A Neural Network Demand System

Julien Boelaert

Documents de Travail du
Centre d'Economie de la Sorbonne

CES Working Papers

A Neural Network Demand System

Julien BOELAERT

2013.81

Maison des Sciences Économiques, 106-112 boulevard de L'Hôpital, 75647 Paris Cedex 13
http://centredeconomiesorbonne.univ-paris1.fr/bandeau-haut/documents-de-travail/
ISSN : 1955-611X

A Neural Network Demand System

Julien Boelaert*

December 5, 2013

Abstract - We introduce a new type of demand system using a feedforward artificial neural network. The neural network demand system is a flexible system that requires few hypotheses, has no roots in consumer theory but may be used to test it. We use the system to estimate demand elasticities on micro data of household consumption in Canada between 2004 and 2008, and compare the results to those of the quadratic almost ideal demand system.

Résumé - Nous présentons un nouveau type de système complet de demande utilisant un réseau de neurones artificiels. Le système de demande à réseau de neurones est un système flexible nécessitant un minimum d'hypothèses. Sa formulation ne provient pas de la théorie du consommateur, mais il peut être utilisé pour la tester. Nous appliquons ce système à l'estimation d'élasticités de la demande des ménages sur des microdonées canadiennes de 2004 à 2008, et comparons les résultats à ceux obtenus à l'aide du système QUAIDS.

JEL C45, D12.
Keywords : Estimating demand systems, Neural networks, Flexible forms, Quadratic almost ideal demand system (QUAIDS)

*Université Paris 1 Panthéon-Sorbonne
Centre d'Economie de la Sorbonne - Axe Economie Mathématique et Jeux
Maison des Sciences Economiques
106-112 boulevard de l'Hôpital, 75013 Paris
julien.boelaert@gmail.com
We thank Prof. S. Langlois of the Sociology Dept. of the Université Laval in Québec for giving us access to the Canadian consumption data.
This work has been supported by an ANR grant # 06-BLAN-0140 MALDI.

1 Introduction

The field of consumer demand analysis has long held a central position in the development both of economic analysis and econometrics. According to A. Deaton [4] "demand analysis is thus in the rare position in econometrics of possessing long interrelated pedigrees on both theoretical and empirical sides." The field has evolved rapidly since H. Working's pioneering work in 1943 [32], as authors have strived to make demand systems both more flexible and more conform to microeconomic theory. Today one of the systems that have come closest to perfection with regard to these two goals is the Quadratic Almost Ideal Demand System [1], a more flexible[1] extension of the Deaton and Muellbauer's celebrated AIDS [5]. The QUAIDS is derived from modified PIGLOG preferences, and is designed to have quadratic Engel curves and a simple price-dependency of the quadratic log-income term. It is widely used and conforms well with survey consumption data. It is the fruit of a long interwoven development of economic theory and econometric techniques, in a long quest for a good specification, spanning nearly 55 years since Working's article. However, its firm roots in economic theory, and the simplicity of the functional forms it is based on, mean that it is built upon strong hypotheses regarding consumer behaviour. It would therefore be interesting to observe how it compares with more flexible, theory-free estimation techniques. In particular, as the main practical purpose of demand systems is the estimation of price- and income elasticities, one might raise the - somewhat provocative - question of whether we needed a century of economic thought to make sense of empirical demand data, when flexible models needing no economic theory may attain the same practical value. Such is the purpose of this article.

A multilayer perceptron is one such technique of machine learning, and it can be readily applied to the problem of consumer demand analysis. It is a feedforward artificial neural network which can be used for classification or regression tasks, and has been famous since the end of the 1980s for being a universal function approximator [27]. Applying this advance in statistical methods to consumption analysis is an attempt at continuing the tradition of interwoven progress of both fields. The multilayer perceptron is particularly well suited to this problem as it can handle many variables, both real and categorical, provided enough data is available, handles high dimension regressions and irrelevant regressors well[2], and yields, among others, analytical expressions of the partial derivatives. One last desirable property is that in some cases it can be restricted to conform with constraints, and thus used to test economic theory.

There have been some previous uses of multilayer perceptrons in demand analysis.

[1] The QUAIDS model is shown to be more flexible than other common specifications in [6]. The more recent EASI demand system by Lewbel and Pendakur [19] seems to yield even more flexibility, and should be investigated in future work.

[2] Numerous or irrelevant regressors are a well-known problem in other flexible regression techniques such as kernel regression or radial basis functions. Because of the local nature of these methods, additional regressors make the problem space grow exponentially, and thus require a large number of additional data, whether they are relevant or not. This is not the case in multilayer perceptrons (if the number of hidden neurons is fixed the dimension of the parameter space grows linearly with the number of regressors) and the training algorithm will normally shrink the weights of irrelevant regressors to values close to 0.

The first studies were confined to particular branches such as tourism [18] or electricity consumption [2]. In 2008 a paper by McAleer *et al* [20] introduced a complete demand system based on neural networks, which demonstrated great flexibility in the inference of Engel curves. Their approach consists in combining neural networks with two existing demand system specifications (AIDS and EASI) : to each specification they add one term defined by a neural network function whose inputs are demographic variables and deflated total expenditure (each specification having its own usual deflator), and another neural-network-based term for the error. Their purpose is to enhance the flexibility of existing specifications in terms of total expenditure. Ours is a different approach, in which the neural network forms a demand system by itself with no additional model specifications. In particular this means that our approach allows for nonlinear (and unplanned or unexpected) interactions between total expenditure, prices and demographic variables, while being easier to implement.

The rest of this article is organised as follows. Section 2 contains a brief presentation of multilayer perceptrons and their use in regression analysis. Section 3 presents the data and the specifications and econometrics of QUAIDS and NNDS. Section 4 details the estimation results of both models on a Canadian dataset. Section 5 closes with some concluding remarks.

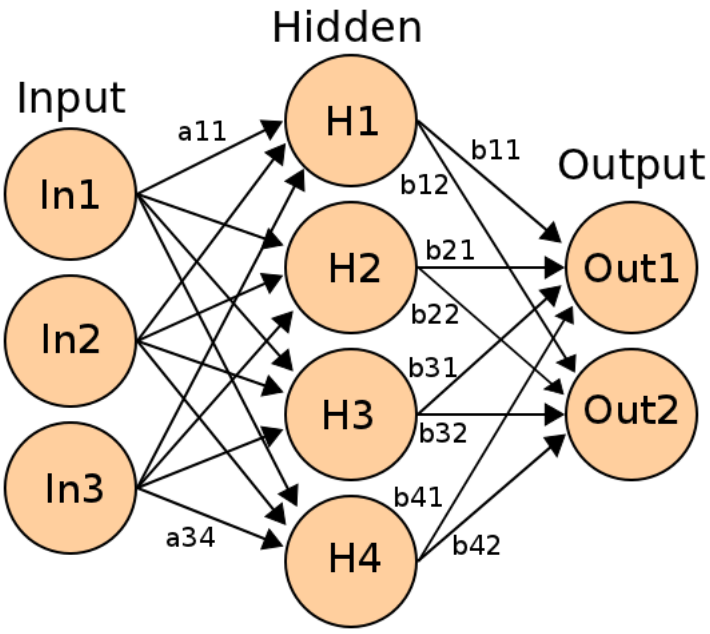
2 Multilayer Perceptrons and their use in regression analysis.

The multilayer perceptron (MLP) is a kind of feedforward artificial neural network, widely used in many areas for supervised classification and regression tasks. Its name derives from the original Perceptron model, a powerful linear discriminant devised in 1957 by F. Rosenblatt (see [27] for details). Since the progresses made in the late 1980s, it has been applied to complex problems in a wide variety of fields, including economics [17], time-series analysis [15], meteorology [9], computer vision [15], medicine [10] and civil engineering [22].

The MLP works as a very simplified and idealized version of a brain : a collection of neurons communicating with other neurons through synapses, sending either inhibating or stimulating signals. Its structure is usually shown in a graph form as in figure 1. As its name suggests, the MLP consists of multiple layers : an input layer, one or more hidden layers, and an output layer. It is called a feedforward network beacause information is transmitted through the network in only one direction, any given neuron passing on information only to neurons in the next layer. While the biological may be entertaining, an MLP is essentially a simple mathematical function, built in the following way :

- Each neuron in the input layer (In1, In2, In3 in figure 1) is an input variable.
- Each synapse (on figure 1, each arrow) has an associated real value called a weight. On figure 1 the weight between nodes In3 and H4 is called a34, while the one between H4 and O3 is called b43.
- Each neuron H_i in a hidden layer is a function of the values of the neurons N_k in

Figure 1: The multilayer perceptron as a graph. This network has three input variables, one hidden layer of four neurons, two output variables, and no bias neurons.



the previous layer and of the synapses a_{ki} that connect them to Hi :

$$H_i = \mathrm{f}(\sum_{k=1}^{p} a_{ki} N_k)$$

where p denotes the number of neurons in the previous layer, a_{ki} is the value of the synapse linking N_k and H_i, and f is called an activation function. A typical choice for f is the logistic function, but other choices are possible. Neuron H2 on figure 1 would therefore equal $H_2 = \mathrm{f}(a_{12} \cdot In_1 + a_{22} \cdot In_2 + a_{32} \cdot In_3)$.

- Each neuron O_i in the output layer corresponds to an output variable and works in the same way as the neurons in the hidden layer : they are a function of the neurons of the last hidden layer and the synapses connecting them to O_i. The choice of activation function for the output neurons depends on the problem at hand : common choices are linear functions, logistic functions or the softmax function, which forces the outputs to be positive and sum to one ($O_i = \frac{exp(\sum_{k=1}^{p} a_{ki} N_k)}{\sum_{j=1}^{q} exp(\sum_{k=1}^{p} a_{kj} N_k)}$ where q is the number of output neurons).

- Each neuron in a hidden or output layer may additionally be connected to a constant term called *bias*, through an extra incoming synapse. If neuron H_2 of figure

1 were to include a bias term, it would equal $H_2 = \mathrm{f}(a_{02} \cdot 1 + a_{12} \cdot In_1 + a_{22} \cdot In_2 + a_{32} \cdot In_3)$.

Thus an MLP with any given structure and set of weights works as a function, mapping each vector in the input space to exactly one vector in the output space. As an illustration, let us explicitly state the function giving the kth output neuron of a fully connected network with a single hidden layer, with activation functions f and g for the hidden and output layers, respectively :

$$O_k = \mathrm{g}\left(b_{0k} + \sum_{j=1}^{N_{hidden}} b_{jk} \cdot \mathrm{f}\left(a_{0j} + \sum_{i=1}^{N_{input}} a_{ij} \cdot I_i\right)\right)$$

It is interesting to note that if f and g are both chosen to be linear, the MLP reduces to a general linear model. Furthermore, if f is chosen to be a sigmoid function, g a linear function, and the network includes only one hidden neuron, the MLP reduces to a logistic regression. It can thus be seen as a generalization of these common techniques.

The power and practical use of the MLP stems from the fact that given the right structure and the right set of weights, it can approximate any function (for proofs and references see Ripley [25]). Thus the multilayer perceptron is often called a *universal function approximator*. This of course raises the question of how to find the right structure and weights ; there is no simple answer to these questions, but a large body of litterature gives some guidelines. The search for the "best" structure of a MLP (number of hidden layers, number of neurons per hidden layer, full or partial connection between layers) is a very complex problem, and Ripley [25] gives some procedures to find such a structure. For most cases a single hidden layer seems sufficient, and the number of hidden neurons depends on the complexity of the function that is to be approximated. As Ripley puts it, "a heuristic reason why feed-forward networks might work well with modest numbers of hidden units is that the first stage allows a projection onto a subspace of much lower dimensionality, within which the approximation can be performed." A common procedure for choosing the number of hidden neurons is cross-validation[3].

For a given structure, the search for the optimal set of weights is called the *training* of the network. The most popular approach is the one pioneered by Rumelhart and McClelland [28], called *back-propagation*. In back-propagation, we fit the MLP by least squares : for a given dataset or point we can easily compute the squared prediction error of the neural network. The analytic form of the network function ensures that computation of the partial derivative of the quadratic error with regard to each weight is straightforward. We can then initialize the weights randomly and use a form of steepest descent - a generalized delta rule - to find the set of weights which minimizes the quadratic

[3]Cross-validation is a procedure in which the dataset is divided in N subsets (typically 10 subsets in what is called "ten-fold cross validation"), and the estimation procedure is run N times. During each run k, all but the k-th subset are used as a training sample, while subset k is used as the test sample on which error statistics are reported. Cross-validation is usually a good estimator of average prediction error, and is widely used in the context of statistical learning, both for robust error reporting and for model selection. For a more detailed treatment see Hastie *et al* [11].

error. However, the cost function mapping weight-sets to quadratic error is generally highly non-linear and multimodal with many local optima, and simple steepest-descent-based methods are prone to get "stuck" in one of these local optima. Many extensions to the back-propagation algorithm have been proposed to overcome this problem and make the search faster (see Ripley [25] or Rojas [27] for a brief overview). Alternative neural network training methods include quasi-Newton methods, conjugate gradient methods, bayesian training (see Neal [21] or Freitas [3]) or gradient-free optimization algorithms such as particle swarm optimization [16] or artificial bee colony optimization [14]. All of these methods are iterative, and necessitate a stopping criterion ; the algorithm can either stop after a fixed number of iterations, or once a given degree of fitness has been reached, or when another criterion is met.
However, as in many machine learning problems, excessive minimization of the cost function can be counter-productive, as too much learning on the original dataset often leads to poor prediction performance, a phenomenon called overlearning. The intuition behind overlearning is that we want the neural network to learn the underlying data-generating process but not the perturbations of the points in our dataset. A common strategy to avoid overlearning is to divide the dataset into a training sample and a testing sample ; the network is trained using only the first sample, and the reported cost after training is calculated on the second sample, thus truly reporting prediction error. This, in turn, can lead to another stopping criterion : stop the learning process once the error on the testing sample starts to rise, in which case a third independant sample is used to compute the final prediction error. Sarle [29] contains a great wealth of practical information on artificial neural networks and their training.

Weight regularization is another popular way of avoiding overlearning, as indicated in [25]. The idea is to avoid high absolute values of the weights, by adding a penalizing term to the cost function during training :

$$\text{cost}(\mathbf{v}) = \frac{1}{N}\sum_{i=1}^{N}(\widehat{y}_i - y_i)^2 + \lambda \cdot \frac{1}{m}\sum_{k=1}^{m} v_k^2 \tag{1}$$

where the v_k are the network weights (previously noted a_{ij} and b_{jk}), m is the number of weights to be penalized, λ is the regularization coefficient, and in this example the network has a single output $\widehat{y}$. As explained in [29], it is best not to penalize the bias terms of the network, and even better to use different penalization coefficients for weights belonging to different layers. The λ coefficient must be chosen carefully to balance bias and variance of the neural network ; a good choice is cross-validation. Regularization proved very useful in our empirical work, because it helped stabilizing the partial derivatives of the network between different runs.

Multilayer perceptrons can be of great value in empirical economics, as they provide a relatively simple tool for regression problems without necessitating any *a priori* model of the underlying relation, the only hypothesis being that there exists a single data-generating function. This flexibility does not come without a few drawbacks when compared to more classical econometrics. One problem inherent to the training process is that one can never be sure of having reached the global minimum of the error function,

and that for a given training samples different runs of the training process will usually give different weights. Usual practice is to run the training algorithm many times to obtain better results and assess the robustness of the learning process. Another problem is that the model-free nature of an MLP means that the estimated parameters - the weights - are not interpretable ; the neural network functions as a black box which gives good predictions but whose inner workings are mysterious. Interpretation does not rely on the analysis of regressor coefficients or the usual statistical significance tests, but rather on predictions for a set of points, or on the analysis of partial derivatives. The trained network being a composition standard functions, it is straightforward by using the chain rule to compute the partial derivatives of each explanatory variable with respect to any regressor. In the case of demand analysis, this can be used to yield demand elasticities. This is one practical advantage of multilayer perceptrons over other flexible regression techniques such as kernel regression or radial basis functions. Another such advantage is the fact that multilayer perceptrons naturally handle irrelevant regressors well by simply setting their weights to low absolute values during the training process, see Sarle [29].

A possible objection to the use of neural networks in demand analysis stems from the fact that multilayer perceptrons are primarily designed for prediction, not interpretation. In the words of Hastie, Tibshirani and Friedman [11],

> [Neural networks] are especially effective in [...] settings where prediction without interpretation is the goal. They are less effective for problems where the goal is to describe the physical process that generated the data and the roles of individual inputs.

One might therefore object that, while being an excellent tool for the estimation of Engel curves (which is inherently an exercise in prediction, the goal being the prediction of budget shares for different levels of total expenditure) neural networks are not suited for the estimation of elasticities. Indeed the estimated elasticities are derived from the partial derivatives of the trained network, which are merely a by-product of the original goal of best prediction. Furthermore, to the best of our knowledge, the statistical properties of the estimated partial derivatives are not precisely known.

We thus turned to simulation to evaluate the performance of the NNDS. First we tried estimating partial derivatives using simulations on simple examples, which yielded good results, except at the bounds (a typical result for semi-parametric models) and for constant partial derivatives (which were systematically estimated as a parabolic shape, with the mean derivative at the approximate desired value). We then simulated data using an AIDS specification with randomly generated parameters, and applied different neural networks specifications to it (expenditures in share-form, in dollars and in logarithmic form, and different specifications of the input variables). The predictions of the trained neural networks were always very close to the true values of the output variable, but interestingly enough the quality of estimation of the elasticities depended on the chosen specification. The specification used in the rest of this section (expenditures as budget shares, prices and total expenditure in logarithmic form) was the one that yielded the best results for the elasticities. This surprising fact might be seen as an confirmation of the view expressed in Hastie *et al* [11] that "there is quite an art in training neural

networks". For the empirical results on Canadian data, we take comfort both in the robustness of the results and in the fact that the elasticities yielded by the neural network closely match those estimated by the QUAIDS model (see section 4).

It is important to note that, contrary to conventional econometrics, *one does not seek parameter identification in feedforward neural networks.* One reason is that the neural network model is not supposed to represent the true data generating process, and that we take no hypotheses concerning the error term. Another reason is that the hidden neurons play symmetric roles in the network, resulting in multiple symmetries in the weights of the network ; therefore the values and variances of single weights are not relevant as they are in an econometric model. Only the outputs of the network (predicted budget shares and predicted elasticities) are relevant.
For this reason, *endogeneity of the regressors (and therefore instrumentation) is not an issue for the estimation of partial derivatives in feedforward neural networks.* To illustrate this, we performed a simulation, running a neural network regressing y on x with the following data generating process :

$$\begin{aligned} y &= 3x + \varepsilon \\ x &= \alpha + \varepsilon \\ \alpha, \varepsilon &\sim N(0,1) \end{aligned}$$

in which it can be seen that x is correlated to ε. As expected, the estimated partial derivative of y in x is around 3.5 instead of 3 (exactly as in linear regression). Whereas in a classical linear regression one wants to retrieve a slope parameter value of 3 (the slope of α in the equivalent model $y = 3\alpha + 4\varepsilon$) using instrumentation to conform with the exogeneity hypothesis on the error term, we argue that this would be counter-productive in a neural network. If the goal of the analysis is to yield the partial derivative of y in x, and we make no assumptions about the true model, then what we want to estimate is the whole effect of x, *ie* the effect of $\alpha + \varepsilon$, not only the effect of α. Following this logic, 3.5 is the correct partial derivative, and instrumentation (which yields a partial derivative of 3) leads to an incorrect estimation. For this reason we do not use instrumentation of the total expenditure term in the following estimations of the neural network demand system.

One last attractive feature of multilayer perceptrons is that they can be modified to handle constraints, linear or non-linear, and thus to test economic hypotheses. Zhang and Constantinides [33] extend the neural network by adding to the cost function a weighted "Lagrange" term representing constraint violation. Running a backpropagation algorithm on this network must then yield the best function approximation that satisfies the constraints. Another solution would be to apply constrained optimization algorithms (see [12] or [13]) to the original network. If we take some additional hypotheses on the distribution of errors and on the fact that the training process did find the global minimum, we can then use a likelihood-ratio test of the unconstrained *vs* the constrained estimation as is done in standard econometrics. The present work makes no use of such constrained neural networks, but it is our feeling that they might be a fruitful line of future research.

3 Model Specification and Econometrics

3.1 Data

We use the microdata of the Survey of Household Spending from Statistics Canada for years 2004 through 2008. Price indexes are drawn from Statistics Canada's Consumer Price Index, using the 2008 Inter-city Indexes of Consumer Price Levels to harmonize price levels between provinces. We aggregated spending categories following the COICOP nomenclature (classification of individual consumption by purpose) which is part of the United Nation's System of National Accounts. This nomenclature comprises twelve categories : Food at home and non-alcoholic beverages; Alcoholic beverages, tobacco and narcotics; Clothing and footwear; Housing, water, electricity, gas and other fuels; Furniture, household equipment and routine household maintenance; Health; Transport; Communication; Recreation and culture; Education; Restaurants and hotels; Miscellaneous goods and services. Transport spendings were disaggregated between personal transport and transport services, leading to a total of 13 categories.

In order to eliminate outliers the datasets were filtered for households with negative total expenditure or negative expenditure for at least one spending category, and for households whose net total expenditure quantile and income quantile differed by more than 30 points. These operations filtered out less than 10% of the sample in each survey, leading to sample sizes of 12,999 for 2004, 13,965 for 2005, 12,894 for 2006, 12,770 for 2007 and 8,820 for 2008, and a total of 61,448 households for the pooled data.

3.2 QUAIDS

We fit the unrestricted QUAIDS model following Banks, Blundell and Lewbel [1] using the following system :

$$w_i = \alpha_i + \sum_{j=1}^{n} \gamma_{ij} \ln p_j + \beta_i \ln \left(\frac{m}{a(\mathbf{p})} \right) + \frac{\lambda_i}{b(\mathbf{p})} \left(\ln \left(\frac{m}{a(\mathbf{p})} \right) \right)^2, \forall i$$

$$\ln a(\mathbf{p}) = \alpha_0 + \sum_{i=1}^{n} \alpha_i \ln p_i + \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \gamma_{ij} \ln p_i \ln p_j$$

$$b(\mathbf{p}) = \prod_{i=1}^{n} p_i^{\beta_i}$$

where w_i is the budget share of good i, p_i the price of good i, $\mathbf{p}$ is a price vector containing all prices, and m is total expenditure. Following [23] α_0 is set to a level somewhat below the minimum log-expenditure in the sample. The α_i are decomposed into an intercept term and control variables : a dummy variable indicating whether the household lives in an urban area and categorical variables for household composition, tenure status, and

household's head education level, so that

$$\alpha_i = \delta_{i0} + \delta_{i1} urban + \sum_{k=2}^{5} \delta_{i1k} hh.comp_k + \sum_{k=2}^{3} \delta_{i2k} tenure_k + \sum_{k=2}^{5} \delta_{i3k} educ_k$$

The system is estimated on the pooled Canadian consumption data using an iterative procedure as recommended in [1]. In the first iteration $\ln a(\mathbf{p})$ and $b(\mathbf{p})$ are initialized using Stone's price index, and the system is estimated conditional to these values. In the next iterations, $\ln a(\mathbf{p})$ and $b(\mathbf{p})$ are updated using the above formula and the system is estimated using these new values, until convergence. We use ordinary least squares equation by equation as the regressors are identical in each equation. To account for possible endogeneity the total expenditure terms are instrumented by the control variables listed above and by the household's head age and squared age and a categorical social status index introduced in [8]. The need for instrumentation is assessed by a Hausman test. Parameter variances are obtained by bootstrapping.

3.3 Neural Network Demand System

To fit our neural network demand system we apply a multilayer perceptron[4] to our consumption data using the following specification :

- the output variables are the budget shares of each class of goods ;
- the input variables are the same as those used for QUAIDS (logarithmic prices, logarithmic total expenditure, a dummy variable indicating if the household lives in an urban area, and categorical variables indicating the household's composition, tenure situation, and education level of the household head) and an additional categorical variable indicating the year of survey. This last variable is added to take inflation into account, which is done implicitly in the QUAIDS estimation procedure. Note that prices and income are taken in logarithmic form for convenience only, as using their initial values instead does not change prediction quality ;
- the neural network consists of the input and output layers and a single hidden layer of ten hidden neurons. The neurons are fully connected, meaning that each neuron is connected to all neurons of the next layer. Trials with different numbers of hidden neurons are detailed below ;
- all neurons are connected to an incoming bias term ;
- the activation function for the hidden neurons is the hyperbolic tangent ;
- the activation function for the output neurons is the identity function ;

[4] The multilayer perceptron was coded in R by the author, with integrated C++ code for faster runtime using the Rcpp package [7]. The resulting program runs much faster than all currently available neural network packages for R, and further work might bring a public release as an R package. Source code available form the author on request.

- input and output variables were standardized as recommended in Sarle [29], and weights were initialized following a uniform distribution on [-0.7, 0.7] as recommended in Hastie *et al* [11] ;
- the neural network is trained using the "RProp"[5] backpropagation procedure as described in [24] with 200 iterations ;
- the training sample is a random sample of 50,000 data points, and the remaining 11,448 points constitute the test sample ;
- 50 runs of training were made, re-drawing a training sample for each run, to assess training robustness.

The choice of the regularization parameter λ (equation 1) is a delicate matter, as its role is to balance variance versus bias. A low value minimizes bias, a high value minimizes variance. In the case of demand analysis, the variance problem manifests itself acutely in the robustness of estimated elasticities : many training runs with no regularization yields very different elasticities. However if λ is set too high, elasticities tend to get closer to one, as the partial derivative term $\frac{\partial w_i}{\partial x}$ tends to get closer to zero[6] in the elasticity formula [7]:

$$El_i^x = 1 + \frac{\partial w_i}{\partial x} \cdot \frac{1}{w_i}$$

We use ten-fold cross-validation to determine the optimal value of λ, using a grid search. In ten-fold cross-validation, the whole sample is divided into ten random sub-samples, and ten iterations are run in which one of the sub-samples is used as a testing sample (used to compute the prediction error) while the other nine are used as the training sample. We find that the value $\lambda = 3$ is the lowest value for which the prediction error starts increasing, and therefore choose this value.

4 Results on Canadian consumption microdata

4.1 Prediction Quality

The estimated QUAIDS model yielded a mean squared error of 0.003578 over the whole sample. Convergence was attained after 5 iterations. Estimation results are presented in tables 3, 4 and 5. Hausman tests using bootstrapped variances strongly rejected equality between instrumented and non-instrumented estimations for all categories except Clothing and Footwear (p-value of 0.15, while all other categories yielded p-values inferior to 10^{-10}), validating the need for instrumentation of the total expenditure terms.

[5] While RProp - which stands for resilient backpropagation - is a relatively old training algorithm by neural network standards, it was shown in Rocha *et al* [26] to be still more effective than most alternatives available in 2003.

[6] A high regularization parameter has the effect of shrinking the network weights to values close to 0, which in turn make the network's partial derivatives close to null functions.

[7] This elasticity formula is specific to the variable specification we used, *viz* consumption expressed as budget shares w_i and prices and total expenditure x in logarithmic form.

For the neural network demand system the best of 50 runs[8] produced a mean squared error of 0.003445 over the test sample, and 0.003365 over the whole sample. The fifty runs showed the robustness of the training procedure (the test-sample-MSE followed a normal distribution of standard deviation 0.00001), and all runs gave results better than QUAIDS in terms of MSE. To determine the optimal number of hidden neurons we used ten-fold cross-validation for numbers of hidden neurons ranging from 1 to 15. Some of the test-sample MSE results are shown in figure 2. Shapiro-Wilks tests did not reject normality of MSE distributions for networks of 5 hidden neurons or more. Two-sample t-tests revealed that 10 hidden neurons gave results significantly different from 9-neuron networks (p-value 0.003), but not significantly different from networks with 11 hidden neurons or more (p-values ranging from 0.4 to 0.9).

This first comparison shows that QUAIDS and NNDS yield mean squared errors of comparable magnitude, with a slight advantage for the neural network when it has two hidden neurons or more. It should be noted that this advantage is due to the flexibility of the neural network, not a higher count of free parameters. Indeed our QUAIDS specification has 351 parameters, which is the number of weights needed by an MLP with 8 hidden neurons ; the chosen 10 hidden neuron specification has 423 independant weights, but a network using only 2 hidden neurons for 95 weights still performs better than QUAIDS. Table 1 shows that both have remarkably similar prediction errors on most categories of goods, except in food at home, for which the NNDS has a clear advantage, and to a lesser degree in furniture, private transport, and miscellaneous goods.

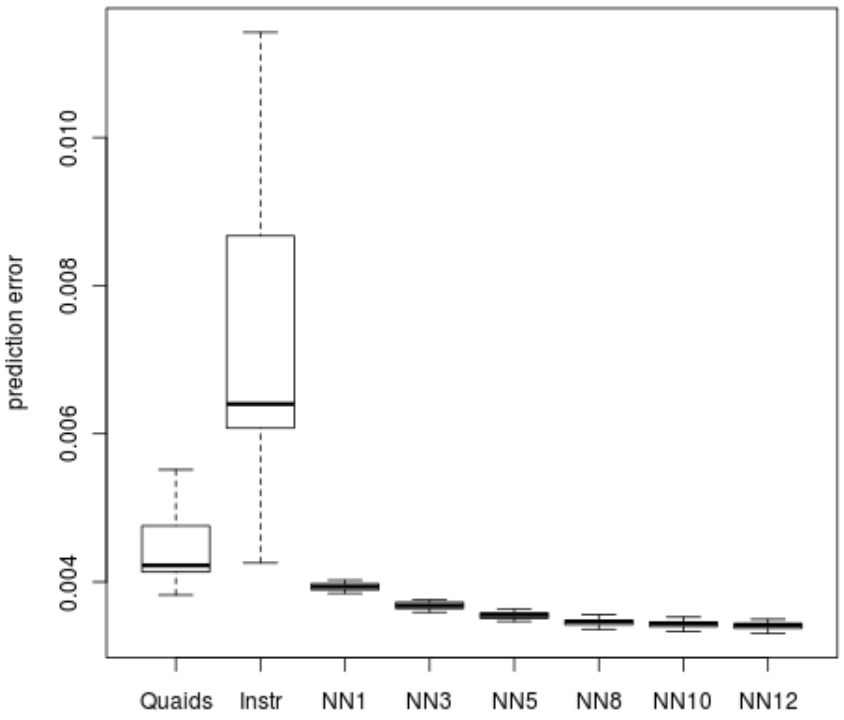
	food	alcohol	clothing	housing	furnish	health	private tr.
quaids	0.0056	0.0018	0.0010	0.0091	0.0025	0.0013	0.0113
nnds	0.0035	0.0018	0.0010	0.0091	0.0025	0.0013	0.0112
	tr. services	communication	recreation	education	restaurant	misc	
quaids	0.0006	0.0003	0.0031	0.0011	0.0016	0.0069	
nnds	0.0006	0.0003	0.0031	0.0011	0.0016	0.0067	

Table 1: Mean Squared Error of QUAIDS and NNDS for each category of goods
Source : Statistics Canada microdata, computations by the author.

In order to get a more just performance comparison of QUAIDS and NNDS we use ten-fold cross-validation for both specifications simultaneously, while measuring prediction error. This puts them both on equal ground, and the resulting prediction error for QUAIDS is found to be significantly higher than the simple mean squared error reported when the whole dataset was used. The results are reported in figure 2. A non-instrumented version of QUAIDS is also included ; comparison between instrumented and non-instrumented versions yields the common result that instrumentation gives better identification at the cost of lower precision and prediction robustness. Comparing QUAIDS to different versions of the NNDS (1, 3, 5, 8, 10 and 12 hidden neurons), we

[8] As described in section 3, for each run a network with a single hidden layer of 10 hidden neurons was trained using 200 iterations of the rprop procedure, after which convergence was attained and test-sample error showed no sign of overlearning.

Figure 2: Comparison of prediction error (by ten-fold cross-validation) for Quaids (with and without instrumentation) and NNDS with different numbers of hidden neurons.



find that prediction errors are close but give a distinct advantage to the neural networks, as their prediction errors are much more stable for different test samples, and a 10 hidden neuron specification always gives better mean error than even a non-instrumented QUAIDS.

The fact that the neural network yields better predictions than QUAIDS should come as no surprise : lowest prediction error is the primary goal of the MLP approach, while in the case of standard econometrics in general (and QUAIDS in particular) the goal is the best identification of parameters. However, the fact that prediction errors of both approaches are relatively close is a great credit to the QUAIDS specification. Indeed if we suppose that the trained weights we have found for the NNDS are close to the optimal weights - and thus that the neural network captures the full relationship linking the input and output variables, giving an upper bound to prediction quality - then we can conclude that the QUAIDS specification is very close to capturing the full effect of the

chosen input variables on consumption behavior. This comes as an empirical validation of the theoretic foundations of the QUAIDS model.

4.2 Income Elasticities and Engel Curves

Table 2 shows the estimated income elasticities for both demand systems. Both systems give credible elasticities, in the expected ranges. In terms of classification between inferior goods, necessities and luxury goods, the results are strictly similar : food at home, alcohol, housing, health and communication are necessities ; clothing, furniture, transport services and education have elasticities close to one, while private transportation, recreation, restaurants & hotels and miscellaneous goods and services are luxuries. The NNDS seems to yield somewhat more conservative elasticities than QUAIDS (*viz* elasticities closer to one), which might indicate that the regularization parameter λ was set too high.

Price elasticities are estimated in the same straightforward manner as income elasticities ; however the results are much less robust, which is why we choose not to reproduce them here. Between different runs price elasticities widely vary, and regularization does not seem to help robustness as it does for income elasticities. We take comfort in the fact that price elasticities were equally badly estimated by QUAIDS, most price coefficients being non significant and own-price elasticities yielding unreasonable values. This result is probably due to the insufficient variation of prices in the sample (one set of prices for each of the five provinces, for each of the five observation years) and the close correlation between price indices for different types of goods.

	food	alcohol	clothing	housing	furnish	health	private tr.
quaids 1st quartile	-0.38	0.56	1.09	0.51	1.05	0.44	1.46
quaids median	0.09	0.78	1.15	0.66	1.13	0.81	1.72
quaids 3rd quartile	0.41	0.90	1.25	0.75	1.30	1.01	2.07
nnds 1st quartile	0.28	0.87	1.10	0.53	1.06	0.71	1.31
nnds median	0.50	0.93	1.15	0.68	1.09	0.82	1.48
nnds 3rd quartile	0.64	0.96	1.23	0.76	1.15	0.90	1.72
	tr. services	comm.	recreation	education	restaurant	misc	
quaids 1st quartile	0.95	0.42	1.29	0.94	1.11	1.24	
quaids median	1.04	0.59	1.45	1.00	1.30	1.43	
quaids 3rd quartile	1.16	0.70	1.71	1.07	1.69	1.83	
nnds 1st quartile	0.99	0.54	1.20	1.05	1.21	1.20	
nnds median	1.00	0.67	1.32	1.10	1.35	1.28	
nnds 3rd quartile	1.00	0.76	1.51	1.22	1.61	1.42	

Table 2: Total expenditure elasticities for all categories, computed using the estimations by quaids and nnds. For each category, datapoints with a budget share lower than 0.01 were excluded.
Source : Statistics Canada microdata, computations by the author.

Figures 3 and 4 plot three types of smoothed Engel curves : one calculated directly from the data, one from the budget shares predicted by QUAIDS, one from those predicted

by NNDS. We can clearly see the roughly quadratic shape of the QUAIDS curves, which is a key hypothesis on which this specification is built - roughly quadratic because our specification allows the parameters to vary according to socio-demographic variables. This quadratic shape is not, however, always shared by the raw data Engel curves. It does in the case of Housing expenditure (figure 4), for which the three curves are very similar. In the case of Food at home however (figure 3), the three curves are only similar for log total expenditures between 10 and 11.5 - which represent 50% of the total sample. Beyond these borders, the forced quadratic shape of the QUAIDS specification seems to make it less accurate than NNDS.

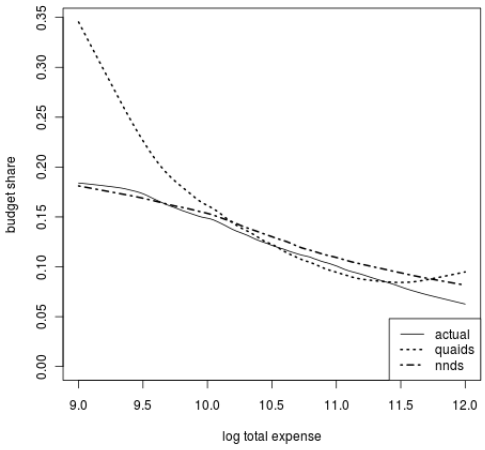


Figure 3: Smoothed Engel curves for Food at home. (Smoothed by lowess with smoothing parameter 0.1)

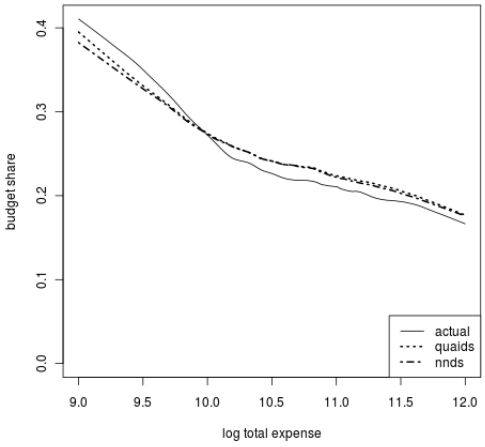


Figure 4: Smoothed Engel curves for Housing. (Smoothed by lowess with smoothing parameter 0.1)

4.3 Testing consumer theory

Alongside estimating demand elasticities, testing consumer theory has long been a main point of interest of demand systems. The goal is to confront the data to the findings of microeconomic theory, and especially to test whether the consumers in a dataset behave rationally, *viz* whether their behavior can be explained by individual maximization of a well-behaved utility function. This is an essential question for theoretical work for obvious reasons, but also for applied work and the construction of usual demand systems themselves, as all interpretations of the parameters of these systems are conditional to the assumption of rationality.

The litterature gives to ways to test for integrability. One is composed of non-parametric tests (see Varian [30] for a survey), in which one seeks to detect choices that contradict one of the axioms of revealed preferences. A limitation of these non-parametric tests is that they need great variations in both income and prices, for if the variation is too low no contradiction is possible (see Varian [31]). The second way

makes use of demand systems, specifically constraining the parameters to make them conform to rationality hypotheses. In our opinion, traditional demand systems have a handicap in this field, due to their many underlying hypotheses. For instance, when testing for integrability using a QUAIDS system, one does not just test whether consumers behave rationally, but whether they behave rationally given that they follow PIGLOG preferences and have quadratic Engel curves. Thus different specifications, with different hypotheses for the specific form of the utility function, will yield different conclusions about the integrability of demand functions. In contrast to this, because it relies on minimal hypotheses, the NNDS might be used to test *just* for rationality. This would go beyond the scope of the present article, but seems worth mentioning. The main difficulty is that neural networks, because their parameters are mostly not identifiable, do not lend themselves easily to statistical tests. However, as mentioned above, there exist some techniques for training constrained networks which could yield interesting results in the case of demand theory. Another possible testing procedure involves setting multiple neural networks in a discrete mixture framework, so as to test whether multiple distinct consumption functions co-exist in the population. This is an interesting field for future research.

5 Concluding remarks

The aim of this article was to compare the quality of the QUAIDS model with that of a new neural network demand system. The results are a credit to both models. Estimation quality is comparable, with a slight advantage for the NNDS. This means that on the one hand, using neural networks we can easily attain state of the art results with no prior knowledge in economics, and on the other hand the QUAIDS model is indeed flexible as it gets results similar to those of a known universal approximator.

However, as stated in section 3, we can never be sure that the training procedure of the neural network has reached the globally optimal set of weights. This means that better training might still get better results, widening the gap between the NNDS and QUAIDS. This raises an interesting point. Theoretically, if a "perfect" training algorithm were found which guaranteed the best set of weights - a highly improbable event -, artificial neural networks could act as benchmarks for empirical problems : the approximator would set an upper bound to the explanatory power of any specification explaining a set of output variables by a set of input variables. One could then test the quality of any specification. Another exciting prospect, as mentioned above, is the possibility of testing economic theory with minimal underlying hypotheses, although much theoretical progress remains to be done in this area.

	food	(sd)	alcohol	(sd).1	clothing	(sd).2	housing	(sd).3
(Intercept)	2.1963	(0.20339)	0.0220	(0.01820)	-0.0253	(0.01326)	0.6747	(0.05852)
ln m	-0.7068	(0.03763)	0.0171	(0.00734)	0.0240	(0.00547)	-0.1030	(0.01813)
(ln m)^2	0.0586	(0.00127)	-0.0025	(0.00074)	-0.0017	(0.00057)	0.0025	(0.00185)
p_food	-1.4585	(0.20981)	-0.1933	(0.06990)	-0.0108	(0.05407)	-0.2014	(0.16055)
p_alco	0.1104	(0.09679)	0.1819	(0.05309)	0.0260	(0.04133)	-0.2112	(0.12162)
p_cloth	0.1711	(0.06754)	0.1077	(0.03478)	-0.1248	(0.02692)	-0.2252	(0.08465)
p_hous	-0.2610	(0.05607)	0.0094	(0.01491)	-0.0410	(0.01200)	-0.0236	(0.03975)
p_furnish	0.4827	(0.11159)	0.0957	(0.06273)	0.0149	(0.04941)	0.2530	(0.15121)
p_health	0.2047	(0.04313)	-0.0262	(0.02013)	-0.0202	(0.01463)	-0.1411	(0.04430)
p_privtr	0.3131	(0.08889)	-0.0440	(0.02449)	0.0269	(0.02010)	-0.0075	(0.06465)
p_trserv	-0.1315	(0.05149)	0.0563	(0.02649)	0.0282	(0.02121)	-0.0594	(0.06297)
p_comm	0.0244	(0.06922)	0.0842	(0.03715)	-0.0805	(0.02690)	-0.0013	(0.08175)
p_recr	-0.1365	(0.07320)	-0.1070	(0.03711)	0.0187	(0.02922)	-0.0430	(0.08952)
p_educ	-0.0032	(0.04622)	-0.0139	(0.02320)	0.0058	(0.01740)	0.1951	(0.05552)
p_resta	0.5447	(0.06754)	0.0732	(0.03469)	0.0295	(0.02863)	0.2899	(0.08406)
p_misc	0.5399	(0.12089)	-0.1147	(0.03831)	-0.0704	(0.03010)	-0.2060	(0.08885)
hh_comp2	0.0140	(0.00112)	-0.0085	(0.00066)	0.0005	(0.00054)	0.0004	(0.00139)
hh_comp3	-0.0179	(0.00115)	-0.0078	(0.00070)	-0.0064	(0.00053)	0.0075	(0.00139)
hh_comp4	-0.0160	(0.00169)	-0.0122	(0.00090)	0.0040	(0.00071)	0.0087	(0.00192)
hh_comp5	-0.1043	(0.00159)	-0.0105	(0.00086)	-0.0106	(0.00062)	0.0377	(0.00180)
tenure2	0.0258	(0.00080)	-0.0071	(0.00049)	-0.0103	(0.00039)	0.0629	(0.00113)
tenure3	0.0281	(0.00088)	-0.0101	(0.00051)	-0.0021	(0.00036)	-0.0745	(0.00107)
urban	0.0041	(0.00081)	-0.0020	(0.00049)	0.0033	(0.00032)	0.0334	(0.00098)
educ2	0.0097	(0.00113)	-0.0017	(0.00066)	0.0029	(0.00042)	-0.0064	(0.00135)
educ3	0.0141	(0.00128)	-0.0004	(0.00079)	0.0021	(0.00049)	-0.0137	(0.00151)
educ4	0.0148	(0.00121)	-0.0068	(0.00070)	0.0032	(0.00047)	-0.0109	(0.00151)
educ5	0.0196	(0.00115)	-0.0117	(0.00063)	0.0040	(0.00046)	-0.0028	(0.00144)

Table 3: Estimated parameters of the QuAIDS model for food, alcohol, clothing and housing expenditures, on pooled Canadian microdata from 2004 to 2008 (standard deviations in parentheses). Instruments for the total expenditure terms : composite social class index (IMPR) dummies, age and squared age of the head of the household.

	furnish	(sd)	health	(sd).1	private tr.	(sd).2	tr. services	(sd).3
(Intercept)	-0.1381	(0.02525)	-0.1245	(0.02658)	-0.4266	(0.08213)	0.0897	(0.01318)
ln m	0.0696	(0.00897)	0.0709	(0.00876)	0.1432	(0.02049)	-0.0316	(0.00460)
(ln m)^2	-0.0060	(0.00089)	-0.0077	(0.00073)	-0.0053	(0.00192)	0.0032	(0.00044)
p_food	0.3042	(0.08367)	0.3604	(0.06803)	0.3557	(0.17882)	0.0441	(0.04417)
p_alco	-0.0703	(0.06302)	-0.1722	(0.04993)	0.0448	(0.13174)	-0.0846	(0.03395)
p_cloth	0.0420	(0.04163)	-0.1214	(0.03124)	-0.0119	(0.08726)	0.0206	(0.02220)
p_hous	0.0409	(0.01855)	0.0013	(0.01538)	0.0430	(0.04129)	-0.0126	(0.01037)
p_furnish	-0.2588	(0.07604)	-0.2450	(0.05945)	-0.0897	(0.15797)	0.0214	(0.04084)
p_health	-0.0115	(0.02327)	-0.0506	(0.01666)	0.0888	(0.05151)	0.0245	(0.01170)
p_privtr	-0.0012	(0.03045)	0.0634	(0.02622)	-0.2169	(0.06874)	0.1098	(0.01723)
p_trserv	0.0550	(0.03220)	-0.0632	(0.02473)	0.0684	(0.06869)	-0.0530	(0.01754)
p_comm	-0.0480	(0.04282)	-0.0638	(0.03108)	-0.1578	(0.08857)	0.0212	(0.02318)
p_recr	0.1128	(0.04491)	0.1193	(0.03699)	0.0386	(0.09255)	0.0785	(0.02469)
p_educ	-0.0144	(0.02682)	0.0494	(0.02042)	-0.0936	(0.06037)	-0.0190	(0.01510)
p_resta	-0.2713	(0.04332)	-0.0722	(0.03012)	-0.1780	(0.08993)	0.0029	(0.02181)
p_misc	0.0845	(0.04555)	-0.0414	(0.03551)	0.2388	(0.10119)	-0.0417	(0.02389)
hh_comp2	0.0092	(0.00072)	-0.0028	(0.00047)	-0.0006	(0.00166)	-0.0046	(0.00041)
hh_comp3	0.0043	(0.00073)	0.0046	(0.00052)	0.0170	(0.00170)	-0.0028	(0.00041)
hh_comp4	0.0171	(0.00106)	-0.0057	(0.00060)	-0.0006	(0.00208)	-0.0033	(0.00051)
hh_comp5	0.0063	(0.00088)	0.0043	(0.00068)	0.0200	(0.00190)	-0.0022	(0.00050)
tenure2	0.0016	(0.00059)	0.0012	(0.00037)	-0.0158	(0.00124)	-0.0107	(0.00030)
tenure3	0.0077	(0.00057)	0.0135	(0.00045)	0.0256	(0.00119)	-0.0066	(0.00029)
urban	-0.0054	(0.00055)	-0.0023	(0.00041)	-0.0321	(0.00124)	0.0050	(0.00022)
educ2	-0.0019	(0.00062)	-0.0088	(0.00055)	-0.0108	(0.00141)	0.0031	(0.00031)
educ3	0.0003	(0.00079)	-0.0084	(0.00065)	-0.0104	(0.00172)	0.0025	(0.00037)
educ4	0.0017	(0.00073)	-0.0081	(0.00060)	-0.0186	(0.00163)	0.0044	(0.00036)
educ5	0.0014	(0.00070)	-0.0080	(0.00059)	-0.0437	(0.00157)	0.0107	(0.00038)

Table 4: Estimated parameters of the QuAIDS model for household equipment, health, private transport and transport services expenditures, on pooled Canadian microdata from 2004 to 2008 (standard deviations in parentheses). Instruments for the total expenditure terms : composite social class index (IMPR) dummies, age and squared age of the head of the household.

	communication	(sd)	recreation	(sd).1	education	(sd).2	restaurant	(sd).3	misc	(sd).4
(Intercept)	0.0950	(0.00812)	-0.0695	(0.02537)	-0.1131	(0.02057)	-0.2883	(0.03491)	-0.8923	(0.09820)
ln m	-0.0136	(0.00291)	0.0217	(0.01037)	0.0465	(0.00637)	0.1173	(0.00848)	0.3447	(0.02111)
(ln m)^2	0.0001	(0.00029)	0.0006	(0.00110)	-0.0042	(0.00059)	-0.0101	(0.00068)	-0.0276	(0.00154)
p_food	-0.0661	(0.02779)	0.0252	(0.09385)	0.1245	(0.05797)	0.3173	(0.07407)	0.3988	(0.15190)
p_alco	0.0514	(0.02144)	0.0364	(0.07122)	-0.0135	(0.04379)	-0.1020	(0.05342)	0.2030	(0.10134)
p_cloth	0.0196	(0.01451)	0.0475	(0.04682)	0.0154	(0.02776)	-0.0204	(0.03613)	0.0797	(0.06938)
p_hous	-0.0029	(0.00621)	0.0337	(0.01997)	0.0328	(0.01373)	0.0344	(0.01775)	0.1456	(0.03686)
p_furnish	0.0435	(0.02607)	-0.0372	(0.08547)	-0.0465	(0.05405)	-0.1126	(0.06299)	-0.1214	(0.12359)
p_health	-0.0100	(0.00786)	0.0635	(0.02567)	-0.0442	(0.01579)	-0.0493	(0.01802)	-0.0285	(0.03888)
p_privtr	0.0081	(0.01065)	-0.0270	(0.03414)	-0.0139	(0.02336)	0.0445	(0.02878)	-0.2553	(0.05859)
p_trserv	0.0179	(0.01109)	0.0142	(0.03648)	0.0195	(0.02266)	-0.0607	(0.02751)	0.1083	(0.05314)
p_comm	-0.0033	(0.01531)	0.0968	(0.04730)	-0.0030	(0.02917)	0.0366	(0.03591)	0.0946	(0.07180)
p_recr	-0.0080	(0.01535)	0.0297	(0.05020)	0.0150	(0.03197)	0.0470	(0.03868)	-0.1650	(0.07168)
p_educ	0.0084	(0.00959)	-0.0802	(0.03178)	-0.0007	(0.01964)	0.0098	(0.02354)	-0.0435	(0.04672)
p_resta	-0.0096	(0.01446)	-0.1141	(0.04588)	-0.1057	(0.02792)	-0.0240	(0.03516)	-0.1652	(0.06811)
p_misc	0.0062	(0.01538)	0.0096	(0.05217)	-0.0156	(0.03074)	-0.1842	(0.04105)	-0.2048	(0.08874)
hh_comp2	-0.0013	(0.00025)	0.0084	(0.00083)	-0.0009	(0.00071)	-0.0065	(0.00062)	-0.0073	(0.00120)
hh_comp3	-0.0034	(0.00027)	0.0044	(0.00085)	-0.0129	(0.00070)	-0.0008	(0.00064)	0.0143	(0.00127)
hh_comp4	0.0011	(0.00038)	0.0143	(0.00103)	-0.0039	(0.00092)	-0.0022	(0.00079)	-0.0013	(0.00156)
hh_comp5	-0.0043	(0.00034)	0.0110	(0.00094)	-0.0096	(0.00077)	0.0124	(0.00080)	0.0498	(0.00150)
tenure2	-0.0012	(0.00021)	-0.0117	(0.00062)	-0.0077	(0.00041)	-0.0120	(0.00048)	-0.0149	(0.00095)
tenure3	0.0006	(0.00021)	0.0055	(0.00058)	-0.0033	(0.00038)	-0.0029	(0.00047)	0.0186	(0.00098)
urban	-0.0016	(0.00019)	-0.0003	(0.00064)	0.0024	(0.00029)	0.0008	(0.00040)	-0.0053	(0.00093)
educ2	0.0024	(0.00025)	0.0074	(0.00071)	0.0046	(0.00037)	0.0035	(0.00053)	-0.0040	(0.00111)
educ3	0.0022	(0.00029)	0.0087	(0.00089)	0.0019	(0.00036)	0.0026	(0.00065)	-0.0015	(0.00133)
educ4	0.0030	(0.00028)	0.0097	(0.00085)	0.0052	(0.00043)	0.0032	(0.00060)	-0.0008	(0.00127)
educ5	0.0023	(0.00026)	0.0079	(0.00083)	0.0114	(0.00047)	0.0060	(0.00060)	0.0029	(0.00125)

Table 5: Estimated parameters of the QuAIDS model for communication, leisure, education, restaurant and hotels, and miscellaneous expenditures, on pooled Canadian microdata from 2004 to 2008 (standard deviations in parentheses). Instruments for the total expenditure terms : composite social class index (IMPR) dummies, age and squared age of the head of the household.

References

[1] J. Banks, R. Blundell, and A. Lewbel. Quadratic engel curves and consumer demand. *Review of Economics and Statistics*, 79(4):527–539, 1997.

[2] G. A. Darbellay and M. Slama. Forecasting the short-term demand for electricity: Do neural networks stand a better chance? *International Journal of Forecasting*, 16(1):71–83, 2000.

[3] J.F.G. de Freitas. *Bayesian methods for neural networks*. PhD thesis, Cambridge University, Trinity College and Engineering Department, 2000.

[4] A. Deaton. Demand analysis. In Z. Griliches and M. D. Intriligator, editors, *Handbook of Econometrics, edition 1*, volume 3, chapter 30, pages 1767–1839. Elsevier, 1986.

[5] A. Deaton and J. Muellbauer. An almost ideal demand system. *The American Economic Review*, 70(3):312–326, 1980.

[6] A. Decoster and F. Vermeulen. Evaluation of the empirical performance of two-stage budgeting aids, quaids and rotterdam models based on weak separability. Technical report, Center for Economic Studies, Catholic University of Leuven, 1998. Discussion Paper DPS 98.07.

[7] D. Eddelbuettel and R. Francois. Rcpp: Seamless R and C++ Integration. *Journal of Statistical Software*, 40(8):1–18, 2011.

[8] F. Gardes, P. Gaubert, and S. Langlois. Pauvreté et convergence des consommations au Canada. *Canadian Review of Sociology and Anthropology / Revue Canadienne de Sociologie et d'Anthropologie*, 2000.

[9] M. W. Gardner and S. R. Dorling. Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences. *Atmospheric Environment*, 32(14–15):2627–2636, 1998.

[10] R. F. Harrison and R. L. Kennedy. Artificial neural network models for prediction of acute coronary syndromes using clinical data from the time of presentation. *Annals of Emergency Medicine*, 46(5):431–439, 2005.

[11] T. Hastie, R. Tibshirani, and J. Friedman. *The elements of statistical learning : data mining, inference and prediction*. Springer, second edition, 2009.

[12] X. Hu and R. C. Eberhart. Solving constrained nonlinear optimization problems with particle swarm optimization. In *Proceedings of the Sixth World Multiconference on Systemics, Cybernetics and Informatics*, 2002.

[13] D. Karaboga and B. Basturk. Artificial bee colony (ABC) optimization algorithm for solving constrained optimization problems. *Advances in Soft Computing: Foundations of Fuzzy Logic and Soft Computing*, 4529/2007(0302-9743):789–798, 2007.

[14] D. Karaboga and B. Basturk. A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. *Journal of Global Optimization*, 39(3):459–471, 2007.

[15] N. K. Kasabov. *Foundations of Neural Networks, Fuzzy Systems, and Knowledge Engineering*. MIT Press, 1996.

[16] J. Kennedy and R. Eberhart. Particle swarm optimization. In *Proceedings of the IEEE International Conference on Neural Networks IV*, page 1942–1948, 1995.

[17] C. M. Kuan and H. White. Artificial neural networks : An econometric perspective. *Econometric Reviews*, 13:1–91, 1994.

[18] R. Law. Back-propagation learning in improving the accuracy of neural network-based tourism demand forecasting. *Tourism Management*, 21(4):331–340, 2000.

[19] A. Lewbel and K. Pendakur. Tricks with Hicks : the EASI demand system. *The American Economic Review*, 99(3):827–863, 2009.

[20] M. McAleer, M. C. Medeiros, and D. Slottje. A neural network demand system with heteroskedastic errors. *Journal of Econometrics*, 147:359–371, 2008.

[21] R. M. Neal. *Bayesian learning for neural networks*. PhD thesis, University of Toronto, Department of Computer Science, 1995.

[22] P. C. Pandey and S. V. Barai. Multilayer perceptron in damage detection of bridge structures. *Computers & Structures*, 54(4):597–608, 1995.

[23] P. Pashardes. Bias in estimating the almost ideal demand system with the stone index approximation. *The Economic Journal*, 103(419):908–915, 1993.

[24] M. Riedmiller and H. Braun. A direct adaptive method for faster backpropagation learning: The rprop algorithm. In H. Ruspini, editor, *Proceedings of the IEEE International Conference on Neural Networks (ICNN) 1993*, pages 586–591, 1993.

[25] B. D. Ripley. *Pattern Recognition and Neural Networks*. Cambridge University Press, 1996.

[26] M. Rocha, P. Cortez, and J. Neves. Evolutionary neural network learning. *Progress in Artificial Intelligence*, page 24–28, 2003.

[27] R. Rojas. *Neural networks*. Springer-Verlag, 1996.

[28] D. E. Rumelhart and J. L. McClelland. *Parallel Distributed Processing : Explorations in the Microstructure of Cognition. Volume 1. Foundations*. MIT Press, 1986.

[29] W. S. Sarle. Neural network FAQ. ftp://ftp.sas.com/pub/neural/FAQ.html, 1997. Consulté en février 2012.

[30] H. Varian. Revealed preference. In M. Szenberg, L. Ramrattan, and A. A. Gottesman, editors, *Samuelsonian economics and the 21st century*. Oxford university press, 2006.

[31] H. Varian. *Introduction à la microéconomie*. de boeck, 6 edition, 2008.

[32] H. Working. Statistical laws of family expenditure. *Journal of the American Statistical Association*, 38(221):43–56, 1943.

[33] S. Zhang and A. G. Constantinides. Lagrange programming neural networks. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, 39(7):441–52, 1992.