Estimation of the Mean Function of Functional Data via Deep Neural Networks

Guanqun Cao

AUBURN UNIVERSITY

Spring School on Models and Data 2021
University of South Carolina
April 10th, 2021

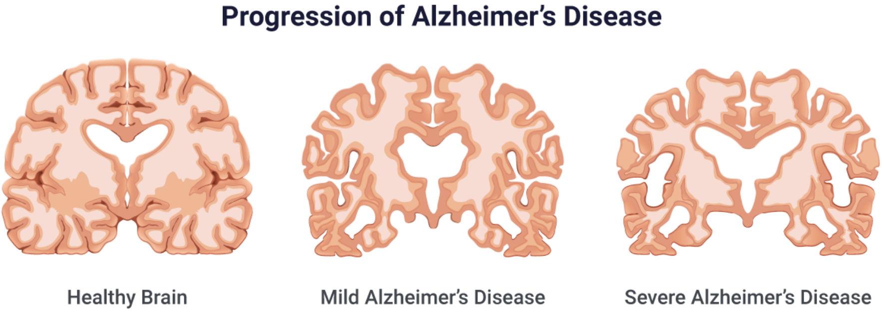
Acknowledgement

- Collaborators
  - Mr. Shuoyang Wang, Auburn University
  - Dr. Zuofeng Shang, New Jersey Institute of Technology
- Partially funded by NSF award DMS-1736470
- ENAR Distinguished Student Paper Award 2021

Outline

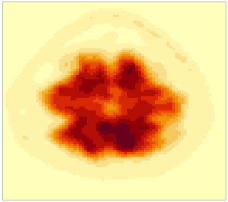
1. **Introduction**
   - Functional regression model
   - Deep neural networks
2. **Methods**
   - Estimation of mean function via deep neural networks
   - Non-asymptotic convergency rate
3. **Real data analysis**

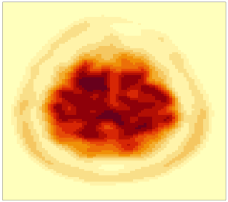
Brain Imaging

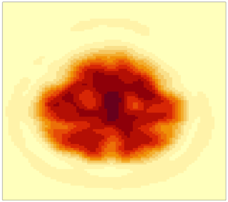


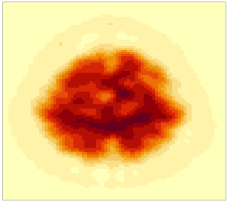
Source: https://www.caring.com/caregivers/alzheimers/

Positron Emission Tomography (PET) Images









Functional regression model

$$Y_{ij} = f_0(\mathbf{X}_j) + \eta(\mathbf{X}_j) + \epsilon_i(\mathbf{X}_j), \quad i = 1, 2, \ldots, n, j = 1, 2, \ldots, N,$$

- $f_0 : \mathbb{R}^d \to \mathbb{R}$, $\mathrm{E}(\mathrm{Y_{ij}}) = \mathrm{f_0}(\mathbf{X}_\mathrm{j})$; $\mathbf{X}_j \in \mathbb{R}^d$;
- $\eta(\cdot)$: individual curve variations; zero mean Gaussian process;
- $\epsilon_i(\cdot)$: zero mean measurement error;
- n: sample size;
- N: number of observations for each subject.

How to estimate mean function $f_0(\cdot)$?

Deep neural networks

Definition

$$f(\mathbf{x}) = \mathbf{W}_L\sigma(W_{L-1}\ldots\sigma(\mathbf{W}_2\sigma(\mathbf{W}_1\mathbf{x}+v_1)+v_2)\ldots+v_{L-1}),$$

- $d = d_0 \to d_1 \to \ldots, \ldots \to d_L \to d_{L+1} = 1$;
- $\sigma(x) = \max(x, 0)$: ReLU activation function;
- $\mathbf{W}_\ell : p_\ell \times p_{\ell+1}$ weight matrix;

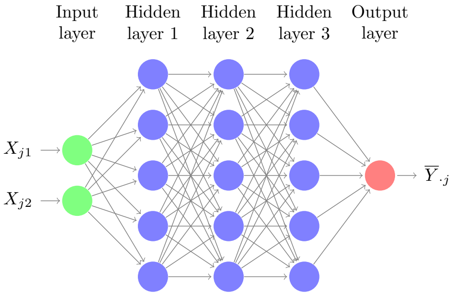
Sparse network space:

$$\mathcal{F}_{DNN}(L, \mathbf{p}, s) = \left\{ f : \max_{\ell=0,\ldots,L} \|\mathbf{W}_\ell\|_\infty + |\mathbf{v}_\ell|_\infty \leq 1, \sum_{\ell=0}^{L} \|\mathbf{W}_\ell\|_0 + |\mathbf{v}_\ell|_0 \leq s \right\}$$

Structured compositions of Hölder Functions

- $g_i : [a_i, b_i]^{d_i} \to [a_{i+1}, b_{i+1}]^{d_{i+1}}$, $g_i = \left(g_{ij}\right)^\top_{j=1,\ldots,d_{i+1}}$, ambient
- Each component g_{ij} is β_i-Hölder function with at most t_i-variate: $\left\{g_{ij} \in \mathcal{C}^{\beta_i}_{t_i}\left([a_i, b_i]^{t_i}, K_i\right), |a_i|, |b_i| \leq K_i\right\}$ intrinsic
- True underlying function space: $\mathcal{G}\left(q, \{d_i, t_i, \beta_i, K_i\}_{i\in[q]}\right)$ consists of $f = g_q \circ g_{q-1} \circ \ldots \circ g_1 \circ g_0$
- Smoothness of $f_i = g_q \circ g_{q-1} \circ \ldots \circ g_i$ $\beta^*_i := \beta_i \prod_{k=i+1}^{q}(\beta_k \wedge 1)$

Functional regression via Deep neural networks



Empirical risk minimization

$$\widehat{f} = \arg\min_{f \in \mathcal{F}_{DNN}} \frac{1}{N} \sum_{j=1}^{N} \left\{ \overline{Y}_{\cdot j} - f\left(\mathbf{X}_j\right) \right\}^2,$$

where $\overline{Y}_{\cdot j} = n^{-1} \sum_{i=1}^{n} Y_{ij}$, $\mathbf{X}_j = (X_{j1}, \ldots, X_{jd})$

Non-asymptotic convergency rate

Theorem 1

Under mild assumptions, with probability greater than $(1-\frac{2}{nN^{\varrho}})^{\log(nN^{\varrho})+1} \to 1$, we have

$$\|\widehat{f}-f_0\|_N^2 \le c(nN^{\varrho})^{-\frac{\theta}{\theta+1}}\log^6(nN^{\varrho}),$$

where $\varrho \ge 0$, $\theta = \min_{i=0,\dots,q} \frac{2\beta_i^*}{t_i}$ and c depend on true function class of f_0.

- $(nN^{\varrho})^{-\frac{\theta}{\theta+1}} = (nN^{\varrho})^{-\alpha}$ and $\alpha = \min_{i=0,\dots,q} \frac{2\beta_i^*}{2\beta_i^*+t_i}$
- If $\varrho = 0$, $\|\widehat{f}-f_0\|_N^2 \le cn^{-\frac{\theta}{\theta+1}}\log^6(n)$

ADNI

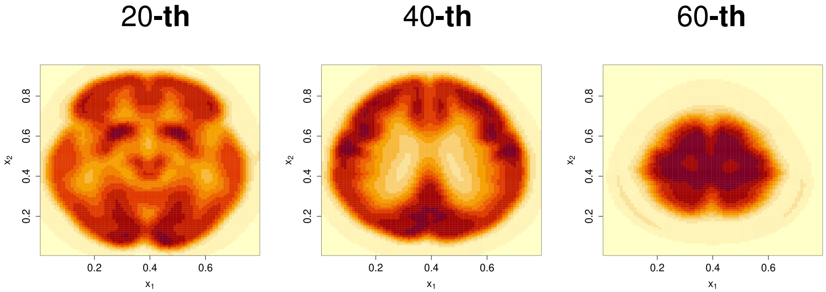
- 79 patients from the AD group.
  - 33 females
  - 46 males

ADNI

ADNI
ALZHEIMER'S DISEASE
NEUROIMAGING INITIATIVE

- 79 patients from the AD group.
  - 33 females
  - 46 males
- reoriented into $79 \times 95 \times 69$ voxels.
- each patient has 69 sliced 2D images with 79×95.

Recovery (79×95) from 3D scans



- Wang, S., Cao, G. and Shang, Z. (2020) Estimation of the Mean Function of Functional Data via Deep Neural Networks. *arxiv.org*

R Package

https://github.com/FDASTATAUBURN/FDADNN.

Assumptions

(A1) The true regression function f_0 has a composition structure.

Deep and wide neural networks

(A2) $\widehat{f} \in \mathcal{F}(L, \mathbf{p}, s)$, s.t.

- Depth: $L \asymp \log(nN^{\varrho})$, $\varrho \geq 0$;

Assumptions

(A1) The true regression function f_0 has a composition structure.

Deep and wide neural networks

(A2) $\widehat{f} \in \mathcal{F}(L, \mathbf{p}, s)$, s.t.

- Depth: $L \asymp \log(nN^{\varrho})$, $\varrho \geq 0$;
- Width: $\min_{l=1,\dots,L} p_\ell \asymp (nN^{\varrho})^{\frac{1}{\theta+1}}$, where $\theta = \min_{i=0,\dots,q} \frac{2\beta_i^*}{t_i}$

Assumptions

(A1) The true regression function f_0 has a composition structure.

Deep and wide neural networks

(A2) $\widehat{f} \in \mathcal{F}(L, \mathbf{p}, s)$, s.t.

- Depth: $L \asymp \log(nN^{\varrho})$, $\varrho \geq 0$;
- Width: $\min_{l=1,\ldots,L} p_\ell \asymp (nN^{\varrho})^{\frac{1}{\theta+1}}$, where $\theta = \min_{i=0,\ldots,q} \frac{2\beta_i^*}{t_i}$
- sparsity: $s \asymp (nN^{\varrho})^{\frac{1}{\theta+1}}$;

Assumptions

(A1) The true regression function f_0 has a composition structure.

Deep and wide neural networks

(A2) $\widehat{f} \in \mathcal{F}(L, \mathbf{p}, s)$, s.t.

- Depth: $L \asymp \log(nN^{\varrho})$, $\varrho \geq 0$;
- Width: $\min_{l=1,\ldots,L} p_{\ell} \asymp (nN^{\varrho})^{\frac{1}{\theta+1}}$, where $\theta = \min_{i=0,\ldots,q} \frac{2\beta_i^*}{t_i}$
- sparsity: $s \asymp (nN^{\varrho})^{\frac{1}{\theta+1}}$;

Assumptions

(A3) The maximal eigenvalue of the kernel matrix is $O(N^{-\varrho})$ for some constant $\varrho \geq 0$.