
Un modèle semi-paramétrique neuronal pour la régression et la discrimination sur données fonctionnelles⁰

Fabrice Rossi* & Briec 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
Briec.Conan-Guez@inria.fr

RÉSUMÉ. L'objectif de cet article est de présenter un nouveau modèle neuronal adapté à la régression sur variables explicatives fonctionnelles. Nous décrivons le perceptron multi-couches (PMC) fonctionnel et résumons les propriétés théoriques de celui-ci, en terme de puissance de calcul et d'estimation des paramètres. Nous proposons ensuite deux techniques de mise en œuvre informatique que nous illustrons dans une application en spectrométrie. Le PMC fonctionnel est ainsi comparé au PMC classique ainsi qu'à des méthodes plus classiques adaptées elles aussi aux données fonctionnelles. Dans l'exemple choisi, il apparaît clairement que les PMC fonctionnels obtiennent de meilleures performances que les autres techniques et représentent la fonction de régression de façon très parcimonieuse.

ABSTRACT. We introduce in the present article a new neural model adapted to regression problems in which explanatory variables are functions. The paper describes the functional Multi-Layer Peceptron (MLP) and summarizes its main theoretical properties: universal approximation and consistent parameter estimation. We propose two possible numerical implementations of the model and show how to apply them to a spectrometric problem. This application example shows that functional MLP achieve better performances than other techniques while using a very low number of numerical parameters.

MOTS-CLÉS : Analyse de données fonctionnelles, Réseaux de neurones, Perceptrons Multi-Couches, Régression, Discrimination, Approximation universelle, Consistance, Spectrométrie.

KEYWORDS: Functional Data Analysis, Neural Networks, Multi-Layer Perceptrons, Regression, Discrimination, Universal approximation, Consistency, Spectrometry.

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

0. CEREMADE preprint 0338 (2003) disponible sur <http://www.ceremade.dauphine.fr>

qui à une date associe la valeur de clôture des indices de cette place. Un autre exemple naturel est celui des mesures climatiques. Comme dans l'exemple boursier, une région peut être représentée par 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. Nous présenterons dans le présent article un autre exemple, l'étude du spectre d'absorbance d'échantillons de viande. Chaque individu est alors décrit par le spectre d'absorbance obtenu par un instrument travaillant sur 100 fréquences dans le proche infra-rouge.

[RAM 97] présente comment les méthodes classiques de l'analyse de données (Analyse en composantes principales, régression linéaire, etc.) ont été adaptées au cas des données fonctionnelles. Parmi les méthodes non citées par cet ouvrage car plus récentes, on peut évoquer entre autres l'extension des approches de type nuées dynamiques ([ABR 03]), la régression inverse par tranches ([FER 00, DAU 01]) et les approches non-paramétriques basées sur des estimateurs à noyaux (utilisées en discrimination/régression, e.g., [FER 01], [FER 02b] et [FER 03] ou pour une modélisation auto-régressive, e.g., [BES 00]). On pourra aussi se reporter à [BES 03] pour une vision générale récente. Nous nous intéressons ici à un cas particulier de modélisation de données fonctionnelles, celui dans lequel nous avons une variable à expliquer. Plus précisément, nous supposons que les individus décrits par une ou plusieurs variables fonctionnelles sont associés à une grandeur à prédire qui peut être une appartenance à une classe ou une valeur numérique au sens large (les vecteurs de \mathbb{R}^n peuvent être traités). Il s'agit donc soit de pratiquer de la discrimination de fonctions (ou de n -uplets de fonctions), soit de faire de la régression multiple sur variables explicatives fonctionnelles. Nous excluons ainsi de notre champs le problème de la classification (cf [ABR 03]).

Dans ce cadre, nous proposons dans le présent article un nouvel outil pour le traitement des données fonctionnelles : le perceptron multi-couches (PMC) fonctionnel. Cet outil est une adaptation au cadre fonctionnel des perceptrons multi-couches définis pour les données numériques classiques. Les avantages des PMC classiques sont connus : ils sont capables d'approcher arbitrairement bien n'importe quelle fonction régulière (cf par exemple [LES 93], [HOR 93] et [STI 99]) de façon parcimonieuse ([BAR 93]), c'est-à-dire en utilisant peu de paramètres numériques (comparativement à d'autres méthodes), ce qui permet d'envisager leur utilisation même quand le nombre de variables explicatives est important. En pratique, les PMC donnent de meilleurs résultats que les méthodes linéaires (cf par exemple [ZHU 97, BAR 97]). De plus les techniques d'estimation des paramètres des PMC numériques sont statistiquement valides (cf [WHI 89] pour une preuve de consistance des estimateurs).

Nous verrons dans la section 2 que le passage à l'aspect fonctionnel est obtenu assez simplement par une généralisation du calcul réalisé par les neurones de la première couche du réseau. Nous verrons comment implémenter en pratique le modèle proposé, au moyen de deux solutions. La première technique proposée est une approche directe assez souple et originale dans le cadre de l'analyse de données fonctionnelles (ADF), permettant l'utilisation de représentations non linéaires de certaines fonctions manipulées. La deuxième technique, plus classique en ADF consiste à utiliser une projection sur une base de splines (ou plus généralement sur une base topologique de l'espace fonctionnel considéré) pour manipuler les fonctions.

Nous verrons dans la section 3 que les PMC fonctionnels possèdent la propriété d'approximation universelle et que l'estimation de leurs paramètres est consistante. Ils partagent ainsi avec les PMC numériques des propriétés fondamentales qui justifient leur utilisation.

Nous concluons cet article par une application des PMC fonctionnels à des données spectro-métriques, présentée dans la section 4. Nous verrons que les PMC fonctionnels semblent très par-

cimonieux comparés aux approches non-paramétriques concurrentes, tout en offrant une meilleure qualité de modélisation.

2. Le perceptron multi-couches fonctionnel

2.1. Un PMC numérique

Rappelons tout d'abord la définition d'un perceptron multi-couches numérique (nous renvoyons le lecteur entre autres à [ZHU 97, BAR 97] pour une présentation plus détaillée). Il s'agit d'une classe de modèles paramétriques obtenue par combinaison de modèles élémentaires non linéaires. Plus précisément, on définit un neurone numérique à n entrées comme un modèle paramétrique à $n + 1$ paramètres réels représenté par la fonction de \mathbb{R}^n dans \mathbb{R} définie par : $N(x) = T(w.x + b)$, où T est la fonction d'activation (une fonction de \mathbb{R} dans \mathbb{R}), w le vecteur de poids synaptiques ($w \in \mathbb{R}^n$) et b le seuil (un réel), le vecteur (w, b) constituant les paramètres du modèle. La fonction T est choisie non linéaire et relativement régulière afin d'assurer la propriété d'approximation universelle pour le PMC complet (cf [HOR 93] pour des conditions minimales sur T).

Le PMC est obtenu en combinant les neurones. La première étape est la combinaison en couches qui consiste à utiliser p neurones à n entrées en parallèle afin de construire un modèle paramétrique à $(n + 1)p$ paramètres qui transforme un vecteur de \mathbb{R}^n en un vecteur de \mathbb{R}^p selon l'équation $H(x) = y$ avec $y_i = T(w_i.x + b_i)$, où (w_i, b_i) désigne les paramètres du neurone i . La deuxième étape consiste à combiner les couches en composant les fonctions réalisées par celles-ci. Si on ajoute par exemple une deuxième couche à q neurones, on obtient globalement un modèle paramétrique à $(n + 1)p + (p + 1)q$ paramètres représenté par la fonction de \mathbb{R}^n dans \mathbb{R}^q définie par $G(x) = z$ avec $z_j = T_2(v_j.y + c_j)$ et $y_i = T(w_i.x + b_i)$. Dans cette expression, T_2 désigne la fonction d'activation de la deuxième couche, alors que les (v_j, c_j) sont les paramètres des neurones de cette couche. On règle la puissance du PMC (c'est-à-dire ses capacités d'approximation) grâce au nombre de couches et au nombre de neurones par couche. Le nombre d'entrées de la première couche correspond au nombre de variables explicatives (à valeurs réelles), alors que le nombre de sortie de la dernière couche correspond au nombre de variables à expliquer (toujours à valeurs réelles).

2.2. Un PMC fonctionnel

L'extension du neurone numérique 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 donnée une mesure σ -finie μ définie sur un espace mesurable X , un neurone de $L^p(\mu)$ dans \mathbb{R} calcule une fonction N donnée par :

$$N(g) = T \left(b + \int fg d\mu \right), \quad (1)$$

où T et b ont les mêmes définitions que pour un neurone numérique, et où f est une **fonction de 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 de poids acceptables. De même, il est parfaitement possible d'étendre le modèle proposé pour définir un neurone à plusieurs entrées fonctionnelles (cf [CON 02a, ROS 03c]) ou à des entrées mixtes.

Comme le neurone fonctionnel proposé ci-dessus 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 deux couches (la première contient ici s neurones et la deuxième un seul), avec une entrée fonctionnelle et une sortie numérique. Un tel réseau calcule la fonction suivante :

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

2.3. Mises en œuvre pratiques

La mise en œuvre informatique du PMC fonctionnel est délicate en raison de trois problèmes.

Tout d'abord, les paramètres du modèle proposé ne sont pas numériques. Or, la manipulation informatique de fonctions est difficile et on doit se contenter de combinaisons de fonctions élémentaires ou de fonctions discrétisées.

De plus, le neurone fonctionnel s'appuie sur un calcul d'intégrales dont la réalisation informatique peut poser des problèmes.

Enfin, il est très rare de disposer de données sous une forme fonctionnelle au sens mathématique du terme (par exemple une définition en terme de fonctions connues). En général, on dispose au contraire pour chaque fonction considérée d'une discrétisation de celle-ci. Ainsi la fonction g est-elle connue par l'intermédiaire d'un m -uplets de couples entrée/sortie, les $(x_j, y_j)_{1 \leq j \leq m}$. Pour prendre en compte le bruit d'observation, on suppose que $y_j = g(x_j) + \epsilon_j$, où ϵ_j désigne un bruit centré indépendant des x_j . On suppose que les x_j sont choisis de façon aléatoire, ce qui constitue une extension importante par rapport au modèle déterministe utilisé en général en ADF.

Nous devons donc trouver une méthode permettant d'adapter le modèle théorique proposé dans la section précédente aux contraintes qui viennent d'être présentées. Dans la suite de cette section, nous traitons la représentation des fonctions de poids puis nous détaillons deux solutions possibles aux problèmes de la représentation des données et du calcul des intégrales. Nous nous contenterons d'étudier le calcul effectué par un neurone fonctionnel, le calcul réalisé par le PMC en découlant naturellement.

2.3.1. Fonctions de poids

Les fonctions de poids sont représentées naturellement grâce à un modèle paramétrique quelconque. Le modèle peut être linéaire (par rapport à ses paramètres) ce qui revient à écrire les fonctions de poids sous la forme :

$$f(x) = \sum_{i=1}^k w_i \phi_i(x), \quad (3)$$

où les $(\phi_i)_{1 \leq i \leq k}$ sont des fonctions faciles à calculer informatiquement. On peut par exemple choisir pour les ϕ_i une base de B-splines, d'ondelettes ou de fonctions trigonométriques. Dans tous les cas, on remplace le paramètre fonctionnel du neurone par un vecteur de paramètres numériques.

On peut aussi utiliser un modèle non linéaire comme par exemple un PMC numérique. On écrit alors les fonctions de poids sous la forme :

$$f(x) = F(w, x), \quad (4)$$

où w désigne le vecteur de paramètres du modèle non linéaire choisi.

2.3.2. Approche directe

L'implémentation la plus simple du modèle théorique consiste à approcher les intégrales par des moyennes. On considère en effet $\int f g d\mu$ avec pour μ une mesure σ -finie. Un choix pratique pour

μ est P_X , la mesure de probabilité associée au choix des x_j , les points d'évaluation des fonctions. Grâce à la loi des grands nombres, on peut approcher $\int fgdP_X$ par $\frac{1}{m} \sum_{j=1}^m y_j f(x_j)$.

Considérons une fonction de poids représentée grâce à un modèle général $F(w, x)$. Un neurone fonctionnel basé sur cette représentation peut être implémenté informatiquement en associant à la liste de m couples d'observations $(x_j, y_j)_{1 \leq j \leq m}$ la valeur numérique suivante :

$$T \left(b + \frac{1}{m} \sum_{j=1}^m y_j F(w, x_j) \right)$$

Intuitivement, on a donc adapté le neurone numérique en lui permettant de traiter de façon consistante des listes d'observations de taille variable. La nature de F intervient dans l'optimisation des calculs. En effet, si F est non linéaire (par rapport à w), on ne peut pas simplifier le calcul de l'expression. Par contre, si F correspond à un modèle linéaire en w , de la forme $F(w, x) = \sum_{i=1}^k w_i \phi_i(x)$, on peut calculer une fois pour toutes les grandeurs suivantes (au nombre de k) :

$$\gamma_i = \frac{1}{m} \sum_{j=1}^m y_j \phi_i(x_j)$$

Le neurone calcule alors :

$$T \left(b + \sum_{i=1}^k w_i \gamma_i \right)$$

En général, k est petit devant m et cette technique permet donc de réduire considérablement le temps de calcul. De plus, k est une constante du modèle, alors que m dépend de la fonction. Avec ce mode de calcul, on peut donc se ramener à un nombre constant d'entrées dans le modèle neuronal. Il est ainsi possible de soumettre les données pré-traitées à une implémentation informatique classique des PMC.

Outre sa simplicité, l'intérêt de cette approche est la possibilité d'utiliser pour F un modèle non linéaire. Or, on sait qu'un modèle de type PMC classique permet de représenter de façon très parcimonieuse des fonctions de \mathbb{R}^k dans \mathbb{R} . Plus précisément, on montre (cf [BAR 93]) que pour $k \geq 3$, un PMC utilise strictement moins de paramètres numériques qu'un modèle linéaire en w pour approcher une fonction donnée (sous certaines conditions de régularité sur la fonction). Plus k est grand, plus la différence est importante. Quand on travaille avec des observations fonctionnelles dont les points de discrétisation appartiennent à un espace de dimension élevée ($k \geq 3$), on peut donc espérer obtenir un modèle plus parcimonieux en utilisant un F non linéaire. L'intérêt pratique est de limiter l'espace mémoire de stockage du modèle et d'accélérer les traitements (une fois l'estimation terminée). Le défaut évident de cette solution réside dans l'estimation des paramètres qui n'est plus un problème quadratique et qui nécessite donc des algorithmes d'optimisation complexes et coûteux.

2.3.3. Approche par projection

En ADF, les fonctions observées sont en général manipulées par l'intermédiaire de représentants réguliers (cf [RAM 97]). Plus précisément, on utilise les observations d'une fonction pour déterminer une approximation de celle-ci simple à manipuler. Même s'il est possible en théorie d'utiliser une méthode d'approximation non linéaire (comme un PMC classique), cela n'a pas beaucoup d'intérêts pratiques. Tout d'abord, estimer un modèle non linéaire est coûteux en temps de calcul. De plus, la représentation obtenue est en général plus difficile à manipuler qu'une représentation pseudo-linéaire. C'est pourquoi les méthodes classiques de l'ADF travaillent en général par projection des fonctions observées sur une base fonctionnelle tronquée, par exemple une base de B-splines, une base d'ondelettes ou encore une base de fonctions trigonométriques. La seule contrainte induite par cette technique est qu'on doit se restreindre à des fonctions dans $L^2(\mu)$.

Étant donnée g , on cherche donc les coefficients $\alpha_i(g)$ qui minimisent :

$$\left\| g - \sum_{i=1}^l \alpha_i(g) \psi_i \right\|^2$$

Comme dans la section précédente, on approche ce problème en posant $\mu = P_X$ et en utilisant la loi des grands nombres pour dire que la norme proposée ci-dessus s'approche par :

$$\frac{1}{m} \sum_{j=1}^m \left(y_j - \sum_{i=1}^l \alpha_i(g) \psi_i(x_j) \right)^2$$

On remplace donc la liste de m couples d'observations $(x_j, y_j)_{1 \leq j \leq m}$ par la fonction $\sum_{i=1}^l \alpha_i(g) \psi_i$. Cette méthode ne résout qu'en partie les problèmes d'implémentation du neurone fonctionnel. Il reste en effet à calculer l'intégrale $\int F(w, x) (\sum_{i=1}^l \alpha_i(g) \psi_i(x)) dP_X(x)$. Par linéarité, il suffit de calculer les $\int F(w, x) \psi_i(x) dP_X(x)$.

L'emploi d'une technique de type Monte Carlo est envisageable pour ce calcul, puisqu'on connaît des listes de points d'observations (les x_i) qui sont par hypothèse distribués selon P_X . Dans cette optique, le seul intérêt de l'étape de projection est la régularisation que celle-ci peut apporter (ce qui permet de lutter contre le bruit d'observations, les ϵ_i). Une autre solution est de choisir pour F un modèle pseudo-linéaire de la forme $F(w, x) = \sum_{i=1}^k w_i \phi_i(x)$. On est ainsi ramené aux calculs des $\int \psi_j \phi_i dP_X$. Ces intégrales se calculent une fois pour toute (par une méthode de Monte Carlo pour respecter la mesure d'intégration P_X). Le neurone calcule alors :

$$T \left(b + \sum_{i=1}^k w_i \left(\sum_{j=1}^l \alpha_j(g) \int \psi_j \phi_i dP_X \right) \right)$$

L'avantage par rapport au cas de F quelconque est double, comme dans la section précédente : certains calculs lourds sont fait au préalable à l'estimation des paramètres du réseau de neurones et la taille variable des observations disparaît car le neurone traite en fait le vecteur (de taille fixe) suivant :

$$\left(\sum_{j=1}^l \alpha_j(g) \int \psi_j \phi_1 dP_X, \dots, \sum_{j=1}^l \alpha_j(g) \int \psi_j \phi_k dP_X \right)$$

2.3.4. Estimation des paramètres

Les deux techniques proposées, combinées à la représentation des fonctions de poids par un modèle paramétrique, conduisent à des versions du PMC fonctionnel qui peuvent être implémentées informatiquement et qui n'utilisent que des paramètres numériques.

Pour fixer les idées, considérons l'approche directe appliquée au cas d'une représentation des fonctions de poids par l'intermédiaire d'une unique fonction F . On suppose connues n fonctions, les $(g_i)_{1 \leq i \leq n}$. La fonction g_i est décrite par la liste de couples $(x_j^i, y_j^i)_{1 \leq j \leq m_i}$. On souhaite appliquer le PMC fonctionnel simple décrit par l'équation 2 à ces données. Avec l'implémentation choisie, l'image par le PMC fonctionnel de la fonction i est :

$$H(\theta, g^i) = \sum_{k=1}^s a_k T \left(b_k + \frac{1}{m_i} \sum_{j=1}^{m_i} y_j^i F(w_k, x_j^i) \right) \quad (5)$$

Dans cette équation, θ regroupe l'ensemble des paramètres du PMC fonctionnel sous forme d'un unique vecteur.

Dans un problème de régression, chaque observation g_i est associée à une grandeur à prédire, notée ici t^i . L'estimation des paramètres du PMC consiste à trouver une valeur de θ (i.e des valeurs des w_k , des a_k et des b_k) telle que $H(\theta, g^i) \simeq t^i$ pour tout i . Pour ce faire, il suffit comme pour un PMC classique, ou plus généralement un modèle paramétrique, de choisir un critère d'erreur d et de chercher les paramètres qui minimisent :

$$\mathcal{E}(\theta) = \sum_{i=1}^n d(H(\theta, g^i), t^i) \quad (6)$$

En pratique, on adapte d au problème, en considérant l'erreur quadratique dans le cas de la régression et l'entropie croisée pour la discrimination. Les paramètres optimaux sont obtenus grâce à un algorithme classique d'optimisation (par exemple un quasi-Newton, cf [PRE 92, CIA 00]).

3. Propriétés théoriques

Nous résumons ici des propriétés théoriques énoncées de façon plus formelle dans [ROS 03c].

3.1. Approximation universelle

Une propriété fondamentale des PMC classiques est celle d'approximation universelle : les PMC sont capables d'approcher arbitrairement bien n'importe quelle fonction régulière (cf [LES 93], [HOR 93] et [STI 99]). En nous basant sur les résultats de [STI 99], nous avons proposé dans [ROS 02a, ROS 02b, ROS 02c, CON 02a, ROS 03c] une démonstration d'une propriété équivalente pour les PMC fonctionnels.

Dans le cadre de l'ADF, on suppose donné un compact K d'un espace fonctionnel $L^p(\mu)$ et une fonction continue G de K dans \mathbb{R} . Dans la pratique, ceci correspond par exemple à une régression simple sur une variable fonctionnelle. Notre théorème indique que pour toute précision $\epsilon > 0$, il existe un PMC fonctionnel à deux couches (de la forme donnée par l'équation 2) qui calcule une fonction H telle que $|H(g) - G(g)| < \epsilon$ pour tout $g \in K$. Le théorème comporte quelques hypothèses techniques qu'on trouvera exposées en détail dans [ROS 02a, CON 02a, ROS 03c]. On peut retenir essentiellement qu'il faut que les fonctions de poids soient représentées par un modèle qui assure une bonne couverture de $L^q(\mu)$ (où q est l'exposant conjugué de p). En pratique, si on suppose que les fonctions observées sont dans $L^2(\mu)$, on peut utiliser pour les fonctions de poids un modèle pseudo-linéaire classique (B-splines, ondelettes, etc.). Dans ce cas, on peut aussi adapter le théorème afin de prendre en compte une étape de projection des fonctions observées (cf [CON 02c, CON 02b, CON 02a]). Si on souhaite traiter des fonctions dans $L^p(\mu)$ avec $p \neq 2$, on peut utiliser des fonctions de poids représentées par un PMC numérique.

Comme pour le cas numérique, le théorème n'est pas constructif. Il indique seulement qu'il existe un PMC permettant d'approcher la fonction, sans donner d'indication sur le nombre de neurones nécessaires dans la première couche ou sur la valeur des paramètres. Dans le cas fonctionnel, une autre limitation vient du fait que le théorème n'est valable que si on évalue de façon exacte les intégrales utilisées dans les neurones fonctionnels. En pratique, cela ne semble pas avoir de conséquences notables.

3.2. Consistance

Une autre propriété fondamentale des PMC classiques est la consistance de l'estimation de leurs paramètres optimaux. On montre en effet (cf [WHI 89]) que les paramètres optimaux obtenus en

utilisant l'erreur empirique construite à partir d'un nombre fini d'observations convergent presque sûrement vers les paramètres optimaux théoriques quand le nombre d'observations tend vers l'infini.

Nous avons démontré dans [ROS 03a, ROS 03b, ROS 03c] un résultat comparable pour les PMC fonctionnels. Le théorème de consistance obtenu tient compte à la fois du nombre fini d'observations fonctionnelles, mais aussi du nombre fini de points de discrétisation pour chaque fonction. Le résultat principal est que les paramètres optimaux déterminés comme indiqué dans la section 2.3.4 convergent presque sûrement vers les paramètres optimaux théoriques quand le nombre de fonctions observées tend vers l'infini et que le nombre de points de discrétisation de chaque fonction tend aussi vers l'infini. La convergence est séquentielle, au sens où le nombre de points de discrétisation des fonctions est lié au nombre de fonctions observées.

Contrairement à ce qui est fait dans la plupart des études théoriques des méthodes de l'ADF, notre résultat est basé sur une discrétisation aléatoire des fonctions observées qui contient donc le cas déterministe comme un cas particulier. Notons que la distribution des points de discrétisation doit être la même pour toutes les fonctions, mais que les points effectivement utilisés peuvent différer de fonction à fonction, de même d'ailleurs que le nombre de points, sans que cela n'ait d'influence sur la propriété de consistance.

Le théorème comporte des hypothèses assez techniques détaillées dans [ROS 03a, ROS 03b, ROS 03c]. Du point de vue pratique, on peut retenir quelques limitations importantes :

- les fonctions observées doivent être continues et définies sur un compact ;
- le modèle paramétrique utilisé pour les fonctions de poids doit être régulier et les paramètres du PMC fonctionnel doivent appartenir à un compact ;
- les observations fonctionnelles doivent être indépendantes et identiquement distribuées, comme les points de discrétisation.

Ces limitations sont comparables à celles imposées dans les autres travaux qui prennent en compte la discrétisation des fonctions, comme par exemple [CAR 03].

4. Application en spectrométrie

4.1. Protocole expérimental

Nous présentons dans cette section les résultats de l'application de nos deux méthodes neuronales (l'approche directe et l'approche par projection) à un problème de spectrométrie. Notre but est avant tout de comparer nos méthodes aux méthodes classiques et fonctionnelles étudiées dans [FER 03]. Les auteurs mettent en œuvre dans cet article une méthode originale qui généralise le principe des estimateurs à noyaux au cas fonctionnel (cf aussi [FER 02b] et [FER 02a]). Ils comparent cette méthode à de nombreuses techniques adaptées aux données fonctionnelles comme l'analyse discriminante pénalisée (*Penalized Discriminant Analysis*, cf [HAS 95]) et la régression PLS multiple [MAR 89], ainsi qu'à des techniques plus classiques comme les arbres de régression [BRE 84] et l'analyse discriminante "souple" proposée dans [HAS 94].

Pour nous confronter aux résultats de [FER 03], nous avons donc repris le protocole expérimental proposé par les auteurs. Le jeu de données d'origine est en effet séparé aléatoirement en un ensemble d'apprentissage, utilisé pour estimer le modèle et un ensemble de test, utilisé pour estimer les performances du modèle obtenu. Nous répétons cette séparation aléatoire 50 fois et comparons ensuite les méthodes en fonction de la médiane du taux de mauvais classement.

Nous avons mis en œuvre sur ces données trois modèles neuronaux : un PMC classique appliqué de façon naïve aux données fonctionnelles considérées comme des observations vectorielles (ce qui est possible car les fonctions étudiées ici sont discrétisées de façon régulière), un PMC fonctionnel utilisant l'approche directe et un PMC fonctionnel utilisant l'approche par projection.

Comme les données sur lesquelles nous travaillons sont des fonctions de \mathbb{R} dans \mathbb{R} , il n'est pas utile (et même néfaste) d'utiliser une représentation non linéaire des fonctions de poids et des observations. De ce fait, nous avons opté pour une représentation par une base de B-splines. Le choix du PMC fonctionnel demande alors le réglage de deux méta-paramètres : le nombre de neurones dans la première couche (en accord avec le résultat d'approximation universelle, nous nous sommes limité à des PMC à deux couches) et le nombre de noeuds dans la base de B-splines (que celle-ci soit utilisée pour une représentation des observations par projection ou pour la représentation des fonctions de poids dans l'approche directe).

L'estimation des modèles neuronaux est assez délicate car ils sont sensibles au phénomène de sur-apprentissage : le réseau de neurones apprend le bruit et les performances sur l'ensemble de test sont bien moins bonnes que celles obtenues sur l'ensemble d'apprentissage. Pour lutter contre ce phénomène, nous avons utilisé une technique classique de pénalisation, le *weight decay* [BIS 95] qui est une reformulation neuronale des techniques liées à la régression *ridge* [CAZ 75]. Il s'agit en effet de pénaliser les grandes valeurs des poids du PMC en ajoutant à l'erreur définie par l'équation 6 le terme $\delta\|\theta\|^2$. La méthode nécessite donc le réglage d'un méta-paramètre, le taux de régularisation δ . Notons que cette technique donne en général de meilleurs résultats que celle de l'arrêt prématuré de l'apprentissage (*early stopping*) utilisée par exemple dans [ZHU 97]. Nous aurions pu choisir un terme de pénalisation plus évolué, comme par exemple la dérivée seconde de la fonction calculée par le PMC (cf [BIS 93]), mais les résultats obtenus montrent que la solution simple retenue donne de très bons résultats sans demander un temps de calcul important et une mise en œuvre complexe.

Pour régler les trois méta-paramètres (deux seulement dans le cas du PMC classique), nous avons utilisé une technique classique de validation croisée dans laquelle l'ensemble d'apprentissage est découpé en 5 parties de tailles identiques.

4.2. Données brutes

Nous travaillons sur des données spectrométriques utilisées pour classer des échantillons selon une propriété physico-chimique qui n'est pas accessible directement (et demande donc une analyse spécifique). Les données utilisées ont été obtenues grâce à un *Tecator Infratec Food and Feed Analyzer*¹ travaillant dans le proche infra-rouge (longueurs d'onde comprises entre 850 et 1050 nanomètres). La mesure est effectuée par transmission à travers un échantillon de viande finement hachée qui est ensuite analysé par un procédé chimique pour déterminer son taux de graisse. Les spectres obtenus correspondent à l'absorbance ($-\log_{10}$ de la transmittance mesurée par l'appareil) pour 100 longueurs d'onde régulièrement réparties entre 850 et 1050 nm. Les échantillons de viande sont répartis en deux classes, selon qu'ils contiennent plus ou moins de 20 % de graisse (on a 77 spectres correspondant à plus de 20 % de graisse et 138 avec moins de 20 % de graisse). Le problème est alors de discriminer les spectres afin d'éviter l'analyse chimique, coûteuse et longue. La figure 1 représente 40 spectres choisis aléatoirement dans les 215 échantillons. La répartition aléatoire entre l'ensemble d'apprentissage et l'ensemble de test place 160 spectres dans le premier et 55 dans le second, sans équilibrer les classes.

Méthode	Premier quartile	Médiane	Moyenne	Troisième quartile
Classique	0	0.019	0.018	0.036
Directe	0.018	0.028	0.036	0.036
Projection	0	0.018	0.018	0.036

TAB. 1 – Taux d'erreur pour les données spectrométriques

1. Les données sont disponibles à l'URL <http://lib.stat.cmu.edu/datasets/tecator>

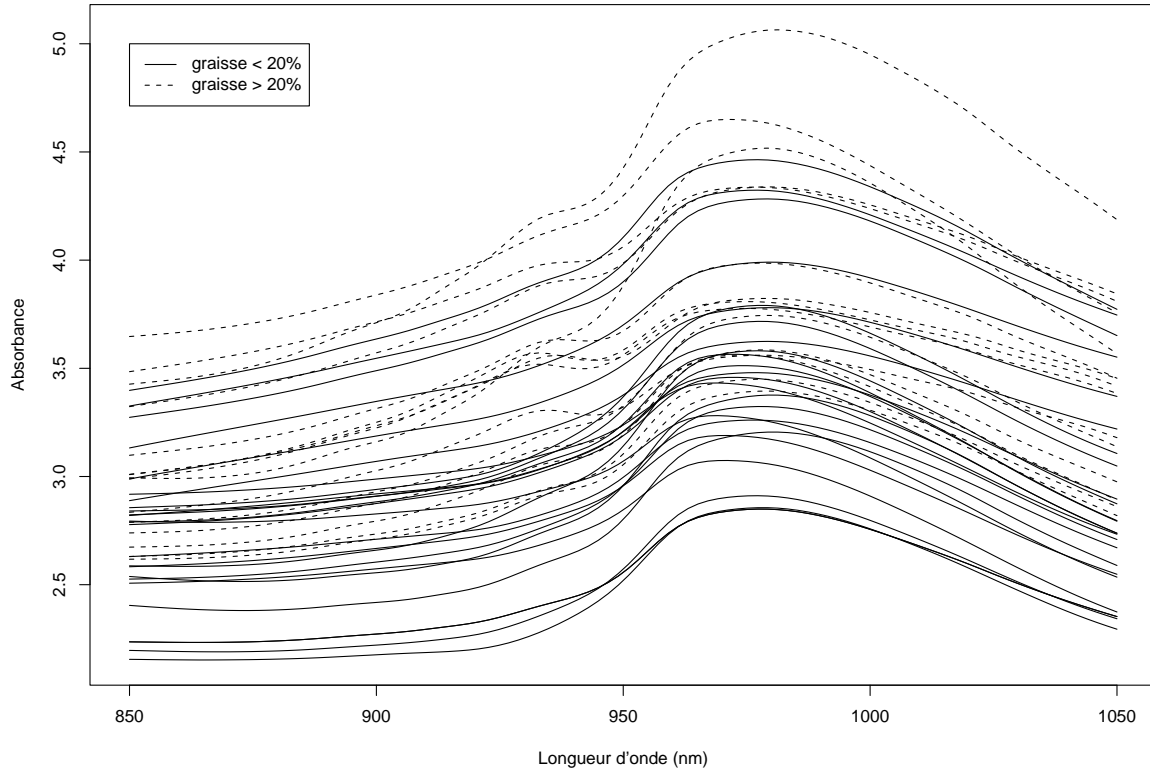


FIG. 1 – 40 spectres

La table 1 résume les taux de mauvais classement obtenu par les trois techniques neuronales étudiées. On constate que les taux d'erreur obtenus sont très bons puisque dans plus de la moitié des tirages (répartitions aléatoires entre l'ensemble d'apprentissage et l'ensemble de test), la méthode fonctionnelle par projection donne moins de 2 % d'erreur de classement. Cependant, cette méthode n'est que marginalement meilleure que l'approche neuronale classique (en fait, une comparaison tirage par tirage indique que l'approche par projection donne de meilleures performances sur 30 tirages parmi 50 et des performances égales sur 13 tirages). De plus, l'approche directe donne de moins bons résultats que les deux autres méthodes. En fait, le seul intérêt des approches fonctionnelles réside dans leur parcimonie car la méthode par projection utilise en moyenne 69 paramètres numériques contre 288 pour l'approche classique, ce qui limite les besoins en stockage et les temps de calcul en phase d'exploitation.

La comparaison avec les résultats rapportés dans [FER 03] montre que les meilleures approches neuronales sont très performantes puisque la meilleure méthode étudiée dans [FER 03] obtient un taux d'erreur médian d'environ 2.2 %. De plus, la méthode en question est une méthode hybride assez complexe qui se base sur un modèle fonctionnel non paramétrique dans lequel les fonctions sont comparées par l'intermédiaire d'une semi-norme construite en utilisant une régression PLS multivariée. D'après [FER 03], une approche non paramétrique fonctionnelle pure donne de mauvais résultats (de l'ordre de 20 % d'erreur de classement en moyenne), vraisemblablement pour des raisons de décalages entre les spectres, comme nous le verrons dans la section suivante. Notons de plus que la méthode hybride utilise 1300 paramètres numériques.

4.3. Pré-traitements fonctionnels

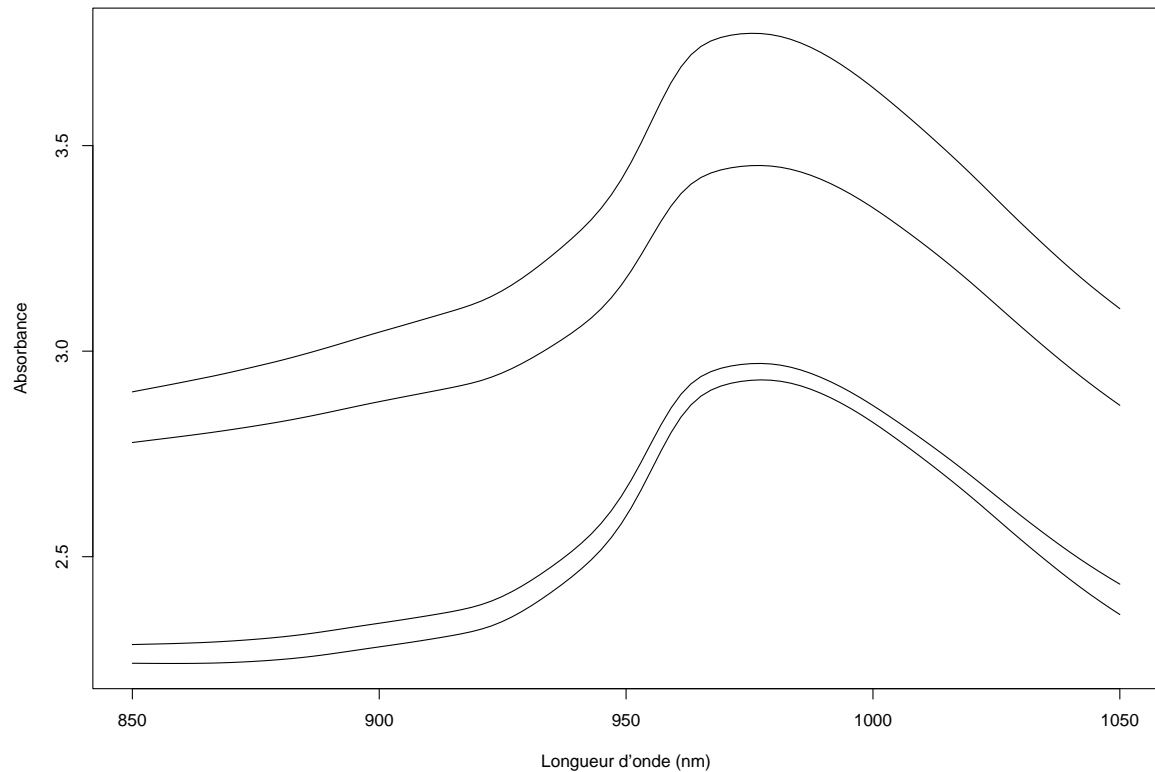


FIG. 2 – 4 spectres avec un même taux de graisse

Cette première expérience peut paraître décevante car même si elle montre l'efficacité des techniques neuronales, elle ne donne pas un clair avantage à l'approche fonctionnelle, excepté pour la parcimonie de la représentation. Cependant, l'approche fonctionnelle ne se limite pas à une technique de mise en œuvre de réseaux de neurones. En effet, elle permet de prendre en compte la nature des observations et de leurs appliquer des pré-traitements fonctionnels. Ferraty et Vieu remarquent en effet dans [FER 02b] que les spectres d'absorbance comportent souvent un effet moyen qui n'apporte aucune information. Plus précisément, la valeur moyenne de l'absorbance n'est pas un bon prédicteur du taux de graisse de l'échantillon de viande. En fait, deux spectres peuvent être très similaires en terme de forme et correspondre à des taux de graisse proches, tout en ayant des valeurs moyennes d'absorbance très différentes. La figure 2 illustre le problème avec quatre spectres correspondants à des échantillons dont la teneur de graisse est de 6.4 %.

Pour remédier à ce problème, Ferraty et Vieu proposent dans [FER 02b] et [FER 03] de traiter les dérivées des spectres (ce qui demande une approche fonctionnelle), et montrent que la dérivée seconde est la plus informative. Comme dans [FER 03], nous avons estimé les dérivées secondes des spectres en approchant chaque spectre sur une base de B-splines (20 noeuds), en calculant les dérivées formelles des fonctions ainsi obtenues et en ré-échantillonnant les dérivées afin d'obtenir 100 observations par fonction. La figure 3 représente les dérivées secondes des 40 spectres de la figure 1. La comparaison entre les deux figures ne donne pas au premier abord un avantage particulier aux dérivées secondes. Cependant, la figure 4 montre que l'effet moyen observé sur la figure 2 disparaît : des spectres de même forme ont des dérivées secondes très proches.

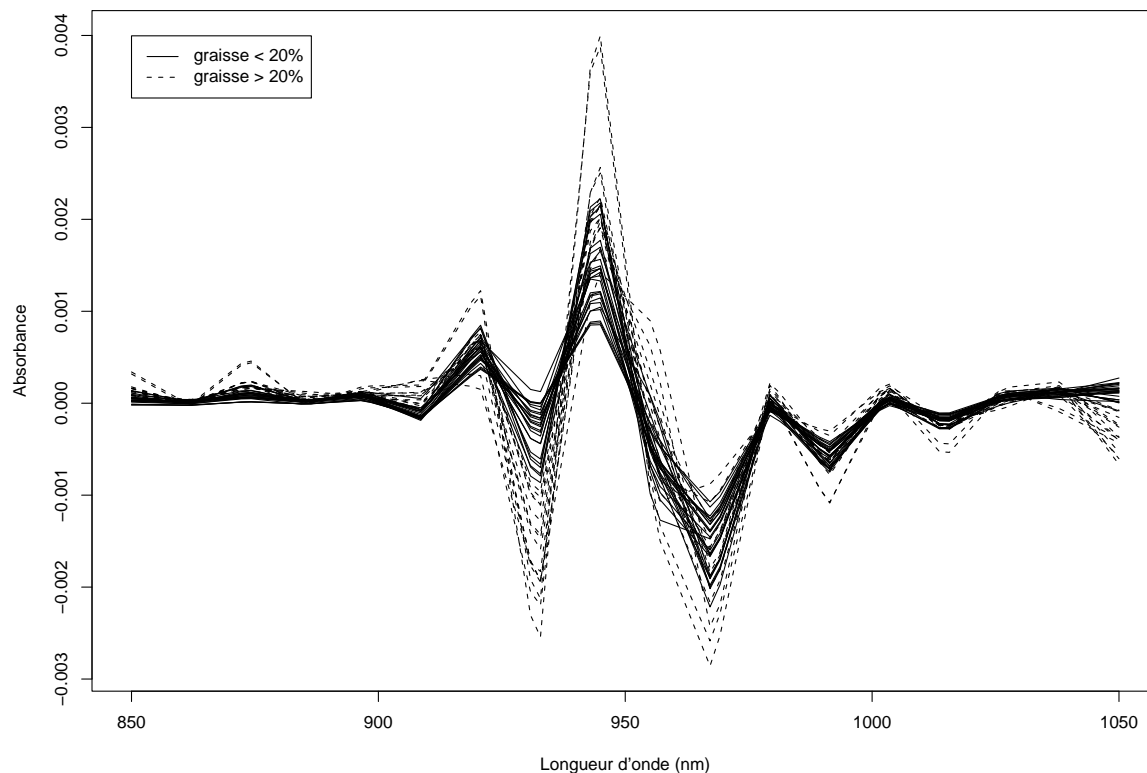


FIG. 3 – Dérivées secondes de 40 spectres

Notons qu'il n'est pas possible de simplement centrer chaque courbe en soustrayant la moyenne car celle-ci apporte tout de même une information sur le taux de graisse. On constate par exemple que parmi les 138 spectres correspondant à un taux de graisse inférieur à 20 %, 75 possèdent une valeur moyenne d'absorbance inférieure à 3, alors que seulement 11 spectres sur les 77 correspondant à un taux de graisse supérieur à 20 % sont dans ce cas.

Méthode	Premier quartile	Médiane	Moyenne	Troisième quartile
Classique	0	0.018	0.013	0.018
Directe	0	0	0.007	0.018
Projection	0	0.009	0.014	0.018

TAB. 2 – Taux d'erreur pour dérivées secondes des données spectrométriques

Après la dérivation, nous avons traité les fonctions obtenues exactement comme les spectres d'origine. La table 2 résume les taux de mauvais classement obtenu par les trois techniques neuronales étudiées. On constate que les performances sont améliorées, en particulier pour la méthode directe qui ne commet aucune erreur de classement pour plus de la moitié des tirages. De plus, les deux méthodes fonctionnelles donnent maintenant de meilleurs résultats que l'approche classique. L'approche non paramétrique proposée dans [FER 03] donne un taux médian d'erreur de classement légèrement inférieur à 2.2 % ce qui la classe en tête des méthodes étudiées dans [FER 03], mais derrière les approches fonctionnelles neuronales. Celles-ci restent de plus très parcimonieuses (82 paramètres

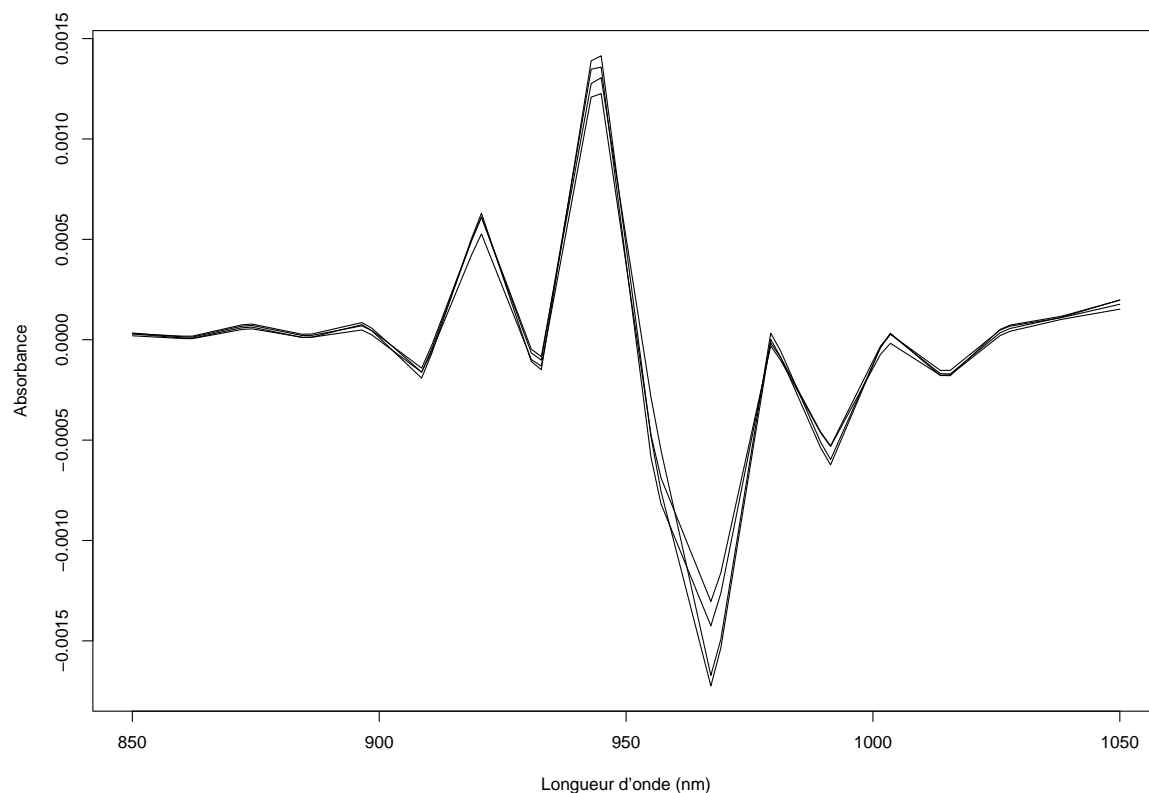


FIG. 4 – Dérivées secondes de 4 spectres avec un même taux de graisse

numériques en moyenne pour l'approche directe, 76 pour l'approche par projection et 391 pour l'approche naïve).

Notons que l'approche naïve doit être considérée en toute rigueur comme une approche mixte. En effet, seule une prise en compte du caractère fonctionnel des observations permet de tester l'effet d'une dérivation des spectres (ou d'ailleurs de toute autre transformation fonctionnelle). Même si le traitement des données transformées est complètement classique, la transformation est elle fonctionnelle.

4.4. *Commentaires*

On peut tirer de ces expériences plusieurs enseignements. Tout d'abord, les approches fonctionnelles neuronales donnent des résultats très satisfaisants, au moins aussi bon qu'un modèle neuronal classique et meilleurs que ceux des méthodes concurrentes explorées dans [FER 03]. De plus, les modèles fonctionnels neuronaux sont parcimonieux et construisent avec peu de ressources un bon résumé des données. On pourra donc envisager leur application à des problèmes contenant de très nombreuses observations (le modèle non paramétrique fonctionnel de [FER 03] utilise un nombre de paramètres proportionnel au nombre de données, même si celles-ci sont très redondantes, ce qui pose des problèmes de stockage et de temps de calcul en phase d'exploitation). Enfin, l'approche fonctionnelle permet d'appliquer des pré-traitements aux données qui tiennent compte de la nature de celles-ci ce qui améliore considérablement les performances sur les exemples étudiés.

L'application présentée montre qu'il est difficile de choisir entre les deux implémentations que nous avons proposées. Cependant, il est parfaitement possible (au prix d'un temps de calcul doublé dans la phase d'estimation) d'utiliser la validation croisée à la fois pour sélectionner les méta-paramètres des deux approches mais aussi pour départager les implémentations. En appliquant cette technique sur les données brutes et sur les dérivées secondes, nous obtenons des résultats très satisfaisants résumés dans la table 3. On constate une très légère détérioration des performances sur les dérivées secondes, mais il est tout de même légitime de conclure que la technique d'implémentation peut être sélectionnée par validation croisée comme les autres méta-paramètres du modèle considéré.

Données	Premier quartile	Médiane	Moyenne	Troisième quartile
Brutes	0	0.018	0.018	0.036
Dérivées secondes	0	0	0.009	0.018

TAB. 3 – Taux d'erreur pour le PMC fonctionnel déterminé par validation croisée

Notons pour finir que selon le jeu de données, la validation croisée sélectionne l'une ou l'autre des implémentations : les données brutes sont souvent associées à l'implémentation par projection, alors que les dérivées secondes correspondent plutôt à l'approche directe, sans pour autant que ces associations soient systématiques. L'observation des courbes plaide pour un meilleur traitement des fonctions très régulières par projection, alors que les fonctions plus bruitées semblent mieux adaptées à un traitement direct. Ces constatations expérimentales doivent être confirmées sur d'autres données.

5. Conclusion

Nous avons proposé dans cet article un modèle neuronal original adapté aux traitements des données fonctionnelles, en particulier la discrimination de telles données et plus généralement la régression sur variables explicatives fonctionnelles. Le perceptron multi-couches (PMC) fonctionnel partage avec la version numérique qu'il généralise des propriétés théoriques fondamentales, l'approximation universelle et la consistance des techniques classiques d'estimation.

Nous avons décrit deux techniques pour la mise en œuvre informatique du PMC fonctionnel, puis nous les avons appliquées à un problème de discrimination de spectres d'absorbance. Les résultats expérimentaux montrent que les PMC fonctionnels sont des outils très satisfaisants. Les taux d'erreurs obtenus sont plus faibles que ceux atteints par d'autres méthodes (fonctionnelles ou classiques) sur les mêmes données. De plus, l'approche fonctionnelle apporte une valeur ajoutée par rapport à un simple PMC numérique, tant en terme de taux d'erreur qu'en terme d'interprétation des données et donc de pré-traitements envisageables. Enfin, la propriété de parcimonie démontrée formellement dans le cas des PMC numériques semble se retrouver expérimentalement dans le cas fonctionnel puisque les excellents taux de bon classement relevés sont obtenus par des modèles utilisant un nombre très réduit de paramètres numériques comparativement au volume des données et aux méthodes concurrentes.

6. Bibliographie

- [ABR 03] ABRAHAM C., CORNILLON P.-A., MATZNER-LOBER E., MOLINARI N., Unsupervised Curve Clustering using B-Splines, *Scandinavian Journal of Statistics*, vol. 30, n° 3, 2003, p. 581–595.
- [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,
- [BAR 97] BARDOS M., ZHU W. H., Comparaison de l'analyse discriminante linéaire et des réseaux de neurones – Applications à la détection de défaillance d'entreprises, *Revue de Statistique Appliquée*, vol. XLV, n° 4, 1997, p. 65–92.

- [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.
- [BES 03] BESSE P., CARDOT H., *Analyse des données (édité par Gérard Govaert)*, Chapitre 6 : Modélisation statistique de données fonctionnelles, p. 167-198, Hermès/Lavoisier, 2003.
- [BIS 93] BISHOP C., Curvature-driven smoothing : a learning algorithm for feedforward networks, *IEEE Trans. on Neural Networks*, vol. 4, n° 5, 1993, p. 882-884.
- [BIS 95] BISHOP C., *Neural Networks for Pattern Recognition*, Oxford Press, 1995.
- [BRE 84] BREIMAN L., FRIEDMAN J. H., OLSHEN R. A., STONE C. J., *Classification and Regression Trees*, Wadsworth, 1984.
- [CAR 03] CARDOT H., FERRATY F., SARDA P., Spline Estimators for the Functional Linear Model, *Statistica Sinica*, vol. 13, 2003, p. 571-591.
- [CAZ 75] CAZES P., Protection de la régression par utilisation de contraintes linéaires et non linéaires, *Revue de Statistique Appliquée*, vol. XXIII, n° 3, 1975, p. 37-57.
- [CIA 00] CIARLET P. G., *Introduction à l'analyse numérique matricielle et à l'optimisation*, Dunod, Février 2000.
- [CON 02a] CONAN-GUEZ B., Modélisation supervisée de données fonctionnelles par perceptron multicouches, Thèse de doctorat, Paris IX Dauphine, Décembre 2002.
- [CON 02b] 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.
- [CON 02c] CONAN-GUEZ B., ROSSI F., Multi-Layer Perceptrons for Functional Data Analysis : a Projection Based Approach, DORRONSORO J. R., Ed., *Artificial Neural Networks - ICANN 2002*, Madrid, August 2002, Springer, p. 667-672.
- [DAU 01] DAUVOIS J., FERRÉ L., YAO A.-F., Un modèle semi-paramétrique pour variables aléatoires hilbertienne, *C. R. Acad. Sci. Paris*, vol. 333, 2001, p. 947-952.
- [FER 00] FERRÉ L., YAO A.-F., Functional sliced inverse regression analysis, rapport n°0248-3289, 2000, Laboratoire de statistique et probabilités, Université Paul Sabatier, Toulouse, A paraître dans *Statistics* (2003).
- [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.
- [FER 02a] FERRATY F., GOIA A., VIEU P., Régression non-paramétrique pour des variables aléatoires fonctionnelles mélangeantes, *C. R. Acad. Sci. Paris*, vol. 334, 2002, p. 217-220, Série I.
- [FER 02b] FERRATY F., VIEU P., The Functional Nonparametric Model and Application to Spectrometric Data, *Computational Statistics*, vol. 17, n° 4, 2002.
- [FER 03] FERRATY F., VIEU P., Curves Discriminations : a Nonparametric Functional Approach, *Computational Statistics and Data Analysis*, vol. 44, n° 1-2, 2003, p. 161-173.
- [HAS 94] HASTIE T., BUJA A., TIBSHIRANI R., Flexible Discriminant Analysis by Optimal Scoring, *J. Amer. Statist. Assoc.*, vol. 89, 1994, p. 1255-1270.
- [HAS 95] HASTIE T., BUJA A., TIBSHIRANI R., Penalized Discriminant Analysis, *Annals of Statistics*, vol. 23, 1995, p. 73-102.
- [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.
- [MAR 89] MARTENS H., NAES T., *Multivariate Calibration*, John Wiley, New York, 1989.
- [PRE 92] PRESS W. H., TEUKOLSKY S. A., VETTERLING W. T., FLANNERY B. P., *Numerical Recipes in C*, Cambridge University Press, second édition, 1992,

-
- [RAM 97] RAMSAY J., SILVERMAN B., *Functional Data Analysis*, Springer Series in Statistics, Springer Verlag, June 1997.
- [ROS 02a] ROSSI F., CONAN-GUEZ B., Modélisation supervisée de données fonctionnelles par perceptron multi-couches, *Actes des neuvièmes journées de la SFC (conférence invitée)*, Toulouse, France, Septembre 2002, p. 93–100.
- [ROS 02b] ROSSI F., CONAN-GUEZ B., FLEURET F., Functional Data Analysis With Multi Layer Perceptrons, *Proceedings of IJCNN 2002 (WCCI 2002)*, vol. 3, Honolulu, Hawaii, USA, May 2002, IEEE/NNS/INNS, p. 2843–2848.
- [ROS 02c] 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.
- [ROS 03a] ROSSI F., CONAN-GUEZ B., Estimation consistante des paramètres d’un modèle semi-paramétrique pour des données fonctionnelles discrétisées aléatoirement, rapport n°0334, Octobre 2003, LISE/CEREMADE, <http://www.ceremade.dauphine.fr/>.
- [ROS 03b] ROSSI F., CONAN-GUEZ B., Estimation consistante d’un modèle paramétrique fonctionnel en présence de discrétisation aléatoire, *Actes des XXXVèmes journées de Statistique de la SFdS*, Lyon, France, Juin 2003, p. 819–822.
- [ROS 03c] ROSSI F., CONAN-GUEZ B., Functional Multi-Layer Perceptron : a Nonlinear Tool for Functional Data Analysis, rapport n°0331, september 2003, LISE/CEREMADE, <http://www.ceremade.dauphine.fr/>.
- [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.
- [WHI 89] WHITE H., Learning in Artificial Neural Networks : A Statistical Perspective, *Neural Computation*, vol. 1, n° 4, 1989, p. 425–464,
- [ZHU 97] ZHU W. H., GUICHENEY C., BERDAGUÉ J.-L., JOUSSET J., Application des réseaux perceptrons multicouches au contrôle de la qualité des aliments par analyse sensorielle – Comparaison des résultats avec différentes méthodes d’analyse discriminante, *Revue de Statistique Appliquée*, vol. XLV, n° 2, 1997, p. 39–57.