Limitations of Lazy Training of Two-layers Neural Networks

Theodor Misiakiewicz

Stanford University

December 11, 2019

Joint work with Behrooz Ghorbani, Song Mei, Andrea Montanari

Two-layers Neural Network (NN) model:

$$f_{\mathsf{NN}}(\boldsymbol{x}; \boldsymbol{a}, \boldsymbol{W}) = \sum_{i=1}^{N} \boldsymbol{a}_{i} \sigma(\boldsymbol{w}_{i}^{\mathsf{T}} \boldsymbol{x})$$

Lazy training regime or Kernel limit [Du et al.,18], [Chizat, Bach,18]

▶ Random Features (RF) model: (second layer linearization)

$$f_{\mathsf{RF}}(\boldsymbol{x}; \boldsymbol{a}) = \sum_{i=1}^{N} \boldsymbol{a}_{i} \sigma(\boldsymbol{w}_{i}^{\mathsf{T}} \boldsymbol{x})$$
 [Rahimi, Recht, 08]

$$f_{\mathsf{NT}}(\boldsymbol{x}; \boldsymbol{a}) = \sum_{i=1}^{N} \langle \boldsymbol{a}_i, \boldsymbol{x} \rangle \sigma'(\boldsymbol{w}_i^{\mathsf{T}} \boldsymbol{x})$$
 [Jacot et al., 18]

► Two-layers Neural Network (NN) model:

$$f_{\mathsf{NN}}(\boldsymbol{x}; \boldsymbol{a}, \boldsymbol{W}) = \sum_{i=1}^{N} \boldsymbol{a}_i \sigma(\boldsymbol{w_i}^\mathsf{T} \boldsymbol{x})$$

Lazy training regime or Kernel limit [Du et al.,18], [Chizat, Bach,18]

▶ Random Features (RF) model: (second layer linearization

$$f_{\mathsf{RF}}(\boldsymbol{x}; \boldsymbol{a}) = \sum_{i=1}^{N} \boldsymbol{a}_{i} \sigma(\boldsymbol{w}_{i}^{\mathsf{T}} \boldsymbol{x})$$
 [Rahimi, Recht, 08]

$$f_{\text{NT}}(\boldsymbol{x}; \boldsymbol{a}) = \sum_{i=1}^{N} \langle \boldsymbol{a}_i, \boldsymbol{x} \rangle \sigma'(\boldsymbol{w}_i^{\mathsf{T}} \boldsymbol{x})$$
 [Jacot et al., 18]

► Two-layers Neural Network (NN) model:

$$f_{\mathsf{NN}}(oldsymbol{x}; oldsymbol{a}, oldsymbol{W}) = \sum_{i=1}^{N} oldsymbol{a}_i \sigma(oldsymbol{w}_i^{\mathsf{T}} oldsymbol{x})$$

Lazy training regime or Kernel limit [Du et al.,18], [Chizat, Bach,18]

▶ Random Features (RF) model: (second layer linearization

$$f_{\mathsf{RF}}(\boldsymbol{x}; \boldsymbol{a}) = \sum_{i=1}^{N} \boldsymbol{a}_{i} \sigma(\boldsymbol{w}_{i}^{\mathsf{T}} \boldsymbol{x})$$
 [Rahimi, Recht, 08]

▶ Neural Tangent (NT) model: (first layer linearization)

$$f_{\mathsf{NT}}(\boldsymbol{x}; \boldsymbol{a}) = \sum_{i=1}^{N} \langle \boldsymbol{a}_i, \boldsymbol{x} \rangle \sigma'(\boldsymbol{w}_i^{\mathsf{T}} \boldsymbol{x})$$
 [Jacot et al., 18]

2/7

► Two-layers Neural Network (NN) model:

$$f_{\mathsf{NN}}(\boldsymbol{x}; \boldsymbol{a}, \boldsymbol{W}) = \sum_{i=1}^{N} \boldsymbol{a}_i \sigma(\boldsymbol{w_i}^\mathsf{T} \boldsymbol{x})$$

Lazy training regime or Kernel limit [Du et al.,18], [Chizat, Bach,18]

▶ Random Features (RF) model: (second layer linearization)

$$f_{\mathsf{RF}}(\boldsymbol{x}; \boldsymbol{a}) = \sum_{i=1}^{N} \boldsymbol{a}_{i} \sigma(\boldsymbol{w}_{i}^{\mathsf{T}} \boldsymbol{x})$$
 [Rahimi, Recht, 08]

$$f_{\mathsf{NT}}(\boldsymbol{x}; \boldsymbol{a}) = \sum_{i=1}^{N} \langle \boldsymbol{a}_i, \boldsymbol{x} \rangle \sigma'(\boldsymbol{w}_i^{\mathsf{T}} \boldsymbol{x})$$
 [Jacot et al., 18]

► Two-layers Neural Network (NN) model:

$$f_{\mathsf{NN}}(\boldsymbol{x}; \boldsymbol{a}, \boldsymbol{W}) = \sum_{i=1}^{N} \boldsymbol{a}_{i} \sigma(\boldsymbol{w_{i}}^{\mathsf{T}} \boldsymbol{x})$$

Lazy training regime or Kernel limit [Du et al.,18], [Chizat, Bach,18]

▶ Random Features (RF) model: (second layer linearization)

$$f_{\mathsf{RF}}(\boldsymbol{x}; \boldsymbol{a}) = \sum_{i=1}^{N} \boldsymbol{a}_{i} \sigma(\boldsymbol{w}_{i}^{\mathsf{T}} \boldsymbol{x})$$
 [Rahimi, Recht, 08]

$$f_{\mathsf{NT}}(\boldsymbol{x}; \boldsymbol{a}) = \sum_{i=1}^{N} \langle \boldsymbol{a}_i, \boldsymbol{x} \rangle \sigma'(\boldsymbol{w}_i^{\mathsf{T}} \boldsymbol{x})$$
 [Jacot et al., 18]

Goal: compare these three models.

Setting:

$$\mathbf{x}_i \sim \mathrm{N}(0, \mathbf{I}_d)$$
,

$$y_i = f_*(oldsymbol{x}_i) \equiv \langle oldsymbol{x}_i, oldsymbol{B} oldsymbol{x}_i
angle + b_0, \qquad ext{with } oldsymbol{B} \succeq 0.$$

- Activation function $\sigma(x) = x^2$
- ▶ N number of neurons: $N/d = \rho$ and d large (high-dimensional regime)

We compare the population squared loss:

$$R_{M,N}(f^*) = \min_{\hat{f} \in \mathcal{F}_M} \mathbb{E}_{oldsymbol{x}} \Big\{ \Big(f_*(oldsymbol{x}) - \hat{f}(oldsymbol{x}) \Big)^2 \Big\}, \qquad M \in \{\mathsf{RF}, \mathsf{NT}\}$$

$$R_{\mathsf{NN},N}(f^*) = \lim_{t o \infty} \lim_{arepsilon o 0} \mathbb{E}_{oldsymbol{x}} \Big\{ \Big(f_*(oldsymbol{x}) - \hat{f}_{\mathrm{SGD}}(oldsymbol{x}; t/arepsilon, arepsilon) \Big)^2 \Big\}$$

Goal: compare these three models.

Setting:

 $ightharpoonup x_i \sim \mathrm{N}(0, \mathbf{I}_d)$,

$$y_i = f_*(\boldsymbol{x}_i) \equiv \langle \boldsymbol{x}_i, \boldsymbol{B} \boldsymbol{x}_i \rangle + b_0, \qquad \text{with } \boldsymbol{B} \succeq 0$$

- Activation function $\sigma(x) = x^2$
- ▶ N number of neurons: $N/d = \rho$ and d large (high-dimensional regime)

We compare the population squared loss:

$$egin{aligned} R_{M,N}(f^*) &= \min_{\hat{f} \in \mathcal{F}_M} \mathbb{E}_{m{x}} \Big\{ \Big(f_*(m{x}) - \hat{f}(m{x}) \Big)^2 \Big\}, \qquad M \in \{\mathsf{RF}, \mathsf{NT}\} \ R_{\mathsf{NN},N}(f^*) &= \lim_{t \to \infty} \lim_{\epsilon \to 0} \mathbb{E}_{m{x}} \Big\{ \Big(f_*(m{x}) - \hat{f}_{\mathrm{SGD}}(m{x}; t/arepsilon, arepsilon) \Big)^2 \Big\} \end{aligned}$$

Goal: compare these three models.

Setting:

 $ightharpoonup x_i \sim \mathrm{N}(0, \mathbf{I}_d)$,

$$y_i = f_*(\boldsymbol{x}_i) \equiv \langle \boldsymbol{x}_i, \boldsymbol{B} \boldsymbol{x}_i \rangle + b_0, \qquad \text{with } \boldsymbol{B} \succeq 0$$

- Activation function $\sigma(x) = x^2$
- N number of neurons: $N/d = \rho$ and d large (high-dimensional regime)

We compare the population squared loss:

$$R_{M,N}(f^*) = \min_{\hat{f} \in \mathcal{F}_M} \mathbb{E}_{m{x}} \Big\{ \Big(f_*(m{x}) - \hat{f}(m{x}) \Big)^2 \Big\}, \qquad M \in \{\mathsf{RF}, \mathsf{NT}\}$$

$$R_{\mathsf{NN},N}(f^*) = \lim_{t o \infty} \lim_{arepsilon o 0} \mathbb{E}_{oldsymbol{x}} \Big\{ \left(f_*(oldsymbol{x}) - \hat{f}_{\mathsf{SGD}}(oldsymbol{x}; t/arepsilon, arepsilon) \Big)^2 \Big\}$$

Results

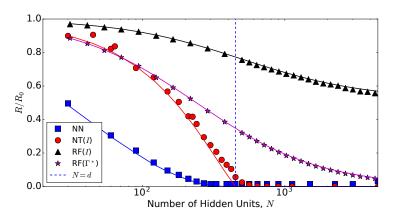


Figure: Population error in fitting a quadratic function in d=450 dimensions for random features (RF), neural tangent (NT), and SGD trained neural networks (NN). Lines are analytical predictions and dots are empirical results.

- ► RF model does not capture quadratic functions
- ► NN model

$$R_{\mathsf{NN},N}(f_*) = \min_{\boldsymbol{W} \in \mathbb{R}^{N \times d}} 2\|\boldsymbol{B} - \boldsymbol{W}\boldsymbol{W}^\mathsf{T}\|_F^2 = 2\sum_{i=N \wedge d+1}^d \lambda_i(\boldsymbol{B})^2$$

with
$$\lambda_1(\mathbf{B}) \geq \lambda_2(\mathbf{B}) \geq \ldots \geq \lambda_d(\mathbf{B}) \geq 0$$
.

 $lackbox{ NT model: } m{W} = [m{w}_1^\mathsf{T}, \dots, m{w}_N^\mathsf{T}]^\mathsf{T} \in \mathbb{R}^{N imes d} \ \mathrm{fixed \ at \ initialization}$

$$R_{\mathsf{NT},N}(f_\star) = \min_{oldsymbol{A} \in \mathbb{R}^{N imes d}} 2 \|oldsymbol{B} - oldsymbol{W} oldsymbol{A}^\mathsf{T} - oldsymbol{A} oldsymbol{W}^\mathsf{T} \|_F^2$$

for N < d, fit along W random subspace of dimension N.



- RF model does not capture quadratic functions
- NN model:

$$R_{\mathsf{NN},N}(f_*) = \min_{m{W} \in \mathbb{R}^{N imes d}} 2 \|m{B} - m{W}m{W}^\mathsf{T}\|_F^2 = 2 \sum_{i=N \wedge d+1}^d \lambda_i(m{B})^2$$

with
$$\lambda_1(\mathbf{B}) \geq \lambda_2(\mathbf{B}) \geq \ldots \geq \lambda_d(\mathbf{B}) \geq 0$$
.

▶ NT model: $W = [w_1^\intercal, \dots, w_N^\intercal]^\intercal \in \mathbb{R}^{N \times d}$ fixed at initialization

$$R_{\mathsf{NT},N}(f_\star) = \min_{oldsymbol{A} \in \mathbb{R}^{N imes d}} 2 \|oldsymbol{B} - oldsymbol{W} oldsymbol{A}^\mathsf{T} - oldsymbol{A} oldsymbol{W}^\mathsf{T} \|_F^2$$

for N < d, fit along W random subspace of dimension N.

5/7

- RF model does not capture quadratic functions
- ► NN model:

$$R_{\mathsf{NN},N}(f_*) = \min_{\boldsymbol{W} \in \mathbb{R}^{N \times d}} 2\|\boldsymbol{B} - \boldsymbol{W}\boldsymbol{W}^\mathsf{T}\|_F^2 = 2\sum_{i=N \wedge d+1}^d \lambda_i(\boldsymbol{B})^2$$

with
$$\lambda_1(\mathbf{B}) \geq \lambda_2(\mathbf{B}) \geq \ldots \geq \lambda_d(\mathbf{B}) \geq 0$$
.

 $lackbox{ riangle}$ NT model: $m{W} = [m{w}_1^\mathsf{T}, \dots, m{w}_N^\mathsf{T}]^\mathsf{T} \in \mathbb{R}^{N imes d}$ fixed at initialization

$$R_{\mathsf{NT},N}(f_\star) = \min_{oldsymbol{A} \in \mathbb{R}^{N imes d}} 2\|oldsymbol{B} - oldsymbol{W} oldsymbol{A}^\mathsf{T} - oldsymbol{A} oldsymbol{W}^\mathsf{T}\|_F^2$$

for N < d, fit along W random subspace of dimension N.

- RF model does not capture quadratic functions
- ► NN model:

$$R_{\mathsf{NN},N}(f_*) = \min_{\boldsymbol{W} \in \mathbb{R}^{N \times d}} 2\|\boldsymbol{B} - \boldsymbol{W}\boldsymbol{W}^\mathsf{T}\|_F^2 = 2\sum_{i=N \wedge d+1}^d \lambda_i(\boldsymbol{B})^2$$

with
$$\lambda_1(\mathbf{B}) \geq \lambda_2(\mathbf{B}) \geq \ldots \geq \lambda_d(\mathbf{B}) \geq 0$$
.

 $lackbox{NT model: } m{W} = [m{w}_1^\mathsf{T}, \dots, m{w}_N^\mathsf{T}]^\mathsf{T} \in \mathbb{R}^{N imes d} \ ext{fixed at initialization}$

$$R_{\mathsf{NT},N}(f_\star) = \min_{oldsymbol{A} \in \mathbb{R}^{N imes d}} 2\|oldsymbol{B} - oldsymbol{W} oldsymbol{A}^\mathsf{T} - oldsymbol{A} oldsymbol{W}^\mathsf{T}\|_F^2$$

for N < d, fit along W random subspace of dimension N.

5/7

Interpretation

► Fully trained NN learns the most important eigendirections, while the NT model remains confined to a random set of directions.

Neural networks are superior to linearized model such as RF and NT, because they can learn a good representation of the data

Mechanism more general: mixture of Gaussians, ReLu activation...

Interpretation

► Fully trained NN learns the most important eigendirections, while the NT model remains confined to a random set of directions.

Neural networks are superior to linearized model such as RF and NT, because they can learn a good representation of the data

Mechanism more general: mixture of Gaussians, ReLu activation...

Interpretation

► Fully trained NN learns the most important eigendirections, while the NT model remains confined to a random set of directions.

Neural networks are superior to linearized model such as RF and NT, because they can learn a good representation of the data

Mechanism more general: mixture of Gaussians, ReLu activation...

Thank you!

For further discussions, you can visit our poster:

 $\begin{array}{c} \text{Poster} \ \# \ 230 \\ \text{East Exhibition Hall B} + \text{C} \\ 5:00 \ \text{-} \ 7:00 \text{pm}, \ \text{Wednesday 11th} \end{array}$

If you have any questions: please email us at misiakie@stanford.edu