Limitations of Lazy Training of Two-layers Neural Networks

Theodor Misiakiewicz
[with Behrooz Ghorbani, Song Mei, Andrea Montanari]

Stanford University

December 2019
We focus on the functional class of two-layers neural networks on $x \in \mathbb{R}^d$:

$$F_{NN} = \left\{ f_{NN}(x) = \sum_{i=1}^{N} a_i \sigma \left( \langle w_i, x \rangle \right), \ a_i \in \mathbb{R}, \ w_i \in \mathbb{R}^d \right\}$$

Consider two ‘linearized’ function sub-classes of $F_{NN}$: fix $W = \{ w_i \}_{i \in [N]}$ with $w_i \in \mathbb{R}^d$

- **Random Features** [Rahimi-Recht 2008]

  $$F_{RF}(W) = \left\{ f_{RF}(x) = \sum_{i=1}^{N} a_i \sigma \left( \langle w_i, x \rangle \right), \ a_i \in \mathbb{R} \right\}$$

- **Neural Tangent Kernel** [Jacot et al. 2018]

  $$F_{NT}(W) = \left\{ f_{NT}(x) = \sum_{i=1}^{N} \langle a_i, x \rangle \sigma' \left( \langle w_i, x \rangle \right), \ a_i \in \mathbb{R}^d \right\}$$
Two interpretations:

- Random features as a computationally efficient approximation of a Kernel
- First order approximation of the model around initialization:

$$f_{NN}(x; a^t, w^t) \approx f_{NN}(x; a^0, w^0) + ((a^t, w^t) - (a^0, w^0))^T \nabla f_{NN}(x; a^0, w^0)$$

$$\approx 0 + f_{RF}(x; a^t - a^0, w^0) + f_{NT}(x; w^t - w^0, w^0)$$

Recent string of articles: gradient descent on NN

right scaling at initialization + sufficient overparametrization

⇒ parameters change hardly during training (aka lazy training [Chizat and Bach 2018])

and model behaves like the above linear model

Are deep neural networks simply efficient linear models?

Limitations of RF and NT in high-dimension [Ghorbani, Mei, M., Montanari 2019]:

If number of neurons $N \leq d^{k+1-\delta}$ for RF or $Nd \leq d^{k+1-\delta}$ for NN, then one cannot fit more than a degree-$k$ polynomial approximation.

Class of NN is in principle much richer than random-features models and can give way better fits. But what about effective networks trained by gradient descent? What are the advantage of learning features against keeping the features fixed in high dimension?

Concretely, can we characterize the gap between NN trained with SGD and random features?
**Setting:** data $x \sim N(0, I_d)$, fit the target function

$$f_*(x) = \langle x, Bx \rangle + b_0, \quad B \succeq 0$$

Look at the population squared loss (number of samples $\to \infty$)

$$R_{M,N}(f^*) = \min_{\hat{f} \in \mathcal{F}_M} \mathbb{E}\left\{ \left( f_*(x) - \hat{f}(x) \right)^2 \right\}, \quad M \in \{ \text{RF, NT} \}$$

$$R_{NN,N}(f^*; k, \varepsilon) = \mathbb{E}\left\{ \left( f_*(x) - \hat{f}_{\text{SGD}}(x; k, \varepsilon) \right)^2 \right\}$$

where $\hat{f}_{\text{SGD}}(x; k, \varepsilon)$ is the two layers neural network with parameters $(a_i, w_i)_{i \in [N]}$ obtained after $k$ steps of SGD with step size $\varepsilon$. 

Theodor Misiakiewicz (Stanford)

NeurIPS 2019, Vancouver

December 2019
Theoretical limits

We consider $N = \rho d$ and $\sigma(x) = x^2$.

Random Features:

$$\frac{R_{RF,N}(f_*)}{\|f_*\|_{L_2}^2} = 1 - \frac{\rho}{1 + \rho} \frac{\text{Tr}(B)^2}{d\|B\|_F^2}$$

Neural Tangent model:

$$\frac{R_{NT,N}(f_*)}{\|f_*\|_{L_2}^2} = (1 - \rho)^2 + \rho(1 - \rho) + \frac{\text{Tr}(B)^2}{d\|B\|_F^2}$$

SGD trained two-layer neural network:

$$\lim_{t \to \infty} \lim_{\epsilon \to 0} \frac{R_{NN,N}(f_*; t/\epsilon, \epsilon)}{\|f_*\|_{L_2}^2} = 1 - \frac{\sum_{i=1}^{d \wedge N} \lambda_i(B)^2}{\|B\|_F^2}$$
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.
Interpretation of results

- RF do not capture quadratic functions in this regime.
- NT fits the quadratic function along a random subspace determined by the fixed random weight $w_i$.
- Fully trained NN fits the $N$ principal eigendirections of $B$.

*Fully trained NN learns the most important eigendirections, surpassing the NT model which is confined to a random set of directions. Hence in this model, neural networks are superior to linearized models such as RF or NT, because they can learn a good representation of the data.*

Mechanism potentially more general. Next slides: results for mixture of two-gaussians and ReLu activation function.
Figure: Population error in fitting a mixture of two Gaussians in $d = 450$ dimensions. Lines are analytical predictions and dots are empirical results.
Figure: Population error in fitting a quadratic function in $d = 450$ dimensions with ReLu activation functions.