December 1995

Treatment of Data in the Development of Intelligent Information Systems Supporting Financial Decisions

Albert Chan
Griffith University

Richard Bonner
Griffith University

TREATMENT OF DATA IN THE DEVELOPMENT OF INTELLIGENT INFORMATION SYSTEMS SUPPORTING FINANCIAL DECISIONS

Albert C. K. Chan and Richard F. Bonner

School of Information Systems & Management Science
Faculty of Commerce and Administration, Griffith University
Nathan-Brisbane, Australia 4111

Solving the unstructured financial problems belongs to the "knowledge-rich" task domain, collectively requiring massive current and historical data. The problem is here discussed from the standpoint of a general strategy for the deployment of intelligent information systems, concentrating on the single most critical but manageable issue: the choice and treatment of data in their development. The training in artificial neural networks is formally the construction of a "problem → solution" schema from a set of examples, and the acquired schemas depend in practice significantly on the form of the data. The suggested preparation of data consists of four basic steps: the collection, analysis, preprocessing, and the separation of data into training and testing sets. The data preparation process is illustrated by the case of a currently developed intelligent system for exchange rate risk management. It is re-emphasised in conclusions that data preparation is only one, albeit vital, aspect which makes the development of intelligent information systems quite different from that of conventional information systems.

1 Introduction

Ever since the days of Eniac, computers have been used as problem solving tools. By noting the common thread through the historical trend of computer applications from supporting structured problems to tackling unstructured problems, the knowledge of the programmer/expert becomes encoded in the program in the form of a typically fix input-output mapping, the "system transfer function" which operates on "data" transforming it into some "more useful" form. The "intelligence" of such white-box systems is thus merely a reflection of the knowledge of their creators. By contrast, the introduction of "artificial neural networks" into business information systems, as it turns out predominantly in the area of financial management (Jagielska, 1993; Trippe & Turban, 1993), in principle short-circuits the traditional role of programmers. Such systems learn by experience (Zurada, 1992): the transfer function (problem → solution) of the system is not *a priori* determined by the programmers but gradually develops as a result of repetitive exposure to data - naturally, in accordance with some in-programmed scheme. The resulting system is a black-box: its transfer function can be observed but is not easily "explained": the "rule extraction" from neural networks is a known problem.

Seen against this background, the data-rich but fairly well understood area of financial management should be ideal for neural network applications. Financial theory is conceptually advanced with all its variables readily measurable, and it should be straightforward to set up a neural network following some general financial model. Able to handle massive data, one would think, a neural network should be an ideal extension of economic and financial models, which provide structure and insight but are in themselves of limited predictive power (Meese & Rogoff, 1983). Indeed, among existing commercial applications of neural network technology, financial applications clearly dominate (Jagielska, 1993). Upon a closer examination, however, and compared to the scope and sophistication of neural network applications in technical areas such as speech (Sejnowski & Rosenberg, 1986) or image (LeCun, *et al*, 1989) recognition, the typical financial neural network applications rank rather low (Taylor, 1993), and their authors have to work hard to justify their choice of neural network technology from dozens of other, usually better understood and theoretically just as suitable, approximation schemes.

This observation, readily verifiable in isolated cases, but, of course, in need of research to give it statistical validity, is sufficiently alarming to make one consider its probable causes and potential consequences. We do this in Section 3, summarising the principles of neural network technology and discussing some of its policy issues. The overlaying general problem could be summarised as the lack of a basic policy on the development, use, and management of intelligent information systems in socio-economic context. To allow a non-technical yet meaningful discussion, we previously build up some cognitive background in Section 2, recalling simple but insightful analogies between human and machine learning. The discussion quickly focuses on a single and directly addressable problem: the treatment of data in the development of intelligent information systems. This problem is expanded in the main Section 4 outlining a scheme of data preparation for neural networks and heuristically discussing its four main steps: the collection, analysis, preprocessing, and the separation of data into training and testing sets. Finally, in Section 5, a current research project (Chan & Bonner, 1995) on intelligent systems in exchange rate risk management is used as illustration.

2 Problem schemas for knowledge-rich task domains

We start by recalling some cognitive concepts of significance for both human and machine problem solving. In particular, we hope to clarify the role of "data" in solving problems, and to prepare ground for Section 3 viewing artificial neural networks as problem schemas.

In essence, "problem solving" is the process of transforming the "data" defining a "problem" into "data" recognised as its "solution", usually by successively applying a series of simple transformations (VanLehn, 1989). According to Webster's dictionary (Mish, 1989), *datum* (pl. *data*), from Latin, literally *gift* or *present*, is "something given or admitted esp. as a basis for reasoning or inference". Thus, etymologically speaking, being "data" is not a property of any thing in isolation but a property of a (thing, transformation scheme) pair.

In one of the task domains in problem solving, the "knowledge-lean" domain, extensively studied by Newell and Simon in their classic (Newell & Simon, 1972), the problem solving transformations are fairly structured and essentially independent of the data. Thus, only little knowledge is required to solve the bulk of problems in such a domain once the transformation techniques have been mastered. In practice, however, problems are usually "contextual", each requiring its own specific transformations. Consequently, although the knowledge required to solve a single practical problem may not be significant, to keep on solving them requires copious amounts of current and historical data. Such practical problems, financial management problems included, belong to a "knowledge-rich" task domain.

It is interesting to recall some known facts explaining how "experts" solve problems in a knowledge-rich task domain (VanLehn, 1989). Roughly, experts recognise a problem as an instance of a familiar problem type, retrieve a solution template from their memory, and generate the problem's solution; they do not, in general, move about searching for a solution. Their behaviour is a product of knowledge gained through learning. As shown by the chess move experiment (Charness, 1981), for example, expertise lies not in having a more powerful overall strategy in problem solving, but rather in having a better knowledge for making elementary decisions.

The template or knowledge representation of a problem and its corresponding solution in an expert's memory is known as a problem schema (VanLehn, 1989); see also Arbib & Hesse (1986) where a theory of schemas is discussed in a broad cognitive and philosophical context. Simply and formally, a problem schema can be thought of as the "problem $\rightarrow$ solution" mapping. It consists of a class of problems the schema applies to together with the corresponding solutions. For example, in the following simple schema:

(a1) *there is a tendency for the currency to move down in the next three months,*
(a2) *the net cash outflow is high,*
(b) *so the cash flow should be hedged,*

the statements (a1) and (a2) describe the problems and the statement (b) describes a solution. Experts would have acquired many large and refined pieces of problem schemas. Each piece is highly specialised and is therefore effective in solving a particular class of problems (Chase & Ericsson, 1981).

Humans and machines alike (Poggio & Girosi, 1990) acquire problem schemas through learning, ie, through "resilient changes in the subject's knowledge about a task domain that is potentially useful in solving further problems" (Simon, 1983). Over time, problem schemas are formed by exposure to collection of problem-solution pairs. Of course, in contrast with simple supervised learning (Zurada, 1992), in more complex forms of learning, the association of a problem with its solution may not always be provided in examples, requiring instead an involved search process which may produce, along the way, new schemas or even re-define the initial problem. Whatever the particulars of the training process, after successful training, when presented with something "close" to the input of a learned pair (the problem), the learner can recall something "close" to the corresponding output (the solution).

We summarise all this on a lighter note with three "facts of life":

(i) **Data is all there is.**
In term of formal modelling of cognition, data is the basic element and all else is derived in terms of its "patterns" or "properties" (Bonner & Chan, 1995). In knowledge-lean domains, patterns are abundant giving rise to a rich transformation structure organising the "raw" data. In knowledge-rich domains, there is little pattern and more data needs to be retained in "raw", problem-specific, form.

(ii) **Learning always takes time but learning from poor data takes very long time.**
That the statement holds for human learning is a matter of direct experience: it takes a lifetime to become an "expert" even though one builds on transferred knowledge of previous generations. Machine learning is no different: the underlying physical process may be speeded up to a degree, but the task of organising experience into problem schemas is identical for man and machine.

(iii) **"Natural" and "artificial" are essentially the same.**
The distinction between "natural" and "artificial" (intelligence, learning, life (Hogeweg, 1993), ...) is simply not practical as long as both phenomena are indistinguishable in any operational (behavioural) sense. As quoted in the paper by Bonner (1993), "what does it matter whether one thinks with jelly or with wires?" The issue is no longer part of science fiction, as intelligent systems are becoming integrated with human decision making. For example, legal problems in trying to maintain a separation between the two by mixing the behavioural and the cognitive, are very real and urgently require solutions.

3 Artificial neural networks: problem schemas of a data-driven technology

Artificial neural networks are essentially as old as the von Neuman computer, and have, over the years, acquired academic standing as an acknowledged field of research. Commercial applications of neural networks are, on the other hand relatively recent but fast growing in number. In economic analysis and forecasting, there are applications to bankruptcy prediction (Koster, *et al*, 1990), stock market prediction (Wan & Chan, 1993; Baba & Kosaki, 1992), macroeconomic evaluation (Li, *et al*, 1991), and exchange rate forecasting (Refenes, 1993); and, in financial risk management, the neural network applications include bond rating (Dutta & Shekhan, 1988), mortgage underwriting (Collins, *et al*, 1988), and hedging strategy decision in foreign exchange market (Chan & Bonner, 1995). We have in this paper opted for calling information systems with neural network modules "intelligent", noting, however, that the words "adaptive" or "learning" could have been possibly more accurate but less established choices.

An artificial neural network is a "neurally inspired" (Rumelhart, 1989) computational scheme with "architecture" conceptually similar to that of its biological counterpart. Briefly, the "input signals" from the external environment are transformed into the "output signals" by the "network transfer function". Just like people, before neural networks can be entrusted with practical decisions, they must be put through a comprehensive training process, in which the parametric weights to the input signals are adjusted iteratively until a certain degree of accuracy of the output has been reached. After training, the values of the weights determine the "network transfer function" which is the problem schema to be used in life problem solving.

3.1 Promises and limitations, revisited

As pointed out in Section 2, the "intelligent" behaviour can be encoded in form of a transfer function of the "system" in question. Artificial neural networks can adapt its transfer function to the data of the environment in an "autonomous" way (Serra & Zanarini, 1990) and can thus exhibit "adaptive" or "intelligent" behaviour. Thus, so far and in principle, one could conclude that neural network technology should have no bounds in self-education, eventually perhaps surpassing all human intelligence.

In the application area here discussed, however, it is hard to find a neural system justifying the use of this technology above other, better known and often more effective computing schemes. Indeed, if restricted to the "feed-forward" networks trained by supervision, the two basic potential advantages of neural networks over classical function approximation schemes, which are the parallel structure and the speed of recall, are rarely utilised at all in existing financial applications. At the same time, the obvious disadvantage of the technology, its low explanatory value, does not seem to bother anyone.

We know now that "raw crunching power" of the computer running some universal algorithm for a giant neural network is not a realistic solution to the problem of learning from massive unorganised data. Recent hardware implementations of neural network learning algorithms, possible analogue computing, and whatever else the immediate future may bring, will of course be a continued improvement, but none of these could be a panacea solution for the problems of combinatorial explosion in dealing with unorganised data. The "contextual approach" is likely to remain for some time the only way out: problems need to be carefully analysed, data carefully selected, architectures and algorithms carefully adapted to the problems. There seem to be no universal short-cuts.

3.2 Policy issues

There is little doubt that the technical disciplines will continue their spectacular achievements in computing technologies, and that, consequently, computer-based information systems will continue to grow in their ability to take on ever more sophisticated intellectual tasks. What is required on the most general level is the establishment of a basic policy on the development, use, and management of intelligent information systems in the various socio-economic domains. Such policy will of course take time to develop as it must be grounded on a thorough understanding of basic issues. We suggest that perhaps sufficient effort is not being put into this development.

To exemplify, among basic issues which such a policy would need to address are the future intellectual needs of information systems developers, users, and managers; and the problem of adapting system development methodologies from "traditional" to intelligent systems. This is, in fact, the underlying problem we are considering in this paper, looking at the basic difference between the development of the two kinds of systems in terms of the required treatment of data.

4 Preparation of data for neural networks

It is assumed that the data preparation process here in question has been preceded by a thorough "system analysis" in the usual sense, resulting in some form of formal "conceptual schema" (Batini, *et al*, 1992), representing the "system" (organisation, firm, process, ...) considered. Such a schema is, as usual, the blueprint for the design of an information system, including its "intelligent" neural network modules. For those modules, however, a schema is only a starting point; their practical performance will essentially depend on the actual data in the schema. Such data will be used to determine the final architecture of the neural networks, and to train the networks. The data needs now to be interactively collected and tested, possibly leading to modifications in the original system schema.

The following general scheme for preparation of data for a neural network seems reasonable (Figure 1).

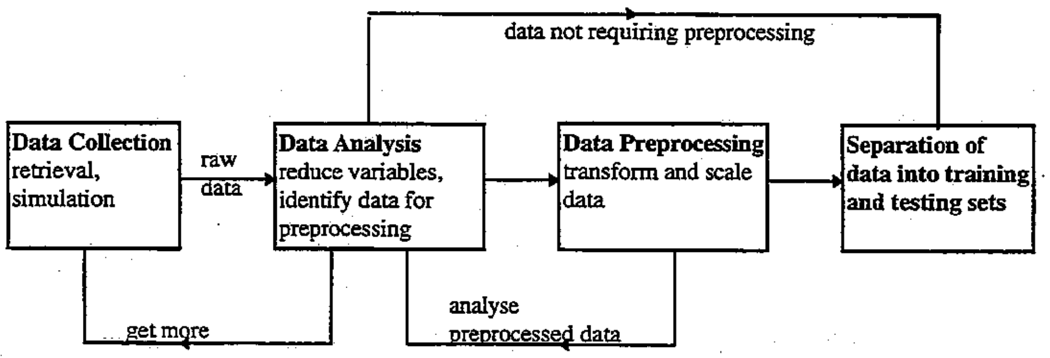

Figure 1: A data preparation scheme

Collected "raw" data is usually noisy, imprecise, and incomplete. Clearly, the training time and the resulting skills, ie the transfer mapping of the network, will depend upon the quality of the training data (Stein, 1993). The collected "raw" data should therefore be analysed before any training is attempted, and should, if possible, be preprocessed to improve its "quality".

The collected data is first analysed by, for example, statistical methods and/or neural network technology, to screen out the "irrelevant" data and to determine what preprocessing, if any, is required. The problem of analysis of the data or, more appropriately, determining the structure of the "analysis → preprocessing → analysis" loop, is a vast and virtually untouched problem, part of which is the determination of a neural network architecture suitable for the class of data to be processed. The "relevance" of data seems more basic, implying a particular need for the involvement of prospective network users and application domain experts in the system development process and especially in its data analysis stage. In this sense, the common perception that "neural networks learn by themselves and thus require little human expertise" is largely a myth.

It should be kept in mind, however, that until a comprehensive theory has been developed, any data preparation scheme will remain a trial-and-error affair to be adapted to each individual situation, guided by basic statistical principles, circumstantial evidence, general knowledge, and any applicable heuristic.

4.1 Data collection

As pointed out, data collection should be preceded by a thorough analysis of the problems so as to identify the variables and understand their relevance. The data should cover the widest range of the problem domain, including not only typical values but also exceptions and conditions at the "boundary" of the domain.

Financial analysis and decisions mostly involve historical data such as previous sales, exchange rates, stock prices, all ordinary index, and consumer price index. Such historical or "secondary" data (Davis & Cosenza, 1985) can be obtained by the retrieval method. The secondary data has been collected by others, such as governments, banks, corporations and universities, for their own purposes, and is published regularly in, for example, newsletters, government publications and bulletins, which are normally available in places such as universities, public libraries, corporations and government departments. The main advantages of the retrieval method are time and cost efficiency. However, the secondary data was not designed specifically to meet a particular problem, and we often cannot assess the accuracy of the data because we know little about the conditions under which its measurement took place, a check for accuracy, relevancy and reliability is still necessary. One method for checking accuracy is to cross-check different sources of data. If the data is obtained from a historically reliable source, it is likely to be reliable, while data from an 'expert' source is likely to be relevant.

Problems may arise when we intend to collect certain historical data such as previous sales and detailed transaction records from a firm. The required historical data may be incomplete in the firm. Collection of data from a firm needs cooperation from its management. Management may be reluctant to disclose data, particularly the data used in strategic decision making, as this data may be leaked to the firm's competitors. Computer simulation then provides a more suitable method of obtaining such kind of data.

Computers can simulate the operations of a firm and generate data to be used in substitute of the real thing. Simulation can generate data within a very short time, and the variables and operational conditions can be adjusted and controlled to produce a set of data well adapted to further processing. However, the operations of a firm are usually complex. It is, of course, not possible to simulate the firm's operations exactly, and simplifications have to be made. On the whole, simulation is a reliable and fast method to obtain data, and is particularly well suited to develop prototype systems.

4.2 Data analysis

Raw data may be noisy, poorly distributed and irregular. Though, in principle, neural networks can adapt to any kind of data, in practice, raw data has been found inappropriate for learning. High quality data for the neural networks improves its input-output mapping, and reduces learning time. However, high quality does not necessarily mean "detailed and precise". In everyday experience, we seldom need detailed data to reason about facts, events and problems. Very often, too much details are confusing and distracting, and may prevent us from organising raw data into useful patterns.

A large number of variables is usually involved in financial decisions. However, there may be correlations among some of these variables, and because of this, the actual number of variables to be considered can be less. Factor analysis is a statistical technique but does not involve sophisticated mathematics to reduce the number of observed variables, by summarising the information contained in the large number of variables into a smaller number of factors. It is done by constructing a matrix of intercorrelations between the variables, and then combining the variables into a new set of *factors* on the basis of relationships in the correlation matrix. The variables with high correlations among themselves are usually combined as one factor. However, the justification of the groupings should be checked by examining the factor loadings (ie correlation coefficients) between the variables and the factors. The grouping of the variables into a factor is justified if there is a high factor loadings, say greater than 0.55.

When human beings make decision, they normally interpret the actual values of the variables as some meaningful forms such as high or low, good or bad. If neural network has to simulate human decision making process, input to the network should better be a kind of rating scales rather than the actual values. For example, the $1,000 net cash flow is meaningless unless it is represented as a certain degree of high or low, which provides a more useful piece of information for decision making. Such variables should be identified for preprocessing.

All financial data can be presented as a collection of real-valued variables. Naturally, the variables would normally have different ranges and variabilities. During training, fluctuation in variables with large ranges will tend to 'overwhelm' the variables with small ranges. Further, variables with similar ranges need not have similar variability. Large variability of some variables would tend to 'distract' the neural network relative to a smaller but perhaps no less important variation in some other variables. In the treatment of a single variable, frequency distribution plot of data is a useful visualisation of its range and variability, and helps to determine whether the variable needs preprocessing.

4.3 Data preprocessing

Data preprocessing is essentially the implementation of the results of data analysis. It involves data transformation and scaling.

Often, the data is transformed using explicit ("elementary") mathematical functions of the original variables, eg logarithm and sine functions. Such functions may be used in combination to transform the whole collective domain of all the variables. The result of transformation can change the data range, or extract the useful information from the raw data. For example, in the analysis of time series, the statistical measures and differentiation techniques are used to extract the "trend" from the data. The variability of exchange rates can thus be measured by the coefficient of variation, and their direction of movement is determined by the first derivatives at specific points.

Scaling is a "procedure for the assignment of numbers to a property of objects in order to impart some of the characteristics of numbers to the properties in questions" (Phillips, 1971). One scaling method which is particularly useful for the continuous financial values is to convert them to "standard scores"[1], and use these as the basis of scaling. The "standard score" method makes the values independent of the units of measure, maps input to roughly equivalent ranges, in practice roughly between -3 and +3, and gives equal value to variations of equal rareness, regardless of the absolute range of the variation. This is particularly useful to minimise the influence of the absolute range and variability of one variable over the other. It is worth to re-analyse the scaled data, for example, by a frequency distribution plot, to examine its range and variability, and to revise the selected scaling method if necessary.

4.4 Separation of data into training and testing sets

It is important to assemble suitable sets of data for training and for testing the network. The data in both sets should be representative of the data on which the neural network will ultimately be used. Further, the training set should be large enough and so structured as to allow easy generalisation to data close by but not in this set.

There are no established rules to select training and testing sets. In practice, the selection seems to depend on the type of problem. In forecasting time series, such as the predictions of future stock prices (Baba & Kosaki, 1992) and foreign exchange rates (Refenes, 1993), both time-dependent, the testing set follows immediately the training set on the time-line. In credit assessment (Klimasauskas, 1993), the selection of training and testing sets has been based on the frequency of occurrence of each of the outcome categories. The automotive diagnostics classification problem (Marko, *et al*, 1990) uses a "leave-k-out" procedure, in which the network is trained and tested for multiple random (N-k, k) partitioning of the N available data sets.

5 Example: an intelligent information system in exchange rate risk management

Firms engaging in international trade are subject to risk from fluctuation of currency exchange rates. Although such risk can be actively managed with suitable hedging strategies, the problem is a complex one, involving large volume of financial data, which can be imprecise and noisy. Small firms rarely have access to required expertise to handle the problem. In a current project (Chan & Bonner, 1995), neural networks technology is used in combination with an expert system to determine a hedging strategy for the exposed net cash flow in a small import/export firm. The project is approaching the stage of a working prototype. We illustrate briefly with some of the experience gained in the treatment of data in the development of the prototype.

5.1 Data collection

The financial data, such as the interest rates, exchange rates, inflation rates, balance of payment, GNP, balance of trade and money supply, were retrieved from public sources such as the bulletins of The Reserve Bank of Australia and The Australian Financial Review. These sources contained

[1] Standard score is the number of standard deviations above or below the mean.

complete and reliable records of these financial data. The firm's data was generated in computer simulation of the firm's importing and exporting activities. As a small firm normally has an unpredicted number of sales contracts, the simulation of its importing/exporting activities thus involved an application of the Monte Carlo technique. The number of contracts obtained by the firm at a given point in the simulation was determined by a chance process described in the form of a probability distribution. The simulation model generated sales and order contracts, and calculated the revenue, expenses and balance every week over the entire period of two years of simulation.

5.2 Data analysis

Factor analysis was used to summarise the data collected by retrieval and simulation methods. A correlation matrix showing the correlation coefficients among all variables was constructed, and, using the principal component analysis technique that the correlated variables were combined as one factor, all the observed variables were then reduced to three factors, as shown in Figure 2.

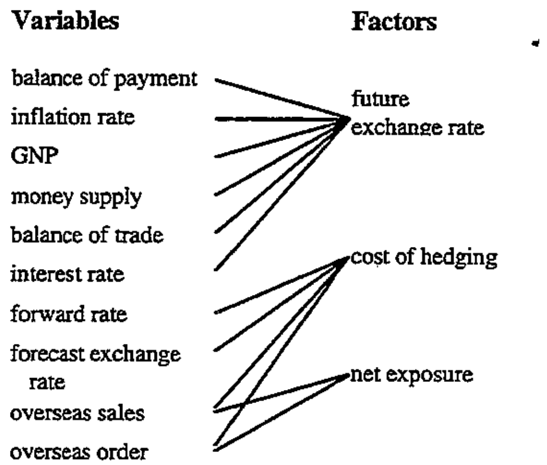

Figure 2: Factor analysis: reduce variables to a less number of factors

5.3 Data preprocessing

While it is sometimes impossible to predict future currency values with much accuracy, the firm can evaluate historical exchange rates in order to assess the potential change of the currency (Madura, 1992). The factor of *future exchange rate* was split into two variables: *currency variability* and *direction of currency movement*, the values of which were obtained by statistical and differential techniques. The standard deviation statistic was used to transform the historical exchange rates into *currency variability*, ie calculating the standard deviation of the historical exchange rates over a past period from a given point which is equivalent to the exposure period; and the *direction of currency movement* at a given point, x_0, was obtained by 'normalising' the first derivative of the line formed by past five data points, according to the formula given by Stein (1993)

$$\frac{dx_0}{dt} = \frac{25x_0 - 48x_{-1} + 36x_{-2} - 16x_{-3} + 3x_{-4}}{12|x_0|}$$

To be financially comparable, the values of *net exposure* were first converted to their present values. The values of the new variable *net exposure present value* and the *cost of hedging* were of little use in financial decisions unless they could be interpreted in a meaningful way (see Section 4.2). For example, a value of *net exposure present value*, say $1,000, could not tell whether the exposure was high or low, and a measure of exposure was related to the size of the firm. For this reason, both the *net exposure present value* and *cost of hedging* had to be scaled.

The first step of scaling the *net exposure present value* was to replace its values by their "standard scores", using the formula

$$\text{standard score} = \frac{x - \bar{x}}{\sigma}$$

where $\bar{x}$ is the mean of all x values,
σ is the standard deviation of all x values.

On basis of the "standard scores", the values were then scaled to the appropriate integers which were more useful than the actual values. The *net exposure present value* was scaled from 1 (lowest) to 4 (highest), indicating the degree of net exposure. Values less than the standard scores of -2 were scaled to 1, between -2 and 0, scaled to 2; between 0 and 2, scaled to 3; and greater than 3, scaled to 4.

The values of *cost of hedging* were scaled in a slightly different way. Firstly, the values were expressed as percentages of the net exposure. Then the percentages were replaced by the standard scores, and scaled in the same way as for the *net exposure present value*.

5.4 Separation of data into training and testing sets

After the data had been preprocessed, there would be four variables input to the neural network: *size of net exposure*, *size of cost of hedging*, *currency variability*, and *currency movement*. The output for the corresponding input was the *hedging decision*: hedged or not hedged. The input-output example pairs were divided into training and testing sets. The testing set was extracted by picking out every tenth example in the data set, in chronological order.

6 Conclusions

"A model is only as good as it is useful to people in thinking about, organising, and using data" (Tsichritzis, 1982). The development of an intelligent information system should be based on a knowledgeable assessment of its true potential and limitations. The understanding of the principles of "intelligence" is essential. An examination of typical such systems presently in use in the area of financial management suggests, however, that these requirements may not have been met in their development. In particular, neural network technology seems often to be used as substitute for thorough analysis of problems. The question "why neural networks?" remains all too often unanswered.

Though neural network technologies are, to degree, self-learning, compared to traditional information systems, both developers' and users' commitment to the development of intelligent information systems is not less vital. In particular, the technologies stand or fall with adequate data, which is not realistic to obtain without users' committed involvement. Further, like any (formal) system, a neural network system is a schema defined by its variables. The question *which variables are "relevant"?*, is thus no less important here than in the development of traditional information systems, and can only be settled with domain expertise. Another issue specific to intelligent system development is the particular need to ascertain the knowledge of the users, and the related questions of decision leverage control and user training. Summing up, a systematical approach to the development of intelligent information systems is lacking and is badly needed.

Whatever the underlying computational scheme, it is clear that the "intelligence" of computer-based information systems is continuously increasing. On the one hand, one can see new and exciting opportunities of extending analytical models with computer-based learning systems which are able to operate in a knowledge-rich environment, and are thus increasingly able to interactively solve practical and unstructured problems on the basis of partial knowledge. This, in particular, would suggest the border area of general financial theories and financial management practice as a very promising application domain. On the other hand, however, it seems rather optimistic to believe that the "intelligence" in the systems will take care of itself. There is a clear need for a basic policy on the development, use, and management of intelligent information systems, both in specific applications domains and in the society in general.

References

Arbib, M. A., and M.A. Hesse (1986) *The Construction of Reality*, Cambridge University Press, Chapter 3.

Baba, N. and M. Kosaki (1992) "An intelligent forecasting system of stock price using neural networks," *International Joint Conference on Neural Networks, (Baltimore, Maryland)*, vol 1, New York: IEEE, pp. 371-7.

Batini, C., *et al.* (1992) *Conceptual Database Design*, Benjamin/Cummings Publishing Co.

Bonner, R. F. (1993) "What can mathematics do for information systems?", *Proceedings 4th Australian Conference on Information Systems*, University of Queensland, St Lucia, Queensland, Australia, September.

Bonner, R. F, and R. Chan (1995) "A mathematical modelling perspective on information systems: basic concepts", working paper, School of Information Systems and Management Science, Faculty of Commerce and Administration, Griffith University, Australia.

Chan, A. C. K., and R. F. Bonner (1995) "A hybrid intelligent system in managing foreign exchange rate risk," to appear in *Proceedings of Symposium on Modelling and Control of National and Regional Economics* (IFAC '95), Gold Coast, Queensland, Australia, July.

Charness, N. (1981) "Search in chess: age and skill differences," *Journal of Experimental Psychology*, **7** (2), pp. 467-76.

Chase, W. G. and K.A. Ericsson (1981) "Skilled memory," in Anderson, J. R. (ed), *Cognitive Skills and Their Acquisition*, Hillsdale, NJ.

Collins, E., S. Gosh, and C.L. Scofield (1988) "An application of multiple neural network learning system to emulation of mortgage underwriting judgements," *International Conference on Neural Networks, Vol II*, IEEE, pp. 443-50.

Davis, D. and R.M. Cosenza (1985) *Business Research for Decision Making*, Boston, MA: Kent Publishing Company.

Dutta, S. and S. Shekhan (1988) "Bond rating: a non-conservative application of neural networks," *International Conference on Neural Networks, Vol II*, IEEE, pp. 443-50.

Hogeweg, P. (1993) "As large as life and twice as natural: bioinformatics and the artificial life paradigm," in Green, D. and T. Bossomaier (eds), *Complex Systems: from Biology to Computation*, Proc. 1st Australian Complex Systems Conf., IOS Press, Amsterdam.

Jagielska, I. (1993) "The application of neural networks to business information systems," *Proceedings of Australian Conference on Information Systems*, University of Queensland, Brisbane, September., pp. 565-574.

Klimasauskas, C. (1993) "Applying neural networks, Part II: A walk through the application process," in Trippe, R. R. and E. Turban (eds), *Neural Networks in Finance and Investing*, Chicago: Probus Publishing Company, pp. 54-63.

Koster, A., N. Sondak, and W. Bourbia (1990), "A business application of artificial neural network systems," *The Journal of Computer Information Systems*, 31 (2), Winter, 1990-91.

LeCun, Y., *et al* (1989) "Backpropagation applied to handwritten zip code recognition," *Neural Computation*, 1 (4), Winter, pp. 541-51.

Li, J., J. Wu, and Y. Zheng (1991), "Neural network to support unstructured economic decision process," *International Joint Conference on Neural Networks*, 3, 18-21 November, Singapore.

Madura, J. (1992) *International Financial Management*, West Publishing Company.

Marko, K. A., L.A. Feldkamp, and G.V. Puskorius (1990) "Automotive diagnostics using trainable classifiers: statistical testing and paradigm selection," *International Joint Conference of Neural Networks (San Diego, CA), Vol. I*, New York: IEEE, pp. 33-8.

Meese, R.A. and K. Rogoff (1983) "The out-of-sample failure of empirical exchange rate models: sampling error or misspecification?" in Frankel, J. (ed), *Exchange Rates and International Macroeconomics*, Chicago: University of Chicago Press, pp. 67-112.

Mish, F. C. (ed) (1989) *Webster's Ninth New Collegiate Dictionary*, Springfield, MA: Merrian-Webster Inc. Publishers.

Newell, A. and H. Simon (1972) *Human Problem Solving*, Englewood Cliffs, NJ: Prentice-Hall.

Phillips, B.S. (1971) *Social Research Strategy and Tactics*, New York, The Macmillan Company.

Poggio, T. and F. Girosi (1990) "Networks for approximation and learning," *Proceedings of the IEEE, Vol 78 (9)*, September, pp. 1481-97.

Refenes, A. N. (1993) "Constructive learning and its application to currency exchange rate forecasting," in Trippe, R. R. and E. Turban (eds), *Neural Networks in Finance and Investing*, Chicago: Probus Publishing Company, pp. 465-93.

Rumelhart, D. E. (1989) "The architecture of mind: a connectionist approach," in Posner, M. I. (ed), *Foundations of Cognitive Science*, Cambridge, London: MIT Press, pp. 133-59.

Serra, R. and G. Zanarini (1990) *Complex Systems and Cognitive Processes*, Berlin: Springer-Verlag.

Simon, H. A. (1983) "Why should machines learn?", in Michalski, R. S., Carbonell, J. G. and Mitchell, T. M. (eds), *Machine Learning: An Artificial Intelligence Approach*, Palo Alto, CA: Tioga.

Sejnowski, T.J. and C.R. Rosenberg (1986) *NETtalk: A Parallel Network that Learns to Read Aloud*, John Hopkins University Electrical Engineering and Computer Science Technical Report (86/01).

Stein, R. (1993) "Preprocessing data for neural networks," *AI Expert*, 9, March, pp. 32-37.

Taylor, J.G. (1993) *The Promise of Neural Networks*, London: Springer-Verlag, Chapter 7.

Trippe, R. R. and E. Turban (eds) (1993) *Neural Networks in Finance and Investing*, Chicago: Probus Publishing Company.

Tsichritzis, D. C. (1982) *Data Models*, Englewood, Cliff, N. J.: Prentice-Hall.

VanLehn , K. (1989) "Problem solving and cognitive skill acquisition," in Posner, M. I. (ed), *Foundations of Cognitive Science*, Cambridge, London: MIT Press, pp. 527-79.

Wan, H.A., and A.K. Chan (1993) "A feedfordward neural network model for Hang Seng Index," *Proceedings 4th Australian Conference on Information Systems*, University of Queensland, St. Lucia, Queensland, Australia, September, pp. 575-86.

Zurada, J. M. (1992) *Introduction to Artificial Neural Systems*, St. Paul: West Publishing Company.