Theoretical Properties of Projection Based Multilayer Perceptrons with Functional Inputs[†]

Fabrice Rossi (fabrice.rossi@inria.fr)
Projet AxIS, INRIA, Domaine de Voluceau, Rocquencourt, B.P. 105, 78153 Le Chesnay Cedex, France

Brieuc Conan-Guez (brieuc.conan-guez@iut.univ-metz.fr)
LITA EA3097, Université de Metz, Ile du Saulcy, F-57045 Metz, France

January 31, 2006

Abstract. Many real world data are sampled functions. As shown by Functional Data Analysis (FDA) methods, spectra, time series, images, gesture recognition data, etc. can be processed more efficiently if their functional nature is taken into account during the data analysis process. This is done by extending standard data analysis methods so that they can apply to functional inputs. A general way to achieve this goal is to compute projections of the functional data onto a finite dimensional sub-space of the functional space. The coordinates of the data on a basis of this sub-space provide standard vector representations of the functions. The obtained vectors can be processed by any standard method.

In [43], this general approach has been used to define projection based Multilayer Perceptrons (MLPs) with functional inputs. We study in this paper important theoretical properties of the proposed model. We show in particular that MLPs with functional inputs are universal approximators: they can approximate to arbitrary accuracy any continuous mapping from a compact sub-space of a functional space to $\mathbb{R}$. Moreover, we provide a consistency result that shows that any mapping from a functional space to $\mathbb{R}$ can be learned thanks to examples by a projection based MLP: the generalization mean square error of the MLP decreases to the smallest possible mean square error on the data when the number of examples goes to infinity.

Keywords: Functional Data Analysis; Multilayer Perceptron; Universal Approximation; Consistency; Projection

1. Introduction

In many practical situations, input data are in fact sampled functions rather than standard high dimensional vectors. This is the case for instance in spectrometry: a discretized spectrum is obtained by measuring the transmittance or the reflectance of an object at different wavelengths. Modern spectrometers can produce very high resolution spectra, with a thousand of observations for each spectrum.

[†] Published in Neural Processing Letters (Volume 23, Number 1, February 2006, pages 55–70). The original publication is available at www.spingerlink.com. DOI: http://dx.doi.org/10.1007/s11063-005-3100-2

Another general example of sampled functions is given by time series. Indeed, a time series is a mapping from a time period to a observation range, for instance the hourly temperature at a weather station over one month. More complex examples can be found in meteorology, for instance rainfall maps, i.e., functions that map geographical coordinates and date to the daily rain level observed at the specified position and date.

Functional Data Analysis (FDA) [4, 37] is a general methodology targeted at data that are better described as functions than as vectors. The main idea is to take advantage of the functional nature of the data to design better data analysis methods than the ones constructed thanks to a vector model. For a comprehensive introduction to FDA methods we refer the reader to [36] in which extensions of classical data analysis tools to functional data, developed since pioneering works such as [17] and [14], are precisely described.

The simplest case of FDA corresponds to a situation in which all considered functions are discretized at the same points. More precisely, if we consider n functions $g^1, \ldots, g^n$ and m sampling points $x_1, \ldots, x_m$, we obtain n vector from $\mathbb{R}^m$, $(g^i(x_1), \ldots, g^i(x_m))$. While direct comparison between vectors remains possible, this type of data suffers from two drawbacks: high dimension vectors and high correlation between variables. [36] focuses on this situation and provides solutions that explicitly use the underlying functions $(g^i)_{1 \leq i \leq n}$. The general methodology uses the fact that most multivariate data analysis methods are based on scalar products and/or distance calculations which can be easily translated from a finite dimensional space to a functional space. A simple example is given by linear regression: if we want to predict a target variable in $\mathbb{R}$, y^i with a linear model on g^i, the classical model on the discretized function tries to model y^i as:

$$y^i = w_0 + \sum_{j=1}^{m} w_j g^i(x_j) + \varepsilon^i \tag{1}$$

A functional version is given by [37, 23, 7]:

$$y^i = w_0 + \int w(x) g^i(x) \mathrm{d}x + \varepsilon^i, \tag{2}$$

in which most numerical parameters of the model $(w_1, \ldots, w_m)$ have been replaced by an unique functional parameter. A simple yet powerful idea to implement the functional version of the model is to estimate each g^i thanks to the corresponding vector $(g^i(x_1), \ldots, g^i(x_m))$ and then to work on the approximated function. Classical solutions are based on spline approximations of both g^i and w (see [23, 33, 8] for

instance). They solve the variable correlation problem by reducing the effective dimension of the functional parameter thanks to regularity assumptions (e.g. bounded second derivative).

A very interesting side effect of estimating g^i thanks to its discretized version is to allow processing of irregularly sampled functions that are quite common in many applications, especially medical ones (see e.g. [6, 24, 27, 38]). Group smoothing techniques have been developed for these types of data: rather than estimating each g^i independently, one can try to optimize the global representation of all examples, either by EM like methods [27, 38] or using hybrid splines and cross-validation [3]. Moreover, functional transformations (such as derivative calculation, see [19, 18]) can be performed on the representation. It is therefore obvious that the functional view of high dimensional data gives much more possibilities than the bare multivariate analysis.

Many classical data analysis tools have been adapted to functional data. Principal Component Analysis was the first method studied in a functional framework by [17] and [14] (see also [15, 36] and [27]). Other linear methods have been studied more recently, such as Canonical Correlation Analysis [30], linear discriminant analysis [22, 26] and linear regression (as presented above [37, 23, 7]). Non linear models such as generalized linear models [25], slice inverse regression [21] and non-parametric kernel based estimation [19, 18] have also been reformulated to work on functional data. Unsupervised classification of functions has also been studied, as a quantization problem in [32] and more traditionally with k-means like approaches [1] or mixture models [28]. Additional references and discussions about functional data analysis can be found in [36].

Neural models have been recently adapted to functional data (see [41, 43, 42, 16, 39]). Building on extensions of multilayer perceptrons (MLPs) to arbitrary inputs studied in [44, 45, 46, 47], we have proposed in [41] a functional multilayer perceptron (FMLP) based on approximate calculation of some integrals. While this model has interesting theoretical properties (cf [41, 40]) and gives very satisfactory results on real world benchmarks, it suffers from the need of a specialized implementation and from long training times. In [43], we have proposed another functional MLP based on projection operators. This method has some advantages over the one studied in [41], especially because the projections can be implemented as a pre-processing step that transforms functions into adapted vector representations. The vectors obtained like this are then processed by a standard neural model. We have shown in [43] that this functional model performs very well on real world data. However, this illustration was only experimental.

In this paper, we study theoretically the capabilities of projection based functional MLPs. We first recall in section 2 the definition of the functional multilayer perceptron and its projection based implementation. In section 3, we show that any continuous function from a compact sub-space of a functional space to $\mathbb{R}$ can be approximated arbitrarily well by projection based FMLPs, which are therefore universal approximators. In section 4, we show that functional MLPs can learn arbitrary mappings from a functional space to $\mathbb{R}$. More precisely, we show that the asymptotic generalization error of functional MLPs converges to the minimum possible error, provided the training is done properly. Proofs are gathered in section 6.

2. Multilayer perceptrons with functional inputs

2.1. Introduction

In this section, we recall the definition of functional multilayer perceptrons given in [43]. We focus on regular functions. More precisely, we denote μ a σ-finite positive Borel measure defined on $\mathbb{R}^p$ and $L^2(\mu)$ the space of measurable real valued functions[1] defined on $\mathbb{R}^p$ and such that $\int f^2 \mathrm{d}\mu < \infty$. $L^2(\mu)$ is a Hilbert space equipped with its natural inner product $\langle f, g \rangle = \int fg \mathrm{d}\mu$ (we denote $\|f\|_2 = \sqrt{\langle f, f \rangle}$).

To avoid cumbersome notations, this paper is restricted to data described by a single function valued variable. However, the results can be easily extended to the case of data described by several functional variables. We also restrict ourselves to one real valued output, but results are also valid for vector valued output.

2.2. Theoretical model

As recalled in the introduction and explained in [36], many data analysis methods are based on the Hilbert structure of the input space rather than on its finite dimension. Using this idea, [43] defines multilayer perceptrons with functional inputs, as recalled here.

A multilayer perceptron (MLP) consists in neurons that perform very simple calculations. Given an input $x \in \mathbb{R}^p$, the output of a neuron is

$$T\left(\beta_0 + \sum_{i=1}^{p} \beta_i x_i\right), \tag{3}$$

[1] More precisely, $L^2(\mu)$ contains equivalence classes of functions that differ only on a μ-negligible set.

where x_i is the i-th coordinate of x, T is an activation function from $\mathbb{R}$ to $\mathbb{R}$, and $\beta_0, \ldots, \beta_p$ are numerical parameters (the weights of the neuron).

The sum $\sum_{i=1}^{p} \beta_i x_i$ is in fact the inner product in $\mathbb{R}^p$ between x and $(\beta_1, \ldots, \beta_p)$. As proposed in [41, 43], a functional neuron can be defined thanks to the inner product in $L^2(\mu)$. Given an input $g \in L^2(\mu)$, the output of a functional neuron is

$$T(\beta_0 + \langle w, g \rangle) = T\left(\beta_0 + \int wg \mathrm{d}\mu\right), \tag{4}$$

where w is a function from $L^2(\mu)$, the "weight function". This functional neuron is in fact a special case of neurons with arbitrary input spaces defined in previous theoretical works [44, 45, 46, 47].

As the output of a generalized neuron is a numerical value, we need such neurons only in the first layer of the MLP. Indeed, the second layer uses only outputs from the first layer which are real numbers and therefore consists in numerical neurons. For example, a single hidden layer perceptron with an unique output neuron maps a functional input g to

$$H(g) = \sum_{l=1}^{L} a_l T\left(\beta_{0l} + \int w_l g \mathrm{d}\mu\right), \tag{5}$$

where L denotes the number of hidden (functional) neurons and $a_1, \ldots, a_L$ are real valued connexion weights of the output neuron (it has a linear activation function).

2.3. Projection

While the model presented in the previous section is a simple generalization of its numerical counterpart, it cannot be used in practice, as only a limited class of functions can be easily manipulated on a computer. Those functions are obtained as combinations (sum, product, composition, etc.) of elementary functions: polynomial functions, trigonometric functions, etc.

In order to solve this problem, FDA methods rely in general on projections. Let us indeed consider a finite p-dimensional subspace of $L^2(\mu)$, denoted V_p. The main principle of projection based FDA methods is to constrain all manipulated functions to belong to V_p rather than to $L^2(\mu)$. This constraint is implemented thanks to an orthogonal projection on V_p. More precisely, let us denote Π_p the orthogonal projection operator on V_p. Given an arbitrary input function g, the output

of a functional neuron constructed thanks to V_p is given by

$$T\left(\beta_0 + \int \Pi_p(w)\Pi_p(g)\mathrm{d}\mu\right). \tag{6}$$

The main advantage of using V_p is that it can be obtained as the vector space spanned by "computer friendly" functions, that is, functions that are easy to evaluate on a computer. One possibility consists in using a Hilbert basis of $L^2(\mu)$, that is a complete orthonormal system $(\phi_k)_{k\in\mathbb{N}^*}$. Useful examples include wavelets and trigonometric functions. Then V_p is defined as the vector space spanned by $(\phi_k)_{1\leq k\leq p}$.

Another possibility consists in using spline spaces, that is vector spaces of piecewise polynomial functions, or more generally, specific V_p that have been chosen because Π_p is easy to calculate and functions in V_p are easy to manipulate.

On the theoretical point of view and in the general case, V_p is given by an orthonormal basis $(\phi_{p,k})_{1\leq k\leq p}$. This basis allows to identify V_p with $\mathbb{R}^p$. We denote π_p the coordinate map, that is the function from $L^2(\mu)$ to $\mathbb{R}^p$ that maps g to the coordinates of $\Pi_p(g)$ on the basis $(\phi_{p,k})_{1\leq k\leq p}$, i.e., to a vector in $\mathbb{R}^p$ such that $\Pi_p(g) = \sum_{k=1}^p \pi_p(g)_k\phi_{p,k}$. We have:

$$\int \Pi_p(w)\Pi_p(g)\mathrm{d}\mu = \sum_{k=1}^{p} \pi_p(w)_k\pi_p(g)_k. \tag{7}$$

This shows, as explained in [43], that the projection approach corresponds to a pre-processing step that transforms functional inputs into finite dimensional inputs. A simple way to implement a projection based functional MLP consists in using a standard MLP to which the p coordinates of the projected functions are submitted (the MLP uses therefore standard vector inputs in $\mathbb{R}^p$). The resulting model gives exactly the same output as a functional MLP build for functional inputs in V_p.

3. Universal approximation

3.1. Definition

This section is dedicated to the approximation capabilities of the functional MLP described in the previous section. We first recall a definition of universal approximation.

If A and B are two topological spaces, we denote $C(A, B)$ the set of continuous functions from A to B.

Definition 1. Let X be a topological space and $\mathcal{B}$ be a set of continuous functions from X to $\mathbb{R}$. We say that $\mathcal{B}$ has the universal approximation property for X if for any compact subset of X, K, $\mathcal{B}$ is dense in $C(K, \mathbb{R})$ for the uniform norm.

In other words, if $\mathcal{B}$ has the universal approximation property for X, for any compact subset K, any continuous function f from K to $\mathbb{R}$, and any requested precision $\epsilon > 0$, there is $g \in \mathcal{B}$ such that $\sup_{x \in K} |f(x) - g(x)| = \|f - g\|_\infty < \epsilon$.

3.2. Projection and universal approximation

When functions are processed thanks to a projection, approximation capabilities depend both on the neural model and on the projection. It is quite obvious that universal approximation cannot be reached if MLPs are constrained to work on a fixed V_p subset. Indeed, most of the functions in $L^2(\mu)$ are very poorly approximated by their projections on V_p for a fixed set of functions $(\phi_k)_{1 \leq k \leq p}$. Therefore, the neural models have not enough information on their actual inputs to provide meaningful outputs. To solve this problem, we need to consider more and more precise projections.

Definition 2. Let us consider a sequence of functions from $L^2(\mu)$ $(\phi_{p,k})_{p \in \mathbb{N}^*, 1 \leq k \leq p}$ such that for each p, $(\phi_{p,k})_{1 \leq k \leq p}$ is an orthonormal system. We denote V_p the subspace of $L^2(\mu)$ spanned by $(\phi_{p,k})_{1 \leq k \leq p}$ and Π_p the orthogonal projection operator on V_p.

Let $\mathcal{G}$ be a subset of $L^2(\mu)$. The sequence $(\Pi_p)_{p \in \mathbb{N}^*}$ (and the corresponding sequence of functions) is said to have the point-wise approximation property for $\mathcal{G}$ if Π_p converges to $Id_{\mathcal{G}}$ on $\mathcal{G}$ for the point-wise convergence: for all $g \in \mathcal{G}$, $\lim_{p \to \infty} \|\Pi_p(g) - g\|_2 = 0$.

A simple example of sequence with the point-wise approximation property for $L^2(\mu)$ is given by any Hilbert basis $(\phi_k)_{k \in \mathbb{N}^*}$ of this space. Indeed, any function g in $L^2(\mu)$ has a series expansion $g = \sum_{k=1}^{\infty} g_k \phi_k$. Therefore, the sequence defined by $\phi_{p,k} = \phi_k$ has obviously the point-wise approximation property.

Thanks to those increasingly accurate projections, we can construct a set of MLP based functions with the universal approximation property for $L^2(\mu)$.

Theorem 1. Let T be a continuous non polynomial function from $\mathbb{R}$ to $\mathbb{R}$ and let $(\phi_{p,k})_{p \in \mathbb{N}^*, 1 \leq k \leq p}$ be a sequence of functions from $L^2(\mu)$ with the point-wise approximation property for $L^2(\mu)$. Let us denote

$\mathcal{S}(T, (\phi_{p,k})_{p\in\mathbb{N}^*, 1\leq k\leq p})$ the set of functions from $L^2(\mu)$ to $\mathbb{R}$ of the form

$$g \mapsto \sum_{l=1}^{L} a_l T\left(\beta_{l0} + \sum_{k=1}^{p} \beta_{lk} \pi_p(g)_k\right),$$

where $L \in \mathbb{N}^*$, $p \in \mathbb{N}^*$, $\beta_{lk} \in \mathbb{R}$ and $a_l \in \mathbb{R}$ (π_p is the coordinate map defined in section 2.3).

Then $\mathcal{S}(T, (\phi_{p,k})_{p\in\mathbb{N}^*, 1\leq k\leq p})$ has the universal approximation property for $L^2(\mu)$.

3.3. Relation to previous works

A lot of work has been done on the topic of universal approximation properties of multilayer perceptrons (see, e.g., [47, 34] for reviews). For functional inputs, pioneering work can be found in [10]. This paper proves that single hidden layer perceptrons with functional inputs have the universal approximation property for $C([a, b], \mathbb{R})$ and $L^p([a, b])$. Those results are based either on the exact calculation of some specific integrals (for $L^p([a, b])$) or on a vector representation of functions based on an evaluation map (for $C([a, b], \mathbb{R})$): g is replaced by $(g(x_1), \ldots, g(x_n))$. Those results have been improved in more recent papers [12, 9].

Other pioneering work can be found in [44]: this paper shows that some specific feed-forward architecture with functional inputs has the universal approximation property. This result relies on perfect calculation of inner products. Generalizations of this result can be found in [46, 47].

Finally, [11] studies a projection based approach for Radial Basis Function Network and [45] studies the approximate realization of the model proposed in [44] thanks to projection. Both works are related to the model proposed in the present paper. The novelty of our approach consists in allowing complex projection methods whereas [11, 45] are limited to truncated basis representation. The complex projection methods covered by Theorem 1, especially those based on spline approximations, have been used successfully in [43] for real world data.

4. Consistency

4.1. Introduction

While universal approximation is an important property, it is not sufficient to ensure that the considered model can be used with success for

some machine learning task. Another problem must be assessed: is it possible to design, from a finite set of examples, a functional MLP such that when the number of examples goes to infinity, the FMLP provides a more and more accurate approximation of the underlying relationship between the input functions and the numerical outputs? This question (the "learnability") has been studied in details in the case of numerical MLP, see e.g. [48, 2, 31].

To give a precise mathematical translation of this question, we introduce the following notations (we follow [31]). We denote (G, Y) a pair of random variables, defined on probability space $(\Omega, \mathcal{M}, P)$, that take their values from $L^2(\mu)$ and $\mathbb{R}$, respectively. Our goal is to predict the value of Y given G. To assess the quality of this prediction, we need an error measure. In this paper, we use the root mean square error, but any L_p-error could be used[2]. Given a function h from $L^2(\mu)$ to $\mathbb{R}$, the root mean square prediction error is defined as

$$\mathcal{C}(h) = \mathbf{E}\left[(h(G) - Y)^2\right]^{\frac{1}{2}}, \tag{8}$$

where $\mathbf{E}[.]$ denotes the expectation. If we assume that $\mathbf{E}\left[|Y|^2\right] < \infty$, then $\mathcal{C}$ is minimized by the conditional expectation of Y given G, i.e., by $h(g) = \mathbf{E}[Y|G = g]$. We denote $\mathcal{C}^*$ the minimal root mean square error, i.e.

$$\mathcal{C}^* = \inf_h \mathcal{C}(h) = \mathbf{E}\left[(\mathbf{E}[Y|G] - Y)^2\right]^{\frac{1}{2}}. \tag{9}$$

We have no information about the distribution of (G, Y), except for n independent, identically distributed (i.i.d.) copies of (G, Y),

$$D_n = ((G^1, Y^1), \ldots, (G^n, Y^n)).$$

Using this data set, we can build a prediction model h_n (from $L^2(\mu)$ to $\mathbb{R}$). The model depends on D_n and its performances are given by the following random variable

$$\mathcal{C}(h_n) = \mathbf{E}\left[(h_n(G) - Y)^2 | D_n\right]^{\frac{1}{2}}. \tag{10}$$

A sequence of prediction models $(h_n)_{n \in \mathbb{N}^*}$ is **universally consistent** (see [31]) if $\mathcal{C}(h_n)$ converges almost surely to $\mathcal{C}^*$, for any distribution (G, Y) satisfying $\mathbf{E}\left[|Y|^2\right] < \infty$. The intuitive interpretation of this condition is that given enough data (when n goes to infinity), the root mean square error of h_n will be arbitrarily close to the best possible root mean square error: we are indeed *learning* the relationship between

[2] There is no relation between the functional input space $L^2(\mu)$ and the use of the mean square error.

Y and G from examples. Another way to look at the condition is to rewrite it into the following equivalent condition:

$$\mathbf{E}\left[(h_n(G) - \mathbf{E}\left[Y|G\right])^2|D_n\right]^{\frac{1}{2}} \xrightarrow[n\to\infty]{} 0 \ a.s. \tag{11}$$

This condition means that $h_n(G)$ is arbitrarily close to $\mathbf{E}\left[Y|G\right]$ for the mean square error.

4.2. Projection and consistency

In this section, we restrict the projection approach to the simple case of sequences of projection spaces constructed thanks to a Hilbert basis of the functional space. More precisely, we assume given $(\phi_p)_{p\in I\!N^*}$ a Hilbert basis of $L^2(\mu)$. We denote V_p the sub-vector space spanned by $(\phi_k)_{1\leq k\leq p}$ and define π_p as in section 2.3. In order to build a consistent learning method based on projection on V_p spaces, we need to adapt the expressive power of the candidate neural networks to the size of the learning set (i.e., to n). Rather than choosing an arbitrary single hidden layer perceptron, we restrict the search to some classes of such perceptrons. More precisely, given $(L_n)_{n\in I\!N^*}$ a sequence of integers and $(\alpha_n)_{n\in I\!N^*}$ a sequence of positive real values, we define $\mathcal{H}_{np}$, a sequence of single hidden layer functional perceptron classes, by:

$$\mathcal{H}_{np} = \left\{ h \in C(L^2(\mu), I\!R) \middle| \right. \\ \left. h(g) = \sum_{l=1}^{L_n} a_l T\left(\beta_{l0} + \sum_{k=1}^{p} \beta_{lk}\pi_p(g)_k\right), \sum_{k=1}^{L_n} |a_l| \leq \alpha_n \right\}. \tag{12}$$

In those classes, L_n and α_n provide a type of regularization by adapting the number of hidden neurons (thanks to L_n) and the magnitude of the weights of the output layer (thanks to α_n) to the size of the learning set. A consistent sequence of models will be obtained by choosing the best single hidden layer perceptron in $\mathcal{H}_{np}$, according to the empirical error (see Theorem 2).

To obtain consistency, we need some technical hypotheses:

(H-1) T is a function from $I\!R$ to $[0,1]$, monotone non decreasing, with $\lim_{x\to\infty} T(x) = 1$ and $\lim_{x\to-\infty} T(x) = 0$;

(H-2) $(L_n)_{n\in I\!N^*}$ and $(\alpha_n)_{n\in I\!N^*}$ are such that

$$\lim_{n\to\infty} L_n = \infty$$
$$\lim_{n\to\infty} \alpha_n = \infty;$$

(H-3) $(L_n)_{n\in\mathbb{N}^*}$ and $(\alpha_n)_{n\in\mathbb{N}^*}$ are such that

$$\lim_{n\to\infty} \frac{L_n \alpha_n^4 \log(L_n \alpha_n)}{n} = 0,$$

and such that there is $\delta > 0$ such that

$$\lim_{n\to\infty} \frac{\alpha_n^4}{n^{1-\delta}} = 0.$$

Hypothesis (H-1) corresponds to a standard requirement for activation functions of multilayer perceptrons. It is fulfilled for instance by $T(x) = 1/(1+e^{-x})$.

Hypothesis (H-2) ensure that the expressive power of the considered classes is not limited asymptotically, as the regularization vanishes asymptotically.

Hypothesis (H-3) corresponds the regularization. The constraints come from [31] (stronger constraints were used in [48]). They control the way the expressive power of $\mathcal{H}_{np}$ grows with n.

Some possible choices for L_n and α_n include $L_n = \lceil \log n \rceil$ (where $\lceil x \rceil$ denotes the smallest integer greater or equal to x) and $\alpha_n = n^{\frac{1}{8}}$.

Under those hypotheses, we have the following consistency result.

Theorem 2. Let h_{np} be a function that minimizes the empirical mean square error in $\mathcal{H}_{np}$, i.e. such that

$$\frac{1}{n}\sum_{i=1}^{n} (h_{np}(G^i) - Y^i)^2 \leq \frac{1}{n}\sum_{i=1}^{n} (h(G^i) - Y^i)^2,$$

for all $h \in \mathcal{H}_{np}$.

Under hypotheses (H-1), (H-2) and (H-3), we have

$$\lim_{p\to\infty} \lim_{n\to\infty} \mathcal{C}(h_{np}) = \mathcal{C}^* \ [a.s.],$$

for all distributions of (G, Y) such that $\mathbf{E}\left[|Y|^2\right] < \infty$.

The theorem means that a functional MLP h_{np} that minimizes its mean square error on a training set with n examples provides a more and more accurate approximation of $\mathbf{E}\left[Y|G\right]$ when n goes to infinity. The theorem provides some rules on L_n and α_n that allow to avoid overfitting and guarantee good generalization. The only limitation of this result comes from the sequential limit: the theorem does not provide guidelines to link p to n.

It should be noted that the theorem could be adapted to any model that is universally consistent in finite dimension.

5. Conclusion

We have demonstrated in this paper two important results for projection based functional multilayer perceptrons: they have the universal approximation property and they can learn arbitrary mapping. Thanks to the representation of the studied functions through projection, we adapt the strong results available for standard numerical MLPs to functional MLPs. This gives a satisfactory theoretical backing to the method proposed in [43].

However, some questions remain open, especially if we want to fully justify the method illustrated in [43]. The more important point is the choice of the projection quality, i.e., of V_p. In [43], it was determined thanks to the input data alone. The goal was to limit the distortion between $\Pi_p(g)$ and g. Further theoretical investigation of this method is needed.

Another possibility is to use a split sample or a re-sampling technique to choose an optimal V_p, as we did in [41]. In practice, this introduces a huge computational load, without much practical gain. However, models constructed like this are universally consistent in the case of classification [5].

In practice, some very good results have been obtained in [20] thanks to an automatic construction of V_p based on a functional version of the Slice Inverse Regression. While the authors provide important theoretical results, the consistency of this method remains an open question.

Finally, the second more important open question is related to the very nature of functional data: in practice, functional data are always given as finite sets of (input, output) pairs. As a consequence, projected functions cannot be exactly computed and are replaced by approximations (see [43] for details). The effects of this approximation on the capabilities of functional MLP have been partially studied in [13] but the consistency of models constructed thanks to approximate values is not yet established.

6. Proofs

6.1. Theorem 1

This theorem is based on results given in [47] for MLPs with arbitrary inputs.

Let us consider a compact subset of $L^2(\mu)$, K. We want to approximate functions in $C(K, \mathbb{R})$ by functions in $\mathcal{S}(T, (\phi_{p,k})_{p\in\mathbb{N}^*, 1\leq k\leq p})$.

6.1.1. *Step one*
As a first step, we prove that the sequence of operator $(\Pi_p)_{p\in I\!N^*}$ converges to Id_K uniformly on K, i.e. for $\eta > 0$, there is P such that for each $p \geq P$ and for each $g \in K$, $\|\Pi_p(g) - g\|_2 < \eta$ (P does not depend on g).

Let us consider $g_0 \in K$, and $K(g_0, r) = B(g_0, r) \cap K$ neighborhood of g_0 in K, where $B(g_0, r)$ denotes the open ball of radius r centered on g_0. As $(\Pi_p)_{p\in I\!N^*}$ has the point-wise approximation property, there is P_0 such that for each $p_0 \geq P_0$, $\|\Pi_{p_0}(g_0) - g_0\|_2 < \eta/2$. For each $g \in K(g_0, r)$, we have $\|\Pi_{p_0}(g) - g\|_2 \leq \|\Pi_{p_0}(g) - \Pi_{p_0}(g_0)\|_2 + \|\Pi_{p_0}(g_0) - g_0\|_2 + \|g_0 - g\|_2$. As requested above, the middle term is smaller than $\eta/2$. As Π_{p_0} is Lipschitz continuous (the Lipschitz constant is 1), $\|\Pi_{p_0}(g) - \Pi_{p_0}(g_0)\|_2 \leq \|g - g_0\|_2$. Therefore, $\|\Pi_{p_0}(g) - g\|_2 \leq \eta/2 + 2\|g - g_0\|_2$. As a consequence, $\forall g \in K(g_0, \eta/4)$, $\|\Pi_{p_0}(g) - g\|_2 < \eta$. As K is compact, it is covered by a finite number of $B(g_i, \eta/4)$ (and therefore of $K(g_i, \eta/4)$). We consider $P = \max P_i$, which allows to conclude.

6.1.2. *Step two*
Let us now denote $\mathcal{S}(T, L^2(\mu))$ the set of functions form $L^2(\mu)$ to $I\!R$ of the form

$$g \mapsto \sum_{l=1}^{L} a_l T\left(\beta_{l0} + \langle w_l, g\rangle\right), \tag{13}$$

where $l \in I\!N^*$, $p \in I\!N^*$, $\beta_{l0} \in I\!R$ and $w_l \in L^2(\mu)$. Then, $\mathcal{S}(T, L^2(\mu))$ has the universal approximation property for $L^2(\mu)$.

Indeed, Corollary 5.1.2 of [47] can be applied, as its conditions are fulfilled:

- $L^2(\mu)$ is locally convex and is isometric to its topological dual;
- as T is continuous and non-polynomial, single hidden layer perceptrons using T as their activation function have the universal approximation property for $I\!R$ (see [34] for instance).

6.1.3. *Step three*
Let us now consider a continuous function F from K to $I\!R$ and let $\epsilon > 0$ be an arbitrary precision.

According to step two, there is $H \in \mathcal{S}(T, L^2(\mu))$, given by equation 13, such that for all $g \in K$, $|H(g) - F(g)| < \frac{\epsilon}{2}$.

As H is continuous on $L^2(\mu)$, for each $g \in K$ there is $\eta(g) > 0$ such that for each $f \in B(g, \eta(g))$, we have $|H(g) - H(f)| \leq \frac{\epsilon}{4}$. As K is com-

pact, it is covered by a finite number of the balls $\left(B\left(g_i, \frac{\eta(g_i)}{2}\right)\right)_{1\leq i\leq N}$. We denote $\eta = \min_{1\leq i\leq N} \eta(g_i)$.

According to Step one of the proof, there is p such that for all $g \in K$, $\|\Pi_p(g) - g\|_2 < \frac{\eta}{2}$. There is i such that g falls in $B\left(g_i, \frac{\eta(g_i)}{2}\right)$ and we have

$$\|\Pi_p(g) - g_i\|_2 \leq \|\Pi_p(g) - g\|_2 + \|g - g_i\|_2 < \eta(g_i),$$

which implies

$$|H(\Pi_p(g)) - H(g)| \leq |H(\Pi_p(g)) - H(g_i)| + |H(g_i) - H(g)| < \frac{\epsilon}{2},$$

by using twice the continuity of H at g_i. We have therefore

$$|H(\Pi_p(g)) - F(g)| < \epsilon.$$

To conclude, we note that $\langle w_l, \Pi_p(g)\rangle = \langle \Pi_p(w_l), \Pi_p(g)\rangle = \sum_{k=1}^{p} \pi_p(w_l)_k \pi_p(g)_k$, which means that $H \circ \Pi_p$ belongs to $\mathcal{S}(T, (\phi_{p,k})_{p\in I\!N^*, 1\leq k\leq p})$.

6.2. Theorem 2

Theorem 2 is based on theorem 3 from [31]. The latter applies to standard MLPs with inputs in $I\!R^d$ and provides universal consistency.

6.2.1. *Step one*

To use theorem 3 from [31], we need to introduce additional notations. We denote $G_p = \pi_p(G)$. As π_p is continuous, G_p is a random variable that takes values from $I\!R^p$. We denote $G_p^i = \pi_p(G^i)$. Obviously, for any p, $D_n^p = ((G_p^1, Y^1), \ldots, (G_p^n, Y^n))$ consists in n i.i.d. copies of (G_p, Y).

If f_n is a measurable function from $I\!R^p$ to $I\!R$ constructed thanks to D_n^p, we denote

$$\mathcal{C}_p(f_n) = \mathbf{E}\left[(f_n(G_p) - Y)^2 | D_n^p\right]^{\frac{1}{2}}.$$

We denote

$$\mathcal{C}_p^* = \inf_f \mathbf{E}\left[(f(G_p) - Y)^2\right]^{\frac{1}{2}},$$

where the infimum is taken over all measurable functions from $I\!R^p$ to $I\!R$. As $\mathbf{E}\left[|Y|^2\right] < \infty$, $\mathcal{C}_p^*$ is reached for f defined by $f(g) = \mathbf{E}\left[Y | G_p = g\right]$.

Each function h in $\mathcal{H}_{np}$ can be written $h = f \circ \pi_p$, where f is chosen in $\mathcal{F}_{np}$ defined by

$$\mathcal{F}_{np} = \left\{ f \in C(\mathbb{R}^p, \mathbb{R}) \middle| \right.$$
$$\left. f(x) = \sum_{l=1}^{L_n} a_l T\left(\beta_{l0} + \sum_{k=1}^{p} \beta_{lk} x_k\right), \text{with} \sum_{k=1}^{L_n} |a_l| \leq \alpha_n \right\},$$

Moreover, a function $f_{np} \in \mathcal{F}_{np}$ such that $h_{np} = f_{np} \circ \pi_p$ has obviously the smallest empirical error among functions in $\mathcal{F}_{np}$, that is

$$\frac{1}{n} \sum_{i=1}^{n} (f_{np}(G_p^i) - Y^i)^2 \leq \frac{1}{n} \sum_{i=1}^{n} (f(G_p^i) - Y^i)^2,$$

for all $f \in \mathcal{F}_{np}$. Then, according to theorem 2 from [31] and thanks to hypothesis on L_n and α_n, for any fixed p, $\lim_{n\to\infty} \mathcal{C}_p(f_{np}) = \mathcal{C}_p^*$. In other words, $\lim_{n\to\infty} \mathcal{C}_p(h_{np}) = \mathcal{C}_p^*$ (almost surely).

6.2.2. *Step two*
We show now that $\lim_{p\to\infty} \mathcal{C}_p^* = \mathcal{C}^*$. Let us consider the sequence of random variables $X_p = \mathbf{E}[Y|G_p]$ and the sequence of σ-fields $\mathcal{M}_p = \sigma(G_p)$. We first show that $(\mathcal{M}_p)_{p\in\mathbb{N}^*}$ is a filtration, i.e., that $\mathcal{M}_p \subset \mathcal{M}_{p+1}$. This is a simple consequence of the definition of G_p. Indeed, $G_p = \pi_p(G)$ and therefore $G_p = \nu_p(G_{p+1})$ where ν_p is the function from $\mathbb{R}^{p+1}$ to $\mathbb{R}^p$ defined by

$$\nu_p(x_1, \ldots, x_p, x_{p+1}) = (x_1, \ldots, x_p).$$

As G_p is the composition of a continuous function and of G_{p+1}, the σ-field generated by G_p is a subset of the σ-field generated by G_{p+1}.

As $\mathbf{E}\left[|Y|^2\right] < \infty$, $\mathbf{E}[|Y|] < \infty$. This allows to apply Lemma 35 from [35] (page 154), from which we conclude that $(X_p)_{p\in\mathbb{N}^*}$ is an uniformly integrable martingale for the $\mathcal{M}_p$ filtration. Therefore, according to Theorem 36 from [35] (page 154), $(X_p)_{p\in\mathbb{N}^*}$ converges almost surely to an integrable random variable X_∞. Moreover, as $X_p = \mathbf{E}[Y|\mathcal{M}_p]$, according to the same theorem, $X_\infty = \mathbf{E}\left[Y \middle| \sigma\left(\bigcup_{p\in\mathbb{N}^*} \mathcal{M}_p\right)\right]$. Obviously, we have $\sigma\left(\bigcup_{p\in\mathbb{N}^*} \mathcal{M}_p\right) = \sigma(G)$ and therefore $(X_p)_{p\in\mathbb{N}^*}$ converges almost surely to $\mathbf{E}[Y|G]$.

Finally, as $\mathbf{E}\left[|Y|^2\right] < \infty$, $\mathbf{E}\left[|X_p|^2\right] \leq \mathbf{E}\left[|Y|^2\right] < \infty$ and therefore, the convergence also happens for the quadratic norm (see Corollary 6.22 from [29]), i.e.

$$\lim_{p\to\infty} \mathbf{E}\left[\left(\mathbf{E}[Y|G_p] - \mathbf{E}[Y|G]\right)^2\right]^{\frac{1}{2}} = 0.$$

This clearly implies $\lim_{p\to\infty} \mathcal{C}_p^* = \mathcal{C}^*$ (almost surely).

Acknowledgements

The authors thank the anonymous referees for their valuable suggestions that help improving this paper.

References

1. Abraham, C., P.-A. Cornillon, E. Matzner-Lober, and N. Molinari: 2003, 'Unsupervised Curve Clustering using B-Splines'. *Scandinavian Journal of Statistics* **30**(3), 581–595.
2. Barron, A. R.: 1994, 'Approximation and Estimation Bounds for Artificial Neural Networks'. *Machine Learning* **14**, 115–133.
3. Besse, P., H. Cardot, and F. Ferraty: 1997, 'Simultaneous non-parametric regressions of unbalanced longitudinal data'. *Computational Statistics and Data Analysis* **24**, 255–270.
4. Besse, P. and J. Ramsay: 1986, 'Principal component analysis of sampled curves'. *Psychometrica* **51**, 285–311.
5. Biau, G., F. Bunea, and M. Wegkamp: 2005, 'Functional Classification in Hilbert Spaces'. *IEEE Transactions on Information Theory* **51**, 2163–2172.
6. Brumback, B. A. and J. A. Rice: 1998, 'Smoothing Spline Models for the Analysis of Nested and Crossed Samples of Curves'. *J. Amer. Statist. Assoc.* **93**, 961–994.
7. Cardot, H., F. Ferraty, and P. Sarda: 1999, 'Functional Linear Model'. *Statist. & Prob. Letters* **45**, 11–22.
8. Cardot, H., F. Ferraty, and P. Sarda: 2003, 'Spline Estimators for the Functional Linear Model'. *Statistica Sinica* **13**, 571–591.
9. Chen, T.: 1998, 'A unified approach for neural network-like approximation of non-linear functional'. *Neural Networks* **11**, 981–983.
10. Chen, T. and H. Chen: 1993, 'Approximation of Continuous Functionals by Neural Networks with Application to Dynamic Systems'. *IEEE Transactions on Neural Networks* **4**(6), 910–918.
11. Chen, T. and H. Chen: 1995a, 'Approximation Capability to Functions of Several Variables, Nonlinear Functionals, and Operators by Radial Basis Function Neural Networks'. *IEEE Transactions on Neural Networks* **6**(4), 904–910.
12. Chen, T. and H. Chen: 1995b, '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.
13. Conan-Guez, B. and F. Rossi: 2002, 'Multilayer Perceptrons for Functional Data Analysis: a Projection Based Approach'. In: J. R. Dorronsoro (ed.): *Artificial Neural Networks – ICANN 2002*. Madrid (Spain), pp. 667–672.
14. Dauxois, J. and A. Pousse: 1976, 'Les analyses factorielles en calcul des probabilités et en statistiques : essai d'étude synthétique'. Thèse d'état, Université Paul Sabatier, Toulouse.

15. Dauxois, J., A. Pousse, and Y. Romain: 1982, 'Asymptotic theory for the principal component analysis of a vector of random function: some applications to statistical inference'. *Journal of Multivariate Analysis* **12**, 136–154.
16. Delannay, N., F. Rossi, B. Conan-Guez, and M. Verleysen: 2004, 'Functional Radial Basis Function Network'. In: *Proceedings of XIIth European Symposium on Artificial Neural Networks (ESANN 2004)*. Bruges (Belgium), pp. 313–318.
17. Deville, J.: 1974, 'Méthodes statistiques et numériques de l'analyse harmonique'. *Annales de l'INSEE* **15**, 3–97.
18. Ferraty, F. and P. Vieu: 2002, 'The Functional Nonparametric Model and Application to Spectrometric Data'. *Computational Statistics* **17**(4).
19. Ferraty, F. and P. Vieu: 2003, 'Curves Discriminations: a Nonparametric Functional Approach'. *Computational Statistics and Data Analysis* **44**(1–2), 161–173.
20. Ferré, L. and N. Villa: 2004, 'Multi-layer Neural Network with functional inputs: an inverse regression approach'. *Submitted to Scandinavian Journal of Statistics*.
21. Ferré, L. and A.-F. Yao: 2003, 'Functional sliced inverse regression analysis'. *Statistics* **37**(6), 475–488.
22. Hastie, T., A. Buja, and R. Tibshirani: 1995, 'Penalized Discriminant Analysis'. *Annals of Statistics* **23**, 73–102.
23. Hastie, T. and C. Mallows: 1993, 'A discussion of "A Statistical View of Some Chemometrics Regression Tools" by I.E. Frank and J.H. Friedman'. *Technometrics* **35**, 140–143.
24. Hoover, D. R., J. A. Rice, C. O. Wu, and L.-P. Yang: 1998, 'Nonparametric Smoothing Estimates of Time-Varying Coefficient Models With Longitudinal Data'. *Biometrika* **85**(4), 809–822.
25. James, G. M.: 2002, 'Generalized Linear Models with Functional Predictor Variables'. *Journal of the Royal Statistical Society Series B* (64), 411–432.
26. James, G. M. and T. J. Hastie: 2001, 'Functional Linear Discriminant Analysis for Irregularly Sampled Curves'. *Journal of the Royal Statistical Society Series B* **63**, 533–550.
27. James, G. M., T. J. Hastie, and C. A. Sugar: 2000, 'Principal component models for sparse functional data'. *Biometrika* **87**(3), 587–602.
28. James, G. M. and C. A. Sugar: 2003, 'Clustering for sparsely sampled functional data'. *Journal of American Statistical Association* **98**, 397–408.
29. Kallenberg, O.: 1997, *Foundations of Modern Probability*, Probability and its Applications. Springer.
30. Leurgans, S., R. Moyeed, and B. Silverman: 1993, 'Canonical Correlation Analysis when the Data are Curves'. *Journal of the Royal Statistical Society B* **55**(3), 725–740.
31. Lugosi, G. and K. Zeger: 1995, 'Nonparametric Estimation via Empirical Risk Minimization'. *IEEE Transactions on Information Theory* **41**(3), 677–687.
32. Luschgy, H. and G. Pages: 2002, 'Functional quantization of Gaussian processes'. *Journal of Functional Analysis* **196**(2), 486–531.
33. Marx, B. D. and P. H. Eilers: 1996, 'Generalized Linear Regression on Sampled Signals with Penalized Likelihood'. In: R. H. A. Forcina, G. M. Marchetti and G. Galmacci (eds.): *Statistical Modelling. Proceedings of the 11th International workshop on Statistical Modelling*. Orvietto.
34. Pinkus, A.: 1999, 'Approximation Theory of the MLP Model in neural networks'. *Acta Numerica* pp. 143–195.

35. Pollard, D.: 2002, *A User's Guide to Measure Theoretic Probability*, Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press.
36. Ramsay, J. and B. Silverman: 1997, *Functional Data Analysis*, Springer Series in Statistics. Springer Verlag.
37. Ramsay, J. O. and C. J. Dalzell: 1991, 'Some tools for functional data analysis (with Discussion)'. *Journal of the Royal Statistical Society Series B* **53**, 539–572.
38. Rice, J. A. and C. O. Wu: 2001, 'Nonparametric Mixed Effects Models for Unequally Sampled Noisy Curves'. *Biometrics* **57**(1), 253–259.
39. Rossi, F. and B. Conan-Guez: 2004, 'Functional Preprocessing for Multilayer Perceptrons'. In: *Proceedings of XIIth European Symposium on Artificial Neural Networks (ESANN 2004)*. Bruges (Belgium), pp. 319–324.
40. Rossi, F. and B. Conan-Guez: 2005a, 'Estimation consistante des paramètres d'un modèle non linéaire pour des données fonctionnelles discrétisées aléatoirement'. *Comptes rendus de l'Académie des Sciences - Série I* **340**(2), 167–170.
41. Rossi, F. and B. Conan-Guez: 2005b, 'Functional Multi-Layer Perceptron: a Nonlinear Tool for Functional Data Analysis'. *Neural Networks* **18**(1), 45–60.
42. Rossi, F., B. Conan-Guez, and A. El Golli: 2004, 'Clustering Functional Data with the SOM algorithm'. In: *Proceedings of XIIth European Symposium on Artificial Neural Networks (ESANN 2004)*. Bruges (Belgium), pp. 305–312.
43. Rossi, F., N. Delannay, B. Conan-Guez, and M. Verleysen: 2005, 'Representation of Functional Data in Neural Networks'. *Neurocomputing* **64**, 183–210.
44. Sandberg, I. W.: 1994, 'General Structures for Classification'. *IEEE Transactions on Circuits and Systems–I: Fundamental Theory and Applications* **41**(5), 372–376.
45. Sandberg, I. W.: 1996, 'Notes on Weighted Norms and Network Approximation of Functionals'. *IEEE Transactions on Circuits and Systems–I: Fundamental Theory and Applications* **43**(7), 600–601.
46. Sandberg, I. W. and L. Xu: 1996, 'Network approximation of input-output maps and functionals'. *Circuits Systems Signal Processing* **15**(6), 711–725.
47. Stinchcombe, M. B.: 1999, 'Neural network approximation of continuous functionals and continuous functions on compactifications'. *Neural Networks* **12**(3), 467–477.
48. White, H.: 1990, 'Connectionist Nonparametric Regression: Mutilayer Feedforward Networks Can Learn Arbitrary Mappings'. *Neural Networks* **3**, 535–549.