathbb{R}^{d_{\mathrm{s}}}, \quad \boldsymbol{x}_2 \in \mathbb{R}^{d-d_{\mathrm{s}}},$$
$$d_{\mathrm{s}} = d^{\eta}, \quad 0 \le \eta \le 1,$$
$$\mathrm{Cov}(\boldsymbol{x}_1) = \mathrm{snr}_{\mathrm{f}} \cdot \mathbf{I}_{d_{\mathrm{s}}}, \quad \mathrm{Cov}(\boldsymbol{x}_2) = \mathbf{I}_{d-d_{\mathrm{s}}},$$
$$\mathrm{snr}_{\mathrm{f}} = d^{\kappa}, \quad 0 \le \kappa < \infty \quad \text{(feature SNR)}.$$

- Response depend on signal features

$$y = f_\star(\boldsymbol{x}) + \varepsilon, \quad f_\star(\boldsymbol{x}) = \varphi(\boldsymbol{x}_1).$$

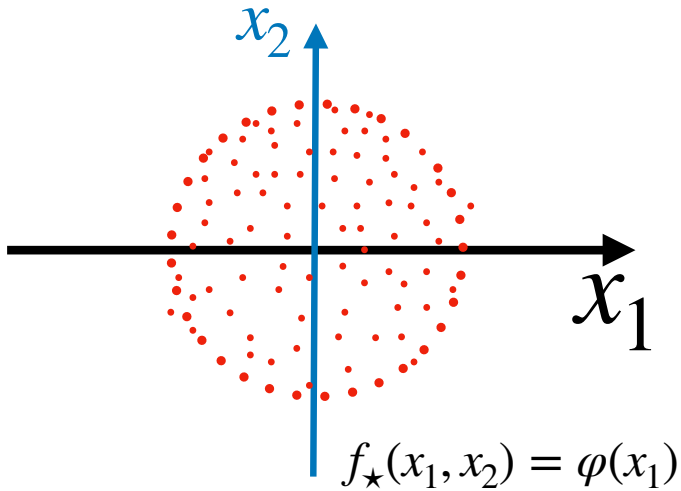


Figure: Isotropic features: $\kappa = 0$, $\mathrm{snr}_{\mathrm{f}} = 1$

- Feature SNR: $\mathrm{snr}_{\mathrm{f}} = d^{\kappa} \ge 1$.
- Effective dimension: $d_{\mathrm{eff}} = d_{\mathrm{s}} \vee (d/\mathrm{snr}_{\mathrm{f}})$. We have $d_{\mathrm{s}} \le d_{\mathrm{eff}} \le d$.
- Larger $\mathrm{snr}_{\mathrm{f}}$ induces smaller d_{eff}.

More precisely: $\boldsymbol{x} \sim \mathrm{Unif}(\mathbb{S}^{d_{\mathrm{s}}}(r\sqrt{d_{\mathrm{s}}})) \times \mathrm{Unif}(\mathbb{S}^{d-d_{\mathrm{s}}}(\sqrt{d}))$. Generalizable to multi-spheres.

Approximation error with N neurons

Approximation error: $R(f_\star, \mathcal{F}) = \inf_{f \in \mathcal{F}} \|f_\star - f\|_{L^2}^2$.

Theorem (Ghorbani, Mei, Misiakiewicz, Montanari, 2020)

Assume $d_{\rm eff}^{\ell+\delta} \le N \le d_{\rm eff}^{\ell+1-\delta}$ *and "generic condition" on* σ, *we have*

$$R(f_\star, \mathcal{F}_{\mathsf{RF},N}(\boldsymbol{W})) = \|\mathsf{P}_{>\ell} f_\star\|_{L^2}^2 + o_{d,\mathbb{P}}(\cdot),$$
$$R(f_\star, \mathcal{F}_{\mathsf{NT},N}(\boldsymbol{W})) = \|\mathsf{P}_{>\ell+1} f_\star\|_{L^2}^2 + o_{d,\mathbb{P}}(\cdot).$$

On the contrary, assume $d_{\rm s}^{\ell+\delta} \le N \le d_{\rm s}^{\ell+1-\delta}$, *we have*

$$R(f_\star, \mathcal{F}_{\mathsf{NN},N}) \le \|\mathsf{P}_{>\ell+1} f_\star\|_{L^2}^2 + o_d(\cdot).$$

Moreover, $R(f_\star, \mathcal{F}_{\mathsf{NN},N})$ *is independent of* $\mathrm{snr}_{\rm f}$.

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

Approximation error with N neurons

Dim $d_{\text{eff}} \equiv d_{\text{s}} \vee (d/\text{snr}_{\text{f}})$ and $d_{\text{s}} \le d_{\text{eff}} \le d$.

To approx. a degree-ℓ poly. in x_1:

- NN need at most ${d_{\text{s}}}^{\ell}$ parameters*.
- RF need ${d_{\text{eff}}}^{\ell}$ parameters.
- NT need ${d_{\text{eff}}}^{\ell-1} \cdot d$ parameters.

Approximation power: NN $\ge$ RF $\ge$ NT.

* If we don't count parameters with value 0.

Extreme case: low feature SNR

Fix $0 < \eta < 1$, low snr_f: $\kappa = 0$.

To approx. a degree-ℓ poly. in x_1:

- NN need at most $d^{\eta\ell}$ parameters*.
- RF need d^{ℓ} parameters.
- NT need d^{ℓ} parameters.

Approximation power: NN $>$ RF $=$ NT.

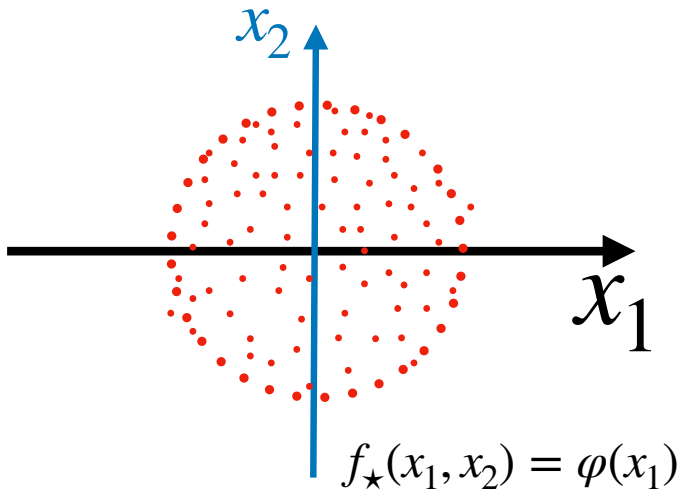


Figure: Isotropic features: $\kappa = 0$, $\mathrm{snr}_f = 1$

* If we don't count parameters with value 0.

Extreme case: high feature SNR

Fix $0 < \eta < 1$, high snr_f: $\kappa \gg 1$.

To approx. a degree-ℓ poly. in x_1:

- NN need at most $d^{\eta\ell}$ parameters*.
- RF need $d^{\eta\ell}$ parameters.
- NT need $d^{\eta(\ell-1)+1}$ parameters.

Approximation power: NN $\sim$ RF $>$ NT.

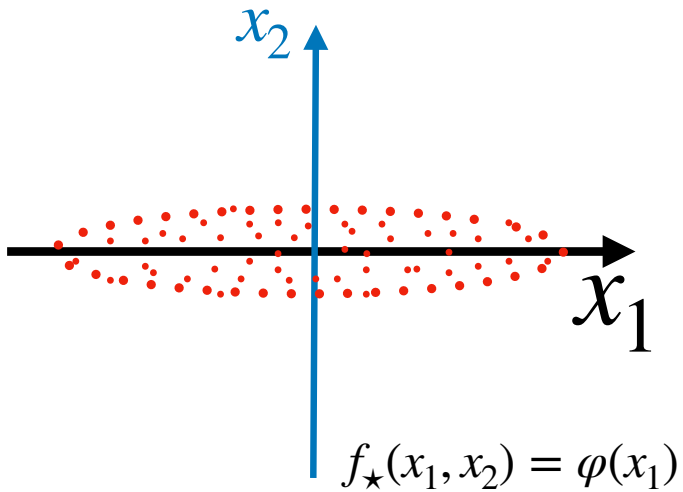
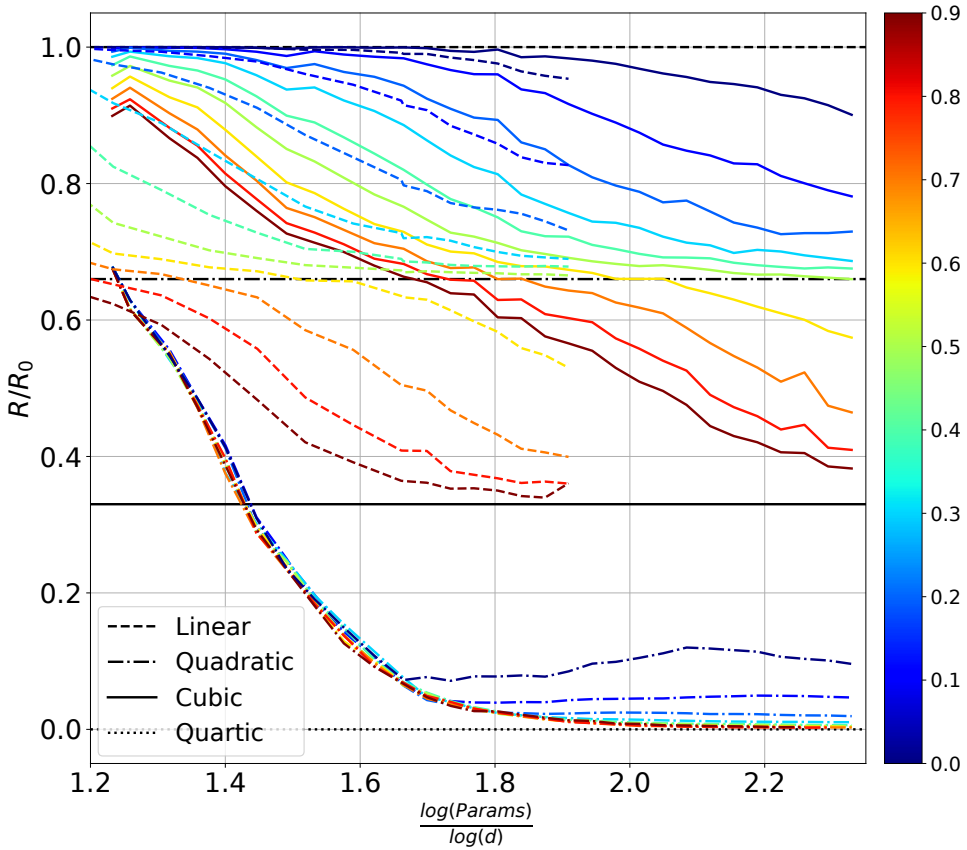


Figure: Anisotropic features: $\kappa > 0$, $\mathrm{snr}_f > 1$

* If we don't count parameters with value 0.

Numerical simulations



Colorbar: $\kappa \in [0, 1]$.
Dot-dashed: NN.
Dashed lines: RF;
Continuous lines: NT;
Dimension: $d = 1024$.
Eff. dim: $d_s = 16$.

Conclusion

(a) Power: NN $\geq$ RF $\geq$ NT. (b) Risk of NN independent of $\mathtt{snr}_f$.
(c) Larger $\mathtt{snr}_f$ induces larger power of $\{\mathrm{RF}, \mathrm{NT}\}$.

Similar results for generalization error with finite samples n

Extreme case: low feature SNR

Fix $0 < \eta < 1$, low $\mathtt{snr}_f$: $\kappa = 0$.

To fit a degree-ℓ poly. in x_1:

- $\exists \nu$, NN need at most $d^{\eta \ell}$ samples.
- $\{\mathsf{RF}, \mathsf{NT}\}$ kernel need d^{ℓ} samples.

Potential generalization power:
NN $>$ Kernel methods.

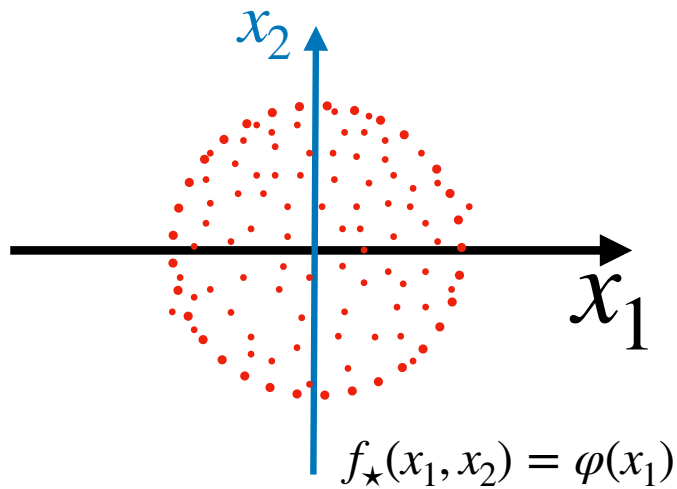


Figure: Isotropic features: $\kappa = 0$, $\mathtt{snr}_f = 1$

Extreme case: high feature SNR

Fix $0 < \eta < 1$, high snr_f: $\kappa \gg 1$.

To fit a degree-ℓ poly. in x_1:

- $\exists \nu$, NN need at most $d^{\eta \ell}$ samples.
- $\{\mathrm{RF}, \mathrm{NT}\}$ kernel need $d^{\eta \ell}$ samples.

Potential generalization power:
NN $\sim$ Kernel methods.

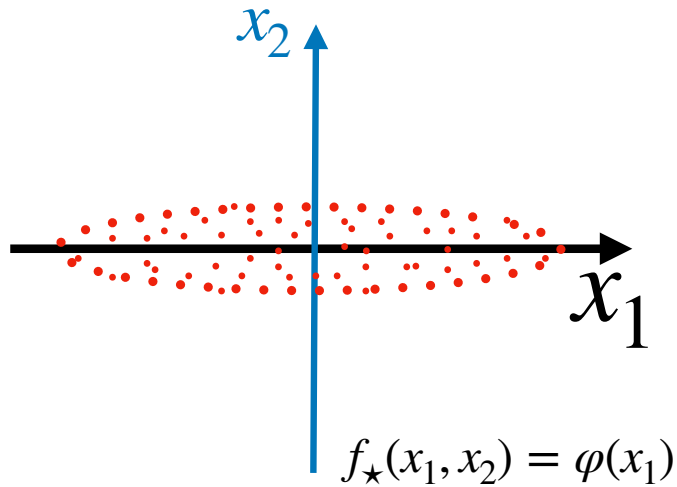


Figure: Anisotropic features:
$\kappa > 0$, $\mathrm{snr}_f > 1$

Implications

Adding isotropic noise in features (i.e., decreasing $\mathtt{snr}_{\mathtt{f}}$),
performance gap between NN and $\{\mathsf{RF}, \mathsf{NT}\}$ becomes larger.

Numerical simulations

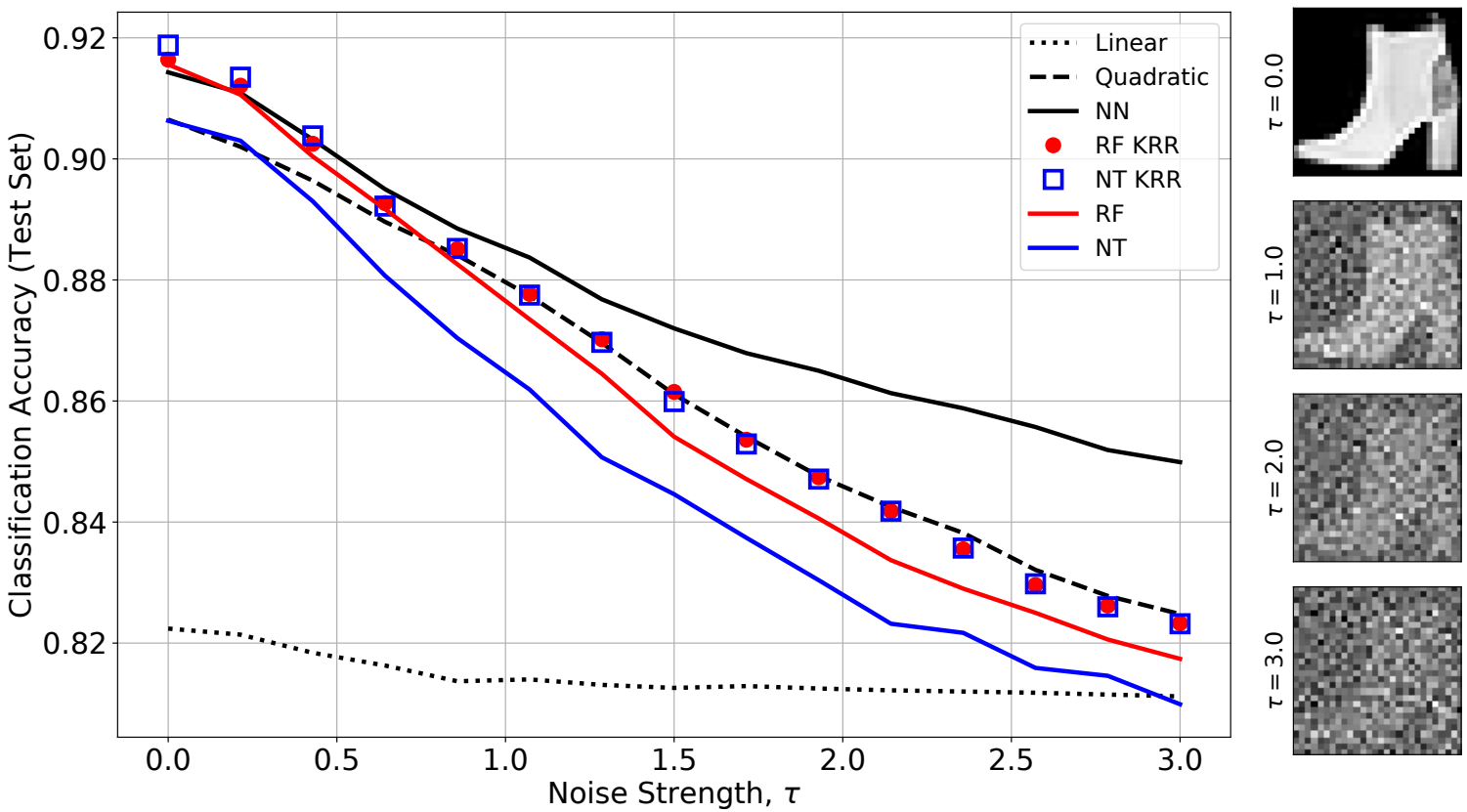


Figure: Underlying assumption: labels depend on low frequency components of images.

Message

In spiked features model, a controlling parameter of the performance gap between NN and $\{\mathrm{RF}, \mathrm{NT}\}$ is

$$\mathrm{snr}_{\mathrm{f}} = \text{Feature SNR} = \frac{\text{Signal features variance}}{\text{Junk features variance}}.$$

- Small $\mathrm{snr}_{\mathrm{f}}$, there is a large separation.
- Large $\mathrm{snr}_{\mathrm{f}}$, $\{\mathrm{RF}, \mathrm{NT}\}$ performs closer to NN.

Somewhat implicitly, NN first finds the signal features (PCA), and then perform kernel methods on these features.

$\mathrm{snr}_{\mathrm{f}} \neq \mathrm{SNR} = \|f_\star\|_{L^2}^2 / \mathbb{E}[\varepsilon^2]$

Thank you!