

Functional Multi-Layer Perceptron: a Nonlinear Tool for Functional Data Analysis

Fabrice Rossi, Brieuc Conan-Guez

► **To cite this version:**

Fabrice Rossi, Brieuc Conan-Guez. Functional Multi-Layer Perceptron: a Nonlinear Tool for Functional Data Analysis. [Research Report] RR-5228, INRIA. 2004, pp.35. inria-00070768

HAL Id: inria-00070768

<https://hal.inria.fr/inria-00070768>

Submitted on 19 May 2006

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Functional Multi-Layer Perceptron: a Nonlinear Tool for Functional Data Analysis

Fabrice Rossi — Brieuc Conan-Guez

N° 5228

Juin 2004

Thème COG



*Rapport
de recherche*

Functional Multi-Layer Perceptron: a Nonlinear Tool for Functional Data Analysis

Fabrice Rossi^{*†}, Brieuc Conan-Guez[‡]

Thème COG — Systèmes cognitifs
Projets AxIS

Rapport de recherche n° 5228 — Juin 2004 — 35 pages

Abstract: This paper is an improved version of [27] in which we study a natural extension of Multi-Layer Perceptrons (MLP) to functional inputs. We show that fundamental results for classical MLP can be extended to functional MLP. We obtain universal approximation results that show the expressive power of functional MLP is comparable to that of numerical MLP. We obtain consistency results which imply that the estimation of optimal parameters for functional MLP is statistically well defined. We finally show on simulated and real world data that the proposed model performs in a very satisfactory way.

Key-words: neural networks, multi-layer perceptron, functional data analysis, universal approximation, consistency, nonlinear regression, curves classification, spectrometric data

* CEREMADE, UMR CNRS 7534, Université Paris-IX Dauphine

† Fabrice.Rossi@inria.fr

‡ Brieuc.Conan-Guez@inria.fr

Perceptron multi-couches fonctionnel : un modèle non-linéaire pour l'analyse de données fonctionnelles

Résumé : Ce rapport est une version améliorée de [27] dans lequel nous étudions une extension naturelle du Perceptron Multi-Couches (PMC) à des entrées fonctionnelles. Nous montrons que les résultats fondamentaux existants pour le PMC classique peuvent être transposés au cas du PMC fonctionnel. Nous obtenons des résultats d'approximation universelle, qui montrent que les capacités d'approximation du PMC fonctionnel sont comparables à celles du PMC classique. Nous obtenons des résultats de consistance, qui montrent que l'estimation des paramètres optimaux du PMC fonctionnel est statistiquement bien définie. Enfin, nous étudions le comportement du PMC fonctionnel sur des données simulées et réelles.

Mots-clés : réseaux de neurones, perceptron multi-couches, analyse de données fonctionnelles, approximation universelle, consistance, régression non-linéaire, discrimination de courbes, données spectrométriques

Contents

1	Introduction	4
2	Functional Multi-Layer Perceptrons	5
2.1	Functional data	5
2.2	Functional neurons	5
2.3	Functional MLP	6
3	Universal approximation	7
3.1	Definitions and notations	7
3.1.1	Functional spaces and metrics	7
3.1.2	One hidden layer perceptrons	7
3.2	Universal approximation with functional MLP	8
3.3	Discussion	8
4	Consistency of Functional MLP learning	9
4.1	Introduction	9
4.2	Probabilistic framework	9
4.2.1	Functional data	9
4.2.2	Parametric model	10
4.2.3	Optimal model and consistency	11
4.2.4	Function observations	12
4.3	Consistency	13
5	Alternative methods for functional inputs	13
5.1	Functions observed at identical points	13
5.2	Function representation	14
6	Experiments	16
6.1	Introduction and experimental setting	16
6.2	Breiman waves	17
6.2.1	Classification results	17
6.2.2	Parameter numbers	18
6.2.3	Pre-smoothing	19
6.2.4	Comments	20
6.3	Spectrometric data	21
6.3.1	Raw data	21
6.3.2	Second order derivatives	22
6.3.3	Comments	23
6.4	Conclusions	23
7	Conclusion	24
	Proofs	25

1 Introduction

Functional Data Analysis (FDA, see [26] for a comprehensive introduction to FDA methods) is an extension of traditional data analysis to functional data. In this framework, each individual is characterized by one or more real valued functions, rather than by a vector of \mathbb{R}^n . An important feature of FDA is its ability to take into account dependencies between numerical measurements that describe an individual, especially smoothness. If we represent for instance the size of a child at different ages by a vector, traditional methods generally consider each value to be independent of the others. In FDA, the size is represented as a function (in general a regular one) that maps measurement times to centimeters.

In order to deal with irregular measurements and to allow numerical manipulation of functions, FDA replaces actual observations by a simple functional representation. Spline based approximation is the most commonly used method, as it represents each individual by a smooth function. Kernel or wavelet based approximations are also used. FDA has been successfully applied to real problems such as climatic variation forecasting ([6]), acidification process studying ([1]), analysis of children size evolution ([26]), land usage prediction based on satellite images ([3]), etc.

In this paper, we focus on a precise yet very general task: we assume that we observe functions associated to a classical target variable. This variable can be for instance a class label, in which case we perform supervised classification. If the variable is a real valued vector, we perform a regression. The key idea is that, whereas individuals are described thanks to functions, we still want to predict a traditional numerical value. In mathematical terms, we have n examples described by $s + 1$ variables, $(g_1^i, \dots, g_s^i, t^i)_{i \in \{1, \dots, n\}}$, where t^i is the target variable (with $t^i \in \mathbb{R}^o$) and where each g_t^i is a function belonging to a given functional space. The problem is to predict t^i based on (g_1^i, \dots, g_s^i) . In the framework of FDA, several methods have been proposed to solve this kind of problem, for instance the linear functional model (see e.g. [17], [25], [26], [8], [9] and [21]), functional discriminant analysis (e.g. [22]), functional Slice Inverse Regression (see [24] for the classical SIR and [16] for its functional version) and non-parametric kernel based functional estimators (see [14], [15] and [13]).

In this paper, we show how Multi-Layer Perceptrons (MLP) can be directly applied to functional data, so as to provide nonlinear semi-parametric function classification and regression. We introduce a major difference with traditional FDA methods: our model works directly with the studied functions, without using a simplified representation. This avoids restrictions on the functional weight representation which can therefore be adapted to the context. For instance, functional data with low dimensional input spaces can be manipulated thanks to generalized linear models (such as splines), whereas MLP are used for functions with high dimensional input spaces.

When functional data are perfectly known, the extension of MLP we propose is a particular case of an extension proposed and studied from purely theoretical point of view in [31]. In [31], the author shows that traditional universal approximation results for MLP can be extended to (almost) arbitrary input spaces, including infinite dimensional vectorial spaces. These results rely on the approximation of continuous linear forms defined on the

MLP input space. In our work, we show how to carry out this kind of approximation in practice, for instance by using traditional MLP. We show this way that functional MLP are universal approximators and therefore that they can be used to model complex dependencies between a real valued target variable and functional inputs.

Moreover, we show that training a parametric functional MLP on a finite number of function examples is statistically valid, as the optimal parameters obtained thanks to those examples provide a consistent estimation of asymptotic optimal parameters, even if we assume limited knowledge on each function example (i.e., each function is only known thanks to a finite number of (input, output) pairs). This is a direct translation of classical results, presented in [32] for instance, available for numerical MLP.

The rest of the paper is organized as follows. In the first part, we assume that we have perfect knowledge of manipulated functions: we start by introducing in section 2 the proposed functional MLP model. Then we show in section 3 how the results of [31] can be adapted to functional MLP to show they are universal approximators. In the second part, we take into account sampling: consistency of functional MLP training is studied in section 4. Section 5 compares our approach to alternative neural solutions, on a theoretical point of view. Then, section 6 gives some experimental results both on simulated and real world data. Proofs are presented in section 7.

2 Functional Multi-Layer Perceptrons

2.1 Functional data

As stated in the introduction, an observation is described by $s + 1$ values, (g_1, \dots, g_s, t) , where each g_l is a function (and $t \in \mathbb{R}^o$). More precisely, we assume that μ_l is a σ -finite positive Borel measure defined on \mathbb{R}^{u_l} and that g_l belongs to $L^{p_l}(\mu_l)$.

2.2 Functional neurons

The extension of numerical neurons to functional inputs is straightforward. Indeed a n input MLP neuron is characterized by a fixed activation function, T , a function from \mathbb{R} to \mathbb{R} , by a vector from \mathbb{R}^n (the weight vector, w) and by a real valued threshold, b . Given a vectorial input $x \in \mathbb{R}^n$, the output of the neuron is $N(x) = T(w.x + b)$.

This formula is based on the linear form $x \mapsto w.x$. When $x = (g_1, \dots, g_s) \in L^{p_1}(\mu_1) \times \dots \times L^{p_s}(\mu_s)$, a linear form can be constructed thanks to integrals, for instance:

$$(g_1, \dots, g_s) \mapsto \sum_{l=1}^s \int f_l g_l d\mu_l, \tag{1}$$

where (f_1, \dots, f_s) are measurable functions chosen such that $f_l g_l \in L^1(\mu_l)$. Using this linear form, we can define a functional neuron:

Definition 1. A functional neuron on $E = L^{p_1}(\mu_1) \times \dots \times L^{p_s}(\mu_s)$ is defined thanks to a fixed activation function T from \mathbb{R} to \mathbb{R} , weight functions f_l (such that $f_l g_l \in L^1(\mu_l)$) and a real valued threshold, b . It calculates

$$N(g_1, \dots, g_s) = T \left(b + \sum_{l=1}^s \int f_l g_l d\mu_l \right). \quad (2)$$

This functional neuron is a special case of general neurons proposed in [29, 30, 31]. The main drawback of this model is that it uses functional weights rather than numerical ones. This problem can be solved by using parametric representation of functions. More precisely, we assume given s functions F_1, \dots, F_s such that (hypothesis H_a):

1. $W_l \subset \mathbb{R}^{v_l}$
2. F_l is a function from $W_l \times \mathbb{R}^{q_l}$ to \mathbb{R}
3. for each $w_l \in W_l$, $F_l(w_l, \cdot) \in L^{q_l}(\mu_l)$ where q_l is the conjugate exponent associated to p_l

For instance, F_l can be implemented thanks to a numerical MLP (in this case, w_l is the weight vector of the MLP) or thanks to the first functions of a topological basis of $L^{q_l}(\mu_l)$ (in this case, we have $F_l(w_l, x) = \sum_{i=1}^{v_l} w_{li} \psi_i(x)$, where $(\psi_i)_{i \in \mathbb{N}}$ is the considered topological basis).

We can now introduce the definition of a parametric functional neuron:

Definition 2. A parametric functional neuron on $E = L^{p_1}(\mu_1) \times \dots \times L^{p_s}(\mu_s)$ is defined thanks to a fixed activation function T from \mathbb{R} to \mathbb{R} , a weight vector $w \in W_1 \times \dots \times W_s$ and a real valued threshold, b . It calculates

$$N(g_1, \dots, g_s) = T \left(b + \sum_{l=1}^s \int F_l(w_l, x) g_l(x) d\mu_l(x) \right). \quad (3)$$

2.3 Functional MLP

As a functional neuron gives a real output, we have to use numerical neurons except in the first layer of a functional MLP. In particular, a one hidden layer parametric functional perceptron with one functional input and one real output computes a function of the following form:

$$H(g) = \sum_{i=1}^k a_i T \left(b_i + \int F_i(w_i, x) g(x) d\mu(x) \right), \quad (4)$$

where the a_i are real numbers, as well as the b_i , and w_i are parameter vectors for F_i .

Of course, it is obvious to extend those definitions to more than one output and/or hidden layer. The only difference between a functional n -hidden layer perceptron and a numerical one is that, as stated above, we use functional neurons only in the first layer. It is also obvious to define a general functional MLP by using functional neurons rather than parametric functional neurons.

3 Universal approximation

3.1 Definitions and notations

We use notations and definitions from [31].

3.1.1 Functional spaces and metrics

We denote $C(A, B)$ the set of continuous functions from A to B , where A and B are two topological spaces. As a special case, C^n is the set of continuous functions from \mathbb{R}^n to \mathbb{R} . M^n is the set of (Borel) measurable functions from \mathbb{R}^n to \mathbb{R} . We denote d_C the metric on M^n that gives uniform convergence over compact subsets:

$$d_C(f, g) = \sum_{n \in \mathbb{N}^*} \frac{1}{2^n} \min \left\{ \sup_{|x| \leq n} |f(x) - g(x)|, 1 \right\}. \quad (5)$$

When K is a compact subset of X a topological space, we define ρ_K a metric on the set of functions from K to \mathbb{R} by:

$$\rho_K(f, g) = \sup_{x \in K} |f(x) - g(x)|. \quad (6)$$

Definition 3. Let X be a metric space with d the associated metric. Let C and S be two subsets of X . S is d -outside dense in C if the d -closure of S contains C , and S is d -inside dense in C if the d -closure of $S \cap C$ contains C .

When $C = X$, d -inside density is equivalent to d -outside density and is simply called d -density.

3.1.2 One hidden layer perceptrons

Definition 4. If T is a function from \mathbb{R} to \mathbb{R} and n a positive integer, S_T^n is the set of functions exactly computed by one hidden layer perceptrons with n inputs and one output, and using T as activation function, i.e. the set of functions of the form $h(x) = \sum_{i=1}^p \beta_i T(w_i \cdot x + b_i)$ where $p \in \mathbb{N}$, $\beta_i \in \mathbb{R}$, and $(w_i, b_i) \in \mathbb{R}^{n+1}$.

Definition 5. If X is a topological vector space, A a subset of X^* and T a function from \mathbb{R} to \mathbb{R} , $S_T^X(A)$ is the set of functions exactly computed by one hidden layer generalized perceptrons with input in X , one real output, and weight forms in A , i.e. the set of functions from X to \mathbb{R} of the form $h(x) = \sum_{i=1}^p \beta_i T(l_i(x) + b_i)$ where $p \in \mathbb{N}$, $\beta_i \in \mathbb{R}$, $b_i \in \mathbb{R}$ and $l_i \in A$.

Note that A can in fact be any set of functions from X to \mathbb{R} , in which case we do not introduce constant terms b_i .

According to this definition, functional one hidden layer perceptrons are a special case of Stinchcombe generalized perceptrons in which X is a product of L^p spaces and A is given by linear forms of the form $l(g_1, \dots, g_s) = \sum_{l=1}^s \int f_l g_l d\mu_l$ (or $l(g_1, \dots, g_s) = \sum_{l=1}^s \int F_l(w_l, x) g_l(x) d\mu_l(x)$ for parametric functional perceptrons).

3.2 Universal approximation with functional MLP

Several approximation results show that $S_T^X(A)$ is inside or outside dense in different functional spaces. Indeed [31] (as well as [30] and [10]) proposes approximation results for $S_T^X(A)$ for almost arbitrary spaces X (see theorem 5.1 and corollaries 5.1.2 and 5.1.3 from [31]). In order to apply those general results to practical cases, complex technical properties have to be satisfied by A . In this section, we show that those properties are satisfied by very general functional one hidden layer perceptrons.

Corollary 1. *Let μ be a finite positive Borel measure on \mathbb{R}^n . Let $1 < p \leq \infty$ be an arbitrary real number and q be the conjugate exponent of p . Let V be a dense subset of $L^q(\mu)$. Let A_V be the set of linear forms on $L^p(\mu)$ of the form $l(f) = \int fg d\mu$, where $g \in V$. Let T be a measurable function from \mathbb{R} to \mathbb{R} such that S_T^1 is d_C -inside (resp. d_C -outside) dense in C^1 . Then $S_T^{L^p(\mu)}(A_V)$ is ρ_K -inside (resp. ρ_K -outside) dense in $C(K, \mathbb{R})$, where K is any compact subset of $L^p(\mu)$.*

Corollary 2. *Let μ be a finite positive compactly supported Borel measure on \mathbb{R}^n . Let T be a measurable function from \mathbb{R} to \mathbb{R} , such that S_T^1 is d_C -inside (resp. d_C -outside) dense in C^1 . Let V be a subset of $L^\infty(\mu)$ d_C -inside (or d_C -outside) dense in C^n . Then $S_T^{L^1(\mu)}(A_V)$ is ρ_K -outside dense in $C(K, \mathbb{R})$, where K is any compact subset of $L^1(\mu)$.*

3.3 Discussion

Corollary 1 shows that as long as we can approximate functions in $L^q(\mu)$ and in C^1 , then an one hidden layer perceptron can be used to approximate functions in $C(K, \mathbb{R})$, where K is a compact subset of $L^p(\mu)$. Previous works give very weak conditions on T that imply d_C inside or outside density of S_T^1 for C^1 , see for instance Theorem 1 in [23] and Theorem 1 in [20]. Basically, T must be non polynomial and Riemann integrable on a non-degenerate compact interval of \mathbb{R} , properties that are obviously satisfied by popular activation functions such as tanh.

The generalized MLP used in corollary 1 uses linear forms in A_V and is therefore a functional MLP with weight functions chosen in V a dense subset of $L^q(\mu)$. In practical situation, weight functions are represented thanks to parametric functions ($F(w, \cdot)$). This constraint does not introduce any problem, as long as we choose a parametric universal approximator for $L^q(\mu)$. Thanks to Theorem 1 of [19], we can use for instance one hidden layer perceptrons based on activation function U (i.e., $V = S_U^n$) as long as U is measurable, bounded and non constant (as $p > 1$, $q < \infty$ and Theorem 1 applies). Other models can be used (B-spline, wavelet, Fourier series, etc.) but imply in general additional restrictions on the considered functional space.

The proof of corollary 1 could be extended to $p = 1$, and therefore, one might wonder why corollary 2 is useful. As pointed out in the introduction of [31], no S_T^n set is dense in $L^\infty(\mu)$. Therefore, corollary 1 main assumption (V is dense in $L^q(\mu)$) cannot be satisfied by MLP based approximation. This reduces greatly the interest of corollary 1 for $p = 1$. That's why corollary 2 is useful: as shown for instance by Theorem 1 of [20], S_U^n can be used to

provide approximation to continuous functions on a compact set. Therefore, the situation for $p = 1$ is quite similar to the one that stands for $p > 1$, except that the measure has to be compactly supported.

This means that when K is a compact subset of a $L^p(\mu)$ functional space, any function from $C(K, \mathbb{R})$ can be approximated to a given precision level by a functional MLP that uses a finite number of parameters (because linear forms can be represented for instance thanks to numerical MLPs). Despite the radical change in the input space dimension (from \mathbb{R}^n to a compact subset of a functional space), we can still effectively approximate continuous functions.

It is very common in FDA to assume that studied functions are smooth, that is at least continuous. If we only consider compact input spaces for those functions, their case is covered by corollary 1. Indeed, continuous functions (or more regular functions) on a compact subset Z of \mathbb{R}^n are obviously elements of $L^\infty(\lambda)$ where λ is the restriction of the Lebesgue measure to Z . Moreover a compact subset K of a space of regular functions (considered with the uniform norm) is a compact subset of $L^\infty(\lambda)$. This means that any continuous function from K to \mathbb{R} can be approximated by a functional MLP as long as $L^1(\lambda)$ can also be approximated (this can be done thanks to $S_{\mathcal{U}}^n$ [19]).

Extension of proposed corollaries to multiple functional inputs is straightforward. In fact, corollaries are based on approximation of linear forms on X the input space of extended neurons. When $X = L^{p_1}(\mu_1) \times \dots \times L^{p_r}(\mu_r)$, approximation of elements of X^* is obtained thanks to approximations of elements of $(L^{p_i}(\mu_i))^*$, because a linear form on X is a linear combination of linear forms on $L^{p_i}(\mu_i)$ (this fact was used to define the functional neuron).

4 Consistency of Functional MLP learning

4.1 Introduction

As explained in the introduction, our goal is to explain a target variable $t \in \mathbb{R}^o$ thanks to functional observations (g_1, \dots, g_s) . Basically, we assume that there is a functional relationship such that $t \simeq F(g_1, \dots, g_s)$ and we try to model F thanks to a functional MLP. Thanks to universal approximation results given in the previous section, we know that any regular F can be approximated by a functional MLP. Nevertheless, an important problem remains: F is obviously unknown and a correct approximation as to be constructed thanks to a limited number of examples of this mapping.

4.2 Probabilistic framework

4.2.1 Functional data

Let us now describe the probabilistic framework of our problem. All random quantities will be defined on a given probability space (Ω, \mathcal{A}, P) . For the sake of simplicity, we consider only the case of an unique functional input. More precisely, we make the following hypothesis (H_b) :

1. Z is a compact subset of \mathbb{R}^u
2. $(G^i, T^i)_{i \in \mathbb{N}}$ is an i.i.d. sequence of random elements with values in $C(Z, \mathbb{R}) \times \mathbb{R}^o$ (i.e., each G^i is a measurable function from Ω to $C(Z, \mathbb{R})$ considered with its Borel sigma algebra and each T^i is a random vector in \mathbb{R}^o , and the sequence is i.i.d.)

Hypothesis on the observed functions are quite different from those of corollaries 1 and 2: on the one hand H_b are stronger than corollaries hypothesis as they consider only continuous functions defined on a compact set, on the other hand they are weaker as observed functions do not belong to a compact subset of $C(Z, \mathbb{R})$.

4.2.2 Parametric model

We try to model the relationship between G^i and T^i thanks to a special kind of parametric model (a parametric functional MLP) that has the following form:

$$H(w, g) = U \left(w_0, \int F_1(w_1, x) g(x) d\mu(x), \dots, \int F_k(w_k, x) g(x) d\mu(x) \right), \quad (7)$$

where $w = (w_0, w_1, \dots, w_k) \in W = W_0 \times W_1 \times \dots \times W_k$, the F_l are parametric models as in parametric neurons, U is a regular function and μ a finite positive Borel measure (defined on Z). This parametric form is quite similar to the one proposed in the context of Slice Inverse Regression by [16]. Our main motivation here is to use a general form that covers functional multi-layer perceptrons without making too much hypothesis on their architecture (number of layers, activation functions, linear terms, etc.). For instance, if U is defined as follows:

$$U(w_0, o_1, \dots, o_k) = \sum_{l=1}^k a_l T(b_l + o_l), \quad (8)$$

with $w_0 = (a_1, b_1, \dots, a_k, b_k)$, then $H(w, g)$ is exactly the output of a functional one hidden layer perceptron, as given by equation 4. As a side effect, we cover any model that uses integrals to transform an input function into a real number.

Some restrictions are needed on F_l functions and on U (hypothesis H_c):

1. for $0 \leq l \leq k$, W_l is a compact subset of \mathbb{R}^{v_l}
2. for $1 \leq l \leq k$, F_l is a function from $W_l \times Z$ to \mathbb{R} such that:
 - (a) for each $x \in Z$, $F_l(\cdot, x)$ is continuous
 - (b) for each $w_l \in W_l$, $F_l(w_l, \cdot)$ is measurable
 - (c) F_l is dominated on W_l , i.e., there is a measurable function $d_l \in L^p(\mu)$ (with $p \geq 1$) such that for for all $w \in W_l$ and $x \in Z$, $|F_l(w, x)| \leq d_l(x)$.
3. U is an uniformly continuous function from $W_0 \times \mathbb{R}^k$ to \mathbb{R}^o

4. U is bounded

Hypothesis H_c are quite natural and are fulfilled in practical settings:

- Compactness of the parameter space is a classical hypothesis in consistency results.
- Useful choices for F_l are numerical MLP and basis expansions: for the former, continuity is mandatory in practice as optimal parameters are obtained thanks to gradient based algorithms (and therefore F_l is in general differentiable with respect to w_l); for the latter, continuity is obvious as F_l is linear with respect to w_l .
- As stated before, when F_l is obtained thanks to a numerical MLP, it is a continuous function. As W_l and Z are compact sets, the domination hypothesis is automatically fulfilled. When F_l is obtained thanks to basis expansion, a natural hypothesis is to assume that basis functions belong to $L^p(\mu)$. Then, compactness of W_l implies again that the domination hypothesis is fulfilled.
- U corresponds to the non functional part of a functional MLP, it is in general natural to assume that it is uniformly continuous. Indeed, popular activation functions such as tanh and the logistic function are uniformly continuous and moreover, W_0 is compact, therefore when U represents a MLP based on standard activation functions, it is uniformly continuous. Moreover, popular activation functions are also bounded and the assumption that U is bounded is also natural.

4.2.3 Optimal model and consistency

The learning phase in neural network applications consists in finding the best parameters for a given task. In our framework, we assume given a distance¹ c on \mathbb{R}^o and we assess the quality of the neural model at the evaluation point G^i thanks to $c(T^i, H(G^i, w))$. We define the global error made by the parametric model H for parameters $w \in W$ by:

$$\lambda(w) = E(c(T^1, H(G^1, w))), \quad (9)$$

where E means expectation. Learning is in fact a parameter estimation problem in which we try to optimize $\lambda(w)$ in order to find a vector $w \in W^*$, where $W^* \subset W$ is the set of minimizer of $\lambda(w)$. The practical problem is that $\lambda(w)$ cannot be exactly calculated and is approximated thanks to a finite number of realizations of (G^i, T^i) . More precisely, we define an empirical error by:

$$\hat{\lambda}_n(w) = \frac{1}{n} \sum_{i=1}^n c(T^i, H(G^i, w)). \quad (10)$$

This empirical error can be minimized to produce \hat{w}_n an estimation of an optimal parameter vector. [32] shows that for numerical MLP, \hat{w}_n is a strongly consistent estimation of an optimal parameter vector. More precisely, if d denotes the distance on W , then

¹ c has not really to be a distance, it can be any continuous positive function.

$\lim_{n \rightarrow \infty} d(\widehat{w}_n, W^*) = 0$ almost surely. Among technical hypothesis needed to ensure this result, we adapt a domination hypothesis to the functional framework (hypothesis H_d): $c(T^i, H(G^i, w))$ has to be dominated, in the sense that there is a positive function c_{\max} from \mathbb{R}^o to \mathbb{R} such that:

1. $\forall w \in W, g \in C(Z, \mathbb{R})$ and $t \in \mathbb{R}^o, c(t, H(g, w)) \leq c_{\max}(t)$
2. $E(c_{\max}(T_1)) < \infty$

For functional MLP, hypothesis H_d are quite natural. Indeed, hypothesis H_c (4) makes $H(g, w)$ bounded and therefore domination turns into an hypothesis on T_1 and c . For instance if c is the Euclidean distance in \mathbb{R}^o , then domination is obtained if T_1 has a second order moment.

Compared to the numerical case, we have two additional difficulties in the functional framework: we are working with random elements with values in a functional space, whereas [32] assumes that observations belong to a finite dimensional space; moreover, perfect knowledge of observed functions is seldom the case and we have to take into account that functions are measured at a finite number of observation points.

4.2.4 Function observations

In practical situations, each observed function is described by a finite number of input/output pairs, such as $(x_j, g(x_j))_{j \in \{1, \dots, m\}}$. We choose the following mathematical model (hypothesis H_e):

1. $(X_j^i)_{i \in \mathbb{N} j \in \mathbb{N}}$ is a sequence of independent sequences of random variables defined on (Ω, \mathcal{A}, P) and with values in Z .
2. All X_j^i are identically distributed and the induced probability measure on Z is $\mu = P_X$.
3. $(\mathcal{E}_j^i)_{i \in \mathbb{N} j \in \mathbb{N}}$ is a sequence of independent sequences of random variables defined on (Ω, \mathcal{A}, P) and with values in \mathbb{R} .
4. For all $i, (\mathcal{E}_j^i)_{j \in \mathbb{N}}$ and $(X_j^i)_{j \in \mathbb{N}}$ are independent.
5. $E(\mathcal{E}_j^i) = 0$ and $E(|\mathcal{E}_j^i|^q) < \infty$, where q is the conjugate exponent to p used in hypothesis H_c (2-c).

For each i , the sequence $(X_j^i)_{j \in \mathbb{N}}$ corresponds to observation points for the function G^i and the sequence $(\mathcal{E}_j^i)_{j \in \mathbb{N}}$ corresponds to measurement errors for these observation points. More precisely, if g^i, x_j^i and ε_j^i are respectively realizations of G^i, X_j^i and \mathcal{E}_j^i , we assume that we observe the sequence: $y_j^i = g^i(x_j^i) + \varepsilon_j^i$. Moreover, we assume that we know only the m^i first values of this sequence.

Hypothesis H_e are natural in this framework, especially independence. The main hypothesis is H_e (2), which says that the way observation points are randomly chosen (i.e.,

P_X) corresponds to the way integrals are calculated (μ). On an intuitive point of view, this means that when an input function is matched to functional weights thanks to integral calculation, probable observation points have more weight than less probable ones. This is quite natural.

As functions are only known thanks to observations, we cannot compute anymore the integrals which are approximated thanks to empirical means. More precisely, we replace $\int F_l(w_l, x) g^i(x) d\mu(x)$ by:

$$\frac{1}{m^i} \sum_{j=1}^{m^i} F_l(w_l, x_j^i) y_j^i. \quad (11)$$

Therefore, the empirical error $\widehat{\lambda}_n(w)$ given in equation 10 is approximated by the following empirical error:

$$\lambda_n^m(w) = \frac{1}{n} \sum_{i=1}^n c \left(t^i, U \left(w_o, \frac{1}{m^i} \sum_{j=1}^{m^i} F_1(w_1, x_j^i) y_j^i, \dots, \frac{1}{m^i} \sum_{j=1}^{m^i} F_k(w_k, x_j^i) y_j^i \right) \right), \quad (12)$$

where t^i is a realization of T^i and $m = \inf_{1 \leq i \leq n} m^i$.

This empirical error, which is based on finite number of numerical values, is easy to evaluate in practice and can be used to obtain empirical optimal parameters, w_n^m . Our goal is to show that w_n^m is a consistent estimator of an optimal parameter vector, i.e. converges to W^* .

4.3 Consistency

Consistency of the proposed estimation of optimal parameters is given by the following theorem:

Theorem 1. *Under hypothesis H_b , H_c , H_d and H_e , we have P -almost surely:*

$$\lim_{n \rightarrow \infty} \lim_{m \rightarrow \infty} d(w_n^m, W^*) = 0.$$

The theorem is an extension of [32]. It suffers from a small limitation: the limit is a sequential one, which means that in order to reach a given distance to W^* , the number of evaluation points for each function (m) depends on the number of functions (n).

5 Alternative methods for functional inputs

5.1 Functions observed at identical points

In some particular cases, functions are all observed thanks to an unique sequence of observation points, that is there is a sequence $(x_j)_{j \in \mathbb{N}}$ such that for any considered function g , we know $g(x_j)$ for all j . Moreover, we assume that we use the same number of observation

points for each function (denoted by m). These cases include for instance situations in which measurement points are under user control (e.g., spectroscopic measurements corresponding to specific frequencies). Of course, this case is covered by theorem 1.

On a practical point of view, the situation is clearly simpler than the general one. Indeed each function g can be considered as a vector in \mathbb{R}^m , i.e., $(g(x_1), \dots, g(x_m))$. Therefore, we can submit these multivariate observations to a numerical MLP. This approach was proposed in [11]. Let us consider the special case of a single hidden layer perceptron with one real output. Such a MLP maps a function g to:

$$V(g) = \sum_{i=1}^k a_i T \left(b_i + \sum_{j=1}^m c_{ij} g(x_j) \right). \quad (13)$$

In such a setting, our model maps g to:

$$H(g) = \sum_{i=1}^k a_i T \left(b_i + \frac{1}{m} \sum_{j=1}^m F_i(w_i, x_j) g(x_j) \right). \quad (14)$$

On a practical point of view, the main advantage of our approach over the numerical one in this setting is the increased flexibility induced by the use of the parametric functions F_i . We can for instance take into account smoothness of observed functions by using simple parametric functions (i.e., MLP with a small number of hidden nodes, B-splines with just a few nodes, etc.). This allows to reduce the number of free parameters in the model while incorporating expert knowledge into it, whereas in the numerical approach, we need in each neuron one connection weight for each function observation point.

Moreover, it is obvious that an appropriate choice of parametric functions F_i allows to reproduce exactly the numerical model, which appears this way as a special case of the functional approach. Indeed each F_i can be an interpolation spline or a kernel based model designed such that for any set of weights c_{ij} , there are weight vectors w_i such that $F_i(w_i, x_j) = c_{ij}$.

Finally, the universal approximation result given in [11] is less general than ours as it relies on uniform sampling.

For all those reasons, we believe that the functional approach is more interesting than the multivariate approach, even for uniformly sampled functions. Experiments exposed in section 6 confirm this point of view.

5.2 Function representation

When functions are not observed at identical evaluation points, there is still a natural alternative approach to ours. The main idea is simply to transform each functional observation into a representation that allows easy manipulation. More precisely, a list of observations $(x_j, g(x_j) + \epsilon_j)_{j \in \{1, \dots, m\}}$ is replaced by an approximation of g , $A(g)$ constructed thanks to the observations.

The only reasonable solution is to use a pseudo-linear model to approximate the input/output mapping for each observed function. Indeed, the number of input functions can be quite large in real world experiments and fitting a non linear model to each function will be very time consuming. Moreover, the only difference between $A(g)$ and g is that the former is known exactly whereas the latter is not. Representation does not solve the function manipulation problem. If we use non linear models, calculation of a scalar product between $A(g)$ and a weight function is still a complex problem that cannot be solved without an approximation method for integral calculation. We are more or less back to our original problem, except that we have now perfectly known functions (hopefully smoothed by the representation algorithm). We do not discuss this approach anymore because it is in fact an extended version of our method (whose theoretical properties remain to be studied).

The case of pseudo-linear models allows to construct what might be seen as an alternative to our approach. Indeed, $A(g)$ is obtained thanks to a truncated basis expansion, a very common approach in FDA thoroughly illustrated in [26] and more recently in [4]. First of all, we need to assume that studied functions belong to $L^2(\mu)$. We chose a free system of $L^2(\mu)$, $(\phi_i)_{1 \leq i \leq p}$. Then each list of observations $(x_j, g(x_j) + \epsilon_j)_{j \in \{1, \dots, m\}}$ is replaced by $A(g)$ the projection of g on the vectorial space spanned by ϕ_1, \dots, ϕ_p , denoted $span(\Phi_p)$. On a practical point of view, we simply calculate numerical parameters $\alpha_i(g)$ that minimize

$$\sum_{j=1}^m \left(g(x_j) + \epsilon_j - \sum_{i=1}^p \alpha_i(g) \phi_i(x_j) \right)^2.$$

This approach has two advantages over the general non linear representation technique. First it is faster as $\alpha_i(g)$ is obtained very efficiently thanks to some simple linear algebra. Second it can lead to a simplify neural model. Indeed we can submit the numerical vector that represents a function $((\alpha_i(g))_{1 \leq i \leq p})$ to a numerical MLP (even if the observation points depend on the function, because p is the same for all functions).

On a theoretical point of view, this solution is in fact a particular case of our approach. Indeed our approach is based on calculating an approximation of $\int fgd\mu$. In $L^2(\mu)$, this is the scalar product. Let us consider the special case where we constraint weight functions f to belong to $span(\Phi_p)$, i.e., $f = \sum_{i=1}^p \beta_i(f) \phi_i$. We have obviously

$$\int fgd\mu = \sum_{i=1}^p \beta_i(f) \int g\phi_i d\mu.$$

If we knew the real projection of g on $span(\Phi_p)$, $\Pi(g)$, we would be able to replace $\int g\phi_i d\mu$ by $\int \Pi(g)\phi_i d\mu$. This is not the case, but we can still assume that $\int g\phi_i d\mu$ is approximately equal to $\sum_{j=1}^p \alpha_j(g) \int \phi_j \phi_i d\mu$. Therefore $\int fgd\mu$ is approximately equal to $\sum_{i=1}^p \sum_{j=1}^p \beta_i(f) \alpha_j(g) \int \phi_j \phi_i d\mu$. Let us denote M the matrix $M_{ij} = \int \phi_i \phi_j d\mu$. As $(\phi_i)_{1 \leq i \leq p}$ is a free system, M is a full rank matrix. If we denote $\gamma(f) = M\beta(f)$, we have

$$\int fgd\mu \simeq \sum_{j=1}^p \gamma_j(f) \alpha_j(g).$$

Moreover, given a vector of coefficients c , we can define a function t by

$$t = \sum_{i=1}^p d_i \phi_i,$$

with $d = M^{-1}c$ such that

$$\int t g d\mu \simeq \sum_{j=1}^p \gamma_j(t) \alpha_j(g) = \sum_{j=1}^p c_j \alpha_j(g).$$

Therefore, a linear combination of the (approximate) coordinates of g on $\text{span}(\Phi_p)$, is always approximately equal to the scalar product of g with a well chosen weight function f . Our method approximates $\int f g d\mu$ by another formula. It is obvious that for the limit case, we end up with identical values and therefore that our approach contains as a special case the representation based approach. As in the previous section, this might be even clearer with a simple one hidden layer perceptron with an unique real output. The representation based approach maps g to

$$V(g) = \sum_{i=1}^k a_i T \left(b_i + \sum_{l=1}^p c_l^i \alpha_l(g) \right), \quad (15)$$

whereas our model gives

$$H(g) = \sum_{i=1}^k a_i T \left(b_i + \frac{1}{m} \sum_{j=1}^m (g(x_j) + \epsilon_j) \left(\sum_{l=1}^p d_l^i \phi_l(x_j) \right) \right). \quad (16)$$

According to the previous discussion, to obtain nearly identical values, we just have to choose d such that $d^i = M^{-1}c^i$, for all i . Of course, on a numerical point of view, results might be slightly different (as will be illustrated in the following section), but the truncated basis approach can still be considered as a different implementation of a special case of our approach. More sophisticated truncated basis approaches, involving for instance a roughness penalty as in [5], depart more from the solution proposed here and should be studied independently.

6 Experiments

6.1 Introduction and experimental setting

In the present section, we illustrate the proposed approach on two supervised classification experiments. The first dataset, studied in section 6.2, consists in the traditional waveform data introduced in [7]. In this synthetic example, the goal is to classify examples into three classes. The second dataset, studied in section 6.3, consists in a real world spectrometric problem in which near infrared absorbance spectra are used to recognize high fat and low fat meat samples.

Both datasets have been used in [15] to illustrate the efficiency of the non-parametric functional kernel based model proposed in the corresponding paper (and also in [14]). We will therefore compare results obtained thanks to neuronal approaches to functional and classical methods used in [15]. Those methods include the above mentioned kernel based model as well as the linear model, Partial Least Square Regression, CART, etc.

We have considered three variations of the Multi Layer Perceptron: the classical MLP applied to raw data, the functional approach presented in this paper and the alternate implementation of the functional approach based on projection on a B-spline basis (see section 5.2).

In all our experiments, we have used a conjugate gradient training algorithm, with 10 different random initializations. To avoid over-fitting, we used a weight decay penalization term. To select both the architecture of the MLP and the value of the weight decay constant, we have used k -fold cross-validation (with $k = 5$). Finally, performances of the selected MLP have been evaluated on a test sample.

6.2 Breiman waves

6.2.1 Classification results

We start our experiments with synthetic data, more precisely with waveform data introduced in [7]. This is a three-class problem in which each class is obtained thanks to convex combination of three shifted triangular waveforms. The generating waveforms are continuous curves defined on $[1, 21]$ by:

$$h_1(t) = \max(6 - |t - 11|, 0), \quad (17)$$

$$h_2(t) = h_1(t - 4), \quad (18)$$

$$h_3(t) = h_1(t + 4). \quad (19)$$

Functions to classify have the following general forms:

$$x(t) = uh_1(t) + (1 - u)h_2(t) \text{ for class 1,} \quad (20)$$

$$x(t) = uh_1(t) + (1 - u)h_3(t) \text{ for class 2,} \quad (21)$$

$$x(t) = uh_2(t) + (1 - u)h_3(t) \text{ for class 3,} \quad (22)$$

where $u \in]0, 1[$. In [7] each function is transformed into a vector from \mathbb{R}^{21} thanks to an uniform sampling on $[1, 21]$. An independent standard Gaussian noise is added to each observation.

In order to stay closer to the functional framework, we follow [15] and work therefore with vectors from \mathbb{R}^{101} which correspond to an uniform sampling of each function on $[1, 21]$. The training sample is obtained exactly as in [15] : we have 150 functions in each class (in order to build such a function the parameter u is chosen uniformly in $]0, 1[$, independently for each function) and an independent standard Gaussian noise is added to each observation. The test sample is generated with the same method but contains 250 functions in each class.

As explained in the introduction, we have compared three neuronal approaches: a naive approach in which \mathbb{R}^{101} vectors are directly submitted to a classical one hidden layer perceptron, our functional approach in which functional weights are represented thanks to B-splines and the alternate implementation of our method based on projection on the same B-splines basis. We refer to [15] for comparison with classical methods and the non parametric functional method introduced in the paper. Table 1 gives the obtained results for the three neural methods (MLP corresponds to the naive approach, FMLP to our functional approach and FpMLP to the alternate implementation of this approach). Results have been averaged over 50 simulations, exactly as in [15], so as to ease comparison with existing results.

Method	Test classification error rate	Standard deviation
MLP	0.098	0.013
FMLP	0.065	0.0096
FpMLP	0.072	0.011

Table 1: Waveform data

Results are very satisfactory. First of all, our functional approaches overcome the classical MLP method (the main functional approach gives the best results). Result summary provided by table 1 does not give complete information. Indeed, as for each simulation the same data set is used for each method, a direct comparison between obtained results is possible. An important result is that for *all* simulations, functional approaches overcome the classical MLP. The mean performance increase is 3.2 percent for our main implementation and 2.6 percent for the alternate projection based implementation. Moreover, the main implementation overcomes the projection based one on 38 simulations (and the mean performance increase is 0.6 percent).

According to results reported in [15], the functional MLP approach outperforms both classical methods (such as CART) and functional ones. The best method studied in [15] achieves a mean classification error rate of 0.072 (with a standard deviation of 0.012). We can therefore conclude that our functional MLP is among the best methods for this dataset and that it overcomes both traditional methods and a classical neural approach. Moreover, as explained in the following section, the obtained functional model is very parsimonious which gives it robustness and efficiency.

6.2.2 Parameter numbers

For all methods, we select the best number of hidden neurons among 2, 3 and 4 hidden neurons. For the functional approaches, weight functions were represented using 5, 7, 10, 15 or 20 B-splines (those numbers have been chosen to keep the architectures as simple as possible). The chosen architecture depends on the simulation, but in general, small architectures are preferred, as summarized by the following tables. Table 2 gives the number of time each B-splines basis has been chosen and table 3 gives the number of time each number of hidden neurons has been chosen.

Number of B-splines	5	7	10	15	20
FMLP	18	17	8	3	4
FpMLP	38	8	4	0	0

Table 2: Number of simulations that select the given number of B-splines

Number of hidden neurons	2	3	4
MLP	10	40	0
FMLP	9	29	12
FpMLP	10	28	12

Table 3: Number of simulations that select the given number of hidden neurons

For our main functional approach, the total number of numerical parameters used varies between 23 and 103, with a mean of 44 (the median is 39 and only 10 simulations needed more than 51 parameters). For the projection based implementation, the total number of numerical parameters used varies between 23 and 63, with a mean of 36 (the median is 33 and only one simulation out of 50 uses more than 51 parameters). The projection based approach uses therefore even less parameters than our main functional approach, but with a slight decrease in the performances.

For the naive approach, cross-validation selects 3 hidden neurons for 10 simulations and 4 hidden neurons for the other 40 simulations. Those values correspond respectively to 321 and 427 numerical parameters (the mean is 406). The naive approach uses therefore far more parameters than functional methods and gives worse results.

The best method studied in [15] is a non-parametric functional method in which functions are first projected on an optimal basis constructed thanks to multivariate partial least squares regression. Optimal results are obtained thanks to a projection on three basis functions (this number is selected thanks to k -fold cross-validation). As the method is kernel based, we have to store all the functions of the training sample. That is, we need to keep a vector of \mathbb{R}^{101} for each basis function (303 numerical parameters) as well as the coordinate of each training function of this basis (3 parameters for each function). We have therefore a total of 1653 numerical parameters.

6.2.3 Pre-smoothing

A possible explanation for the poor performances of the standard MLP is that Breiman waves are very noisy. One side effect of using function representation, either for the functional weights or for the data themselves, is to smooth the waves. It is therefore quite natural to investigate the effect of applying a spline smoothing method on the waves before submitting them to a standard MLP.

In order to implement a fair comparison, we have used the following method: we calculate coordinates of training and test waves on each B-spline basis considered in the previous

series of experiments. These coordinates are used to reconstruct smooth versions of the waves that are sampled exactly as the original waves (101 points regularly spaced in $[1, 21]$). The obtained \mathbb{R}^{101} vectors are then submitted to a classical one hidden layer perceptron. The number of B-splines used for the smoothing phase, the number of hidden neurons and the weight decay are then selected by k -fold cross validation (with $k = 5$).

The test set performances are much better than with the basic MLP approach. Indeed, the mean error rate is now 0.073 (with a standard deviation of 0.012), which is comparable to the non-parametric approach of [15] and to the projection based implementation of the functional MLP. Nevertheless, a direct comparison shows that in fact our main implementation performs better than the smoothing approach for 42 simulations on 50. The projection based implementation obtains better results for 27 simulations. The basic MLP approach obtains better results than the smoothing approach for 1 simulation out of 50.

It seems therefore that smoothing plays an important role in obtaining good performances, but also that it does not help in reducing the number of parameters. Indeed, the mean number of parameters used by the smoothing approach is 406. With only 44 parameters, our main implementation obtains slightly better results.

6.2.4 Comments

Table 4 summarizes the result obtained on the Breiman waves. It is clear that the functional MLP approach gives very satisfactory results on those data. The obtained classification rate is slightly better than the best results reported in [15], which means that the MLP approach performs better than both traditional approaches and functional approaches. Moreover, the functional MLP approach also overcomes a naive MLP modeling of the raw multivariate data, as well as a more complex method in which a spline smoothing is performed on the raw data before submitting them to a classical MLP. Finally, the obtained model is very parsimonious: the MLP classifier will be faster than the kernel based one (after training).

Model	Parameters	Error rate	Training time
FMLP	44	0.065	4.5
FpMLP	36	0.072	3.8
Non parametric	1653	0.072	0
MLP with smoothing	406	0.073	5.9
MLP	406	0.098	1

Table 4: Results summary

Table 4 shows also the relative cost of the studied methods in terms of training time: the total training time of the classical MLP applied on raw data has been chosen as the reference training time (the values include the cross validation phase). As the non parametric approach of [15] involves almost no training phase (except for the selection of the kernel width), it has been considered as almost instantaneous compared to MLP training. Most of the cost comes from the model selection phase. Indeed, for the basic MLP, we just have to select

the weight decay parameter and the number of hidden neurons. On the contrary, all other methods involve the selection of the representation basis (here the number of B-splines). An interesting point is that the functional approaches are faster to train than the smoothing approach, give better results and produce very parsimonious summary of the data.

Compared to a classical MLP, the functional approach implies to use around 4.5 times more processing power in the training phase. Fortunately, the training is done only once and allows to produce a very small footprint solution than can be implemented on a small device such as a cell phone or a PDA, and with recognition performances that are significantly better than those of the classical MLP.

6.3 Spectrometric data

6.3.1 Raw data

Our next example is a real world classification problem of spectrometric data from food industry. Each observation is the near infrared absorbance spectrum of a meat sample (finely chopped), recorded on a Tecator Infratec Food and Feed Analyser. More precisely, an observation consists in a 100 channel spectrum of absorbances in the wavelength range 850–1050 nm. The goal is to classify meat samples into high fat samples and low fat samples. The first class consists in meat samples with less than 20% of fat, whereas the second class contains all other meat samples. We have a total of 215 spectra. Data are not organized into a training sample and a test sample, therefore, we follow exactly the evaluation method described in [15]: we select randomly 160 training spectra and 55 test spectra. We repeat this operation 50 times and give the average classification error rate.

We have compared the three approaches described in the introduction. The preprocessing experimented in section 6.2.3 was not considered here because absorbance spectra are very smooth and a B-spline basis projection as no noticeable smoothing effect on those functions. Table 5 gives statistical summaries of the classification error rate obtained by those neural methods. In this situation, only the alternate implementation of the functional approach

Method	First quartile	Mean	Median	Third quartile
MLP	0	0.019	0.018	0.036
FMLP	0.018	0.028	0.036	0.036
FpMLP	0	0.018	0.018	0.036

Table 5: Error rate for Spectrometric curves

gives satisfactory results. Indeed, the naive MLP approach gives better results than our main functional implementation. As in the previous section, generated data sets are identical for each method and a direct comparison between obtained results is possible. The naive method performs better than the FMLP method on 21 data sets (identical performances are obtained on 27 simulations).

But the FpMLP method still performs better than the naive approach. The average performance improvement is only 0.001, but FpMLP performs better than MLP on 30 simulations (identical performances are obtained on 13 simulations). We can therefore conclude that the best functional approach gives slightly better performances than the MLP approach. Moreover, the best method reported in [15] obtains a median classification rate of approximately 0.022, which shows again that neural methods perform very well. Additionally, the best method reported in [15] is as in previous section a mixed method that uses a functional non parametric model on functions projected on an optimal basis generated thanks to non functional multivariate partial least squares regression. [15] reports that a pure functional approach (in which functional principal component analysis is used to design an optimal projection) gives very bad results (the mean error rate is 0.2). On the contrary, our methods are pure functional methods and still give the best results.

Moreover, functional methods use a small number of numerical parameters. For all methods, we select the best number of hidden neurons among 2, 3 and 4 hidden neurons. For the functional approaches, weight functions were represented using 15 or 20 B-splines. In general, methods choose a small number of neurons, as shown in tables 6 and 7.

Number of B-splines	15	20
FMLP	12	38
FpMLP	24	26

Table 6: Number of simulations that select the given number of B-splines

Number of hidden neurons	2	3	4
MLP	24	16	10
FMLP	32	11	7
FpMLP	18	17	15

Table 7: Number of simulations that select the given number of hidden neurons

The classical MLP approach uses between 213 and 423 parameters, with a mean of 288 parameters. The main functional approach uses between 43 and 103 parameters (the mean is 62), whereas the projection based approach has the same range of parameter numbers with a higher mean (69). The best method reported in [15] uses 1300 parameters (almost 19 times more than our best method) with slightly worse performances.

6.3.2 Second order derivatives

[14] and [15] point out that the second derivative of the spectrum is in general more informative than the spectrum itself. The non parametric approach proposed in [15] has been used with a second derivative based semi-metric and achieved better results than the optimal projection based method. Indeed, the median error rate of a pure functional approach is now slightly less than 0.022. This method turns out to be the best overall method.

We have therefore applied our functional MLP approaches to the second derivative of the spectrum. As in [15], we evaluate the spectrum thanks to a B-spline representation. The second derivative of the B-spline is calculated exactly and sampled uniformly on $[850, 1050]$ as the original data. We obtain therefore new functional data that we model as normal functional data (that is we forget the preprocessing phase).

Table 8 gives statistical summaries of the classification error rate obtained by the neural methods applied to the second order derivatives.

Method	First quartile	Mean	Median	Third quartile
MLP	0	0.013	0.018	0.018
FMLP	0	0.007	0	0.018
FpMLP	0	0.014	0.009	0.018

Table 8: Error rate for second order derivatives of the Spectrometric curves

We obtain very satisfactory results as all neural methods perform better than results reported in [15]. Moreover, the best results are obtained by our main functional MLP implementation. A direct comparison between results obtained for each simulation shows that FMLP overcomes MLP on 15 simulations (identical performances are obtained on 31 simulations). The FMLP also overcomes FpMLP on 19 simulations (identical performances are obtained on 20 simulations). In fact FMLP provides perfect classification of the test set for 34 simulations, whereas this number drops to 25 for FpMLP and to 24 for MLP.

We do not report completely architecture selection results as they are very similar to those obtained on the raw functional data. MLP uses a mean number of 391 parameters, FMLP 82 and FpMLP 76.

6.3.3 Comments

As in the Breiman wave experiments, an appropriate functional MLP model allows to obtain very good recognition rate that cannot be reached by a classical MLP. Moreover, the optimal functional MLP uses a small number of parameters, which eases its real world implementation. We have not reported here training times as they are comparable to values reported in table 4: the price to pay for higher recognition rate and lower parameter number is a higher training time than the one needed for a classical MLP, mainly because of the additional parameter (the number of B-splines) that has to be chosen by cross validation.

6.4 Conclusions

In both experiments (on simulated data and on real world data), functional multi-layer perceptrons perform in a very satisfactory way. They are at least as good as functional and traditional methods presented in [15]. Moreover, they also overcome a naive MLP modeling of the raw multivariate data. A way to obtain correct results with a classical MLP

is to perform a kind of functional preprocessing: a spline smoothing for noisy data such as the Breiman wave or a derivative calculation for smooth data such as the absorbance spectra. But even those mixed approaches do not perform as well as the functional MLP. Another important practical property is the small number of numerical parameters used by the functional neural methods: this allows an easier implementation on devices with limited resources such as PDA, cell phones and more generally embedded devices.

Of course, additional experiments on real world data are needed to fully understand advantages and shortcomings of the proposed functional MLP. While the model has been compared to traditional classification methods thanks to experiments conducted in [15], additional comparisons, especially to recent methods such as support vector machines (see e.g. [12]) or boosted classification trees (see e.g. [18]), are also needed.

An interesting open research topic is to develop automatic tuning of weight function representation. We have used here a brute force k -fold cross-validation method but [15] shows that automatic design of projection basis can improve performances. Moreover, this might reduce the training time of functional MLP which remains the only negative part of the proposed approach compared to classical MLP (when the latter is used without functional preprocessing).

7 Conclusion

In this paper, we have introduced Functional Multi-Layer Perceptrons (FMLP), a simple extension of MLP to functional data. The proposed model is very interesting on a theoretical point of view because it shares with its numerical counterpart useful properties.

We have indeed shown that FMLP are universal approximators, that is they can approximate continuous mappings from a compact subset of a functional space to \mathbb{R} with arbitrary precision. For a given function to approximate to a given accuracy, the approximating FMLP uses a finite number of numerical parameters.

Moreover, we have shown that parameter estimation for FMLP is consistent: optimal parameters estimated thanks to a finite number of functions known at a finite number of measurement points converge to the set of true optimal parameters when the size of the data increases.

We have also shown on simulated and real world data that the FMLP performs in a very satisfactory way. Performances are in general better than those obtained by non functional methods (including neural methods) and at least as good as other functional methods. Moreover, the functional approach gives much more parsimonious representation of studied data, a property that enhance the robustness of the obtained models and allows also an easier implementation on devices with limited processing power. We believe therefore that Functional Multi-Layer Perceptrons are a valuable tool for data analysis when a functional representation of input variables is possible.

Acknowledgement

The authors thank participants to the working group STAPH (<http://www.lsp.ups-tlse.fr/Fp/Ferraty/staph.html>) on Functional Statistics at the University Paul Sabatier of Toulouse, for very interesting and stimulating discussions. The authors thank especially Frédéric Ferraty and Philippe Vieu for sharing data and simulation results. The authors thank also anonymous referees for their valuable suggestions that help improving this paper.

Proofs

Proof. (corollary 1) If $1 < p < \infty$, we know that $L^q(\mu)$ (with $q < \infty$) can be identified with $(L^p(\mu))^*$ (see for instance [28]). More precisely, for each $l \in (L^p(\mu))^*$ there is an unique function $f \in L^q(\mu)$ so that $l(g) = \int fg d\mu$. By hypothesis, V is dense in $L^q(\mu)$. This obviously implies that A_V is dense in $(L^p(\mu))^*$ for the weak $*$ topology. We can therefore apply corollary 5.1.3 of [31] (note that corollary 5.1.3 is given for the outside density case, but the author states explicitly that a similar inside corollary is valid).

If $p = \infty$, we cannot apply directly corollary 5.1.3 from [31] as the dual of $L^\infty(\mu)$ is not $L^1(\mu)$. Let us nevertheless consider A the set of affine functions on $L^\infty(\mu)$ defined by $l(f) = \alpha + \int fg d\mu$, where α is an arbitrary real number and g is an arbitrary function from $V \subset L^1(\mu)$. A is obviously a vectorial space which contains constant functions of $C(K, \mathbb{R})$. Let us now show that A separates points in K . Let u and v be two distinct functions of K . The function $f = u - v$ is a non zero function belonging to $L^\infty(\mu)$. We can assume that the measurable set $H = \{x \in \mathbb{R}^n \mid f(x) > 0\}$ has non zero finite measure (if it is not the case, replace f by $-f$). Then, obviously $\int f \chi_H d\mu > 0$, that is $\int u \chi_H d\mu \neq \int v \chi_H d\mu$. As μ is finite, χ_H belongs to $L^1(\mu)$. As V is dense in $L^1(\mu)$, there is a sequence h_k of functions in V that converges to χ_H . We have obviously

$$\left| \int f(h_k - \chi_H) d\mu \right| \leq |f|_\infty \left| \int h_k - \chi_H d\mu \right|.$$

Therefore, there is an index k such that $\int f h_k d\mu > 0$, that is there is a function $h_k \in V$ such that $\int u h_k d\mu \neq \int v h_k d\mu$. Therefore, A separates points in K . The conclusion is then obtained by applying theorem 5.1 of [31]. \square

Proof. (corollary 2) As μ is a finite Borel measure on \mathbb{R}^n , it is regular ([28], theorem 2.18), and we can apply Lusin theorem ([28], theorem and corollary 2.23). We know therefore that for any function f in $L^\infty(\mu)$, there is a sequence of compactly supported continuous functions g_k that converges punctually to f and such that $|g_k|_\infty \leq |f|_\infty$. A simple application of Lebesgue dominated convergence theorem shows that for any function h in $L^1(\mu)$, $\int g_k h d\mu \rightarrow_{k \rightarrow \infty} \int f h d\mu$. Then, as μ is compactly supported, there is a compact K such that $\int g_k h d\mu = \int_K g_k h d\mu$. Then, thanks to hypothesis, each g_k can be approximated by a function ϕ_k in V such that $\sup_{x \in K} |g_k(x) - \phi_k(x)| < \frac{1}{k}$. In this case $|\int_K g_k h d\mu - \int_K \phi_k h d\mu| < \frac{1}{k} \|h\|_1$. As μ is compactly supported, this allows to conclude

that $\int \phi_k h d\mu \xrightarrow{k \rightarrow \infty} \int f h d\mu$. Therefore, the set of linear forms A_V is dense for the weak * topology in $(L^1(\mu))^*$, provided that μ is finite and compactly supported. The conclusion is then obtained by applying corollary 5.1.3 from [31]. \square

Proof. (theorem 1) The proof is quite technical and can be cut into several parts:

1. We need first a quite general Uniform Strong Law of Large Numbers (USLLN) which will be obtained thanks to a general result of [2].
2. Then we show that integral approximations used in the definition of $\lambda_n^m(w)$ have a kind of uniform convergence property.
3. Using both results, we show that $\lambda_n^m(w)$ converges almost surely uniformly to $\lambda(w)$.
4. The conclusion is obtained thanks to a simple lemma on approximation of the minimizers of a function.

part 1

A very general Uniform Strong Law of Large Numbers (USLLN) is given in [2]. It is based on complex assumptions, so we propose to simplify it into the following corollary:

Corollary 3. *Let X be an arbitrary metric space considered with its Borel sigma algebra. Let (Ω, \mathcal{A}, P) be a probability space on which is defined a sequence of independent identically distributed random elements, Z_t with values in X . Let W be a compact metric space. Let l be a function from $W \times X$ to \mathbb{R} . We assume that the following conditions hold:*

1. *For each $w \in W$, $l(w, \cdot)$ is a measurable function from X to \mathbb{R} .*
2. *For each $x \in X$, $l(\cdot, x)$ is a continuous function from W to \mathbb{R} .*
3. *there is a positive measurable function d (from X to \mathbb{R}) such that for all $x \in X$ and for all $w \in W$, $|l(w, x)| \leq d(x)$.*
4. *$E(d(Z_t)) < \infty$.*

Then we have:

$$\sup_{w \in W} \left| \frac{1}{n} \sum_{i=1}^n l(w, Z_i) - E(l(w, Z_t)) \right| \xrightarrow[n \rightarrow \infty]{a.s.} 0.$$

In order to prove this corollary, we need first a simple lemma:

Lemma 1. *Let l be a function from $W \times X$ to \mathbb{R} , where W is a separable metric space and X is a metric space (considered with its Borel sigma algebra). If l is continuous on W for each fixed $x \in X$ and measurable on X for each fixed $w \in W$, then the function $f(x) = \sup_{w \in W} l(w, x)$ is measurable.*

Proof. (lemma 1) As W is separable, there is a denombrable set $W' = \{w_i \mid i \in \mathbb{N}^*\}$ dense in W . Let us show that $f(x) = \sup_{w \in W'} l(w, x)$. Let us consider a fixed $x \in X$. Let ϵ be an arbitrary positive real number. By definition of f , there is $w \in W$ such that $l(w, x) \geq f(x) - \frac{\epsilon}{2}$. As $l(\cdot, x)$ is continuous in w , there is η such that $|w' - w| < \eta$ implies $|l(w', x) - l(w, x)| < \frac{\epsilon}{2}$, which implies $l(w', x) \geq f(x) - \epsilon$. As W' is dense in W , there is $w' \in W'$ such that $|w' - w| < \eta$. This implies $f(x) \geq \sup_{w \in W'} l(w, x) \geq f(x) - \epsilon$. As this is true for each ϵ , we have obviously $f(x) = \sup_{w \in W'} l(w, x)$. Therefore, $f(x) = \sup_{i \in \mathbb{N}} l(w_i, x)$. As each function $l(w_i, x)$ is measurable, the sup is also measurable. \square

We can now proceed to the proof of the corollary:

Proof. (corollary 3) We obtain corollary 3 as a consequence of Andrews' theorem ([2]). We have to check three assumptions:

1. Assumption A 1 is fulfilled as W is compact (W corresponds to Θ in Andrews' paper)
2. Assumption A 2 breaks into two sub-assumptions:
 - (a) Assumption A 2 (a) can be translated with our notation into the following assumption: for all w_0 (and all i), $l(w_0, Z_i)$, $\sup_{w \in W(w_0, \eta)} l(w, Z_i)$ and $\inf_{w \in W(w_0, \eta)} l(w, Z_i)$ are random variables (where $W(w_0, \eta) = B(w_0, \epsilon) \cap W$, and $B(w_0, \epsilon)$ is the closed ball centered on w_0 with radius ϵ).

$l(w_0, Z_i)$ is a random variable thanks to assumption 1 of corollary 3. Thanks to assumptions 1 and 2 of corollary 3 and due to the fact that a compact set is separable, lemma 1 can be applied to l and to $W(w_0, \eta)$, and allows to conclude that $\sup_{w \in W(w_0, \eta)} l(w, Z_i)$ is a random variable. The case of $\inf_{w \in W(w_0, \eta)} l(w, Z_i)$ is handled thanks to the same lemma applied to $-l$.

Assumption A 2 (a) is therefore fulfilled.

- (b) Assumption A 2 (b) translates in our case into the assumption that $\sup_{w \in W(w_0, \eta)} l(w, Z_i)$ and $\inf_{w \in W(w_0, \eta)} l(w, Z_i)$ satisfy a point-wise strong law of large numbers, that is for any fixed w_0 :

$$\lim_{n \rightarrow \infty} \frac{1}{n} \sum_{i=1}^n \sup_{w \in W(w_0, \eta)} l(w, Z_i) = E \left(\sup_{w \in W(w_0, \eta)} l(w, Z_i) \right) \text{ P a.s.}$$

As shown in the previous point, both $\left(\sup_{w \in W(w_0, \eta)} l(w, Z_i) \right)_{i \in \mathbb{N}^*}$ and $\left(\inf_{w \in W(w_0, \eta)} l(w, Z_i) \right)_{i \in \mathbb{N}^*}$ are sequences of independent identically distributed random variables. Moreover, thanks to assumptions 3 and 4 of corollary 3, they are integrable and therefore the strong law of large numbers applies: assumption A 2 (b) is therefore fulfilled.

3. Assumption A3 translates in our case into the following assumption:

$$\lim_{\eta \rightarrow 0} \sup_{n \geq 1} \left| \frac{1}{n} \sum_{i=1}^n \left(E \left(\sup_{w \in W(w_0, \eta)} l(w, Z_i) \right) - E(l(w, Z_i)) \right) \right| = 0.$$

A similar equation has to be fulfilled by $E(\inf_{w \in W(w_0, \eta)} l(w, Z_i))$.

As l is continuous with respect to w for a fixed x , we have the following point-wise convergence:

$$\lim_{\eta \rightarrow 0} \sup_{w \in W(w_0, \eta)} l(w, \cdot) = l(w_0, \cdot).$$

Thanks to assumptions 3 and 4 of corollary 3, we can apply Lebesgue dominated convergence which implies:

$$\lim_{\eta \rightarrow 0} E \left(\sup_{w \in W(w_0, \eta)} l(w, Z_i) \right) = E(l(w_0, Z_i)).$$

Finally, as Z_i are identically distributed, assumption A3 can be simplified into:

$$\lim_{\eta \rightarrow 0} \left| E \left(\sup_{w \in W(w_0, \eta)} l(w, Z_1) \right) - E(l(w, Z_1)) \right| = 0,$$

which is exactly what we have just proven. The case of $E(\inf_{w \in W(w_0, \eta)} l(w, Z_i))$ can be obtained exactly the same way.

Assumption A3 is therefore fulfilled.

As the assumptions are fulfilled, we can apply Andrews' theorem which gives exactly the conclusion of corollary 3. \square

part 2

Let us define:

$$M_l^i(g, w_l)(\omega)_m = \frac{1}{m} \sum_{j=1}^m F_l(w_l, X_j^i(\omega)) (g(X_j^i(\omega)) + \mathcal{E}_j^i(\omega)),$$

which can be simplified into $M_l^i(g, w_l)_m$ when ω is obvious, and

$$M_l(g, w_l) = \int F_l(w_l, x) g(x) d\mu(x)$$

We prove now the following lemma:

Lemma 2. *Let us define*

$$\Omega_l = \left\{ \omega \in \Omega \mid \lim_{m \rightarrow \infty} \frac{1}{m} \sum_{j=1}^m d_l(X_j^i) = \int d_l d\mu \right\}$$

and

$$B_l^i = \left\{ \omega \in \Omega \mid \forall g \in C(Z, \mathbb{R}), \lim_{m \rightarrow \infty} \sup_{w_l \in W_l} |M_l^i(g, w_l)(\omega)_m - M_l(g, w_l)| = 0 \right\}.$$

Under hypothesis H_a , H_c and H_e , $B_l^i \cap \Omega_l$ is measurable and $P(B_l^i \cap \Omega_l) = 1$.

Proof. (lemma 2) The proof is based on the separability of $C(Z, \mathbb{R})$ and on corollary 3. Let us first note that $P(\Omega_l) = 1$ thanks to hypothesis H_c (2-c) and the strong law of large numbers. Let us first show that the set

$$B_l^i(g) = \left\{ \omega \in \Omega \mid \lim_{m \rightarrow \infty} \sup_{w_l \in W_l} |M_l^i(g, w_l)(\omega)_m - M_l(g, w_l)| = 0 \right\}$$

is such that $P(B_l^i(g)) = 1$ for any $g \in C(Z, \mathbb{R})$.

This can be obtained by applying corollary 3 to the function ψ from $W_l \times (Z \times \mathbb{R})$ to \mathbb{R} defined as follows

$$\psi(w_l, (x, e)) = F_l(w_l, x)(g(x) + e),$$

and to the sequence of random elements $(X_j^i, \mathcal{E}_j^i)_{j \in \mathbb{N}}$. Corollary 3 applies because:

- W_l is compact (hypothesis H_c (1))
- hypothesis H_c (2-b) implies that ψ is measurable with respect to its second variable
- hypothesis H_c (2-a) and $g \in C(Z, \mathbb{R})$ implies that ψ is continuous with respect to its first variable
- hypothesis H_c (2-c) implies $\forall x \in Z, w_l \in W_l$ and $\forall e \in \mathbb{R}, |\psi(w_l, (x, e))| \leq d_l(x)(|g(x)| + |e|)$
- as g is continuous on the compact set Z , $g \in L^q(\mu)$ and therefore $d_l(x)|g(x)| \in L^1(\mu)$ (according to hypothesis H_c (2-c))
- as $E(|\mathcal{E}_j^i|^q) < \infty$ (hypothesis H_e (5)), $E(d_l(X_j^i) |\mathcal{E}_j^i|) < \infty$
- $(X_j^i, \mathcal{E}_j^i)_{j \in \mathbb{N}}$ is i.i.d. (hypothesis H_e)

Therefore, we have:

$$\sup_{w_l \in W_l} |M_l^i(g, w_l)_m - E(F_l(w_l, X_1^i)(g(X_1^i) + \mathcal{E}_1^i))| \xrightarrow{a.s.} 0.$$

By definition, $E(F_l(w_l, X_1^i)g(X_1^i)) = M_l(g, w_l)$ and by independence and hypothesis H_e (5):

$$E(F_l(w_l, X_1^i)\mathcal{E}_1^i) = E(F_l(w_l, X_1^i)) E(\mathcal{E}_1^i) = 0.$$

Therefore:

$$\sup_{w_l \in W_l} |M_l^i(g, w_l)_m - M_l(g, w_l)| \xrightarrow{a.s.}_{m \rightarrow \infty} 0,$$

which means that $P(B_l^i(g)) = 1$.

As $C(Z, \mathbb{R})$ is separable, there is a sequence $(h_t)_{t \in \mathbb{N}}$ dense in $C(Z, \mathbb{R})$ (for the uniform norm). Let us denote $A_l^i = \Omega_l \cap \bigcap_{t \in \mathbb{N}} B_l^i(h_t)$. A_l^i is measurable and $P(A_l^i) = 1$. Obviously, $B_l^i \cap \Omega_l \subset A_l^i$. Let us now show that $B_l^i \cap \Omega_l = A_l^i$.

Let $\omega \in A_l^i$. As $\omega \in \Omega_l$, $\frac{1}{m} \sum_{j=1}^m d_l(X_j^i(\omega))$ is a convergent sequence and is therefore bounded, so there is $\gamma_l^i(\omega) > 1$ such that for all m , $\left| \frac{1}{m} \sum_{j=1}^m d_l(X_j^i(\omega)) \right| < \gamma_l^i(\omega)$. Moreover, we can choose $\gamma_l^i(\omega)$ such that $\gamma_l^i(\omega) > E(d_l(X_1^i))$.

Let $g \in C(Z, \mathbb{R})$. For any $\epsilon > 0$, there is $t \in \mathbb{N}$ such that $\rho_Z(g, h_t) < \frac{\epsilon}{3\gamma_l^i(\omega)}$. This obviously implies for all $w_l \in W_l$ and for all m both $|M_l^i(w_l, g)(\omega)_m - M_l^i(w_l, h_t)(\omega)_m| < \frac{\epsilon}{3}$ and $|M_l(w_l, g) - M_l(w_l, h_t)| < \frac{\epsilon}{3}$. As $\omega \in A_l^i$, $M_l^i(w_l, h_t)(\omega)_m$ converges to $M_l(w_l, h_t)$ uniformly on W_l . Therefore there is M such that $m > M$ implies $\sup_{w_l \in W_l} |M_l^i(w_l, h_t)(\omega)_m - M_l(w_l, h_t)| < \frac{\epsilon}{3}$. Then $m > M$ implies $\sup_{w_l \in W_l} |M_l^i(w_l, g)(\omega)_m - M_l(w_l, g)| < \epsilon$. As this is true for any ϵ , we conclude that $M_l^i(w_l, g)(\omega)_m$ converges uniformly on W_l to $M_l(w_l, g)$, and therefore that $\omega \in B_l^i(g) \cap \Omega_l$. As this is true for all $g, \omega \in B_l^i \cap \Omega_l$. Therefore, $B_l^i \cap \Omega_l = A_l^i$, which gives the conclusion of the lemma. \square

part 3

Let us now apply corollary 3 to $\widehat{\lambda}_n(w)$, more precisely to the function from $W \times (C(Z, \mathbb{R}) \times \mathbb{R}^o)$ to \mathbb{R} define by:

$$k(w, g, t) = c \left(t, U \left(w_0, \int F_1(w_1, x) g(x) d\mu(x), \dots, \int F_k(w_k, x) g(x) d\mu(x) \right) \right).$$

This is possible according to the following reasons:

- W is compact
- k is continuous on w for each (g, t) , according to hypotheses H_c and because, as a continuous function defined on a compact set, g belongs to $L^q(\mu)$. Indeed, $w_l \mapsto \int F_l(w_l, x) g(x) d\mu(x)$ is continuous for each g : as F_l is continuous on w for each x , the function $F_l(w'_l, \cdot) g(\cdot)$ converges punctually to $F_l(w_l, \cdot) g(\cdot)$ when w'_l converges to w_l . Moreover, $|F_l(w, \cdot) g(\cdot)|$ is dominated on W_l by $d_l(\cdot)|g(\cdot)|$, which is integrable (by hypothesis). Thanks to dominated convergence theorem, this obviously implies the continuity of $w_l \mapsto \int F_l(w_l, x) g(x) d\mu(x)$.

- k is measurable with respect (g, t) for each w . This is a direct consequence of hypotheses H_c and of the fact that $g \mapsto \int F_l(w_l, x) g(x) d\mu(x)$ is continuous for each w_l
- hypothesis H_d implies that $k(w, g, t) \leq c_{max}(t)$ for all w, g and t , with $E(c_{max}(T_1)) < \infty$
- $(G^i, T^i)_{i \in \mathbb{N}}$ is i.i.d.

According to corollary 3, we therefore have

$$\sup_{w \in W} \left| \frac{1}{n} \sum_{i=1}^n k(w, G^i, T^i) - E(k(w, G^1, T^1)) \right| \xrightarrow{a.s.} 0,$$

that is

$$\sup_{w \in W} \left| \widehat{\lambda}_n(w) - \lambda(w) \right| \xrightarrow{a.s.} 0. \quad (23)$$

Let us call C the set of probability 1 for which this uniform convergence occurs. Let us now consider $D = C \cap \bigcap_{i \in \mathbb{N}} \bigcap_{l \in \mathbb{N}} (B_l^i \cap \Omega_l)$. According to lemma 2, $P(D) = 1$. Let ω be an arbitrary element of D and denote for simplicity $g^i = G^i(\omega)$ and $t^i = T^i(\omega)$. Let $\epsilon > 0$ be an arbitrary real number. According to equation 23, there is N such that for each $n \geq N$,

$$\sup_{w \in W} \left| \frac{1}{n} \sum_{i=1}^n k(w, g^i, t^i) - \lambda(w) \right| < \frac{\epsilon}{2}. \quad (24)$$

We handle here the case where c is not a distance on \mathbb{R}^o but simply a continuous positive function. As U is bounded and uniformly continuous, the function $l(t, w_0, u) = c(t, U(w_0, u))$ from $\mathbb{R}^o \times W_0 \times \mathbb{R}^k$ is uniformly continuous with respect to (w_0, u) . That is, for each t^i , there is $\eta_i > 0$ such that for each w_0 and $(u, u') \in \mathbb{R}^k \times \mathbb{R}^k$, $\|u - u'\| < \eta \Rightarrow \|l(t^i, w_0, u) - l(t^i, w_0, u')\| < \frac{\epsilon}{2}$. As $\omega \in \bigcap_{i \in \mathbb{N}} \bigcap_{l \in \mathbb{N}} (B_l^i \cap \Omega_l)$, for each i , there is S^i such that $m^i \geq S^i$ implies for all l

$$\sup_{w_l \in W_l} |M_l^i(w_l, g^i)(\omega)_{m^i} - M_l(w_l, g^i)| < \eta_i.$$

Let us call $S_n = \sup_{i \leq n} S^i$. Then for $m \geq S_n$, for all w and for all $i \leq n$

$$\begin{aligned} & \left\| c(t^i, U(w_0, M_1^i(w_1, g^i)(\omega)_m, \dots, M_k^i(w_k, g^i)(\omega)_m)) \right. \\ & \quad \left. - c(t^i, U(w_0, M_1(w_1, g^i), \dots, M_k(w_k, g^i))) \right\| < \frac{\epsilon}{2}, \end{aligned}$$

that is for all $w \in W$

$$\left| \frac{1}{n} \sum_{i=1}^n c(t^i, U(w_0, M_1^i(w_1, g^i)(\omega)_m, \dots, M_k^i(w_k, g^i)(\omega)_m)) - \frac{1}{n} \sum_{i=1}^n k(w, g^i, t^i) \right| < \frac{\epsilon}{2}.$$

Combined with equation 24, this gives that for $n \geq N$ and $m \geq M(n)$:

$$\sup_{w \in W} |\lambda_n^m(w) - \lambda(w)| < \epsilon.$$

Therefore for almost all ω (i.e., for $\omega \in D$), we have:

$$\lim_{n \rightarrow \infty} \lim_{m \rightarrow \infty} \sup_{w \in W} |\lambda_n^m(w) - \lambda(w)| = 0. \quad (25)$$

part 4

The final conclusion of the theorem is obtained exactly as in [32]. We use the following lemma:

Lemma 3. *Let W be a compact set (considered with the metric d) and $(f_i^j)_{i \in \mathbb{N}, j \in \mathbb{N}}$ a sequence of sequences of real valued continuous functions that converges uniformly to a continuous function f , that is $\lim_{n \rightarrow \infty} \lim_{m \rightarrow \infty} \rho_W(f_n^m, f) = 0$. Let us call W^* the set of minimizers of f and let w_i^j be a minimizer of f_i^j . Then*

$$\lim_{n \rightarrow \infty} \lim_{m \rightarrow \infty} d(w_n^m, W^*) = 0.$$

Proof. (lemma 3) First of all, it is clear that we just have to prove that the set of accumulation points of $(w_i^j)_{i \in \mathbb{N}, j \in \mathbb{N}}$, Acc , is included into W^* . Indeed assume that both $Acc \subset W^*$ and that the conclusion of the theorem does not hold. This implies that there is an infinite subsequence of $(w_i^j)_{i \in \mathbb{N}, j \in \mathbb{N}}$ which distance to W^* remains above a fixed positive number. As W is compact this subsequence has at least one accumulation point which cannot belong to W^* . As this accumulation point is also an accumulation point of the full sequence, this contradicts our main hypothesis.

Let us now consider w^0 an accumulation point of the sequence. Strictly speaking, w^0 is the limit of a subsequence of the main sequence, but to simplify the proof, we assume that $w^0 = \lim_{n \rightarrow \infty} \lim_{m \rightarrow \infty} w_n^m$.

Let $\epsilon > 0$. f is uniformly continuous on W and therefore there is η such that $|w' - w| < \eta$ implies $|f(w) - f(w')| < \epsilon$. By uniform convergence, there is N such that for each $n > N$, there is M_n such that $m > M_n$ implies for all $w \in W$, $|f_n^m(w) - f(w)| < \epsilon$. Moreover, we can choose N and M_n such that $n > N$ and $m > M_n$ imply $|w_n^m - w^0| < \eta$. Therefore, $n > N$ and $m > M_n$ imply $|f_n^m(w_n^m) - f(w^0)| < 2\epsilon$.

As w_n^m is a minimizer of f_n^m , for all w , $f_n^m(w_n^m) - f_n^m(w) \leq 0$, which implies (by uniform convergence), $f_n^m(w_n^m) - f(w) \leq \epsilon$. Therefore, $f(w^0) - f(w) \leq 3\epsilon$. As this is true for all ϵ , we conclude that $f(w^0) - f(w) \leq 0$ for all w and therefore that $w^0 \in W^*$. Therefore $Acc \subset W^*$. \square

The conclusion of the theorem is obtained by applying lemma 3 to all $\omega \in D$. For such a ω , the uniform convergence of λ_n^m to λ translates into the convergence of any minimizer of λ_n^m to the set of minimizers of λ . \square

References

- [1] Christophe Abraham, Pierre-André Cornillon, Eric Matzner-Lober, and Nicolas Molinari. Unsupervised curve clustering using b-splines. *Scandinavian Journal of Statistics*, 30(3):581–595, September 2003.
- [2] Donald W. K. Andrews. Consistency in nonlinear econometric models: A generic uniform law of large numbers. *Econometrica*, 55(6):1465–1471, November 1987.
- [3] P. Besse, H. Cardot, R. Faivre, and M. Goulard. Statistical modelling of functional data. *Applied Stochastic Models in Business and Industry*, 2004. to be published.
- [4] Philippe Besse and Hervé Cardot. *Analyse des données (édité par Gérard Govaert)*, chapter 6 : Modélisation statistique de données fonctionnelles, pages 167–198. Hermès/Lavoisier, 2003.
- [5] Philippe Besse, Hervé Cardot, and Frédéric Ferraty. Simultaneous non-parametric regressions of unbalanced longitudinal data. *Computational Statistics and Data Analysis*, 24:255–270, 1997.
- [6] Philippe Besse, Hervé Cardot, and David Stephenson. Autoregressive forecasting of some functional climatic variations. *Scandinavian Journal of Statistics*, 4:673–688, 2000.
- [7] Leo Breiman, Jerome H. Friedman, Richard A. Olshen, and Charles J. Stone. *Classification and Regression Trees*. Wadsworth, 1984.
- [8] Hervé Cardot, Frédéric Ferraty, and Pascal Sarda. Functional linear model. *Statist. & Prob. Letters*, 45:11–22, 1999.
- [9] Hervé Cardot, Frédéric Ferraty, and Pascal Sarda. Spline estimators for the functional linear model. *Statistica Sinica*, 13:571–591, 2003.
- [10] Tianping Chen. A unified approach for neural network-like approximation of non-linear functional. *Neural Networks*, 11:981–983, 1998.
- [11] Tianping Chen and Hong Chen. Universal approximation to nonlinear operators by neural networks with arbitrary activation functions and its application to dynamical systems. *IEEE Transactions on Neural Networks*, 6(4):911–917, July 1995.
- [12] Nello Cristianini and John Shawe-Taylor. *An Introduction to Support Vector Machines*. Cambridge University Press, Cambridge, UK, 2000.
- [13] Frédéric Ferraty, Aldo Goia, and Philippe Vieu. Functional nonparametric model for time series: a fractal approach for dimension reduction. *TEST*, 11(2):317–344, December 2002.
- [14] Frédéric Ferraty and Philippe Vieu. The functional nonparametric model and application to spectrometric data. *Computational Statistics*, 17(4), 2002.

- [15] Frédéric Ferraty and Philippe Vieu. Curves discriminations: a nonparametric functional approach. *Computational Statistics and Data Analysis*, 44(1–2):161–173, 2003.
- [16] Louis Ferré and Anne-Françoise Yao. Functional sliced inverse regression analysis. *Statistics*, 37(6):475–488, November/December 2003.
- [17] T. Hastie and C. Mallows. A discussion of "a statistical view of some chemometrics regression tools" by i.e. frank and j.h. friedman. *Technometrics*, 35:140–143, 1993.
- [18] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer-Verlag, 2001.
- [19] Kurt Hornik. Approximation capabilities of multilayer feedforward networks. *Neural Networks*, 4(2):251–257, 1991.
- [20] Kurt Hornik. Some new results on neural network approximation. *Neural Networks*, 6(8):1069–1072, 1993.
- [21] Gareth M. James. Generalized linear models with functional predictor variables. *Journal of the Royal Statistical Society Series B*, (64):411–432, 2002.
- [22] Gareth M. James and Trevor J. Hastie. Functional linear discriminant analysis of irregularly sampled curves. *Journal of the Royal Statistical Society Series B*, 63:533–550, 2001.
- [23] Moshe Leshno, Vladimir Ya. Lin, Allan Pinkus, and Shimon Schocken. Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. *Neural Networks*, 6(6):861–867, 1993.
- [24] Ker-Chau Li. Sliced inverse regression for dimension reduction. *J. Amer. Statist. Assoc.*, 86:316–342, 1991.
- [25] B. D. Marx and P. H. Eilers. Generalized linear regression on sampled signals with penalized likelihood. In R. Hatzinger A. Forcina, G. M. Marchetti and G. Galmacci, editors, *Statistical Modelling. Proceedings of the 11th International workshop on Statistical Modelling*, Orvieto, 1996.
- [26] Jim Ramsay and Bernard Silverman. *Functional Data Analysis*. Springer Series in Statistics. Springer Verlag, June 1997.
- [27] Fabrice Rossi and Briec Conan-Guez. Functional multi-layer perceptron: a nonlinear tool for functional data analysis. Technical Report 0331, LISE/CEREMADE Université Paris IX, <http://www.ceremade.dauphine.fr/CMD/preprints03/0331.pdf>, September 2003.
- [28] Walter Rudin. *Real and complex Analysis*. Mc Graw Hill, 1974.

-
- [29] Irwin W. Sandberg. Notes on weighted norms and network approximation of functionals. *IEEE Transactions on Circuits and Systems-I: Fundamental Theory and Applications*, 43(7):600–601, July 1996.
- [30] Irwin W. Sandberg and Lilian Xu. Network approximation of input-output maps and functionals. *Circuits Systems Signal Processing*, 15(6):711–725, 1996.
- [31] Maxwell B. Stinchcombe. Neural network approximation of continuous functionals and continuous functions on compactifications. *Neural Networks*, 12(3):467–477, 1999.
- [32] Halbert White. Learning in Artificial Neural Networks: A Statistical Perspective. *Neural Computation*, 1(4):425–464, 1989.



Unité de recherche INRIA Rocquencourt
Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)

Unité de recherche INRIA Futurs : Parc Club Orsay Université - ZAC des Vignes
4, rue Jacques Monod - 91893 ORSAY Cedex (France)

Unité de recherche INRIA Lorraine : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)

Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)

Unité de recherche INRIA Rhône-Alpes : 655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier (France)

Unité de recherche INRIA Sophia Antipolis : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
<http://www.inria.fr>
ISSN 0249-6399