# Deep Neural Network Classifier for Multi-Dimensional Functional Data

Shuoyang Wang

AUBURN UNIVERSITY

Yale University

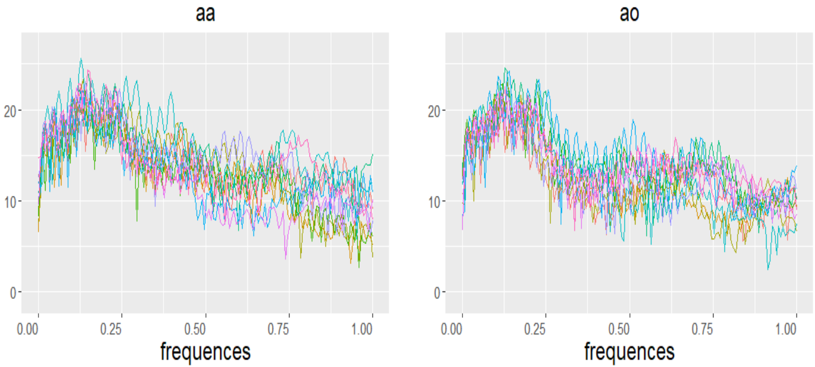
*Joint work with Guanqun Cao (AU) & Zuofeng Shang (NJIT)*

2022 Symposium on Data Science and Statistics

*Jun 8th, 2022*

Partially funded by NSF award DMS-1736470

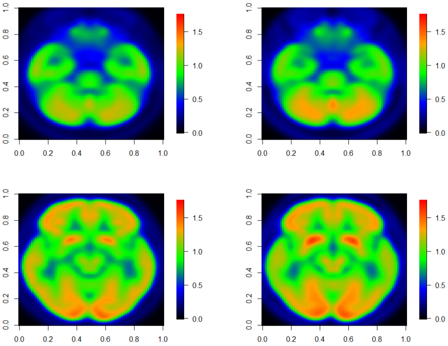
# Motivating example: speech data




aa $\Longleftrightarrow$ dark
ao $\Longleftrightarrow$ water

# Motivating example: Alzheimer disease

**EMCI** **AD**




EMCI : early mild cognitive impairment
AD : Alzheimer disease

# Literature Review

## Functional data classification (one-dimensional)

- k-nearest neighbor classifiers [Biau et al., 2005, Biau et al., 2010].

# Literature Review

## Functional data classification (one-dimensional)

- k-nearest neighbor classifiers [Biau et al., 2005, Biau et al., 2010].
- logistic regression [Araki et al., 2009].

# Literature Review

**Functional data classification (one-dimensional)**

- k-nearest neighbor classifiers [Biau et al., 2005, Biau et al., 2010].
- logistic regression [Araki et al., 2009].
- Functional discriminant analysis [Delaigle and Hall, 2012].

# Literature Review

**Functional data classification (one-dimensional)**

- k-nearest neighbor classifiers [Biau et al., 2005, Biau et al., 2010].
- logistic regression [Araki et al., 2009].
- Functional discriminant analysis [Delaigle and Hall, 2012].
- Bayesian classifier [Dai et al., 2017].

Which classifier(s) can achieve optimality?

# Framework for functional data classification

- Class label $Y = \{1, -1\}$
- $X(\boldsymbol{s}) = \sum_{j=1}^{\infty} \xi_j \psi_j(\boldsymbol{s}) \Longleftrightarrow (\xi_1, \xi_2, \dots)^\intercal$
- Density for functional observations $h = (h_1, h_{-1}) \in \mathcal{H}$
  - Domain for $h$ has infinite dimension
  - Likelihood ratio $h_1/h_{-1}$ has composition structures of $q$ layers
  - The $u$-th composition layer:
    - $\beta_u$-Hölder functions
    - Intrinsic dimension bounded by $c_u < \infty$
- Noise condition with $\alpha$

$$\mathrm{P}\left(\left|\frac{h_1 - h_{-1}}{h_1 + h_{-1}}\right| \leq x\right) \leq Cx^{\alpha}, \qquad \forall x > 0. \tag{1}$$

# Statistical optimality for binary classification

## Misclassification risk for $\widehat{G}$

$$R_h(\widehat{G}_n) := \mathbb{E}_h[\mathbb{I}\{\widehat{G}_n(X) \neq Y\} | \{X_i, Y_i\}_{i=1}^n]$$

## Excess misclassification risk (EMR)

$$\mathcal{E}_h(\widehat{G}) := \mathbb{E}[R_h(\widehat{G}_n) - R_h(G^*)], \quad G^* \textit{ is the Bayesian classifier}$$

## Minimax excess misclassification risk (MEMR)

$$\max_{h \in \mathcal{H}} \mathcal{E}_h(\widetilde{G}) \asymp \inf_{\widehat{G}} \max_{h \in \mathcal{H}} \mathcal{E}_h(\widehat{G})$$

$\widetilde{G}$ is statistically optimal!

# Optimality for Functional Deep Neural Network (FDNN) Classifier

## MEMR lower bound

$$\inf_{\widehat{G}} \sup_{h \in \mathcal{H}} E\left[R_h(\widehat{G}) - R_h(G^*)\right] \gtrsim \left(\frac{1}{n}\right)^{S_0}.$$

## EMR for FDNN classifier

$$\sup_{h \in \mathcal{H}} E\left[R_h(\widehat{G}^{FDNN}) - R_h(G^*)\right] \lesssim \left(\frac{\log^3 n}{n}\right)^{S_0},$$

$S_0 = \min_{u=0,\ldots,q} \frac{\beta_u^*(\alpha+1)}{\beta_u^*(\alpha+2)+c_u}$, $\beta_u^* = \beta_u \prod_{w=u+1}^{q} \beta_w \wedge 1$.

$c_u \uparrow \Longleftrightarrow$ easier to classify (Bayesian) $\Longleftrightarrow$ slower convergence

$c_u \downarrow \Longleftrightarrow$ harder to classify (Bayesian) $\Longleftrightarrow$ faster convergence

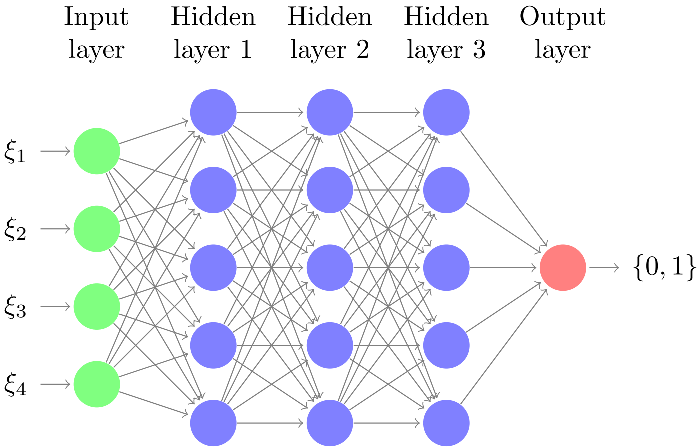
# Data driven architecture

## Hyperparameters for DNN

- Number of inputs: $\left(n\log^{-3} n\right)^{S_0/\rho} \lesssim J \lesssim (n\log^{-3} n)^{S_1}$;
- Depth: $L \asymp \log n$;
- Width: $\max_{1\leq \ell \leq L} p_\ell \asymp (n\log^{-3} n)^{S_1}$;
- Maximal weight: $B \asymp (n\log^{-3} n)^{S_2}$,

$$S_1 = \max_{0\leq u\leq q} \frac{c_u}{\beta_u^*(\alpha+2)+c_u}, S_2 = \min_{0\leq u\leq q} \frac{1}{\beta_u^*(\alpha+2)+c_u}, \rho > 0.$$

# Deep neural network structures ($J = 4$)



# Functional deep neural networks (FDNN) classifier

## Deep neural networks

Deep neural networks with ReLU active function and hinge loss

# Functional deep neural networks (FDNN) classifier

## Deep neural networks

Deep neural networks with ReLU active function and hinge loss

## Projection scores as inputs

- Use the first $J$ projection scores $\{\xi_i^{(j)}\}_{j=1}^J$ as inputs of DNN
- Classification rule: $\widehat{G}^{FDNN}(Z) = \mathbb{I}\left(\widehat{f}_\phi(\boldsymbol{z}) < 0\right)$

# Functional deep neural networks (FDNN) classifier

## Deep neural networks

Deep neural networks with ReLU active function and hinge loss

## Projection scores as inputs

- Use the first $J$ projection scores $\{\xi_i^{(j)}\}_{j=1}^{J}$ as inputs of DNN
- Classification rule: $\widehat{G}^{FDNN}(Z) = \mathbb{I}\left(\widehat{f}_{\phi}(\boldsymbol{z}) < 0\right)$

## Choice of ingredients

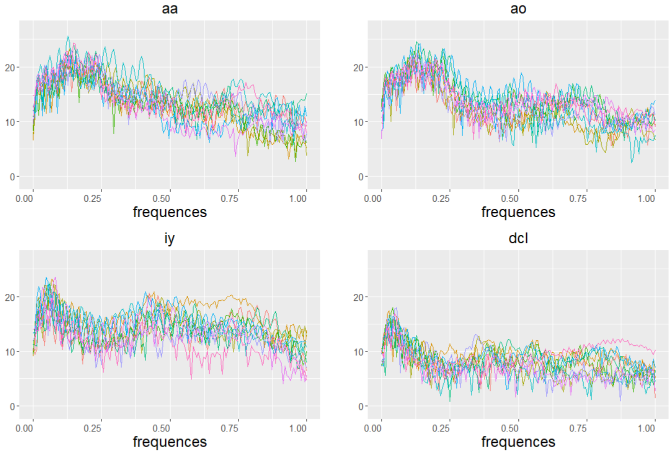
- Data driven network structure
- Flexible choice of basis functions

# Network hyperparameter selection

## Data-splitting method

- Step 1. Randomly divide the whole sample $\left(\{\xi_i^{(j)}\}_{j=1}^{J}, Y_i\right)$'s into two subsets indexed by $\mathcal{I}_{train}$ and $\mathcal{I}_{test}$, respectively, with about $|\mathcal{I}_{train}| = 0.8n$ and $|\mathcal{I}_{test}| = 0.2n$.
- Step 2. For each $(L, J, \boldsymbol{p}, B)$, we train a DNN based on subset $\mathcal{I}_{train}$, and then calculate the testing error based on subset $\mathcal{I}_{test}$ as
$$\mathrm{err}(L, J, \boldsymbol{p}, B) = \frac{1}{|\mathcal{I}_{test}|} \sum_{i \in \mathcal{I}_{test}} I(\widehat{f}_{L,J,\boldsymbol{p},B}(\widehat{\xi}_i^J) Y_i < 0).$$
- Step 3. Choose $(L, J, \boldsymbol{p}, B)$, possibly from a preselected set, to minimize $\mathrm{err}(L, J, \boldsymbol{p}, B)$.

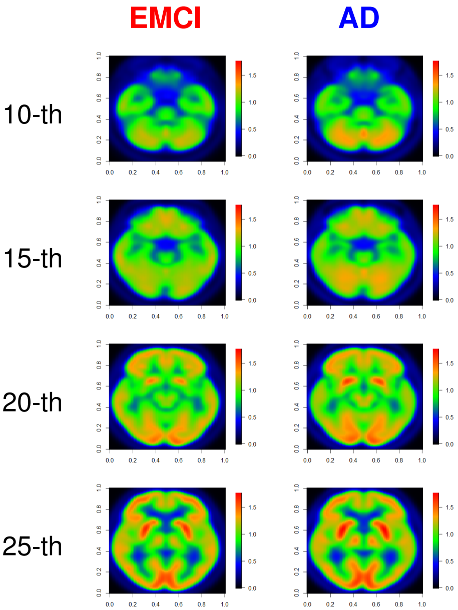
# Speech data



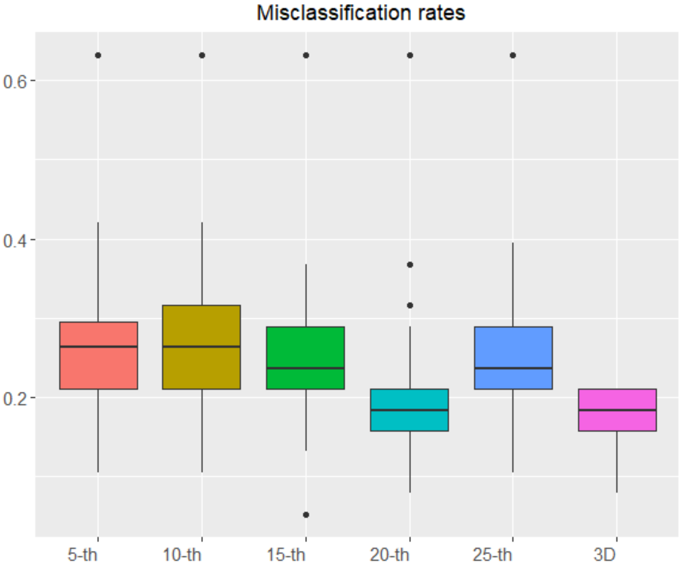
## Speech data misclassification rates

| Phonemes | FDNN | QD | NB |
|---|---|---|---|
| "aa" vs "ao" | 20.744 | 25.402 | 25.378 |
| "aa" vs "iy" | 0.193 | 0.288 | 0.273 |
| "ao" vs "iy" | 0.183 | 0.578 | 0.232 |
| "ao" vs "dcl" | 0.229 | 0.391 | 0.472 |

# ADNI data



# ADNI data misclassification rates



## Takeaways

**Motivation:** Whether and which of existing functional classifiers are statistically optimal? How to construct an optimal functional classifier with better performances?

## Takeaways

**Motivation:** Whether and which of existing functional classifiers are statistically optimal? How to construct an optimal functional classifier with better performances?

**Novelty:** Establish the first minimax theory for multi-dimensional functional data classification problem under non-Gaussian assumption.

## Takeaways

**Motivation:** Whether and which of existing functional classifiers are statistically optimal? How to construct an optimal functional classifier with better performances?

**Novelty:** Establish the first minimax theory for multi-dimensional functional data classification problem under non-Gaussian assumption.

**Importance:** Understand how the “best" functional classifier looks like, as well as provide a guidance to find the “best" classifiers.

рахмат
danke
謝謝
spas
ngiyabonga
teşekkür ederim
Баярлалаа
mersi
kia ora
barka
welalin
tack
mahalo
tapadh leat
спасибо
faafetai lava
vinaka
спасибі
blagodaram
dank je
misaotra
matondo
paldies
grazzi
gracias
хвала
asante
manana
obrigada
tenki
murakoze
kiitos
dankie
nanni
nandri
dhanyavad
mauruuru
hvala
köszönöm
thank you
dankon
aciū
akun
bedankt
bayarlalaa
gràcie
enkosi
dziękuję
djiere dieuf
tau
mochchakkeram
дякую
mamnun
chokrane
chnorakaloutioun
gratias ago
gràcies
sulpáy
taiku
go raibh maith agat
sobodi
děkuji
obrigado
sagolun
sukriya
kop khun krap
arigatô
takk
dakujem
trugarez
mèsi
didi madloba
ありがとう
tanemirt
grazie
najis tuke
kam sah hamnida
rahmat
terima kasih
rahmet
diolch
dhanyavadagalu
shukriya
merce
мерси
তোমাকে ধন্যবাদ
감사합니다
xièxie
ευχαριστώ
merci

# References I

Araki, Y., Konishi, S., Kawano, S., and Matsui, H. (2009).
Functional logistic discrimination via regularized basis expansions.
*Communications in Statistics. Theory and Methods*, 38(16-17):2944–2957.

Biau, G., Bunea, F., and Wegkamp, M. (2005).
Functional classification in hilbert spaces.
*IEEE Trans. Info. Theory*, 51:2163â2172.

Biau, G., Cérou, F., and Guyader, A. (2010).
Rates of convergence of the functional k-nearest neighbor estimate.
*IEEE Trans. Info. Theory*, 56:2034–2040.

Dai, X., Müller, H.-G., and Yao, F. (2017).
Optimal Bayes classifiers for functional data and density ratios.
*Biometrika*, 104(3):545–560.

Delaigle, A. and Hall, P. (2012).
Achieving near-perfect classification for functional data.
*Journal of the Royal Statistical Society, Series B*, 74:267–286.