

# **Neural Networks for Economic and Financial Modelling**

**Andrea Beltratti, Sergio Margarita  
and Pietro Terna**  
University of Turin, Italy



**INTERNATIONAL THOMSON COMPUTER PRESS**  
**I T P** An International Thomson Publishing Company

---

London • Bonn • Boston • Johannesburg • Madrid • Melbourne • Mexico City • New York • Paris  
Singapore • Tokyo • Toronto • Albany, NY • Belmont, CA • Cincinnati, OH • Detroit, MI

# 4

## The cross-target method

### Outline of the chapter

In this chapter we introduce an ANN technique developed to build AAAs without using a priori economic rules. Section 4.1 deals with the meaning of developing such models and introduces the structure of the method. The core of the technique is explained first and then some sophistications are introduced, the possibility of tuning AAAs' behaviour.

In section 4.2 the method is gradually applied, from simple to complex cases, also allowing interaction and imitation among agents. Experiments start, in section 4.2.1, with a simple analysis of motion behaviour, made by agents foraging for food, learning while acting and relearning from their experience. Learning by imitation is then introduced, developing agents able to do a specific task and studying their internal structures.

In sections 4.2.2 agents are supposed to behave in an economic context, starting with one-agent models describing a consumer reacting to price changes and a risk-averse stock market agent: the capabilities of the two agents come from micro-mechanisms developed by this method. The framework of the stock market model is described in section 4.2.3.

In section 4.2.4 we introduce external objectives in one-agent models; in sections 4.2.5 and 4.2.6 we use models of interacting agents, with two different populations and introducing both external objectives and external proposals: we will see high rationality behaviour emerging from randomness and imitation.

### 4.1 The cross-target method

We must, first of all, explain the name and the idea that exists behind the name. The name 'cross-targets' (CTs) comes from the technique used to figure out targets in a class of models founded upon artificial adaptive agents whose main characteristic is developing some kind of internal consistency. Our agents are developed by neural networks; we can specify two types of outputs of the ANN upon which we build each AAA: (1) actions to be performed and (2) evaluations of the effects of those actions. With our technique, both the targets necessary to train the network from the point of view of the actions, and those connected with the effects, are built in a crossed way. The former are built in a consistent way with the outputs of the network concerning the guesses of the effects, in order to develop the capability of deciding actions close to the expected results. The latter are similarly built in a consistent way with the outputs of the

network concerning the guesses of the actions; here, we will improve the agent's capability to estimate the effects emerging from the actions that the agent itself is deciding.

What is the idea behind such a mechanism? Our hypothesis is that an agent acting in an economic environment must develop and adapt his or her capability to evaluate in a coherent way what he or she has to do in order to obtain a specific result or to appreciate the consequences of a specific action. The same is true if the agent is interacting with other agents of the same population or of other populations.

Beyond consistency, we can add other characteristics, mainly to obtain the possibility of tuning agents to make experiments.

#### 4.1.1 Developing models without a priori rules

The CTs technique attributes a central role to learning mechanisms and can be applied without introducing, either explicitly or implicitly, economic rules in order to influence or to characterize agents' behaviour. The aim is to conduct economic experiments without the influence of any prior economic hypothesis (Terna, 1991, 1992a, 1992b, 1993a, 1993b).

We will see that CTs can reproduce economic subjects' behaviour, often in internal 'ingenuous' ways, but externally with apparently complex results. Mainly, economic behaviour, simple or complex, can appear directly as a by-product of developing consistency between (1) decisions about actions and (2) guesses about effects. For an external observer, this kind of AAA is apparently operating with goals and plans. Obviously, it has no such symbolic entities, which are inventions of the observer. The similarity that we recall here is that the observations and analyses about real-world agents' behaviour can suffer from the same bias.

Certainly, the CT algorithm introduced here is not the only way of dealing with AAAs, but it represents an interesting tool because it doesn't require injections of rules, optimizing behaviour or planning capabilities, but only a limited computational ability: that necessary to take simple decisions and to compare guesses with results.

#### 4.1.2 The cross-target technique

Following other authors' work (Parisi *et al.*, 1990), we choose the neural approach to develop CTs mostly as a consequence of the intrinsic adaptive capabilities of neural functions. Here we will use feed-forward multilayer ones.

Targets in learning process are: (1) on one side, the actual effects of the actions made by the simulated subject; (2) on the other side, the actions needed to match guessed effects.

Figure 4.1 describes an AAA's learning and behaviour in a CT scheme. The AAA has to produce guesses about both its own actions and their effects, on the basis of an information set (the input elements are  $I_1, \dots, I_k$ ). Actual

effects are estimated through the guessed actions, also taking into account the consequences from the interaction among agents, if any; the results are used to train the mechanism that guesses the effects. Actions that we measure to be necessary to match guessed effects are, on the contrary, employed to train the decision mechanism about actions. In the last case we have to use inverse rules, even though some problems arise when the inverse is indeterminate.

The CTs method, introduced to develop economic subjects' autonomous behaviour, can also be interpreted as a general algorithm useful for building behavioural models without using constrained or unconstrained optimization techniques. The kernel of the method, conveniently based upon ANNs (but it could also be conceivable with the aid of other mathematical tools), is learning by guessing and doing: the control capabilities of the subject can be developed without defining either goals or maximizing objectives.

The CT method can appear to be related to the Temporal Difference (TD) Learning of Barto and Sutton (Sutton, 1988; Tesauro, 1992), which learns from the differences between temporally successive predictions – or action outcomes – of the system, having a final target perfectly known at the end of the run. In the TD method we have a special and powerful case of true supervised learning, where an external teacher can suggest

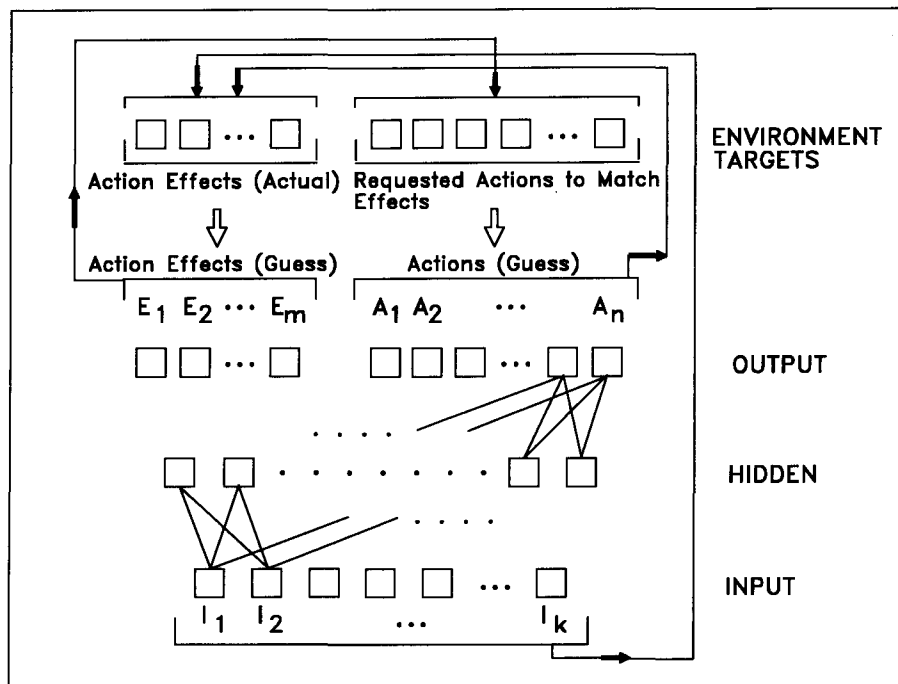


Figure 4.1 The cross-target construction.

correct target values. Also TD, like CT, addresses the issue of consistent learning, but with delayed feedback founded upon a true target value; CT uses immediate tentative targets, self-generated and never corrected by an external teacher. The aim of CT is in effect to generate time paths for relevant variables, without any final or intermediate externally known objective, operating only with simple rules to adapt both behaviour and predictions.

Now we may introduce some technical explanations of CTs, with the aid of the general scheme of Figure 4.1, observing that (1) the inputs of the model are mainly data coming from the environment or from other agents' behaviour, (2) they can be dependent on or independent of the previous actions of the simulated artificial subject, and (3) targets are known only when actions take place.

The CT algorithm is a learning and acting one: action is necessary to produce the information from which we can construct targets to train the ANN that simulates the subject. A training set cannot be constructed here in the usual way because the rules linking the inputs and outputs of the ANN have 'to be discovered' by the experiments led by AAAs. Learning and acting take place in four steps each 'day'; a day is the sum of the four steps required to perform a full cycle of estimation of outputs and of back-propagation of errors, correcting the neural network weights. Initial weights are randomized in a given range.

Looking at Figure 4.1, the four steps can be introduced in the following sequence.

1. Outputs of the ANN: The actions to be accomplished, reported on the right side of Figure 4.1, and the effects of these actions, reported on the left side of the same figure, are guessed following inputs and network weights.
2. Targets for the left side of the network: The targets for the effects supposed to arise from actions, as guessed in the left side of the output layer in Figure 4.1, are figured out by the independently guessed actions. In this way, guesses about effects become closer to the true consequences of actual actions.
3. Targets for the right side of the network: The differences measured in step 2 between targets and ANN outputs on the effect side can be inversely interpreted as starting points for action modifications, to match the guessed effects. So they are used to build the targets for the mechanism that guesses the actions. Since the inverses of the formulas shown below are often undefined, corrections are shared randomly among all the targets to be constructed; besides, when several corrections concern a target, only the one with the largest module is chosen. In this way, we would like to imitate the actual behaviour of a subject requested to obey several independent and inconsistent commands: probably the most imperative, here the largest value, will be followed.

4. Back-propagation: Learning takes place, correcting weights in order to obtain guessed effects closer to the consequences of guessed actions, and guessed actions more consistent with guessed effects. Thus, we have two learning processes, both based upon the guesses of the elements of the opposite side of the network.

This double-sided process of adaptation, with interaction among agents and long-term learning introduced at the end of this section, ensures the emergence of non-trivial self-developed behaviour, from the point of view of the time paths of the values generated by the outcomes of the agents.

We can now explain in a formal way the acting and learning algorithm of CTs, introducing a generic effect  $E_1$  arising from two actions, named  $A_1$  and  $A_2$ . The target for the effect is:

$$E_1' = f(A_1, A_2) \quad (4.1)$$

where  $f$  is a definition, linking actions to effects on an accounting basis.

Our aim here is to obtain an output  $E_1$  (the guess made by the network) closer to  $E_1'$ , which is the correct measure of the effect of actions  $A_1$  and  $A_2$ . The error related to  $E_1$  is:

$$e = E_1' - E_1$$

or, by convention in ANN development, one half of the square of  $E_1' - E_1$ . To minimize the error, we back-propagate it through network weights.

Our aim now is to find actions, as outputs of our network, more consistent with the outputs produced by the effect side. So we have to correct  $A_1$  and  $A_2$  to make them closer to  $A_1'$  and  $A_2'$ , which are actions consistent with the output  $E_1$ . We cannot figure out the targets for  $A_1$  and  $A_2$  separately. From equation 4.1 we have:

$$A_1 = g_1(E_1', A_2) \quad (4.2)$$

$$A_2 = g_2(E_1', A_1) \quad (4.3)$$

Choosing a random value  $\tau_1$  from a random uniform distribution whose support is the closed interval  $[0, 1]$  and setting  $\tau_2 = 1 - \tau_1$ , from equations 4.2 and 4.3 we obtain:

$$A_1' = g_1(E_1' - e \cdot \tau_1, A_2) \quad (4.4)$$

$$A_2' = g_2(E_1' - e \tau_2, A_1) \quad (4.5)$$

Functions  $g_1$  and  $g_2$ , being obtained from definitions that link actions to effects mainly on an accounting basis, usually have linear specifications; so equations 4.4 and 4.5 generally give solutions that are globally consistent. The errors to be minimized are:

$$a_1 = A_1' - A_1$$

$$a_2 = A_2' - A_2$$

Equations 4.4 and 4.5 would be unacceptable as inversions of true dynamic functions, but they are used here as a simplifying tool (mainly for the presence of random separation obtained by  $\tau_1$  and  $\tau_2$  values), always to generate time paths for variables, without a priori or external suggestions.

When the actions determine multiple effects, they are included in multiple definitions of effects. So, those actions will be affected by several corrections; as reported in 3 above, only the largest absolute value is chosen.

Input and target variability, generated in both deterministic and random ways, is required to ensure the economic plausibility of the experiments, but is also necessary to ensure that the outputs and the targets of the ANN change. Lacking such variability, on the basis of the initial random weights of the network and following CTs, in most cases all outputs would be frozen at about 0.5, with perfect but merely apparent learning results.

With the proper variability, we repeat for a given number of cycles (days) the four steps introduced to describe Figure 4.1. The learning following the fourth step of each day gives a sort of local adaptation to the changes of the environment.

Analysing the changes in the weights during the process we can show that the matrix of weights linking input to hidden elements has little or no changes, while the matrix of weights from hidden to output layer changes in a relevant way. Only hidden-output weight changes determine the continuous adaptations of ANN responses to the environment modifications, as the output values of hidden layer elements stay almost constant.

This situation is not caused by linear separability of the problem (see section 3.3.1 where the XOR function represents a classic case of a linear inseparable problem), as we can verify by applying the long-term learning explained above, but it is the consequence of very small changes in targets (generated by the CT method) and of a reduced number of learning cycles.

The resulting network is certainly under-trained (see also section 3.1.5). Consequently, the simulated economic agent develops a local ability to make decisions, but it has difficulties in coping with large environmental changes. This case resembles an actual consumer who is not able to determine his/her demand if prices change dramatically.

This is short-term learning as opposed to long-term learning, in analogy with the psychologists' distinction between short- and long-term memory: (1) the learning and acting phase produces neural agents continuously modifying their weights, in a local way, to adapt to environmental changes; (2) *ex post*, relearning the weights of the neural networks engaged in the experiment and using as data the historical records of the events that have occurred, we can obtain ANNs also able to react, without subsequent learning, to major changes in environmental conditions.

The second type of learning can also take place periodically upon a short segment of data (for example, every fifty days referring to the previous one hundred days – or upon a sample of the full historical data set, always with satisfying results).

We will introduce two examples of long-term learning: the first in section 4.2.1, related to a little experiment of motion, and the second at the end of section 4.2.6, to develop a long-term trained agent behaving in the stock market. From the structure of such an agent we will extract the rules implicitly developed, as in section 3.3.5.

### 4.1.3 Cross-targets, external goals and external proposals

With CTs we obtain agents that behave on the basis of the development of consistency among guesses about their actions and related effects. This sort of consistency is sufficient to obtain self-developed micro-mechanisms, as reported in the connectionist robot literature (Connell, 1990), which are very simple, but sufficient to characterize realistic economic behaviour.

To improve the capabilities of CTs in the development of economic experiments and to offer to the experimenter useful tools directed to the tuning of the agents' behaviour, some improvements on the method can be introduced: (1) the use of external objectives (EOs), to direct the guesses of the effects, and (2) the use of external proposals (EPs) to influence the guesses about actions. EPs and EOs are external targets: the EO replaces the cross-target to train the specific output processing element, but the original CT target survives for the crossed training of the actions; any EP represents one of the multiple targets – from which only the largest is chosen – used to train the side of the model that guesses the actions.

Simple examples of EOs are the following: for a household, obtaining a good match between food purchase and food requirement, increasing money at a constant rate or not working more than a maximum, etc.; for an entrepreneur, maintaining a constant difference between costs and benefits, etc.

EPs suggest actions: the source of the suggestions can also be randomness, which is for example sufficient in section 4.2.6 to explain in a radical way what apparently could be the effects of reason. Another kind of EP is imitation, which is a powerful means of exchanging information among agents; imitation is well known to sociologists, but it is almost unknown in economic models, where agents exchange information mainly by prices.

In conclusion, EOs are targets used to train the side of the ANN that guesses the effects, while EPs represent one of the multiple targets used to train the side of the model that guesses the actions.

## 4.2 Experiments with cross-targets

We now introduce several experiments developed with the CT method, mainly in the field of economics. Before doing so, however, we will present two analyses of artificial behaviour: (1) subjects foraging for food on a plane, with imitation of other subjects' actions; (2) subjects learning to

solve a problem by imitation. The role of these experiments is to introduce some technique capabilities in our work, such as learning repetition and imitation, starting with very simple situations in a stylized environment.

In the examples, we will look both for emerging aggregate behaviour and try to discover the internal mechanisms, often micro-mechanisms, generating the behaviour of each agent. Some very interesting results in the connectionist literature about robots (e.g. Beer, 1990; Connell, 1990; Mel, 1990) stress the importance of uncorrelated mechanisms governing independent single minimal aspects of the apparent global action of a connectionist robot, whose complexity is the consequence of those independent functions.

With Connell (1990), we have: 'Like Simon's metaphorical ant (Simon, 1969) the complexity of a creature's action is not necessarily due to deep cognitive introspection, but rather to the complexity of the environment it lives in.' R.A. Brooks, introducing Connell's book (Connell, 1990): '... has shown how a robot can appear to an observer to be successfully carrying out high level tasks, seemingly with goals and motivations, persistence and plans. In fact, as we read his description we find that the robot has no such entities. They are inventions of the observer.'

For an observer, the behaviour produced by the AAAs of our models also seems to match the actions of realistic economic actors, appearing to be rational, as if they operated with goals and plans. Obviously, our agents have no such symbolic entities, which exist only in the observer's mind. The assumption is that the observations of economists about real-world agents' behaviour could suffer from the same bias.

Finally, we stress the following aspect: operating only on the basis of the development of agents' consistency, CTs develop behaviour without optimization, always explaining actions and their effects in a simple and 'parsimonious' way.

#### 4.2.1 Simple preliminary experiments

We now introduce the two preliminary experiments proposed in section 4.2, which are useful both for explaining the role of the repeated *ex-post* learning proposed at the end of sections 3.1.5 and 4.1.2, and for presenting the capabilities of random or imitative behaviour in explaining complex situations.

##### (a) *Agents foraging for food*

The experiment on the motion of agents foraging for food is built upon the following scheme.

On a plane with  $(x, y)$  coordinates, the subject is initially in  $(10, 10)$  while the food is fixed in  $(0, 0)$ . The ANN simulating the subject has as inputs (which are the targets founded in the previous cycle):

- $X(t-1)$ , the position in the  $x$  direction at time  $t-1$ ;  
 $Y(t-1)$ , the position in the  $y$  direction at time  $t-1$ ;  
 $dX(t-1)$ , the step in the directions  $x$ , at time  $t-1$  (bounded in the range  $\pm 1$ );  
 $dY(t-1)$ , the step in the directions  $y$ , at time  $t-1$  (bounded in the range  $\pm 1$ ).

Using CT terminology, the same ANN produces as outputs two guesses about effects and two guesses about actions.

Guesses about effects are:

$$X(t), Y(t)$$

Guesses about actions are  $dX(t)$  and  $dY(t)$ , all with the same meaning of the input values.

Targets are:

$$\begin{aligned}
 X'(t) &= X(t-1) + dX(t); \\
 Y'(t) &= Y(t-1) + dY(t); \\
 dX'(t) &= dX(t) + X(t) - X'(t), \text{ which is the correct action to match } X(t), \\
 &\quad \text{increasing } dX(t) \text{ if } X(t) - X'(t) > 0 \text{ and decreasing it in the opposite case;} \\
 dY'(t) &= dY(t) + Y(t) - Y'(t), \text{ as before.}
 \end{aligned}$$

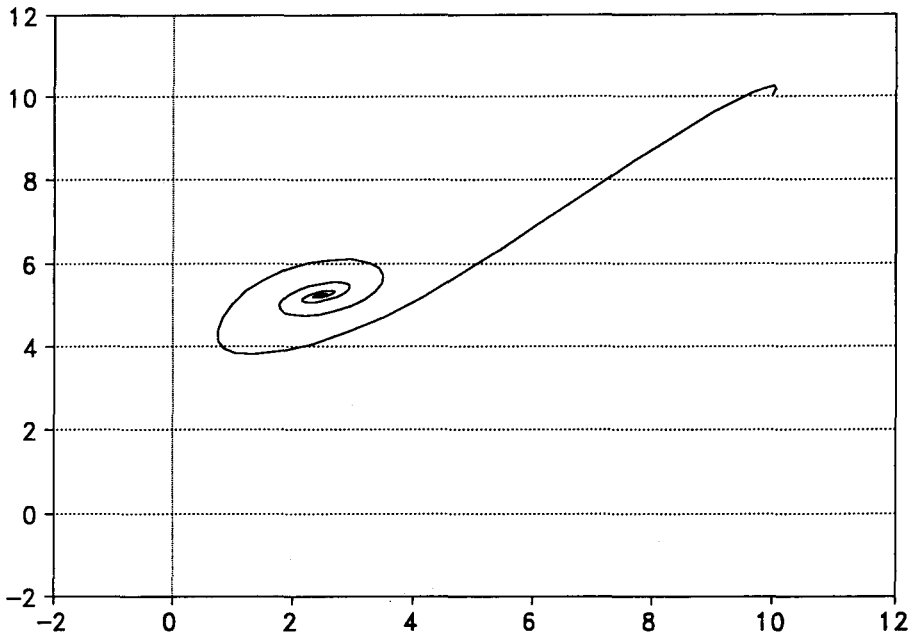


Figure 4.2 *Moving towards food, without EO.*

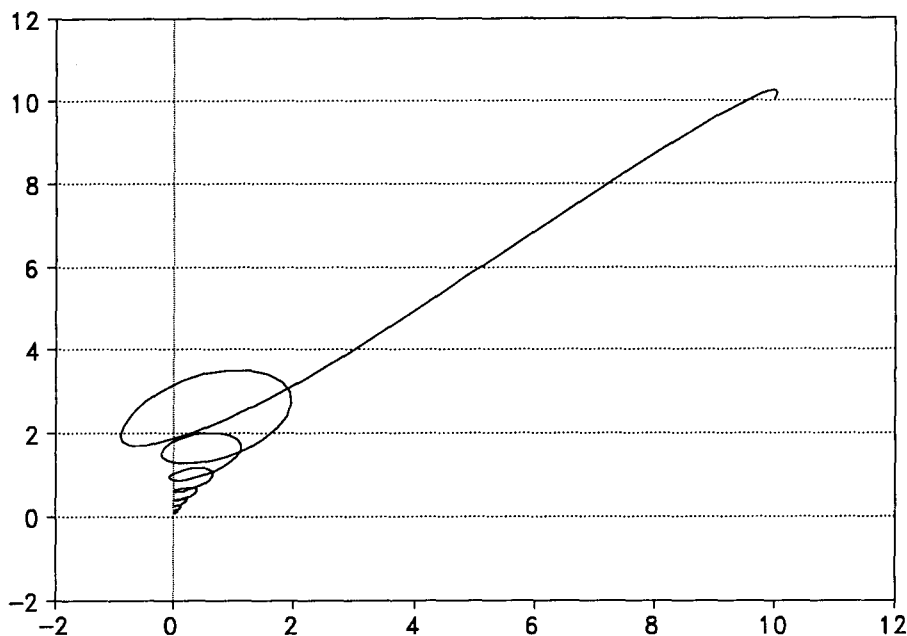


Figure 4.3 *Moving towards food, with EO.*

Positions  $X(t)$  and  $Y(t)$  also have the meaning of measuring the distance of the artificial subject from the food (the distance is evaluated by employing rectangular coordinates). In Figure 4.2 we report the position of the agent in 200 cycles of acting and learning; in each cycle, representing one day or step, the movement is determined by the action outputs of the ANN. Each position is linked with the succeeding one, plotting a continuous line. The agent goes towards the food on the basis of a simple implicit mechanism, which also explains the locking situation in the middle of the path.

The mechanism works in the following way: at the beginning of the experiment the ANN produces random outputs, but in a small interval around the central value between the minimum and maximum ones. This effect is always present and is easily explained by considering the consequence of the initial random choice of the weights, which gives on average a null sum of the inputs of the sigmoidal transformation. In the case of the logistic functions, that input gives an output about 0.5, which also corresponds to the simple mean between minimum and maximum values. The initial guesses about the effects of the movement give estimated positions around the point  $(0, 0)$ , where the food is placed, with large variability; this result emerges from the range  $[-20, 20]$  assigned to the spatial coordinates. CTs immediately correct these wrong estimates, but they also correct the guesses about actions (the movements), to develop

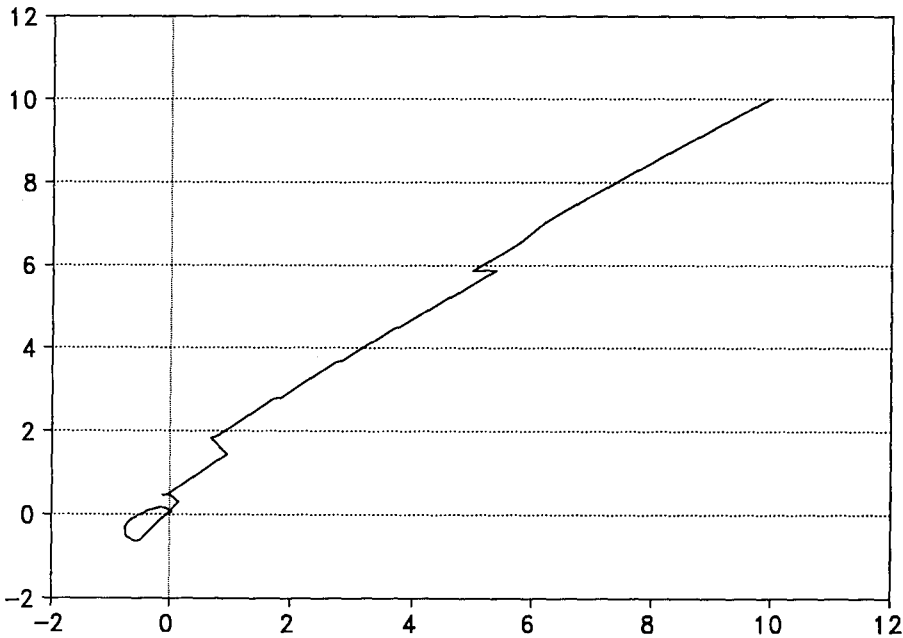


Figure 4.4 *Moving towards food, following an ANN with weight coming from repeated learning.*

their consistency with the guesses of effects. So, the artificial agent moves in the correct direction, but the process rapidly goes into a locking situation, with mutual consistency between effects and actions.

If we now impose an EO on the side of the effects, namely the target of reducing the distance from the food, in each cycle, to 75% of the distance in the previous cycle (the measure of the distance from the current position to the point (0, 0) is directly obtained by the rectangular coordinates of the position), the food is easily gained, as reported in Figure 4.3. We underline that no suggestion is introduced about the direction of the movement.

In Figure 4.4 we present the case of Figure 4.3 again, but with an ANN whose weights come from a repeated learning process of 200 000 cycles. The learning process is applied upon the full 200 cycles of historical data of a single acting and learning run, in sequential order, with a 0.9 momentum coefficient.

The simulated agent reaches the food in a few steps (fewer than one hundred), going directly towards it, despite some uncertainty.

We can now produce two GWMs (see section 3.3.5), the first related to the agent described in Figure 4.3, representing the effect of short-term learning (see section 4.1.2), the second related to the agent of Figure 4.4, with the effects of long-term learning.

The GWM of the agent of Figure 4.3 is the following; the rows represent the input values, that is, the targets of the previous cycle, while the columns contain the output values.

|          | Outputs | X(t)    | Y(t)    | dX(t)   | dY(t)   |
|----------|---------|---------|---------|---------|---------|
| Inputs   |         |         |         |         |         |
| X'(t-1)  |         | -0.1035 | -0.0171 | 0.0330  | 0.0017  |
| Y'(t-1)  |         | -0.0406 | -0.0308 | -0.0449 | -0.0213 |
| dX'(t-1) |         | 0.0381  | 0.0245  | 0.0144  | -0.0125 |
| dY'(t-1) |         | -0.0140 | -0.0227 | 0.1088  | 0.0290  |

Clearly, with short-term learning, we have a limited amount of weight correction, here only 200, determining soft links between each input and each output; our ANN rapidly reacts in each cycle or day to the targets coming from the CT mechanism mainly because, by construction, they are never very different from the output values; so the adaptation is obtained by changing the weights slightly, mainly those from the hidden layer and the output layer. We can observe this kind of soft ANN structure in the GWM matrix.

We can also observe that other adaptive functions or algorithms could be used, such as classifier systems, but with a lack of generality and without the simple passage from short-to-long term learning that we are considering here.

Going to the long term, with the agent of Figure 4.4, we have the following GWM, showing hard input-output links.

|          | Outputs | X(t)      | Y(t)     | dX(t)    | dY(t)   |
|----------|---------|-----------|----------|----------|---------|
| Inputs   |         |           |          |          |         |
| X'(t-1)  |         | - 16.9111 | - 5.7756 | 7.2730   | 40.0658 |
| Y'(t-1)  |         | -8.3577   | -1.6650  | 6.1658   | 20.0175 |
| dX'(t-1) |         | -75.9463  | -46.7457 | 182.9647 | 81.7869 |
| dY'(t-1) |         | -71.1736  | -46.5281 | 99.4387  | 42.3390 |

To interpret this second GWM matrix, we will extract the implicit rules developed by long-term learning. However, first of all we observe its values: guesses about the position on the plane, evaluating the effect of the actions, are influenced mainly by the action inputs. The reasons for this result, which seems to be counter-intuitive, come from the strong links joining input and output values about actions. The actions are strongly influenced by the previous ones and so guesses about the position on the plane can be founded upon previous actions, while actually they depend on the current ones. As an interpretation we can suggest that the subject has

memorized completely the sequence of steps necessary to reach the food and acts to reproduce that sequence, regardless of the position in each day or after each step. Is this global interpretation confirmed by a technical analysis?

Considering the ANN weights after the relearning process, that is, the ANN upon which the agent of Figure 4.4 is founded, we analyse the derivatives between each output and all the inputs. For each output variable we consider: (1) derivatives calculated for a hypothetical mean point of input, where all variables assume their mean value (in the ANN metrics, all the values are 0.5). These derivatives are called Mean Point Derivatives (MPDs); (2) derivatives calculated for each pattern. In order to facilitate the analysis of these results, we calculate the mean and the standard deviation of the derivatives of each pattern. These derivatives are called Mean Derivatives (MDs).

#### Output variable $X(t)$

| inputs  | $X'(t-1)$ | $Y'(t-1)$ | $dX'(t-1)$ | $dY'(t-1)$ |
|---------|-----------|-----------|------------|------------|
| MPD     | 0.024     | 0.008     | 0.009      | 0.028      |
| MD mean | -0.016    | -0.009    | -0.144     | -0.127     |
| MD stdv | 0.146     | 0.063     | 0.530      | 0.558      |

#### Output variable $Y(t)$

| inputs  | $X'(t-1)$ | $Y'(t-1)$ | $dX'(t-1)$ | $dY'(t-1)$ |
|---------|-----------|-----------|------------|------------|
| MPD     | 0.009     | 0.003     | 0.003      | 0.010      |
| MD mean | -0.014    | -0.004    | -0.128     | -0.127     |
| MD stdv | 0.088     | 0.027     | 0.461      | 0.494      |

#### Output variable $dX(t)$

| inputs  | $X'(t-1)$ | $Y'(t-1)$ | $dX'(t-1)$ | $dY'(t-1)$ |
|---------|-----------|-----------|------------|------------|
| MPD     | -0.071    | -0.014    | 0.064      | -0.127     |
| MD mean | -0.033    | 0.000     | 0.315      | 0.092      |
| MD stdv | 0.190     | 0.060     | 0.831      | 0.842      |

#### Output variable $dY(t)$

| inputs  | $X'(t-1)$ | $Y'(t-1)$ | $dX'(t-1)$ | $dY'(t-1)$ |
|---------|-----------|-----------|------------|------------|
| MPD     | 0.036     | 0.019     | 0.088      | 0.003      |
| MD mean | 0.080     | 0.040     | 0.140      | 0.041      |
| MD stdv | 0.176     | 0.081     | 0.202      | 0.184      |

To read these data we observe both the value of the derivative at the average point and the standard deviation of the average of pattern derivatives: a

large standard deviation can signal the existence of situations where a modification of the input value has an important role, even if the mean does not appear to be relevant. We confirm here the previous interpretation: all the output variables are mainly affected by the third and fourth input values; we can see however that a location coordinate, that is,  $X'(t-1)$ , also has a role. The sequence of steps is self-determined, but also controlled by a spatial coordinate. We can also investigate the implicit rules developed by the training process, following the method proposed in section 3.3.5.

In a CT exercise, the values of  $R^2$  are not interesting, since the targets can be interpreted as values suggesting corrections to develop consistency, and not values to be approximated. To apply that method here, we have to define the maximum  $R^2$  result artificially, in order to get the  $R^2$  values obtained by introducing only subsets of the input values into the ANN; as in section 3.3.5, the excluded inputs are set to their intermediate value between minimum and maximum. The artificial  $R^2$  is set to  $R^2 = 1$ , corresponding to the result that we obtain by applying the ANN to a set of data in which the targets are exactly the output values of the ANN after the relearning process. (Alternatively, we could use the outputs vs the targets of the relearning task, in which case  $R^2 < 1$ .)

From the GWM values and MD analysis we can adopt the following sequence in the additive rule construction, for all output variables:  $dX(t-1)$ ,  $dY(t-1)$ ,  $X(t-1)$ ,  $Y(t-1)$ .

The  $X(t)$  reconstruction with limited input gives the following  $R^2$  results:

| Variables in input                  | $R^2$  |
|-------------------------------------|--------|
| $dX'(t-1)$                          | -0.357 |
| $dX'(t-1)$ , $dY'(t-1)$             | 0.121  |
| $dX'(t-1)$ , $dY'(t-1)$ , $X'(t-1)$ | 0.940  |

The  $Y(t)$  reconstruction gives:

| Variables in input                  | $R^2$  |
|-------------------------------------|--------|
| $dX'(t-1)$                          | -0.304 |
| $dX'(t-1)$ , $dY'(t-1)$             | 0.588  |
| $dX'(t-1)$ , $dY'(t-1)$ , $X'(t-1)$ | 0.980  |

The results show that data related to the previous movements are essential in explaining guesses about the agent's current position, but only if they are jointly considered. Furthermore, we have a near perfect ANN performance when we introduce a spatial coordinate (the  $X$ ).

The  $dX(t)$  reconstruction gives:

| Variables in input            | $R^2$ |
|-------------------------------|-------|
| $dX'(t-1)$                    | 0.758 |
| $dX'(t-1), dY'(t-1)$          | 0.849 |
| $dX'(t-1), dY'(t-1), X'(t-1)$ | 0.985 |

The  $dY(t)$  reconstruction gives:

| Variables in input            | $R^2$ |
|-------------------------------|-------|
| $dX'(t-1)$                    | 0.915 |
| $dX'(t-1), dY'(t-1)$          | 0.696 |
| $dX'(t-1), dY'(t-1), X'(t-1)$ | 0.959 |

These results point to the specific importance of the previous steps and, in the case of  $dY(t)$ , of  $dX'(t-1)$ . Therefore the movement in the  $x$  direction is the key in interpreting the behaviour of our AAA.

Certainly, we are discovering here a sophisticated agent structure with interesting characteristics, reliable, but not easy to foresee before the experiment. This result underlines the importance of the experimental approach: we see complex results emerging from simple structures, built in a parsimonious way, without optimization capabilities. We are developing simple

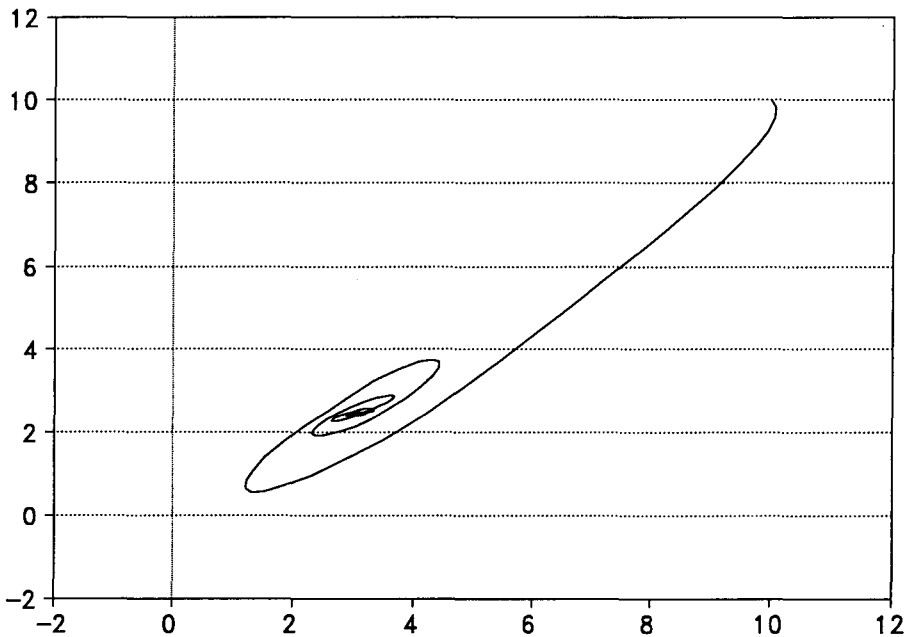


Figure 4.5 *Moving towards food, with EPs coming from imitation.*

agents, or mechanisms, that are capable of achieving a goal. Again, is that also the explanation of the behaviour of real agents? To understand more about this issue, consider what happens if we introduce an imitative habit in the model of the agent foraging for food.

The last analysis of this model is developing a population of similar agents, all acting without an EO, like the one in Figure 4.2, but with an EP (see section 4.1.3). In Figure 4.5 we have the itinerary towards the food of an agent which has as an EP the instruction to imitate the current movement, that is, the step in the x and y directions, of the other agents.

The imitative mechanism works as follows: the agent whose action is to be imitated is chosen randomly; the imitation occurs only if the action of that agent is greater in absolute value than that of the imitating agent. In the case of imitation, a random factor in the range  $[0,2]$  amplifies or mitigates the two measures of the action (which is a step defined in the x and y directions).

The artificial agents, without an EO, come close to the food. The reason is that the choice of imitating another agent reproduces a situation of inconsistency among CTs, from which – as seen before – the implicit simple mechanism driving the agent restarts, avoiding locking situations. Also note that if the imitated action is directed towards the food, the consequence is immediately convenient; if the imitated action goes in the opposite direction, the process corrects the error rapidly, always strength-

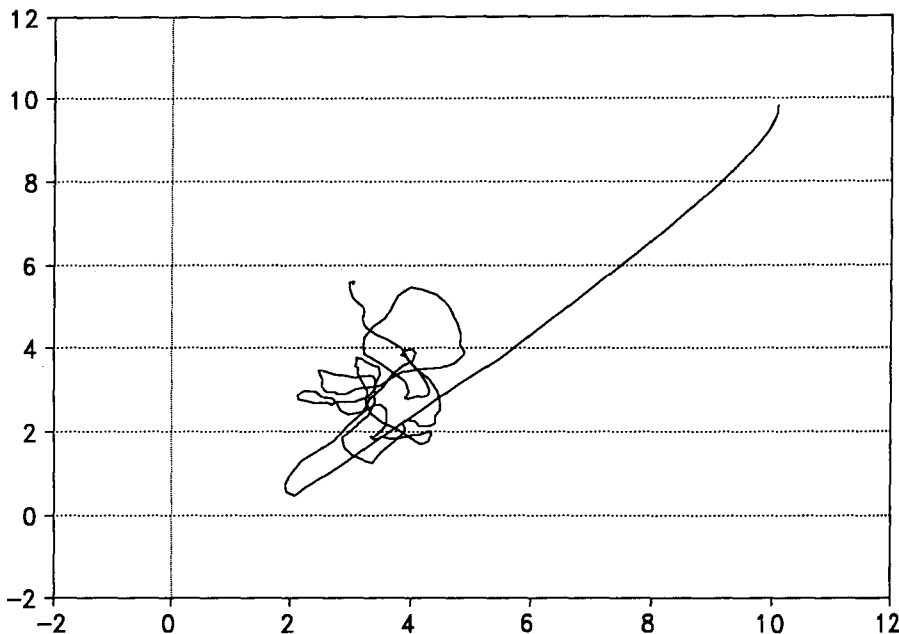


Figure 4.6 *Moving towards food, with random EPs.*

ening the global convergence process. In addition, when all agents are close to the food, all act with small steps and the imitative process loses importance.

If the EP is randomly generated, as in the case of Figure 4.6, the effect of avoiding locking situations is replicated, but the random behaviour prevails: it is more difficult to recognize a plan in this kind of artificial crazy agent, which is in any case capable of going close to the food.

Finally, note that in these experiments imitation and chance seem to have the same effect, but imitation can develop more subtle situations, as one might expect.

### *Learning to solve problems by imitation*

The second experiment with imitation involves agents learning to solve a problem. Here we have several ANNs doing numerical products of their two inputs, which belong to a given range and are equal for all the ANNs. Some of them receive correct teaching, with a target obtained by multiplying the two inputs; the others carry out their imitative behaviour, using as their target the output of an ANN chosen randomly. The imitated ANN can be one with correct teaching or one that imitates others.

Here the meaning of imitation is more relevant than that implicit in the previous experiment, where imitation is mainly a source of noise. Surprisingly, apparent rationality emerges from noise: this is the power of experimental techniques. In this second case, the surprises are the capabilities emerging in the imitative ANNs and of the strong similarities that we can measure in the GWM of the imitating ANNs.

In a first run we have ten ANNs, one with the correct target (this ANN is named T) and the other nine ANNs (named I) imitating randomly. In a second run, there are ten T ANNs and 90 I ones.

In the first run, the GWM of the one ANN of type T ( $i_1$  and  $i_2$  are the input values,  $o_1$  the output one) is:

|       |       |
|-------|-------|
|       | $o_1$ |
| $i_1$ | 18.45 |
| $i_2$ | 19.83 |

The GWMs of the nine ANNs of type I are presented as mean values, with standard deviations in brackets; the matrices composing the GWMs are different for each ANN, while GWMs (as a phenotype) are strongly similar:

|       |       |        |
|-------|-------|--------|
|       | $o_1$ |        |
| $i_1$ | 10.78 | (0.32) |
| $i_2$ | 10.68 | (0.30) |

In the second run, the GWMs of the ten ANNs of type T show high variability; we summarize them in the following data:

|    |       |        |
|----|-------|--------|
|    | o1    |        |
| i1 | 17.05 | (2.89) |
| i2 | 18.31 | (2.22) |

The 90 ANNs of type I are again very similar in terms of GWMs:

|    |       |        |
|----|-------|--------|
|    | o1    |        |
| i1 | 10.99 | (0.42) |
| i2 | 10.90 | (0.39) |

The result in terms of similarity can be explained on the basis of the following consideration: the I ANNs evolve towards a 'mean' of the different existing structures, while the T ANNs maintain their structural differences untouched, producing analogous results. We therefore discover the influence of the environment in the development of both agents' capabilities and their structures.

#### 4.2.2 Economic regularities in one-agent models

Now we introduce the theme of the economic regularities arising from CT models, when used in an economic situation. Starting with the agent's actions in one-agent models, the following examples of simple mechanisms directly developed by CTs will be introduced: (1) the capability of reacting to price changes; (2) a characterization of portfolio decisions.

##### *Reacting to price changes*

The first simulated agent is making two independent guesses on the basis of a single input, the price  $p_a$ , which changes exogenously by means of a sin function plus a random noise. Guesses are: the quantity  $q_a$  of the good  $C_a$  to be acquired; the expenditure  $x$ . Cross-targets are easily established here with:

$$x' = p_a \cdot q_a$$

$$q_a' = q_a + (x - x')/p_a = x/p_a$$

where, on the one hand,  $x'$  is the true expenditure consistent with the decision  $q_a$ ; on the other hand,  $q_a'$  is the correct decision necessary to acquire the quantity of A consistent with the guess  $x$ . If  $x' > x$ , the guess of  $q_a$  must be increased by the amount  $(x - x')/p$  and vice versa.

In Figure 4.7 we have prices and guessed quantities ( $q_a\_g$ ) for a thousand days of learning and acting; in the figure, we have chosen a sample of one day every ten days. The simulated subject has self-developed its

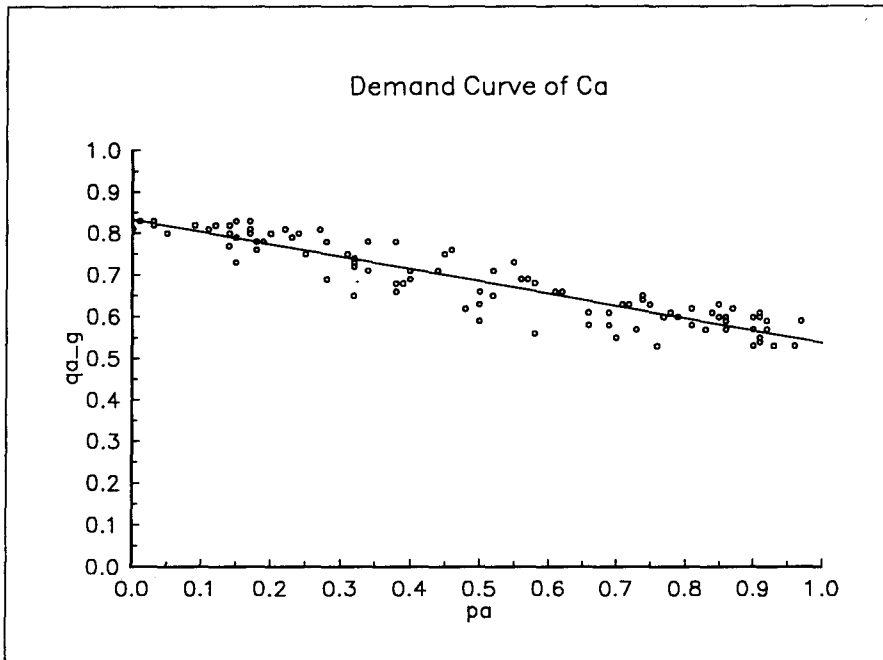


Figure 4.7 *Experimental reactions to different price levels.*

capability of reacting to price changes, smoothing the effects on its level of expense by quantity adjustment: what is emerging is a kind of demand curve without optimization. What is surprising is that the mechanism, albeit very simple, is autonomously and endogenously developed by the model.

Things go as follows: when the price changes, the cross-target mechanism determines two symmetrical corrections. Let us suppose that the price is augmented. The neural network outcomes, at the beginning of the learning process, are very conservative. Consequently, in short-term runs (see section 4.1.2 for details of short-vs long-term learning), the guesses about the expenditure (the effect) and the quantity (the action) adapt themselves very slowly to the changing input. The expenditure evaluation will therefore be underestimated and corrected by increasing it; vice versa, the quantity, which has to be consistent with the guess about the expense, which is underestimated, has to be corrected by decreasing it. We thus have two symmetrical corrections, less relevant than those that we could obtain separately by acting upon only one of them, keeping the price or the quantity constant. Realistically, the results are neither strictly exact nor deterministic. Is this a demand curve? In a strict sense it is not, but it describes common behaviour very well!

*Agents in a stock market*

We introduce an agent behaving in a stock market, starting with neither EO nor EP. The general framework of the model is described in section 4.2.3.

Here we summarize the model structure, emphasizing the rules related to the behaviour of a single agent model. Rules related to the interaction between populations of agents will be introduced in section 4.2.5, which also describes the market structure.

The AAA decides actions, as guesses about actions that it will make: there are two decisions, both related to the quantities of shares to buy  $A_t$  and to sell  $V_t$  (the double contract could be simplified considering only the net amount of the two transactions, but by considering them independently we can better investigate the rationality of the AAA). The AAA also makes guesses about the effects of its actions, first of all the actual contracts  $Ac_t$  and  $Vc_t$  that it will stipulate with another subject (in a one-agent model, the hypothesis is that the AAA always finds its counterpart, so  $Ac_t = A_t$  and  $Vc_t = V_t$ ). Exchanges are regulated at the opening price of the day, equal to the closing one of the previous day,  $p_{t-1}$ . Other guesses about effects are:  $M_t$ , the quantity of money after buying and selling;  $S_t$ , the quantity of shares, always after buying and selling;  $W_t$ , wealth valued as the sum of  $M_t$  and  $S_t$ , pricing shares at the closing price of the day,  $p_t$ . This difference in prices is the key to interpreting our AAA behaviour, both in this case without EO and in section 4.2.4. The inputs of the ANN emulating the AAA are:  $M_{t-1}$ , the quantity of money owned at time  $t-1$ ;  $S_{t-1}$ , the quantity of shares at the same time;  $W_{t-1}$ , global wealth (money and shares);  $A_{t-1}$ ,  $V_{t-1}$ ,  $A_{t-2}$ ,  $V_{t-2}$ , the quantities of shares owned bought (A as acquisition) and sold (V as vendor) at time  $t-1$  or  $t-2$ ;  $p_{t-3}$ ,  $p_{t-2}$ ,  $p_{t-1}$ , exogenous prices at the specified time.

We refer here to equations 4.1 to 4.7 in section 4.2.3. Due to the independence of the agents, we replace (4.1) and (4.2) with:

$$*Ac_t = A_t$$

$$*Vc_t = V_t$$

(For practical reasons, in this case the computer model is over-simplified, eliminating the  $Ac$  and  $Vc$  variables.)

Only the graph of one agent per experiment is shown: all values other than  $p_t$  are denoted by  $_{gm}$  where  $g$  means guess and  $m$  is the number identifying the agent (an interacting one or behaving independently in a population of independent agents). In Figures 4.8 to 4.13, we will mainly look at  $M_{_g}$  and  $S_{_g}$ , to discover their positive or negative relations with the price cycle.

In Figure 4.8, without EO and without EP, we have an agent acting as if it were trying to avoid risk: it sells all the shares and keeps the money. This result arises because of the CT mechanism; in our model, share prices change continuously and those changes determine errors among the guesses

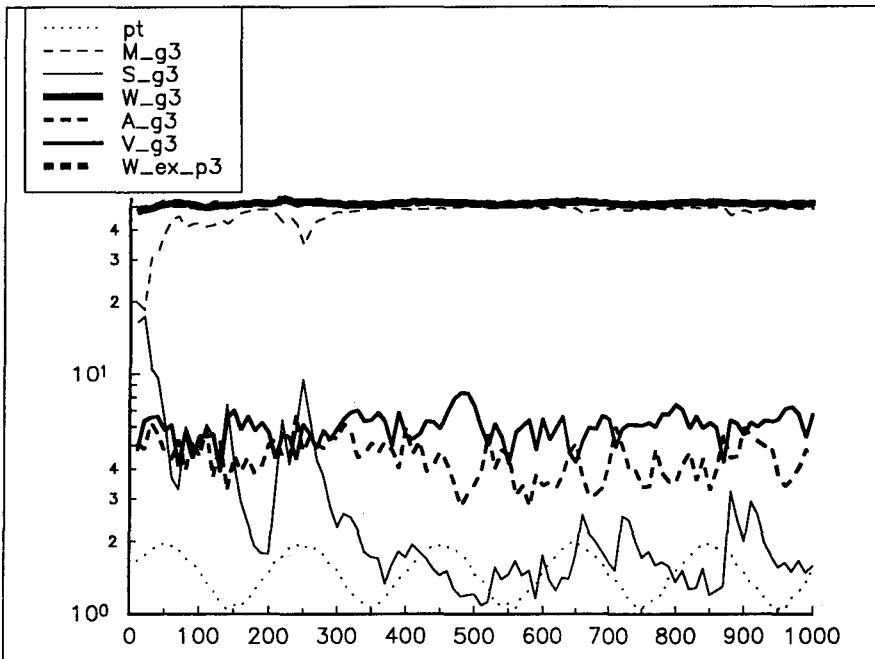


Figure 4.8 *Agent trying to avoid risk, without EO.*

and the targets related to the effects of the actions. Without external suggestions, the agent chooses to hold only money, greatly simplifying the task of developing consistency between the guesses of the actions and those of their effects. Again, we have the emergence of sophisticated behaviour from a simple self-developed micro-mechanism.

#### 4.2.3 The general framework of the stock market model

We now introduce, following the first quick explanation of the previous section, the general framework of the stock market model used until the end of the chapter. However, the reading of this section can be skipped, coming back to it after sections 4.2.4 to 4.2.6 which present the results from the experiments.

The model shown here follows Terna (1993b); globally we have 10 inputs, 13 hidden elements and 7 outputs, as described in section 4.2.2 (b).

We adopt the following operators:

- $H[x_1, x_2, \dots, x_n] = x_i$ , where  $x_i$  is the largest absolute value;
- $C_i$ ,  $i$ -th random value uniformly distributed in the  $[0, 1]$  range; we operate here with 20 agents; the  $C$  operators (such as  $C_{61}$  in equation

4.1; or  $C_{31}$ ,  $C_{32}$  and  $C_{33}$  of equations 4.9 and 4.10; or  $C_{81}$  in the EPZ definitions of section 4.2.6) are numbered taking into account the presence of the other agents;

- $\{i \text{ n } s_1 s_2 \dots s_n\}$  list operator number  $i$ , which chooses randomly among  $n$  agents, avoiding the choice of the agent itself; the complete list is shown only in the first declaration; in subsequent uses the operator is written in the short form  $\{i\}$ ; so  $V_t\{i \text{ n } s_1 s_2 \dots s_n\}$ , or  $V_t\{i\}$ , means the value  $V_t$  of one of the  $n$  agents, randomly chosen. Several list operators can appear in a model, with different numbers. In each simulation period, or 'day', the value of each list operator is kept constant.

The equations describing the targets are reported below; variables with a little star on the left are targets; they will have eventually have EO or EP on the left. In the following experiments we have 20 agents. The complex notation adopted here is strictly related to the necessity of writing rules for CT determination.

$$\begin{aligned} *A_{ct} = & \\ (V_t\{1 \ 20 \ 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9 \ 10 \ 11 \ 12 \ 13 \ 14 \ 15 \ 16 \ 17 \ 18 \ 19 \ 20\} & \\ - A_t) \cdot C_{61} + A_t & \end{aligned} \quad (4.1)$$

$$*V_{ct} = (A_t\{1\} - V_t) \cdot (1 - C_{61}) + V_t \quad (4.2)$$

$$*M_t = *M_{t-1} + (*V_{ct} - *A_{ct}) \cdot p_{t-1} \quad (4.3)$$

$$*S_t = *S_{t-1} + *A_{ct} - *V_{ct} \quad (4.4)$$

$$*W_t = *M_t + *S_t \cdot p_t \quad (4.5)$$

$$\begin{aligned} *A_t = H[(A_{ct} - *A_{ct}), - C_1 \cdot (M_t - *M_t)/p_{t-1}, C_2 \cdot (S_t - *S_t), - \\ C_3 \cdot (W_t - *W_t)/(p_{t-1} - p_t)] + A_t \end{aligned} \quad (4.6)$$

$$\begin{aligned} *V_t = H[(V_{ct} - *V_{ct}), (1 - C_1) \cdot (M_t - *M_t)/p_{t-1}, - \\ (1 - C_2) \cdot (S_t - *S_t), (1 - C_3) \cdot (W_t - *W_t)/(p_{t-1} - p_t)] + V_t \end{aligned} \quad (4.7)$$

This is the general framework of our agents, with neither EO nor EP. The implicit goal of this cross-target neural model is the development of coherence between the effect and the action of the outputs. Everybody acts (buying or selling) at the price  $p_{t-1}$ , which is the closing price of the previous day, known by all agents; after the action – at the end of the day – the agents know the closing price of the day ( $p_t$ ); we remember that all prices are exogenous. The goal of equations 4.1 and 4.2 is stating a random

matching between the demand and the supply of any pair of exchanging subjects; other solutions (the min value, the arithmetic mean) do not give significant differences.

Subdividing agents into two populations, we can introduce for the first one an external objective (EO) like the one expressed in equation 4.8, with the obvious meaning of improving wealth at a daily fixed rate:

$$EO^*W_t = EO^*W_{t-1} \cdot 1.0005 \quad (4.8)$$

The cross-targets original equation 4.5 always runs to determine  $^*W_t$ , which is employed in equations 4.6 and 4.7. EO agents (first population) adjust the weights with which they guess their actions to take advantage of the difference between  $p_{t-1}$  and  $p_t$ .

The experiments are conducted on the basis of an exogenous price, generated by a sinusoidal function with min 1.05, amplitude 0.9 and a random perturbation of  $\pm 0.05$ , giving a complete range [1, 2]; the starting price is about 1.5. In equations 4.6 and 4.7 the corrections are made to  $A_t$  and  $V_t$  values which influence the proper  $Ac_t$  and  $Vc_t$  values, that is, the effects upon which the determination of all the other effects is founded.

Finally, to run the experiments of section 4.2.6, and only for the agents of the second population of those cases, equations 4.6 and 4.7 must be replaced with:

$$^*A_t = H[(Ac_t - ^*Ac_t), -C_{31} \cdot (M_t - ^*M_t)/p_{t-1}, \\ C_{32} \cdot (S_t - ^*S_t), -C_{33} \cdot (W_t - ^*W_t)/(p_{t-1} - p_t), EPZ] + A_t \quad (4.9)$$

$$^*V_t = H[(Vc_t - ^*Vc_t), (1 - C_{31}) \cdot (M_t - ^*M_t)/p_{t-1}, \\ -(1 - C_{32}) \cdot (S_t - ^*S_t), \\ (1 - C_{33}) \cdot (W_t - ^*W_t)/(p_{t-1} - p_t), EPZ] + V_t \quad (4.10)$$

to account for EP, as defined in section 4.2.6 itself.

#### 4.2.4 External goals in one-agent models

Until this section, the role of the experimenter in designing models was limited to the definition of the input and output variables and of the accounting rules linking actions and effects. Economic theory and behavioural prescriptions were unnecessary in this first phase. Some theoretical hypotheses have however been introduced in the definition of variables, such as the choice in section 4.2.3 of using the  $p_{t-1}$  price in accounting for the effect of share bargaining and the price  $p_t$  in the evaluation of wealth, so introducing a lag of length 1 in the strategy of the agent.

We now look for the effect of a simple external goal. Always using the model of section 4.2.3 and with equations 4.1 and 4.2 modified as in section 4.2.2, we adopt the target  $W$  definition of equation 4.8, imposing

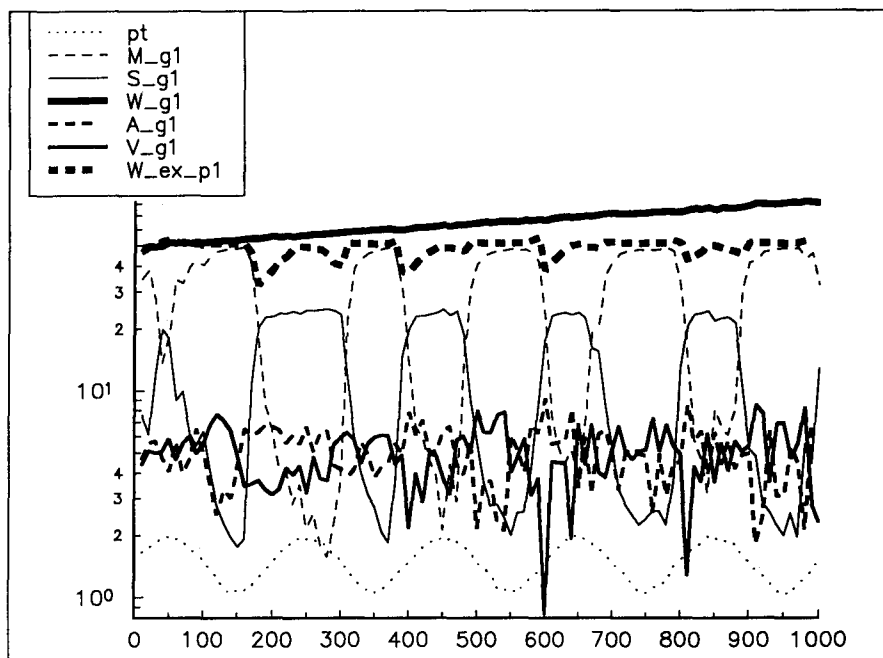


Figure 4.9 *Cyclical behaviour induced by EO.*

an increase of the variable in each cycle. Our stock market agent now has the goal of increasing its wealth.

The introduction of the goal modifies the agent's portfolio decisions, developing a short-term strategy. The CT mechanism smoothly propagates external correction applied to the effects of actions, to action guesses. The interest of this approach is to evaluate the cognitive consequences of simple objectives, such as improving wealth.

It is meaningful to compare results obtained both with and without external goals. With EO the agent is compelled to augment  $W$  and so it acts in the short term, as in Figure 4.9, selling when the price diminishes and vice versa. This strategy is incorrect in a long-term view, but it is very frequent in small investor decisions. Again, it is completely self-developed, as a consequence of CT consistency and of an EO which corrects only guesses about effects.

#### 4.2.5 Interacting cross-target agents

Here we use CT agents interacting with or without external goals, in many-agent and many-population models. In the next section we will show in one table the results of all the experiments with interacting agents: the experiment of this section is labelled Exp. A. Equations 4.1 and 4.2 are

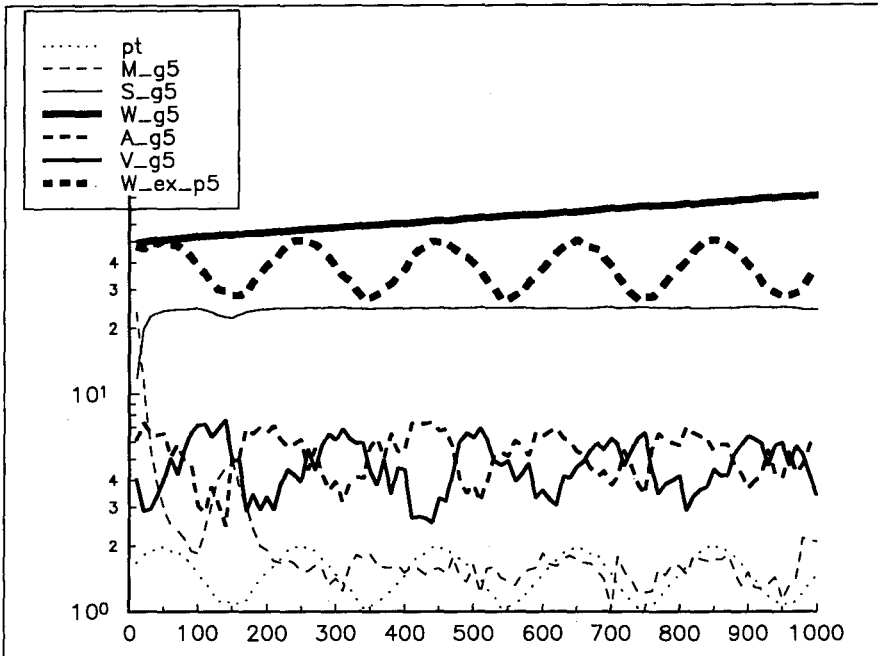


Figure 4.10 *Interacting agent with EO, without counterpart.*

used as specified in section 4.2.3, eliminating the corrections of sections 4.2.2(b) and 4.2.4.

In Exp. A we have 20 interacting AAAs of two types: one population of ten agents acting with the EO of equation 4.8 and the other, also of ten agents, without an EO. In the first case all the agents are similar to the one described in section 4.2.4; in the second case, all the agents are similar to the one described in section 4.2.2(b).

Always relating to the complete presentation of section 4.2.3, we have to introduce here the market rule and the conditions of the exchange among AAAs. Our attempt is to reproduce a market in which each AAA has to find another AAA in order to have a counterpart to bargain with. The counterpart, the same for both selling and buying, is chosen randomly each day; the differences between the quantities  $A_t$  and  $V_t$  or  $V_t$  and  $A_t$  of the two AAAs is randomly divided among them. Other criteria, such as the min value or the average value, do not give significant differences.

The agents of the first population cannot easily find a counterpart for their exchanges, either among themselves because of the analogies in their actions, or in the second population, whose objective is keeping only money. So they are compelled to keep shares in an attempt to augment  $W$ , as in Figure 4.10. Those of the second population are risk-averse, but as they have no EO, they are more oscillating in their erratic behaviour,

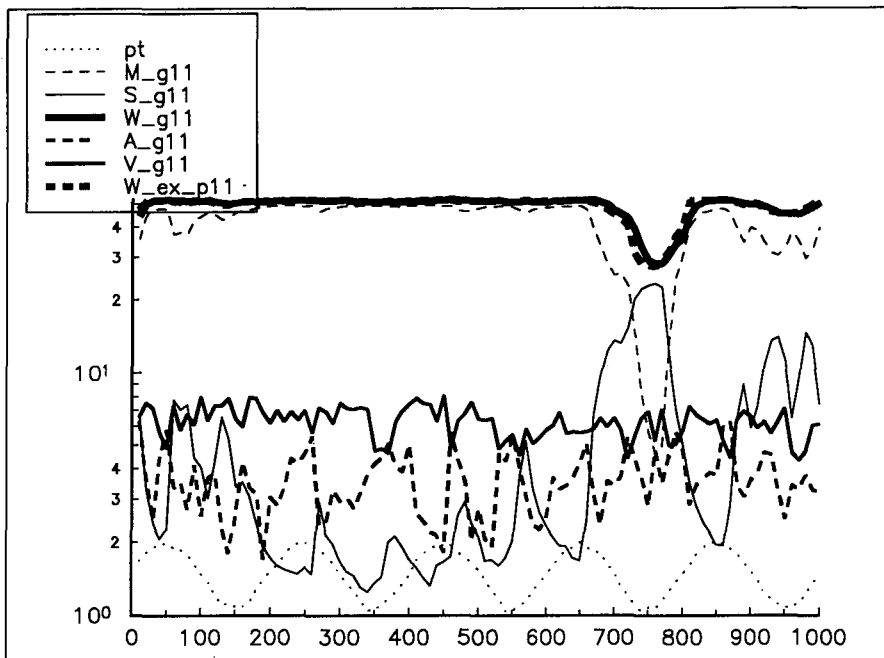


Figure 4.11 *Erratic behaviour of an interacting agent without EO.*

with spurious effects of constant, limited cyclical or limited anti-cyclical behaviour. In Figure 4.11 we have an example of constant behaviour, with two episodes of anti-cyclical acting.

The whole effect of our simple rules is somewhat surprising; the appearance of sophisticated behaviour, such as the anti-cyclical behaviour, will be explained in the following section.

#### 4.2.6 External proposals in agents' interaction

Always using 20 agents in two populations, we develop four experiments (from B to E). As in section 4.2.5 the first population receives an EO from equation 4.8. Targets related to the  $A_t$  and  $V_t$  values of the agents of the second population are influenced by four types of EP. Equations 4.6 and 4.7 are replaced by equations 4.9 and 4.10 to account for EP in the targets of the agents of the second population.

EP influences learning on the side of action determinations: we use imitation and random hints here, with a high degree of similarity to the experiments involving agents foraging for food in section 4.2.1(a).

Imitation is introduced as a means to explain behaviour complexity and to obtain results which resemble a reasoning process.

As well as using imitation, we carry out experiments on the effects of random hints, clearly representing a source of noise. In these experiments, imitation is also a source of noise, as in section 4.2.1(a); however, the use of imitation and randomness together seems to be very powerful.

We now define the term  $EPZ$  of equations 4.9 and 4.10, in four different ways.

- Exp. B: First population with the EO of equation 4.8. Second population with the following EP, showing the consequence of randomness in action determination:

$$EPZ = C_{81} \cdot 2 \cdot K$$

$K$  represents the maximum amount of shares that can be bought or sold daily and 2 is an amplifying coefficient; see section 4.2.3 for the meaning of  $C$  and its index.

- Exp. C: First population with the EO of equation 4.8. Second population with the following EP, showing the consequences of generic imitation, with the agents of the second population imitating all agents. The list operator  $\{ \}$  is explained in section 4.2.3.

$$EPZ =$$

$$A_t\{2 \ 20 \ 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9 \ 10 \ 11 \ 12 \ 13 \ 14 \ 15 \ 16 \ 17 \ 18 \ 19 \ 20 \}$$

in equation 4.9

$$EPZ = V_t\{2\} \text{ in equation 4.10}$$

- Exp. D: First population with the EO of equation 4.8. Second population with the following EP, showing the consequences of specific imitation, with agents belonging to the second population imitating agents of the first population.

$$EPZ = A_t\{2 \ 10 \ 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9 \ 10\} \text{ in equation 4.9}$$

$$EPZ = V_t\{2\} \text{ in equation 4.10}$$

- Exp. E: First population with the EO of equation 4.8. Second population with the following EP, showing the consequences of generic imitation plus randomness. The  $H$  operator is introduced in section 4.2.3.

$$EPZ =$$

$$H[A_t\{2 \ 20 \ 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9 \ 10 \ 11 \ 12 \ 13 \ 14 \ 15 \ 16 \ 17 \ 18 \ 19 \ 20\}, \\ C_{81} \cdot 2 \cdot K]$$

in equation 4.9

$$EPZ = H[V_t\{2\}, C_{81} \cdot 2 \cdot K] \text{ in equation 4.10}$$

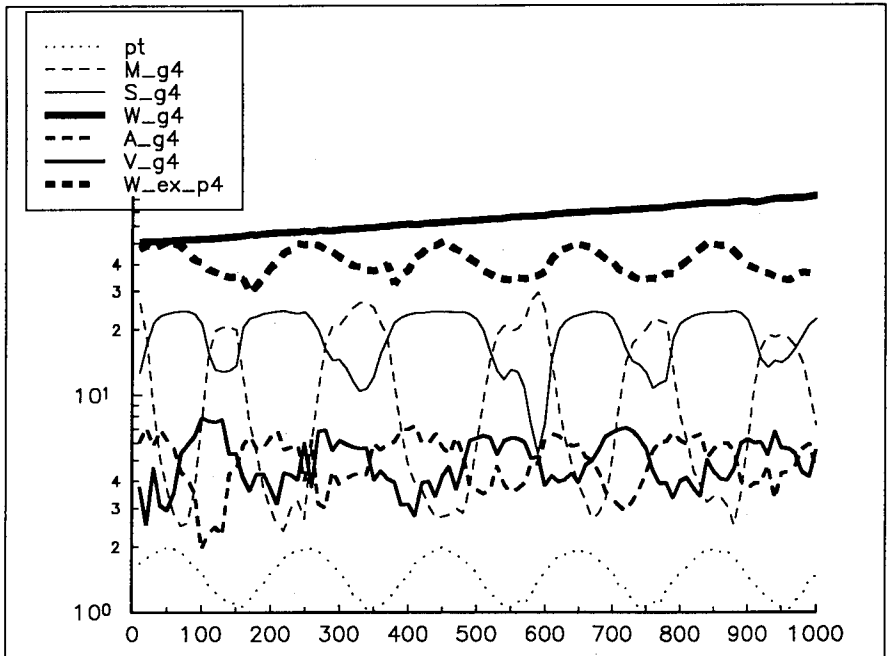


Figure 4.12 *Cyclical behaviour of an EO interacting agent, with counterpart.*

By introducing randomness (Exp. B), generic imitation with the second population imitating all the agents (Exp. C), specific imitation, with the second population imitating the first one (Exp. D) and finally imitation plus randomness, we allow the agents of the first population to find a counterpart for their exchanges, again showing the behaviour of section 4.2.4; the agents of the second population imitate or accept random suggestion and buy or sell against their natural inclination to avoid risk; then actions to avoid risk prevail again, thus producing anti-cyclical behaviour. We show here two representative cases of Exp. E, with an agent of the first population in Figure 4.12 and an agent of the second population in Figure 4.13.

Summarizing, agents 1–10 of the first population and agents 11–20 of the second population show opposite behaviour in almost all the experiments (Table 4.1).

We have to consider the importance of both imitation and randomness in the emergence of ‘rational’ behaviour, and to notice that the global effect is not simply the sum of the two individual effects. We stress the importance of these experiments, explaining what is apparently the consequence of reasoning as the result of random shocks and of imitation in a constrained environment.

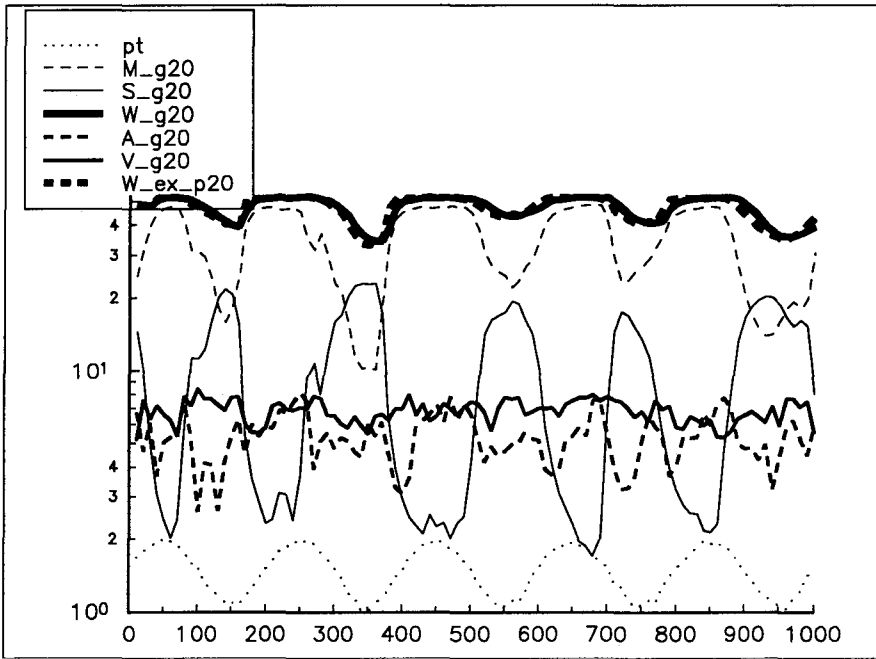


Figure 4.13 *Anti-cyclical behaviour of an interacting EP agent.*

To deeply analyse the behaviour of the agents of Exp. E, and especially of AAAs 4 and 20 shown in Figures 4.12 and 4.13, we adopt here the long-term learning strategy introduced in section 4.1.2. At the end of the experimental cycle of 1000 days, each AAA relearns from its own experience (emerging from the data that it has generated as a result of its behaviour), repeatedly adapting the weights of the ANN upon which it is founded.

Data used in the relearning task are the outputs of the ANN simulating the agent behaviour during short-term learning and acting; outputs are preferred to targets to refer to the actual agents' actions. The relearning is extended to 100 epochs, corresponding to 100 000 cycles of weight corrections.

After relearning, the ANNs simulating the AAAs produce outputs by applying the weights to the historical inputs, without the effects of CT rules.  $R^2$  is determined from historical outputs (those of the learning and acting short-term phase) and actual outputs after relearning. In evaluating  $R^2$  we have to remember that situations in which agent interaction produces unexpected results always exist: this kind of event reduces the capability of an ANN to measure the outcomes of a given input vector.

We will now evaluate the additive analysis of output values connected with  $M_t$  and  $S_t$ , following section 3.3.5. We adopt a simplified notation, with  $M$  instead of  $M_t$ ,  $M_{-1}$  instead of  $M_{t-1}$ , etc.).

**Table 4.1** Mean correlation coefficients between  $p_i$  and  $M_g$ ,  $S_g$ , first and second population (Exp. A from section 4.2.5)

| Ag. [#] | Exp. A |        | Exp. B |        | Exp. C |        | Exp. D |        | Exp. E |        |
|---------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
|         | $M_g$  | $S_g$  | $M_g$  | $S_g$  | $M_g$  | $S_g$  | $M_g$  | $S_g$  | $M_g$  | $S_g$  |
| 1+10    | 0.101  | -0.069 | -0.316 | 0.401  | 0.030  | 0.034  | 0.140  | -0.089 | -0.622 | 0.634  |
| 11+20   | 0.102  | -0.164 | 0.422  | -0.475 | 0.119  | -0.185 | -0.026 | -0.057 | 0.619  | -0.636 |

**Table 4.2** AAA number 4

| Output   | Input    |          |          |          |          |          |          |          |          |          |
|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|
|          | $M_{-1}$ | $S_{-1}$ | $W_{-1}$ | $A_{-1}$ | $V_{-1}$ | $A_{-2}$ | $V_{-2}$ | $P_{-3}$ | $P_{-2}$ | $P_{-1}$ |
| GWM      |          |          |          |          |          |          |          |          |          |          |
| M        | 15.0     | -2.1     | -4.4     | 4.7      | -5.2     | 5.3      | -5.3     | -6.9     | -2.9     | 7.2      |
| S        | -14.8    | 7.1      | 2.1      | -5.9     | 6.8      | -5.1     | 6.0      | 7.3      | 2.6      | -6.4     |
| MPD      |          |          |          |          |          |          |          |          |          |          |
| M        | 0.836    | -0.168   | -0.318   | 0.269    | -0.305   | 0.310    | -0.311   | -0.378   | -0.135   | 0.456    |
| S        | -0.835   | 0.356    | 0.148    | -0.323   | 0.374    | -0.346   | 0.339    | 0.429    | 0.142    | -0.450   |
| MD       |          |          |          |          |          |          |          |          |          |          |
| M        | 0.434    | -0.053   | -0.116   | 0.138    | -0.156   | 0.155    | -0.154   | -0.233   | -0.105   | 0.209    |
| S        | -0.433   | 0.191    | 0.002    | -0.176   | 0.201    | -0.180   | 0.176    | 0.250    | 0.094    | -0.216   |
| MD (sqm) |          |          |          |          |          |          |          |          |          |          |
| M        | 0.194    | 0.046    | 0.106    | 0.066    | 0.079    | 0.079    | 0.082    | 0.101    | 0.047    | 0.129    |
| S        | 0.210    | 0.092    | 0.084    | 0.080    | 0.102    | 0.088    | 0.093    | 0.111    | 0.040    | 0.112    |

**Table 4.3** AAA number 20

| Output   | Input    |          |          |          |          |          |          |          |          |          |
|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|
|          | $M_{-1}$ | $S_{-1}$ | $W_{-1}$ | $A_{-1}$ | $V_{-1}$ | $A_{-2}$ | $V_{-2}$ | $P_{-3}$ | $P_{-2}$ | $P_{-1}$ |
| GWM      |          |          |          |          |          |          |          |          |          |          |
| M        | 18.8     | 4.9      | 13.0     | 4.0      | -2.0     | 4.4      | -2.4     | 4.6      | 1.0      | -2.2     |
| S        | -14.3    | 5.3      | -7.4     | -4.5     | 3.8      | -4.0     | 3.3      | -4.2     | -1.6     | 0.9      |
| MPD      |          |          |          |          |          |          |          |          |          |          |
| M        | 0.869    | 0.221    | 0.612    | 0.204    | -0.141   | 0.221    | -0.155   | 0.215    | 0.048    | -0.113   |
| S        | -0.578   | 0.140    | -0.357   | -0.261   | 0.172    | -0.236   | 0.152    | -0.198   | -0.083   | 0.041    |
| MD       |          |          |          |          |          |          |          |          |          |          |
| M        | 0.647    | 0.163    | 0.448    | 0.143    | -0.100   | 0.156    | -0.109   | 0.161    | 0.037    | -0.083   |
| S        | -0.418   | 0.120    | -0.243   | -0.173   | 0.117    | -0.156   | 0.101    | -0.138   | -0.056   | 0.034    |
| MD (sqm) |          |          |          |          |          |          |          |          |          |          |
| M        | 0.170    | 0.037    | 0.129    | 0.063    | 0.039    | 0.065    | 0.043    | 0.047    | 0.013    | 0.020    |
| S        | 0.123    | 0.035    | 0.091    | 0.073    | 0.032    | 0.074    | 0.033    | 0.047    | 0.025    | 0.009    |

Referring to agents 4 and 20 of Figures 4.12 and 4.13, the additive searching strategy is applied by introducing clauses (variables) to the rules until they become valid in explaining the AAA's behaviour about M and S, namely the quantity of money and shares that the subject is holding. Rule building is founded upon the data of Tables 4.2 and 4.3, reporting: the GWM matrix defined in section 3.3.5; the Mean Point Derivatives (MPDs), introduced in section 4.2.1(a), and the Mean Derivatives (MDs) also introduced in section 4.2.1(a) calculated for each pattern; to simplify the analysis of these results, we calculate the mean and the standard deviation of the derivatives of each pattern.

The following results explain the determination of M and S, with the  $R^2$  value obtained by introducing only the included variables in the ANN input and keeping the other one constant, as in sections 3.3.5 and 4.2.1(a).

- Agent 4, rule explaining M:

| Variables included in input                      | $R^2$<br>(with full input: 0.873) |
|--------------------------------------------------|-----------------------------------|
| $M_{-1}$                                         | 0.724                             |
| $M_{-1}, p_{-3}$                                 | 0.457                             |
| $M_{-1}, p_{-3}, p_{-1}$                         | 0.730                             |
| $M_{-1}, p_{-3}, p_{-1}, W_{-1}$                 | 0.769                             |
| $M_{-1}, p_{-3}, p_{-1}, W_{-1}, A_{-2}$         | 0.761                             |
| $M_{-1}, p_{-3}, p_{-1}, W_{-1}, A_{-2}, V_{-2}$ | 0.776                             |

- Agent 4, rule explaining S:

| Variables included in input              | $R^2$<br>(with full input: 0.866) |
|------------------------------------------|-----------------------------------|
| $M_{-1}$                                 | 0.701                             |
| $M_{-1}, p_{-3}$                         | 0.611                             |
| $M_{-1}, p_{-3}, p_{-1}$                 | 0.709                             |
| $M_{-1}, p_{-3}, p_{-1}, S_{-1}$         | 0.810                             |
| $M_{-1}, p_{-3}, p_{-1}, S_{-1}, V_{-1}$ | 0.816                             |

- Agent 20, rule explaining M:

| Variables included in input | $R^2$<br>(with full input: 0.945) |
|-----------------------------|-----------------------------------|
| $M_{-1}$                    | 0.781                             |
| $M_{-1}, W_{-1}$            | 0.832                             |

|                                          |       |
|------------------------------------------|-------|
| $M_{-1}, W_{-1}, S_{-1}$                 | 0.888 |
| $M_{-1}, W_{-1}, S_{-1}, p_{-3}$         | 0.923 |
| $M_{-1}, W_{-1}, S_{-1}, p_{-3}, A_{-2}$ | 0.925 |

- Agent 20, rule explaining S:

|                                          |                                   |
|------------------------------------------|-----------------------------------|
| Variables included in input              | $R^2$<br>(with full input: 0.945) |
| $M_{-1}$                                 | 0.756                             |
| $M_{-1}, W_{-1}$                         | 0.812                             |
| $M_{-1}, W_{-1}, A_{-1}$                 | 0.821                             |
| $M_{-1}, W_{-1}, A_{-1}, A_{-2}$         | 0.816                             |
| $M_{-1}, W_{-1}, A_{-1}, A_{-2}, p_{-3}$ | 0.908                             |

Note that agent 4, which behaves according to the EO applied to  $W$  and develops unproductive cyclical behaviour, assesses  $M$  on the basis of  $M_{-1}$ , two prices and  $W$ . Notice that if we consider the two prices separately,  $R^2$  diminishes: clearly those variables interact in a nonlinear way. Agent 4 assesses  $S$  on the basis of the same variables, plus  $S_{-1}$ . Therefore, both variables are explained by their preceding value, with a relevant effect from  $M_{-1}$  and prices.

Agent 20, influenced by the EP and developing anti-cyclical behaviour, in an apparently rational way, chooses  $M$  mainly on the basis of  $M_{-1}$ ,  $W_{-1}$ ,  $S_{-1}$ , with a small effect on prices. The same agent establishes  $S$  by  $M_{-1}$  and  $W_{-1}$ , taking into account the quantities of shares bought in the previous days. We underline the interesting presence of  $W_{-1}$  in this scheme: a link to wealth in a context in which the previous EO does not operate upon  $W$ ; this is a case of apparent rationality, emerging from randomness and imitation. Links emerging from input to output variables in our learning and adapting agents have the main characteristic of being totally self-developed, without using learning strategies such as least squares with a priori functions or complex credit assignment techniques as found in CSs. As a conclusion, from the models presented in this chapter we learn that simple micro-mechanisms are sufficient to develop complex behaviour and that the CT structure is sufficient to develop those mechanisms autonomously. These are only demonstrations of the plausibility of some intuitions and absolutely not proofs or verifications of those intuitions. However, this is a possible way to conduct artificial experiments, also useful for investigating and understanding the reasons for and the consequences of the actions of real economic subjects.

## **Neural Networks for Economic and Financial Modelling**

**Copyright © 1996 Andrea Beltratti, Sergio Margarita and Pietro Terna**

**ITP** A division of International Thomson Publishing Inc.  
The ITP logo is a trademark under licence

---

All rights reserved. No part of this work covered by the copyright may be reproduced or used in any form or by any means – graphic, electronic, or mechanical, including photocopying, recording, taping, or information storage and retrieval systems – without the written permission of the Publisher, except in accordance with the provisions of the Copyright Designs and Patents Act 1988.

Whilst the Publisher has taken all reasonable care in the preparation of this book, the Publisher makes no representation, express or implied, with regard to the accuracy of the information contained in this book and cannot accept any legal responsibility or liability for any errors or omissions from the book or the consequences thereof.

Products and services that are referred to in this book may be either trademarks and/or registered trademarks of their respective owners. The Publisher/s and Author/s make no claim to these trademarks.

### **British Library Cataloguing-in-Publication Data**

A catalogue record for this book is available from the British Library

### **Library of Congress Cataloging-in-Publication Data**

A catalog record for this book is available from the Library of Congress

---

### **First printed 1996**

Commissioning Editor: Liz Israel Oppedijk

Cover Designed by Buttons Eventures

Typeset in the UK by Florencetype Limited

Printed in the UK by Cambridge University Press

---

### **ISBN 1-85032-169-8**

International Thomson Computer Press  
Berkshire House  
High Holborn  
London WC1V 7AA  
UK

International Thomson Computer Press  
20 Park Plaza  
14th Floor  
Boston MA 02116  
USA

**<http://www.thomson.com/itcp.html>**

Imprints of International Thomson Publishing