Available at
www.ElsevierMathematics.com
POWERED BY SCIENCE DIRECT®

Computational Statistics & Data Analysis 44 (2004) 547 – 570

COMPUTATIONAL STATISTICS & DATA ANALYSIS

www.elsevier.com/locate/csda

Comparison of the performance of multi-layer perceptron and linear regression for epidemiological data

Jean Gaudart*, Bernard Giusiano, Laetitia Huiart

Department of BioMathematics, Medical Statistics, Informatics, Faculty of Medicine, University of Aix-Marseille II, 27 bd J. Moulin 13385 Marseille Cedex 05, France

Received 1 March 2001; received in revised form 1 August 2002

Abstract

Neural networks are used increasingly as statistical models. The performance of multilayer perceptron (MLP) and that of linear regression (LR) were compared, with regard to the quality of prediction and estimation and the robustness to deviations from underlying assumptions of normality, homoscedasticity and independence of errors. Taking into account those deviations, five designs were constructed, and, for each of them, 3000 data were simulated. The comparison between connectionist and linear models was achieved by graphic means including prediction intervals, as well as by classical criteria including goodness-of-fit and relative errors. The empirical distribution of estimations and the stability of MLP and LR were studied by re-sampling methods. MLP and linear regression had comparable performance and robustness. Despite the flexibility of connectionist models, their predictions were stable. The empirical variances of weight estimations result from the distributed representation of the information among the processing elements. This emphasizes the major role of variances of weight estimations in the interpretation of neural networks. This needs, however, to be confirmed by further studies. Therefore MLP could be useful statistical models, as long as convergence conditions are respected.

Keywords: Neural networks; Perceptron; Linear regression; Prediction; Estimation

1. Introduction

Neural networks are modeling tools for neurophysiology and artificial intelligence (Reggia, 1993). They are also used as statistical models instead of classical approaches.

* Corresponding author. Tel.: +33-(0)4-91-79-19-10; fax: +33-(0)4-91-79-40-13.
E-mail address: jean.gaudart@medecine.univ-mrs.fr (J. Gaudart).

0167-9473/$ - see front matter © 2002 Elsevier B.V. All rights reserved.
PII: S0167-9473(02)00257-8

In the medical field, they are applied to an increasing range of epidemiological problems, using mostly multi-layer perceptron. Examples of applications of neural networks to medical data include Baxt (1995); Bottaci et al. (1997); Cross et al. (1995); Fogel et al. (1998); Guh et al. (1998); Lapeer et al. (1995); Ottenbacher et al. (2001); Sonke et al. (2000).

Many epidemiological studies provide insufficient information regarding the statistical properties of the covariates studied, such as the normality of their distribution patterns, or the existence of colinearity. In addition there is often no information about the link function between variables, and the linearity has to be assumed. In recent epidemiological studies using neural networks, multi-layer perceptron (MLP) appears to be a solution to those problems, as it has been proven that three-layer perceptron networks are theoretically universal approximators (Hornik et al., 1989). Moreover, some works suggest that they can match or exceed the performance of classical statistical methods regarding their goodness of fit and their estimation and prediction capability. Connectionist models are seen as flexible methods, and used as a generalization of regression methods (Mariani et al., 1997). But their frequent utilization as “black boxes” is controversial (Schwarzer et al., 2000). Thus the use of connectionist models as a particular class of statistical models is an ongoing problem (Flexer, 1996).

Neural networks are directed and weighted graphs, where nodes are processing elements (PEs) with an inner state called activation. Those PEs are usually arranged in layers and are connected to many PEs in other layers via directed arcs. Associated with each connection is a real-valued weight w_{ij}. Each PE processes the input vector X it receives via these connections. Usually, this process consists in transformation of this input vector by an activation function $h(W;X)$, and then by a transfer function $g(h(W;X))$. The simplest PE possible has a linear activation function and an identity transfer function. Its output is

$$y_{\mathrm{L}} = g(h(X)) = g(W^{\mathrm{T}}X) = W^{\mathrm{T}}X.$$

After this process, the PE provides the continuous value y_{L}, called local output value, to other PEs via its outgoing weighted connections. In feed-forward models, connections run forward from input to hidden PEs and from hidden PEs to output ones. Deciding how the PEs are connected, how the PEs process their information and how the connection weights are estimated all contribute to creating a neural network. Those models depend on the number, the design and the connections of the PEs (architecture), and also depend on the weights as well as their estimation methods (learning rules). In other words, the architecture specifies the model; weights between processing elements are the parameters of the model and the learning rule is the estimation method.

Each PE analyzes one part of the problem, thus the information is distributed among the processing elements. The output of the network, $Y = f_W(X)$, is a combination of local functions. This combination depends on the number of hidden PEs, and on the different classes of activation and transfer functions. Thus, complex overall behavior could result from simple local behavior.

To study the statistical behavior of neural networks, it is necessary to compare them to classical tools, by formal comparisons and simulations. During the last few years several comparisons have been published, including logistical models (Schumacher et al., 1996; Vach et al., 1996), principal components extractions (Nicole, 2000), time series analysis (Lisi and Schiavo, 1999) and autoregressive models (Tian et al., 1997), Cox regressions (Xiang et al., 2000). The conditions for use and the robustness of connectionist models are two frequently asked questions (Cheng and Titterington, 1994; Capobianco, 2000). Our study is in keeping with those works and focuses on comparison between MLP and linear regression.

The aim of this study is to provide a comparative evaluation of those two methods for simulated data sets and variables. The quality of the models and the effect of deviations from underlying assumptions of normality, independence and homoscedasticity are compared regarding their ability to predict and estimate.

The paper is organized as follows. The models are described in the following section. The learning rules, and some procedures to improve them, are presented in Section 3. In Section 4, the simulations are described. In Section 5 we report on the experimental results obtained. Finally, the discussion section analyzes the comparisons and the estimations, and attempts to discuss the conditions for use and interpretation of the MLP as a statistical model.

2. Models

Linear regression, which is a well-known statistical model, was used for modeling simulated data. Whereas connectionist models use heuristic and iterative algorithm, linear regression uses more formalized solutions (not iterative). The estimation of the coefficients of the linear equation has been formalized by Gauss using the least square estimations, which are unbiased and have the least variance among all unbiased estimators.

If $Y = \beta X + \varepsilon$, then a good estimator of the vector β of the coefficients is given by

$$b = (X^{\mathrm{T}}X)^{-1}X^{\mathrm{T}}Y,$$

and its variance-covariance matrix is given by

$$V(b) = \sigma^2(X^{\mathrm{T}}X)^{-1}.$$

It is proven that for one coefficient b_j, the Wald statistics, $b_j/v(b_j)$, follows a student distribution, if we assume the normal distribution of the covariates.

For prediction, we can use the equation $Y_{\mathrm{p}} = X_{\mathrm{p}}^{\mathrm{T}}B$, where Y_{p} is normally distributed with mean $X_{\mathrm{p}}^{\mathrm{T}}\beta_{\mathrm{p}}$ and variance $\sigma\sqrt{(X_{\mathrm{p}}^{\mathrm{T}}(X^{\mathrm{T}}X)^{-1}X_{\mathrm{p}})}$.

So we know the distribution patterns of the coefficients of the LR and the distribution of the predictions, which are well described elsewhere (e.g. Saporta, 1990). To analyze the robustness of the model, the underlying assumptions (normality, independence and homoscedasticity of the errors) were not always respected when simulating the data (see Section 4). The coefficients of regressions were estimated using the ordinary least-square error estimator (OLSE) or the weighted least-square error estimator (WLSE) for the heteroscedastic context of simulation. Statistical significance was set at 0.05. The linear models were performed using the statistical software SPSS 10.0.5 (SPSS Inc. 1999, Chicago IL).

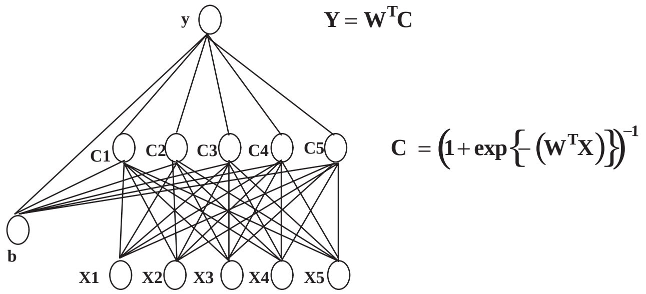

Fig. 1. Scheme of a feed-forward multi-layer perceptron.

For connectionist modeling, the architecture of the network as well as the algorithms of minimization and optimization need to be specified. Of the many connectionist models, such as Radial Basis Function networks or wavelet networks, the multi-layer perceptron (MLP) is one that has been extremely useful in many applications and has been extensively analyzed theoretically. In this article, we focused on the MLP model that is the most commonly used connectionist model in the medical field. But a further step in this work will be to compare other neural network models to classical statistical models in order to obtain a more exhaustive approach. We performed a perceptron model with one hidden layer, and a learning algorithm based on the delta-rule (see Section 3). From the theoretical viewpoint, the approximation properties of this delta-rule are shown in Gascuel (1997) and Hornik et al. (1989). The MLP with a single hidden layer, using squashing transfer functions (non-decreasing and bounded functions) can approximate, to any desired degree of accuracy, any measurable function. These theoretical results show the existence of MLP networks. The network was performed using the connectionist software Neural Works Professional II/plus, (NeuralWare Inc. 1993, Pittsburgh PA). The starting architecture, shown in Fig. 1, was performed as follows:

- An input layer with 5 PEs, for the 5 input covariates. Those first neurons were obviously only input buffers.
- A hidden layer consisting initially in 5 PEs, with linear activation functions (weighted sums), logistical transfer functions, and learning capabilities.
- An output layer, consisting in a single PE, with a linear activation function (weighted sum), an identity transfer function, and learning capability.

One extra PE, called Bias, was connected to the hidden and output layers. This neuron has no input, and a constant output equal to 1. This constant is similar to the intercept coefficient in standard statistical models. The data are randomly presented to the network. Thus, every output y is computed, for every input vector, as follows:

$$y = \sum_{j=0}^{5} w_{jk} c_j = \sum_{j=0}^{5} w_{jk} \left(1 + \exp\left\{ - \sum_{i=0}^{5} w_{ij} x_i \right\} \right)^{-1},$$

where w_{jk} are the weights between each hidden PE c_j and the single output PE k; w_{ij} are the weights between an input PE x_i and a hidden PE c_j; x_0 is the bias output associated with the input layer; c_0 is the bias output associated with the hidden layer; (here, $x_0 = c_0 = 1$).

3. Learning rule

The delta-rule is a supervised learning rule used for minimization (Medler, 1998; Vapnik and Bottou, 1993). This method is often effective. It is a first-order steepest-descent-like method for non linear data that has been used extensively for the training of neural networks (Bertsekas, 1996). The aim of this method is to find the matrix W of weights so that W minimizes the error function $E(D, Y)$. D is the desired output $D = f_W(X)$, and Y is the obtained output $Y = f_{\hat{W}}(X)$. This estimation of the weights matrix is given by

$$\hat{W} = \arg\min_W (E(D, Y)), \quad \text{with } E(D, Y) = \frac{1}{2}\|D - Y\|_2^2.$$

The error is used to adjust the weights in the network so that when the same input vector X is presented again, the network produces a response slightly closer to the desired response D. The weights are modified at each iteration, for every input vector X, as follows:

$$w(t+1) = w(t) - \Delta w(t), \quad \text{with } \Delta w(t) = \eta \frac{\partial E}{\partial w}.$$

This instantaneous gradient has convergence properties, and is easy to use. If we have an activation function such as

$$h(X) = W^{\mathrm{T}} X,$$

and a transfer function such as

$$g(X) = (1 + \exp\{-W^{\mathrm{T}} X\})^{-1},$$

then, for an input vector X, we have an easy solution:

$$\varDelta_X w_{ij} = \eta \delta_{X_j} y_{X_i},$$

where y_{X_i} is the local output for each PE i, η is a real value called learning rate.

For a given vector X, the δ_{X_j} coefficient attached to an output PE j is written as follows:

$$\delta_{X_j} = (d_{X_j} - y_{X_j}) g_j'(h_{X_j}),$$

and for a hidden PE i:

$$\delta_{X_i} = \sum_{j=1}^{S} (\delta_{X_j} \times w_{ij}) g_j'(h_{X_j}).$$

Back-propagating the error could produce defaults on generalization. Those defaults are frequently due to learning problems. A local minimum on the surface of the error produces an estimation bias of the weight matrix. A bad specification of the architecture or bad choices in the learning parameters leads to learning defaults or to over-learning (over-fitting). To optimize convergence and generalization we have modified the delta-rule, using five heuristics (the algorithms for optimization are more formally described elsewhere (e.g. Gallinari, 1997)). With the first heuristic, called the cumulative delta-rule, an accumulative period called epoch is defined. The errors accumulate over the number of iterations set by the epoch. The weights are updated after one epoch using the cumulative gradient. Thus, this learning rule depends less on the size of the learning set, and improves the rapidity of convergence. The epoch size has to be chosen between pattern and batch mode of learning: not too weak to avoid oscillations of estimations, and not too strong to protect convergence.

The minimizations of the error function during the previous epochs are altered by the new modification of weights, at the current epoch. To keep one part of the successive minimizations we used inertial correction. Two coefficients are introduced: a learning rate, and a tendency parameter called momentum. The learning rate η decreases by rate ρ at each πth learning epoch. The choice of this learning rate is a frequently raised problem. To modify weights according to a linear function of partial derivatives, the error function is assumed to be locally linear. The learning rate quantifies this “locality”. When a high learning rate improves convergence, it increases the risk that the error stays within a local minimum. So a second inertial correction is added. The momentum μ is a tendency term introduced into the backpropagation algorithm. One part of the previous gradient is used to compute the current one. Thus, weights are modified according to the direction defined by all previous modifications. Weights are modified according to the following computation:

$$\Delta_X w_{ij}(t) = \eta \times \delta_{X_j} \times y_{X_i}(t) + \mu[\Delta_X w_{ij}(t-1)].$$

Those learning parameters have to be chosen according to previous work and to experience.

Training with noise is an effective approach for smoothing the estimator (Intrator, 1999). With the addition of a Gaussian noise, the output is provided by an input space around the input vector. It does improve convergence by partially resolving the problem of local minima on the error space. Thus, the network approximates a smoothing function between input and output vectors (Gallinari, 1997).

With the basic learning algorithm, the error decreases continuously to a minimum (minimizing the bias of the estimation). But after many learning iterations, the network loses its capability of generalization (increasing the variance of the estimation). This is known as the trade-off between variance and bias (Intrator, 1999). An algorithm was introduced to find the appropriate number of learning iterations. The test set is used to check, at specified iterations, the capability of generalization during the learning phase. Thus, 4 parameters have to be chosen: the maximum number of iterations, iterations at which the model is to be tested, the criterion of decision (here the root mean square error for the test sample), and the minimum number of “bad” test results, as defined by the criterion, before learning stops.

When it has enough hidden processing elements, the network finds unbiased estimations, but capabilities of prediction are not preserved (Fogelman Soulié, 1997). In

```
Repeat
    Prune list ←∅
    Learn and test the NN
    Performance_level ← global error made on the test set
    For i ←1 to the number of PEs
         For $\text{Output}_i$ ←0 to 1
                   Learn and test the NN
                   If performance ≥ 0.95*Performance_level
                   Then add $\text{PE}_i$ to the prune list
         Next $\text{Output}_i$
    Re-enable $\text{PE}_i$
    Next i
    Sort the prune list into decreasing order based on performance
    Disable the first PE of the prune list
As long as a PE remains in the prune list.
```

Fig. 2. Pruning algorithm.

many linear models, the Fisher information matrices are always positive definite under general conditions. In MLP, however, the Fisher information matrix can be singular and, therefore, many statistical techniques based on asymptotic theory cannot be applied properly. Fukumizu has proven that the Fisher information matrix of a three-layer perceptron is positive definite if and only if the MLP is reducible. That is if there is no hidden unit that makes no contribution to the output and there is no pair of hidden units that could be collapsed into a single unit without altering the performances (Fukumizu, 1996). That implies that if a Fisher information matrix of a MLP is singular, we should search redundant hidden units and prune them until the MLP becomes irreducible. The pruning algorithm approximates the error surface around the minimum, then analyzes disturbances in the network when removing PEs. This heuristic prunes the PEs whose removal does not entail much disturbance. The pruning algorithm has two steps: a learning procedure up to minimization, and a pruning step to remove non-useful PEs. Usefulness of a PE is defined by the variation of the global error E when the local output is set to zero. Thus, the usefulness of the analyzed PE is computed with an additional pass of the backpropagation algorithm. In addition, the same test is performed with the output of the analyzed PE set to 1. Fig. 2 shows the pruning algorithm.

4. Simulation

We generated 5 models, with variations in the distribution of errors, and in co-linearity. For each model, we simulated one set of 3 samples, with 1000 subjects each.

- Designs 1, 2 and 5:

$$Y = \alpha + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \beta_4 X_4 + \beta_5 X_5 + \varepsilon = \alpha + \beta^{\mathrm{T}} X + \varepsilon$$

Table 1
Parameter values

Parameters	α	β_1	β_2	β_3	β_4	β_5	γ	v
Values	1.17	−0.66	2.98	2.14	0	0.03	0.2	0.9

Table 2
Characteristics of the covariates X

	Mean	Variance	Minimum	Maximum
X_1	8.882	7.046	0.638	17.454
X_2	5.554	0.162	4.268	6.947
X_3	6.922	5.376	−0.662	13.166
X_4	10.409	0.0009	10.329	10.489
X_5	8.361	10.945	−0.693	18.051

where errors were generated according to a Normal distribution (with a mean $m = 0$, and a variance $v = 1$ (design 1)), or to a Uniform distribution on [0,1] (design 5), or to a Normal heteroscedastic distribution (design 2).

- Design 3, with a term of interaction between 2 covariates:

$$Y = \alpha + \beta^{\mathrm{T}}X + \gamma X_3X_5 + \varepsilon,$$

where errors were generated according to a Normal distribution ($m = 0,\ v = 1$).

- Design 4, with autoregressive and mobile average parameters (AR3MA3):

$$Y = \alpha + \beta^{\mathrm{T}}X + AR3MA3.$$

Coefficients of those designs, shown in Table 1, were generated according to a Uniform distribution on $[-5; 5]$, apart from β_4. To study the ability of MLP and linear regression for modeling non-explanatory covariates, β_4 was set to 0 (thus, to obtain a non-explanatory interpretation of the fourth covariate X_4).

According to a Normal distribution, 5 covariates X_i were generated. The means (m) and standard deviations (SD) were generated according to a Uniform distribution on $[-12; 12]$ (absolute value for SD).

Errors were generated according to the specified designs. Table 2 shows the characteristics of the covariates X.

The variables Y were simulated for each design, using the 5 covariates, the error, and for the AR3MA3 design, the auto-correlation term. Table 3 shows the characteristics of the variables Y.

The procedure was divided into 3 steps: the learning step, the test step (those 2 steps were associated here in one single procedure using the optimization algorithm), and the prediction step. For each step, 3 samples, with 1000 subjects each, were simulated. At the first step, the network estimated weights, according to the learning sample (and to the test sample for the optimization heuristic). An input vector (5 values) for each subject was presented to the input layer. During the estimation of those weights, their generalization capabilities were verified with the test sample. When the results were

Table 3
Characteristics of the variables Y

Design	$g(X)$	Mean	Variance	Minimum	Maximum
1	$Y=\alpha+\beta^{\mathrm{T}}X+\varepsilon$ $\varepsilon\sim \mathrm{N}[0;1]$	26.932	31.019	10.349	43.665
2	$Y=\alpha+\beta^{\mathrm{T}}X+\varepsilon$ $\varepsilon\sim \mathrm{N}[0;v^2\times(f(X))]$	26.98	35.105	10.093	47.086
3	$Y=\alpha+\beta^{\mathrm{T}}X+\gamma X_3X_5+\varepsilon$ $\varepsilon\sim \mathrm{N}[0;1]$	38.518	111.358	8.923	77.447
4	$Y=\alpha+\beta^{\mathrm{T}}X+AR3MA3$ $\varepsilon\sim \mathrm{N}[0;1]$	40.17	116.64	12.78	71.77
5	$Y=\alpha+\beta^{\mathrm{T}}X+\varepsilon$ $\varepsilon\sim \mathrm{U}[0;1]$	27.427	29.945	10.998	45.665

not appropriate, the estimations were modified. The algorithms for minimization and optimization, associated in one single procedure, are described in Section 3. In sums, the learning rule used is the modified delta-rule, optimized by the methods mentioned. A set of 1000 subjects was used for learning. Learning run with an epoch size of 57 input vectors, with a momentum $\mu=0.7$, a learning rate $\eta=0.5$ decreasing by rate $\rho=0.5$ every $\pi=10\,000$ epochs. Every 10 learning iterations, a sample of the test set (1000 subjects) was used for testing. The learning phase stopped as soon as 10 consecutive tests presented a criterion worse than the minimum root mean square error criterion. Otherwise, the learning phase continued until 500 000 epoch iterations had been reached (this high number was chosen to avoid stopping the algorithm before the best estimation was found). After this learning and testing algorithm, the pruning phase found the minimum MLP (Sussman, 1992).

To estimate the linear model, we used the first two sets of data (2000 subjects). The third sample was used to evaluate the prediction capabilities of the MLP and of the linear regression. We computed the criteria of comparison with the value predicted by the MLP or by the regression model for each input vector.

The predictions made by the MLP and the linear regression were first compared. Then, both of them were compared with the desired true values. Those comparisons were achieved graphically. For the MLP, we computed prediction intervals around each predicted value, as described by Hwang and Ding (1997), after assuming normality of distribution of errors.

Those computations were performed with the software Mathematica 2.2 (Wolfram Research Inc. 1993, Champaign IL). The size of the prediction intervals and the regression prediction membership of those intervals were analyzed graphically.

To compare prediction errors, we used a criterion based on relative errors, given by

$$C(RE)=\sum_{i=1}^{N_{\mathrm{p}}}\left[\left(\frac{f_{\hat{W}}(X_{n+i})-f_W(X_{n+i})}{f_W(X_{n+i})}\right)^2\right],$$

where $f_{\hat{W}}(X_{n+i})$ is the predicted value, $f_W(X_{n+i})$ is the value to be predicted and N_{p} is the size of the prediction sample.

We also used the standard goodness-of-fit criteria (Zucchini, 2000), after checking that the distribution of errors was graphically Normal. Those criteria were computed as follows:

- Log likelihood: $L = \ln(\ell)$,
- Akaïke criterion[1]: $AIC = -2L + 2p$,
- Simplified Kullback-Leibler criterion (see footnote 1): $C(KL) = \dfrac{AIC}{2N_p}$,
- Schwarz criterion[2]: $BIC = 2L - p\ln N_p$,

where ℓ is the likelihood, and p the number of coefficients in the linear regression or the number of weights in the MLP.

Least-square error estimators used for linear regression have well described qualities, whereas the quality of learning methods used by connectionist models is not that well known. We studied the quality and the stability of estimations of the MLP. The same models as those specified by the prediction phase were used for this estimation study. Thus, we used the pruned architecture, and the number of learning iterations previously determined. The learning parameters also remained the same. We used Bootstrap method to re-sample the set of 3000 simulated subjects, except for the AR3MA3 design. We obtained 250 samples including 500 subjects each. For the fourth design (AR3MA3), we generated new samples to protect the auto-correlated structure of the data. The estimation process was repeated 250 times for each design, using a different sample each time. To initialize the weights in the MLP, a different generating integer was used for each of the 250 networks per design. Weight estimations are frequently used to interpret the models (Duh et al., 1998). We studied the distribution of weights in MLP to analyze the qualities of those methods of estimation, and the stability of estimations. We built empirical distributions of the weight estimations with the bootstrap method.

To study the prediction stability of the models a subject was randomly chosen among the 3000 simulated subjects per design ($p = 1/3000$). To evaluate the prediction stability, this subject was used in the 250 MLP per design.

Because the linearity was not always respected in our simulations, we evaluated the prediction stability of the linear regression, using the same 250 samples per design for estimation and one subject per LR design for prediction. The prediction stability was evaluated based on the mean, the variance and the error of prediction of each of the 250 models per design, for MLP and LR.

5. Results

With linear regression, whatever the design was, no coefficient associated with the covariate X_4 was significant, and the Wald statistic (statistic of the coefficient test to

[1] The smallest value, the best goodness of fit.

[2] The greatest value, the best goodness of fit.

Table 4
Linear regression: estimation of coefficients and Wald statistics associated (Sample size: 2000 subjects simulated for each design)

$g(X)$	α Estimation (statistic)	β_1 Estimation (statistic)	β_2 Estimation (statistic)	β_3 Estimation (statistic)	β_4 Estimation (statistic)	β_5 Estimation (statistic)
$Y=\alpha+\beta^{\mathrm{T}}X+\varepsilon$ $\varepsilon\sim \mathrm{N}[0;1]$	8.046 (0.724)	−0.671 (−56.97)	3.044 (39.14)	2.146 (159.37)	−0.695 (−0.65)	3.77×10^{-2} (3.99)
$Y=\alpha+\beta^{\mathrm{T}}X+\varepsilon$ $\varepsilon\sim \mathrm{N}[0;v^2\times(f(X))]$	14.32 (0.632)	−0.607 (−25.325)	2.888 (18.287)	2.231 (80.637)	−1.314 (−0.605)	2.97×10^{-2} (1.539)
$Y=\alpha+\beta^{\mathrm{T}}X+\gamma X_3X_5+\varepsilon$ $\varepsilon\sim \mathrm{N}[0;1]$	−29.321 (−1.47)	−0.695 (−32.925)	2.983 (21.377)	3.847 (159.156)	1.839 (0.96)	1.397 (82.365)
$Y=\alpha+\beta^{\mathrm{T}}X+AR3MA3$ $\varepsilon\sim \mathrm{N}[0;1]$	−64.95 (−0.62)	−0.943 (−8.486)	2.81 (3.827)	2.028 (15.951)	8.089 (0.803)	-4.07×10^{-2} (−0.456)
$Y=\alpha+\beta^{\mathrm{T}}X+\varepsilon$ $\varepsilon\sim \mathrm{U}[0;1]$	0.164 (0.256)	−0.66 (−971.89)	2.98 (664.48)	2.14 (2755.66)	7.36×10^{-2} (1.19)	0.118 (216.94)

Table 5
Optimum number of hidden PEs and associated number of iterations

Design	1	2	3	4	5
Number of hidden PEs	3	3	2	2	5
(total number of weights)	(22)	(22)	(15)	(15)	(36)
Learning iterations	19 400	22 700	36 000	7900	51 600

zero) of the coefficient associated with the covariate X_3 was always the greatest. Only the second design presented a non-significant coefficient associated with X_5. Table 4 shows the results of the estimations.

For each design, the optimum architecture and optimum number of learning iterations, shown in Table 5, were obtained by the optimized delta-rule used for modeling the MLP. Thus, the design 5 (where the errors were Uniform) had the greatest number of hidden processing elements, and learning iterations.

Scatter plots of the results for each design are presented in Figs. 3–7. Predictions of the MLP are shown for each predicted value of the linear regression. Prediction intervals around each prediction of the MLP or of the linear regression are represented depending on the simulated predicting values.

Prediction intervals of the MLP seemed similar for each design, and linear regression predictions and predicting values were all within the prediction intervals of the MLP.

For designs 1 and 5, where the underlying assumptions were acceptable, prediction errors of the MLP were small and of the same size order as the prediction errors of the linear regression. In design 2, where simulated errors were heteroscedastics,

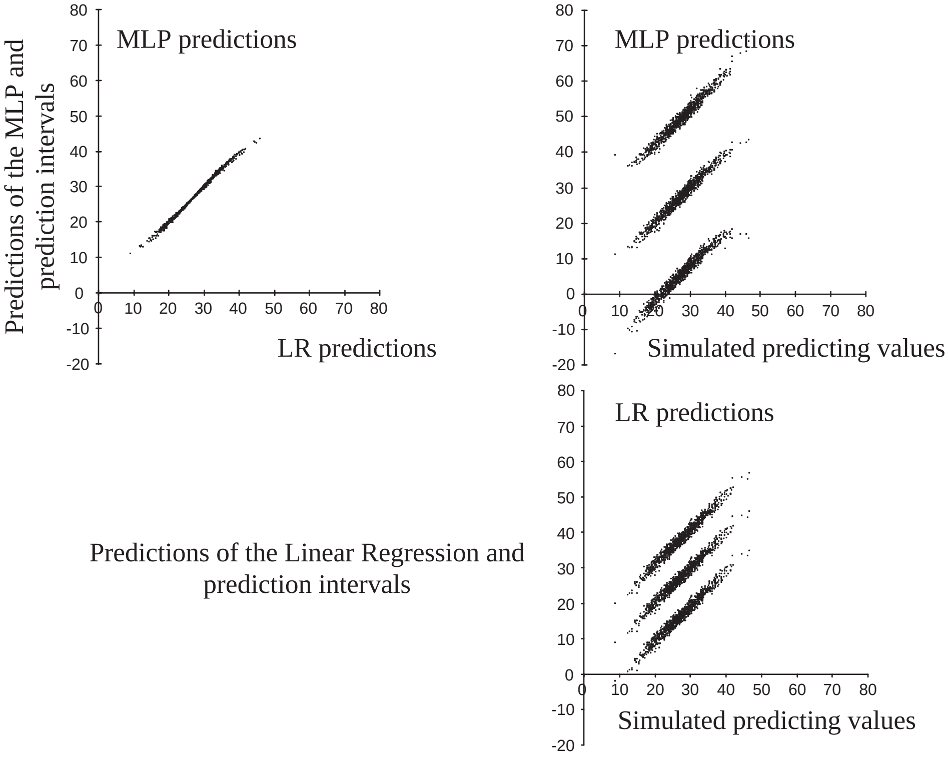

Fig. 3. Design 1, scatter plots of the 1000 predictions and their prediction intervals.

predictions of either the MLP or the linear regression were not so accurate (despite the use of WLSE estimator for the linear regression). Predictions of those two models (in the design 2) were of the same size order. In design 3, in spite of the interaction between the 2 covariates X_3 and X_5, the 2 models (MLP and linear regression models) were correctly and similarly predicting. But, in design 4, which included the AR3MA3 process, the predictions were far from the predicting values, for the MLP, as well as the linear model. Again, those two models (MLP and linear regression models) were similar regarding their predictions, but the prediction intervals of the linear regression seemed to be smaller.

All the criteria used for the comparison of the 2 models were of the same size order. Table 6 shows those criteria. It implied that the goodness-of-fit was similar for those 2 models. Data were better fitted by the linear regression than by the MLP for designs 1, 2, and 5. The MLP was more suitable for design 4. The Akaïke and the Schwarz criteria showed contradictory results for design 3. Only the BIC was in favor of the linear regression. But in fact, the design included a multiplicative interaction, and so, some coefficients were correlated with each other. More than the AIC, the BIC depends on the number of coefficients. In this design, the BIC associated to the MLP was poor probably because of the interaction and the high number of weights. Some of

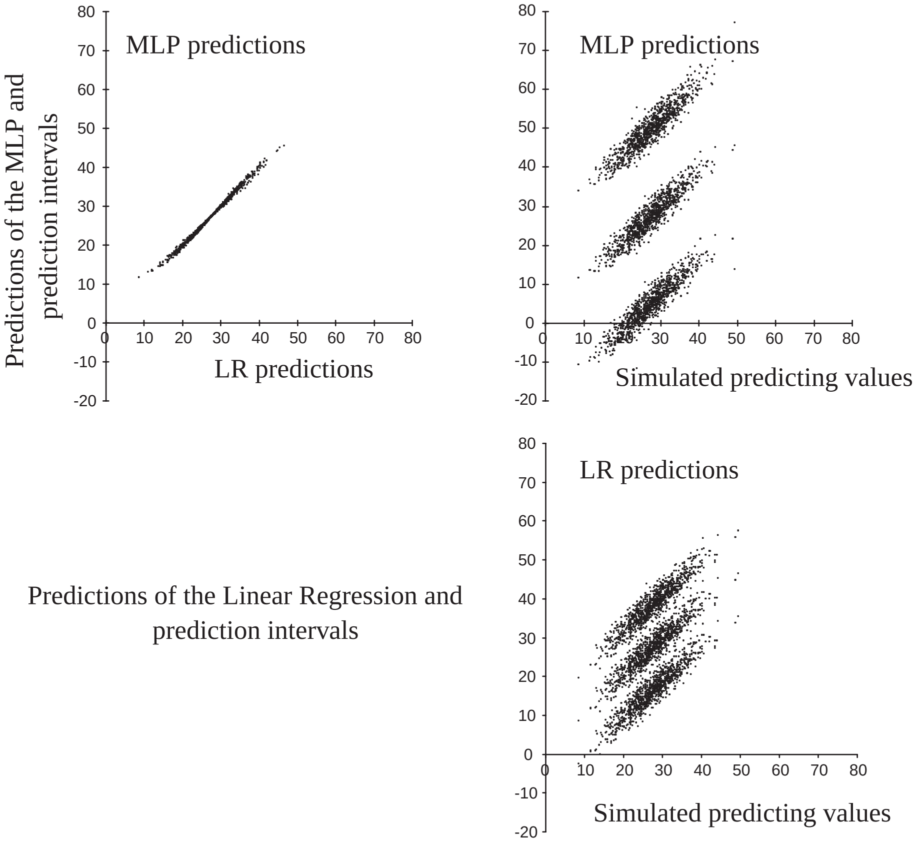

Fig. 4. Design 2, scatter plots of the 1000 predictions and their prediction intervals.

those weights may have been generated only because of the presence of an interaction factor and would have been unnecessary otherwise. The relative error criterion did not present the same problem. For this design, it constituted the best criterion to analyze the relative performances of each model. Thus, according to this criterion, design 3 was best fitted by MLP.

Weight estimations of the MLP were studied by a re-sampling method. Box-plots of their empirical distributions are shown in Figs. 8–12.

The connections are identified by the names of the two PEs link. b represents the bias PE, X_i an input PE, C_i a hidden PE and y the output PE.

The empirical distributions of the weights were more or less distant from a Normal distribution, depending on the connections weighted. In fact, no a priori theoretical distribution could be assumed. But, there were some similarities between designs 1, 2 and 5. Those 3 simulation designs were very close, being linear models with either a Normal or an asymptotically Normal distribution of the errors. This could explain why the weights between the same processing elements were similar in the 3 designs, with

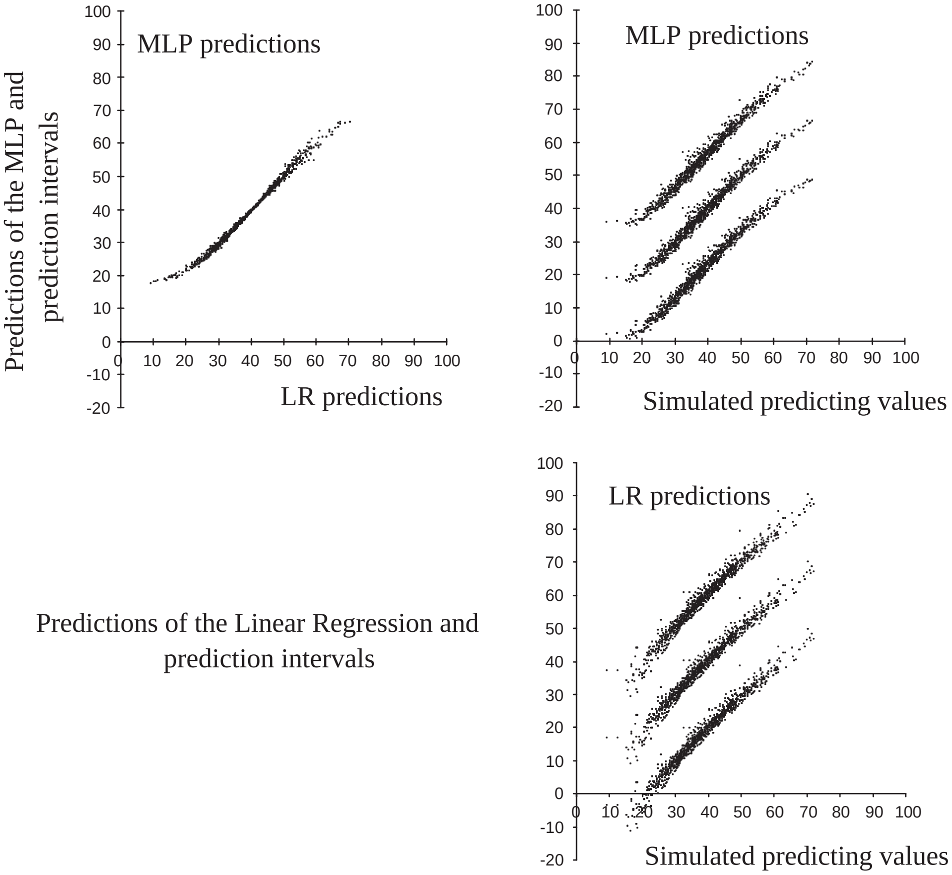

Fig. 5. Design 3, scatter plots of the 1000 predictions and their prediction intervals.

the exception of the weights attached to the fourth and the fifth hidden PEs, which were only presented in the fifth MLP.

The value of a weight gives an indication of the importance of the connections between processing elements. These values are used in that way to analyze the connectionist model (Duh et al., 1998), and especially to interpret the importance of a covariate. One could then expect that the empirical means of the weights would reflect this statement. But in our experience, after re-sampling, whatever the design was, those means were poorly informative. For example, the means of the weights associated to the covariate X_3 could be positive or negative, in spite of a large positive simulated coefficient. Even though the covariate X_4 was not an explicative variable ($\beta_4 = 0$), some empirical means of the weights attached to this covariate were not equal to 0. Thus, those empirical means were not very contributive to the analysis of our MLP.

The empirical variances seemed to be more informative. Indeed, whatever the design was, the weights attached to the covariate X_3 had the largest variances. These were

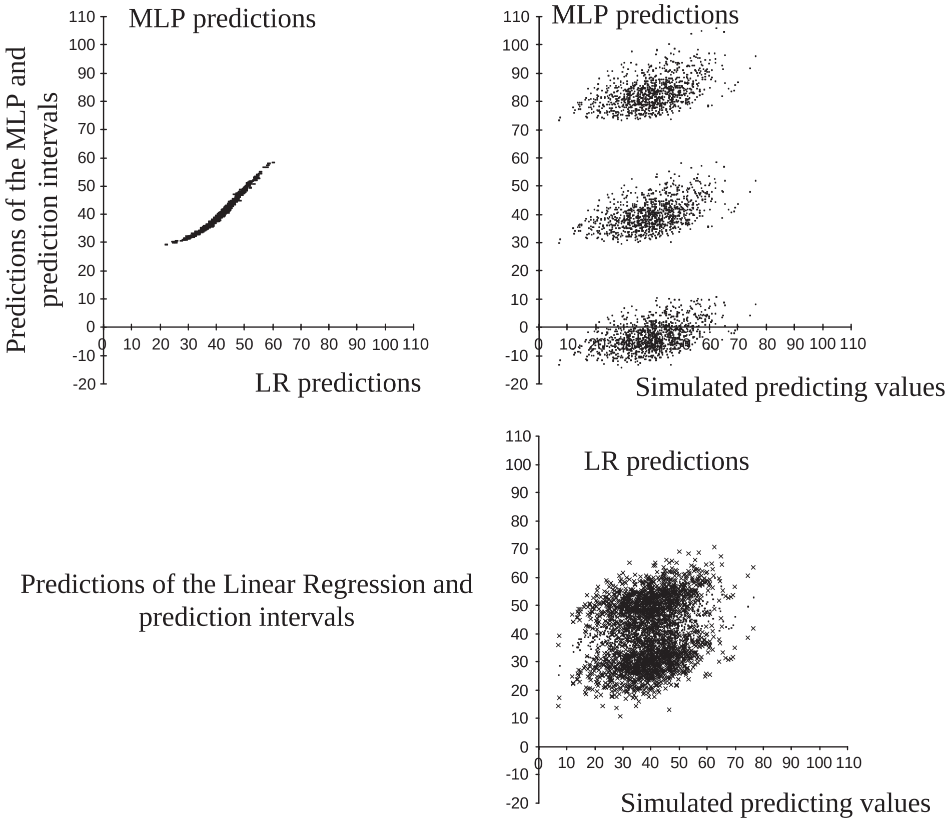

Fig. 6. Design 4, scatter plots of the 1000 predictions and their prediction intervals.

the only variances which were greater than 1, except for design 3, where the empirical variances attached to the covariate X_5 were also greater than 1. This could be explained by the interaction term X_3X_5 included in this design. Furthermore, the linear regression showed that Wald statistics of the estimated coefficients attached to the covariate X_3 in designs 1 to 5, and attached to the covariates X_3 and X_5 in design 3, were the largest. Thus, we consider the variances of the weights to be strongly contributive for the interpretation of the MLP results for epidemiological data, and this analysis will be discussed further in Section 6.

The stability of the models was studied by the empirical distributions of the predictions and of the prediction errors. The results showed the reproducibility of the estimation algorithms for MLP, and for LR the robustness to deviations from underlying assumptions. Characteristics of those distributions are shown in Tables 7 and 8.

The predictions of the MLP were stable, whatever the design was, since their variances were null (or small for the fourth design), and since their ranges were small (the largest range (1.32) was presented by the fourth design). In addition, prediction

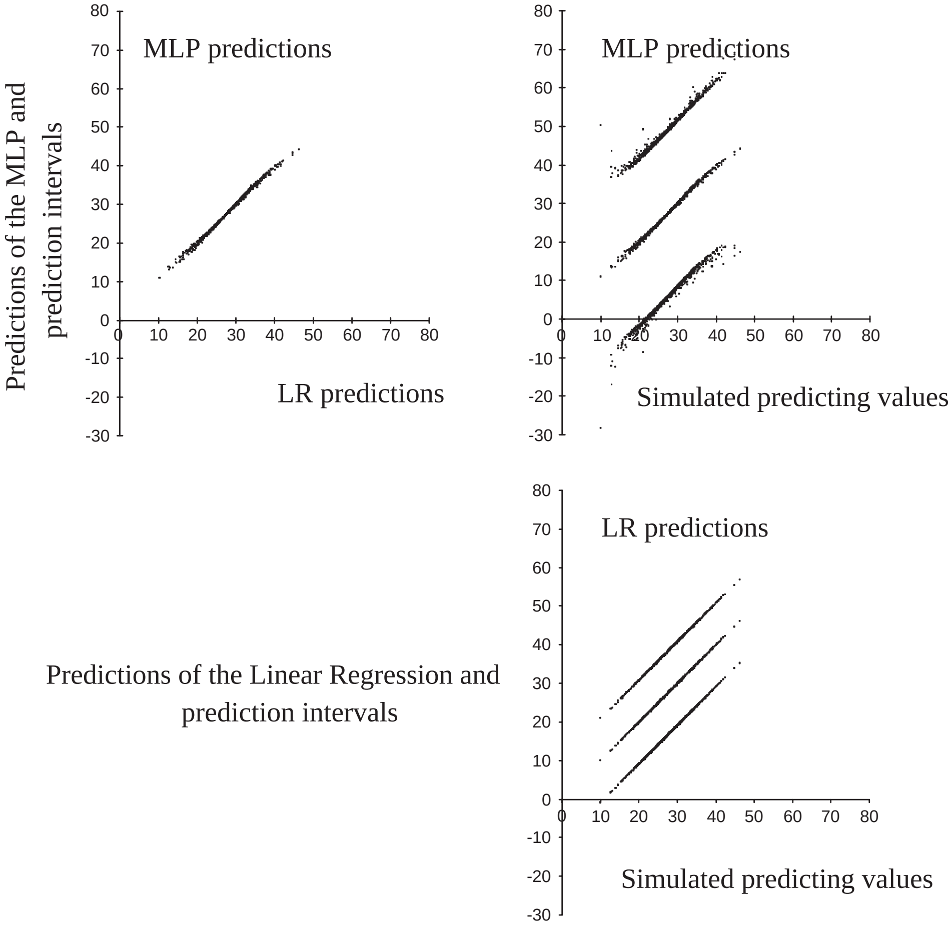

Fig. 7. Design 5, scatter plots of the 1000 predictions and their prediction intervals.

errors were on average very small and their variances were null or approaching 0. Graphically, predictions and prediction errors were empirically normally distributed.

Except for design 4, the predictions of linear regression were stable, and the errors were small. For the ARMA design (design 4), linear regression was not an adequate model. The variance of the prediction was high (37.34), even if the mean of the prediction errors is small (2.72).

6. Discussion

Our study shows that the use of multilayer perceptron is comparable, for epidemiological data, to linear regression regarding the predictions, the goodness-of-fit and the

Table 6
Goodness-of-fit criteria and relative errors for the 5 designs (optimum results are darkened)

Design	$g(X)$	Model(p)	L	AIC	c(kl)	BIC	c(re)
1	$Y = \alpha + \beta^{\mathrm{T}}X + \varepsilon$	MLP(22)	**−3129.49**	6302.99	3.15	−6410.96	1.76
	$\varepsilon \sim \mathrm{N}[0;1]$	LR(6)	−3133.79	**6279.57**	**3.14**	**−6309.02**	**1.63**
2	$Y = \alpha + \beta^{\mathrm{T}}X + \varepsilon$	MLP(22)	−3176.65	6397.30	3.19	−6505.27	7.24
	$\varepsilon \sim \mathrm{N}[0;v^2 \times (f(X))]$	LR(6)	**−3176.53**	**6365.07**	**3.18**	**−6394.52**	**6.94**
3	$Y = \alpha + \beta^{\mathrm{T}}X + \gamma X_3 X_5 + \varepsilon$	MLP(15)	**−3739.65**	**7509.3**	**3.76**	−7582.92	**3.53**
	$\varepsilon \sim \mathrm{N}[0;1]$	LR(6)	−3760.31	7532.61	3.77	**−7562.06**	4.45
4	$Y = \alpha + \beta^{\mathrm{T}}X + AR3MA3$	MLP(15)	**−4478.67**	**8987.34**	**4.49**	**−9060.96**	**136.78**
	$\varepsilon \sim \mathrm{N}[0;1]$	LR(6)	−4634.08	9280.16	4.64	−9309.61	139.73
5	$Y = \alpha + \beta^{\mathrm{T}}X + \varepsilon$	MLP(36)	**−3129.76**	6331.52	3.17	−6508.20	0.2
	$\varepsilon \sim \mathrm{U}[0;1]$	LR(6)	−3130.92	**6273.83**	**3.14**	**−6303.28**	**0.01**

L: log-likelihood; AIC: Akaïke information criterion; BIC: Bayesian information criterion; c(kl): Simplified Kullback-Leibler criterion; c(re): relative errors criterion; MLP: multilayer perceptron; LR: linear regression; p: number of coefficients of the model.

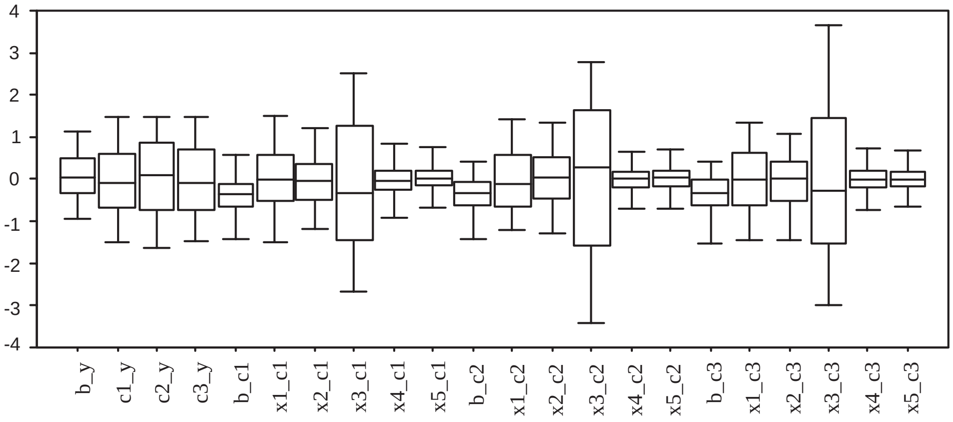

Fig. 8. Design 1: empirical distributions of the weight estimations ($n = 250$) for each connection.

effect of deviations from underlying assumptions of normality, homoscedasticity and independence of the errors. The five designs show that the predictions of those two models are very close. They are also close to the predicting values, except for the fourth design, where the auto-correlation term alters both models. In addition, similar qualities for both models were revealed by analogous goodness-of-fit and relative error criteria.

In the analysis of epidemiological data, one must often make many assumptions about the data, and must sometimes limit the analysis. By contrast from a practical point of

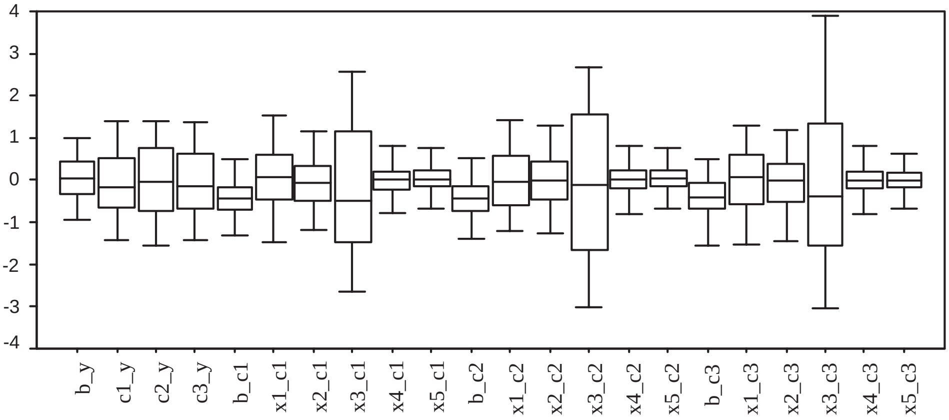

Fig. 9. Design 2: empirical distributions of the weight estimations ($n = 250$) for each connection.

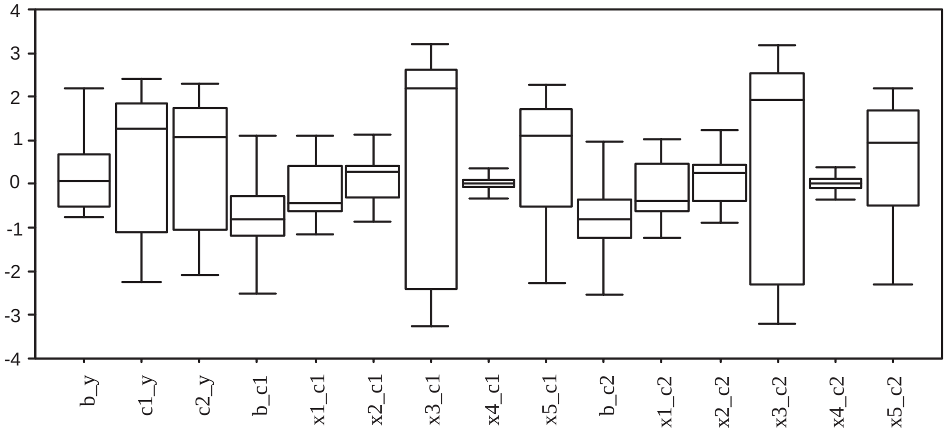

Fig. 10. Design 3: empirical distributions of the weight estimations ($n = 250$).

view, neural networks are basically non-parametric, although their learning parameters and their weights parameterize them. Whereas the underlying assumptions of normality, homoscedasticity and independence of the errors are required to use linear regression model, the connectionist model does not depend on these assumptions. This model uses numerical and iterative estimation methods to control the convergence and the generalization. Thus, the neural network is considered as a semi-parametric model (Capobianco, 2000). In addition, the unspecified interactions have an effect on the weight estimations. Ignoring colinearity between covariates does not affect the goodness-of-fit or the prediction capabilities of the MLP. The linear regression does not have this capability and interaction terms have to be specified before modeling.

Even though neural networks are flexible tools, the user has to specify methods and basic parameters, guided by his experience. But there is a range of possibilities

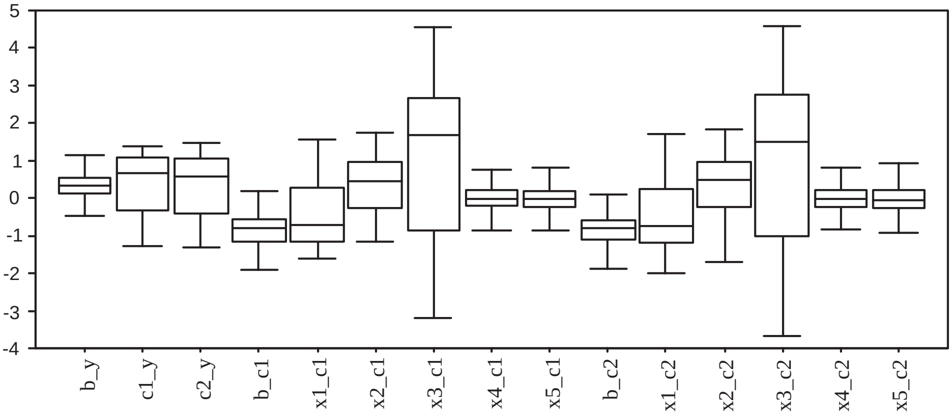

Fig. 11. Design 4: empirical distributions of the weight estimations ($n = 250$).

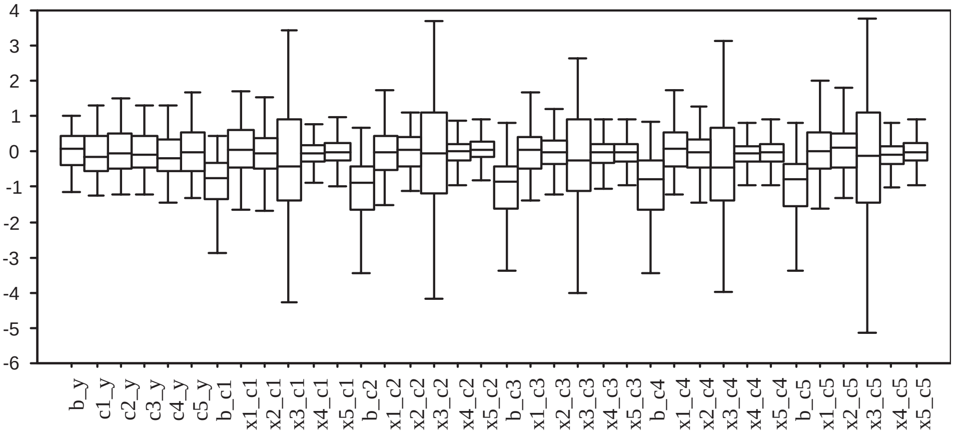

Fig. 12. Design 5: empirical distributions of the weight estimations ($n = 250$).

to construct such networks. We chose simple processing elements, with demonstrated properties (Hornik et al., 1989). Similar results could be obtained with other transfer functions (instead of the sigmoid functions) from the class of squashing functions (non-decreasing and bounded functions), which has well known properties.

The complexity of the model is based on the architecture, which is determined here by the number of hidden PEs. This architecture defines the class of computable functions: it is the first parameter used to control the network. To preserve convergence properties, some restrictions have to be respected, particularly concerning the Vapnik-Chervonenkis dimension (VC-dimension) (see Gascuel, 1997; Bottou, 1997; Vapnik and Bottou, 1993). Karpinski and Mac Intyre (1995) showed that for a MLP with sigmoid transfer functions, the consistency is valid if the number of hidden PEs, C, is such that $C < \mathrm{O}(\sqrt[4]{n})$, where n is the number of inputs, and O() the Landau

Table 7
MLP: characteristics of the empirical distributions of the prediction and of the prediction error (250 predictions of the same value using the 250 estimated MLP for each design)

Design		Mean	Variance	Minimum	Median	Maximum
1	Prediction	0.23	0.00	0.20	0.23	0.26
	Error	0.07	0.00	0.05	0.07	0.10
2	Prediction	0.12	0.00	0.06	0.12	0.16
	Error	0.03	0.00	0.00	0.03	0.07
3	Prediction	0.73	0.00	0.70	0.73	0.77
	Error	0.04	0.00	0.01	0.04	0.08
4	Prediction	0.79	0.11	0.04	0.84	1.36
	Error	0.31	0.03	0.00	0.34	0.73
5	Prediction	0.14	0.00	0.11	0.14	0.18
	Error	0.01	0.00	0.00	0.01	0.05

Table 8
Linear regressions: characteristics of the empirical distributions of the prediction and of the prediction error (250 predictions of the same value using the 250 estimated LR for each design)

Design		Mean	Variance	Minimum	Median	Maximum
1	Prediction	29.67	0.01	29.45	29.68	29.89
	Error	1.01	0.01	0.79	1.02	1.23
2	Prediction	29.74	0.02	29.35	29.73	30.07
	Error	0.35	0.02	−0.05	0.35	0.69
3	Prediction	42.15	0.02	41.78	42.14	42.45
	Error	1.10	0.02	0.73	1.09	1.40
4	Prediction	42.58	37.31	35.40	46.31	49.08
	Error	2.68	37.31	−4.50	6.41	9.18
5	Prediction	30.18	0.00	30.07	30.18	30.30
	Error	0.03	0.00	−0.08	0.03	0.15

notation. This condition binds the number of hidden PEs to the size of the learning set. In addition redundant hidden PEs imply that the Fisher information matrix of the MLP will be singular (Fukumizu, 1996). Those conditions have to be respected, to preserve the convergence of the model and the use of statistical properties.

Back-propagation of the gradient (the delta rule) as a learning method is often effective, and is supported by stochastic as well as deterministic convergence analyses (Bertsekas, 1996). However, this method typically has a slow convergence rate. In addition, because the algorithm chooses iteratively the function that best fits the training data, the flexibility of the model could create a problem by fitting the noise of the training data. This problem called over-learning leads to prediction defects. The presence of local minima on the error function is another problem that limits the convergence and thus, the prediction capabilities. With the modifications introduced in the learning algorithm, we assume that the convergence occurred, and that the generalization capabilities were sufficient to predict. Some other methods exist, they are all numerical and iterative heuristics (Gallinari, 1997). However they have an influence on the quality of estimations and predictions, and on the interpretation possibilities of the model. Despite this, those heuristics have to be systematically used to control the complexity and the nature of the estimated solutions. Furthermore, the weakness of the delta-rule has to be compared to that of other methods such as Levenberg-Marquardt algorithm or the extended Kalman filtering (Bertsekas, 1996).

The non-reproducibility of the results and the multiplicity of models obtained by heuristics are two frequently pointed out problems. But, our results show that the MLP, built as described, gives reproducible predictions. The variances and the ranges of the predictions are very small, in spite of the differences between the learning sets. Thus, the prediction capabilities of this neural network are stable, whatever the deviation from the linear model. In the same way, the predictions made during the estimation analysis respect the Normality assumption of distributions. The work is based on a single realization of the simulation. Although we used a re-sampling method to estimate the stability of the MLP, replications of the simulations will need to be carried out to analyze the stability.

Weight estimations are classically used to interpret the MLP (Duh et al., 1998). But our results show that their empirical means are poorly informative. The interpretation of a covariate cannot be grounded on the weight estimation of its connection to a hidden PE, whereas coefficients of linear regression are directly used to interpret the importance of a covariate. In fact, connectionist model and linear regression should not be interpreted in similar ways. A particularity of the neural networks is that, to link input vectors to the output vectors, the information is distributed among the processing elements (Hinton et al., 1986). This capacity of parallel distributed processing can be compared to data analysis methods (Nicole, 2000). This distributed representation of the information varies according to the learning data, and thus weight estimations vary as well. The importance of empirical variances of weight estimations in our work is the result of the variation of this distributed representation of the information, from one learning set to another. Indeed, if, according to the learning set, the hidden PEs are not used in the same way to process the learning data, then the information of a covariate has an effect on different hidden PE. The weight estimations depend on the hidden PE used to process the greatest part of the information. A connection between an input PE and a hidden PE, considered as a strong connection after one learning phase, might be considered as a weak one after another learning phase with a different learning set. Fig. 13 illustrates the variations of weight estimations.

This variation of the strength of a connection, or “rocking motion”, leads us to interpret the variances of the weight estimations as indicators of the covariate importance in the neural network. Moreover, the Wald statistics attached to the coefficients of the linear regression corroborate our interpretation. Even though the principle of the distributed representation of the information in neural networks has been well studied (Hinton et al., 1986), as far as we know, no other study has so far pointed out the major role of variances of weight estimations in the interpretation of a neural network. Further studies are required to confirm our findings.

Our study, using simulated data, provides a basis for considering neural networks as a particular class of statistical models for the analysis of epidemiological data. Moreover, they cannot be used as “black boxes” to obtain appropriate models

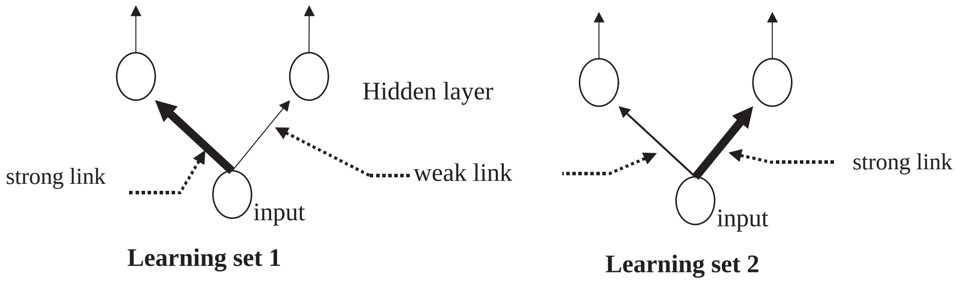

Fig. 13. Representation of the variations of weight estimations between an input PE and a hidden PE, for 2 different learning sets.

(Schwartz et al., 2000). The results show that MLP cannot replace linear regression models. Nevertheless, if MLP are not the most appropriate model for the analysis of epidemiological data, they are useful to take unspecified colinearity or non-linear functions into account. Due to the initialization of weights with small values, the MLP have an initially linear behavior by the linearization of transfer functions. Step by step, during the learning procedure, weights are modified, leading to non-linear solutions. This is how the MLP can model non-linear regression functions. They are sometimes seen as simple representations of non-linear regression models, using different algorithms for numerical solutions (back-propagation of the gradient, Newton-Raphson, Gauss-Newton or Extended Kalman Filter methods).

In fact, none of the neural networks outperformed linear regression when linear regression is used optimally. MLP are useful when the real underlying regression function cannot be approached by classical methods. But the existence of such functions in epidemiological applications is doubtful, and so is the practical interpretation of their covariates. Whereas, neural networks are very useful for pattern recognition (Vach et al., 1996).

The distributed representation pattern of the information is an interesting contribution of neural networks to classical statistics. This representation is defined by the neural network and not by the user a priori. A parallel could be made between this distribution pattern of the information and the principal components extraction, where the hidden PEs are compared to the principal components. So, using MLP seems to be comparable to regression on principal components, linear or non-linear regression depending on the transfer function.

The presented MLP is a robust model, but it is one of the numerous different connectionist models. The exploration and the systematic comparison of those models to classical statistical models have to be continued, to improve their use as classifying, predicting, or estimating models (Fogelman Soulié, 1997).

Acknowledgements

We thank Dr Bernard Fichet for many valuable discussions, Co-Editor and the anonymous reviewers for their helpful comments.

References

Baxt, W.G., 1995. Application of artificial neural networks to clinical medicine. Lancet 346, 1135–1138.

Bertsekas, D.P., 1996. Incremental least squares methods and the extended Kalman filter. SIAM J. Optimiz. 6, 807–822.

Bottaci, L.., et al., 1997. Artificial neural networks applied to outcome prediction for colorectal cancer patients in separate institutions. Lancet 350, 469–472.

Bottou, L., 1997. La mise en oeuvre des idées de VN Vapnik. In: Thiria, S., Lechevallier, Y., Gascuel, O., Canu, S. (Eds.), Statistiques et méthodes neuronales. Dunod, Paris, pp. 262–274.

Capobianco, E., 2000. Neural networks and statistical inference: seeking robust and efficient learning. Comput. Statist. Data Anal. 21, 443–454.

Cheng, B., Titterington, D.M., 1994. Neural networks: a review from a statistical perspective. Statist. Sci. 9, 2–54.

Cross, S., Harrisson, R.F., Kennedy, R.L., 1995. Introduction to neural networks. Lancet 346, 1075–1079.

Duh, M.S., Walker, A.M., Ayanian, J.Z., 1998. Epidemiologic interpretation of artificial neural networks. Amer. J. Epidemiol. 147, 1112–1122.

Flexer, A., 1996. Connectionist and statistician, friends or foes? Proceedings of the 13th European Meeting on Cybernetics and Systems Research, Vol. 6, pp. 995–1004.

Fogel, D.B., Wasson, E.C., Boughton, E.M., Porto, V.W., Angeline, P.J., 1998. Linear and neural models for classifying breast masses. IEEE Trans. Med. Imag. 17, 485–488.

Fogelman Soulié, F., 1997. Réseaux de neurones et statistiques. In: Thiria, S., Lechevallier, Y., Gascuel, O., Canu, S. (Eds.), Statistiques et méthodes neuronales. Dunod, Paris, pp. 1–19.

Fukumizu, K., 1996. A regularity condition of the information matrix of a multilayer perceptron network. Neural Networks 9, 871–879.

Gallinari, P., 1997. Heuristique pour la généralisation. In: Thiria, S., Lechevallier, Y., Gascuel, O., Canu, S. (Eds.), Statistiques et méthodes neuronales. Dunod, Paris, pp. 230–243.

Gascuel, O., 1997. La dimension de Vapnik-Chervonenkis. In: Thiria, S., Lechevallier, Y., Gascuel, O., Canu, S. (Eds.), Statistiques et méthodes neuronales. Dunod, Paris, pp. 244–261.

Guh, J.Y., Yang, C.Y., Yang, J.M., Chen, L.M., Lai, Y.H., 1998. Prediction of equilibrated postdialysis BUN by an artificial neural network in high-efficiency hemodialysis. Amer. J. Kidney Dis. 31, 638–646.

Hinton, G.E., Rumelhart, D.E., McClelland, J.L., 1986. Distributed representations. In: Rumelhart, D.E., McClelland, J.L. (Eds.), PDP research group, Parallel Distributed Processing. MIT Press, Cambridge, MA, pp. 77–109.

Hornik, K., Stinchcombe, M., White, H., 1989. Multilayer feedforward networks are universal approximators. Neural Networks 2, 359–366.

Hwang, J.T.G., Ding, A.A., 1997. Prediction intervals for artificial neural networks. J. Amer. Statist. Assoc. 92, 748–757.

Intrator, N., 1999. Robust prediction in many-parameter models. In: Kay, J.M., Titterington, D.M. (Eds.), Statistics and Neural Networks: Advances at the Interface. Oxford University Press, Oxford, pp. 97–128.

Karpinski, M., Mac Intyre, A., 1995. Bounding VC-dimensional of neural networks: progress and prospect. In: Vitanyi, P. (Ed.), Proc. of the 2nd European Conference on Computational Learning Theory, Barcelona, Spain, Lecture Notes in Artificial Intelligence, number 904, Springer, Berlin, pp. 337–341.

Lapeer, R.J.A., Dalton, K.J., Prager, R.W., Forsström, J.J., Selbmann, H.K., Derom, R., 1995. Application of neural networks to the ranking of perinatal variables influencing birth weight. Scand. J. Clin. Lab. Invest. 55, 83–93.

Lisi, F., Schiavo, R.A., 1999. A comparison between neural networks and chaotic models for exchange rate prediction. Comput. Statist. Data Anal. 30, 87–102.

Mariani, L.., et al., 1997. Prognostic factors for metachronous contralateral breast cancer. Breast Cancer Res. Treat. 44, 167–178.

Medler, D.A., 1998. A brief history of connectionism. Neural Comput. Survey 1, 61–101.

Nicole, S., 2000. Feedforward neural networks for principal components extraction. Comput. Statist. Data Anal. 33, 425–437.

Ottenbacher, K.J., Smith, P.M., Illig, S.B., Linn, R.T., Fiedler, R.C., Granger, C.V., 2001. Comparison of logistic regression and neural networks to predict rehospitalization in patients with stroke. J. Clin. Epidemiol. 54, 1159–1165.

Reggia, J.A., 1993. Neural computation in medicine. Artificial Intelligence Med. 5, 143–157.

Saporta, G., 1990. La régression multiple et le modèle linéaire général. In: Saporta, G. (Ed.), Probabilités, analyse des données et statistique. Technip, Paris, pp. 375–402.

Schumacher, M., Roβner, R., Vach, W., 1996. Neural networks and logistic regression: part 1. Comput. Statist. Data Anal. 21, 661–682.

Schwarzer, G., Vach, W., Schumacher, M., 2000. On the misuses of artificial neural networks for prognostic and diagnostic classification in oncology. Statist. Med. 19, 541–561.

Sonke, G.S., Heskes, T., Verbeek, A.L., De la Rosette, J.J., Kiemeney, L.A., 2000. Prediction of bladder outlet obstruction in men with lower urinary tract symptoms using artificial neural networks. J. Urol. 163, 300–305.

Sussman, H.J., 1992. Uniqueness of the weights for minimal feed-forward nets with a given input-output map. Neural Networks 5, 589–593.

Tian, J., Juhola, M., Gröfors, T., 1997. AR parameter estimation by a feedback neural network. Comput. Statist. Data Anal. 25, 17–24.

Vach, W., Roβner, R., Schumacher, M., 1996. Neural networks and logistic regression: part 2. Comput. Statist. Data Anal. 21, 683–701.

Vapnik, V., Bottou, L., 1993. Local algorithm for pattern recognition and dependencies estimation. Neural Comput. 5, 893–909.

Xiang, A., Lapuerta, P., Ryutov, A., Buckley, J., Azen, S., 2000. Comparison of the performance of neural network methods and Cox regression for censored survival data. Comput. Statist. Data Anal. 34, 243–257.

Zucchini, W., 2000. An introduction to model selection. J. Math. Psy. 44, 41–61.