ESANN'2002 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 24-26 April 2002, d-side publi., ISBN 2-930307-02-1, pp. 7-12

ESANN'2002 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 24-26 April 2002, d-side publi., ISBN 2-930307-02-1, pp. 7-12

Theoretical Properties of Functional Multi Layer Perceptrons

Fabrice Rossi*, Brieuc Conan-Guez† and François Fleuret†

∗ LISE/CEREMADE, UMR CNRS 7534, Université Paris-IX Dauphine,
Place du Maréchal de Lattre de Tassigny, 75016 Paris, France
† 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 numerical 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 consistent.

1 Introduction

It is quite common in current application to deal with high volume data that can be considered as functions. This is the case for instance for time series (which are mapping between a date and a value), weather data (which are time-varying geographical mappings), etc. Functional Data Analysis (FDA, see [4]) is an extension of traditional data analysis to this kind of functional data. In FDA, each individual is characterized by one or more real valued functions, rather than by a vector of $\mathbb{R}$. 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 real 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.

Most FDA methods share two common aspects: they are linear and they use basis expansion to represent functions with a finite number of parameters (for instance thanks to a spline based representation of each input function). In this paper, we propose a natural extension of Multi Layer Perceptrons (MLP) that enables to make non linear modeling of function data, without using a finite representation.

2 Functional Multi Layer Perceptrons

2.1 From numerical neurons to functional neurons

A n inputs MLP neuron is built thanks to a non linear activation function T (from $\mathbb{R}$ to $\mathbb{R}$), a numerical threshold b and connection weights, i.e. a vector $w \in \mathbb{R}^n$. Such a neuron maps an input vector $x \in \mathbb{R}^n$ to $T(b + wx)$, where wx is the scalar product of w and x, and can be seen as the image of x by the linear form defined by w.

As was already mentioned and studied in [7], there is no reason to limit neurons to finite dimensional inputs, as long as we consider w as a general linear form. Indeed, if E is a vectorial space and E^* its dual, we can define a neuron that maps elements of E to real numbers, thanks to an activation function T, a numerical threshold b and a "weight form", i.e. an element w of E^*. Such a neuron maps an input vector $x \in E$ to $T(b + w(x)) \in \mathbb{R}$.

This very general approach, which makes no assumption on the dimension of E, has been fruitful to prove very broad approximation results in [7]. The main drawback of the proposed model is that it relies on (approximate) manipulation of linear forms on an arbitrary vectorial space, which is not easy in general. That's why we specialize the model to functional spaces as follows.

We denote μ a finite positive Borel measure on $\mathbb{R}^n$ (the rational for using such a measure will be explained in section 4), and $L^p(\mu)$ the space of measurable functions from $\mathbb{R}^n$ to $\mathbb{R}$ such that $\int |f|^p d\mu < \infty$. Then we can define a neuron that maps elements of $L^p(\mu)$ to $\mathbb{R}$ thanks to an activation function T, a numerical threshold b and a weight function, i.e. a function $w \in L^q(\mu)$ (where q is the conjugate exponent of p). Such a neuron maps an input function g to $T(b + \int wg\, d\mu) \in \mathbb{R}$.

2.2 Functional MLP

As a functional neuron give a numerical output, we can define a functional MLP by combining numerical neurons with functional neurons. The first hidden layer of the network consists exclusively in functional neurons, whereas subsequent layers are constructed exclusively with numerical neurons. For instance an one hidden layer functional MLP with real output computes the following function:

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

where w_i are functions of the functional weight space.

3 Universal approximation

The practical usefulness of MLP is directly related to one of their most important properties: they are universal approximators (e.g. [3]). M. Stinchcombe has demonstrated in [7] that universal approximation is possible even if the input space of the MLP is an (almost) arbitrary vectorial space. Unfortunately, the proposed theorems use quite complex assumptions on linear form approximations. We propose here simplified versions that are directly applicable to FDA.

3.1 Theoretical results

Following [7], we introduce a definition:

Definition 1. If X is a topological vector space, A a set of functions from X to $\mathbb{R}$ 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 $p \in N$, $\beta_i \in \mathbb{R}$, $b_i \in \mathbb{R}$ and $l_i \in A$.

We have the following results:

Corollary 1. *Let μ be a finite positive Borel measure on $\mathbb{R}^n$. Let $1 < p \leq \infty$ be an arbitrary real number and q be the conjugate exponent of p. Let 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 $L^p(\mu)$ and $C(K, \mathbb{R})$ is the set of continuous functions from K to $\mathbb{R}$.*

Proof. We give here a sketch of the proof which can be found in [5]. For $p < \infty$, A_M is dense in $(L^p(\mu))^* = L^q(\mu)$ and therefore, corollary 5.1.3 of [7] applies (hypothesis on T allow to satisfy hypothesis of this corollary, thanks to [2]).

For $p = \infty$, we show that the set of functions defined on $L^\infty(\mu)$ by $l(f) = \alpha + \int fg\, d\mu$ where $g \in M$ separates points in K, thanks to approximation of elements of $L^1(\mu)$. Then, we apply theorem 5.1 of [7]. □

For $p = 1$, we must add an additional condition:

Corollary 2. *Let μ be a finite positive compactly supported Borel measure on $\mathbb{R}^n$. Let T be a measurable function from $\mathbb{R}$ to $\mathbb{R}$, 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 $L^1(\mu)$.*

Proof. We prove that A_M is dense in $(L^1(\mu))^* = L^\infty(\mu)$ for the weak-$*$ topology thanks to a two steps process: first we approximate a function in $L^\infty(\mu)$ by a compactly supported continuous function (thanks to Lusin theorem, e.g. [6]). Then we approximate this function on the support of the measure μ thanks to hypothesis on M. The conclusion is obtained thanks to corollary 5.1.3 of [7]. □

3.2 Practical consequences

Corollaries 1 and 2 show that as long as we can approximate functions in $L^q(\mu)$ or in $C(\mathbb{R}^n, \mathbb{R})$ (with elements of M), we can approximate continuous functions from a compact subset of $L^p(\mu)$.

On a practical point of view, M can be implemented by numerical MLPs. Indeed, [1] provides density results in $L^p(\mu)$ spaces ($p < \infty$) for MLP calculated functions, which is exactly what is needed for corollary 1. For corollary 2 we need universal approximation on compacta of continuous functions, which can again be done by numerical MLP, according to results of [2].

Corollaries 1 and 2 allows therefore to conclude that given a continuous function from a compact subset of a $L^p(\mu)$ space to $\mathbb{R}$ and a given precision, there is a functional MLP (constructed thanks to numerical MLP) that approximates the given function to the specified accuracy. The unexpected result is that the approximating MLP uses a finite number of numerical parameters, exactly as in the case of finite dimensional inputs.

4 Consistency

4.1 Probabilistic framework

In practical situations, input functions are not completely known but only through a finite set of input/output pairs, i.e., $(x_i, g(x_i))$. In general, the x_i are randomly chosen measurement points. To give a probabilistic meaning to this model, we assume given a probability space $\mathcal{P} = (\Omega, \mathcal{A}, P)$, on which is defined a sequence of sequences of independent identically distributed random variables, $(X_i^j)_{i\in\mathbb{N}}$ with value in Z, a metric space considered with its Borel sigma algebra. We call P_X the finite measure induced on Z by $X = X_1^1$ (this observation measure plays the role of μ in the universal approximation results). We assume define on $\mathcal{P}$ a sequence of independent identically distributed random elements $(G^j)_{j\in\mathbb{N}}$ with values in $L^p(P_X)$ and we denote $G = G^1$.

Let us now consider a one hidden layer functional perceptron that theoretically computes $H(a, b, w, g^j) = \sum_{i=1}^k a_i T\left(b_i + \int w_i g^j \, dP_X\right)$, where g^j is a realization of G^j, and each w_i belongs to $L^q(P_X)$. We replace this exact calculation which is not practically possible by the following approximation (a random variable) :

$$\widehat{H}(a, b, w, G^j)^m = \sum_{i=1}^{k} a_i T\left(b_i + \frac{1}{m}\sum_{l=1}^{m} w_i(X_l^j) G^j(X_l^j)\right) \tag{2}$$

In practical settings, we will compute a realization of $\widehat{H}^m$ for each input function realization associated with its evaluation points (which are themselves realization of the evaluation sequences).

4.2 Parametric approach

As explained in section 3.2, we propose to rely on numerical MLP for practical representation of weight functions. More generally, we can use parametric regressors, that is a easily computable function F from $W \times Z$ to $\mathbb{R}$, where W is a finite dimensional weight space (numerical MLPs are obviously a special case of parametric regressors when the number of parameters is fixed). With parametric regressors, equation 2 is replaced by:

$$\widehat{H}(a,b,w,G^j)^m = \sum_{i=1}^{k} a_i T\left(b_i + \frac{1}{m}\sum_{l=1}^{m} F_i(w_i, X_l^j)G^j(X_l^j)\right) \tag{3}$$

4.3 Consistency result

The parametric approach just proposed allows to tune a given functional MLP for a proposed task. In regression or discrimination problems, each studied function G^j is associated to a real value Y^j ($(Y^j)_{j\in\mathbb{N}}$ is a sequence of independent identically distributed random variables defined on $\mathcal{P}$ and we denote $Y = Y^1$). We want the MLP to approximate the mapping from G^i to Y^i and we measure the quality of this approximation thanks to a cost function, l. In order to simplify the presentation, we consider as in [8] that l models both the cost function and the calculation done by the functional MLP, so that it can be considered as a function from $L^p(P_X) \times \mathbb{R} \times W$, where W is a compact subset of $\mathbb{R}^q$ which corresponds to all numerical parameters used in the functional MLP (this includes parameters directly used by the functional MLP as well as parameters used by embedded parametric regressors).

The goal of MLP training is to minimize $\lambda(w) = E(l(G, Y, w))$. Unfortunately, it is not possible to calculate exactly λ which is replaced by the random variable $\widehat{\lambda}_n(w) = \frac{1}{n}\sum_{j=1}^{n} l(G^j, Y^j, w)$. In [8], H. White shows that for finite dimensional input spaces and under regularity assumptions on l, $\widehat{\lambda}_n$ converges almost surely uniformly on W to λ, which allows to conclude that estimated optimal parameters (i.e., solution to $\min_{w\in W}\widehat{\lambda}_n(w)$) converge almost surely to optimal parameters (i.e. to the set W^* of solution to $\min_{w\in W}\lambda(w)$). We have demonstrated in [5] that this result can be extended to infinite dimensional input spaces, i.e. to $L^p(P_X)$.

Unfortunately, we cannot directly rely on this result for practical situations, because we cannot compute exactly H, but rather $\widehat{H}^m$. We define therefore $\widehat{l}^m$ by exactly the same rational as l except that the exact output of the functional MLP is replaced by $\widehat{H}^m$. This allows to define $\widehat{\lambda}_n^m(w) = \frac{1}{n}\sum_{j=1}^{N}\widehat{l}^{m^j}(G^j, Y^j, w)$, where $m = \inf_{1\leq j\leq n} m_j$, and $\widehat{w}_n^m$ a solution to $\min_{w\in W}\widehat{\lambda}_n^m(w)$.

We show in [5] that under regularity assumptions on the cost function l and on the parametric regressors, $\lim_{n\to\infty}\lim_{m\to\infty} d(\widehat{w}_n^m, W^*) = 0$.

The practical meaning of this result is quite similar to the corresponding result for numerical MLP: we do not make systematic errors when we estimate optimal parameters for a functional MLP using a finite number of input functions known at a finite number of measurement points, because the estimated optimal parameters converge almost surely to the “true” optimal parameters.

5 Conclusion

We have proposed in this paper an extension of Multi Layer Perceptrons that allows processing of functional inputs. We have demonstrated that the proposed model is an universal approximator, as numerical MLP are. We have also demonstrated that despite a very limited knowledge (we have in general a finite number of example functions and a finite number of evaluation points for each function), it is possible to consistently estimate optimal parameters on available data (as in the case of numerical MLP).

Those theoretical results show that functional MLP share with numerical MLP their fundamental properties and that they can therefore be considered as a possible way to introduce non linear modeling in Functional Data Analysis. Further work is needed to assess their practical usefulness on real data and to compare them both with linear FDA methods and with basis expansion representation techniques used in FDA.

References

[1] Kurt Hornik. Approximation capabilities of multilayer feedforward networks. *Neural Networks*, 4(2):251–257, 1991.

[2] Kurt Hornik. Some new results on neural network approximation. *Neural Networks*, 6(8):1069–1072, 1993.

[3] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2:359–366, 1989.

[4] Jim Ramsay and Bernard Silverman. *Functional Data Analysis.* Springer Series in Statistics. Springer Verlag, June 1997.

[5] Fabrice Rossi, Brieuc Conan-Guez, and François Fleuret. Functional multi layer perceptrons. Technical report, LISE/CEREMADE & INRIA, 2001.

[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.