Deep Neural Network Classifier for Multi-dimensional Functional Data

3 authors:

Shuoyang Wang
University of Louisville
20 PUBLICATIONS   108 CITATIONS

SEE PROFILE

Guanqun Cao
Auburn University
20 PUBLICATIONS   118 CITATIONS

SEE PROFILE

Zuofeng Shang
New Jersey Institute of Technology
84 PUBLICATIONS   717 CITATIONS

SEE PROFILE

Deep Neural Network Classifier for Multi-dimensional Functional Data

Shuoyang Wang[1], Guanqun Cao[1], and Zuofeng Shang[2]

[1]Department Mathematics and Statistics, Auburn University, U.S.A.
[2]Department of Mathematical Sciences, New Jersey Institute of Technology, U.S.A.

Abstract: We propose a new approach, called as functional deep neural network (FDNN), for classifying multi-dimensional functional data. Specifically, a deep neural network is trained based on the principle components of the training data which shall be used to predict the class label of a future data function. Unlike the popular functional discriminant analysis approaches which rely on Gaussian assumption, the proposed FDNN approach applies to general non-Gaussian multi-dimensional functional data. Moreover, when the log density ratio possesses a locally connected functional modular structure, we show that FDNN achieves minimax optimality. The superiority of our approach is demonstrated through both simulated and real-world datasets.

Key words and phrases: Functional classification; Functional data analysis; Functional neural networks; Minimax excess misclassification risk; Multi-dimensional functional data.

1 Introduction

Due to modern advanced technology, complex functional data are ubiquitous. A fundamental problem in functional data analysis is to classify a data function based on training samples. A typical 1D example is the speech recognition data extracted from the TIMIT database, in which the training samples are digitized speech curves of American English speakers from different phoneme groups, and the task is to predict the phoneme of a new speech curve. Typical 2D and 3D examples include the brain imaging data extracted from Early Mild Cognitive Impairment (EMCI) or Alzeheimer's Disease (AD), in which the training samples are digitized brain images, and the task is to predict the stage of a new patient. Besides above examples, functional data classification has wide applications in various fields such as machine learning, genetics, agriculture, chemimetrics and artificial intelligence [34, 21, 31, 8]. Recent monographs [16, 19] provide comprehensive and general discussions on this field.

Classical multivariate analysis techniques such as logistic regression or discriminant analysis no longer work for functional data due to its intrinsically infinite dimensionality [39]. A mainstream technique in functional data classification is based on functional principle component analysis (FPCA) such as functional discriminant analysis [33, 11, 10, 12, 14, 9, 4, 29, 1, 41]. Functional discriminant analysis requires data function being Gaussian process, under which the decision boundary is characterized by a linear or quadratic polynomial so that classic discriminant analysis approach can accurately recover

1

the decision boundary. Gaussian assumption is restrictive and often violated in practice. When data distributions are general non-Gaussian, the resulting decision boundary is often complicated which cannot be accurately recovered by existing approaches. Our aim is to construct a new functional classifier to overcome this challenge.

In this paper, we propose a new approach, called as functional deep neural network (FDNN), for multi-dimensional functional data classification. We start from FPCA to extract the functional principle components of the data functions, and then train a DNN based on these FPCs as well as their corresponding class labels. As demonstrated through numerical studies, our FDNN approach performs well in classifying complex curve or imaging data. Moreover, our FDNN has desirable theoretical properties. Intuitively, when the network architectures are suitably selected, DNN shall have large expressive power (see [30, 42]) so that functional Bayes classifier can be accurately recovered, even though data distributions are complex. Specifically, we show that, when the log-ratio of the population densities demonstrates a locally connected functional modular structure, our FDNN is proven minimax optimal. The proposed functional modular structure is useful to overcome the infinite dimensionality of functional data, and is meaningful as demonstrated in various examples (see Section 5). Relevant modular structures have been recently adopted by researchers in nonparametric regression and classification to characterize the local behavior of the multivariate input variables, based on which DNN approaches are proven to overcome the "curse of dimensionality." See [32, 3, 25, 24, 40, 22, 17, 18, 5].

The rest of this article is organized as follows. In Section 2 we review functional Bayes classifier in general setting. In Section 3, we propose FDNN classifier. In Section 4, we establish theoretical properties of FDNN under suitable technical assumptions. Section 5 provides three progressive examples to demonstrate the validity of these technical assumptions. In Section 6, performances of FDNN and its competitors are demonstrated through simulation studies. In Section 7, we apply FDNN to speech recognition data and Alzeheimer's Disease data. Section 8 summarizes the conclusions. Technical proofs are provided in Appendix and a supplement document. R programs for implementing our method are provided on GitHub; see Section 6.

2 Functional Bayes classifier under non-Gaussianity

In this section, we review functional Bayes classifier for binary classification. Let $X(s), s \in \mathcal{S} := [0, 1]^d$ be a random process with $\int_{\mathcal{S}} \mathbb{E}X(s)^2 ds < \infty$, and $Y \in \{-1, 1\}$ be a uniform random class label such that, under $Y = k$, $X(s)$ has unknown mean function $\mu_k(s)$ and unknown covariance function $\Omega_k(s, s')$, for $s, s' \in \mathcal{S}$. Suppose that Ω_k satisfies a Karhunen–Loéve decomposition:

$$\Omega_k(s, s') = \sum_{j=1}^{\infty} \lambda_{kj} \psi_{kj}(s) \psi_{kj}(s'), s, s' \in \mathcal{S}, \tag{1}$$

where $\psi_{kj}, j \geq 1$ is an orthonormal basis of $L^2(\mathcal{S})$ with respect to the usual L^2 inner product, and $\lambda_{k1} \geq \lambda_{k2} \geq \cdots > 0$ are nonincreasing positive eigenvalues. Notably, (1) requires the covariance functions being decomposed in terms of the same eigenfunctions, which is a common assumption in functional classification literature; see [10] and [9]. Further relaxation of this assumption is discussed in Section .

Under $Y = k$, write $X(s) = \sum_{j=1}^{\infty} \xi_j \psi_{kj}(s)$, where ξ_j's are pairwise uncorrelated random coefficients. Let $\boldsymbol{\xi} = (\xi_1, \xi_2, \ldots)$ and $h_k(\cdot)$ be the unknown conditional density of $\boldsymbol{\xi}$ under $Y = k$. Define

$Q^*(\cdot)$ as the log density ratio functional between the two classes:

$$Q^*(\boldsymbol{\xi}) = \log\left(\frac{h_1(\boldsymbol{\xi})}{h_{-1}(\boldsymbol{\xi})}\right).$$

The functional Bayes rule for classifying a data function $X \in L^2(\mathcal{S})$ thus has an expression

$$G^*(X) = \begin{cases} 1, & Q^*(\boldsymbol{\xi}) \geq 0, \\ -1, & Q^*(\boldsymbol{\xi}) < 0. \end{cases} \tag{2}$$

Direct estimation of Q^* is infeasible due to the infinite dimensionality of the input. A common practice is to estimate its finite-dimensional truncation. For $J \geq 1$, let $\boldsymbol{\xi}_J = (\xi_1, \ldots, \xi_J)^\top$ be the leading J components of $\boldsymbol{\xi}$ and $h_k^{(J)}(\cdot)$ be the marginal density of $\boldsymbol{\xi}_J$ under $Y = k$, for $k = \pm 1$. Define the truncated log density ratio

$$Q_J^*(\boldsymbol{\xi}_J) = \log\left(\frac{h_1^{(J)}(\boldsymbol{\xi}_J)}{h_{-1}^{(J)}(\boldsymbol{\xi}_J)}\right),$$

which is the log density ratio of $h_1^{(J)}$ to $h_{-1}^{(J)}$. The intuition is that, when J is large, $h_k^{(J)}$ approaches h_k so that Q_J^* is an accurate approximation of Q^*. Our aim is to design an efficient method to estimate Q_J^*, which will in turn estimate Q^*.

3 Functional deep neural network classifier

Suppose we observe n i.i.d. training samples $\{(X_i(s), Y_i) : 1 \leq i \leq n, s \in \mathcal{S}\}$, which are independent of $X(s), s \in \mathcal{S}$ to be classified. For $k = \pm 1$, define sample covariance function

$$\widehat{\Omega}_k(s, s') = \frac{1}{n_k} \sum_{i \in I_k} (X_i(s) - \bar{X}_k(s))(X_i(s') - \bar{X}_k(s')), \quad s, s' \in \mathcal{S},$$

where I_k is the collection of i such that $Y_i = k$, $n_k := |I_k|$ and $\bar{X}_k(s) = \frac{1}{n_k} \sum_{i \in I_k} X_i(s)$ is the sample mean function of class k. Perform Karhunen–Loéve decomposition for $\widehat{\Omega}_k$:

$$\widehat{\Omega}_k(s, s') = \sum_{j=1}^{\infty} \widehat{\lambda}_{kj} \widehat{\psi}_{kj}(s) \widehat{\psi}_{kj}(s'), s, s' \in \mathcal{S},$$

and write the sample data function X_i, under $Y_i = k$, as

$$X_i(s) = \sum_{j=1}^{\infty} \widehat{\xi}_{ij} \widehat{\psi}_{kj}(s), i = 1, \ldots, n.$$

Intuitively, $\widehat{\boldsymbol{\xi}}^{(i)} := (\widehat{\xi}_{i1}, \widehat{\xi}_{i2}, \ldots)$ is an estimator of $\boldsymbol{\xi}^{(i)} := (\xi_{i1}, \xi_{i2}, \ldots)$, in which ξ_{ij} are unobservable random coefficients of X_i with respect to the population basis ψ_{kj}. Hence, it is natural to design classifiers based on $\widehat{\boldsymbol{\xi}}^{(i)}$'s.

Let $\widehat{\boldsymbol{\xi}}_J^{(i)} = (\widehat{\xi}_{i1}, \ldots, \widehat{\xi}_{iJ})^\top$ be the J-dimensional truncation of $\widehat{\boldsymbol{\xi}}^{(i)}$ for $i = 1, \ldots, n$. When X_i's are Gaussian processes, various classifiers have been proposed such as centroid classifier [10], QDA [12] and nonparametric Bayes classifier [9]. When X_i's are non-Gaussian, one major challenge is the

underlying complicated form of the conditional densities h_1 and h_{-1} so that estimation of Q_J^* is typically difficult. Inspired by the rich approximation power of DNN, in this section, we propose a new classifier called FDNN (functional+DNN) that can accurately estimate Bayes classifiers even when h_1 and h_{-1} are non-Gaussian complicated.

We will train a DNN to estimate Q_J^* based on $\widehat{\boldsymbol{\xi}}_J^{(i)}$'s. In what follows, we will describe our method in details. Let σ denote the rectifier linear unit (ReLU) activation function, i.e., $\sigma(x) = (x)_+$ for $x \in \mathbb{R}$. For any real vectors $\boldsymbol{V} = (v_1, \ldots, v_w)^\top$ and $\boldsymbol{y} = (y_1, \ldots, y_w)^\top$, define the shift activation function $\sigma_{\boldsymbol{V}}(\boldsymbol{y}) = (\sigma(y_1 - v_1), \ldots, \sigma(y_w - v_w))^\top$. For $L \geq 1$, $\boldsymbol{p} = (p_1, \ldots, p_L) \in \mathbb{N}^L$, let $\mathcal{F}(L, J, \boldsymbol{p})$ denote the class of fully connected feedforward DNN with J inputs, L hidden layers and, for $l = 1, \ldots, L$, p_l nodes on the lth hidden layer. Equivalently, any $f \in \mathcal{F}(L, J, \boldsymbol{p})$ has an expression

$$f(\boldsymbol{x}) = \mathbf{W}_L \sigma_{\boldsymbol{V}_L} \mathbf{W}_{L-1} \sigma_{\boldsymbol{V}_{L-1}} \ldots \mathbf{W}_1 \sigma_{\boldsymbol{V}_1} \mathbf{W}_0 \boldsymbol{x}, \quad \boldsymbol{x} \in \mathbb{R}^J, \tag{3}$$

where $\mathbf{W}_l \in \mathbb{R}^{p_{l+1} \times p_l}$, for $l = 0, \ldots, L$, are weight matrices, $\boldsymbol{V}_l \in \mathbb{R}^{p_l}$, for $l = 1, \ldots, L$, are shift vectors. Here we adopt the convention that $p_0 = J$ and $p_{L+1} = 1$.

We consider the following class of DNN:

$$\mathcal{F}(L, J, \boldsymbol{p}, B) = \left\{ f \in \mathcal{F}(L, J, \boldsymbol{p}) : \max_{0 \leq l \leq L} \|\mathbf{W}_l\|_\infty \leq B, \max_{1 \leq l \leq L} \|\mathbf{v}_l\|_\infty \leq B \right\},$$

where $\| \cdot \|_\infty$ denotes the maximum-entry norm of a matrix/vector or supnorm of a function, and $B > 0$ controls the largest weights and shifts.

Given the training data $(\boldsymbol{\xi}_J^{(1)}, Y_1), \ldots, (\boldsymbol{\xi}_J^{(n)}, Y_n)$, let

$$\widehat{f}_\phi(\cdot) = \arg \min_{f \in \mathcal{F}(L, J, \boldsymbol{p}, B)} \frac{1}{n} \sum_{i=1}^n \phi(f(\widehat{\boldsymbol{\xi}}_J^{(i)}) Y_i), \tag{4}$$

where $\phi(x) = \max(1 - x, 0)$ denotes the hinge loss. We then propose the following FDNN classifier: for $X \in L^2(\mathcal{S})$,

$$\widehat{G}^{FDNN}(X) = \begin{cases} 1, & \widehat{f}_\phi(\boldsymbol{\xi}_J) \geq 0, \\ -1, & \widehat{f}_\phi(\boldsymbol{\xi}_J) < 0. \end{cases} \tag{5}$$

In practice, we suggest the following data-splitting method for selecting $(L, J, \boldsymbol{p}, B)$:

- Step 1. Randomly divide the whole sample $(\widehat{\boldsymbol{\xi}}_J^{(i)}, Y_i)$'s into two subsets indexed by $\mathcal{I}_1$ and $\mathcal{I}_2$, respectively, with about $|\mathcal{I}_1| = 0.8n$ and $|\mathcal{I}_2| = 0.2n$.

- Step 2. For each $(L, J, \boldsymbol{p}, B)$, we train a DNN $\widehat{f}_{L, J, \boldsymbol{p}, B}$ using (4) based on subset $\mathcal{I}_1$, and then calculate the testing error based on subset $\mathcal{I}_2$ as

$$\text{err}(L, J, \boldsymbol{p}, B) = \frac{1}{|\mathcal{I}_2|} \sum_{i \in \mathcal{I}_2} I(\widehat{f}_{L, J, \boldsymbol{p}, B}(\widehat{\boldsymbol{\xi}}_J^{(i)}) Y_i < 0). \tag{6}$$

- Step 3. Choose $(L, J, \boldsymbol{p}, B)$, possibly from a preselected set, to minimize $\text{err}(L, J, \boldsymbol{p}, B)$.

4   Minimax optimality of FDNN

For a generic functional classifier $\widehat{G}$, its excess misclassification risk is defined as $\mathcal{E}_h(\widehat{G}) := \mathbb{E}[R_h(\widehat{G}) - R_h(G^*)]$, where $R_h(\widehat{G}) := \mathbb{E}_h[\mathbb{I}\{\widehat{G}(X) \neq Y\}]$ is the misclassification risk of $\widehat{G}$ taken with respect to (X, Y) under $h := \{h_1, h_{-1}\}$, with Y the true class label of X. A central task is to design $\widehat{G}$ that achieves minimax excess misclassification risk (MEMR), i.e.,

$$\max_{h \in \mathcal{H}} \mathcal{E}_h(\widehat{G}) \asymp \inf_{\widehat{G}} \max_{h \in \mathcal{H}} \mathcal{E}_h(\widehat{G}), \tag{7}$$

where $\mathcal{H}$ is a proper class of h to be described later and the infimum is taken over all classifiers based on training samples. Classifiers satisfying (7) are called as minimax optimal.

There is a rich literature on construction of minimax optimal classifiers when data dimension is fixed or diverging. For instance, classic nonparametric approaches, such as ones directly estimating Bayes classifier nonparametrically, are proven minimax optimal in fixed-dimension regime [26, 37, 38, 20, 14, 13, 27, 17]. When data are high-dimensional Gaussian, discriminant analysis approaches are proven minimax optimal [7, 6]. On the other hand, under functional Gaussian data, researchers have proposed various functional classifiers, including functional quadratic discriminant analysis (FQDA) [33, 11, 10, 12, 14, 9, 4, 29, 7, 6]. Gaussianity leads to a linear or quadratic polynomial Q^* which can be effectively estimated by FQDA, based on which [41] showed that FQDA is minimax optimal. It is still unclear how to design optimal functional classifiers when data are non-Gaussian, a gap that the present article attempts to close.

In this section, we will establish minimax optimality of FDNN classifier under non-Gaussian functional data. For technical convenience, assume that the two populations have common known basis, i.e., $\psi_{+1j}(\cdot) = \psi_{-1j}(\cdot)$. Therefore, we can train FDNN classifier based on $\boldsymbol{\xi}_J^{(i)} := (\xi_{i1}, \ldots, \xi_{iJ})^\top$, for $i = 1, \ldots, n$. We will first derive an upper bound for the excess misclassification risk of our FDNN classifier, and then derive a lower bound for the MEMR which matches the above upper bound. Therefore, our FDNN is able to achieve sharp rate of MEMR. Extensions to general basis are possible with more tedious technical arguments.

Before proceeding further, we introduce some technical assumptions. At high levels, our assumptions are different from those proposed under Gaussian case. For instance, in either high- or infinite-dimensional Gaussian data, it is well known that density ratio between two Gaussian population densities has an explicit expression in terms of mean difference and variance ratio, which impacts the sharp rate of MEMR. More precisely, in high-dimensional Gaussian data classification, the rate depends on the number of nonzero components of mean difference vector [7, 6]; in Gaussian functional data classification, the rate depends on the decay orders of both mean difference series and variance ratio series [41]. Nonetheless, in general non-Gaussian case, likelihood ratio does not have an explicit expression, therefore, one cannot simply use mean or variance discrepancy to characterize the sharp rate of MEMR.

In traditional non-Gaussian multivariate data classification, a common strategy is to assume smooth density ratio and controllable noise, under which minimax optimal classifiers were proposed (see [26, 37, 2, 18] and references therein). In the functional data framework, the input variable of Q^* is infinite-dimensional, hence, the above strategy no longer works. We instead propose a set of functional conditions on Q^* under which minimax optimality shall be established. Such conditions are viewed as infinite-dimensional extensions of [2] and [32].

For $t \geq 1$, a measurable subset $D \subset \mathbb{R}^t$ and constants $\beta, K > 0$, define

$$
\begin{aligned}
&\mathcal{C}^\beta(D, K) \\
&= \left\{ f : D \mapsto \mathbb{R} \mid \sum_{\boldsymbol{\alpha}: |\boldsymbol{\alpha}| < \beta} \|\partial^{\boldsymbol{\alpha}} f\|_\infty + \sum_{\boldsymbol{\alpha}: |\boldsymbol{\alpha}| = \lfloor \beta \rfloor} \sup_{\boldsymbol{x}, \boldsymbol{x}' \in D, \boldsymbol{x} \neq \boldsymbol{x}'} \frac{|\partial^{\boldsymbol{\alpha}} f(\boldsymbol{x}) - \partial^{\boldsymbol{\alpha}} f(\boldsymbol{x}')|}{\|\boldsymbol{x} - \boldsymbol{x}'\|_\infty^{\beta - \lfloor \beta \rfloor}} \leq K \right\},
\end{aligned}
$$

where $\partial^{\boldsymbol{\alpha}} = \partial^{\alpha_1} \ldots \partial^{\alpha_t}$ denotes the partial differential operator with multi-index $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_t) \in \mathbb{N}^t$, $|\boldsymbol{\alpha}| = \alpha_1 + \cdots + \alpha_t$. Equivalently, $\mathcal{C}^\beta(D, K)$ is the ball of β-Hölder smooth functions on D with radius K. A function $f : \mathbb{R}^t \to \mathbb{R}$ is said to be locally β-Hölder smooth if for any $a, b \in \mathbb{R}$, there exists a constant K (possibly depending on a, b) such that $f \in \mathcal{C}^\beta([a, b]^t, K)$.

For $q \geq 0, J \geq 1$, let $d_0 = J$ and $d_{q+1} = 1$. For $\boldsymbol{d} = (d_1, \ldots, d_q) \in \mathbb{N}_+^q$, $\boldsymbol{t} = (t_0, \ldots, t_q) \in \mathbb{N}_+^{q+1}$ with $t_u \leq d_u$ for $u = 0, \ldots, q$, $\boldsymbol{\beta} := (\beta_0, \ldots, \beta_q) \in \mathbb{R}_+^{q+1}$, let $\mathcal{G}(q, J, \boldsymbol{d}, \boldsymbol{t}, \boldsymbol{\beta})$ be the class of functions g satisfying a modular expression

$$
g(\boldsymbol{x}) = g_q \circ \cdots \circ g_0(\boldsymbol{x}), \quad \forall \boldsymbol{x} \in \mathbb{R}^{d_0}, \tag{8}
$$

where $g_u = (g_{u1}, \ldots, g_{ud_{u+1}}) : \mathbb{R}^{d_u} \mapsto \mathbb{R}^{d_{u+1}}$ and $g_{uv} : \mathbb{R}^{t_u} \mapsto \mathbb{R}$ are locally β_u-Hölder smooth. The d_u arguments of g_u are locally connected in the sense that each component g_{uv} only relies on $t_u (\leq d_u)$ arguments. Similar structures have been considered by [32, 3, 25, 24, 40, 22, 17, 18, 17] in multivariate regression or classification to overcome high-dimensionality. Generalized additive model [15] and tensor product space ANOVA model [23] are special cases; see [25].

Let $\mathcal{H}^* \equiv \mathcal{H}^*(q, \boldsymbol{d}, \boldsymbol{t}, \boldsymbol{\beta})$ be the class of population densities $h = \{h_1, h_{-1}\}$ of $\boldsymbol{\xi}$ such that, for any $J \geq 1$, $Q_J^* \in \mathcal{G}(q, J, \boldsymbol{d}, \boldsymbol{t}, \boldsymbol{\beta})$. Equivalently, for any $h \in \mathcal{H}^*$ and $J \geq 1$, the corresponding truncated log density ratio Q_J^* has a modular structure (8) with certain smoothness. Although Q_J^* has J arguments, it involves at most $t_0 d_1$ effective arguments, implying that the two population densities differ by a small number of variables. Relevant conditions are necessary for high-dimensional classification. For instance, in high-dimensional Gaussian data classification, [7, 6] show that, to consistently estimate Bayes classifier, it is necessary that the mean vectors differ at a small number of components. The modular structure holds for arbitrary J, which may be viewed as a functional extension of [32]. Note that the density class $\mathcal{H}^*$ covers many popular models studied in literature, either Gaussian or non-Gaussian; see Section 5. Moreover, we introduce the following regularity conditions on Q^*.

Assumption 1. *(Functional Tsybakov noise condition) There exist constants $C > 0$ and $\alpha \geq 0$ such that*

$$
\mathbb{P}\left(\left| \frac{1 - \exp\{-Q^*(\boldsymbol{\xi})\}}{1 + \exp\{-Q^*(\boldsymbol{\xi})\}} \right| \leq x \right) \leq C x^\alpha, \quad \forall x > 0. \tag{9}
$$

Assumption 2. *(Approximation error of Q_J^*) There exist a constant $J_0 \geq 1$ and decreasing functions $\epsilon(\cdot) : [1, \infty) \to \mathbb{R}_+$ and $\Gamma(\cdot) : [0, \infty) \to \mathbb{R}_+$, with $\sup_{J \geq 1} J^\varrho \epsilon(J) < \infty$ for some $\varrho > 0$ and $\int_0^\infty \Gamma(x) dx < \infty$, such that for any $J \geq J_0$ and $x > 0$,*

$$
\mathbb{P}\left(|Q^*(\boldsymbol{\xi}) - Q_J^*(\boldsymbol{\xi}_J)| \geq x \right) \leq \epsilon(J) \Gamma(x). \tag{10}
$$

Assumption 1 characterizes the discrepancy between Q^* and random guess. Specifically, it requires that the probability of Q^* close to 0 by x is upper bounded by an order x^α. Assumption 1 is a functional extension of the classic *Tsybakov noise condition*, which is necessary in establishing minimax classification in multivariate case [26, 37]). Assumption 2 provides an upper bound on the probability of Q^*

differing from Q_J^* by at least x, which approaches zero if either J or x tends to infinity, implying that Q_J^* is an accurate approximation of Q^*. Both assumptions can be verified in several concrete examples included in Section 5.

Our MEMR results will be based on the following class of population densities of $\boldsymbol{\xi}$:

$$\mathcal{H} \quad \equiv \quad \mathcal{H}\left(q, \boldsymbol{d}, \boldsymbol{t}, \boldsymbol{\beta}, \alpha, C, \epsilon(\cdot), \Gamma(\cdot)\right) = \{h \in \mathcal{H}^* : Q^* \text{ satisfies both Assumptions 1 and 2}\}.$$

Finally, we introduce an assumption on the orders of $(L, J, \boldsymbol{p}, B)$, under which the exact rate of MEMR shall be established. Let

$$S_0 = \min_{0 \le u \le q} \frac{\beta_u^*(\alpha + 1)}{\beta_u^*(\alpha + 2) + t_u}, S_1 = \max_{0 \le u \le q} \frac{t_u}{\beta_u^*(\alpha + 2) + t_u}, S_2 = \min_{0 \le u \le q} \frac{1}{\beta_u^*(\alpha + 2) + t_u},$$

where $\beta_u^* := \beta_u \prod_{k=u+1}^q (\beta_k \wedge 1)$.

Assumption 3. *The DNN class $\mathcal{F}(L, J, \boldsymbol{p}, B)$ satisfies*

(a) $L \asymp \log n$;

(b) $\left(n \log^{-3} n\right)^{S_0/\rho} \lesssim J \lesssim (n \log^{-3} n)^{S_1}$;

(c) $\max_{1 \le \ell \le L} p_\ell \asymp (n \log^{-3} n)^{S_1}$;

(d) $B \asymp (n \log^{-3} n)^{S_2}$.

Assumption 3(a), (c), and (d) provide exact orders on $L, \boldsymbol{p}, B$, respectively. Assumption 3(b) provides a range on J. Notably, this condition implies $\varrho \ge S_0/S_1$, i.e., the function $\epsilon(J)$ rapidly converges to zero when $J \to \infty$.

Theorem 4.1. *There exist positive constants C_1, C_2, depending on $q, \boldsymbol{d}, \boldsymbol{t}, \boldsymbol{\beta}, \alpha, C, \epsilon(\cdot)$, and $\Gamma(\cdot)$, such that the following results hold:*

(i) $\inf_{\widehat{G}} \sup_{h \in \mathcal{H}} \mathcal{E}_h(\widehat{G}) \ge C_1 n^{-S_0}$, *where the infimum is taken over all classifiers $\widehat{G}$ based on training samples;*

(ii) under Assumption 3, it holds that

$$\sup_{h \in \mathcal{H}} \mathcal{E}_h(\widehat{G}^{FDNN}) \le C_2 \left(\frac{\log^3 n}{n}\right)^{S_0}.$$

Theorem 4.1 establishes a nonasymptotic rate for the MEMR which is of order n^{-S_0}. Moreover, the proposed FDNN classifier is able to achieve this rate up to a logarithmic factor, and hence, is minimax optimal. Since S_0 involves the intrinsic dimensions t_u's rather than the original dimensions d_u's, the rate of MEMR is typically fast, demonstrating the theoretical advantage of our FDNN classifier.

5   Examples

The minimax results in Section 4 are based on parameter space $\mathcal{H}$. In this section, we provide some concrete examples to demonstrate the validity of such space.

5.1 Gaussian functional data with independent coefficients

Suppose that, under $Y = k$, the random coefficients ξ_j are independent Gaussian with mean μ_{kj} and variance λ_{kj}. Define $M = \{j : \mu_{1j} \neq \mu_{-1j}\}$ and $N = \{j : \lambda_{1j} \neq \lambda_{-1j}\}$. Assume that M, N are mutually disjoint with common cardinality ω. It can be shown that, for any $J \geq J_0 := \max M \cup N$, $Q_J^*(\boldsymbol{\xi}_J) = g_1(g_0(\boldsymbol{\xi}_J))$, where g_0 has components $g_{0j}(\xi_j) = a_j \xi_j^2 + b_j \xi_j + c_j$ for some constants a_j, b_j, c_j depending on $\mu_{1j}, \mu_{-1j}, \lambda_{1j}, \lambda_{-1j}$, and $g_1(g_0(\boldsymbol{\xi}_J)) = \sum_{j \in M \cup N} g_{0j}(\xi_j)$. Clearly, $d_0 = J$ and $t_0 = 1$, and $M \cup N$ has cardinality 2ω, $d_1 = t_1 = 2\omega$. So $Q_J^* \in \mathcal{G}(1, J, 2\omega, (1, 2\omega), \beta)$ for any $\beta > 0$. Meanwhile, since $Q_J^* = 0$ for all $J \geq J_0$, and for any function $\epsilon(\cdot)$ with exponential tails and any density $\Gamma(\cdot)$, Assumption 1 holds for $\alpha = 1$ and Assumption 2 holds for J_0.

5.2 Student's t functional data with independent coefficients

Suppose that, under $Y = k$, ξ_j are independent Student's t variables $t_{\nu_{kj}}$, where $\nu_{kj} \geq 1$ are degrees of freedom of the t variables. Define $M = \{j : \mu_{1j} \neq \mu_{-1j}\}$ whose cardinality is ω. It can be shown that, for any $J \geq J_0 := \max M \cup N$, $Q_J^*(\boldsymbol{\xi}_J) = g_1(g_0(\boldsymbol{\xi}_J))$, where g_0 has components

$$g_{0j}(\xi_j) = \log e_j - \frac{\nu_{1j} + 1}{2} \log\left(1 + \frac{\xi_j^2}{\nu_{1j}}\right) + \frac{\nu_{-1j} + 1}{2} \log\left(1 + \frac{\xi_j^2}{\nu_{-1j}}\right),$$

for some constant e_j depending on ν_{kj}, and $g_1(g_0(\boldsymbol{\xi}_J)) = \sum_{j \in M \cup N} g_{0j}(\xi_j)$. Similar to Section 5.1, we have $Q_J^* \in \mathcal{G}(1, J, 2\omega, (1, 2\omega), \beta)$ for any $\beta > 0$. Assumptions 1 and 2 can be similarly verified as well.

5.3 Student's t functional data with dependent coefficients

We consider an extension of Section 5.2 which involves dependent coefficients. Let $p \geq 1$ and $\nu \geq 2$ be integers. Suppose that, under $Y = k$, $\boldsymbol{\zeta}_j := (\xi_j, \xi_{j+1}, \ldots, \xi_{j+p-1})^\top$, $j = 1, p+1, 2p+1, \ldots$ are independent multivariate Student's t vectors following $t_\nu(\boldsymbol{\mu}_{kj}, \boldsymbol{\Sigma}_{kj})$, where $\boldsymbol{\mu}_{kj} \in \mathbb{R}^p$ and positive definite $\boldsymbol{\Sigma}_{kj}$ is $p \times p$ positive definite. Define $M = \{j : \boldsymbol{\mu}_{1j} \neq \boldsymbol{\mu}_{-1j}\}$ and $N = \{j : \boldsymbol{\Sigma}_{1j} \neq \boldsymbol{\Sigma}_{-1j}\}$. Assume sets M and N are mutually disjoint with common cardinality ω. For any $J \geq J_0 := J_0 := \max M \cup N + p - 1$, then it can be shown that

$$Q_J^*(\boldsymbol{\xi}_J)$$
$$= \sum_{j \in M \cup N} \left\{ \frac{1}{2} \log\left(\frac{|\boldsymbol{\Sigma}_{-1j}|}{|\boldsymbol{\Sigma}_{1j}|}\right)^{1/2} + \frac{\nu + p}{2} \log\left(\frac{1 + \nu^{-1}(\boldsymbol{\zeta}_j - \boldsymbol{\mu}_{-1j})^\top \boldsymbol{\Sigma}_{-1j}^{-1}(\boldsymbol{\zeta}_j - \boldsymbol{\mu}_{-1j})}{1 + \nu^{-1}(\boldsymbol{\zeta}_j - \boldsymbol{\mu}_{1j})^\top \boldsymbol{\Sigma}_{1j}^{-1}(\boldsymbol{\zeta}_j - \boldsymbol{\mu}_{1j})}\right) \right\}.$$

Note that there are 2ω terms in the above sum. Similar to Sectoions 5.1, we have $Q_J^* \in \mathcal{G}(1, J, 2\omega, (1, 2\omega), \beta)$ for any $\beta > 0$. Assumptions 1 and 2 can be similarly verified as well.

6 Simulation study

In this section, we examine the performances of FDNN and two competitors, quadratic discriminant method (QD) proposed in [12] and the nonparametric Bayes classifier (NB) proposed in [9], through simulation studies. Our studies involve both $d = 1$ and $d = 2$, corresponding to 1D and 2D functional data, respectively. All experiments are conducted in R. We summarize R codes and examples for the proposed FDNN algorithms on GitHub (https://github.com/FDASTATAUBURN/fdnn-classification). For 1D functional data, we considered two data generation processes (DGP).

n	DGP1			DGP2		
	FDNN	QD	NB	FDNN	QD	NB
40	31.76(0.10)	38.58(0.02)	38.33(0.02)	16.69(0.04)	39.99(0.01)	39.26(0.03)
100	18.82(0.10)	37.91(0.02)	41.03(0.02)	13.20(0.01)	38.42(0.09)	40.27(0.03)
200	13.19(0.10)	37.35(0.02)	39.92(0.02)	12.29(0.01)	42.63(0.02)	39.84(0.04)
400	9.62(0.04)	36.75(0.02)	38.54(0.02)	12.40(0.01)	43.98(0.09)	38.51(0.04)

Table 1: Misclassification rates (%) with standard errors in brackets for DGP1 and DGP2

	n			
	40	100	200	400
DGP 3	0.170(0.066)	0.148(0.055)	0.139(0.054)	0.127(0.040)
DGP 4	0.139(0.055)	0.127(0.014)	0.127(0.040)	0.123(0.011)

Table 2: Misclassification rates (%) with standard errors in brackets for DGP3 and DGP4 with the FDNN classifier

- *DGP1:* Generate $X(s) = \sum_{j=1}^{3} \xi_j \psi_j(s)$, $s \in [0,1]$, where $\psi_1(s) = \log(s+2)$, $\psi_2(s) = s$ and $\psi_3(s) = s^3$. Under class k, generate independently $(\xi_1, \xi_2, \xi_3)^\top \sim N(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$, where $\boldsymbol{\mu}_1 = (-1, 2, -3)^\top$, $\boldsymbol{\Sigma}_1 = \operatorname{diag}\left(\frac{3}{5}, \frac{2}{5}, \frac{1}{5}\right)$, $\boldsymbol{\mu}_{-1} = \left(-\frac{1}{2}, \frac{5}{2}, -\frac{5}{2}\right)^\top$, $\boldsymbol{\Sigma}_{-1} = \operatorname{diag}\left(\frac{9}{10}, \frac{1}{2}, \frac{3}{10}\right)$.

- *DGP2:* Generate $X(s) = \sum_{j=1}^{3} \xi_j \psi_j(s)$, $s \in [0,1]$, where $\psi_j(s)$'s are the same as in DGP1. Under class 1, generate independently $(\xi_1, \xi_2, \xi_3)^\top \sim N(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$, where $\boldsymbol{\mu}_1 = (-1, 2, -3)^\top$, $\boldsymbol{\Sigma}_1 = \operatorname{diag}(3, 2, 1)$; under class -1, generate independently $\xi_j \sim t_{7-2j}$, $j = 1, 2, 3$.

For 2D functional data, we considered two DGPs:

- *DGP3:* Generate $X(s_1, s_2) = \sum_{j=1}^{4} \xi_j \psi_j(s_1, s_2)$, $0 \leq s_1, s_2 \leq 1$, where $\psi_1(s_1, s_2) = s_1 s_2$, $\psi_2(s_1, s_2) = s_1 s_2^2$, $\psi_3(s_1, s_2) = s_1^2 s_2$, $\psi_4(s_1, s_2) = s_1^2 s_2^2$. Under class k, generate independently $(\xi_1, \xi_2, \xi_3, \xi_4)^\top \sim N(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$, where $\boldsymbol{\mu}_1 = (8, -6, 4, -2)^\top$, $\boldsymbol{\Sigma}_1 = \operatorname{diag}(8, 6, 4, 2)$, $\boldsymbol{\mu}_{-1} = \left(-\frac{7}{2}, -\frac{5}{2}, \frac{3}{2}, -\frac{1}{2}\right)^\top$, $\boldsymbol{\Sigma}_{-1} = \operatorname{diag}\left(\frac{9}{2}, \frac{7}{2}, \frac{5}{2}, \frac{3}{2}\right)$.

- *DGP4:* Generate $X(s_1, s_2) = \sum_{j=1}^{4} \xi_j \psi_j(s_1, s_2)$, $0 \leq s_1, s_2 \leq 1$, where $\psi_j(s_1, s_2)$'s are the same as in DGP3. For $j = 1, 2, 3, 4$, under class 1, generate independently $\xi_j \sim t_{2j}(0)$; under class -1, generate independently $\xi_j \sim t_{2j+1}(\mu_j)$, with non-central parameter $\mu_1 = 2$, $\mu_2 = \frac{3}{2}$, $\mu_3 = 1$, and $\mu_4 = \frac{1}{2}$.

In each DGP, we generated n training data functions and 500 testing data functions, with $n = 40, 100, 200, 400$. Each data function was sampled over 50 grid points in the respective domain. Misclassification rates were evaluated based on 100 replicated datasets. Network parameters were selected based on training data using Steps 1-3 in Section 3. Table 1 and Table 2 summarizes the misclassification rates with standard deviations for 1D and 2D functional data, respectively. Specifically, the proposed FDNN outperforms QDA and NB in all settings. Though FDNN has larger standard deviation than QD and NB, the values decrease when n becomes large. As no existing alternative methods for 2D functional data classification available, we only present the performance of the proposed FDNN classifier. We observe that the misclassification risk decreases as the sample size increase for both Gaussian and non-Gaussian functional data.
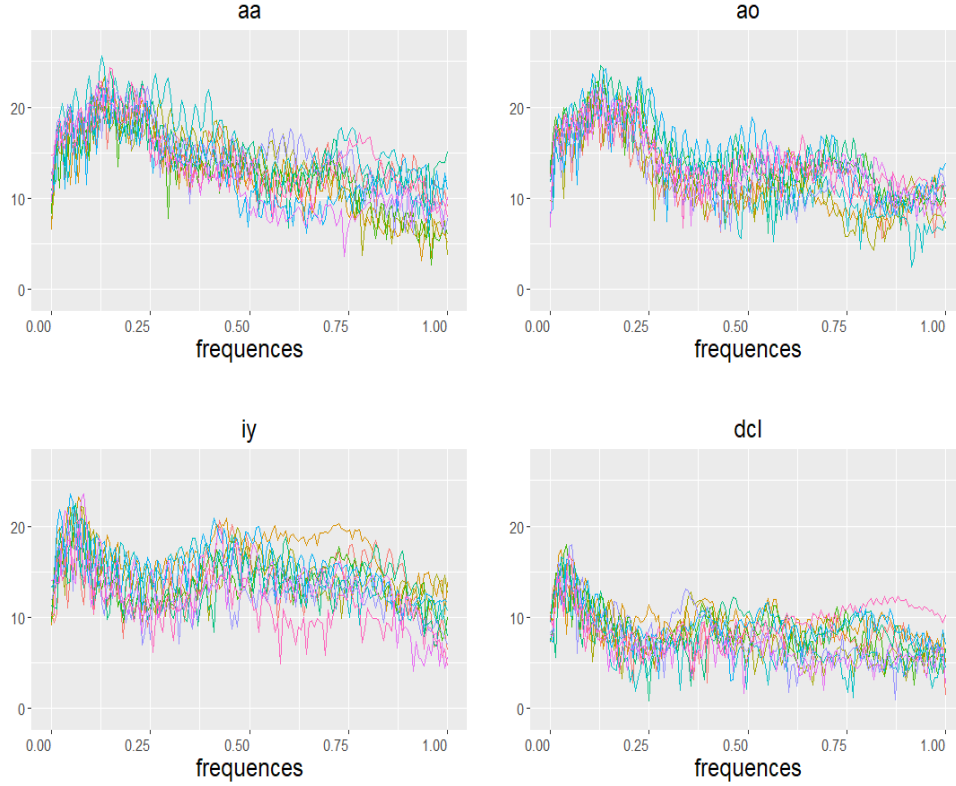
Figure 1: A sample of 10 log-periodograms per class

7 Real data illustrations

7.1 TIMIT database

This benchmark data example was extracted from the TIMIT database (`https://catalog.ldc.upenn.edu/LDC93s1`), which is a widely used resource for research in speech recognition and functional data classification. The data set we used was constructed by selecting four phonemes for classification based on digitized speech from this database. From each speech frame, a log-periodogram transformation is applied so as to cast the speech data in a form suitable for speech recognition. The five phonemes in this data set are transcribed as follows: "sh" as in "she", "dcl" as in "dark", "iy" as the vowel in "she", "aa" as the vowel in "dark", and "ao" as the first vowel in "water". For illustration purpose, we focus on the "aa", "ao", "iy" and "dcl" phoneme classes. Each speech frame is represented by $n = 400$ samples at a 16-kHz sampling rate; the first $M = 150$ frequencies from each subject are retained. Figure 1 displays 10 log-periodograms for each class phoneme.

We randomly select training sample size $n_1 = n_2 = 300$ to train the three classifiers and the rest of 100 samples remained as the test samples. Network parameters were selected based on training data using Steps 1-3 in Section 3. Table 3 reports the mean percentage (averaged over the 100 repetitions) of misclassified test curves. Overall, FDNN outperformed QD and NB in all the four classification tasks. As depicted in the Figure 1, "aa" and "ao" phoneme class trajectories looks extremely similar to each other, the misclassification rate is fairly larger than other classification results. The proposed classifiers

Table 3: Misclassification rates (%) with standard errors in brackets for Speech Recognition data.

Classes	FDNN	QD	NB
"aa" vs "ao"	20.744(0.016)	25.402(0.026)	25.378(0.021)
"aa" vs "iy"	0.193(0.002)	0.288(0.005)	0.273(0.006)
"ao" vs "iy"	0.183(0.004)	0.578(0.005)	0.232(0.005)
"ao" vs "dcl"	0.229(0.002)	0.391(0.005)	0.472(0.006)

FDNN still provides smallest risks and smaller standard errors compared with QD and NB classifiers. For "ao" vs"iy", the misclassification rate of FDNN is less than one third of that of QD; For "ao" vs"dcl", the misclassification rate of FDNN is nearly half of that of NB.

7.2 ADNI database

The dataset used in the preparation of this article were obtained from the ADNI database (`adni.loni.usc.edu`). The ADNI is a longitudinal multicenter study designed to develop clinical, imaging, genetic, and biochemical biomarkers for the early detection and tracking of AD. From this database, we collect PET data from 79 patients in AD group, and 45 patients in EMCI group. This PET dataset has been spatially normalized and post-processed. These AD patients have three to six times doctor visits and we select the PET scans obtained in the third visits. People in EMCI group only have the second visit, and we select the PET scans obtained in the second visits. For AD group, patients' age ranges from 59 to 88 and average age is 76.49, and there are 33 females and 46 males among these 79 subjects. For EMCI group, patients' age ranges from 57 to 89 and average age is 72.33, and there are 26 females and 19 males among these 45 subjects. All scans were reoriented into $79 \times 95 \times 68$ voxels, which means each patient has 68 sliced 2D images with 79×95 pixels. For 2D case, it means each subject has $N = 79 \times 95 = 7,505$ observed pixels for each selected image slice. For 3D case, the observed number of voxels for each patient's brain sample is $N = 79 \times 95 \times 68$.

It is well known that Alzheimer's disease destroys neurons and their connections in hippocampus, the entorhinal cortex, and the cerebral cortex. These parts are corresponding to the first 25 slices. Therefore, for our 2D case study, we specifically select the 5-th, 10-th, 15-th, 20-th and 25-th slices from 68 slices for each patient. We aim to conduct classification based on the information of those slices respectively; see [28]. Figure 2 shows the averaged 2D images for two groups at each slice. For 3D case, we focus on the total 25 slices, so the 3D data is observed on $79 \times 95 \times 25$ points. Figure 3 demonstrates the misclassification rates for both 2D and 3D brain imaging data. There are several interesting finds. First, given a single slice 2D imaging data, the misclassification rates tend to be larger than using total 25 slices data (3D data). It indicates that 3D data contains more helpful information to decrease the misclassification risk. Second, the 20-th slice provides the lowest one among all 2D data. It is a promising finding for neurologists, as this smallest risk indicates this particular slice presents useful information to distinguish the EMCI and AD groups. Further medical checkups are meaningful for this special location in the brain.

8 Conclusion

We propose a new FDNN classifier for classifying non-Gaussian complex function data. Our contributions are twofold. First, we establish sharp convergence rates for MEMR when data are of functional type, and the result can be applied to a large scope of functional data with complex density functions.
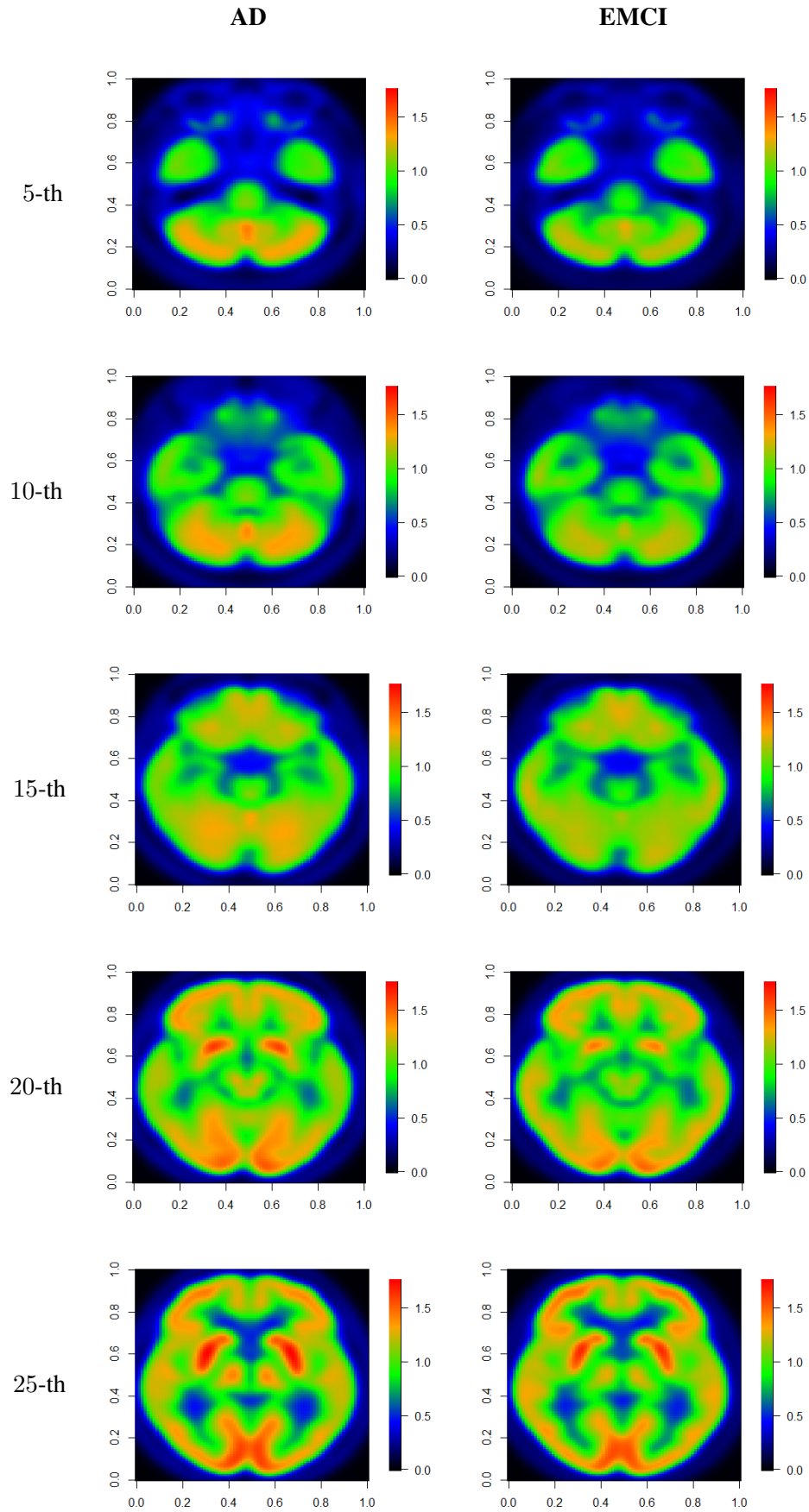
Figure 2: Averaged images of the 5-th, the 10-th, the 15-th, the 20-th and the 25-th slices of EMCI (left column) group and AD group (right column).
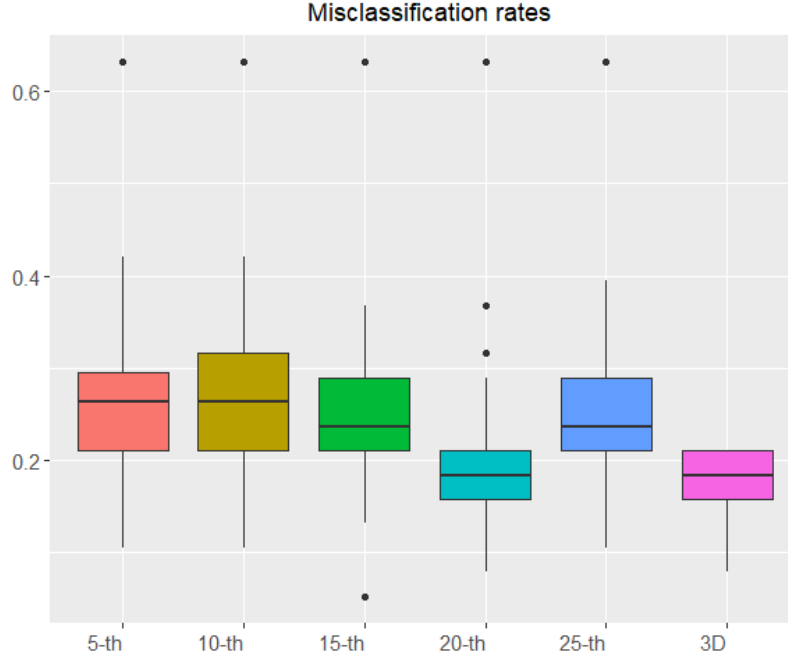
Figure 3: Grouped boxplot of misclassification rates for the 5-th, the 10-th, the 15-th, the 20-th , the 25-th slices and 3D data of the first 25 slices between EMCI and AD groups.

The proposed FDNN is able to attain the sharp rate. Second, our FDNN classifier is able to handle various 1D or multi-dimensional complex functional data. As demonstrated through extensive simulated and real-data examples, the proposed FDNN classifier has outstanding performances in both Gaussian and non-Gaussian settings.

Acknowledgement

References

[1] Jorge Adrover, Matias Salibian-Barrera, and Ruben Zamar. Globally robust inference for the location and simple linear regression models. *Journal of Statistical Planning and Inference*,

119(2):353–375, 2004.

[2] Jean-Yves Audibert and Alexandre B. Tsybakov. Fast learning rates for plug-in classifiers. *The Annals of Statistics*, 35:608–633, 2007.

[3] B. Bauer and M. Kohler. On deep learning as a remedy for the curse of dimensionality in nonparametric regression. *The Annals of Statistics*, 47:2261–2285, 2019.

[4] J. R. Berrendero, A. Cuevas, and J. L. Torrecilla. On the use of reproducing kernel hilbert spaces in functional classification. *Journal of the American Statistical Association*, 113(523):1210–1218, 2018.

[5] Thijs Bos and Johannes Schmidt-Hieber. Convergence rates of deep relu networks for multiclass classification. *arXiv:2108.00969*, 2021.

[6] T. Tony Cai and Linjun Zhang. A convex optimization approach to high-dimensional sparse quadratic discriminant analysis. *arXiv:1912.02872*, 2019.

[7] T. Tony Cai and Linjun Zhang. High dimensional linear discriminant analysis: optimality, adaptive algorithm and missing data. *Journal of the Royal Statistical Society. Series B. Statistical Methodology*, 81(4):675–705, 2019.

[8] F. Chamroukhi and H. Glotin. Mixture model-based functional discriminant analysis for curve classification. *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2012.

[9] Xiongtao Dai, Hans-Georg Müller, and Fang Yao. Optimal Bayes classifiers for functional data and density ratios. *Biometrika*, 104(3):545–560, 2017.

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

[11] A. Delaigle, P. Hall, and N. Bathia. Componentwise classification and clustering of functional data. *Biometrika*, 99(2):299–313, 2012.

[12] Aurore Delaigle and Peter Hall. Classification using censored functional data. *Journal of the American Statistical Association*, 108(504):1269–1283, 2013.

[13] Farzan Farnia and David Tse. A minimax approach to supervised learning. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, page 4240–4248, Red Hook, NY, USA, 2016.

[14] Pedro Galeano, Esdras Joseph, and Rosa E. Lillo. The Mahalanobis distance for functional data with applications to classification. *Technometrics*, 57(2):281–291, 2015.

[15] Travor J. Hastie and Robert J. Tibshirani. *Generalized Additive Models*. Chapman & Hall/CRC, 1990.

[16] T. Hsing and R. Eubank. *Theoretical foundations of functional data analysis, with an introduction to linear operators*. Wiley Series in Probability and Statistics. John Wiley & Sons, Ltd., Chichester, 2015.

[17] Tianyang Hu, Zuofeng Shang, and Guang Cheng. Sharp rate of convergence for deep neural network classifiers under the teacher-student setting. *arXiv:2001.06892*, 2020.

[18] Yongdai Kim, Ilsang Ohn, and Dongha Kim. Fast convergence rates of deep neural networks for classification. *Neural Networks*, 138:179–197, 2021.

[19] P. Kokoszka and M. Reimherr. *Introduction to functional data analysis*. Texts in Statistical Science Series. CRC Press, Boca Raton, FL, 2017.

[20] Guillaume Lecué. Classification with minimax fast rates for classes of Bayes rules with sparse representation. *Electronic Journal of Statistics*, 2:741–773, 2008.

[21] X. Leng and H.G. Müller. Classification using functional data analysis for temporal gene expression data. *Bioinformatics*, 22:68–76, 2006.

[22] Kexuan Li, Fangfang Wang, Ruiqi Liu, Fan Yang, and Zuofeng Shang. Calibrating multi-dimensional complex ode from noisy data via deep neural networks. *arXiv:2106.03591*, 2021.

[23] Yi Lin. Tensor product space anova models. *The Annals of Statistics*, 28:734 – 755, 2000.

[24] Ruiqi Liu, Ben Boukai, and Zuofeng Shang. Optimal nonparametric inference via deep neural network. *Journal of Mathematical Analysis and Applications*, 505:125561, 2022.

[25] Ruiqi Liu, Zuofeng Shang, and Guang Cheng. On deep instrumental variables estimate. *arXiv:2004.14954*, 2021.

[26] Enno Mammen and Alexandre B. Tsybakov. Smooth discrimination analysis. *The Annals of Statistics*, 27:1808–1829, 1999.

[27] Santiago Mazuelas, Andrea Zanoni, and Aritz Perez. Minimax classification with 0-1 loss and performance guarantees. *arXiv:2010.07964*, 2020.

[28] Yangling Mu and Fred H. Gage. Adult hippocampal neurogenesis and its role in alzheimer's disease. *Mol Neurodegener.*, 2011.

[29] Juhyun Park, Jeongyoun Ahn, and Yongho Jeon. Sparse functional linear discriminant analysis. *arXiv:2012.06488*, 2020.

[30] Philipp Petersen and Felix Voigtlaender. Optimal approximation of piecewise smooth functions using deep relu neural networks. *Neural Networks*, 108:296–330, 2018.

[31] Fabric Rossi, Delannay Nicolas, Brieuc Conan-Guez, and Michel Verleysen. Representation of functional data in neural networks. *Neurocomputing*, 64:183–210, 2005.

[32] J. Schmidt-Hieber. Nonparametric regression using deep neural networks with relu activation function. *The Annals of Statistics*, 48(4):1875–1897, 2020.

[33] H. Shin. An extension of fisher's discriminant analysis for stochastic processes. *Journal of Multivariate Analysis*, 99:1191—-1216, 2008.

[34] J. Song, W. Deng, H. Lee, and D. Kwon. Optimal classification for time-course gene expression data using functional data analysis. *Biometrika*, 103(1):147–159, 2016.

[35] Ingo Steinwart and Andreas Christmann. *Support vector machines*. Springer Science and Business Media, 2008.

[36] Taiji Suzuki. Adaptivity of deep relu network for learning in besov and mixed smooth besov spaces: optimal rate and curse of dimensionality. *arXiv preprint arXiv:1810.08033*, 2018.

[37] Alexandre B. Tsybakov. Optimal aggregation of classifiers in statistical learning. *The Annals of Statistics*, 32:135–166, 2004.

[38] Alexandre B. Tsybakov. *Introduction to nonparametric estimation. Springer Series in Statistics*. Springer, New York, 2009.

[39] J.L. Wang, J. M. Chiou, and H. G. Müller. Functional data analysis. *Annual Review of Statistics and Its Application*, 3:257–295, 2016.

[40] Shuoyang Wang, Guanqun Cao, and Zuofeng Shang. Estimation of the mean function of functional data via deep neural networks. *Stat*, e393, 2021.

[41] Shuoyang Wang, Zuofeng Shang, Guanqun Cao, and Jun S. Liu. Optimal classification for functional data. *arXiv:2103.00569*, 2021.

[42] Dmitry Yarotsky. Error bounds for approximations with deep relu networks. *Neural Network*, 94:103–114, 2021.