Artificial Neural Networks in Financial Modelling

Le Reti Neurali Artificiali nella Modellizzazione Finanziaria

Crescenzio Gallo[1]
Dipartimento di Scienze Economiche, Matematiche e Statistiche
Università di Foggia
Via IV Novembre, 1 - 71100 Foggia (Italy)

Abstract: The study of Artificial Neural Networks derives from first trials to translate in mathematical models the principles of biological "processing". An Artificial Neural Network deals with generating, in the fastest times, an implicit and predictive model of the evolution of a system. In particular, it derives from experience its ability to be able to recognize some behaviours or situations and to "suggest" how to take them into account. This work illustrates an approach to the use of Artificial Neural Networks for Financial Modelling; we aim to explore the structural differences (and implications) between one- and multi- agent and population models. In one-population models, ANNs are involved as forecasting devices with wealth-maximizing agents (in which agents make decisions so as to achieve an utility maximization following non-linear models to do forecasting), while in multi-population models agents do not follow predetermined rules, but tend to create their own behavioural rules as market data are collected. In particular, it is important to analyze diversities between one-agent and one-population models; in fact, in building one-population model it is possible to illustrate the market equilibrium endogenously, which is not possible in one-agent model where all the environmental characteristics are taken as given and beyond the control of the single agent. A particular application we aim to study is the one regarding "customer profiling", in which (based on personal and direct relationships) the "buying" behaviour of each customer can be defined, making use of behavioural inference models such as the ones offered by Artificial Neural Networks much better than traditional statistical methodologies.

Keywords: Artificial Neural Network, Financial Modelling, Customer Profiling.

1 Introduction

An increasing field of research in artificial neural networks (ANN) [Kohonen (1984), Rumelhart and McClelland (1986), Wasserman (1989)] is the one mainly concerned with interactions between economics and computer science, studying their potential applications to economics (and to modelling economies with artificial agents). Being a lot of "literature" already available on financial applications, we based our research interests on a theoretical approach to artificial neural networks, aiming to study their potential applications to financial data, too.

The mainstream in actual research programs can be sintethized by the following question: is it possible to use ANNs to study the behaviour of economic enivronments, mainly in those situations not fully covered by "official" economic theories? This can be done using ad hoc-written software for representing the economic behaviour of agents in a market [Caudill and Butler (1992)], not with the objective of developing new algorithms or economic models, but with the target of looking at economic issues in a new manner, that of ANNs, hoping to understand whether the so created artificial market replicates some of real market phenomena. The spreading of forecasting tools, alternative to those traditionally used by financial analysts, made the widening of both methdological features and relevant aspects for the application to financial data series necessary. Particularly,

[1] Phone: +39 3204394567 E-mail: c.gallo@unifg.it

neural networks are ideal for financial applications because they allow non-random and non-linear dynamics recognition. This feature derives from the fact that networks "learn" [Parisi *et al.* (1990)] how phenomena evolve in time thanks to a relatively long series of trials allowing the optimization of the weights linking inputs to outputs through intermediate layers neurons. To minimize the overall network construction time it is possible to follow a sequence of operations conditioning the final result, and then the total forecasting effectiveness.

2 Research outline

Artificial neural networks represents an easily "customizable" tool for modelling learning behaviour of agents and for studying a lot of problems very difficult to analyze with standard economic models. Very often economists have to operate dramatic simplifications of real world, especially regarding the learning mechanisms of agents and the learning-action link which derives from them; now, we want to investigate in the direction of creating an artificial model of economics [Lane (1993a,b)] with adaptive agents [Cliff *et al.* (1992)], simulated by the nodes of a neural network. This work is divided in a first part, illustrating some significant models of artificial agents; subsequently, a series of practical problems are outlined relevant to the interactions between economy and computer theory, followed by examples of ANNs modelling economic agents and showing behavioural rules derived from simple initial requirements, evolving towards quite complex simulated environments. At the end, we point out interactions of artificial agents in a series of situations, especially those belonging to differents sectors of the economy, the main interest being finalized to the interpretation of agents' actions in terms of simple behavioural rules.

3 Artificial Neural Networks

In this section we introduce the main features of artificial neural networks (ANNs), avoiding biological aspects and concentrating upon their applications in financial and economic problems. First, we describe the general architecture of ANNs, pointing out the artificial model, and then the training and learning mechanisms, such as competitive, genetic and back-propagation.

We may define an artificial neural network as a mathematical model made of a great number of elements organized in levels; an ANN may also be viewed as a "computing system" composed by a high number of simple interconnected elements which process information modifying its dynamic response to external solicitations; more precisely [Hecth-Nielsen (1987)] *"an ANN is a dynamic system represented with a directed graph; it may process information producing a state as a result of initial or continuous input; nodes are the processing elements and edges are the information channels; every element produces a single output signal which may travel upon one or more channels"*.

The building of a neural network goes through the definition of various steps:

1. identifying the forecasting target;
2. building the data set upon which to activate neural network learning;
3. activating network learning, with the choice of the architecture and parameters necessary for the definition of the connection weights between neurons;
4. generalizing the output for financial markets forecasting.

3.1 Identifying the forecasting target

The first step in buinding a neural network for financial applications is the one requiring the definition of the analysis target. In general, we may state that neural networks allow to face with financial problems otherwise analyzed with essentially linear methodologies. Actually, phenomena (and not only financial ones) only rarely manifest themselves in linear form and however never keep in time this regularity.

Neural networks application areas are very diversified. According to the classification proposed by the *International Conference on Neural Networks in the Capital Markets* (NNCM) they spread from portfolio management, payoff curve models estimate, evaluation of shares and bonds, trading, coverage and arbitrage strategies, cointegration, volatility and correlation analysis, to forecasting. In the latter, analysis target can be relative to share, bond or future prices, interest or exchange rates.

Once decided the within of surveying, it is necessary to establish the borders of the measurement of the phenomenon. The choice of the more useful forecasting pointer depends on specific and external factors of the operator. In the first case, it is possible that the objective is to use the result of the neural network for the implementazione of a trading system. Different is the case of an analyst whishing to characterize the various alternatives of market: in this case it is not so important to know that a market is in rise, but rather to have an idea of the entity of that rise for being able to confront it with dynamics forecasted on the other markets. It will therefore be needed a previsional system being able to signal the future price or rate variation.

Among the external factors to the analyst that can condition the choice are the possible incompleteness or irregularity of an historical series, the noise due to market inefficiency elements that can condition the analysis negatively.

A further aspect relative to the definition of the forecasting target is the one that must concur to decide the frequency of data upon which to obtain the output. Also in this case the factors that must suggest the choice depend on endogenous and exogenous motivations. The first are to be characterized in the temporal horizon on which the analysis activity is made: if to the traders a forecast to short expiration will turn out interesting, for patrimony or investment fund managers a less frequent indication will be sufficient. Generally, the greater is the incidence of data bias the more a reduced regularity of the series is advisable, in order to work on mean values that cancel irregular fluctuations around a path more easy recognizable by the neural network.

3.2 Building the data set

The relevant steps in building a neural network are:

- the collection;
- the analysis and transformation;
- the selection of the input and output variables.

Information gathering must meet some fundamental principles: first, data must be recovered from markets regularly, in order to guarantee the historical series continuity, at least with the chosen forecast frequency; moreover, it is fundamental to implement a periodic data set feeding, in order to update series parameters. A particularly delicate phase of the construction of the neural network is that of the input and output data preparation. In addition to the traditional statistical analysis issues, there is the necessity to leave in the historical series all the necessary elements for the non-linear learning of the network and to eliminate those that can condition negatively its results.

A first problem to resolve is that of data selection. This aspect is intuitively decisive, as the learning of whichever system, biological or artificial, depends on available information: if some of these data lack, also the behavior turns out in some way to be conditioned. In the case of a forecast, supposing the network to recognize a pattern, it would have theoretically to be put in the same cognitive conditions of a market analyst.

The factors that determine the choices on the markets must be recognizable by the network: therefore are important both the theoretical models and the information that condition the economic expectations of the markets. Naturally, not all the information will turn out - ex post - relevant: therefore, a continuous calibration of the database is required, with the objective to eliminate data not modifying the informative content of the network. Substantially, the analysis would have to follow this steps:

a) wide definition of the first database;
b) first learning of the net;
c) evaluation of the informative content of the single variables;
d) analysis of the matrix of correlation between the input variables;
e) progressive elimination of the less less meaningful and more autocorrelate variables;
f) subsequent learning of the network with the reduced database.

This process tends to identify the optimal data sets depending on the analyzed problem, according to an iterative scheme that must guarantee the maximum network learning ability with the minimal informative effort. In particular, the observation of the correlation coefficients let alone of the related graphs allows to estimate the nature of the eventual interrelation inside the input variables set and between these and the one to forecast. From such process derives the generation of a correlation matrix among the variables to analyze, whose main diagonal is characterized from unitary values.

The linear correlation coefficient between two variables x and y varies in the range $[-1;+1]$ and measures the mean variation of dependent variable x when the independent variable y increases by a unit. The generation of the correlation table allows to remove from the input variables set that more intensely correlated, in order to avoid potentially ambiguous or delayed signals. More wide are the data spread around the regression line, the greater turns out to be noise component of data that can disturb the learning process of the neural network.

The neural network for the forecast of historical financial series can be set up using different typologies of information:

a) those directly tied to the output variable (market given);
b) those resulting operatingly connected with the output variable (intermarket given);
c) finally, those depending on the fundamental components (fundamental data).

The one described can be considered the general model. In fact, the contribution of intermarket or fundamental variables not always turns out meaningful in order to explain the phenomenon: this can depend on the frequency of the observations, the existence of obstacles to the movement of capitals or, still, on autocorrelation factors that prevent from adding knowledge to the neural network. The general outline of data necessary for neural networks processing is that of 1.

In turn, market data can be processed by means of technical analysis pointers that allow to deepen some dynamics and tendency and cycle reversal marks. These further information is relevant for the neural network, since the objective is to recognize behavior models that the only historical output series does not necessarily exhibit. The pointers of technical analysis more frequently used can be calculated also for historical series of financial variables determined on different markets; to these market elements at least two fundamental elements join: the economic cycle and the dynamics of price indices.

A problem related to the insertion of fundamental data is that of the survey frequency. Two solutions are preferable:

a) not using the historical series of the effective values (which are only known ex-post) but those expected by the market;
b) replacing fundamental data with a *dummy* variable measuring the only time of the information: in such a way it is necessary to add an historical series whose values are null in the moments in which no important information is communicated to the market, while they are

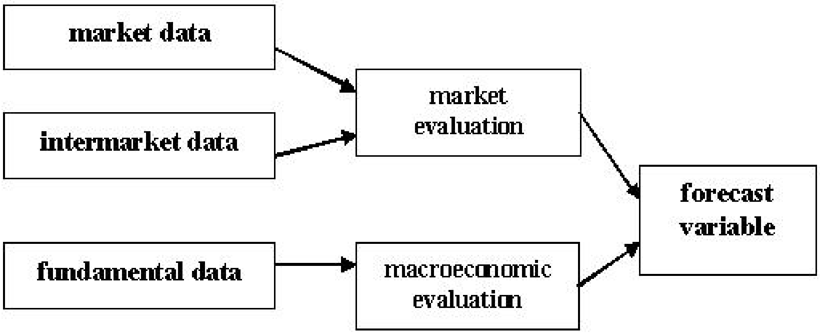

Figure 1: *Input variables of neural networks for financial forecasting*

unitary in correspondence of the information that can have a meaningful impact for dynamics of the series to forecast.

3.3 Activating network learning

The construction of the neural network goes necessarily through some steps that allows to fix the parameters useful for the learning suitable to the solution of the problem. The first phase of the analysis is that of the choice of the architecture more suited for the learning. The delicacy of this phase depends on the fact that the connection mechanism of input nodes between themselves and between them and the output ones, through the hidden layers, turns into a decisive element for the success of the operation.

In literature exist some algorithms proposed in order to optimize the choice of the architecture (pruning algorithm, polynomial time algorithm, canonical decomposition technique, network information criterion), but it is thought above all that experience and test phase are more reliable criteria, if do not exist direct references to the problem to resolve. The characteristics of the architectures can moreover constitute a further element of choice: each of these options contains some parameters that must be defined by the analyst, and the choices can also involve numerous attempts at sharpening for the location of the better result.

Generally it can be thought that the parameters to determine in the definition of the architecture are:

a) the temporal subdivision of the database;
b) the number of the hidden layers and the neurons to insert in every layer;
c) the connection mechanisms between the different layers;
d) the activation function;
e) the rules of learning;
f) the updating of neurons' connection weights.

3.3.1 The temporal phases of learning and forecasting

Once the shape of output and the content of the data set from which to extract input variables have been defined, the historical series must be subdivided in subperiods, which delimit the learning (training set) and evaluation sphere. This last is distinguished in test and generalization set. In short, the network learns trying to recognize the dynamics of the training set, verifies its adaptation on the test set and then applies itself on a data set (generalization) that it has not never been able to observe. The period of generalization can also be ignored, in the case in which the analyst had at her disposal data external to the original set and only subsequently she wanted to apply network results to those data: but this solution implies a wider time for the verification of the goodness of the network.

The subperiods extraction criteria are of various kinds:

a) random, in which the analyst only defines the proportion percentage of the three subperiods;
b) defined in quantitative terms but training, test and generalisation data turn out to be in variable position;
c) training and test data are fixed but the network extracts in a random way: the generalization series instead is held to the end of the historical series;
d) the historical series is rigidly subdivided in a period of training, one of test and one of generalization, following this chronological order.[2]

Universally valid rules do not exist for the subdivision of the historical series to analyze; they go from the suggestion to use $2/3$ of the total of the series for the training to the indication of a number of data used for learning not less than $4/5$ of the total of the series: the most adopted solutions are, respectively, $(60\%; 20\%; 20\%)$ and $(60\%; 30\%; 10\%)$.[3]

3.3.2 The number of layers and neurons

In relation to the number of layers, numerous methodological contributions are found according to which a single hidden layer would be sufficient to approximate the more recurrent non-linear functions with high accuracy degree. For this reason, are numerous the empiric forecasting works who estimate the network with a single hidden layer; the limit of this approach is bound to the necessity of the use of a high number of neurons, that limits the learning process. The use of networks with two hidden layers turns out more effective for forecasting problems on high frequency data.

With reference to the numerosity of the neurons that must be assigned to every hidden layer, the criterion to adopt in the choice is that of *overlearning*'s risk minimization, that occurs when an excessive number of neurons is inserted which design nearly perfectly the pattern of the historical series but are not able to generate a reliable forecast, because they reduce the contribution of the inputs. Viceversa, the risk of assigning a too much low number of neurons is to reduce the potential of learning of the neural network.

So, it is necessary to find a trade-off solution between a too much high or too much low number of neurons. The formulas proposed in literature are several and, in some cases, conflicting:

$$n_{hl} = 2 \cdot n_{in} + 1 \tag{1}$$

$$n_{hl} = 2 \cdot n_{in} \tag{2}$$

$$n_{hl} = n_{in} \tag{3}$$

$$n_{hl} = \frac{n_{in} + n_{out}}{2} + \sqrt{n_t} \tag{4}$$

where

n_{hl} is the number of hidden layers;
n_{in} is the number of inputs;
n_{out} is the number of outputs;
n_t is the number of observations contained in the training set.

[2] This last criterion is the one suggested within networks applied to financial market forecast.

[3] In literature and some application software, the test set is called "validation set" while the generalization set is called "production set".

3.3.3 The architectural connections

The Back Propagation architecture (widely used and diffused also to software level, for its ability to generalize results for a wide number of financial problems) is defined *supervisioned architecture*, whose learning is conditioned both by input and output variables. Numerous possibilities of connection exist:

- standard connections;
- jump connections;
- repeated connections.

Standard connections are between input and output, passing through one or more hidden layers, to which have been assigned neurons that facilitate the learning of the network. The connections, apart from the number of hidden layers (that can be arbitrarily high, but at the cost of a higher learning time), are directed and do not provide for jumps or loops:

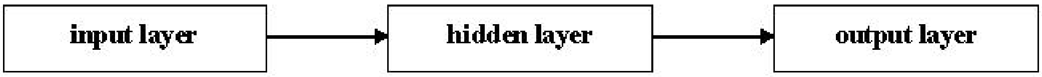

Figure 2: *Back-propagation architecture with one hidden layer and standard connections.*

Jump connections provide instead for the network to assign connection weights also between neurons not in adjacent layers: in the simplest case the input layer introduces a series of connections not only with the hidden layer but also with the output layer. This connecting ramification is further articulated if the number of hidden layers increases.

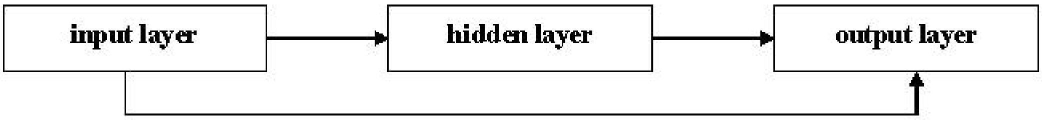

Figure 3: *Back-propagation architecture with one hidden layer and jump connections.*

Repeated connections neural networks turn out useful for the analysis of the financial historical series in how they recognize sequences created within markets. These connections provide for the possibility that the neurons assigned to hidden layers can return on input variable with iterative processes so as to precisely quantify the connecting weight.

3.3.4 The activation functions

The fourth problem during the construction of the neural network is that of the activation function that conditions neurons' link. Typical activation functions are:

- Linear $f(x) = x$

- Logistic $f(x) = \frac{1}{1+e^{-x}}$

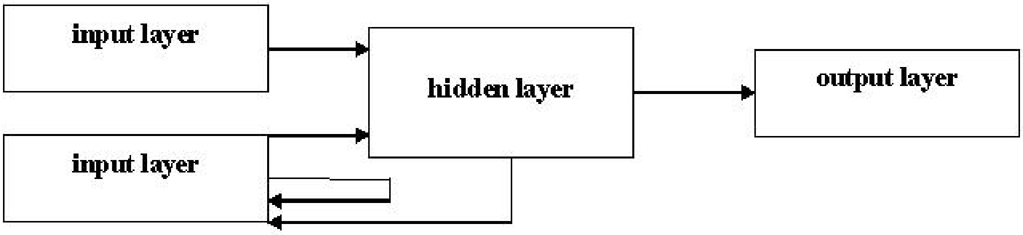

Figure 4: *Back-propagation architecture with one hidden layer and repeated connections.*

- Symmetric logistic $f(x) = \left(\frac{2}{1+e^{-x}}\right)^{-1}$
- Hyperbolic tangent $f(x) = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}}$
- Corrected tangent $f(x) = \tanh(c \cdot x)$
- Sinusoidal $f(x) = \sin(x)$
- Gaussian $f(x) = e^{-x^2}$
- Inverse gaussian $f(x) = 1 - e^{-x^2}$

A theoretically acceptable rule does not exist in order to define the activation function of the various layers. The *linear* function is generally used for the layer containing the output of the neural network[4]. Less effective is, instead, the use of the linear function in hidden layers, above all if these are characterize by a high number of neurons, which would therefore become connected just on a functional base, willing to be overtaken with the use of the network itself. The relevant limit of the linear function is that it does not allow an adequate fitting for historical series characterized by a persistent trend.

The *logistic* and *symmetrical logistic* functions have the characteristic to vary within a range, $[0;1]$ and $[-1;1]$ respectively. The first turns out to be particularly useful in the hidden layers of the networks applied to financial historical series. Some problems introduce dynamic characteristics that turn out to be picked in preciser measure by the symmetrical function, above all in the input and hidden layers.

The *hyperbolic tangent* function allows to adapt the network in reliable way in the hidden layers (in particular in three-layers networks), above all in the case in which the analyst choices a logistic or linear function for the output.

The *sinusoidal* function is generally adopted by researchers and it is suggested to standardize input and output inside the range $[-1;+1]$.

The *Gaussian* function lends itself to the location of particular dynamic processes with two hidden parallel layers architectures, with a tangent function in the second layer.

[4] The reason is that this function, albeit more rigid than the alternatives, avoids the stretchening of the result towards the minimum or the maximum.

3.3.5 The learning algorithm

A further problem consists in the choice of the learning rule; in particular, it is necessary to decide with which change rate the network must modify the definition of the weights of the neurons as to the significance of the error. The risk in deciding a too high rate learning is to condition the network to oscillate too much widely with a consequent loss of the correct evolution of the historical series. In this way a non-convergence phenomenon could happen that must be corrected adequately reducing the learning rate.

In alternative, it is possible to assign a correct value to the momentum. It is relative to the proportion with which the variation of the last weight caught up from the neural network joins to the new weight. In this way the network can learn also to a high rate but it does not risk to oscillate because it recovers a high quota of the last caught up weight. The analyst must then decide the starting level of the weight to give to the connection between neurons and to estimate if the observations are characterized from a high rate noise; in this case it will be opportune to maintain the value of the momentum high. The learning process of the network passes naturally through the progressive location of the best fitting value of these parameters. A numerous series of trials is therefore needed that may generate considerably different outcomes. The suggestion is to avoid radical variations of the parameters.

3.3.6 The error indicators of the neural network

Once the initial characteristics of the neural network have been selected it is possible to define the stop criteria of the learning process. These criteria can first of all be connected to the training or test set: if the objective is to adequately describe the studied phenomenon it is preferable to block the learning on the training set, while if a neural network with previsional purposes is wanted it is preferable to block its learning on the test set.

In the second place, the learning parameters are generally tied to the error pointers made from the network:

- mean error;
- maximum error;
- number of ages without improvement of the error.

Fixing a value to these parameters the network will stop itself once the desired value has been reached. Generally, it is simpler to fix a high number of ages (that are conditioned by the numerosity of observations of the training set) that the network analyzes without improving the error. It can be thought that a value between $10,000$ and $60,000$ ages is sufficiently sure in order to block a network, which by now with great difficulty can learn better than how much it has already made until that moment. Naturally, the choice depends also on the speed with which the network reaches these values.

An element for the acceptance of a network is that one of the convergence. If also the errors turn out modest but the oscillation has carried to a high divergence, it is opportune to verify the adequacy of the parameters. Once a correct error dynamics has been verified, at least in graphical terms, it is necessary to measure it quantitatively. The learning programs of the historical series tend to use numerous statistical error pointers; between these are:

- the determination index (R^2);
- the Mean Absolute Error (MAE);
- the Mean Absolute Percentage Error (MAPE);
- the Mean Square Error (MSE).

These are pointers who measure, in various ways, the differential between the original output and the one estimated by the network; only if the inputs, the neurons, the activation functions and all the parameters previously described were perfectly able to characterize the original phenomenon, the deviations between real and estimated output would be null, thus optimizing the cited error pointers.

A pointer often used in order to optimize the neural network is the Theil's T coefficient, having the advantage to vary within a standardized interval $[0, 1]$, where 0 is obtained with perfect previsional models. The limit of these error pointers is their being based on a concept of symmetrical deviation from the real value, while in finance the single error is only measured in terms of loss. It would be therefore opportune to estimate network weights on the base of the obtained profits; in practical terms, filter strategies can be adopted in order to remove the problem a posteriori.

3.4 Generalizing the output for financial markets forecasting

Once that the pointers are favorable to the choice of the neural network, it is necessary to verify its forecasting goodness. It is in fact possible that a model succeeds to optimally describe the training and the test set but then turning out quite inadequate as far as its generalization, that is - in the financial case - the forecast.

The analyst will have therefore to test, with the same techniques, the neural network on the generalization set. First of all, the already described error pointers will have to be measured on the historical series never observed by the network. In case these had to turn out significantly worse and, however, not acceptable by the analyst on the base of her original targets, the network will have to be further tested.

The financial forecast is also often conditioned by the ability of the model to timely characterize the cyclical reversals of the phenomenon. This property can first of all be verified with an adequate graphical representation.

It is possible to use the original output and that one estimated by the network in order to quantitatively measure the delay with which it learns the cicle reversal. Numerous are the instruments useful in order to analyze the previsional ability of the neural network: at every cicle reversal the phenomenon values' series must record one variation of the sign. If, therefore, it passes from an increasing trend to a decreasing one, the signs of the variations will change from "$+$" to "$-$".

4 Models of Artificial Neural Networks

It may be now important to point out that economic analysis has, by now, sistematically avoided questions about how agents make choices when they interact with real, evolving world: this is so because most known formal models are not very well suited to such problems. But new techniques in computer-based modelling offer the possibility to test "artificial agents" in "artificial worlds" in a variety of ways, from one- to multi-agent formalizations.

In one-agent models there is a single agent, interacting only with the environment influencing its learning, while in a many-agent model there can be a single or multi-population scheme with agents interacting and improving their behavioural skills. In a many-agent model, interaction and nonlinearity makes its results highly unpredictable and complex; a minimal change to agents, rules, goals and environment can drastically modify the whole system, so showing the enormous simulating capability of such models. In particular, the most useful scenario is that of multi-populations interacting with each other, where populations can differ for

a) the input sctructures;
b) the outcomes (which stands for the actions or forecasts made by the agents);

c) the rules applied to agents' expectations;
d) the agents' structure used to develop endogenous rules;
e) the explicit goals, wired into agents' behaviour;
f) the implicit goals, suggested as trainig elements;
g) external suggestions to the agents' actions.

In the first step of construction of such models, we have models of agents interacting with the economic world: this situation is represented with equations which may include the underlying economic theory or simple connections among the variables; in a second step, there are interacting agents generating aggregate results influencing the environment responses. The third step regards agents' interactions, with agents reciprocally influencing by their actions: this is the most complex (and realistic) case, in which agents do not exchange information only by prices, but also learn iteratively by observing other agents' behaviour.

In all these models an important role is played by "consistency" between actions and effects, as outlined in the cross-target technique [Terna (1993)], an ANN technique developed to build adaptive artificial agents without using a priori economic rules. In it, we have a generic effect F_1 (arising from two actions S_1 and S_2) whose target is:

$$F'_1 = f(S_1, S_2) \tag{5}$$

The objective is to obtain an output F_1 (the guess made by the network) closer to F'_1 (the correct measure of S_1 and S_2 effect), with an error $e = \frac{1}{2}(F'_1 - F_1)^2$ or, in other words, to find actions (as output of the networks) more consistent with the outputs produced by the effect side. So we have to correct S_1 and S_2 to make them closer to S'_1 and S'_2 (actions consistent with the output F_1). From equation 5 we have:

$$S_1 = h_1(F'_1, S_2) \tag{6}$$

$$S_2 = h_2(F'_1, S_1) \tag{7}$$

Picking a random value α_1[5] and setting $\alpha_2 = 1 - \alpha_1$, from equations 6 and 7 we obtain:

$$S'_1 = h_1(F'_1 - e \cdot \alpha_1, S_2) \tag{8}$$

$$S'_2 = h_2(F'_1 - e \cdot \alpha_2, S_1) \tag{9}$$

Functions h_1 and h_2 usually are linear, so equations 8 and 9 normally produce globally consistent solutions. The errors to be minimized are:

$$e_1 = S'_1 - S_1$$

$$e_2 = S'_2 - S_2$$

Equations 8 and 9 are used as a simplifying tool, especially for the random separation obtained by α_1 and α_2.

Cross-target method can be used to develop a simple example mechanism of the capability of reacting to price changes, in which the simulated agent makes two independent guesses (based upon a single input, the price a_p[6]): the quantity a_q of the good a_g to be acquired and the expenditure s. Applying cross-target we have:

$$s' = a_p \cdot a_q$$

[5] From a random uniform distribution based on closed interval $[0, 1]$

[6] Changing exogenously by a sin function and a random noise.

$$a'_q = a_q + (s - s')/a_p = s/a_p$$

where s' is consistent with guess a_q and a'_q is the correct decision to acquire the quantity consistent with guess s. If $s' > s$, a_q must be increased by $(s - s')/a_p$ and vice versa. In Figure 5 are illustrated prices (on the horizontal axis) and guessed quantities (on the vertical axis), with a kind of "demand curve" autonomously developed by the model[7]: this can be e.g. viewed as a practical mechanism to develop a "price cap" for volatile markets, as the recently-started electricity italian market (see D'Ecclesia and Gallo (2002) for a deep analysis and a proposed theoretical model of price control).

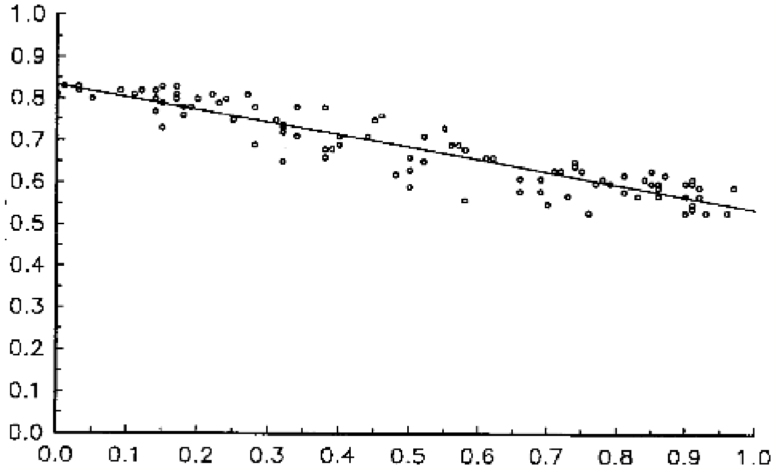

Figure 5: *Example of experimental results for cross-target method.*

5 Conclusion

In previous sections we have considered diverse issues relative to artificial neural networks construction, as economic and financial models, artificial agents and markets. Many dynamic models assume agents with knowledge of the economic environment in which they operate, in which case learning is not so relevant. Not often it is assumed that agents lack of such a knowledge and act according to models iteratively approximating as more data become available: here learning is crucial for the analysis and its behaviour is associated with econometric algorithms. ANNs may be useful for understanding the basis of learning behaviour, and they have many features of real world learning and reasoning, with a robust association between inputs and outputs also in presence of noise in the inputs, besides an "inductive" and adaptive learning behaviour as the environment changes.

Much work has to be done to better understand how such learning takes place, especially in how agents learn and behave. In particular, the study of financial markets leads practictioners to rely not only upon official academic theories, but to use ANNs to get a right analysis of financial markets; obviously, a precise evaluation of the ANNs' goodness may be done only when an artificial financial market is fully developed, but today we are still far from that. The implications of ANNs for the efficiency of financial markets shows that it is its very idea that must be redesigned, in order to come to a more practical definition that considers as well computing resources now available.

[7] On price changes, the cross-target mechianism determines two symmetrical corrections

References

Caudill M. and Butler C. (1992) *Understanding Neural Networks: Computer Explorations*, volume 1 and 2, MIT Press, Cambridge, MA.

Cliff D., Harvey I. and Husbands P. (1992) Incremental evolution of neural network architectures for adaptive behaviour, Research paper n. 256, University of Sussex, Brighton.

D'Ecclesia R. and Gallo C. (2002) Price-caps and efficient pricing for the electricity italian market, *Quaderni del Dipartimento di Matematica, Statistica, Informatica e Applicazioni*, vol. 5, Università di Bergamo.

Hecth-Nielsen (1987) *Neurocomputer Applications: Neural Computers*, Springer-Verlag, Berlin.

Kohonen T. (1984) Self-organization and associative memory, in: *Series in Information Sciences*, Springer-Verlag, Berlin.

Lane D.A. (1993a) Artificial worlds and economics, part i, *Journal of Evolutionary Economics*, 3, 89–107.

Lane D.A. (1993b) Artificial worlds and economics, part ii, *Journal of Evolutionary Economics*, 3, 89–107.

Parisi D., Cecconi F. and Nolfi S. (1990) Econets: Neural networks that learn in an environment, *Network*, 2, 149–168.

Rumelhart D.E. and McClelland J.L. (1986) *Parallel Distributed Processing: Exploration in the Microstructure of Cognition, Vol. 1: Foundations*, MIT Press, Cambridge, MA.

Terna P. (1993) Artificial interacting agents for stock market experiments: the cross-target method, in: *Proceedings of ICANN 93,* (eds S. Gielen and B. Kappen), Springer-Verlag, Berlin.

Wasserman P.D. (1989) *Neural Computing: Theory and Practice*, Van Nostrand Reinhold, New York, NY.