PERGAMON

Neural Networks 15 (2002) 237–246

Neural Networks

www.elsevier.com/locate/neunet

Contributed article

A methodology to explain neural network classification

Raphael Féraud, Fabrice Clérot*

France Télécom R&D, 2, avenue Pierre Marzin, 22300 Lannion, France

Received 3 October 2000; accepted 3 August 2001

Abstract

Neural networks are still frustrating tools in the data mining arsenal. They exhibit excellent modelling performance, but do not give a clue about the structure of their models. We propose a methodology to explain the classification obtained by a multilayer perceptron. We introduce the concept of 'causal importance' and define a saliency measurement allowing the selection of relevant variables. Once the model is trained with the relevant variables only, we define a clustering of the data built from the hidden layer representation. Combining the saliency and the causal importance on a cluster by cluster basis allows an interpretation of the neural network classifier to be built. We illustrate the performances of this methodology on three benchmark datasets. © 2002 Published by Elsevier Science Ltd.

Keywords: Classification; Clustering; Knowledge extraction; Neural network; Saliency

1. Introduction

Neural networks techniques are now widely recognised as powerful modelling tools and most major vendors have included them in their data mining software. Modelling, however, is only a part of the data mining process and practitioners are still often reluctant towards neural network techniques since they do not yet allow a simple interpretation of the models they build.

Extracting knowledge from the data does not reduce to the construction of a good classifier, even if the ultimate task is a classification task; we also would like to know why a given individual is in a given class, that is analyse the influence[1] of the input variables on the classification. We would also like to know how input variables can be modified so as to bring an individual from one class to another. We shall introduce the concept of '*causal importance*' which extends the 'what if?' analysis to the level of a set of individuals and explains how this extension allows the classification to be interpreted.

For variable elimination, we must define an indicator allowing the variables to be sorted according to their influence on the neural network performance in terms of generalisation error. Such an indicator is usually called the *predictive importance* or *saliency*. We shall describe how this indicator can be built from the neural network trained with all the inputs.

Given the classifier, the interpretation can be done at various levels; it can be done globally by analysing the classification behaviour as a function of the inputs, but this does not give us any knowledge on the behaviour of individuals. It can also be done at the individual level by the so-called 'what if?' analysis, but such an analysis is not scalable. Indeed, the extracted knowledge will be much more valuable if we can first attach the individuals to clusters and then analyse the influence of the input variables on these clusters. Of course, this only makes sense if the clustering is built from the classifier and does not degrade the classification performance (i.e. clusters should be as homogeneous as possible in terms of class labels). In a feedforward perceptron with one hidden layer used as a classifier, the second layer of weights only performs a linear separation in the representation space built by the outputs of the hidden units. The relationship between feedforward perceptrons with one hidden layer and discriminant analysis is well established (Gallinari, Thiria, Badran & Fogelman-Soulié, 1991) with a formal equivalence in the case of linear networks. By analogy to the linear case, we shall, therefore, refer to the hidden units as 'factors'. The clustering is performed in factor space since all the

* Corresponding author. Tel.: +33-2-96-05-21-20; fax: +33-2-96-05-23-58.

E-mail address: fabrice.clerot@francetelecom.com (F. Clérot).

[1] The terminology in the area of neural network interpretation is heavily overloaded; in this article, we chose to use the term 'influence' in a very loose way. The term 'saliency' refers to the measure of the importance of a variable for the accuracy of the model. We introduce the term 'causal importance' for the effect of a variable on the output of the model (i.e. how changing a variable changes the classification behaviour (not the classification performance), given the model).

0893-6080/02/$ - see front matter © 2002 Published by Elsevier Science Ltd.
PII: S0893-6080(01)00127-7

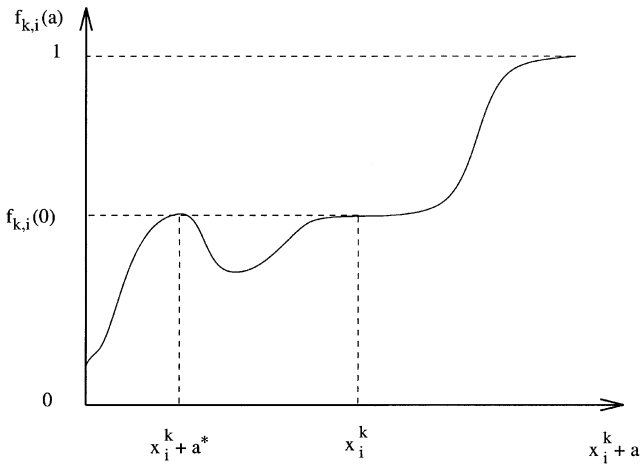

Fig. 1. The output of the neural network versus a. For a large value of a, the example is classified as an example of class 1, and for a small value of a as an example of class 0: the variable x_i is important for the classification of this example.

necessary information for the classification lies there. The number of clusters is adjusted so as to recover a good homogeneity of the clusters. We shall define the concept of causal importance and saliency to the cluster level. We shall use these measurements to interpret each important variable of each cluster. Since individuals are attached to clusters, conclusions drawn on the clusters also apply to them.

Our general framework to tailor feedforward neural networks with one hidden layer to a practical data mining process is, therefore, the following.

1. Train a neural network with all the input variables.
2. Modelling phase:
   2.1. variable selection to eliminate irrelevant inputs; and
   2.2. train a neural network with only the relevant inputs.
3. Interpretation phase:
   3.1. automatic partition of the data in the representation space defined by the hidden layer;
   3.2. interpretation of each obtained cluster; and
   3.3. analysis of the influence of the inputs on the clusters.

The next section starts with a short presentation of the 'what if?' analysis and a review of previous work in this area and presents our analysis of the input variable influence. Section 3 presents our framework for the interpretation of neural network classification. Section 4 shows the results obtained by applying these techniques to three benchmark datasets.

2. Analysis of an input variable influence

2.1. Motivation and previous works

When a classifier has been built, a question often raised in practice is '*What* would happen to this individual *if* this variable was set to a different value?'. A simple way to answer this question is to plot the variation of the output of the classifier for this individual versus the variation of the variable (Fig. 1).

Let f be the multilayer perceptron input–output mapping, we define:

$$f_{k,i}(a) = f\left(x_0^k, \ldots, x_i^k + a, \ldots, x_n^k\right)$$

where x_i^k is the value of the input variable i for the example k, and $a \in \mathbb{R}$.

Since multilayer perceptrons are a non-linear model, the function $f_{k,i}(a)$ can be non-monotonous. Hence, the influence of an input variable cannot be evaluated by a local measurement: in the illustrative example of Fig. 1, the gradient of the output with respect to the input x_i is null for the individual k even though the variable x_i is important for its classification.

The measurement of the difference of the output of a neural network with respect to the variation of an input variable by a fixed value h provides a more global information and can be applied to discrete variables. However, the choice of h depends on each input variable and on the function itself: a small value has the same drawback as the partial derivatives (local information and not well suited for discrete variables), a large value can be misleading if the function with respect to an input i is non-monotonous, or periodic (in the example of Fig. 1, the choice of $h = a^*$ would be misleading). The same criticism applies when these measurements (see Moody, 1994; Réfénes, Zapranis & Utans, 1994 for partial derivatives and Baxt & White, 1995, for differences) are averaged on the whole training set.

The weights provide non-local information on the influence of an input variable. The input importance can be obtained by analysing the size of the weights (Garson, 1991). A regularisation technique has to be applied during the training process to ensure that only relevant features will, indeed, survive after convergence (Burkitt, 1992; Rumelhart, Hinton & Williams, 1986). The main criticism regarding this interpretation is that the non-linearities of the hidden units can make the size of the weights misleading in some situations (Sarle, 1998), even if a penalisation term can be added to the contribution of weights for saturated hidden neurons so as to compensate for these non-linearities (Mak & Blanning, 1998).

It is also possible to rely on more sophisticated heuristics to determine the important features, from the inverse Hessian matrix of the error for instance (Hassibi & Stork, 1993; Le Cun, Denker & Solla, 1990). The functional Taylor series of the error E with respect to the vector of weights W:

$$\partial E = \left(\frac{\partial E}{\partial W}\right)^T \cdot \partial W + \frac{1}{2}\partial W^T \cdot \frac{\partial^2 E}{\partial W^2} \cdot \partial W + O(\|\partial W\|^3)$$

Considering that a local minimum is reached after the training, the first order term of this equation vanishes and the third order term is insignificant. However, a zero first

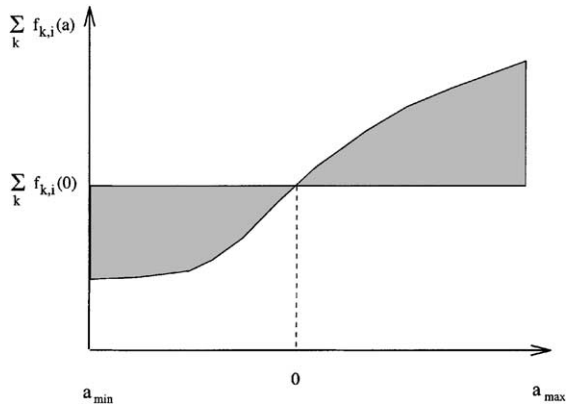

Fig. 2. The saliency of the input variable i is the area between the curve of causal importance and the reference curve (the curve obtained when the influence of the input variable is null) weighted by the prior of a.

order term implies that a local minimum of the error function is reached. Using structural risk minimisation framework (Vapnik, 1995), this assumption can be wrong since to avoid over-training, the optimisation of the training error is stopped when the error increases on a validation set. Then, the first order term can be higher than the second order term and this saliency measurement can be misleading.

A characterisation of the 'what if?' simulation is that it relies on the generalisation capabilities of the model since the output of the model is calculated with values of the variables which can be away from the training set; for instance, a discrete variable can be treated as a continuous one. We shall extend the 'what if?' simulation to define the causal importance and the saliency measurement.

2.2. Causal importance

We define the causal importance of the input i of the model f as:

$$CI(a|x_i,f) = \int_{x_i} P(x_i) f_i(x_i + a) \mathrm{d}x_i$$

where $f_i(x)$ is the function defined by the neural network when all the inputs $j \neq i$ are fixed and $P(x_i)$ the probability to observe the value x_i.

Approximating this probability by the empirical distribution on the training set, we obtain:

$$CI(a|x_i,f) = \sum_{k=0}^{N} f_{k,i}(a)$$

where N is the number of examples, which allows a simple interpretation of our definition of the causal importance as an extension of the *what if* simulation to a set of examples.

This measurement allows the effect of the variation of an input when all the others are fixed to be explored. In the case of a linear model, $CI(a|x_i,f)$ is a linear function with slope Nw_i

$$CI(a|x_i,f) = N \sum_{k=0}^{m} w_j \bar{x}_j + Nw_i \times a = K + Nw_i \times a$$

where $\bar{x}_j$ is the mean value of the variable x_j, m the number of input variables, and K a constant ($K = w_0$ for centred variables). We recover the fact that the weights are sufficient to analyse the causal importance of an input variable i for a linear model. In the general case, only the non-linear function $CI(a|x_i,f)$ allows the causal importance of an input variable i to be shown.

This approach has in common with the standard 'what if?' simulation the fact that it relies on the generalisation capabilities of the model. Therefore, $CI(a|x_i,f)$ does not analyse the causal importance of an input i for the classification itself. The plot of the function $CI(a|x_i,f)$ shows the causal importance of an input i for the model used to perform the classification task.

2.3. Predictive importance or saliency

The variable selection task consists of suppressing irrelevant input variables. When correlated variables have been removed, we need to measure and to compare each saliency of each input to select the smallest set of input variables which does not degrade the performances. The input variables with a low saliency are removed. The causal importance cannot be used directly. We can define the saliency as:

$$S(x_i|f) = \int_a \left| \int_{x_i} P(x_i)(f_i(x_i + a) - f_i(x_i)) \mathrm{d}x_i \right| \mathrm{d}a$$

where $f_i(x_i)$ is the function defined by the neural network when all the inputs $j \neq i$ are fixed and $P(x_i)$ the probability to observe the value x_i. This saliency is a scalar. The value of the saliency of x_i is null when $f_i(x)$ is a constant for all x: x_i is irrelevant for the model. If the perturbation around the input x_i changes the output of the model, the area between the $f_i(x_i + a)$ and $f_i(x_i)$ increases (Fig. 2) and $S(x_i|f)$ measures the strength of the perturbation. This saliency measurement is, therefore, non-local and it depends on the interval of variation of a.

The definition above does not take into account the true interval of variation of x_i. For example, if x_i is a positive variable, using the previous formula we can evaluate the saliency for x_i negative. To take into account the possible values of the input variable, we use:

$$S(x_i|f) = \int_a \left| \int_{x_i} P(x_i) P(a|x_i)(f_i(x_i + a) - f_i(x_i)) \mathrm{d}x_i \right| \mathrm{d}a$$

where $P(a|x_i)$ is a prior on the possible values of x_i: x_i can be discrete, positive, bounded, etc. The knowledge needed to define the prior depends only on the type of the input variable. For instance, for a binary variable, the prior can

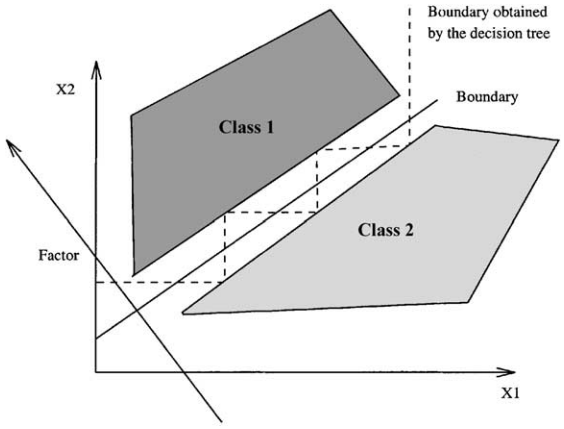

Fig. 3. The staircase effect: on this toy problem, the true variable is in a one-dimensional factor space obtained by a linear combination of the inputs. The splits produced by the tree can obtain accurate performance in classification, but cannot be used to interpret the classification.

be: $P(x_i + a = 1) = 1/2$ and $P(x_i + a = 0) = 1/2$ and 0 elsewhere.

For a linear model and continuous variables, the above definition reduces to:

$$S(x_i|f) = \int_a \left| \int_{x_i} P(a, x_i) w_i a \mathrm{d}x_i \right| \mathrm{d}a = |w_i| \int_a P(a)|a|\mathrm{d}a$$

$$= K|w_i|$$

where K is a constant. Therefore, in the case of linear models, our saliency measurement $S(x_i|f)$ orders the predictive importance of the input variables by the size of weights, as expected.

Approximating $P(x_i)$ by the empirical distribution of the training set, we obtain the following definition of the saliency:

$$S(x_i|f) \approx \frac{1}{N} \int_a \left| \sum_k P(a|x_i)\left(f_{i,k}(a) - f_{i,k}(0)\right) \right| \mathrm{d}a$$

Since this definition of the saliency does not rely on the specifities of the model, but only on its output variations, it can be applied to any model and not only to multilayer perceptrons.

In the above, we detailed the case of a classification between two classes. This approach can be extended to the case of a classification between n classes: in this case, $S(x_i|f)$ is a vector with n components and the saliency index can be defined, for instance, as the norm of this vector.

The same approach can also be applied in the case of regression problems.

3. Interpretation of neural network classification

3.1. Previous work

A first approach to interpret the classification obtained by a neural network is to cluster the examples using sets of simple rules reproducing the output of the model. Decision trees, trained to reproduce the output of the neural network, allow the clusters (the terminal nodes) and the rules (the set of ordered splits) to be obtained directly. This property makes the decision tree very popular in the data mining community. The algorithm TREPAN (Craven & Shavlik, 1996) uses a decision tree to approximate the function reached by a neural network. The extracted decision tree uses the generalisation ability of the multilayer perceptron to produce a lesser number of nodes than a standard decision tree built directly on the data. It exhibits better performances in generalisation. However, the decision tree is not well suited to explain a neural network classification. The neural network produces non-linear factors on the hidden layer which have to be approximated by a series of splits in the input space (Fig. 3). Such an approximation may be accurate, but at the cost of generating many leaves, making the a posteriori interpretation of the tree difficult.

A more interesting approach is to build a decision tree in the representation space of the hidden layer instead of the input space. Such a standard decision tree directly translates into an oblique tree (an oblique tree is a decision tree where the splits are made according to a linear combination of the input variables (Shesha & Sastry, 1999)) in the input space since the activation functions are monotonous. Such oblique decision trees have been shown to be much smaller and at least as accurate as decision trees built directly in the input space (Setiono & Liu, 1999).

Interpretation of an oblique tree in terms of the input variables is, however, very difficult since the same variables can appear with opposite influences in the successive splits defining a leaf. We can have, for instance, a leaf defined by the successive splits $x_1 + x_2 + x_3 > S_1$ and $x_1 - x_2 + x_4 > S_2$; the influence of x_2 on this leaf is difficult to assess.

Another approach is to simplify the neural network to obtain a model as simple as possible (Duch, Adamczak & Grabczewski, 1998; Jackson & Craven, 1996; Vaughn, 1999). In this case of constrained learning, there is a trade-off to find between the complexity and the accuracy of the neural network.

- A very sparse neural network is easy to interpret, but its accuracy can be low for real-world problems.
- A more complex neural network can perform better, but cannot be interpreted.

3.2. Clustering data in factor space

Our approach is to separate the clustering from the interpretation of the clusters. The criterion we use to produce a partition is based on the factor space (the hidden layer) where the data are linearly separable. For each example k, we calculate $I_{k,j}$, the relative contribution of the factor j

Table 1
Results for the variable selection task on the three datasets

Database	Number of inputs	Classification all inputs (%)	Number of selected inputs	Classification of selected inputs (%)
Voting	15	93	3	92
Churn	13	95	7	95
Heart	20	81	4	83

(the hidden unit j) to the output of the neural network:

$$I_{k,j} = \frac{w_j\left(z_j^k - \bar{z}_j\right)}{\sum_p \left|w_p\left(z_p^k - \bar{z}_p\right)\right|}$$

where w_j is the weight between the hidden unit j to the output neuron, z_j^k is the output of the hidden unit j for the example k, and $\bar{z}_j$ is the mean value of z_j. Notice that $I_{k,j} \in [-1, 1]$. A negative (resp. positive) value shows a negative (resp. positive) influence towards the output $y = 1$. A large absolute value means a large influence of the feature detected by the hidden neuron j.

To produce the partition, we evaluate $I_{k,j}$ for each hidden unit to obtain a vector I_k with h components (h is the number of hidden neurons) representative of each example. We train an unsupervised Kohonen map (Kohonen, 1984) with a one-dimensional topology using the N vectors I_k as the input. The number of clusters will be the number of neurons of the map: each neuron of the Kohonen map codes a mean position in the space of the I_k and corresponds to a cluster composed of the nearest examples. The number of neurons of the map is increased until the resulting segmentation exhibits a classification performance similar to the model.

3.3. Interpreting the influence of variables on a cluster by cluster basis

At this point, we can analyse the influence of the input variables on a cluster by cluster basis. For a given cluster, we determine the most important variables and investigate their causal importance. To select the most important variables for a given cluster, we introduce the saliency *relative to a cluster* C_m as:

$$S(x_i|f,m) = \frac{1}{N}\int_a \left|\sum_{k|x^k \in C_m} P(a|x_i)\left(f_{i,k}(a) - f_{i,k}(0)\right)\right| \mathrm{d}a$$

and we introduce the causal importance of an input variable *relative to a cluster* C_m as:

$$CI(a|x_i,f,m) = \sum_{k|x^k \in C_m} f_{k,i}(a)$$

4. Experimental results

4.1. The datasets

We use three benchmark datasets to test the accuracy of the proposed methodology.

- Congressional voting dataset (UC-Irvine): 15 ordered discrete input variables, 435 examples. We remove the physician fee freeze feature to compare our results with other works.
- Heart disease dataset: six continuous input variables, one discrete ordered input variable, three binary input variables, and three nominal variables. We recode the three nominal input variables as binary variables to obtain 20 input variables. This set contains 271 examples.
- Churn dataset: 20 input variables. We suppress the first three (state, phone number, area code). The input variables total day billing, total evening billing and total night billing are linearly correlated, respectively, with total day minutes, total evening minutes and total night minutes. We analyse two binary input variables, and the 11 continuous independent input variables. The set contains 5000 examples.

The input variables are centred–reduced. We use a validation set to select the optimal architecture and to stop the training. Ten neural networks are trained for each architecture. The results are shown on the test sets for the three benchmark databases for variable selection, clustering and interpretation.

4.2. Results for the variable selection task

We discuss the variable selection for the voting dataset. A summary of the results for the three datasets is given in Table 1.

Considering that the discrete input variables for the voting dataset (voted for, paired for, and announced for) can be viewed as a continuous variable coding the agreement for a law, the prior we use to evaluate the saliency of each input variable is very simple: $P(a|x_i)$ has a uniform distribution for $a + x_i \in \left[-2Var(x_i), 2Var(x_i)\right]$.

Fig. 4 shows the results our saliency measurement $S(x_i|f)$ and of the saliency evaluation for all inputs according to the input removal technique defined as

$$\sum_{k=1}^{N} \left|f\left(x_1^k, \ldots, x_i^k, \ldots, x_n^k\right) - f\left(x_1^k, \ldots, x_i^k = 0, \ldots, x_n^k\right)\right|$$

for centred variables; this technique is usually considered as the most reasonable one (Sarle, 1998).

The order of the input variables sorted by the two saliency measurements is different. According to the proposed

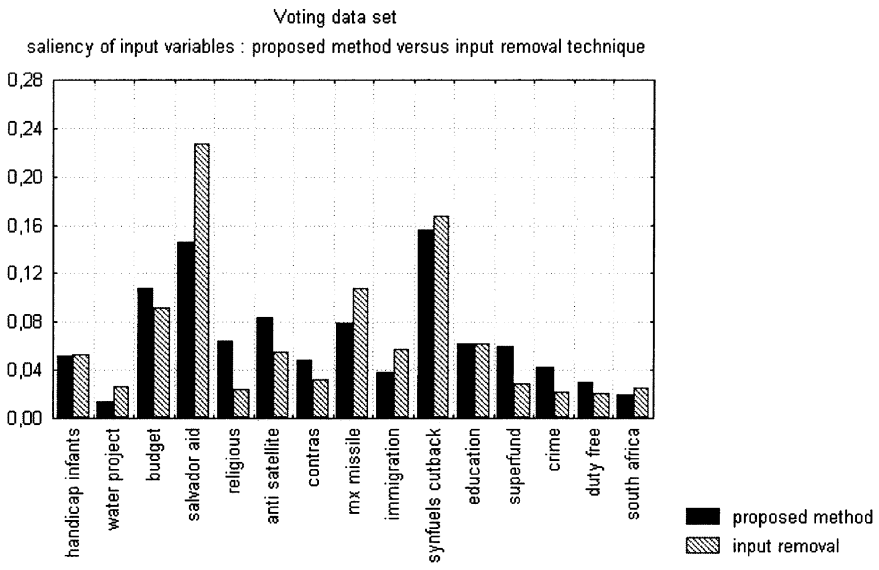

Fig. 4. The saliency according to input removal technique and the saliency according to the proposed method. Each value corresponds to the mean value on 10 trainings.

method, the smallest subset of input variables which does not degrade the classification performance, is composed of three variables (Fig. 5): *budget resolution*, *Salvador aid*, and *synfuel corporation cutback*. The classification performance of a model using these three variables only is 92%. If we select the set of three most important variables according to the input removal technique (*budget resolution*, *mx missile*, and *synfuel corporation cutback*), the obtained classification performance is only 88%, showing that our technique selected a better ordering for the most important variables.

On the three benchmark databases, the proposed saliency measurement allows input variables without degrading performances to be discarded. This variable selection simplifies the next step: the clustering and interpretation of the classification.

4.3. Results for the interpretation

We use the churn dataset as an illustrative example to discuss the clustering and interpretation ability of our methodology. The results of clustering and interpretation for the three datasets are given in Tables 2– 4.

Churn (or attrition) is an increasingly important problem in the telecommunication arena because of increased competition and because of technical innovations allowing easier migration from one provider to another. Modelling churn and more importantly *understanding* churn is an important step of pro-active fidelisation programmes as currently developed by most providers: the reasons for churn may differ for different groups of clients and a segmentation and its explanation is mandatory to take the appropriate actions.

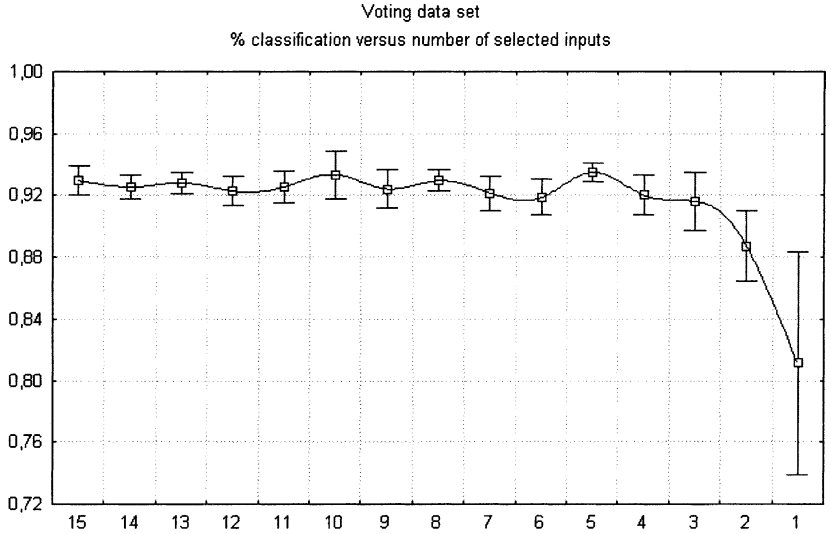

Fig. 5. Results for the variable selection tasks for the voting dataset. Each value corresponds to the mean value on 10 trainings. The error bar shows the 95% confidence interval.

Table 2
Results for the clustering for the churn dataset

Cluster	Interpretation	Class of the cluster	Classification
1	Perfect except for those who call the customer service	Non-churner	92% (426/461)
2	Need a plan for day minutes	Non-churner	97% (368/380)
3	Need voice mail plan, don't need international plan	Churner	79% (45/57)
4	Need to call (or be called by) the customer service	Non-churner	57% (58/102)
Total			90% (897/1000)

Table 3
Results for the clustering for the voting dataset

Cluster	Interpretation	Class of the cluster	Classification
1	Against Salvador aid and for budget	Democrat	94% (82/87)
2	For Salvador aid and against synfuels corporation cutback	Republican	89% (55/62)
Total			92% (137/149)

Using the saliency measurement, we selected seven independent input variables. The number of hidden units is determined using the structural minimisation framework (Vapnik, 1995). We obtain a neural network composed of seven input neurons, and nine hidden neurons.

We proceed then to the clustering of the data in factor space as described in Section 3.2. The classification performance of clustering with an increasing number of clusters is given in Fig. 6. From this figure, we chose a clustering with four clusters.

We plot the mean value of centred–reduced input variables for each cluster (Fig. 7). The mean value of an input does not correspond to its predictive or causal importance. However, the mean values of each cluster inform us about the 'nature' of the cluster as explained in the legend of Fig. 7.

To develop the interpretation of the neural network classification, we plot the saliency of input variables for each cluster (Fig. 8). We concentrate our analysis on the cluster 3 which gathers the churners with heavy usage (see Fig. 7). For this cluster, we analyse the variables which have the largest saliency (see Fig. 8): international plan, voice mail plan and total day minutes. Fig. 9 shows the causal importance of these three variables relative to cluster 3.

The causal importance of the input variable *international plan* indicates that this plan does not correspond to the usage of the clients of the cluster 3: many short international calls (from Figs. 7 and 9). For these clients, increasing the value of this input variable does not imply a decrease of the ratio of clusters. Moreover, decreasing the value of this input variable implies a decrease of the ratio of churners. The causal importance of the input variable *total day minute* is very simple: the heavier their usage, the more these clients churn. The analysis of the causal importance of the input variable *voice mail plan* shows that this plan corresponds to the usage of these clients (Fig. 9). Moreover, these clients have a low ratio of voice mail plan (Fig. 7). We conclude

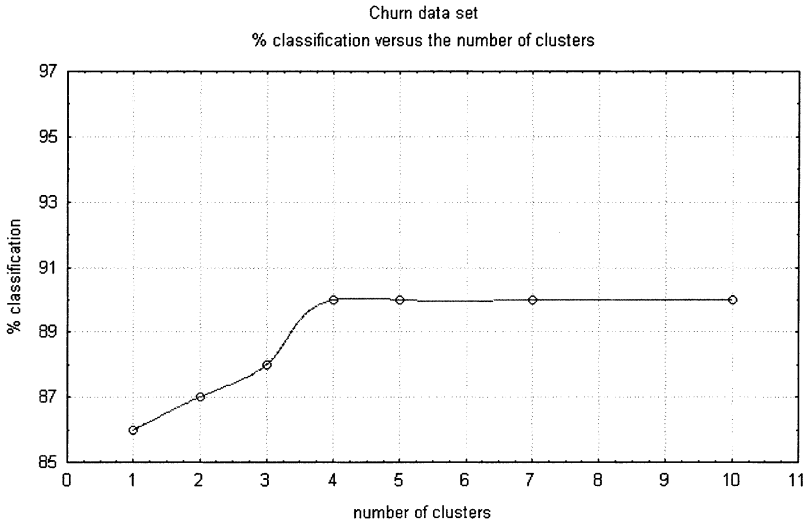

Fig. 6. The choice of the number of clusters corresponds to a trade-off between the accuracy of the classification and the number of clusters.

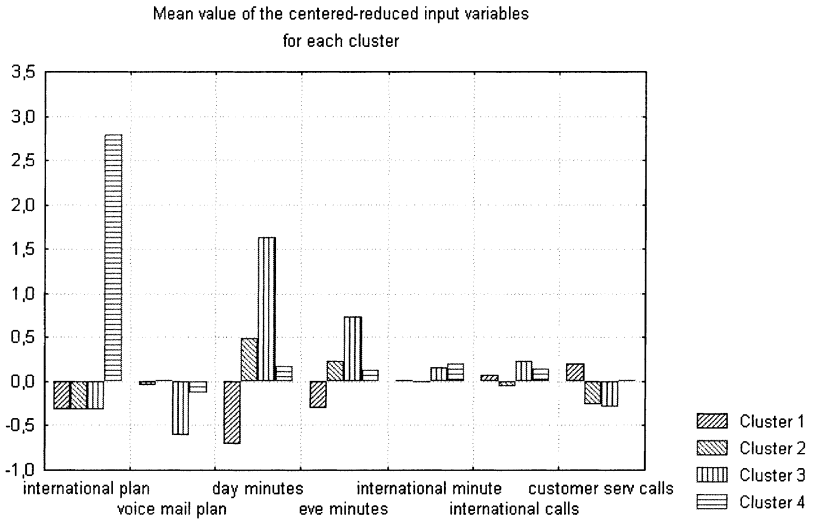

Fig. 7. The mean value of input variables for each cluster. The cluster 1 (non-churner) is composed of clients with light usage. The cluster 2 (non-churner) is composed of clients with moderate usage. The cluster 3 (churner) is composed of clients with heavy usage. The cluster 4 (non-churner) is composed of clients with moderate national usage, heavy international usage and using an international plan.

Table 4
Results for the clustering for the heart disease dataset

Cluster	Interpretation	Class of the cluster	Classification
1	Chest pain type 4 and high number of major vessels increases risk	Disease	76% (35/46)
2	Low value of number of major vessels decreases risk	No disease	91% (42/46)
Total			84% (77/92)

that these clients have a heavy usage, both national and international, and are presumably angry with their bills (the heavier the usage, the more they churn); offering them the international plan would not help since it does not fit their usage, as reflected by the causal importance of this variable. Offering them the voice mail plan might help fidelising them.

The same analysis is applied to the three datasets and leads to the interpretations given in Tables 2–4.

5. Discussion

Defining a performance measure for the interpretation of classifiers is a formidable task which is beyond the scope of this paper. Moreover, most previous works in the neural network area do not make their interpretation of the classification results explicit. Below, we shall only give a few numerical comparisons of our results with previous works.

We can compare the complexity of the model obtained.

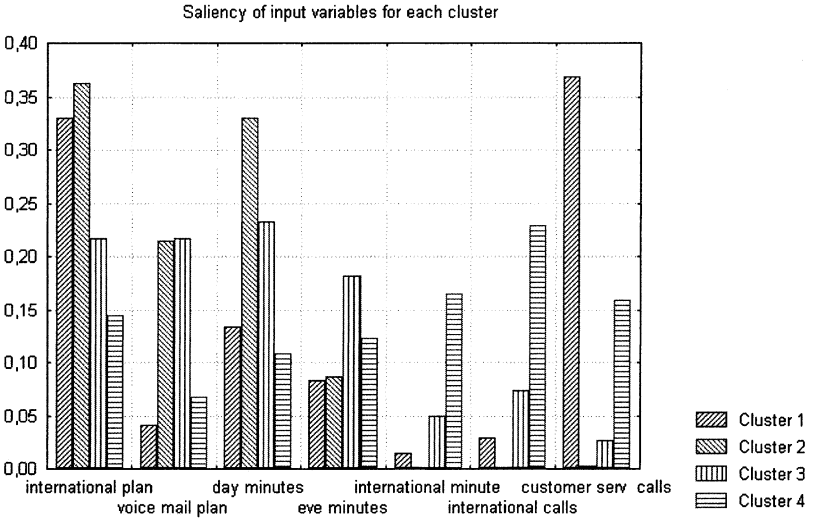

Fig. 8. Saliency of input variables for each cluster. The saliency measurements are normalised by cluster for the comparison.

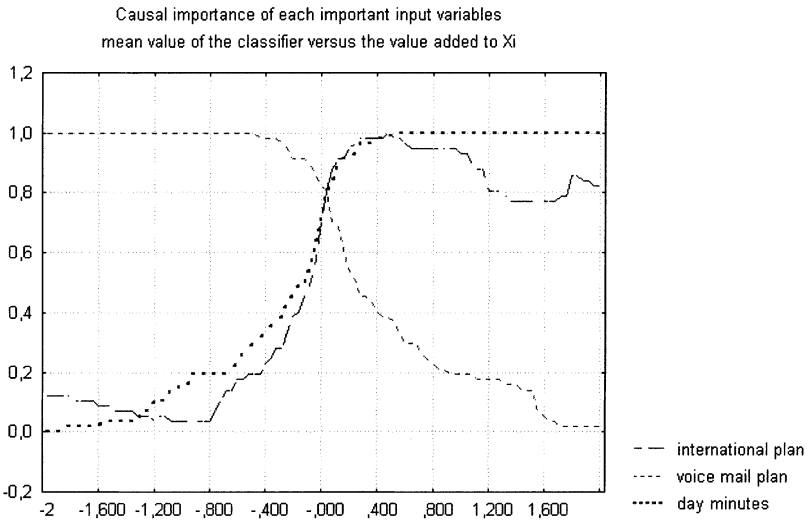

Fig. 9. Causal importance of the three input variables which have the largest saliency for the cluster 3.

Learning sparse perceptron (Jackson & Craven, 1996), obtained a perceptron composed of 12 weights trained on the voting dataset. Using our saliency measurement to select the smallest set of inputs, and structural risk minimisation framework to select the number of hidden units, we obtain for the same performance with a multilayer perceptron composed of three inputs and two hidden neurons, which corresponds to 11 weights. In this case, our methodology leads to a network of similar complexity, even if minimising complexity (defined as the number of weights) was not our goal.

The algorithm TREPAN (Craven & Savlik, 1996) learns the function defined by the multilayer perceptron with a decision tree. On the voting and heart disease databases, this algorithm uses 21 splits on input variables. Using our methodology, we found that, respectively, for the voting database and the heart disease database, three and four input variables with two hidden neurons are sufficient. Moreover, the interpretation of the classification relies on two clusters only and is fairly simple in both cases. We suspect the staircase effect (see Fig. 3) to explain the high number of splits reported for TREPAN.

Regarding the computation cost of the techniques described above, we note the following:

- our variable selection scheme differs from the input removal technique only by the choice of a different criterion for the ordering of the variables. The criterion proposed here, the *saliency*, is more costly as it requires the scoring for each variable of a few (typically 20) offset versions of the training set instead of just one offset version for the input removal technique. This cost is, however, negligible as compared to the training cost of the neural network itself, so that computing the *saliency* has a negligible impact on the overall efficiency of the variable selection task; and
- the interpretation task requires building one-dimensional Kohonen maps in feature space with a number of units ranging from two to typically 20 (adding more units may increase the classification performance of the clustering, but at the cost of the interpretation: remember that the neural network delivers the performance and that the clustering in feature space should primarily allow the interpretation of the model). Such small maps are fast to train.

Therefore, the variable elimination and model interpretation phases proposed here do not add much cost to the overall processing, which is dominated by the training of the neural network.

6. Conclusions

The saliency measurement allows to select the smallest set of input variables to simplify a neural network, when the input variables are uncorrelated. This saliency measurement can be applied to non-linear models, and it can be extended to classifications between n classes or regression tasks.

The clustering algorithm and the graphical analysis of the causal importance relative to each cluster allow the neural network classification to be interpreted without adding constraints on the training and without using another classifier. Hence, our methodology preserves the performance reached by the neural network.

References

Baxt, W. G., & White, H. (1995). Bootstrapping confidence intervals for clinical inputs variable effects in a network trained to identify the presence of acute myocardial infarction. *Neural Computation*, *7*, 624–638.

Burkitt, A. N. (1992). Refined pruning techniques for feed-forward neural networks. *Computer Systems*, *6*, 479–494.

Craven, M. W., & Shavlik, J. W. (1996). Extracting tree-structured

representations of trained network. *Neural Information Processing Systems*, *8*, 24–30.

Duch, W., Adamczak, R., & Grabczewski, K. (1998). Extraction of logical rules from neural network. *Neural Processing Letters*, *7*, 211–219.

Gallinari, P., Thiria, S., Badran, F., & Fogelman-Soulié, F. (1991). On the relations between discriminant analysis and multilayer perceptrons. *Neural Networks*, *4*, 349–360.

Garson, D. (1991). Interpreting neural-network connection strengths. *AI Expert*, 47–51.

Hassibi, B., & Stork, D. G. (1993). Optimal brain surgeon. *Neural Information Processing*, *5*, 164–171.

Jackson, J. C., & Craven, M. W. (1996). Learning sparse perceptrons. *Neural Information Processing Systems*, *8*, 654–660.

Kohonen, T. (1984). *Self-organization and associative memory*, Springer-Verlag, New York.

Le Cun, Y., Denker, J., & Solla, S. (1990). Optimal brain damage. *Neural Information Processing Systems*, *2*, 598–605.

Mak, B., & Blanning, R. W. (1998). An empirical measure of element contribution in neural networks. *IEEE Transactions on Systems, Man, and Cybernetics*, *28* (4), 561–564.

Moody, J. (1994). *Prediction risk and architecture selection for neural networks. From statistics to neural networks-theory and pattern recognition*, Springer-Verlag, New York.

Réfénes, A. N., Zapranis, A., & Utans, J. (1994). Stock performance using neural networks: a comparative study with regression models. *Neural Networks*, *7*, 375–388.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning internal representations by error propagation. In: *Parallel distributed processing: explorations in the microstructures of cognition* (vol. 1). Bradford Books/MIT Press, Cambridge, MA, pp. 318–362.

Sarle, W. S. (1998). *How to measure the importance of inputs?* Technical Report, SAS Institute Inc., Cary, NC, USA. ftp://ftp.sas.com/pub/neural/FAQ.html

Setiono, R., & Liu, H. (1999). A connectionist approach to generating oblique decision trees. *IEEE Transactions on Systems, Man, and Cybernetics*, *29*(3), 440–444.

Shah, S., & Sastry, P. S. (1999). New algorithms for learning and pruning oblique decision trees. *IEEE Transactions on Systems, Man, and Cybernetics*, *29*(4), 494–505.

Vapnik, V. (1995). *The nature of statistical learning theory*, New York/Heidelberg/Berlin: Springer-Verlag.

Vaughn, M. L. (1999). Derivation of the multilayer perceptron weight constraints for direct network interpretation and knowledge discovery. *Neural Networks*, *12*, 1259–1271.