Modélisation supervisée de données fonctionnelles par perceptron multi-couches[0]

Fabrice Rossi[1]∗ **& Brieuc Conan-Guez**†

∗ *LISE/CEREMADE, UMR CNRS 7534, Université Paris-IX Dauphine,*
Place du Maréchal de Lattre de Tassigny,
75016 Paris, France
Fabrice.Rossi@dauphine.fr
† *INRIA-Rocquencourt, Domaine De Voluceau,*
BP 105 Bâtiment 18
78153 Le Chesnay Cedex, France
Brieuc.Conan-Guez@inria.fr

RÉSUMÉ. *Nous décrivons dans cet article une adaptation des perceptrons multi-couches aux données fonctionnelles. L'adaptation proposée permet une modélisation supervisée (discrimination et/ou régression) non linéaire de données fonctionnelles, sans représentation préalable de celles-ci sur une base. Nous montrons que le modèle proposé est un approximateur universel. De plus, l'estimation des paramètres optimaux d'un perceptron multi-couches fonctionnel paramétrique est consistante.*

MOTS-CLÉS : *Perceptrons multi-couches, Données fonctionnelles, Discrimination, Régression, Approximation universelle, Estimation paramétrique*

1. Introduction

Dans de nombreuses situations, il est naturel de décrire un individu par une ou plusieurs fonctions. Le cas le plus classique est celui d'un individu observé pendant un certain temps au cours duquel plusieurs mesures sont effectuées. On peut considérer par exemple plusieurs indices boursiers étudiés pendant une même période : chaque place boursière est observée par l'intermédiaire d'une fonction qui à une date associe la valeur de clôture des indices associés à cette place. Un autre exemple naturel est celui des mesures climatiques. Comme dans l'exemple boursier, une région peut être observée grâce à une fonction qui à une date associe des grandeurs comme la température moyenne de la journée correspondante, la quantité de précipitation, etc. Quand on dispose de données à haute résolution, on peut aussi prendre un point de vue géographique, la région étant alors décrite par une fonction qui à des coordonnées géographiques associe les grandeurs climatiques observées. On peut ainsi étudier la série temporelle des fonctions quotidiennes. [RAM 97] présente comment les méthodes classiques de l'analyse des données (Analyse en composantes principales, régression linéaire, etc.) ont été adaptées aux cas des données fonctionnelles. Parmi les méthodes non citées par cet ouvrage, on peut évoquer entre autres l'extension des approches de type nuées dynamiques ([ABR 01]) et les approches non-paramétriques basées sur des estimateurs à noyaux (utilisées en discrimination/régression, e.g., [FER 01] ou pour une modélisation auto-régressive, e.g., [BES 00]).

0. Conférence invitée publiée dans les actes des Neuvièmes journées de la Société Francophone de Classification.
Disponible à `http://apiacoa.org/publications/2002/sfc02.pdf`
1. Les coordonnées actuelles de Fabrice Rossi sont disponibles à l'URL `http://apiacoa.org/`

La principale limitation des méthodes proposées réside dans la représentation des fonctions manipulées. Il est clair que pour des données réelles, il est illusoire de compter sur une connaissance exacte des fonctions correspondant à l'observation d'un individu. En général, l'individu est connu par l'intermédiaire d'un nombre fini d'observations. Chaque observation est un couple entrée/sortie, comme par exemple une date de mesure associée à la valeur mesurée. Les méthodes de l'ADF doivent donc passer par une phase de modélisation des individus afin de transformer une suite finie d'observations en une fonction. Si on laisse de côté le cas particulier des situations dans lesquelles une connaissance experte du problème permet de proposer un modèle *a priori*, l'ADF travaille en général par régularisation : chaque individu est représenté par une (ou plusieurs) fonction régulière décrite par ses coordonnées dans une base fonctionnelle, en général des *splines* (les séries trigonométriques sont aussi utilisées, en particulier pour des données supposées périodiques). Ce mode de représentation est linéaire au sens où la fonction représentant un individu dépend linéairement des observations associées.

Or, on peut obtenir une représentation bien plus précise quand on utilise un modèle non linéaire. [BAR 93] montre en effet qu'à partir de la dimension trois (pour l'espace de départ des fonctions étudiées), une approximation par perceptron à une couche cachée est strictement plus efficace qu'une approximation par un modèle linéaire, au sens où il faut plus de coefficients numériques avec un modèle linéaire pour obtenir la même qualité d'approximation que celle obtenue par le perceptron. Pour la dimension deux, la qualité d'approximation est la même, alors que pour la dimension un, les modèles linéaires dominent. Cependant, il est illusoire de vouloir représenter *chaque* individu par un perceptron à une couche cachée. En effet, l'estimation d'un modèle non linéaire est un problème numérique bien plus délicat que celui d'un modèle linéaire, ce qui se traduit par un temps de calcul très important et difficilement acceptable dans la pratique. De plus, les bases utilisées pour les modèles linéaires sont en général orthonormées, ce qui accélère considérablement les calculs ultérieurs (une fois les individus représentés sur la base). Ce type d'accélération n'est pas envisageable pour un modèle neuronal. Dans la pratique, les individus ne peuvent donc pas vraiment être représentés par des modèles non linéaires.

Cependant, les méthodes d'ADF travaillent toutes par "comparaison" avec des modèles fonctionnels maîtrisés. L'ACP fonctionnelle réduit la dimension des données en trouvant des axes fonctionnels prépondérants, les méthodes à noyaux utilisent des noyaux fonctionnels, de même que les méthodes de type nuées dynamiques, etc. Or, les fonctions "modèles" sont elles aussi représentées par des modèles linéaires (ce qui permet l'accélération des calculs évoquée plus haut). Nous proposons dans cette communication un modèle neuronal permettant une modélisation non linéaire, basée sur la suppression de la phase de représentation des individus (qui seront donc manipulés directement sous forme d'une suite finie d'observations). Dans un perceptron multi-couches (PMC) fonctionnel, les neurones de la première couche utilisent des poids fonctionnels que nous représentons par des PMC numériques (classiques) ou par des modèles linéaires, en fonction de la dimension de l'espace de départ. De ce fait, les individus sont "comparés" avec des fonctions représentées par des modèles éventuellement non linéaires, mais en nombre limité, ce qui réduit les problèmes de temps de calcul évoqués plus haut. La souplesse du modèle permet de toujours utiliser la représentation la plus adaptée au problème.

L'article s'attache avant tout à décrire le modèle proposé et à donner les résultats théoriques qui autorisent son utilisation dans le cadre supervisé (discrimination et/ou régression). Nous montrons en particulier qu'on peut étendre deux propriétés très importantes des PMC numériques au cas fonctionnel. Les PMC fonctionnels sont en effet des approximateurs universels. De plus, pour une précision donnée et une fonction à approcher fixée, le PMC fonctionnel qui réalise l'approximation de la fonction utilise un nombre fini de paramètres numériques et peut donc être mis en œuvre informatiquement. Enfin, l'estimation des paramètres optimaux d'un PMC fonctionnel paramétrique à partir de données empiriques est consistante, ce qui confirme la possibilité d'une mise en œuvre effective du modèle.

2. Perceptron multi-couches fonctionnel

2.1. *Neurone fonctionnel*

Un neurone numérique (classique) à n entrées calcule une fonction N de $\mathbb{R}^n$ dans $\mathbb{R}$ définie par : $N(x) = T(w.x + b)$, où T est la fonction d'activation (de $\mathbb{R}$ dans $\mathbb{R}$), w le vecteur de poids synaptiques ($w \in \mathbb{R}^n$) et b le seuil (un réel).

L'extension au cas fonctionnel ne pose aucun problème, en remplaçant le produit scalaire $w.x$ dans $\mathbb{R}^n$ par son équivalent dans l'espace fonctionnel considéré, ce qui a été proposé dans [SAN 96a], [SAN 96b] et [STI 99]. Plus précisément, étant donné une mesure σ-finie μ définie sur un espace mesurable X, un neurone de $L^p(\mu)$ dans $\mathbb{R}$ calcule une fonction N définie par :

$$N(g) = T\left(b + \int fg \, \mathrm{d}\mu\right), \tag{1}$$

où T et b ont les mêmes définitions que pour un neurone numérique, et où f est une **fonction poids** (qui remplace donc le vecteur de poids du neurone numérique). Si $f \in L^q(\mu)$, où q est l'exposant conjugué de p, $N(g)$ est définie pour tout $g \in L^p(\mu)$. De façon plus générale, on peut considérer une partie de $L^p(\mu)$ (ou un autre espace fonctionnel défini sur X) et donc un ensemble plus général de fonctions poids acceptables. De même, il est parfaitement possible d'étendre le modèle proposé pour définir un neurone à plusieurs entrées fonctionnelles (cf [ROS 01]).

Notons que le modèle proposé est un cas particulier du modèle plus général dans lequel on remplace $w.x$ par $l(x)$, où l est une forme linéaire continue définie sur l'espace vectoriel normé d'entrée du neurone généralisé (cf [ROS 01]).

2.2. *PMC fonctionnel*

Comme le neurone fonctionnel proposé dans la section précédente produit une sortie réelle, la construction d'un PMC fonctionnel ne pose pas de problème : il suffit de commencer par une première couche constituée de neurones fonctionnels, puis d'utiliser exclusivement des neurones numériques dans les couches suivantes. Le cas le plus simple est celui d'un perceptron à une couche cachée, avec une entrée fonctionnelle et une sortie numérique. Un tel réseau calcule la fonction suivante :

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

2.3. *Mise en œuvre pratique*

Plusieurs problèmes se posent pour une réalisation informatique d'un PMC fonctionnel :

1. comme nous l'avons souligné en introduction, les fonctions qui décrivent les individus ne sont connues que par des suites finies de couples entrées/sorties ;
2. les fonctions poids doivent être représentées d'une façon informatiquement acceptable ;
3. le calcul des intégrales est délicat.

2.3.1. *Neurone paramétrique fonctionnel*

La représentation des fonctions poids est possible par diverses techniques. Nous nous focalisons dans cet article sur une approche paramétrique. Plus précisément, pour tout neurone fonctionnel, on suppose donné un espace de paramètres W (en général $\mathbb{R}^t$) et une fonction F de $W \times X$ dans $\mathbb{R}$. Le neurone fonctionnel basé sur F (et W) calcule la fonction :

$$N(w, g) = T\left(b + \int F(w, x) g(x) \, \mathrm{d}\mu(x)\right) \tag{3}$$

Le réglage du neurone fonctionnel se fait par l'intermédiaire du paramètre w. Dans la pratique, F peut être obtenue par diverses techniques comme par exemple un PMC numérique ou un modèle linéaire. Le modèle proposé est donc très souple puisqu'il permet d'utiliser la fonction la plus adaptée à la dimension de l'espace d'entrée (X) des fonctions étudiées.

2.3.2. *Modèle probabiliste*

Pour résoudre les autres problèmes, nous modélisons l'observation des individus de façon probabiliste. Plus précisément, l'individu représenté par la fonction g est connu par la suite finie de couples $(x_i, g(x_i))$. On suppose que les x_i sont des réalisations de la suite de variables aléatoires X_i indépendantes et identiquement distribuées, à valeurs dans X, et définies sur $(\Omega, \mathcal{A}, P)$. Dans ce modèle, μ est la mesure induite sur X par X_0.

Dans ce cas, par définition, $\int fg \,\mathrm{d}\mu = E(f(X_0)g(X_0))$. Cette dernière grandeur est finie (par hypothèse sur les fonctions poids du neurone fonctionnel) et la loi forte des grands nombres permet l'approximation suivante :

$$\lim_{m\to\infty} \frac{1}{m} \sum_{i=1}^{m} f(X_i)g(X_i) = \int fg \,\mathrm{d}\mu \quad P \text{ p.s.} \tag{4}$$

Si la fonction d'activation T d'un neurone fonctionnel est continue, on peut donc approcher la sortie de ce neurone par la variable aléatoire suivante :

$$\widehat{N}_m(g) = T\left(b + \frac{1}{m} \sum_{i=1}^{m} f(X_i)g(X_i)\right) \tag{5}$$

Par extension, on peut aussi approcher la sortie d'un PMC fonctionnel par une suite de variables aléatoires qui converge vers la sortie théorique. Dans la pratique, en combinant le modèle probabiliste avec l'approche paramétrique, on obtient la suite de variables aléatoires suivante (pour l'exemple du perceptron à une couche cachée donné par l'équation 2) :

$$\widehat{H}(g, w)_m = \sum_{i=1}^{k} a_i T\left(b_i + \frac{1}{m} \sum_{j=1}^{m} F_i(w_i, X_j)g(X_j)\right), \tag{6}$$

où $w = (a_1, \ldots, a_k, b_1, \ldots, b_k, w_1, \ldots, w_i)$.

Le modèle pratique obtenu est assez proche de celui proposé par [CHE 95], avec une différence cruciale : dans le modèle de [CHE 95], les points d'évaluation des fonctions ne sont pas déterminés par les données. L'existence de points d'évaluation vérifiant certaines propriétés est garantie par un théorème non constructif, ce qui limite considérablement la portée pratique de ce modèle, contrairement au notre.

Notons que comme annoncé en introduction, les individus sont traités directement, sans modélisation préalable par une fonction régulière.

3. Approximation universelle

3.1. *Introduction*

La puissance des PMC numériques est garantie par un ensemble de résultats d'approximation universelle qui établissent que toute fonction régulière peut être approchée arbitrairement bien par un PMC (cf par exemple [LES 93], [HOR 93] et [STI 99]). Dans la pratique, ces résultats permettent d'utiliser des PMC pour approcher n'importe quelle fonction d'un ensemble d'individus vers un autre ensemble, à des fins de régression non linéaire ou de discrimination (modélisation supervisée).

Des résultats similaires sont disponibles pour les PMC fonctionnels ([CHE 95], [SAN 96b] et [STI 99]), mais ils ne s'appliquent pas directement à notre modèle, soit parce qu'ils sont trop limités ([CHE 95]), soit, au contraire, parce qu'ils sont trop généraux et font des hypothèses presque

aussi complexes à vérifier que le théorème lui-même ([SAN 96b] et [STI 99]). C'est pourquoi nous proposons dans la section suivante deux corollaires adaptés à notre modèle.

3.2. *Notations et résultats*

On note $C(A, B)$ l'ensemble des fonctions continues de A dans B (qui sont deux espaces topologiques).

Définition 1. Soit X un espace vectoriel normé, A un sous-ensemble de X^* et T une fonction de $\mathbb{R}$ dans $\mathbb{R}$. On note $S_T^X(A)$ l'ensemble des fonctions de X dans $\mathbb{R}$ de la forme $h(x) = \sum_{i=1}^{k} a_i T(l_i(x) + b_i)$, où k est un entier positif, les a_i et les b_i sont des réels et les l_i sont des éléments de A.

$S_T^X(A)$ est en fait l'ensemble des fonctions exactement réalisables par un perceptron généralisé à une couche cachée dans lequel les neurones cachés sont les neurones généralisés évoqués dans la section 2.1. Pour obtenir un PMC fonctionnel, il suffit de considérer pour X un espace fonctionnel de type $L^p(\mu)$ et pour A les formes linéaires engendrées par les fonctions de $L^q(\mu)$, comme dans les corollaires suivants :

Corollaire 1. *Soit μ une mesure de Borel finie définie sur $\mathbb{R}^n$. Soit un réel $1 < p \leq \infty$ et son exposant conjugué q. Soit V un sous-espace dense de $L^q(\mu)$. Soit A_V l'ensemble des formes linéaires sur $L^p(\mu)$ construites grâce aux éléments de V, c'est-à-dire de la forme $l(f) = \int fg \, \mathrm{d}\mu$ pour un $g \in V$. Soit T une fonction de $\mathbb{R}$ dans $\mathbb{R}$, continue et non polynomiale.*

Alors $S_T^{L^p(\mu)}(A_V)$ est dense dans $C(K, \mathbb{R})$ pour tout compact K de $L^p(\mu)$.

Corollaire 2. *Soit μ une mesure de Borel finie à support compact définie sur $\mathbb{R}^n$. Soit V un sous-espace de $L^\infty(\mu)$ uniformément dense sur les compacts dans $C(\mathbb{R}^n, \mathbb{R})$. Soit T une fonction de $\mathbb{R}$ dans $\mathbb{R}$, continue et non polynomiale.*

Alors $S_T^{L^1(\mu)}(A_V)$ est dense dans $C(K, \mathbb{R})$ pour tout compact K de $L^1(\mu)$.

Pour une démonstration de versions légèrement plus générales de ces corollaires, voir [ROS 01] et [ROS 02b].

3.3. *Discussion*

Les corollaires proposés montrent que si les fonctions poids sont bien représentées (i.e., sont choisies dans des ensembles V suffisamment "gros"), un perceptron fonctionnel à une couche caché peut approcher arbitrairement bien une fonction continue d'un compact de l'espace fonctionnel considéré dans $\mathbb{R}$. On obtient donc une généralisation au cas fonctionnel des résultats connus pour le cas numérique. Quelques remarques s'imposent :

- les résultats de [HOR 91] montrent qu'on peut prendre pour V (dans le cas du corollaire 1) l'ensemble des fonctions exactement calculable par un perceptron numérique à une couche cachée (en prenant par exemple une fonction d'activation continue et non polynomiale). Or, si on fixe une précision d'approximation, le PMC fonctionnel fourni par le corollaire 1 utilise bien entendu un nombre fini de neurones. Comme chaque neurone est paramétré par deux coefficients numériques et une fonction poids, et que cette fonction poids est elle-même *exactement* calculée par un PMC numérique, la fonction de $K \subset L^p(\mu)$ dans $\mathbb{R}$ est approchée grâce à un nombre *fini* de paramètres numériques. On peut donc espérer pouvoir réaliser dans la pratique cette forme d'approximation ;
- on peut montrer que l'hypothèse $1 < p$ du corollaire 1 peut se relâcher en $1 \leq p$. Mais comme l'indique [STI 99], il est très difficile de construire un sous-ensemble dense de $L^\infty(\mu)$. En particulier, si μ n'a pas un support compact, aucun ensemble de fonctions calculées par PMC n'est dense dans $L^\infty(\mu)$. Ceci explique le recours au corollaire 2. Comme pour le corollaire 1, on peut prendre pour V des fonctions calculées par un perceptron numérique à une couche cachée (cf [HOR 93]) ;

– les résultats obtenus s'appliquent au calcul exact réalisé par le PMC fonctionnel. Dans la pratique, on approche *en chaque point* ce calcul par la réalisation d'une variable aléatoire $\widehat{H}(g,w)_m$. La quantité de données (i.e., la valeur de m) nécessaire à une bonne approximation en g dépend de g.

4. Consistance

4.1. *Résultat*

Dans la pratique, nous estimons les paramètres optimaux d'un PMC fonctionnel paramétrique à partir de données doublement incomplètes. Nous travaillons en effet à partir d'un ensemble fini de fonctions (i.e., d'individus). De plus, chaque fonction est connue grâce à un nombre fini de couples entrée/sortie. L'estimation reste cependant consistante, comme le montre le théorème suivant :

Theorem 1. *Soit X_j^i une suite de variables aléatoires indépendantes et identiquement distribuées, définies sur $(\Omega, \mathcal{A}, P)$ et à valeur dans Z, un compact de $\mathbb{R}^s$. On note μ la probabilité induite sur Z par X_0^0 (on note X cette dernière v.a.).*

Soit K un compact de l'espace $C(Z,\mathbb{R})$ (muni de la norme infinie). Soit (G^i, Y^i) une suite de couples de variables aléatoires indépendantes et identiquement distribuées, définies sur $(\Omega, \mathcal{A}, P)$ et à valeur dans $K \times \mathbb{R}^t$. On note $Y = Y^0$.

Soit $k > 0$ un entier, $q \geq 1$ un réel, et $F^1, \ldots, F^k$, k fonctions telles que :

1. F^l est une fonction de $W^l \times Z$ dans $\mathbb{R}$

2. W^l est un espace métrique compact

3. pour tout $x \in Z$, $F^l(.,x)$ est continue

4. pour tout $w_h^l \in W^l$, $F^l(w_h^l,.)$ est mesurable

5. il existe une fonction mesurable d^l de Z dans $\mathbb{R}$, élément de $L^q(\mu)$, telle que pour tout $w_h^l \in W^l$ et tout $x \in Z$, $|F^l(w_h^l, x)| \leq d^l(x)$

Soit enfin l, une fonction de $\mathbb{R}^k \times \mathbb{R}^t \times W_o$ dans $\mathbb{R}$ (où W_o est un espace métrique compact), telle que :

1. pour tout $y \in \mathbb{R}^t$, $l(.,y,.)$ est uniformément continue

2. pour tout $w_o \in W_o$, $l(.,.,w_o)$ est mesurable

3. il existe une fonction mesurable d' de $\mathbb{R}^t$ dans $\mathbb{R}$ telle que $|l(x,y,w_o)| < d'(y)$ pour tout x et tout w_o

4. $E(d'(Y)) < \infty$

On définit alors pour tout $\omega \in \Omega$:

$$\lambda_n^m(w_h, w_o)(\omega) = \frac{1}{n}\sum_{i=1}^{n} l\left(\frac{1}{m_i}\sum_{j=1}^{m_i} F^1\left(w_h^1, X_j^i(\omega)\right) G^i(\omega)\left(X_j^i(\omega)\right), \ldots, \frac{1}{m_i}\sum_{j=1}^{m_i} F^k\left(w_h^k, X_j^i(\omega)\right) G^i(\omega)\left(X_j^i(\omega)\right), Y^i(\omega), w_o\right), \quad (7)$$

où m est défini par $m = \inf_{1\leq i\leq n} m_i$; et :

$$\lambda(w_h, w_o) = E\left(l\left(\int F^1\left(w_h^1, x\right) G(x) d\mu(x), \ldots, \int F^k\left(w_h^k, x\right) G(x) d\mu(x), Y, w_o\right)\right) \quad (8)$$

Alors pour tout couple (n,m) et tout ω la fonction $\lambda_n^m(\omega)$ possède un minimum qu'on note

$$w_n^m(\omega) = \arg \min_{w \in W_h \times W_o} \lambda_n^m(w_h, w_o)(\omega)$$

Soit W^* l'ensemble des points qui réalisent le minimum de $\lambda(w_h, w_o)$. Pour P presque tout ω, on a :

$$\lim_{n\to\infty} \lim_{m\to\infty} d(w_n^m(\omega), W^*) = 0 \tag{9}$$

Pour une démonstration d'une version plus générale de ce théorème, voir [ROS 01].

4.2. *Discussion*

Le résultat énoncé est assez technique. On peut l'interpréter en pratique grâce aux remarques suivantes :

- les individus sont modélisés grâce à deux suites de variables aléatoires. La suite G^i correspond aux individus eux-mêmes et donc à des variables aléatoires fonctionnelles. Pour chaque individu (i.e., chaque G^i), on dispose d'une suite d'observations, les X_j^i, comme indiqué à la section 2.3.2. Enfin, comme nous sommes dans un modèle supervisé, chaque individu est associé à une "cible", Y^i, encore une fois modélisée par une suite de v.a. ;
- les F^l correspondent à la représentation paramétrique des fonctions poids, comme indiqué à la section 2.3.1 ;
- la partie la plus éloignée du modèle de PMC fonctionnel proposé dans l'article est la fonction l. En fait, cette fonction est la composée de la fonction de coût qui mesure la qualité des paramètres du PMC (c'est pourquoi on utilise Y, la cible, dans la définition de $\lambda(w_h, w_o)$) et de la "fin" du PMC fonctionnel étudié. Cette fin correspond à toute la partie numérique du PMC fonctionnel. Le paramètre w_o regroupe tous les paramètres numériques. Si on considère le cas d'un perceptron fonctionnel à une couche cachée, avec une erreur quadratique, on peut définir l de la façon suivante :

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

  avec $w_o = (a_1, b_1, \ldots, a_k, b_k) \in \mathbb{R}^{2k}$. L'idée est donc de séparer le calcul fonctionnel, qui introduit une approximation (une moyenne remplace une intégrale), du calcul numérique qui est exact ;
- $\lambda(w_h, w_o)$ correspond à l'erreur théorique effectuée par le modèle, dans le cas de calculs exacts et d'une connaissance parfaite ;
- $\lambda_n^m(w_h, w_o)(\omega)$ correspond à l'erreur empirique, obtenue avec des données limitées et un calcul approximatif.

Au final, le théorème indique simplement que le calcul des paramètres optimaux d'un PMC fonctionnel à partir de l'erreur empirique (évaluée grâce au calcul approché de la sortie du PMC) est consistant, au sens où ces paramètres convergent presque sûrement vers les paramètres optimaux quand la connaissance des données augmente. La seule limitation du résultat est que la convergence proposée est séquentielle : pour une précision souhaitée pour les paramètres optimaux, le nombre de points d'évaluation nécessaires dépend du nombre d'individus considérés.

5. Conclusion

Le modèle de PMC fonctionnel proposé dans cet article est intéressant pour diverses raisons :

- le modèle partage avec les PMC numériques des propriétés importantes (approximation universelle et consistance) qui justifient son utilisation dans la pratique pour le traitement supervisé des données fonctionnelles, au même titre que les PMC numériques pour les données classiques ;
- le modèle ne nécessite pas de modélisation préalable des individus et ne demande que des hypothèses très faibles sur les fonctions sous-jacentes (qui peuvent en particulier ne pas être régulières). Il est donc plus général que beaucoup d'autres approches classiques en ADF ;
- la représentation des fonctions poids est très souple et peut donc être adaptée à la dimension de l'espace d'entrée des fonctions traitées (modèle non linéaire en grande dimension, linéaire en petite dimension).

Dans [ROS 02a], nous avons comparé, sur des données synthétiques, les PMC numériques et les PMC fonctionnels. Nous avons montré que l'approche fonctionnelle non linéaire permet une représentation plus économique (en terme de nombre de paramètres) que l'approche numérique.

La manipulation de données fonctionnelles par PMC n'implique cependant pas automatiquement l'utilisation de l'approche directe proposée dans le présent article. Dans [CON 02], nous proposons une approche plus classique qui se base sur la représentation des individus grâce à un modèle linéaire. Ce nouveau modèle possède les mêmes propriétés théoriques que celui du présent article (approximation universelle et estimation consistante). Comme annoncé en introduction, le présent modèle est plus souple que celui de [CON 02], ce qui permet de l'adapter à une dimension d'espace d'entrée élevée. Par contre, les temps de calcul induit par le présent modèle sont bien plus élevés que ceux du modèle de [CON 02]. Nous travaillons actuellement sur une applications des différents modèles à des données réelles pour mieux comprendre les avantages et inconvénients de chacun.

6. Bibliographie

[ABR 01] Abraham C., Cornillon P.-A., Matzner-Lober E., Molinari N., Unsupervised Curve Clustering using B-Splines, rapport n°00–04, October 2001, ENSAM–INRA–UM II–Montpellier.

[BAR 93] Barron A. R., Universal Approximation Bounds for Superpositions of a Sigmoidal Function, *IEEE Trans. Information Theory*, vol. 39, n° 3, 1993, p. 930-945,

[BES 00] Besse P., Cardot H., Stephenson D., Autoregressive forecasting of some functional climatic variations, *Scandinavian Journal of Statistics*, vol. 4, 2000, p. 673-688.

[CHE 95] Chen T., Chen H., Universal Approximation to Nonlinear Operators by Neural Networks with Arbitrary Activation Functions and Its Application to Dynamical Systems, *IEEE Transactions on Neural Networks*, vol. 6, n° 4, 1995, p. 911–917.

[CON 02] Conan-Guez B., Rossi F., Approche Régularisée du Traitement de Données Fonctionnelles par un Perceptron Multi-Couches, *Actes des neuvièmes journées de la SFC*, Toulouse, France, Septembre 2002, p. 169–172.

[FER 01] Ferraty F., Vieu P., Statistique Fonctionnelle : Modèles de régression pour variables aléatoires uni, multi et infiniment dimensionnées, rapport n°LSP-2001-03, 2001, Laboratoire de Statistique et Probabilités, Université Paul Sabatier, Toulouse, France.

[HOR 91] Hornik K., Approximation Capabilities of multilayer feedforward networks, *Neural Networks*, vol. 4, n° 2, 1991, p. 251-257.

[HOR 93] Hornik K., Some new results on neural network approximation, *Neural Networks*, vol. 6, n° 8, 1993, p. 1069-1072.

[LES 93] Leshno M., Lin V. Y., Pinkus A., Schocken S., Multilayer Feedforward Networks With a Nonpolynomial Activation Function Can Approximate Any Function, *Neural Networks*, vol. 6, n° 6, 1993, p. 861–867.

[RAM 97] Ramsay J., Silverman B., *Functional Data Analysis*, Springer Series in Statistics, Springer Verlag, June 1997.

[ROS 01] Rossi F., Conan-Guez B., Fleuret F., Functional Multi Layer Perceptrons, rapport n°0134, december 2001, LISE/CEREMADE & INRIA, `http://www.ceremade.dauphine.fr/`.

[ROS 02a] Rossi F., Conan-Guez B., Fleuret F., Functional Data Analysis With Multi Layer Perceptrons, *Proceedings of IJCNN 2002 (WCCI 2002)*, vol. 3, Honolulu, Hawai, USA, May 2002, IEEE/NNS/INNS, p. 2843–2848.

[ROS 02b] Rossi F., Conan-Guez B., Fleuret F., Theoretical Properties of Functional Multi Layer Perceptrons, *Proceedings of ESANN 2002*, Bruges, Belgium, April 2002, p. 7–12.

[SAN 96a] Sandberg I. W., Notes on Weighted Norms and Network Approximation of Functionals, *IEEE Transactions on Circuits and Systems–I : Fundamental Theory and Applications*, vol. 43, n° 7, 1996, p. 600–601.

[SAN 96b] Sandberg I. W., Xu L., Network approximation of input-output maps and functionals, *Circuits Systems Signal Processing*, vol. 15, n° 6, 1996, p. 711–725.

[STI 99] Stinchcombe M. B., Neural network approximation of continuous functionals and continuous functions on compactifications, *Neural Networks*, vol. 12, n° 3, 1999, p. 467-477.