Statistical Foundations of Learning

Debarghya Ghoshdastidar

School of Computation, Information and Technology

Technical University of Munich

Regression with Infinite Width Neural Networks

Focus and Outline

Approximating (infinitely) wide neural networks

- With randomly initialised parameters, wide NN $\approx$ Gaussian process
- Linearisation of NN with Taylor approximation leads to a kernel model

Outline:

- Neural Network Gaussian process
- Random feature model
- Neural tangent kernel

Recap: Gaussian process

- $\{f(x) : x \in \mathbb{R}^p\}$ is a Gaussian process on $\mathbb{R}^p$ if there exist
  - a function $\mu : \mathbb{R}^p \to \mathbb{R}$ and
  - a positive semi-definite kernel $k : \mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R}$

  such that, for any m and $x_1, \ldots, x_m \in \mathbb{R}^p$,

$$\begin{bmatrix} f(x_1) \\ \vdots \\ f(x_m) \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mu(x_1) \\ \vdots \\ \mu(x_m) \end{bmatrix}, K \right) \quad \text{where } K_{ij} = k(x_i, x_j)$$

- Example: $f(x) = w^\top \phi(x)$ where $\phi : \mathbb{R}^p \to \mathbb{R}^N$ and $w \sim \mathcal{N}(0, I_{N\times N})$
  - Here, $\mu(x) = \mathbb{E}[f(x)] = 0$ and
  - $k(x, x') = \mathbb{E}[f(x)f(x')] = \phi(x)^\top \mathbb{E}[ww^\top]\phi(x') = \phi(x)^\top \phi(x')$

Recap: Gaussian process regression

- Given training $S = \{(x_i, y_i = f(x_i))\}_{i=1,\ldots,m}$

- Let x be a test point, $X = [x_1 \ldots x_m], y = [y_1 \ldots y_m]^\top, k(x, X) = [k(x, x_1) \ldots k(x, x_m)]$

- We know $(f(x), f(x_1), \ldots, f(x_m))$ is Gaussian with covariance $\begin{pmatrix} k(x,x) & k(x,X) \\ k(X,x) & K \end{pmatrix}$

- Hence, conditioned on $f(x_1) = y_1, \ldots, f(x_m) = y_m$
$$f(x) \mid \{f(X) = y\} \sim \mathcal{N}\left(k(x,X)K^{-1}y \;,\; k(x,x) - k(x,X)K^{-1}k(X,x)\right)$$

- Exercise: If $y_i = f(x_i) + \epsilon_i, \quad \epsilon_i \sim \mathcal{N}(0, \lambda)$, $\epsilon_1, \ldots, \epsilon_m$ iid, then show that
$$f(x) \mid y \sim \mathcal{N}\Big(\underbrace{k(x,X)(K+\lambda I)^{-1}y}_{\text{kernel ridge regression}} \;,\; k(x,x) - k(x,X)(K+\lambda I)^{-1}k(X,x)\Big)$$

2-layer NN with random initialisation

Theorem InfNN.1 (Infinite width NNs are GPs)

Define 2-layer NN $f : \mathbb{R}^p \to \mathbb{R}$, *with hidden layer of width* N *as*

$$f(x) = \frac{1}{\sqrt{N}} v^\top \sigma(Wx)$$

$W \in \mathbb{R}^{N \times p}$ *has entries* $W^{ij} \sim \mathcal{N}(0,1)$ *iid and* $v \in \mathbb{R}^N$ *has entries* $v^1, \ldots, v^N \sim_{iid} \mathcal{N}(0,1)$.
As $N \to \infty$, $\{f(x) : x \in \mathbb{R}^p\}$ *converges to a Gaussian process with* $\mu(x) = 0$ *and*

$$k(x, x') = \mathbb{E}_{(z,z')}[\sigma(z)\sigma(z')], \quad \text{where} \quad \begin{pmatrix} z \\ z' \end{pmatrix} \sim \mathcal{N}\left(0, \begin{pmatrix} \|x\|^2 & x^\top x' \\ x^\top x' & \|x'\|^2 \end{pmatrix}\right)$$

assuming $\sup_x k(x,x) < \infty$

- f could also be parameterised as $f(x) = v^\top \sigma(Wx)$, where $v^1, \ldots, v^N \sim_{iid} \mathcal{N}(0, \frac{1}{N})$

Proof

- View $W = \begin{bmatrix} (w^1)^\top \\ \vdots \\ (w^N)^\top \end{bmatrix}$, and for any x, define $z = \begin{bmatrix} z^1 \\ \vdots \\ z^N \end{bmatrix}$ with $z^i = x^\top w^i$

- Given x (or conditioned on x), observe that $z^1, \ldots, z^N \sim_{iid} \mathcal{N}(0, \|x\|^2)$
(why are they independent?)

- Since $z^1, \ldots, z^N$ are iid, $\sigma(z^1), \ldots, \sigma(z^N)$ are also iid

- $f(x)$ is a sum of independent random variables. By central limit theorem,

$$f(x) = \frac{1}{\sqrt{N}} \sum_{i=1}^{N} v^i \sigma(z^i) \xrightarrow{N\to\infty} \mathcal{N}\big(0, \mathbb{E}_z[\sigma^2(z)]\big)$$

- **Exercise:** For $\{x_1, \ldots, x_m\}$, show $\{f(x_1), \ldots, f(x_m)\}$ are jointly Gaussian with covariance k

NN-GP: Deep NNs are also GP

Theorem InfNN.2 (Neural Network Gaussian Process (NN-GP))

Consider a L*-layer NN* $f_L : \mathbb{R}^p \to \mathbb{R}$*, where* N_l *denotes width of* l*-th layer* $(N_0 = p, N_L = 1)$*.*
For $l = 1, \ldots, L-1$*, let the output of* l*-th layer be defined as*

$$f_l = \sigma(W_{l-1} f_{l-1}), \ \textit{where}\ W_{l-1} \in \mathbb{R}^{N_{l-1} \times N_l}\ \textit{with}\ (W_{l-1})^{ij} \sim \mathcal{N}\left(0, \frac{1}{N_{l-1}}\right)$$

and $f_L = v^\top f_{l-1}$ *with* $v^i \sim \mathcal{N}(0, \frac{1}{N_{L-1}})$
If $\min\{N_1, \ldots, N_{L-1}\} \to \infty$*, then* $\{f_L(x) : x \in \mathbb{R}^p\}$ *converges to a GP, called as a NN-GP, with* $\mu(x) = 0$ *and covariance kernel* $k_L(x, x')$*, which is recursively defined as*

$$k_0(x, x') = \frac{1}{p} x^\top x' \qquad k_l(x, x') = \mathbb{E}_{(z,z')}[\sigma(z)\sigma(z')]$$

where $\begin{pmatrix} z \\ z' \end{pmatrix} \sim \mathcal{N}\left(0, \begin{pmatrix} k_{l-1}(x,x) & k_{l-1}(x,x') \\ k_{l-1}(x',x) & k_{l-1}(x',x') \end{pmatrix}\right)$

Predictive distribution of NN-GP, and implications

- Let $\{f_L(x) : x \in \mathbb{R}^p\}$ be NN-GP with covariance kernel $k_L(x, x')$

- Given training data $\{(x_1, y_1), \ldots, (x_m, y_m)\}$, predictive distribution for test data x

$$f(x) \mid X, y \sim \mathcal{N}\Big(\underbrace{k_L(x, X)(K_L + \lambda I)^{-1} y}_{\text{kernel ridge regression}},\; k_L(x, x) - k_L(x, X)(K_L + \lambda I)^{-1} k_L(X, x)\Big)$$

- Prediction (from kernel) can be expressed $f(x) = \langle v, \phi_x \rangle$, Why is v, ϕ_x?
  - $\phi_x = z_{L-1}$, output of last hidden layer
  - v corresponds to trained weights of output layer

- NN-GP corresponds to infinitely wide lazy-trained NNs
  - only output layer is trained, and other layers are only randomly initialised

Random feature model / Lazy-trained NN

- Lazy-trained NN:
  - $f(x) = v^\top \sigma(Wx)$, where only v is trainable and W is randomly initialised but not trained
  - One can also define a deep lazy trained NN, with only output layer trained. But such a model can be approximated well by a single hidden layer lazy NN **(why?)**
- Random feature model:
  - A psd kernel $k : \mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R}$ has a random feature approximation if there exists a distribution $\mathcal{D}_w$ on $\mathbb{R}^p$ and a nonlinear map $\sigma : \mathbb{R} \to \mathbb{R}^k$ such that, for any $x, x' \in \mathbb{R}^p$,

$$k(x,x') = \mathbb{E}_{w \sim \mathcal{D}_w}[\sigma(w^\top x)\sigma(w^\top x')] = \lim_{N\to\infty} \frac{1}{N}\sum_{i=1}^{N} \sigma(w_i^\top x)\sigma(w_i^\top x') \quad w_1, \ldots, w_N \sim_{iid} \mathcal{D}_w$$

  - Example: Gaussian kernel

$$e^{-\frac{\|x-x'\|^2}{2}} = \underbrace{\mathbb{E}_{w\sim\mathcal{N}(0,I)}\left[\cos\left(w^\top(x-y)\right)\right]}_{\text{special case of Bochner's theorem}} = \mathbb{E}_w\left[\begin{pmatrix} \cos(w^\top x) \\ \sin(w^\top x) \end{pmatrix}^\top \begin{pmatrix} \cos(w^\top x') \\ \sin(w^\top x') \end{pmatrix}\right]$$

Focus and Outline

Approximating (infinitely) wide neural networks

- With randomly initialised parameters, wide NN $\approx$ Gaussian process
- Linearisation of NN with Taylor approximation leads to a kernel model

Outline:

- Neural Network Gaussian process
- Random feature model
- Neural tangent kernel

Recap: Taylor approximation of multivariate function

- Consider a function $g : \mathbb{R}^k \to \mathbb{R}$

- Let $\theta_0 \in \mathbb{R}^k$ and assume g is twice differentiable on a ball $\mathcal{B}$ around θ_0

- 2^{nd} order Taylor approximation of g:
  For any $\theta \in \mathcal{B}$,
  $$g(\theta) = g(\theta_0) + (\theta - \theta_0)^\top \nabla g(\theta_0) + \frac{1}{2}(\theta - \theta_0)^\top H(g(\xi))(\theta - \theta_0)$$
  for some ξ that lies on line segment joining θ and θ_0

  - $\nabla g(\theta_0) \in \mathbb{R}^k$ is gradient of $g(\cdot)$ computed at θ_0

  - $H(g(\xi)) \in \mathbb{R}^{k \times k}$ is Hessian of $g(\cdot)$ computed at ξ ... $[H(g(\theta))]^{ij} = \dfrac{\partial^2 g(\theta)}{\partial \theta^i \partial \theta^j}$

Linearisation of neural network (w.r.t. parameters)

- Consider 2-layer NN: $f(x) = \frac{1}{\sqrt{N}} v^\top \sigma(Wx)$ $\qquad v \in \mathbb{R}^N, W \in \mathbb{R}^{N \times p}$
  - We will view $f(x)$ as $f_x(\theta)$, function of parameters $\theta = \left(\{v^i\}_{i=1,\ldots,N}, \{W^{ij}\}_{i=1,\ldots,N;j=1,\ldots,p}\right)$
- Let $\theta_0 = (v_0, W_0)$ be the parameters at (random) initialisation. By Taylor's theorem

$$f_x(\theta) = f_x(\theta_0) + (\theta - \theta_0)^\top \nabla f_x(\theta_0) + \frac{1}{2}(\theta - \theta_0)^\top H(f_x(\xi))(\theta - \theta_0)$$

- Linear approximation of NN (linear w.r.t θ):

$$\tilde{f}_x(\theta) \;=\; f_x(\theta_0) + (\theta - \theta_0)^\top \nabla f_x(\theta_0) \;=\; \theta^\top \underbrace{\nabla f_x(\theta_0)}_{=:\,\phi_x} + \underbrace{\left(f_x(\theta_0) - \theta_0^\top \nabla f_x(\theta_0)\right)}_{=:\,b_{x,\theta_0}\ \text{constant w.r.t}\ \theta}$$

Closer look at $\phi_x = \nabla f_x(\theta_0)$

- $\nabla f_x(\theta) \in \mathbb{R}^{Np+N}$ is a vector with coordinates $\dfrac{\partial f_x}{\partial v^i}, \dfrac{\partial f_x}{\partial W^{ij}}, \; i = 1, \ldots, N; j = 1, \ldots, p$

$$\frac{\partial f_x}{\partial v^i} = \frac{1}{\sqrt{N}} \underbrace{\sigma\left(\sum_l W^{il} x^l\right)}_{\sigma(x^\top w^i)} \qquad \frac{\partial f_x}{\partial W^{ij}} = \frac{1}{\sqrt{N}} v^i x^j \underbrace{\sigma'\left(\sum_l W^{il} x^l\right)}_{\sigma'(x^\top w^i)}$$

$\ldots \sigma'(z) = \dfrac{d}{dz}\sigma(z)$, and $(w^i)^\top = i$-th row of W

- Consider the kernel $k(x, x') = \langle \phi_x, \phi_{x'} \rangle = \langle \nabla f_x(\theta_0), \nabla f_{x'}(\theta_0) \rangle$

$$\begin{aligned} k(x, x') &= \sum_{i=1}^{N} \frac{\partial f_x}{\partial v_0^i} \frac{\partial f_{x'}}{\partial v_0^i} + \sum_{i=1}^{N} \sum_{j=1}^{p} \frac{\partial f_x}{\partial W_0^{ij}} \frac{\partial f_{x'}}{\partial W_0^{ij}} \\ &= \frac{1}{N} \sum_{i=1}^{N} \sigma(x^\top w_0^i) \sigma(x'^\top w_0^i) + \frac{1}{N} \sum_{i=1}^{N} (v_0^i)^2 \sigma'(x^\top w_0^i) \sigma'(x'^\top w_0^i) \sum_{j=1}^{p} x^j x'^j \end{aligned}$$

Neural tangent kernel

- Assume Gaussian initialisation, $v_0^i \sim \mathcal{N}(0,1), W_0^{ij} \sim \mathcal{N}(0,1)$.
  Using law of large numbers,

$$\lim_{N\to\infty} k(x,x') = \underbrace{\mathbb{E}_{w\sim\mathcal{N}(0,I)}\left[\sigma(x^\top w)\sigma(x'^\top w)\right]}_{\text{kernel of NNGP}} + \underbrace{(x^\top x')\cdot\mathbb{E}_{w\sim\mathcal{N}(0,I)}\left[\sigma'(x^\top w)\sigma'(x'^\top w)\right]}_{\text{correction term}}$$

  - NNGP kernel $= \langle \nabla_v f_x, \nabla_v f_{x'} \rangle$ corresponds to random feature model (training only output layer)

  - correction term $= \langle \nabla_W f_x, \nabla_W f_{x'} \rangle$ arises from training first layer

- Above (limiting) kernel is neural tangent kernel (NTK)
  For NNs with more layers, NTK is defined recursively over layers

Closer look at b_{x,θ_0}

$$\tilde{f}_x(\theta) \;=\; \theta^\top \underbrace{\nabla f_x(\theta_0)}_{=:\,\phi_x} + \underbrace{\Big(f_x(\theta_0) - \theta_0^\top \nabla f_x(\theta_0)\Big)}_{=:\,b_{x,\theta_0}\ \text{constant w.r.t}\ \theta}$$

- For ReLU, $\sigma(z) = \max\{z, 0\}$, verify that $\sigma'(z) = z\sigma(z)$
  - **Exercise:** Using above fact, show that $b_{x,\theta_0} = 0$ for all x, θ_0
  - Caveat: ReLU is not differentiable everywhere. So Taylor approx. needs care
- For arbitrary σ, we use Taylor approximation of $f(\theta_0), \nabla f(\theta_0)$ around origin

$$b_{x,\theta_0} = f_x(\theta_0) - \theta_0^\top \nabla f_x(\theta_0) = \left(f(0) + \theta_0^\top \nabla f(0) + \frac{1}{2}\theta_0^\top H(f(\xi_1))\theta_0\right) - \theta_0^\top \left(\nabla f(0) + H(f(\xi_2))\theta_0\right)$$

for some ξ_1, ξ_2 between 0 and θ_0

Error of NTK approximation

$$f_x(\theta) - \underbrace{\theta^\top \nabla f_x(\theta_0)}_{\langle \theta, \phi_x \rangle} = b_{x,\theta_0} + \frac{1}{2}\theta_0^\top H(f_x(\xi))\theta_0 = \theta_0^\top \left(\frac{1}{2} H(f_x(\xi_1)) - H(f_x(\xi_2)) + \frac{1}{2} H(f_x(\xi)) \right) \theta_0$$

where $\xi_1 = c_1\theta_0;\ \xi_2 = c_2\theta_0;\ \xi = (c\theta + (1-c)\theta_0);\ c, c_1, c_2 \in [0,1]$

- Main proof idea:
  - $H(f_x(\xi))$ is sparse and, if σ is "smooth" and $\|x\| = O(1)$, then $\|H(f_x(\xi))\| = O\left(\frac{\|\xi\|_\infty}{\sqrt{N}}\right)$
  - In interpolating regime (opt train error = 0), $\|\theta_t - \theta_0\| = O(1)$ for all iterations of gradient descent. Hence, $\|\xi\|_\infty, \|\xi_1\|_\infty, \|\xi_2\|_\infty = O(\|\theta_0\|_\infty)$.
- Caveat: With $\mathcal{N}(0,1)$ initialisation of θ_0, we get $\|\theta_0\|_\infty = O(\ln N)$, $\|\theta_0\|_2^2 \approx N(p+1)$
  Also, above still gives $\theta_0^\top H(f_x(\xi))\theta_0 = O(1)$, and not vanishing as $N \to \infty$

ReLU Neural Tangent Kernel

- Relu NTK on $\mathcal{X}$ = unit sphere in $\mathbb{R}^p$
  - Exact form: $k(x,x') = x^\top x' \left(\frac{1}{2} - \frac{\arccos(x^\top x')}{2\pi}\right)$ (Bietti, Mairal, *NeurIPS*, 2019)
  - RKHS is same as the RKHS of Laplace kernel $e^{-\gamma\|x-\bar{x}\|}$ (Chen, Xu, *ICLR*, 2021)
    Hence, ReLU NTK is universal kernel on unit sphere in $\mathbb{R}^p$
- One can use previous generalisation error bounds for kernels to comment on generalisation in infinite width neural networks