Session Number: POSTER PAPER SESSION

Paper Prepared for the 29th General Conference of
The International Association for Research in Income and Wealth

Joensuu, Finland, August 20 – 26, 2006

A neural network architecture for data editing in the Bank of Italy's business surveys

Claudia Biancotti, Leandro D'Aurizio and Raffaele Tartaglia-Polcini

For additional information please contact:

Claudia Biancotti, Leandro D'Aurizio and Raffaele Tartaglia-Polcini
Bank of Italy, Economic Research Department, via Nazionale 91, 00184 Rome, Italy
E-Mail: raffaele.tartagliapolcini@bancaditalia.it
FAX : + 39 06 4792 4764
Telephone: + 39 06 4792 1

This paper is posted on the following websites: http://www.iariw.org

A neural network architecture for data editing in the Bank of Italy's business surveys

Claudia Biancotti(*), Leandro D'Aurizio(*) and Raffaele Tartaglia-Polcini(*)

Abstract

This paper presents an application of neural network models to predictive classification for data quality control. Our goal is the identification of data affected by measurement error in the Bank of Italy's business surveys. We build an architecture consisting of three feed-forward networks (for variables related to employment, sales and investment respectively): the networks are trained on input matrices extracted from the error-free final survey database for the 2003 wave, and subjected to stochastic transformations reproducing known error patterns. A binary indicator of unit perturbation is used as the output variable. The networks are trained with the Resilient Propagation learning algorithm. On the training and validation sets, correct predictions occur on about 90 per cent of the records for employment, 94 per cent for sales, and 75 per cent for investment. On independent test sets, the respective quotas average 92, 80 and 70 per cent. Neural networks as classifiers perform much better than logistic regression, one of the most popular competing methods, on our data. They appear to provide a valid means for improvement in the efficiency of the quality control process and, ultimately, in the reliability of survey data.

Keywords: data quality, data editing, binary classification, neural networks, measurement error.

CONTENTS

1. Introduction 4
2. The case for neural networks 5
3. Data quality in the Bank of Italy's business surveys 6
4. The data 7
   4.1 The initial dataset and the error-generating process 7
   4.2 The selection of the variables 8
   4.3 The training and validation datasets 9
5. Network architecture, estimation strategy and results 10
   5.1 The general features of the architecture 10
   5.2 The evaluation 10
   5.3 Employment 11
   5.4 Sales 14
   5.5 Investment 17
   5.6 An interpretation of the results in terms of density estimation 21
6. Conclusions and further developments 22
Appendix: Methodological issues 25
   Basics on Neural Networks 25
   Basics on the applied architecture 26
   The software 28
References 29

(*) Bank of Italy, Economic Research Department.

1. Introduction

Attention to quality has become a central preoccupation for data producers. In a statistical framework, quality control is routinely called "editing". A definition widely agreed upon goes as follows: “[data] *editing is the activity aimed at gathering intelligence related to significant differences in the data for analytical purposes, providing feedback that can lead to improvements in data collection and processing, reducing the level of error present in the data and ensuring a degree of consistency, integrity, and coherence*” (Chinnappa et al. 1990). Under the Total Quality Management approach, quality equates to fitness of survey microdata for purposes of research or statistical information. Brackstone (1999) lists six dimensions of quality: *relevance, accuracy, timeliness, accessibility, interpretability* and *coherence.* The concept of interpretability is also meant to encompass the provision of useful indications about the accuracy of information, in the guise of documentation and metadata. In this context, interpretability is enhanced both by making the user aware of the correct application of the editing methods employed to rectify wrong data, and providing synthetical indicators about the spotted anomalies. Granquist and Kovar (1997) give a thorough account of the resource problems induced by the need for data quality control, and relate some problems of the traditional editing methods. According to them, costs related to quality may sum up to 40 per cent of the whole cost of a typical business survey.

Strategies for automated and semi-automated error detection are constantly improving. Macro-level editing methods are mostly concerned with maximizing the reliability of summary statistics and regression-based estimates. In an effort towards timeliness and cost control, measurement error resulting in serious distortions of the final results is fought, while mistakes of little consequence are overlooked (see for example Battipaglia, 2002). A macro-editing procedure normally ensures quality enhancement of the final estimates through the selection of the significant units to re-contact. The most evident drawback of these techniques is their low “hit rate” (i.e. the share of the number of flags that result in changes of the original data) (Hedlin 2003): for re-contacted firms, confirmations of apparently anomalous values are far more numerous than revisions. There is also a price to pay: progressive segmentation of the sample between frequently monitored (since they are more influential on the final estimate) and less frequently monitored (less influential) units occurs. Typically, for estimates of totals and rates of change, the big firms are more likely to be considered influential and therefore re-contacted.

Micro-level approaches, generally more time-consuming and expensive, complement the macro approach and may provide a remedy to this problem by allowing a reliability assessment of single items, regardless of their impact on a pre-defined set of estimates. This is very important whenever data end up being used for a wide variety of research purposes that are not predictable *ex ante*.

This paper explores the use of artificial neural networks (ANNs) as a micro-editing tool for the Bank of Italy's business surveys. It has a strong applied orientation, dealing with a specific neural network architecture selected for error detection and data editing for the Bank of Italy's annual business surveys. According to Breiman (1994, 2001), the very nature of neural network modeling makes it highly dependent on the particular data at hand and the search of the best network in terms of predictive accuracy is strongly problem–oriented. Section 2 briefly makes the case for the use of neural networks in our context. Section 3 presents the surveys and their data quality issues. Section 4 provides details of the data sets used for our experiment. Section 5 discusses network architecture, estimation strategies and results. Section 6 concludes and indicates the way for further developments. For the sake of brevity, the methodological tenets of neural computing are not explained in the paper: the Appendix provides a brief summary and references.

2. The case for neural networks

Supporters and critics of ANNs balance their arguments. A proven high predictive capability stands against some unresolved issues that still punctuate the existing theoretical framework: for example, the absence of optimality criteria for the choice of the network topology (e.g., how many hidden layers, how many hidden units, how to place arcs), the activation function, the learning algorithm, the weight adjustment rule; the want of a quality assessment for the predicted values.

From an applied mathematical perspective, however, the developers of neural networks can claim the *universal approximation property* (Hornik, Stinchcombe and White 1989; for a heuristic, suggestive proof and extensions, see Ripley 1996): any real-valued, continuous function of real variables can be uniformly approximated on compacta by a neural network with a logistic or threshold squasher and one hidden layer.

From this point of view, ANNs can be seen as computer-intensive, specification-free nonlinear fitters; many existing multivariate regression models can be seen as restrictions of this general framework. In our research, aimed at efficient error-spotting and effective data quality enhancement rather than interpretability, ANNs come as a tool allowing detection of mistakes on many different variables, taking into account their interaction without having to describe the connections between observed features of the respondent and erring propensity by way of possibly under-performing "structural" equations.

Although neural computing is a relatively new tool in the context of editing, studies on the subject can be dated back to Roddick (1993); discussions of the topic from a mixed theoretical-empirical standpoint can be found in Nordbotten (1995, 1996), Larsen and Madsen (1999), Biancotti and Tartaglia-Polcini (2004) among others. Implementation details of ANN-based editing strategies feature in several operational manuals of national statistical offices and research institutions. Machine-learning methods for predictive classification are interesting in that they have the potential to outperform their traditional counterparts (such as discriminant analysis or logistic regression) because of their ability to quickly adapt to changes in the structure of phenomena (Rohwer, Wynne-Jones and Wysotzki, 1997). In laymen's terms, if we can train a computer to see mistakes in a dataset, without being too explicit in what kind of error to look for, we can save vast amounts of time and achieve greater accuracy under given budget and time constraints.

3. Data quality in the Bank of Italy's business surveys

The Bank of Italy has been carrying out business surveys since 1972, on a yearly basis. Up until 1998 only manufacturing firms with more than 49 employees were included in the sample; the reference population gradually grew starting from the following year, and now covers firms with at least 20 employees belonging both to the industrial sector and the private services (excluding banking and insurance). The sample includes some 4,000 firms. Units were originally chosen at random under a stratified design, and always re-contacted in all waves following the one during which they had been drawn, provided that they still belonged to the target population. Refusals and firms no longer in the target population are routinely replaced with similar units. Interviews are made in the first months of the year $t+1$ and concern data about the years $t-1$ and t, together with forecasts for the current year $t+1$. Data collection is conducted within the Bank, through the local branches.

The collected variables range from levels of sales and investment to those of indebtedness and other sources of financing (expressed in thousands of euro). Typical sources of measurement error arising in this context are:

1. response in euro instead of thousands of euro, or in former national currency units instead of euro: this latter problem is particularly acute in the years immediately following the adoption of the new currency;
2. misclassification of aggregates to be included in the amounts;
3. mergers or acquisitions not correctly handled[1];
4. misreadings in the paper questionnaire (either manual or through optical character recognition).

Data quality is taken care of in successive steps. Simple checks are already implemented at the data-entry level with the use of CAPI (Computer Aided Personal Interviewing)[2]. A further level

[1] Firms that went through mergers or acquisitions are considered only if data come from the same set of local units and employees for the three years $t-1$, t and $t+1$. This happens either by fictitiously pre-dating the merger/acquisition event to the beginning of year $t-1$ or by postponing it to the end of year $t+1$. Measurement error arises from mistakes in these adjustments.

of control consists in checking for admissible ranges of quantitative variables, based on past distributions or subject matter specialists' opinions values outside the expected range must be double-checked by the interviewer and a flag activated in case the value is confirmed. The final stage relies on a selective editing procedure. A linear model for the published estimates (for example, rates of change for sales and investment) is fitted and the predicted value is trusted as the "true" one under the model. A first-order Taylor approximation (the "score") evaluates the contribution of the single units to this prediction. Units with the highest score level are labeled as outliers. Anomalous units are normally re-contacted.

4. The data

4.1 The initial dataset and the error-generating process

In order to conduct our experiment, we build a dataset based on actual business survey microdata. We generate errors by perturbing large subsets of correctly valued quantitative survey variables, more prone to measurement errors.

We use simulated errors instead of historically recorded ones because by doing so we can accurately keep track of correct and wrong data.

We need to produce a significant percentage of errors, since the estimation of a neural network for classification calls for a fair-sized class of wrong records.. The choice enables the training to yield robust results, given that training an artificial neural network for classification is an iterative process liable to produce wrong results if one class size is too low. The frequency of correct data hovers around the threshold of 50-55 % and is certainly too low if compared to real survey scenarios; but, as discussed below, the set of rules upon which the perturbations are based carefully reproduce actual error patterns to provide a realistic training set for the network. The learning process is not biased by the high frequency of errors, which is merely instrumental to its iterative nature.

Our simulated datasets are based on the final archives for the business surveys conducted during 2004. Typically, questions are asked about the two previous years (2002 and 2003 in this case). We assume that these archives include no leftover errors, having undergone the control and editing procedures detailed in Section 3.

Five main categories of perturbations are introduced, covering the widest possible range of error types: substitution of the original value of a variable with a zero; generation or elimination of a zero digit in a variable; units/thousands/millions arbitrary transformations (addition of subtraction of three zero digits at the end of a variable); swapping of values of a randomly chosen variable

[2] For example, negative values are not accepted if they do not have economic meaning; answers to discretely coded variables can only belong to a pre-defined list.

between two firms surveyed by the same interviewer; random generation of tail values[3] from the empirical distribution, in order to account for residual sources of measurement error.

4.2 The selection of the variables

The selection of the variables to be considered in the experiment poses a number of problems. The scope of our networks should not be confined to predicting whether a particular record somewhere contains an error. While already a relatively interesting accomplishment from the machine-learning standpoint, it is not operationally useful since a generic indication would force us to go through all the suspect variables manually in order to single out the mistake. We need some pointers to where the error is within the record, ideally to the level of the specific variable. The perfect editing device should be trained on all the 200-odd survey questions, but this is easier said than done. A first important hurdle is represented by our estimation algorithm that – in the fashion of nearly all estimation algorithms – does not allow for missing values: we are forced to exclude variables that are seriously affected by item non-response. The elimination of all records with at least one missing element would train the network on a sample that is too narrow; performing estimates on the basis of imputed values would be even worse because standard methods of imputation artificially reduce the variance in the regressors, hurting the informational content of the results. Moreover, in order to obtain results that can be extended to a typical wave of the survey we have to exclude one-shot sections: alternatively, the model should be modified for every wave of the survey. We also forego questions routinely asked only to a part of the sample, because they might have peculiar error patterns, correlated with the very inclusion of a firm in the subsample. Forecasts are left out because of the impossibility of separating measurement error from forecasting error, and of the high frequency of missing values.

These difficulties lead us to choose the solution of disclosing errors only for variables related to the three core topics investigated in the questionnaire: employment, sales and investment. These phenomena, anyway, deserve maximum priority in any plan of quality improvement: the survey's main objective consists in the evaluation of short-term dynamics of macroeconomic aggregates. The relevant estimates are the only ones that get published every year without exception, and they tend to draw the highest attention from the readership.

There are twenty such core variables: twelve for employment, four for sales, and four for investment. The perturbation algorithms presented in Section 5 are applied to these twenty variables, and then three neural networks are separately trained to localize the general presence of errors within each of the groups.

[3] These perturbed data are obtained by the following steps: the value to perturb is placed in the lower or upper tail if it is respectively below or above the median. A random percentile is then extracted to define the tail dimension and a final random drawing sets the position of the perturbed value inside the tail.

Some non-perturbed stratification variables, such as the geographical location and the sector of activity of the firm are also kept on the restricted dataset, which comprises 2,970 observations, out of around 4,200 firms participating in the survey.

4.3 The training and validation datasets

In order to train the networks, we produce three datasets of the same size with generated errors for employment, sales and investment. Mistakes are generated independently for the three variables categories, each containing 45-50 percent of wrong data. Each dataset is then randomly split in two equally sized samples, to be used respectively for training and validation.

Network architecture building starts from a decision on how to submit the data to the network: the same set of information can be presented in a variety of ways that differ a lot where explanatory power is concerned.

Since the chosen neural network acts like an enhanced multinomial logit (see the Appendix), we assumed that the use of stratification variables (strictly related to sampling design) as inputs would improve the classification task. This is proven effective only for sales, whereas we resolved to get rid of these for employment and investment, as they showed no contribution to learning.

As a rule, raw variables possess a limited solving potential and must be flanked by some transforms, typically ratios. Ratios works better than the raw input variables because a neural network, as much as any other binary classifier, has the primary goal of dividing space into acceptance and rejection regions. This is going to be harder as the dimensionality of the space increases, especially when the type of partition that has to be found on each dimension is not trivial. This is particularly true for outlier detection: the presence of "low" and "high" outliers would call for a partition of the space in separate regions. No banal threshold function would work; if we used hidden nodes with a logistic activation, we would have needed at least two nodes to perform this very simple activity on a single variable[4]. Ratios reduce the dimensionality of the problem, allowing for lighter networks and reducing the risk of overfitting, and are, moreover, able to turn data consistency problems into simpler range problems.

This point can be highlighted through a simple example: suppose the national firm–level average of yearly hours worked is around 300,000. The network must be trained to understand that both 1,000 and 5,000,000 are wrong, but must also beware if a firm declares 150,000 in 2002 and 500,000 in 2003; if only the raw values are used as inputs, a very large number of hidden nodes is needed to have a decent performance. If the ratio of hours worked per employee in 2002 to hours worked per employee in 2003 is used as input, the problem is reduced to finding ratios outside a

[4] If we were working on two variables, at least four hidden nodes would be required to spot both "low" and "high" outliers in each of those. An experiment showed that if both variables took wrong albeit reasonable values, the network could not extrapolate a rule, because of a dominating range effect.

reasonable range (say 0.8 to 1.2). Cases of both values multiplied by 1,000 are not caught, but they are anyway marked as wrong by adding a specific control on yearly hours worked per capita.

5. Network architecture, estimation strategy and results

5.1 The general features of the architecture

Three separate networks are trained: one catches mistakes in employment variables, one in sales, one in investment. We find that errors in employment can be singled out based on predictors of the same group, e.g. number of hours worked per employee, or percentage of overtime. In the case of sales and investment, the task is not quite as self-contained and some employment-related predictors are also needed, in the form of ratios per employee. This result calls for some extra care in how the networks are used: while a wrong number on investment will not affect how the system detects mistakes on employment, the vice versa is not true. An operational hierarchy is necessary: the employment network is to be run first and then the other networks, after getting rid of the mistakes singled out by the first one.

Topologically, the networks are built on the traditional feed-forward scheme. They feature large hidden layers with logistic activations, with the addition of one or two discrete–value nodes, directly connected with input nodes.

As for the estimation strategy, we always use the Resilient Propagation (Rprop) algorithm: traditional learning devices like vanilla backpropagation, or slight variations thereof, were no match to our problems nor produced good performances in terms of MSE. The mechanics of the Rprop algorithm are quite simple: signs are used instead of levels ("Manhattan learning") for derivatives and the learning rate is dynamically updated (Riedmiller and Braun 1993; see the Appendix). This has been shown to be suitable for data as heavy-tailed as ours.

We select the number of hidden nodes, keeping an eye simultaneously on training and validation error, so as to avoid overfitting and to keep their number to a minimum. We also tried more than one hidden layer, to no better results: this is not surprising, given the universal approximation property enjoyed by feed-forward neural networks with one hidden layer. The hidden nodes can be seen as detectors of abstract features of the space of variables, so that their number accounts for the dimension of the (possibly nonlinear) principal components space that filters the lower-variance noise signal (Rohwer, Wynne-Jones and Wysotzki, 1994).

5.2 The evaluation

The random perturbation of data regarded as "correct" and the following analysis of how well “wrong” data are pinpointed is a classical tool to evaluate editing techniques (ET) (see, for example, Barcaroli *et al*., 1997). Simple indexes help us in the assessment, for example the Error

Identification Capability (EIC), defined as the percentage of correctly identified wrong values. Two kinds of errors feature in any editing process: a) marking correct data as erroneous (identification error of wrong values); b) marking errors as correct data (identification error of correct values). An ideal ET should keep the frequency of the two errors to a minimum.

These simple evaluation tools are illustrated by the following two-way contingency table, which we will use in the course of our analysis.

True value	**Predicted value**	
	correct	wrong
correct	N_{11}	N_{12}
wrong	N_{21}	N_{22}

The EIC is $\frac{N_{22}}{N_{12}+N_{22}}*100$, while the two previously mentioned errors are the cells corresponding to the values N_{12} and N_{21} cases: they are kept to a minimum if $N_{11}+N_{22}$ (number of correct predictions) is as close as possible to the total $N_{11}+N_{12}+N_{21}+N_{22}$.

For each experiment, the network performance will be assessed against the logistic model[5] (a traditional binary classifier) for the training set, for the validation set and for two test sets: the latter are two equally-sized samples drawn from the original dataset and comprising all types of perturbations. Each table cell will present the number of cases and the total, row and column percentages.

In order to evaluate the network stability, we also present a graph with the MSEs on the training and validation datasets against the number of iterations (the MSEs for the test sets are omitted for brevity). The two plots should ideally decrease and keep close after the first iterations.

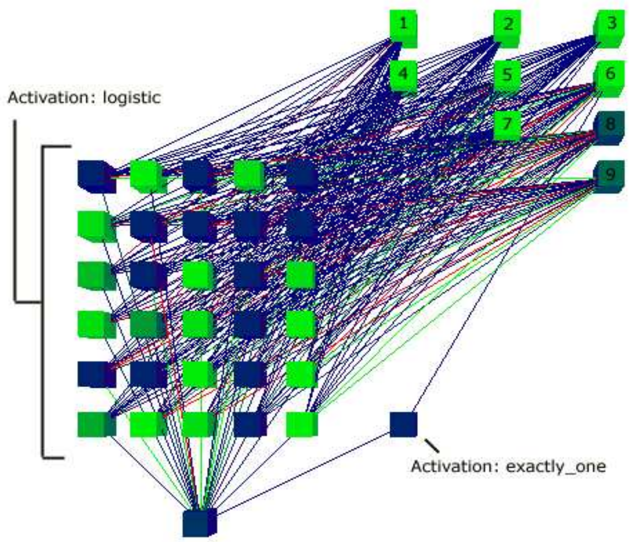
5.3 Employment

The employment variables affected by error in our dataset are as follows: average number of employees, end-of-the-year number of employees, total number of hours worked in a year, percentage of overtime hours, number of new hires, number of layoffs. Values for the year 2002 and the year 2003 (both drawn from the 2004 wave of the survey) are provided for each of those variables. As stated above, raw variables showed a rather limited predictive power: after an extensive number of trials, nine transforms were selected for inclusion in the network. Figure 1 illustrates the network topology; the list of variables is in the legend below.

[5] The logistic models presented feature the same input variables used for the neural network. We also tried to use other covariates, including all the stratification variables included in the survey design, to no substantial improvement of the results.

Figure 1

Neural network for error detection: Employment

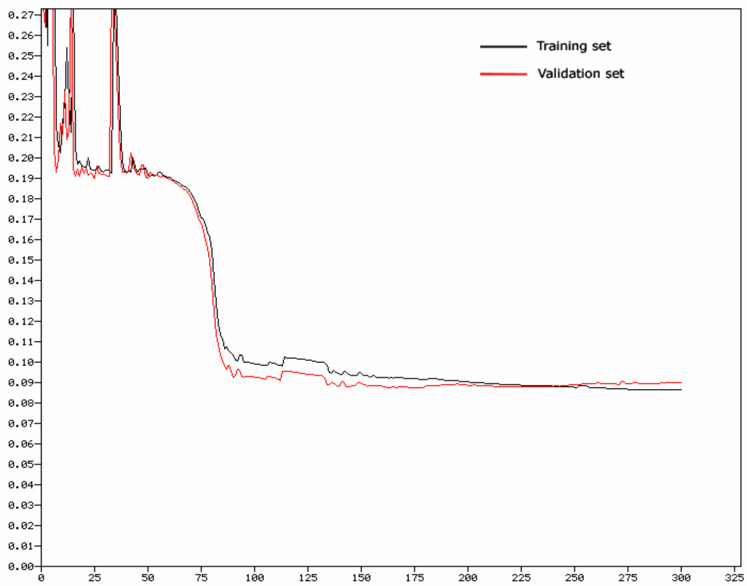


1. Ratio of total hours worked, 2002 to total hours worked, 2003; 2. Hours worked per employee, 2002; 3. Hours worked per employee, 2003; 4. Ratio of hours worked per employee, 2002 to hours worked per employee, 2003; 5. Ratio of average number of employees, 2002 to average number of employees, 2003; 6. Consistency check: takes the value 1 when the end-of-the-year number of employees, 2003 corresponds to the sum of the end-of-the-year number of employees, 2002 and the new hires during 2003 minus the layoffs during 2003, and 0 otherwise; 7. Ratio of share of overtime, 2002 to share of overtime, 2003; 8. Ratio of new hires in 2002 to average number of employees, 2002; 9. Ratio of layoffs in 2002 to average number of employees, 2002.

The learning and validation curves are shown in Figure 2. The performance of the network is detailed in Tables 1a-1d.

Figure 2

Training and validation MSE: Employment



Tables 1a-1d

EMPLOYMENT: CONFUSION MATRIX FOR THE TRAINING SET

(frequency, percent, row percent, column percent)

		PREDICTED				Total
		Correct		Wrong		
		ANN	Logistic	ANN	Logistic	
TRUE	Correct	796 53.78 100.00 84.14	791 53.45 99.37 73.72	0 0.00 0.00 0.00	5 0.34 0.63 1.23	796 53.78
	Wrong	150 10.14 21.93 15.86	282 19.05 41.23 26.28	534 36.08 78.07 100.00	402 27.16 58.77 98.77	684 46.22
	Total	946 63.92	1073 72.50	534 36.08	407 27.50	1480 100.00

Correct predictions: ANN 89.86 percent; logistic regression 80.61 percent

EMPLOYMENT: CONFUSION MATRIX FOR THE VALIDATION SET

(frequency, percent, row percent, column percent)

		PREDICTED				Total
		Correct		Wrong		
		ANN	Logistic	ANN	Logistic	
TRUE	Correct	817 55.24 98.55 85.64	822 55.58 99.16 76.32	12 0.81 1.45 2.29	7 0.47 0.84 1.74	829 56.05
	Wrong	137 9.26 21.08 14.36	255 17.24 39.23 23.68	513 34.69 78.92 97.71	395 26.71 60.77 98.26	650 43.95
	Total	954 64.5	1077 72.82	525 35.5	402 27.18	1479 100.00

Correct predictions: ANN 89.93 percent; logistic regression 82.29 percent

EMPLOYMENT: CONFUSION MATRIX FOR THE TEST SET I

(frequency, percent, row percent, column percent)

		PREDICTED				Total
		Correct		Wrong		
		ANN	Logistic	ANN	Logistic	
TRUE	Correct	1109 74.98 99.02 90.75	1107 74.85 98.84 86.48	11 0.74 0.98 4.28	13 0.88 1.16 6.53	1120 75.73
	Wrong	113 7.64 31.48 9.25	173 11.7 48.19 13.52	246 16.63 68.52 95.72	186 12.58 51.81 93.47	359 24.27
	Total	1222 82.62	1280 86.54	257 17.38	199 13.46	1479 100.00

Correct predictions: ANN 91.61 percent; logistic regression 87.43 percent

EMPLOYMENT: CONFUSION MATRIX FOR THE TEST SET II

(frequency, percent, row percent, column percent)

		PREDICTED				Total
		Correct		Wrong		
		ANN	Logistic	ANN	Logistic	
TRUE	Correct	1114 75.27 99.02 91.09	1115 75.34 99.11 86.37	11 0.74 0.98 4.28	10 0.68 0.89 5.29	1125 76.01
	Wrong	109 7.36 30.7 8.91	176 11.89 49.58 13.63	246 16.62 69.3 95.72	179 12.09 50.42 94.71	355 23.99
	Total	1223 82.64	1291 87.23	257 17.36	189 12.77	1480 100.00

Correct predictions: ANN 91.89 percent; logistic regression 85.43 percent

On the training dataset, 100% of non-erroneous records are recognized correctly; conditional on an error being present, the hit rate is of 78.07 per cent. The network shows a satisfactory performance: the logistic model, totals 99.37 per cent on correct information, but only 58.77 per cent on erroneous records. By looking at individual data, we find out that the mistakes not seen by the network are, typically, those that are neither outliers nor are able to alter the ratio structure. A perturbed value is not spotted if it falls within a reasonable range and its use in subsequent estimations does not produce anything weird, meaning that the network is as bad at understanding mistakes looking like correct information exactly as human experts may be.

The generalization shows that the results change very little when the network is fitted on the validation dataset: 99.82 per cent of the non-erroneous records and 78.92 of the erroneous ones are correctly classified. On the test sets, the hit rate is still above 99 per cent for the non-perturbed records, but falls to around 70 per cent for the ones containing mistakes. Microdata inspection shows that this behavior mostly depends on unspotted digit-swapping perturbations, whenever they do not affect the first digits on the left.

5.4 Sales

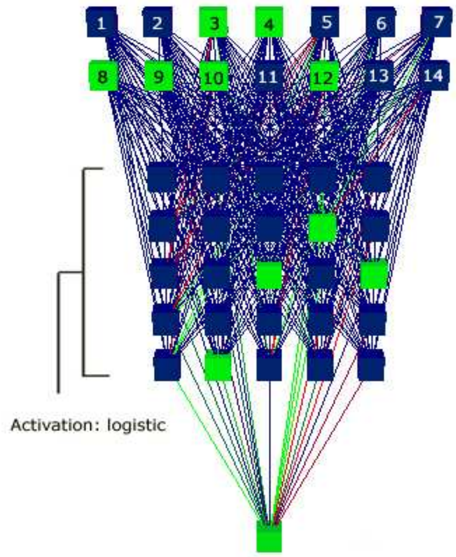
Four variables concerning the firm's overall sales are included in the simulation experiment: total sales for 2002 and 2003, together with the export sales for both years. The implemented network draws largely from the experience accumulated in setting up the one for the employment variables: transforms of the original variables are similarly included, together with the original levels that are to be checked.

The two ratios introduced are the level of total sales per employee and the quota of export sales over total sales. In this case, too, ratios are useful alongside raw variables not so much as a shield against risks of overfitting (less likely than in the employment case due to the smaller number of variables involved) but rather for the fundamental role they play in speeding up the learning and generalization performances of the network.

Dichotomous variables indicating the presence of zeroes are added: they are a sure indication of error for total sales (hardly can a firm survive without making any sales over a whole year). For export sales these indicators, together with the information about firm size and sector of economic activity, can help pinpoint suspicious zeroes. Inspection of micro data shows that the stratification variables explain a greater deal of variability for sales rather than employment: particularly for export sales, the combined information of economic activity and firm size acts as a powerful discriminant in separating "true" from "false" zeros. The topology is shown in Figure 3. Figure 4 shows the MSE evaluated on the training set and the validation set. Tables 2a – 2d present the performance of the network.

Figure 3

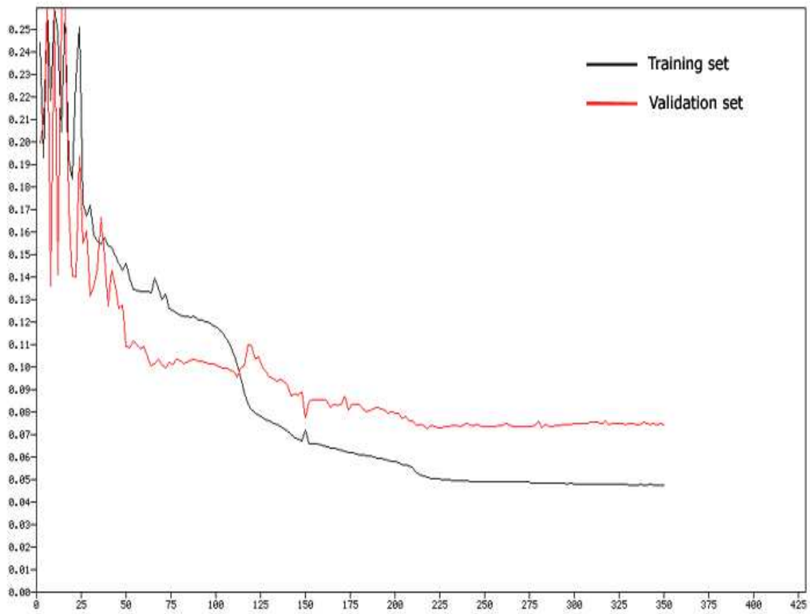
Neural network for error detection: Sales



1. Firm size class; 2. Sector of economic activity; 3. Total sales, 2002; 4. Total sales, 2003; 5. Export sales, 2002; 6. Export sales, 2003; 7. Total sales per employee, 2002; 8. Total sales per employee, 2003; 9. Export sales per employee, 2002; 10. Export sales per employee, 2003; 11. Boolean indicator taking the value 1 if total sales, 2002 = 0, and 0 otherwise; 12. Boolean indicator taking the value 1 if total sales, 2003 = 0, and 0 otherwise; 13. Boolean indicator taking the value 1 if export sales, 2002 = 0, and 0 otherwise; 14. Boolean indicator taking the value 1 if export sales, 2003 = 0, and 0 otherwise.

Figure 4

Training and validation MSE: Sales



Tables 2a-2d

SALES: CONFUSION MATRIX FOR THE TRAINING SET

(frequency, percent, row percent, column percent)

		PREDICTED				Total
		Correct		Wrong		
		ANN	Logistic	ANN	Logistic	
TRUE	**Correct**	702 47.3 96.43 94.99	621 41.85 85.3 63.37	26 1.75 3.57 3.49	107 7.21 14.7 21.23	728 49.06
	Wrong	37 2.49 4.89 5.01	359 24.19 47.49 36.63	719 48.45 95.11 96.51	397 26.75 52.51 78.77	756 50.94
	Total	739 49.8	980 66.04	745 50.2	504 33.96	1484 100.00

Correct predictions: ANN 95.75 percent; logistic regression 68.60 percent

SALES: CONFUSION MATRIX FOR THE VALIDATION SET

(frequency, percent, row percent, column percent)

		PREDICTED				Total
		Correct		Wrong		
		ANN	Logistic	ANN	Logistic	
TRUE	**Correct**	875 58.96 95.21 93.88	581 39.15 63.22 68.03	44 2.96 4.79 7.97	338 22.78 36.78 53.65	919 61.93
	Wrong	57 3.84 10.09 6.12	273 18.4 48.32 31.97	508 34.23 89.91 92.03	292 19.68 51.68 46.35	565 38.07
	Total	739 49.8	980 66.04	745 50.2	504 33.96	1484 100.00

Correct predictions: ANN 93.19 percent; logistic regression 58.83 percent

SALES: CONFUSION MATRIX FOR THE TEST SET I

(frequency, percent, row percent, column percent)

		PREDICTED				Total
		Correct		Wrong		
		ANN	Logistic	ANN	Logistic	
TRUE	**Correct**	622 41.91 84.74 80.88	638 42.99 86.92 62.24	112 7.55 15.26 15.66	96 6.47 13.08 20.92	734 49.46
	Wrong	147 9.91 19.6 19.12	387 26.08 51.6 37.76	603 40.63 80.4 84.34	363 24.46 48.4 79.08	750 50.54
	Total	769 51.82	1025 69.07	715 48.18	459 30.93	1484 100.00

Correct predictions: ANN 82.54 percent; logistic regression 67.45 percent

SALES: CONFUSION MATRIX FOR THE TEST SET II

(frequency, percent, row percent, column percent)

		PREDICTED				Total
		Correct		Wrong		
		ANN	Logistic	ANN	Logistic	
TRUE	**Correct**	616 41.51 68.44 87.62	618 41.64 68.67 68.82	284 19.14 31.56 36.36	282 19 31.33 48.12	900 60.65
	Wrong	87 5.86 14.9 12.38	280 18.87 47.95 31.18	497 33.49 85.1 63.64	304 20.49 52.05 51.88	584 39.35
	Total	703 47.37	898 60.51	781 52.63	586 39.49	1484 100.00

Correct predictions: ANN 75.00 percent; logistic regression 62.13 percent.

The training produces an almost perfect match between true and predicted error flags (95.75 of matches and 95.11 of true errors correctly identified). The same good performance is shown for the validation sample. Results are obviously less good, but still acceptable, when the network is run on the test sets. The logistic model is again outperformed.

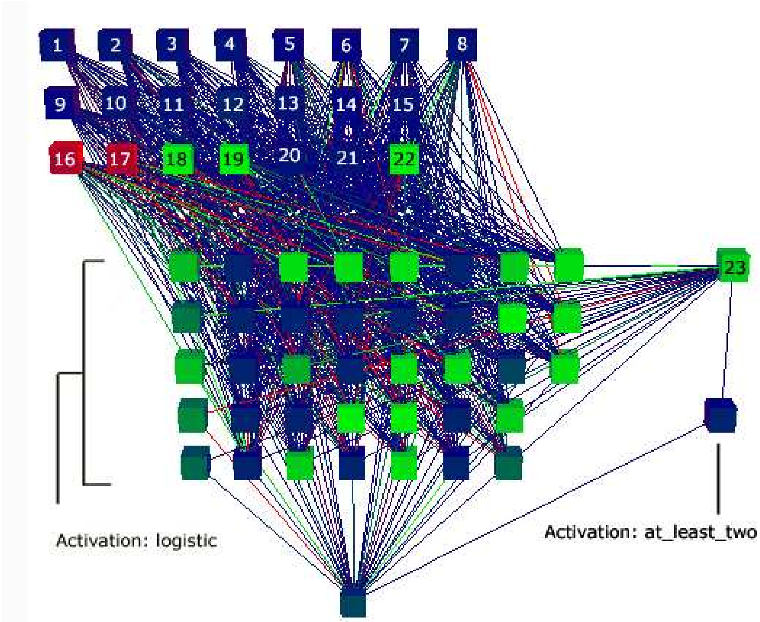
5.5 Investment

The raw survey variables concerning investment are only four: equipment, machinery and real estate in 2002; software and intellectual property in 2002; the same two variables, but referred to 2003. In order to properly train the network, however, we need more variables. Figure 5 shows the network topology and presents a short description of the 23 input variables. Those can be divided in four broad groups: a) levels of the original variables; b) ratios calculated with respect to non-investment variables (e.g. investment per employee, investment to sales); c) variations in levels and ratios; d) dummy indicators of data structure. The number of employees needed to build some of the ratios are taken from the non-perturbed dataset: in assuming they are correct, we are embracing the hierarchical approach advocated in Section 5.1.

At a first glance at the list of inputs, one can already tell that predicting errors on investment is noticeably more difficult than catching mistakes on employment or sales. The intrinsic variability of the phenomenon is much larger, and much less patterned: for example, it is possible for a firm to invest a huge amount on a given year, and very little in the following year. To make matters worse, two firms can be very similar in other respects and have substantially different investment behavior: it then becomes almost impossible to find a contrast term in order to tell which part of the variability comes from an error and which comes from real-life heterogeneity. Finally, a special problem emerges where the treatment of zeros is concerned. Some of them are actually true: it is plausible that a firm stops investing for one year. Some are false and should be substituted by some unknown amount, other ones are equally false, but should be classified instead as item non-response.

Figure 5

Neural network for error detection: Investment



1. Tangible investment, 2002; 2. Tangible investment, 2003; 3. Intangible investment, 2002; 4. Intangible investment, 2003; 5. Tangible Investment per employee, 2002; 6 Tangible Investment per employee, 2003; 7. Intangible Investment per employee, 2002; 8. Intangible Investment per employee, 2003; 9. Ratio of Tangible investment to total sales, 2002; 10.Ratio of Tangible investment to total sales, 2003; 11. Ratio of Intangible investment to total sales, 2002; 12. Ratio of Intangible investment to total sales, 2003; 13. Sum of total investment, 2002 and total investment, 2003; 14. Total investment per employee, calculated based on (13.) and average employment, 2002 and 2003; 15. Tangible investment: signed variation between 2002 and 2003; 16. Intangible investment: signed variation between 2002 and 2003; 17. Tangible investment: signed relative variation between 2002 and 2003; 18. Intangible investment: signed relative variation between 2002 and 2003; 19. Boolean indicator taking the value 1 if Tangible investment, 2002 = 0, and 0 otherwise; 20. Boolean indicator taking the value 1 if tangible investment, 2003 = 0, and 0 otherwise; 21. Boolean indicator taking the value 1 if intangible investment, 2002 = 0, and 0 otherwise; 22. Boolean indicator taking the value 1 if intangible investment, 2003 = 0, and 0 otherwise; 23. Sum of the four boolean indicators sub (19.), (20.), (21.), and (22.).

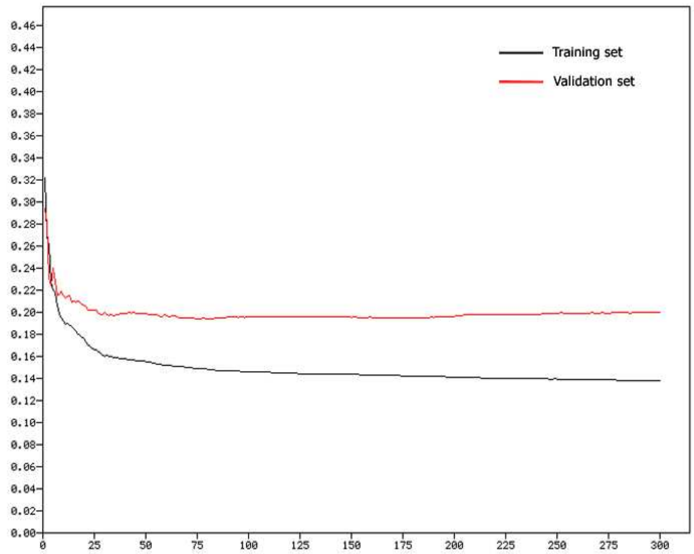
This scenario implies that the approach used in the case of employment, i.e. a straightforward reduction of the problem dimensionality by moulding it into a combination of several simple range problems and consistency checks, cannot work for investment. The concept of acceptable range is less clearly defined: if a structure similar to the one *sub* 5.3 were proposed, only the few major outliers would be caught. Prior knowledge of a different kind is needed by the network. Through trial and error we find that the level, ratio and variation variables are all useful to fine-tune the performance of the algorithm, but the big jump in performance terms is achieved by introducing a set of four dummy variables that detail whether each of the four raw variables is zero-valued.

These indicators put the network on the right track in two ways. First, while investment can be equal to zero in one year and in one aggregation (either machinery or software), the more zeros we find, the more suspicious the record looks: it is very unlikely for a firm to refrain totally from any investment for two consecutive years. Second, firms that invest for two years in a row on a given item might show definite patterns, different from those of firms that only choose to invest during one of the two years. For example, suppose that the former do not invest less than the latter, on average: this will be reflected in the variance of the distribution of investment-to-sales and investment-per-employee ratios over the two years. If the learning process is helped by inserting a simple Boolean indicator that signals whether a firm reportedly invested or not during each year, it will tell true zeros from false zeros more easily: the network may deduce a requirement for consistency of this indicator with the patterns seen on related variables. For similar reasons, we also added variables that contain the sum of investments in all fields over the two years.

Figure 6 shows the MSE evaluated on the training and on the validation set and Tables 3a-3d present the results.

Figure 6

Training and validation MSE: investment



Tables 3a-3d

INVESTMENT: CONFUSION MATRIX FOR THE TRAINING SET

(frequency, percent, row percent, column percent)

		PREDICTED				Total
		Correct		Wrong		
		ANN	Logistic	ANN	Logistic	
TRUE	Correct	631 42.52 85.04 76.76	644 43.4 86.79 64.02	111 7.48 14.96 16.77	98 6.6 13.21 20.5	742 50
	Wrong	191 12.87 25.74 23.24	362 24.39 48.79 35.98	551 37.13 74.26 83.23	380 25.61 51.21 79.5	742 50
	Total	822 55.39	1006 67.79	662 44.61	478 32.21	1484 100.00

Correct predictions: ANN 79.65 percent; logistic regression 69.01 percent

INVESTMENT: CONFUSION MATRIX FOR THE VALIDATION SET

(frequency, percent, row percent, column percent)

		PREDICTED				Total
		Correct		Wrong		
		ANN	Logistic	ANN	Logistic	
TRUE	Correct	615 41.61 74.73 75	643 43.5 78.13 66.84	208 14.07 25.27 31.61	180 12.18 21.87 34.88	823 55.68
	Wrong	205 13.87 31.3 25	319 21.58 48.7 33.16	450 30.45 68.7 68.39	336 22.73 51.3 65.12	655 44.32
	Total	820 55.48	962 65.09	658 44.52	516 34.91	1478 100.00

Correct predictions: ANN 72.06 percent; logistic regression 66.23 percent

INVESTMENT: CONFUSION MATRIX FOR THE TEST SET I

(frequency, percent, row percent, column percent)

		PREDICTED				Total
		Correct		Wrong		
		ANN	Logistic	ANN	Logistic	
TRUE	Correct	874 58.89 75.93 84.85	954 64.29 82.88 79.17	277 18.67 24.07 61.01	197 13.27 17.12 70.61	1151 77.56
	Wrong	156 10.51 46.85 15.15	251 16.91 75.38 20.83	177 11.93 53.15 38.99	82 5.53 24.62 29.39	333 22.44
	Total	1030 69.41	1205 81.2	454 30.59	279 18.8	1484 100.00

Correct predictions: ANN 70.82 percent; logistic regression 69.82 percent

INVESTMENT: CONFUSION MATRIX FOR THE TEST SET II

(frequency, percent, row percent, column percent)

		PREDICTED				Total
		Correct		Wrong		
		ANN	Logistic	ANN	Logistic	
TRUE	Correct	854 57.78 74.85 86	911 61.64 79.84 78.40	287 19.42 25.15 59.18	230 15.56 20.16 72.78	1141 77.2
	Wrong	139 9.4 41.25 14	251 16.98 74.48 21.60	198 13.4 58.75 40.82	86 5.82 25.52 27.22	337 22.8
	Total	993 67.19	1162 78.62	485 32.81	316 21.38	1478 100.00

Correct predictions: ANN 71.18 percent; logistic regression 67.43 percent

On the training set, 79.65 percent of all records are correctly classified; the hit rate is of 72.06 per cent on the validation set, and respectively 70.82 and 71.18 percent on the two test sets. The correct predictions, conditional on no mistakes, range between 75 and 85 per cent depending on the dataset, both for the network and the competing logistic model. On the other hand, the network again outperforms the logistic model on the very task of error-spotting (70 per cent of the wrong records, against 50 per cent for the logistic model). On the test sets, the network shows a relatively poorer performance, as anticipated[7]; it only catches, respectively, 53.15 and 58.75 percent of the errors, still better than the logistic predictor (25 per cent in both cases).

Predictions formulated on investment are on the whole less certain than those referring to other phenomena. The average distance between target and forecast is considerably larger than in the case of employment or sales, as Figure 4 clearly shows, but the hit rate is not dramatically smaller: since all predictions smaller than 0.5 are mapped to zero and the rest to one, the hit rate is only partly affected by the fact that investment predictions are closer to the center of the (0,1) interval than the other predictions. This can be clearly seen in the kernel approximations shown in section 5.6 below.

5.6 An interpretation of the results in terms of density estimation

All the results are produced according to the generally accepted convention of translating numbers below 0.5 as a prediction of correctness (value 0), so that records with a network output like 0.45 are going to be evaluated as correct: if we estimate a kernel density function of the predictions respectively of absence and presence of errors (Figures 7 and 8), to account for the shape of the empirical distribution function, we get a clearer understanding of the functioning of the networks. The excellent performance in error identification of sales is shown by the density peaks respectively located very close to zero and one. Slightly less good, but equally satisfying, is the capability of the network for employment, which presents a negligible uncertainty in correctly identifying wrong values (as shown by the hump on the left of Figure 8). The estimated distributions confirm that investment predictions are more concentrated towards the center of the interval (0,1): for investment, the network is apparently more cautious, whereas for employment and sales is more clear-cut in assessing the probability of a record containing mistakes.

---

[7] As explained in Section 5.2, in the test phase we re-introduced perturbations such as swap of neighboring digits, or substitution of a randomly chosen digit with a randomly selected one: we expect this to lower the hit rate, acting as a sort of “stress test”.

Figure 7

Distribution of network predictions,
conditional on absence of errors (0's)
(kernel density estimates, rescaled)

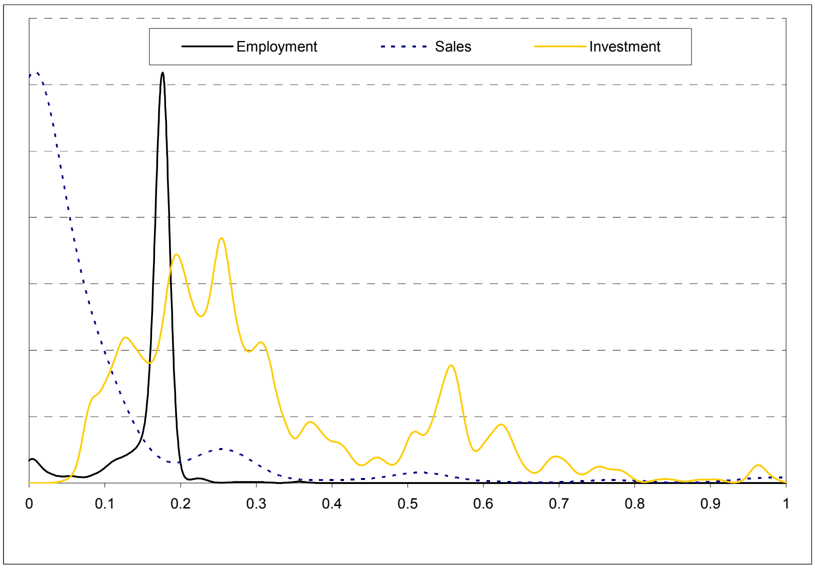
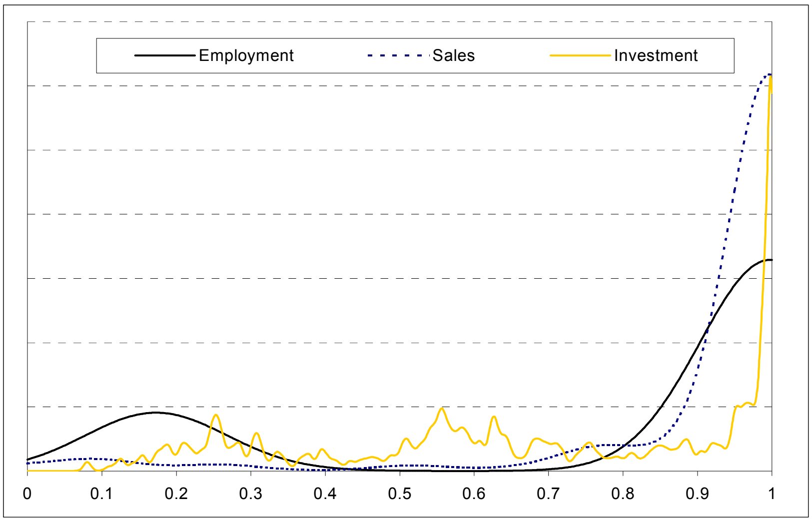


Figure 8

Distribution of network predictions,
conditional on presence of errors (1's)
(kernel density estimates, rescaled)



6. Conclusions and further developments

An application of neural network models to predictive classification for data quality control is presented. Three feed-forward networks (for employment, sales and investment) are trained on a set of records from the most recent wave of the Bank of Italy's business surveys, in order to identify and correct measurement error. The output variable is a binary proxy of unobservable measurement error; the input variables include raw survey variables and many transforms thereof. The networks are trained on simulated datasets, based on perturbations of the original survey dataset intended to reproduce real types of measurement error.

Common mistakes can be recognized by a properly trained neural network with a satisfying level of accuracy. The percentage of correct predictions on training datasets is of 89.86 percent for employment, 95.75 percent for sales and 79.65 percent for investment. Validation sets yield similar results. Out-of-sample generalizations are also good, albeit the test sets includes types of errors not used in the learning phase, which just slightly lower the hit rates conditional on error presence. The networks consistently outperform the logistic predictor, which is the most common alternative classification method.

The performance shown by the networks turns out to depend crucially on how prior knowledge is plugged in the architecture, and on the choice of a learning algorithm. As far as the first issue is concerned, transforms are introduced to reduce the dimensionality of the learning task: consistency problems are therefore turned into simpler range problems. Other improvements derive by simple discrete indicators of data structure that act as simple blueprints that the network broadly follows in building acceptance and rejection regions. As for the learning algorithm, the resilient propagation (Rprop) delivers results largely superior to those of the standard backpropagation (see Appendix). This probably depends on the adaptive nature of the former, which fine-tunes the learning parameter based on the data structure: a particularly useful feature in complex datasets.

In general, errors on investments appear harder to spot than those on employment and sales: a larger proportion of network predictions is actually incorrect and, when correct, less certain. This can be explained on the basis of the limited stability of investment: records with correct and wrong values can look too similar for the network to discriminate.

The effectiveness shown by the networks in flagging erroneous groups of variables suggests a twofold direction for further research. At the macro level, we could try to streamline the existing data editing process. As seen in Section 2, traditional selective editing procedures suffer from a low "hit rate", which hovers around a fifth of the flagged records.

The currently used data editing method split the records in two groups: the thoroughly edited records, the so-called "critical stream" (the most influential units); and the loosely edited records, the "non-critical stream" (the less influential units). On the "critical stream", the integration of network flagging with traditional (score-based) flagging can be experimented.

On the "non-critical stream", the use of ANN could become an effective editing tool in data quality assessment for micro level analysis. Neural network error predictions are most precious here: traditional microdata quality assessment is a very complex task, largely based on subject-matter expert advice, whereas the ANN learning process guarantees that units are compared to each other by exploiting known regularities found in units flagged as correct. The flexible learning mechanism enables to easily insert in the model any useful new features of error-free units.

Appendix: Methodological issues

Basics on Neural Networks

Artificial neural networks (ANNs) have raised growing interest in the last dozen of years also out of their native field of application, which traditionally was discriminant analysis and pattern recognition. The name recalls the very activity of the human brain, structured in collection of information, learning by training and subsequent validation; it carries a suggestive flavor and may possibly raise misplaced enthusiasm for a new promising tool.

Research in ANNs dates back to the studies of McCulloch and Pitts (1943) but gained momentum only starting from the Sixties, as growingly available computing power made possible the widespread implementation of complex algorithms. The seminal idea of a linear combination of observed values to separate (classify) units is found in the work of Rosenblatt (1962), The re-emergence of ANNs in the Eighties is marked by the textbook by Rumelhart and McClelland (1986).

From an econometric point of view, ANNs can definitely be viewed as powerful prediction tools with strong connections with the classical framework of multivariate regression models. Systematization of ANN theory under the category of statistical computing started with the classical paper of Cheng and Titterington (1994); a discussion which punctually parallels ANNs and non-linear regression tools, comparing also the respective jargon, can be found in Martin and Tan (1997).

Neural networks can be fully represented by means of partially connected graphs. Units, or nodes, are called "neurons" in the ANN parlance; nodes and arches are organized in layers. The first layer contains inputs, i.e. right-hand side variables; the last final layer contains outputs, i.e. left-hand side variables. Layers between the initial and the final ones are called "hidden" layers, as their values are fed from within the network and do not represent observed variables.

In a feed-forward network, information flows from input to output only, with no arches reverting to a node belonging to a preceding layer. This is perfectly intuitive as we maintain constant in mind the comparison with regressions models, to which feed-forward ANNs can be reconducted. Impulses (=values) coming from input and intermediate layers are magnified or inhibited through appropriate weights and then fed (=weighted sum) to the subsequent layer. Prior to that, the signals are processed through a function, called a "squasher" or activation, that calibrates the sensitivity of the nodes to the stimuli. Activation functions are typically sigmoidal (e.g. logistic or cumulative normal) or threshold.

"Learning" means that the output of the network is compared to desired output by means of a loss function. If the value of the loss function is not minimum, an appropriate rule (typically the gradient method) adjusts the weights, whose values are propagated back through the network. This way of iterating is not unique, but typical, and called backpropagation. The learning process ends when the distance between desired and predicted output is minimum, i.e. global loss is minimum.

If the desired value of the output is compared to the network for learning (=fitting) we speak of "supervised" learning. This is so far the only type of ANNs that has proven of use for concrete applications, as unsupervised learning poses expectedly formidable methodological problems.

When it comes to predict a zero-one classifying variable, as in our case, the logit modeling framework appears a natural competitor. In fact, there is perfect coincidence between a particular class of artificial neural networks, i.e. two-layer, feed-forward networks with a single logistic activation function (at the output node), and multinomial logit (Ripley 1996); the enhancement offered by neural networks equipped with hidden layers belongs in the fact that they can natively account for nonlinearities in the hypothesized relationship (Bentz and Merunka 2000).

Basics on the applied architecture

A neural network is fully defined once we have chosen the following features: a) the network topology; b) the input transfer function; c) the weight adjustment rule; d) the output transformation function. What follows is in common with the three networks we trained.

a) We chose the classical feed-forward network topology. A feed-forward network has vertices which can be numbered so that all connections go from each vertex to another with a higher number. Nodes are organized in layers with connections available only from lower-number to higher-number layers. In practice, signals flow in one direction only (from input to output). This recalls the familiar feature of ordinary regression models, where we feed values in right-hand variables and get the resulting output from the left-hand side.

Given the universal approximation property shown, for example, in Hornik, Stinchcombe and White (1989), one hidden layer is enough for our aims. Such a network can be represented by the expression (Ripley, 1996)

$$y_k = f_k\left(\alpha_k + \sum_{j=1}^{k} w_{jk} f_j\left(\alpha_j + \sum_{i=1}^{j} w_{ij} f_i(x_i)\right)\right)$$

explicitly, a (possibly) non-linear regression function. The $f(\cdot)$'s are called the activation functions; w_{ij} are the weights assigned to each node and account for the approximation of the y_k through the network. As seen in the second paragraph, a network with linear input and a single logistic output can be seen as an extension of the logistic regression; they coincide if we get rid of the hidden layer.

b) The hidden layer activation function has been chosen as logistic, as this type of activation has proven able to learn and converge on our data.

c) We chose Resilient backpropagation (Rprop) (Riedmiller and Braun 1993) as the rule for weight updating. This represents a major improvement to traditional backpropagation algorithm.

The basics of vanilla backpropagation can be briefly expounded as follows. If we have "examples" (**x**, **t**) (**t** fitting **y**) and the output of the network is ***y***=*f*(**x**;**w**) the parameter vector **w** should minimize *d*(**t**,**y**), where *d* is an average distance function. If this distance is convex, e.g. a sum of squares, the "steepest descent" method can be used for updating the weights: the rule is of the type

$$w^{(n+1)} = w^{(n)} - \eta \frac{\partial d}{\partial w}\Big|_{w=w^{(n)}} + \zeta(w^{(n+1)} - w^{(n)})$$

Where η marks the rate of adjustment $\eta \leq 1$ and is called the "learning rate". ζ, called "momentum", is an optional term which allows additional flexibility in updating during the iterations. This procedure is in fact a least square fit achieved through an iterative algorithm (due to the non-linear structure of the model).

Backpropagation, being basically a gradient convergence method bound by the choice of a fixed "learning rate" and making use of the first derivative only, can be rather slow, as seen in the literature. Moreover, it is rather likely to stuck into paralysis (if changes in the weights become too small) or to get trapped into local minima.

A remedy to these drawbacks seems to have been found in Rprop. Here the updates are no longer proportional to the partial derivative, as they use an independent step size for every connection. The direction is defined only by the sign of the partial derivative. The formulae for Rprop are intuitive but a bit bulky: the learning rule for the update-values Δ^t_{ij} is

$$\Delta^t_{ij} = \begin{Bmatrix} \eta^+ \Delta^{t-1}_{ij}, \dfrac{\partial E^{t-1}}{\partial w_{ij}} \cdot \dfrac{\partial E^t}{\partial w_{ij}} > 0 \\ \eta^- \Delta^{t-1}_{ij}, \dfrac{\partial E^{t-1}}{\partial w_{ij}} \cdot \dfrac{\partial E^t}{\partial w_{ij}} < 0 \\ \Delta^{t-1}_{ij} \, otherwise \\ \end{Bmatrix}$$

and, consequently, the learning rule for the weights is described by $w_{ij}^{t+1} = w_{ij}^t + \Delta w_{ij}^t$, where

$$\Delta w^t_{ij} = \begin{Bmatrix} -\Delta^t_{ij}, & \dfrac{\partial E^t}{\partial w_{ij}} > 0 \\ +\Delta^t_{ij}, & \dfrac{\partial E^t}{\partial w_{ij}} < 0 \\ 0 & otherwise \\ \end{Bmatrix}$$

Whenever a change of sign occurs for the partial derivative of a weight, (signaling that the last update was too big and the procedure got trapped in a local minimum) the update-value is decreased by a fixed factor. If the derivative retains its sign instead, the update-value is increased to speed convergence. Once the update value has been adapted, the weights are updated in their turn by the intuitive rule: if the derivative is positive, the weight is decreased, and vice-versa.

The software

The experiments were conducted by means of JavaNNS software package (Java Neural Networks Simulator, version 1.1) under a PC Windows NT (ver. 4.0) architecture. For most computationally intensive tasks, the UNIX version of SNNS (Stuttgart Neural Network Simulator) package has been run on a RISC system.

JavaNNS is a simulator for neural networks developed at the Wilhelm-Schickard-Institute for Computer Science (WSI) in Tübingen, Germany. It is based on the SNNS 4.2 kernel, with a new graphical user interface written in Java. Currently, JavaNNS is distributed by the University of Tübingen only as a binary file. It is not public domain, but is available free of charge. All relevant information on this package can be found at the Web address:

http://www-ra.informatik.uni-tuebingen.de/software/JavaNNS/welcome_e.html

References

Barcaroli, G., D'Aurizio, L. (1997), Evaluating editing procedures: the simulation approach, *Working Paper, Conference of European Statisticians, Work Session on Statistical Data Editing,* Prague.

Battipaglia, P. (2002), Selective editing to increase efficiency in survey data processing – An application to the Bank of Italy's Business Survey on Industrial Firms, *Irving Fisher Committee Bulletin,* n. 13, December.

Bentz, Y. and D. Merunka (2000), Neural networks and the multinomial logit for brand choice modelling: a hybrid approach, *Journal of Forecasting*, 19, 177-200.

Biancotti, C. and R. Tartaglia-Polcini (2005), Artificial Neural Networks for Data Editing, *Irving Fisher Committee Bulletin,* n. 21*,* July.

Breiman, L. (1994), Comment of Neural networks: a review from a statistical perspective, by Cheng, B. And Titterington, M., *Statistical Science*, vol. 9 n. 1, 2-54.

Breiman, L. (2001), Statistical modeling: the two cultures, *Statistical Science*, vol. 16 n. 3, 199-231.

Brackstone, G. (1999), Managing data quality in a statistical agency, *Survey Methodology*, vol. 25 n. 2, 139-149.

Cheng, B. and D. M. Titterington (1994), Neural networks: a review from a statistical perspective (with discussion), *Statistical Science*, 9, 2-54.

Chinnappa, N. *et al.* (1990), "Macro editing at Statistics Canada", unpublished report of the Statistics Canada working group on strategies for macro editing, prepared for the Statistics Canada Advisory Committee on statistical methods (January), Ottawa: Statistics Canada.

Cooper, J.C.B. (1999), Artificial neural networks versus multivariate statistics: an application from economics, *Journal of Applied Statistics*, vol. 26 n. 8, 909-921.

Graepel, T., R. Herbrich *et al.* (2001), Neural networks in economics: background, applications and new developments. Department of computer science, Technical University of Berlin.

Granquist, L. and J.G. Kovar (1997), "Editing of survey data: how much is enough?" in: Survey measurement and process quality, edited by Lyberg, Biemer *et al.* New York: Wiley.

Hedlin, D. (2003), Score functions to reduce Business Survey Editing at the U.K. Office for National Statistics, Journal of Official Statistics, Vol. 19 n. 2, pp. 177-199.

Hornik, K., M. Stinchcombe and H. White (1989), Multi-layer feedforward networks as universal approximators, *Neural Networks*, 2, 359-366.

Larsen, B. and B. Madsen (1999), Error identification and imputations with neural networks, UN/ECE Work Session on Statistical Data Editing, Working Paper 26.

Martin, V. L. and C. Tan (1997), Artificial neural networks, in: Creedy, J. and V. L. Martin (eds.), *Nonlinear economic models*, Cheltenham: Edward Elgar.

Rohwer, R, M. Wynne-Jones and F. Wysotzki (1995), Neural Networks. In Michie, D., D.J. Spiegelhalter, C.C. Taylor (editors), Machine learning, neural and statistical classification. Hertfordshire: Ellis Horwood.

Nordbotten, S. (1995), Editing statistical records by neural networks, *Journal of Official Statistics* Vol. 11 n. 4, 391-411.

Nordbotten, S. (1996), Editing and imputation by means of neural networks, *Statistical Journal of the United Nations Economic Commission for Europe*, 119-129.

Riedmiller, M. and Braun, H. (1993), "A Direct Adaptive Method for Faster Backpropagation Learning: The RPROP Algorithm", Proceedings of the IEEE International Conference on Neural Networks 1993, San Francisco: IEEE.

Ripley, B.D. (1996), Pattern recognition and neural networks. Cambridge: Cambridge University Press.

Rivière, P. (2002), General data editing tools are often unsuitable to use in complex business surveys: why?, Conference of European Statisticians, UNECE Work Session on Data Editing, Helsinki, Working Paper n. 30.

Roddick, H. (1993), Data editing using neural network. Tech. Rep., Systems Development Division, Statistics Canada.

Rosenblatt, F. (1962), Principles of Neurodynamics. Washington: Spartan books.

Rumelhart, D.E. and J.L. MacClelland (1986), Parallel distributed processing, Cambridge: MIT Press.

Thompson, K.J. and R.S. Sigman (1999), Stastical Methods for developing ratio edit tolerances for economic data. Journal of Official Statistics, 15, 517-535.