

# Functional Multi-Layer Perceptrons\*

Fabrice Rossi<sup>1†</sup>, Briec Conan-Guez<sup>2</sup> and François Fleuret<sup>2</sup>

December 2001

<sup>1</sup> LISE/CEREMADE, UMR CNRS 7534, Université Paris-IX Dauphine,  
Place du Maréchal de Lattre de Tassigny, 75016 Paris, France

<sup>2</sup> INRIA, Domaine de Voluceau, Rocquencourt, B.P. 105  
78153 Le Chesnay Cedex, France

## Abstract

In this paper, 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 the one of numerical MLP. We obtain consistency results which imply that optimal parameters estimation for functional MLP is statistically well defined.

## 1 Introduction

Functional Data Analysis (FDA, see [5]) 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  $\mathbb{R}^n$ . The main advantage of FDA is to take into account dependencies between numerical measurements that describe an individual. If we represent for instance the size of a child at different ages by a vector, traditional methods consider each value to be independent from the others. In FDA, the size is represented as a regular function that maps measurement times to centimeters. FDA methods take explicitly into account the smoothness assumptions on the size function.

In this paper, we show how Multi-Layer Perceptrons (MLP) can be directly applied to functional data, so as to provide non linear function classification and regression. The non linear modeling that can be obtained by functional MLP is not the only difference with classical linear FDA methods. In general those methods rely on a finite dimensional representation of studied functions, for instance thanks to a spline based approximation (see [5] for a presentation of traditional FDA methods). In our model, we work directly with the studied functions, without using a simplified representation (section 5 describes precisely how this can be done on a practical point of view).

The extension of MLP we propose is very close to an extension proposed and studied on a pure theoretical point of view in [7]. In [7], M. Stinchcombe 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 approximation of continuous linear forms defined on the MLP input space. In our work, we show how to practically realize this kind of approximation, for instance by using traditional MLP.

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 the real optimal parameters. This is a direct translation of classical results, exposed in [8] for instance, available for numerical MLP.

---

\*CEREMADE preprint 0134 (2001) available at <http://www.ceremade.dauphine.fr>

†Up to date contact informations for Fabrice Rossi are available at <http://apiacoo.org/>

The rest of the paper is organized as follows: we start by introducing in section 2 the proposed functional MLP model. Then we show in section 3 how results of [7] can be adapted to functional MLP to show they are universal approximators. In section 4 we introduce calculation of derivative for parametric functional MLP by an adapted back-propagation algorithm as well as our first consistency result. In section 5 we show how we can practically manipulate functions thanks to finite set of (input, output) pairs and we introduce the full consistency result adapted to this limited knowledge. We end this last section with a short discussion on links between numerical MLP and functional MLP in some particular cases.

## 2 From vectors to functions

### 2.1 Numerical neurons

A multi-layer perceptron (MLP) is built with simple units called neurons. A  $n$  inputs 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  $T(w \cdot x + b)$ .

### 2.2 Functional neurons

It is quite simple to define a functional neuron, i.e., a simple unit that mimics the calculation of a numerical neuron, but deals with functional inputs.

If  $E$  is a vectorial space and  $E^*$  its dual, then a functional neuron is characterized by a “weight form”,  $w \in E^*$ , an activation function  $T$  from  $\mathbb{R}$  to  $\mathbb{R}$  and a real valued threshold,  $b$ . Given an input  $x \in E$ , the output of the neuron is  $T(w(x) + b)$ . Obviously, if  $E = \mathbb{R}^n$ , a functional neuron is in fact a numerical neuron. Our goal is not here to radically change the neuron model. It would be possible to allow arbitrary “weight functions”, that is to use for  $w$  any function from  $E$  to  $\mathbb{R}$ . We think that without the linear restriction, we are not anymore working with neural networks and therefore that this kind of extension is outside the scope of this article.

Of course, representing elements in  $E^*$  is not easy when  $E$  is not a finite dimensional vectorial space. Fortunately, some particular cases are easier to handle. Let  $\mu$  be a  $\sigma$ -finite positive measure on a measurable space  $X$ . Then when  $1 \leq p < \infty$ ,  $(L^p(\mu))^*$  can be identified with  $L^q(\mu)$ , where  $q$  is  $p$  conjugate exponent. More precisely, for each continuous linear form  $w$  on  $L^p(\mu)$ , there is an unique function  $f$  in  $L^q(\mu)$  such that for each  $g \in L^p(\mu)$ ,  $w(g) = \int fg d\mu$  (see [6] for instance). In this case a functional neuron on  $L^p(\mu)$  is characterized by a function  $f$  in  $L^q(\mu)$ , an activation function  $T$  from  $\mathbb{R}$  to  $\mathbb{R}$  and a real valued threshold,  $b$ . It computes the function  $H(g) = T(b + \int fg d\mu)$ .

More generally, we can define  $H(g) = T(b + \int fg d\mu)$  as long as the product  $fg$  is  $\mu$ -integrable for all  $g \in E$ . If we consider for instance  $E = C_0(\mathbb{R}^n, \mathbb{R})$ , the set of compactly supported continuous functions from  $\mathbb{R}^n$  to  $\mathbb{R}$ , we can take  $f \in L^\infty(\mu)$ . In such cases, using a simple representation for the functional neuron introduces a restriction on its weight set, which is now strictly included into  $E^*$ .

Using functions rather than linear forms might seem a matter of convenience. In fact, the very first assumption of FDA is that we are able to directly manipulate functions: FDA methods show that working with functions is almost as easy as working with vectors of  $\mathbb{R}^n$ . If we assume that integrals can be calculated (see section 5 for further discussion about this), then using linear forms which can be represented by integrals is almost as easy as using functions. Therefore, we believe that functional neurons have a similar complexity as other FDA methods. Moreover, we will introduce further simplifications in this model in subsection 4.1, without loosing any computing power.

## 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 functional perceptron with real output computes a function of the following form:

$$H(g) = \sum_{i=1}^k a_i T(b_i + w_i(g)), \quad (1)$$

where the  $a_i$  are real numbers, as well the  $b_i$ , and  $w_i$  are continuous linear forms on  $E$ .

As in previous section, we can consider the case of linear forms that are defined thanks to an integral, for instance:

$$H(g) = \sum_{i=1}^k a_i T\left(b_i + \int f_i g d\mu\right), \quad (2)$$

where  $f_i$  are functions of a given functional space.

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.

## 2.4 Multiple inputs

In this article, we restrict ourselves to the case where each individual is defined by a single function from a subset of  $\mathbb{R}^n$  to  $\mathbb{R}$ . As explained below, it is very simple to extend the proposed model and results to multiple inputs, but the presentation is simpler when we deal with only one input.

Allowing each individual to be described by one function with vectorial values (in  $\mathbb{R}^p$ ) is a special case of allowing multiple real valued functions for each individual: the unique vectorial valued function is replaced by its coordinate functions.

Let us therefore focus on the case where each individual is described by  $p$  real valued functions. This case is obviously included in the general description proposed in section 2.2. Let us consider indeed that each individual belongs to  $E_1 \times E_2 \dots \times E_p$ , where each  $E_i$  is a vectorial space of real valued functions. Then obviously,  $E = E_1 \times E_2 \dots \times E_p$  is also a vectorial space. Using elements in  $E^*$ , we can define a multi-functional neuron exactly as it is done in section 2.2. Moreover, if we can use a simple representation for elements of  $E_i^*$  for all  $i$ , then this representation is also valid for  $E^*$ , as a linear form on  $E$  is a linear combination of linear forms on  $E_i$ . Therefore, we define a functional neuron with  $p$  inputs as follows:  $H(g_1, \dots, g_p) = T(b + \sum_{i=1}^p w_i(g_i))$ , where each  $w_i$  belongs to  $E_i^*$ . The output of a functional MLP with  $p$  inputs is given by the following equation:

$$H(g_1, \dots, g_p) = \sum_{i=1}^k a_i T\left(b_i + \sum_{j=1}^p w_{ij}(g_j)\right)$$

As in the previous section,  $w_{ij}$  can be represented by an integral.

# 3 Computing power

## 3.1 Definitions and existing results

Following [7], we introduce some definitions:

**Definition 1.** 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  $(w_i, b_i) \in \mathbb{R}^{n+1}$ .

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 functional 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  $b_i$ .

Several approximation results show that  $S_T^n$  and  $S_T^X(A)$  are inside or outside dense in different functional spaces. For instance, we have [4]:

**Theorem 1 (Hornik 93).** *If  $T$  a measurable function from  $\mathbb{R}$  to  $\mathbb{R}$  is non polynomial and Riemann integrable on some compact interval (not reduced to one point) of  $\mathbb{R}$ , then  $S_T^n$  contains a subset that is uniformly dense on compacta in  $C(\mathbb{R}^n, \mathbb{R})$  (the space of continuous functions from  $\mathbb{R}^n$  to  $\mathbb{R}$ ).*

and [3]:

**Theorem 2 (Hornik 91).** *If  $T$  from  $\mathbb{R}$  to  $\mathbb{R}$  is measurable, bounded and non constant, and if  $\mu$  is a finite measure on  $\mathbb{R}^n$ , then  $S_T^n$  is dense in  $L^p(\mu)$  for  $1 \leq p < \infty$ .*

In the very general case, we have [7]:

**Theorem 3 (Stinchcombe 99).** *Suppose that  $S_T^1$  contains a subset that is uniformly dense on compacta in  $C(\mathbb{R}, \mathbb{R})$ . Let  $X$  be a topological space,  $K$  a compact subset of  $X$  and  $A$  a vector space of real valued measurable functions on  $X$ . If the intersection of  $C(K, \mathbb{R})$  and the closure of  $A$  according to the uniform norm contains the constant functions and separates points in  $K$ , then for each compact set  $K \subset X$ ,  $S_T^X(A)$  contains a set that is dense for the uniform norm in  $C(K, \mathbb{R})$ .*

In the infinite dimensional context, the previous theorem can be simplified into the following corollary:

**Corollary 1 (Stinchcombe 99).** *Suppose that  $S_T^1$  contains a subset that is uniformly dense on compacta in  $C(\mathbb{R}, \mathbb{R})$ . Let  $X$  be a topological vector space and let  $A$  be a dense subset of  $X^*$  (considered with the weak  $*$  topology). Then for each compact set  $K \subset X$ ,  $S_T^X(A)$  contains a set that is dense for the uniform norm in  $C(K, \mathbb{R})$ .*

## 3.2 Corollaries for functional MLP

Our goal is to adapt the recalled results to functional MLP. Of course, theorem 3 and corollary 1 give more or less everything that is needed on a theoretical point of view: as long as  $A$  and  $T$  satisfy some quite general properties,  $S_T^X(A)$  is powerful enough to allow approximation of continuous functions from a compact subset of  $X$  to  $\mathbb{R}$ .

But on the practical point of view, this is not directly helpful, as manipulating elements of  $A$ , (in general, continuous linear forms on  $X$ ), is not easy in the general case. In this section, we show that a kind of two levels approximation is in fact implied by theorem 3: if we can approximate linear forms on  $X$ , then we can also approximate functions in  $C(K, \mathbb{R})$ .

**Corollary 2.** *Let  $\mu$  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  $M$  be a dense subset of  $L^q(\mu)$ . Let  $A_M$  be the set of linear forms on  $L^p(\mu)$  of the form  $l(f) = \int fg d\mu$ , where  $g \in M$ . Let  $T$  be a measurable function from  $\mathbb{R}$  to  $\mathbb{R}$  that is non polynomial and Riemann integrable on some compact interval (not reduced to one point) of  $\mathbb{R}$ . Then  $S_T^{L^p(\mu)}(A_M)$  contains a set that is dense for the uniform norm in  $C(K, \mathbb{R})$ , where  $K$  is any compact subset of  $L^p(\mu)$ .*

This corollary shows that as long as we can approximate functions in  $L^q(\mu)$ , then a functional MLP can be used to approximate functions in  $C(K, \mathbb{R})$ , where  $K$  is a compact subset of  $L^p(\mu)$ . Theorem 2 provides general approximation results for  $L^q(\mu)$ , which means that  $M$  can be  $S_U^n$  for instance, for a well chosen  $U$ .

**Corollary 3.** *Let  $\mu$  be a finite positive compactly supported Borel measure on  $\mathbb{R}^n$ . Let  $T$  be a measurable function from  $\mathbb{R}$  to  $\mathbb{R}$ , that is non polynomial and Riemann integrable on some compact interval (not reduced to one point) of  $\mathbb{R}$ . Let  $M$  be a subset of  $L^\infty(\mu)$  that contains a set which is uniformly dense on compacta in  $C(\mathbb{R}^n, \mathbb{R})$ . Then  $S_T^{L^1(\mu)}(A_M)$  contains a set that is dense for the uniform norm in  $C(K, \mathbb{R})$ , where  $K$  is a compact subset of  $L^1(\mu)$ .*

The proof of corollary 2 could be extended to  $p = 1$ , and therefore, one might wonder why corollary 3 is useful. As pointed out in [7], no  $S_T^n$  set is dense in  $L^\infty(\mu)$ . Therefore, corollary 2 main assumption ( $M$  is dense  $L^q(\mu)$ ) cannot be satisfied by MLP based approximation. This reduces greatly the interest of corollary 2 for  $p = 1$ . That's why corollary 3 is useful: as shown by theorem 1,  $S_T^n$  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 are represented thanks to numerical MLPs). This means that 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 2. Indeed, continuous functions (or more regular functions) on a compact subset  $W$  of  $\mathbb{R}^n$  are obviously elements of  $L^\infty(\lambda)$  where  $\lambda$  is the restriction of the Lebesgue measure to  $W$ . 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.

### 3.3 Multiple inputs

Corollaries proposed in the previous subsection are based on theorem 3 and corollary 1. Both results make very few assumptions on the input space of "extended" MLP. Those assumptions are obviously fulfilled by products of  $L^p$  spaces such as  $X = L^{p_1}(\mu_1) \times \dots \times L^{p_r}(\mu_r)$ . In order to apply corollary 1 we basically need to be able to approximate elements of  $X^*$ . As an element of  $X^*$  is a linear combination of elements of  $(L^{p_i}(\mu_i))^*$ , this does not introduce any problem as long as  $1 < p_i < \infty$  (see proof of corollary 2, subsection 7.1). In this case, it is easy to apply corollary 1 and therefore to extend corollary 2 to  $r$  inputs functional MLP.

The case of  $p_i = \infty$  is handled in a similar way, but relies directly on theorem 3. For  $p_i = 1$ , we have to add the hypothesis that the corresponding measure  $\mu_i$  is compactly supported. Anyway, the  $r$  inputs fulfill conditions of corollaries 2 and 3, we can apply theorem 3 to conclude that universal approximation is again possible.

## 4 Training Functional MLP

### 4.1 Parametric approach

The simplest way to use functional MLP in practical settings is to rely on parametric approximation of functions used to represent linear forms. More precisely, we assume that  $W$  is a compact subset of  $\mathbb{R}^t$  and that we have a function  $F$  from  $W \times \mathbb{R}^n$  such that for each  $w \in W$ , the function  $x \mapsto F(w, x)$  belongs to  $L^q(\mu)$ , where  $\mu$  be a finite positive Borel measure on  $\mathbb{R}^n$ . Rather than working with an arbitrary dense subset of  $L^q(\mu)$ , we define  $M_{F,W} = \{g \in L^q(\mu) \mid \exists w \in W, \forall x \in \mathbb{R}^n F(w, x) = g(x)\}$ . We call  $F$  a parametric regressor.

A parametric regressor allows to define a functional neuron as a function  $H$  from  $W \times L^p(\mu)$  to  $\mathbb{R}$  as follows:

$$H(w, b, g) = T \left( b + \int F(w, x)g(x)d\mu(x) \right), \quad (3)$$

where  $T$  is the activation function of the neuron, a function from  $\mathbb{R}$  to  $\mathbb{R}$ , and  $b$  is a real number (the threshold of the neuron).

Of course, it is possible to use a different parametric regressor for each functional neuron in the functional MLP. Therefore, a one hidden layer functional perceptron computes the following function from  $W_h \times \mathbb{R}^{2k} \times L^p(\mu)$  to  $\mathbb{R}$ :

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

where  $W_h = W^1 \times \dots \times W^k$  (and therefore  $w_h = (w_h^1, \dots, w_h^k)$ ) and  $w_o = (a_1, b_1, a_2, b_2, \dots, a_k, b_k)$ .

In the parametric framework, training a neural network means adjusting its parameters in order to minimize a given objective function, in general a distance between a target function and the function calculated by the network. One way to optimize such a distance is to use gradient based algorithms, which implies to compute the derivative of the distance with respect to the different numerical parameters, thanks to back-propagation (more precisely a generalized one, see [2]). In order to do this, we must compute the derivative of the function calculated by a functional neuron with respect to the parameters of its parametric regressor.

## 4.2 Derivatives for a functional neuron

We need to derivate the mapping  $w \mapsto H(w, b, g) = T \left( b + \int F(w, x) g(x) d\mu(x) \right)$ . If we assume the partial derivative  $\frac{\partial F}{\partial w}$  exists  $\mu$ -almost everywhere, is measurable, and that there is a positive function  $f$  in  $L^q(\mu)$  such that  $|\frac{\partial F}{\partial w}(w, x)| \leq f(x)$  for all  $\mu$ -almost  $x$ , then the mapping  $w \mapsto \int F(w, x) g(x) d\mu(x)$  is differentiable, and we have:

$$\frac{\partial \int F(w, x) g(x) d\mu(x)}{\partial w}(w, g) = \int \frac{\partial F}{\partial w}(w, x) g(x) d\mu(x)$$

Now, if the activation function  $T$  itself is derivable, we have:

$$\frac{\partial H}{\partial w}(w, b, g) = T' \left( b + \int F(w, x) g(x) d\mu(x) \right) \int \frac{\partial F}{\partial w}(w, x) g(x) d\mu(x)$$

If  $F$  is computed thanks to a MLP,  $\frac{\partial F}{\partial w}$  can be computed efficiently thanks to an extended back propagation (see [2]). If  $F$  provides a vectorial output, the extended back propagation is much faster than a basic one, because we can compute  $\frac{\partial F}{\partial w}(w, x) g(x)$  directly rather than computing first  $\frac{\partial F}{\partial w}$  and then the scalar product (details can be found in [1]).

## 4.3 MLP case

The case of functional MLPs is covered by the extended back propagation defined in [2, 1]. The main requirement of this work is to be able to compute the derivative of the output of each neuron with respect to its parameters, and with respect to its inputs (except for input neurons). We are exactly in this case for functional neurons as previous section allows to compute the needed derivative (functional neurons can only be used as input neurons).

## 4.4 Consistency

Our goal is to obtain an approximation of a mapping from a set of functions to  $\mathbb{R}$ . There is no *a priori* reason for this to be possible if we know only a finite number of input/output pairs. It is well known that a statistical analysis of neural network learning allows to prove that neural model parameters estimation is consistent (see [8]). Our aim is to prove a similar result for functional neural networks.

Following White [8], we prove a general result that makes minimal assumptions on the error function and on the exact calculation done by the functional parametric model. We have:

**Theorem 4.** *Let  $X$  be an arbitrary metric space considered with its Borel sigma algebra. Let  $(\Omega, \mathcal{A}, P)$  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  $w \in W$ ,  $|l(w, x)| < d(x)$ .*
4.  *$E(d(Z_t)) < \infty$ .*

*Then for each  $n$ , there exists a solution  $\hat{w}_n$  to the problem  $\min_{w \in W} \hat{\lambda}_n(w)$ , where  $\hat{\lambda}_n(w) = \frac{1}{n} \sum_{i=1}^n l(w, Z_i)$ . If  $W^*$  is the set of minimizers on  $W$  of the function  $\lambda(w) = E(l(w, Z_t))$ , then  $d(\hat{w}_n, W^*) \rightarrow 0$   $P$  almost surely.*

This theorem is an extension of theorem 1 from [8]. In [8],  $X$  and  $W$  are finite dimensional vectorial spaces. The ability to extend the theorem to arbitrary dimensions allows us to apply it to our practical problem. We consider that the training set of our functional neural network is made of a finite number of input/output pairs,  $(g_i, y_i)$  where  $g_i$  is a function in  $L^p(\mu)$  and  $y_i$  is the corresponding expected output (a real number). Each example is considered as a realization of a random variable  $Z_i$  with value in  $X = L^p(\mu) \times \mathbb{R}$ . The  $Z_i$  are considered independent and identically distributed.  $W$  is the set of usable parameters for the functional neural network and  $l$  combines both the mapping performed by the MLP and an error measure (for instance the quadratic error). The practical meaning of the theorem is that by optimizing the network parameters on a finite number of samples, we do not make systematic errors: the estimation is strongly consistent in the sense that if we increase the number of samples, the network parameters converge to the set of optimal parameters (almost surely).

As  $X$  can be an arbitrary metric space, it is obvious that this theorem applies to the case of multiple functional inputs.

## 5 Practical setting

### 5.1 Introduction

The proposed framework allows to precisely define how a neural network can be used to compute function from a functional space to  $\mathbb{R}$ . In practical settings, input functions are in general only known through a sample set, i.e., we know a set of input/output pairs  $(x_i, f(x_i))$ . In general, each  $x_i$  corresponds to a measurement that has been randomly chosen. Moreover, it is quite common to have different evaluation points for each input functions.

In the previous section, we showed that optimal parameters obtained with a finite number of samples converge almost surely to real optimal parameters. The problem is that the previous theorem relies on exact calculation of the error made by the network. In practical settings, we have only approximate knowledge of this error. We show in this section that the consistency result can be extended to the practical case.

### 5.2 Probabilistic framework

Let us assume that we have a sequence of sequences of evaluation points,  $(x_i^j)_{i \in N}$  such that each  $x_i^j$  is the realization of a random variable  $X_i^j$  with value in  $\mathbb{R}^n$  (the  $X_i^j$  variables are independent identically distributed random variables on the probability space  $(\Omega, \mathcal{A}, P)$  and we denote  $P_X$  the probability measure associated to every  $X_i^j$ ).

Let us consider for instance a functional one hidden layer perceptron that computes:

$$H(a, b, w, g) = \sum_{i=1}^k a_i T \left( b_i + \int w_i g dP_X \right),$$

where  $g \in L^p(P_X)$  and each  $w_i$  belongs to  $L^q(P_X)$ . We can define the following random variable:

$$\widehat{H}(a, b, w, g)_N^j = \sum_{i=1}^k a_i T \left( b_i + \frac{1}{N} \sum_{l=1}^N w_i(X_l^j) g(X_l^j) \right)$$

In practical setting, we have a sequence of functions  $g^j$ , each of which is associated to the corresponding sequence of evaluation points  $(x_i^j)_{i \in N}$  and we replace  $H(a, b, w, g^j)$  by the corresponding realization of  $\widehat{H}(a, b, w, g^j)_{m_j}^j$ , where  $m_j$  is the number of evaluation points taken into account for the actual calculation. Under reasonable assumptions,  $\widehat{H}(a, b, w, g^j)_{m_j}^j$  converges almost surely to  $H(a, b, w, g^j)$  when  $m_j$  goes to infinity. The problem is that theorem 4 relies on exact calculation of the error criterion, that is on exact calculation of each  $H(a, b, w, g^j)$  in order to provide almost sure convergence. Therefore, it cannot be applied directly to the practical case.

### 5.3 Consistency

Fortunately, we have the following theorem:

**Theorem 5.** *Let  $Z$  be an arbitrary metric space considered with its Borel sigma algebra. Let  $(\Omega, \mathcal{A}, P)$  be a probability space on which is defined a sequence of independent identically distributed random elements,  $X_j^i$ , with values in  $Z$  and let us denote  $P_X$  the induced measure on  $Z$  and  $X = X_1^1$ . Let  $k$  be a positive integer and  $F^1, \dots, F^k$  be  $k$  parametric regressors such that for each  $l$ :*

1.  $F^l$  is a function from  $W^j \times Z$  to  $\mathbb{R}$
2.  $W^l$  is a compact set
3. for each  $x \in Z$ ,  $F^l(\cdot, x)$  is a continuous function from  $W^l$  to  $\mathbb{R}$
4. for each  $w_h^l \in W^l$ ,  $F^l(w_h^l, \cdot)$  is a measurable function from  $Z$  to  $\mathbb{R}$
5. the function from  $X$  to  $\mathbb{R}$ ,  $\sup_{w \in W_h^l} |F^l(w, \cdot)|$  belongs to  $L^q(P_X)$ .

We denote  $W_h = W^1 \times \dots \times W^k$ . Let  $G^i$  be a sequence of independent identically distributed random elements defined on  $(\Omega, \mathcal{A}, P)$  with value in  $L^p(P_X)$  ( $p$  and  $q$  are conjugate exponents) and let us denote  $G = G^1$ .

Let  $O$  be an arbitrary metric space considered with its Borel sigma algebra and let  $Y^i$  be a sequence of independent identically distributed random elements defined on  $(\Omega, \mathcal{A}, P)$  with value in  $O$ . We denote  $Y = Y^1$ .

Let  $l$  be a function from  $\mathbb{R}^k \times O \times W_o$  to  $\mathbb{R}$ , where  $W_o$  is a compact set. We assume that:

1. for each  $y \in O$ ,  $l(\cdot, y, \cdot)$  is uniformly continuous on  $\mathbb{R}^k \times W_o$
2. for each  $w_o \in W_o$ ,  $l(\cdot, \cdot, w_o)$  is measurable on  $\mathbb{R}^k \times O$
3. there is a measurable function  $d'$  from  $O$  to  $\mathbb{R}$  such that  $|l(x, y, w_o)| < d'(y)$  for all  $x$  and  $w_o$
4.  $E(d'(Y)) < \infty$

We define

$$\widehat{\lambda}_n^m(w_h, w_o) = \frac{1}{n} \sum_{i=1}^n l \left( \frac{1}{m_i} \sum_{j=1}^{m_i} F^1(w_h^1, X_j^i) G^i(X_j^i), \dots, \frac{1}{m_i} \sum_{j=1}^{m_i} F^k(w_h^k, X_j^i) G^i(X_j^i), Y^i, w_o \right),$$

with  $m = \inf_{1 \leq i \leq n} m_i$ , and

$$\lambda(w_h, w_o) = E \left( l \left( E \left( F^1(w_h^1, X) G(X) \right), \dots, E \left( F^k(w_h^k, X) G(X) \right), Y, w_o \right) \right)$$

Then for each  $n$  and  $m$ , there exists a solution  $\widehat{w}_n^m$  to the problem  $\min_{w \in W_h \times W_o} \widehat{\lambda}_n^m(w_h, w_o)$ . If  $W^*$  is the set of minimizers of  $\lambda(w_h, w_o)$ , then  $\lim_{n \rightarrow \infty} \lim_{m \rightarrow \infty} d(\widehat{w}_n^m, W^*) = 0$   $P$  almost surely.

This theorem gives a direct answer to our problem. It shows indeed that even if we replace  $H(a, b, w, g^j)$  the exact output of a functional MLP by an approximated output  $\widehat{H}(a, b, w, g^j)_{m_j}^j$ , a consistent estimation of optimal parameters is still possible. The formulation of the theorem is a little bit technical because we hide into  $l$  both part of the functional MLP (everything except the integral calculation) and the error function. If we consider for instance a quadratic error, we can define  $l$  as follows :

$$l(u_1, \dots, u_k, y, w_o) = \left( \sum_{i=1}^k a_i T(b_i + u_i) - y \right)^2,$$

where  $w_o = (a_1, b_1, \dots, a_k, b_k) \in \mathbb{R}^{2k}$ . Each  $u_i$  is the output of the weight form associated to hidden neuron  $i$ .

The practical meaning of the theorem is that if both the number of functions and the number of evaluation points for each function go to infinity, the estimated optimal parameters converge almost surely to the true optimal parameters. The only limitation of the theorem is that the number of evaluation points needed to achieve a given accuracy depends on the number of functions.

## 5.4 Link with traditional approaches

The proposed model can be compared to traditional multi-layer perceptron if we make some restrictive hypothesis on the considered practical setting. Input functions are sometimes only known through a finite set of input/output pairs. Let us restrict further this assumption in this section: we assume that for each functional input submitted to the functional network, the input sampling points are identical. To simplify further the analysis, we assume that each functional input is evaluated at point  $j \in \{1, \dots, N\}$ . In this case, a functional input is simply described by a vector in  $\mathbb{R}^N$ .

In this particular case, the estimated output of a functional MLP can be rewritten as follows:

$$\widehat{H}(a, b, w, g)_N = \sum_{i=1}^k a_i T \left( b_i + \frac{1}{N} \sum_{j=1}^N w_i(j) g_j \right)$$

Let us now consider a traditional one hidden layer perceptron, based on the same activation function. Given a  $N$  dimensional input, its output is given by a formula of the form:

$$U(\alpha, \beta, p, g) = \sum_{i=1}^l \alpha_i T \left( \beta_i + \sum_{j=1}^N p_{ij} g_j \right)$$

Models are obviously very close. The main difference is that in the traditional approach, the connection weights ( $p_{ij}$ ) are freely chosen, whereas in the functional approach, they are replaced by the output of parametric regressors ( $\frac{1}{N} w_i(j)$ ). This difference have important consequences for complexity tuning of the neural network.

In the traditional model, the only meta parameter that can be tuned is the number of hidden neurons  $l$ . The number of numerical parameters is  $l(2 + N)$  and is directly proportional to the number of inputs. On the contrary, the functional model allows to choose both the number of hidden neurons  $k$  and the complexity of internal parametric regressors. If we consider the case of numerical one hidden layer perceptrons as parametric regressors, the number of numerical parameters is  $k(2 + 3m)$ , where  $m$  is the number of hidden neurons in each internal MLP. The main advantage of this approach is that the number of parameters is not directly linked to the number of inputs.

In fact, the proposed model is a kind of regularization method. Rather than allowing arbitrary weights, we restrict them to weight sets that can be computed by a function. It is traditional to use weight sharing when the number of input of the neural network is high. For image analysis for instance, it is quite common to use the linear combination part of the neural network in a convolution mode: the real set of weights is far smaller than the needed set. Missing weights are replaced by 0 (i.e.,  $w_i(j) = 0$ ) and a global set of weight is used to define a non linear moving filter (each  $w_i$  is obtained through a translation of a main  $w$ ).

## 6 Conclusion

We have introduced in this paper Functional Multi-Layer Perceptrons (FMLP), a natural extension of MLP to functional inputs. We have shown that the proposed model shares with numerical MLP two fundamental properties:

1. FMLP are universal approximators
2. FMLP parameters estimation is consistent

Those properties give strong theoretical foundations to FMLP. We need now to study practical behavior of the proposed model, especially on real world data. We hope to extend previously reported successes of Functional Data Analysis which were mostly based on linear models.

## 7 Mathematical Appendix

### 7.1 Corollary 2

*Proof.* If  $1 < p < \infty$ , we know that  $L^q(\mu)$  (with  $q < \infty$ ) can be identified with  $(L^p(\mu))^*$  (see for instance [6]). More precisely, for each  $l \in (L^p(\mu))^*$  there is a unique function  $f \in L^q(\mu)$  so that  $l(g) = \int fg d\mu$ . By hypothesis,  $M$  is dense in  $L^q(\mu)$ . This obviously implies that  $A_M$  is dense in  $(L^p(\mu))^*$  for the weak  $*$  topology. Hypothesis on  $T$  allows to apply theorem 1 which implies that  $S_T^1$  contains a subset that is uniformly dense on compacta in  $C(\mathbb{R}, \mathbb{R})$ . The conclusion is then obtained by applying corollary 1.

If  $p = \infty$ , we cannot apply directly corollary 1 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  $\alpha$  is an arbitrary real number and  $g$  is an arbitrary function from  $M \subset L^1(\mu)$ .  $A$  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  $\mu$  is finite,  $\chi_H$  belongs to  $L^1(\mu)$ . As  $M$  is dense in  $L^1(\mu)$ , there is a sequence  $h_k$  of functions in  $M$  that converges to  $\chi_H$ . 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  $h_k \in M$  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 3.  $\square$

### 7.2 Corollary 3

*Proof.* Thanks to Lusin theorem (e.g., [6]), we know 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  $\mu$  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  $\phi_k$  in  $M$  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  $\mu$  is compactly supported, this allows to conclude that  $\int \phi_k h d\mu \rightarrow_{k \rightarrow \infty} \int f h d\mu$ . Therefore, the set of linear forms  $A_M$  is dense for the weak  $*$  topology in  $(L^1(\mu))^*$ , provided that  $\mu$  is finite and compactly supported. The conclusion of the corollary is then obtained by applying corollary 1.  $\square$

### 7.3 Theorem 4

We need first an uniform strong law of large numbers:

**Theorem 6.** *Let  $X$  be an arbitrary metric space considered with its Borel sigma algebra. Let  $(\Omega, \mathcal{A}, P)$  be a probability space on which is defined a sequence of independent identically distributed random variables,  $Z_t$ . 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  $w \in W$ ,  $|l(w, x)| < d(x)$ .
4.  $E(d(Z_t)) < \infty$ .

Then we have:

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

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

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

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

We can now give the proof of the theorem:

*Proof.* Let us consider  $w_0 \in W$ . We call  $W(w_0, \epsilon) = B(w_0, \epsilon) \cap W$ , a compact neighborhood of  $w_0$ . Note that compactness of  $W(w_0, \epsilon)$  implies that this set is separable and therefore that lemma 1 can be applied. Let us show we have  $\lim_{\epsilon \rightarrow 0} E \left( \sup_{w \in W(w_0, \epsilon)} l(w, Z_t) \right) = E(l(w_0, Z_t))$ . First of all, thanks to the lemma, we know that  $\sup_{w \in W(w_0, \epsilon)} l(w, \cdot)$  is measurable. Next, it is obvious that for each  $x \in X$ ,  $\sup_{w \in W(w_0, \epsilon)} l(w, x) \rightarrow l(w_0, x)$  when  $\epsilon$  goes to 0, by continuity of  $l$ . Moreover,  $\left| \sup_{w \in W(w_0, \epsilon)} l(w, x) \right| < d(x)$ . Then the proposed result holds thanks to dominated convergence. Obviously, this result is also true for  $E \left( \inf_{w \in W(w_0, \epsilon)} l(w, Z_t) \right)$ .

Let us fix an arbitrary real number  $\epsilon > 0$ . Then, for each  $w_0 \in W$ , there is  $\eta > 0$  (that depends on  $w_0$ ) such that  $E \left( \sup_{w \in W(w_0, \eta)} l(w, Z_t) \right) \leq E(l(w_0, Z_t)) + \frac{\epsilon}{3}$  and  $E \left( \inf_{w \in W(w_0, \eta)} l(w, Z_t) \right) \geq E(l(w_0, Z_t)) - \frac{\epsilon}{3}$ . Then as

$$\sup_{w \in W(w_0, \eta)} \left( \frac{1}{n} \sum_{i=1}^n l(w, Z_i) - E(l(w, Z_t)) \right) \leq \frac{1}{n} \sum_{i=1}^n \sup_{w \in W(w_0, \eta)} l(w, Z_i) - E \left( \inf_{w \in W(w_0, \epsilon)} l(w, Z_t) \right)$$

we have

$$\sup_{w \in W(w_0, \eta)} \left( \frac{1}{n} \sum_{i=1}^n l(w, Z_i) - E(l(w, Z_t)) \right) \leq \frac{1}{n} \sum_{i=1}^n \sup_{w \in W(w_0, \eta)} l(w, Z_i) - E(l(w_0, Z_t)) + \frac{\epsilon}{3}$$

But  $\sup_{w \in W(w_0, \eta)} l(w, x)$  is measurable (and integrable when composed with  $Z_t$  by domination by  $d$ ), and therefore, the strong law of large numbers says that, almost surely,

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

Let us consider a converging realization,  $z_i$ . Then there is  $N$  such that  $n \geq N$  implies

$$\frac{1}{n} \sum_{i=1}^n \sup_{w \in W(w_0, \eta)} l(w, z_i) \leq E \left( \sup_{w \in W(w_0, \eta)} l(w, Z_t) \right) + \frac{\epsilon}{3} \leq E(l(w_0, Z_t)) + \frac{2\epsilon}{3}$$

Therefore,  $n \geq N$  implies

$$\sup_{w \in W(w_0, \eta)} \left( \frac{1}{n} \sum_{i=1}^n l(w, z_i) - E(l(w, Z_t)) \right) \leq \epsilon$$

As  $W$  is compact, it is covered by a finite number of  $W(w_j, \eta_j)$ . As this covering is finite, convergence occurs almost surely simultaneously on each  $W(w_j, \eta_j)$ . For a converging realization  $z_i$ , there is a  $N$  such that  $n \geq N$  implies the majoration is valid on each  $W(w_j, \eta_j)$  and therefore that

$$\sup_{w \in W} \left( \frac{1}{n} \sum_{i=1}^n l(w, z_i) - E(l(w, Z_t)) \right) \leq \epsilon$$

A similar proof shows that for each converging realization  $z_i$  (which occurs almost surely) there is  $N$  such that  $n \geq N$  implies

$$\inf_{w \in W} \left( \frac{1}{n} \sum_{i=1}^n l(w, z_i) - E(l(w, Z_t)) \right) \leq \epsilon$$

Therefore, for each converging realization  $z_i$  (which occurs almost surely) there is  $N$  such that  $n \geq N$  implies

$$\sup_{w \in W} \left| \frac{1}{n} \sum_{i=1}^n l(w, z_i) - E(l(w, Z_t)) \right| \leq \epsilon$$

Therefore,

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

□

With this uniform strong law of large numbers, we can prove theorem 4 exactly as the corresponding theorem in [8].

*Proof.* We have the following steps:

1.  $\lambda(w)$  is continuous. By continuity of  $l$ ,  $l(w', x)$  converges punctually to  $l(w, x)$  when  $w'$  converges to  $w$ . Moreover,  $l$  is dominated by  $d$ . Therefore, the dominated convergence theorem implies that  $E(l(w', Z_t))$  converges to  $E(l(w, Z_t))$  when  $w'$  converges to  $w$ .
2. let us consider a realization  $z_i$  for which uniform convergence of  $\hat{\lambda}_n$  to  $\lambda$  occurs. Of course, each  $\hat{\lambda}_n$  is continuous on a compact set and therefore, there is a sequence  $\hat{w}_n$  of minimizers. Because  $W$  is compact, this sequence has at least an accumulation point,  $w_0$ , with a subsequence  $\hat{w}_{n'}$  that converges to it. Let  $\epsilon$  be an arbitrary positive real number.  $\lambda$  is uniformly continuous on  $W$  and therefore there is  $\eta$  such that  $|w' - w| < \eta$  implies  $|\lambda(w) - \lambda(w')| < \epsilon$ . By uniform convergence, for  $n'$  sufficiently large,  $\|\hat{\lambda}_{n'} - \lambda\|_\infty < \epsilon$ . For  $n'$  sufficiently large, we have also  $|\hat{w}_{n'} - w_0| < \eta$ . This implies  $|\hat{\lambda}_{n'}(\hat{w}_{n'}) - \lambda(w_0)| < 2\epsilon$ . This implies that for any  $w$ ,  $\lambda(w_0) - \lambda(w) \leq 3\epsilon$ , because  $\hat{w}_{n'}$  optimality implies  $\hat{\lambda}_{n'}(\hat{w}_{n'}) - \hat{\lambda}_{n'}(w) \leq 0$ ,  $\hat{\lambda}_{n'}(w) - \lambda(w) \leq \epsilon$  by uniform convergence and we have just proved that  $\lambda(w_0) - \hat{\lambda}_{n'}(\hat{w}_{n'}) < 2\epsilon$ . As this is true for all  $\epsilon$ , we have for all  $w$ ,  $\lambda(w_0) \leq \lambda(w)$ , which shows that  $w_0 \in W^*$ .
3. finally, let us assume that we don't have  $d(\hat{w}_n, W^*) \rightarrow 0$ . Then there is a positive real number  $\epsilon$  and a subsequence,  $\hat{w}_{n'}$  such that  $d(\hat{w}_{n'}, W^*) > \epsilon$  for all  $n'$ . But  $\hat{w}_{n'}$  is still a sequence of minimizer in a compact set and as therefore an accumulation point in  $W^*$  which is impossible as  $d(\hat{w}_{n'}, W^*) > \epsilon$ .

□

### 7.4 Theorem 5

*Proof.* First of all, we apply the uniform strong law of large numbers to  $F^l(w_h^l, X)G(X)$  (theorem 6). This is possible according to the following reasons:

- the function  $h(x, w) = F^l(w, x)g(x)$ , where  $g$  belongs to  $L^p(P_X)$ , is continuous with respect to  $w$  for each  $x$ , according to hypotheses on  $F^l$
- $h$  is also measurable with respect to  $x$  for each  $w$ , again as a direct consequence of the hypotheses
- according to lemma 1 applied to  $|h|$ , the function  $d(x) = \sup_{w \in W_h^l} |F^l(w, x)g(x)|$  is measurable
- $E(d(X)) \leq E\left(\left(\sup_{w \in W_h^l} |F^l(w, X)|\right)^q\right)^{\frac{1}{q}} E(|g(X)|^p)^{\frac{1}{p}} < \infty$  again based on hypotheses on  $F^l$  and  $g$ .

According to theorem 6, we therefore have

$$\sup_{w \in W_h^l} \left| \frac{1}{m_i} \sum_{j=1}^{m_i} F^l(w, X_j^i)g(X_j^i) - E(F^l(w, X)g(X)) \right| \xrightarrow[m_i \rightarrow \infty]{a.s.} 0$$

for any measurable function  $g \in L^p(P_X)$ .

As a second step, we apply again the uniform strong law of large numbers to the function  $k(w_h, w_o, g, y) = l(E(F^1(w_h^1, X)g(X)), \dots, E(F^k(w_h^k, X)g(X)), y, w_o)$ . This is possible according to the following reasons:

- the function  $k'((w_h, w_o), (g, y)) = k(w_h, w_o, g, y)$  is continuous on  $w = (w_h, w_o)$  for each  $x = (g, y)$ , according to hypotheses on  $l$  and on  $F^1, \dots, F^k$ , and provided that  $g$  belongs to  $L^p(P_X)$ . Indeed,  $E(F^j(w_h^j, X)g(X))$  is continuous on  $w_h^j$  for each  $g$ : as  $F^j$  is continuous on  $w$  for each  $x$ , the function  $F^j(w', \cdot)g(\cdot)$  converges punctually to  $F^j(w, \cdot)g(\cdot)$  when  $w'$  converges to  $w$ . Moreover,  $|F^j(w, \cdot)| |g(\cdot)|$  is dominated by  $\sup_{w \in W_h^l} |F^l(w, \cdot)| |g(\cdot)|$ , which is integrable by hypothesis. Thanks to dominated convergence theorem, this obviously implies the continuity of  $E(F^j(w_h^j, X)g(X))$ .
- $k'$  is measurable with respect  $(g, y)$  for each  $(w_h, w_o)$ . This is a direct consequence of hypotheses on  $l$  and of the fact that  $E(F^j(w_h^j, X)g(X))$  is continuous on  $g$  for each  $w_h^j$
- according to lemma 1 applied to  $|k'|$ , the function

$$c(g, y) = \sup_{(w_h, w_o) \in W_h \times W_o} |k'((w_h, w_o), (g, y))|$$

is measurable

- $E(c(G, Y)) < \infty$  again by hypothesis on  $l$

According to theorem 6, we therefore have

$$\sup_{(w_h, w_o) \in W_h \times W_o} \left| \frac{1}{n} \sum_{i=1}^n k(w_h, w_o, G^i, Y^i) - E(k(w_h, w_o, G, Y)) \right| \xrightarrow[n \rightarrow \infty]{a.s.} 0$$

Now we move to the final step. Let us consider a sequence of  $G^i$  and  $Y^i$  for which the previous convergence occurs (such sequences occur almost surely), respectively  $g^i$  and  $y^i$ . Let  $\varepsilon > 0$  be an arbitrary real number. According to the previous result, there is  $N$  such that for each  $n \geq N$ ,

$$\sup_{(w_h, w_o) \in W_h \times W_o} \left| \frac{1}{n} \sum_{i=1}^n k(w_h, w_o, g^i, y^i) - E(k(w_h, w_o, G, Y)) \right| < \frac{\varepsilon}{2}$$

Let us fix  $n$  bigger than  $N$ . As  $l$  is uniformly continuous with respect to  $z$  and  $w$ , for each  $y^i$  there is  $\eta^i > 0$  such that for each  $w$ ,  $|l(z, w, y^i) - l(z', w, y^i)| < \frac{\varepsilon}{2}$  as long as  $|z - z'| < \eta^i$ . Now remember that for any  $g$  we have recalled a uniform and almost sure convergence just above. As we have a denumerable number of functions, we have an almost sure convergence for all those functions together. Therefore we can consider with probability one converging sequences of  $X_j^i, x_j^i$ . For each  $g^i$ , there is  $M_i$  such that  $m_i \geq M_i$  implies  $\sup_{w \in W_h} \left| \frac{1}{m_i} \sum_{j=1}^{m_i} F^l(w, x_j^i) g^i(x_j^i) - E(F^l(w, X) g^i(X)) \right| < \eta_i$  for each  $l$ . Let us call  $M^n = \sup_{i \leq n} M_i$ . For  $m \geq M^n$ , we have for each  $i \leq n$  and for all  $(w_h, w_o)$ :

$$\left| l \left( \frac{1}{m} \sum_{j=1}^m F^1(w_h^1, x_j^1) g^1(x_j^1), \dots, \frac{1}{m} \sum_{j=1}^m F^k(w_h^k, x_j^k) g^k(x_j^k), y^i, w_o \right) - k(w_h, w_o, g^i, y^i) \right| < \frac{\varepsilon}{2}$$

This obviously implies that for all  $(w_h, w_o)$  :

$$\left| \lambda_n^m(w_h, w_o) - \frac{1}{n} \sum_{i=1}^n k(w_h, w_o, g^i, y^i) \right| < \frac{\varepsilon}{2},$$

where  $\lambda_n^m(w_h, w_o)$  is the corresponding realization of  $\widehat{\lambda}_n^m(w_h, w_o)$ . Combining this inequality with the first obtained in this final step, we obtain the following conclusion: if  $\varepsilon > 0$  is an arbitrary real number, there is almost surely  $N$  such that for each  $n \geq N$ , there is almost surely  $M^n$  such that for each  $m \geq M^n$ ,  $\sup_{(w_h, w_o) \in W_h \times W_o} |\lambda_n^m(w_h, w_o) - \lambda(w_h, w_o)| < \varepsilon$ . Therefore,

$$\lim_{n \rightarrow \infty} \lim_{m \rightarrow \infty} \sup_{(w_h, w_o) \in W_h \times W_o} \left| \widehat{\lambda}_n^m(w_h, w_o) - \lambda(w_h, w_o) \right| = 0 \text{ P a.s.}$$

The final conclusion of the theorem can be obtained exactly as for theorem 4 (see previous section).  $\square$

## References

- [1] Cédric Gégout, Bernard Girau, and Fabrice Rossi. A General Feed-Forward Neural Network Model. Technical report NC-TR-95-041, NeuroCOLT, Royal Holloway, University of London, May 1995. Available at <http://apiacoa.org/publications/1995/neurocolt1995.pdf>.
- [2] Cédric Gégout, Bernard Girau, and Fabrice Rossi. Generic Back-Propagation in Arbitrary Feedforward Neural Networks. In D. W. Pearson, N. C. Steele, and R. F. Albrecht, editors, *Int. Conf. on Artificial Neural Nets and Genetic Algorithms*, pages 168–171, Alès, April 1995. Springer Verlag.
- [3] Kurt Hornik. Approximation capabilities of multilayer feedforward networks. *Neural Networks*, 4(2):251–257, 1991.
- [4] Kurt Hornik. Some new results on neural network approximation. *Neural Networks*, 6(8):1069–1072, 1993.
- [5] Jim Ramsay and Bernard Silverman. *Functional Data Analysis*. Springer Series in Statistics. Springer Verlag, June 1997.
- [6] Walter Rudin. *Real and complex Analysis*. Mc Graw Hill, 1974.
- [7] Maxwell B. Stinchcombe. Neural network approximation of continuous functionals and continuous functions on compactifications. *Neural Networks*, 12(3):467–477, 1999.
- [8] Halbert White. Learning in Artificial Neural Networks: A Statistical Perspective. *Neural Computation*, 1(4):425–464, 1989.