Journal of Forecasting
J. Forecast. **21**, 559–577 (2002)
Published online 15 August 2002 in Wiley InterScience (www.interscience.wiley.com). **DOI**: 10.1002/for.835

Neural Network Pruning Applied to Real Exchange Rate Analysis

JOHAN F. KAASHOEK* AND HERMAN K. VAN DIJK
Econometric Institute, Erasmus University Rotterdam, The Netherlands

ABSTRACT

Neural networks are fitted to real exchange rates of several industrialized countries. The size and topology of the networks is found through the use of multiple correlation coefficients, principal component analysis of residuals and graphical analysis of network output per hidden layer cell and input layer cell. These pruned neural networks are good approximations to varying non-linear trends in real exchange rates. Non-linear dynamic analysis shows that the long-term equilibrium values of several European currencies correspond to the actual values within the European Monetary System. Based on its long-term equilibrium value, the Euro appears to be undervalued *vis-à-vis* the US dollar at the introduction of the Euro on 1 January 1999. Copyright © 2002 John Wiley & Sons, Ltd.

KEY WORDS neural networks size reduction; non-linear dynamics; varying trends in real exchange rates

INTRODUCTION

Neural networks are flexible models for handling complex data patterns of economic variables. This feature has led to the diffusion and implementation of neural network models in the fields of economics and econometrics; see e.g. (Gallant and White, 1988; White, 1989; Kaashoek and van Dijk, 2002). However, the price of this flexibility is the danger of overfitting. That is, fitting the noise process may easily occur and bad predictive behaviour may be the result; see Bishop (1995). Possible overfitting of neural networks may be explained as follows. A simple neural network model consist of three layers each of which contains a number of cells. The three layers are an input layer with a certain number of cells, a hidden layer with a large number of cells and an output layer with few cells. Given the large number of cells in the hidden layer the network output may encompass almost entirely the spectrum of the real observed output variable. Thus, a crucial point is to develop practical procedures to reduce or 'prune' the size of the hidden layer. Such a strategy can also be applied to 'prune' the size of the input layer.

A threefold procedure for reducing the size of the network is proposed in this paper. The basic idea is what is called by Theil (1971), in the context of linear regression, the incremental contribution

* Correspondence to: Johan F. Kaashoek, Econometric Institute (H11-31), Erasmus University Rotterdam, Burg. Oudlaan 50, NL-3062 PA Rotterdam, The Netherlands. E-mail: kaashoek@few.eur.nl

of explanatory variables. That is, how much is the reduction of the explained variance of the dependent variable when we exclude an explanatory variable. We apply this idea to neural networks by excluding hidden layer cells and/or input cells from the networks.

Our starting point is a graphical comparison of network output and observed data with only one cell excluded and with all other cells included. Next, in order to get a quantification of network performance with one cell excluded, the reduced contribution is measured in terms of multiple correlation coefficients. A variable with a low incremental contribution will be a candidate to be excluded from the model. Third, we calculate the principal components of the set of residuals obtained by omitting successively one network cell. The vector representing the first principal component may reveal which cell can be excluded from the network. As a pruning method, it falls in the category of cell reduction methods; see e.g. Mozer and Smolensky (1989). Our approach is less numerically intensive since the contribution of cells is based on the outcome of only one optimization procedure with all variables included.

We emphasize that our method is a descriptive technique which can be useful for exploratory data analysis. The particular network, which results from the cell-pruning procedure, may be used for dynamic analysis and prediction. That is, by making use of a simple recursive procedure, the network generates a data series called orbit. The generated orbit may indicate the presence of non-linear trends in the data. Further, long-run stability and equilibrium values may be computed. Predictive properties may be investigated using scoring rules like mean square prediction error and information criteria. For a statistical approach to deleting cells we refer to White (2000).

As actual data we use the logarithms of monthly dollar real exchange rates of several industrialized countries for the period 1957-1998. We determine the varying trend, the stability and the long-term equilibrium values of the exchange rates. Next, we investigate the exchange rates within a number of countries of the European Monetary Union. Our results indicate that within the EMU-countries the exchange rates were at 1 January 1999 'properly fixed' in the sense that the actual values of these exchange rates are conform the estimated long term values. The same subject is investigated for the Euro/US dollar exchange rate. Here we find evidence that at 1 January 1999 the value of the Euro compared to the US dollar was fixed at a rate which is 10% lower than the long-term equilibrium value.

The contents of the paper is organized as follows. In the next section the graphical analysis, the incremental contribution of cells and the principal component analysis of residuals are explained in the context of a standard feedforward neural network. In the remainder of the paper the pruning procedures are applied to several examples. In the third section data are generated from a 'true' neural network. The fourth section deals with the logarithms of real exchange rates of several industrialized countries. The final section contains our conclusions.

NETWORK PRUNING

The functional form of the network used in this paper may be summarized as:

$$y_t = h_t'c + d + \varepsilon_t \tag{1}$$

$$h_t = G(Ax_t + b) \tag{2}$$

where $t = 1, \ldots, T$. The scalar variable y_t denotes the real output variable at time t. The $1 \times H$ vector h_t' denotes the layer with hidden cells. These hidden or unobserved cells are connected to the $I \times 1$ vector of inputs x through a vector of non-linear functions $G = (g_1, g_2, \ldots, g_H)'$ with as typical element

$$g_h(z) = \frac{1}{1 + e^{-z}}$$

The disturbances ε_t are stochastic variables. We note that in the present paper the input vector x is given as the set of lagged real output variables $(y_{t-1}, y_{t-2}, \ldots, y_{t-I})$. The $H \times 1$ vectors b, c, and d and the $H \times I$ matrix A consists of unknown coefficients. Network output will be denoted as $\hat{y}_t$. A network with I input cells and H hidden cells will be denoted as $nn(I, H)$.

As an example, consider a $nn(1, 2)$ network. This neural network with one input cell and two hidden layer cells, has as functional form:

$$y_t = d + \frac{c_1}{1 + e^{-a_1 y_{t-1} - b_1}} + \frac{c_2}{1 + e^{-a_2 y_{t-1} - b_2}} + \varepsilon_t \qquad (3)$$

In order to determine the parameters of the network we minimize the sum of the squared differences of real output y_t and the network output $\hat{y}_t$, $t = 1, \ldots, T$ with respect to A, b, c and d. As a nonlinear optimization procedure we apply the variable metric method of Davidon, Fletcher and Powell, see Press *et al.* (1988).

Graphical analysis

An simple way to look at neural network performance is to compare the graphs of real output data (t, y_t) and neural network estimates $(t, \hat{y}_t)$.

Consider now network (1) with hidden layer cell h left out; this is equivalent with putting c_h equal to zero. All other parameters are left the same. Without this hidden layer cell h, the network produces an output called $\hat{y}_{-h}$. The graphs of $(t, (\hat{y}_{-h})_t)$ are compared to the graph of (t, y_t) and this comparison may give evidence of the contribution of hidden cell h in explaining the variance of y_t.

In a similar way the importance of input cells $y_{t-1}, \ldots, y_{t-I}$ can be examined. Let $(\hat{y}_{-i})_t$, $i = 1, \ldots, I$ be neural network output with inclusion of all cells except input cell (variable) i (adjusted for mean differences). Then again, visual inspection of the graphs of (t, y_t) and $(t, (\hat{y}_{-i})_t)$ may show evidence for inclusion or exclusion of input cell i. This graphical analysis was already used by Koopmans (1937) and Tinbergen (1939) within the context of the linear regression model.

Incremental contributions of cells

A natural candidate for quantification of the network performance is the square of the correlation coefficient of y and $\hat{y}$

$$R^2 = \frac{(\hat{y}'y)^2}{(y'y)(\hat{y}'\hat{y})} \qquad (4)$$

where $\hat{y}$ is the vector of network output points. Note that y and $\hat{y}$ are adjusted for the mean.

The network performance with only one cell deleted can be measured in a similar way. For instance, if the contribution of hidden cell h is put to zero ($c_h = 0$), then the network will produce an output $\hat{y}_{-h}$ with errors

$$e_{-h} = y - \hat{y}_{-h} \qquad (5)$$

This reduced network can be measured by the square of the correlation coefficient R^2_{-h} between y and $\hat{y}_{-h}$ with

$$R^2_{-h} = \frac{(\hat{y}'_{-h}y)^2}{(y'y)(\hat{y}'_{-h}\hat{y}_{-h})} \tag{6}$$

where y and $\hat{y}_{-h}$, are adjusted for the mean.[1]

Now the incremental contribution of cell h is given as the following difference:

$$R^2 - R^2_{-h} \tag{7}$$

If the value in (7) is low for some h compared to all other values, then this cell is a candidate for exclusion from the network. In our experience, cells with contribution less then one tenth of the cells with highest contribution are to be considered as having a *low* contribution; the cell with lowest contribution is the first candidate for exclusion.

Note that for a linear model with constant term (see e.g. equation (1)), the R^2 of equation (4) equals to

$$R^2_{lin} = 1 - \frac{e'e}{y'y} \tag{8}$$

with

$$e = y - \hat{y} \tag{9}$$

Suppose the h-variable is left out, and the reduced linear model is *estimated* again with errors $\hat{e}_{-h}$ then the incremental contribution of variable h, is given as the difference between the (linear) correlation coefficients (see Theil, (1971)); in formula:

$$\frac{\hat{e}'_{-h}\hat{e}_{-h} - e'e}{y'y} \tag{10}$$

The notation $\hat{e}_{-h}$ is used to emphasize that these residuals are the result of an additional regression of the reduced linear model while the errors given in equation (5), in the linear case, would be simply the result of putting a parameter h to zero. Since equation (10) is based on re-estimating the model after exclusion of a variable, the decision to leave out a network cell based on its low contribution measured by equation (7) is *conservative* with respect to the one which is based on the value given in (10). However, this approach has the obvious advantage that the quantities used are based on one non-linear regression of a non-linear model with possible non-identified parameters. Moreover, after the exclusion of a cell, optimization is prolonged with all parameters (except the one left out) equal to the results obtained in the foregoing optimization round.

The same procedure can be applied to reduce the number of input layer cells. In this case, $\{\hat{y}_{-i}(t)\}$ is network output, given network parameters estimates, without input cell i. The contribution of

[1] Apart from consistency in the definition of multiple correlation coefficients, which are defined in deviation of means, the inclusion of the constant d in the network definition is motivated by the possibility to adjust easily network output $\{(\hat{y}_{-h})_t\}$ for differences in mean.

input cell i is put to zero ($A_{hi} = 0, h = 1, \ldots H$), then the reduced network can be quantified by the square of the correlation coefficient R^2_{-i} between y and $\hat{y}_{-i}$ with

$$R^2_{-i} = \frac{(\hat{y}'_{-i}y)^2}{(y'y)(\hat{y}'_{-i}\hat{y}_{-i})} \tag{11}$$

where y and $\hat{y}_{-i}$ are adjusted for the mean. The contribution of cell i is measured as

$$R^2 - R^2_{-i} \tag{12}$$

The relative value of incremental contributions in R^2 can be used in evaluating whether an input cell can be omitted or not.

Incremental contribution through principal components analysis of network residuals

For the hidden layer cells we define the matrix:

$$E_{-H} = (e_{-1}, e_{-2}, \ldots, e_{-H}) \tag{13}$$

with e_{-h}, $h = 1, \ldots, H$ defined in equation (5). A principal component analysis on the matrix E_{-H}, i.e. the calculation of the orthonormal eigenvectors and eigenvalues of the symmetric matrix $E'_{-H}E_{-H}$, will give the principal components of E_{-H}. The first principal component, corresponding to the maximal eigenvalue, will have maximal variance since the amount of variance of each principal component is proportional to the corresponding eigenvalue; see e.g Malinvaud (1970) and Theil (1971). Hence the first component or better, the eigenvector v_{max} at largest eigenvalue λ_{max} of $E'_{-H}E_{-H}$, defines the linear combination of elements e_{-h} with the largest variance. Otherwise stated: the vector v_{max} gives the worst case combination with respect to omitting cells. Moreover, the elements of this vector v_{max} reveal which variable may be omitted: the cell with index h for which the corresponding element in the first principal component is minimal in absolute sense, may be excluded: its exclusion of the model does not contribute very much to the worst case!

Whether a decision for exclusion and/or inclusion can be based on the factors (=eigenvector) of the first principal component only will depend on the relative weight of this component. Again by the above statement, the relative importance of each component is proportional to the corresponding eigenvalue. Hence, the weight w_k of the kth component is given as the relative magnitude of the corresponding eigenvalue λ_k:

$$w_k = \lambda_k / \sum_{k=1}^{H} \lambda_k \tag{14}$$

As a pragmatic rule, based on our experience principal components with a weight below 0.7 are insignificant.

Similarly, for the input layer cells, we define the matrix:

$$E_{-I} = (e_{-1}, e_{-2}, \ldots, e_{-I}) \tag{15}$$

where e_{-i}, $i = 1, \ldots, I$ are defined as

$$e_{-i} = y - \hat{y}_{-i} \tag{16}$$

Again, the first principal component of E_{-I} may give evidence which input layer cell can be excluded. Of course, economic and time-series analysis may have a stronger impact on the exclusion decision than in the case of hidden layer cells.

We summarize the proposed procedure as follows. Cells are candidates for exclusion if they have a minimal incremental contribution in R^2, measured by equation (7), and/or are the smallest component (in absolute sense) of the first principal component. Both quantities measure the contribution of one cell only and do not reveal (at first sight) for instance 'anti-symmetric' output of pair of cells. The term 'anti-symmetric' output refers to the possibility that one cell generates errors while another cell generates errors with a similar pattern but with reverse sign; see e.g. Figures 4 and 5 below. Such behaviour can be found by graphical analysis and is also revealed by the principal component analysis since both cells will have components in the first principal component of the same order of magnitude but with reverse sign; see e.g. Table V.

We emphasize that our procedures are descriptive and meant for exploratory data analysis. Statements like 'cells with contribution less then one tenth of the cells with highest contribution are to be considered as having a *low* contribution' and 'principal components with a weight below 0.7 are insignificant' are purely based on practical experience. The descriptive statistics used, like R^2, have usually no simple expression for their statistical distribution. An informal explanation is that our class of neural network models is similar to that of threshold models with partially identified parameters. The issue is that the parameters in the logistic function are not well identified when some of the c parameters in equation (1) have a value close to zero. Classical distribution theory in partially identified models is a topic outside the scope of the present paper; see e.g. Phillips (1989). For a recent statistical analysis of the 'pruning' of neural networks we refer to White (2000). The purpose of the present paper is to develop practical tools for a descriptive analysis of a flexible neural network model. The resulting model is 'validated' by investigating its long-term dynamic properties and its predictive performance. In this context one may use scoring rules like the Akaike, Schwartz or Bayesian Information Criterion. Our experience with the two descriptive tools and a graphical method are positive in the sense that the resulting neural network model shows good in-sample predictive performance.

AN EXAMPLE OF A TRUE NEURAL NETWORK

We start with an example which illustrates the pruning method explained above. The data used in this section are generated by a two-dimensional model:

$$\begin{aligned} y_{2,t} &= y_{1,t-1} \\ y_{1,t} &= F_1(y_{1,t-1}, y_{2,t-1}) \end{aligned} \tag{17}$$

or equivalently written

$$y_{2,t} = F_1(y_{2,t-1}, y_{2,t-2}) \tag{18}$$

with F_1 is the function $\mathbb{R}^2 \rightarrow \mathbb{R}$ given by a $nn(2,2)$ neural network. The observed data, denoted as $NN0202$, are only one-dimensional: $\{y_{1,t}\} \equiv y_t$; the sample size is 500.

Applying the procedures as explained above the original (true) neural network is to be found again starting with a neural network with four variable inputs (y_{t-1}, y_{t-2}, y_{t-3}, y_{t-4}), one constant input and six hidden layer cells. The results of an optimization are reported in Table I.

Table I. Incremental contribution in R^2 and principal components

Network (4,6) on Data: $NN0202$						
Network total result: $R^2 = 0.9999$						
Cell excluded:	$-H1$	$-H2$	$-H3$	$-H4$	$-H5$	$-H6$
R^2_{inc}	0.0000	0.1764	0.5934	0.0912	0.0045	0.0112
Eigenvector at first principal component of $E'_{-H}E_{-H}$ (weight = 70.30%)						
Cell excluded:	$-H1$	$-H2$	$-H3$	$-H4$	$-H5$	$-H6$
	−0.0000	0.5001	−0.8647	−0.0198	0.0685	−0.0153
Cell excluded:	$-y_{t-4}$	$-y_{t-3}$	$-y_{t-2}$	$-y_{t-1}$		
R^2_{inc}	0.0000	0.0000	0.9579	0.9934		
Cell excluded:	$-y_{t-4}$	$-y_{t-3}$	$-y_{t-2}$	$-y_{t-1}$		
Eigenvector at first principal component of $E'_{-I}E_{-I}$ (weight = 92.19%)						
	−0.0000	0.0001	−0.6923	−0.7200		

Table II. Incremental contribution in R^2 and principal components

Network (2,4) on Data: $NN0202$				
Network total result: $R^2 = 0.9999$				
Cell excluded:	$-H1$	$-H2$	$-H3$	$-H4$
R^2_{inc}	0.5592	0.9565	0.0678	0.0635
Eigenvector at first principal component of $E'_{-H}E_{-H}$ (weight = 93.58%)				
Cell excluded:	$-H1$	$-H2$	$-H3$	$-H4$
	0.6753	−0.7374	−0.0281	−0.0265
Cell excluded:	$-y_{t-2}$	$-y_{t-1}$		
R^2_{inc}	0.9860	0.9948		
Eigenvector at first principal component of $E'_{-I}E_{-I}$ (weight = 92.80%)				
Cell excluded:	$-y_{t-2}$	$-y_{t-1}$		
	0.7300	−0.6800		

With respect to hidden layer cells, comparing the incremental contributions and the eigenvectors of $E'_{-H}E_{-H}$, hidden layer cells 1 and 5 may be excluded. Moreover, it is obvious that input cells 1 with y_{t-4}, and 2 with y_{t-3} can be excluded. This gives a network with two inputs and four hidden layer cells, a $nn(4, 6)$ network. In Table II the results of an further optimization run are reported.

Table II shows that hidden layer cells 3 and 4 can be excluded now; see e.g. the remarkable pattern in the eigenvectors of $E'_{-H}E_{-H}$. Hence the original size of the true neural network is ‘reconstructed’ indeed.

PRUNED NEURAL NETS APPROXIMATE VARYING TRENDS IN REAL EXCHANGE RATES

Empirical implementation of pruning

The data used in this subsection are the logarithm of US dollar real exchange rates, period January 1957 to March 1998. These data are the extended data of Schotman and van Dijk (1991) who fitted a subset of the same data to a linear auto-regressive model of order one, AR1 for the period 1973–1988.

The Yen-US dollar data are used for the empirical implementation of pruning techniques. These are denoted as *JPUS* and shown in Figure 1.

All data are scaled down to the interval [0.1,0.9] and fed to an initial network which is rather large: $nn(5, 10)$. The results of optimization are summarized in Table III where apart from the incremental contribution measured by R^2, only the principal components (eigenvectors of $E'_{-H}E_{-H}$ and their proportional weights) of hidden layer residuals are given.

From Table III one can conclude that hidden layer cells 3 and 9 may be excluded. For these two cells the incremental contributions are very low. Moreover, the factors in the first principal component (with a relative weight of 93.72%) are also very low for these cells.

Although all input variables, except y_{t-1} have a rather low contribution (not reported here), only reduction of hidden layer cells is applied at this stage. Therefore, optimization is continued after

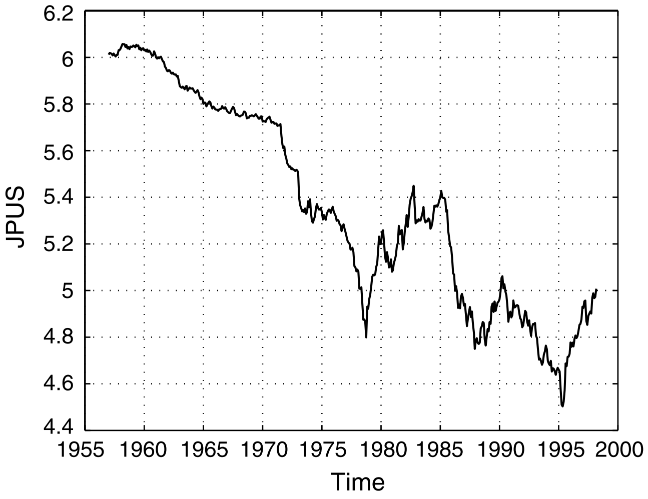

Figure 1. *JPUS* data (unscaled)

Table III. Incremental contribution in R^2 and first principal component

Network (5,10) on Data: *JPUS* Network total result: $R^2 = 0.9965$					
Cell excluded:	$-H1$	$-H2$	$-H3$	$-H4$	$-H5$
R^2_{inc}	0.9192	0.0763	0.0000	0.0001	0.0833
Cell excluded:	$-H6$	$-H7$	$-H8$	$-H9$	$-H10$
R^2_{inc}	0.0300	0.0258	0.0892	0.0000	0.0063
Eigenvector at first principal component of $E'_{-H}E_{-H}$ (weight = 93.72%)					
Cell excluded:	$-H1$	$-H2$	$-H3$	$-H4$	$-H5$
	−0.3919	−0.0683	−0.0047	0.0195	−0.2444
Cell excluded:	$-H6$	$-H7$	$-H8$	$-H9$	$-H10$
	−0.6730	0.3501	0.1671	0.0134	0.4218

Table IV. Incremental contribution in R^2 and first principal component

Network (5, 8) on Data: *JPUS* Network total result: $R^2 = 0.9967$					
Cell excluded:	$-H1$	$-H2$	$-H3$	$-H4$	$-H5$
R^2_{inc}	0.62452	0.9187	0.9706	0.0066	0.8405
Cell excluded:	$-H6$	$-H7$	$-H8$		
R^2_{inc}	0.4708	0.4945	0.0033		
Eigenvector at first principal component of $E'_{-H}E_{-H}$ (weight = 86.30%)					
Cell excluded:	$-H1$	$-H2$	$-H3$	$-H4$	$-H5$
	-0.5030	-0.3241	-0.0657	0.0039	-0.0083
Cell excluded:	$-H6$	$-H7$	$-H8$		
	0.02536	0.7980	0.0092		

removing hidden layer cells 3 and 9. In Table IV the results are summarized. Again, only results on hidden layer cells are reported.

Table IV, and Figures 2 and 3 show that, at least, hidden cells 4 and 8 are candidates for exclusion. First hidden cells 4 and 8 are excluded (based on low factors in the principal component of $E'_{-H}E_{-H}$) and after an additional optimization, still three more hidden layer cells could be excluded so finally a network with only three hidden layer cells was obtained. The results are reported in Table V.

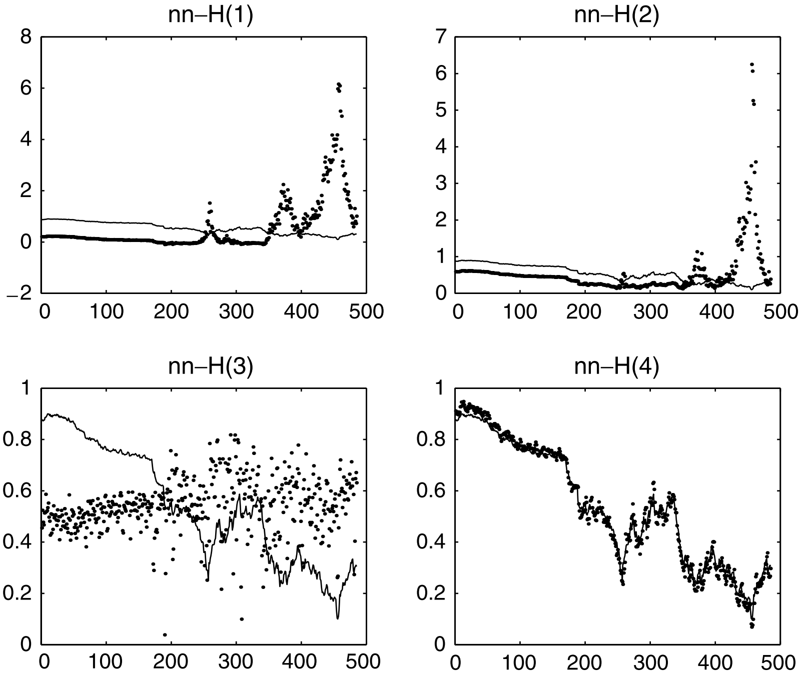

Figure 2. *JPUS* data (scaled to [0.1,0.9]) and $nn(5, 8)$ network output (thick dots) without hidden layer cells 1, 2, 3 and 4 respectively

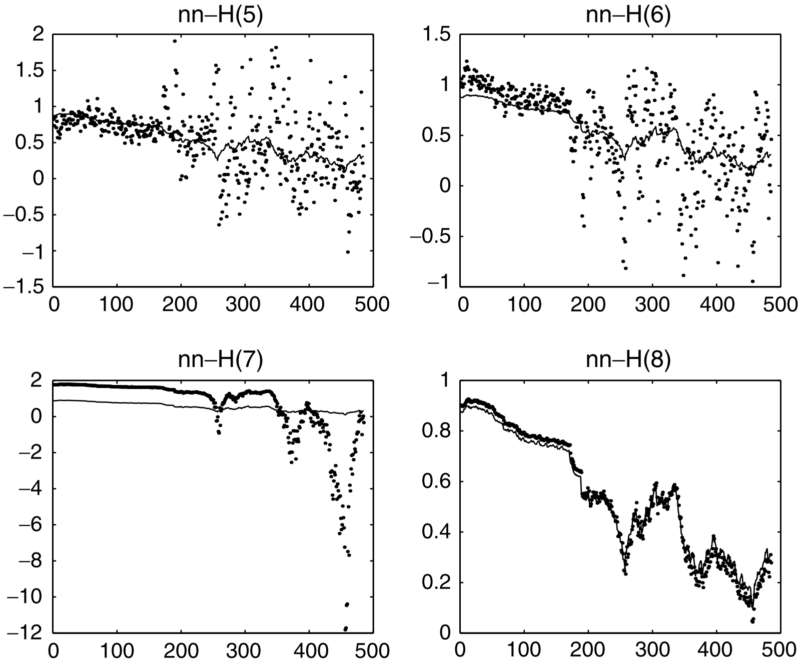

Figure 3. *JPUS* data (scaled to [0.1,0.9]) and $nn(5, 8)$ network output (thick dots) without hidden layer cells 5, 6, 7 and 8 respectively

Table V. Incremental contribution in R^2 and first principal component

Network (5, 3) on Data: *JPUS* Network total result: $R^2 = 0.9965$					
Cell excluded:	$-H1$	$-H2$	$-H3$		
R^2_{inc}	0.8233	0.9960	0.9159		
Eigenvector at first principal component of $E'_{-H}E_{-H}$ (weight = 93.37%)					
Cell excluded:	$-H1$	$-H2$	$-H3$		
	−0.7074	0.0243	0.7064		
Cell excluded:	$-I1$ (y_{t-5})	$-I2$ (y_{t-4})	$-I3$ (y_{t-3})	$-I4$ (y_{t-2})	$-I5$ (y_{t-1})
R^2_{inc}	0.0197	0.2584	0.9070	0.0003	0.9012
Eigenvector at first principal component of $E'_{-I}E_{-I}$ (weight = 95.35%)					
Cell excluded:	$-I1$ (y_{t-5})	$-I2$ (y_{t-4})	$-I3$ (y_{t-3})	$-I4$ (y_{t-2})	$-I5$ (y_{t-1})
	−0.0011	0.0203	0.9858	−0.0082	0.1663

Now all hidden layer cells have a rather large contribution. The second hidden layer cell ($H2$) has a small factor in the first principal component. However, in the second principal component (with a weight of 6.58%), the second cell has a factor equal to 0.9997, so there is no reason to exclude cell $H2$. However, the graphs of network output with exclusion of one hidden layer cell, respectively, show a remarkable pattern: it seems that the contribution of cell $H1$ and cell $H3$ are based on only very limited input values. Above all, the output of those cells seems to be 'anti-symmetric';

see Figure 4 which shows the graphs of network output minus one hidden layer cell (compared to actual data) and Figure 5 which shows the graphs of network output based on only one hidden layer cell each.

Note also that in the principal component hidden cells $H1$ and $H3$ are present with the same order of magnitude but with reverse sign. Based on those graphs a further reduction is applied resulting in a network with only one hidden layer cell. After optimization the network performance can be summarized by $R^2 = 0.9962$ which hardly differs from the one with three hidden layer cells.

With respect to the input variables, the reduction to only one hidden cell has a remarkable effect on the importance of the input variables. While according to Table V, the variable y_{t-1} has a small factor in the first principal component (but a high contribution in R^2), in the case of only hidden layer cell only the variable y_{t-1} is important; all other input variables have a small contribution! To visualize this effect, two figures are supplied: both figures show graphs of network output minus the input of one input cell but Figure 6 applies to the case of three hidden cells while in Figure 7 the number hidden layer cells is only 1.

Therefore, in the case of only one hidden layer cell with only input cell $I5$ active, one is tempted to reduce the number of input cells to 1, with variable y_{t-1} as input. After optimization, this one input- and one hidden layer cell $nn(1, 1)$ network has a performance quantified by $R^2 = 0.9962$, which is only slightly worse than the R^2 for a $nn(5, 3)$ network! This $nn(1, 1)$ neural network has as functional form:

$$y_t = d + \frac{c}{1 + e^{-ay_{t-1}-b}} \tag{19}$$

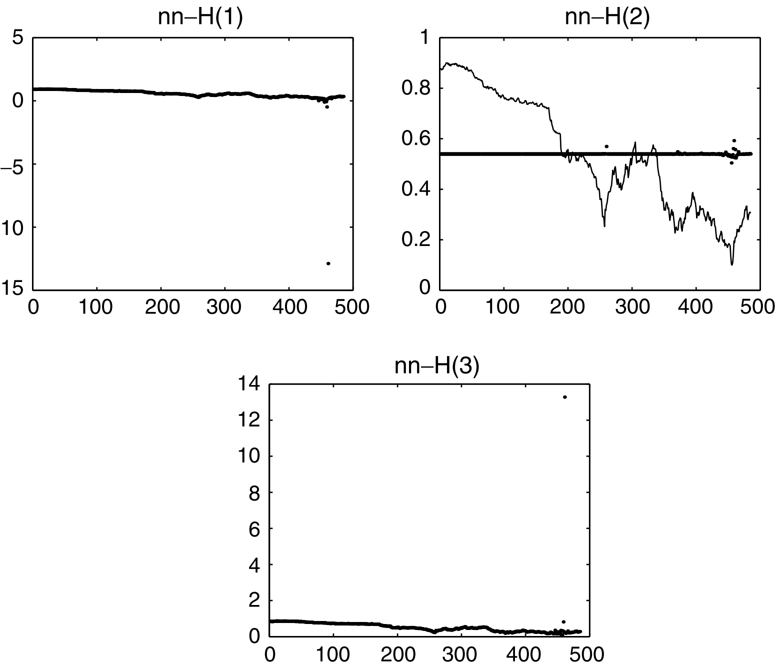

Figure 4. $nn(5, 3)$ network output without one hidden layer cell ($H1$, $H2$ and $H3$, respectively) compared with actual data

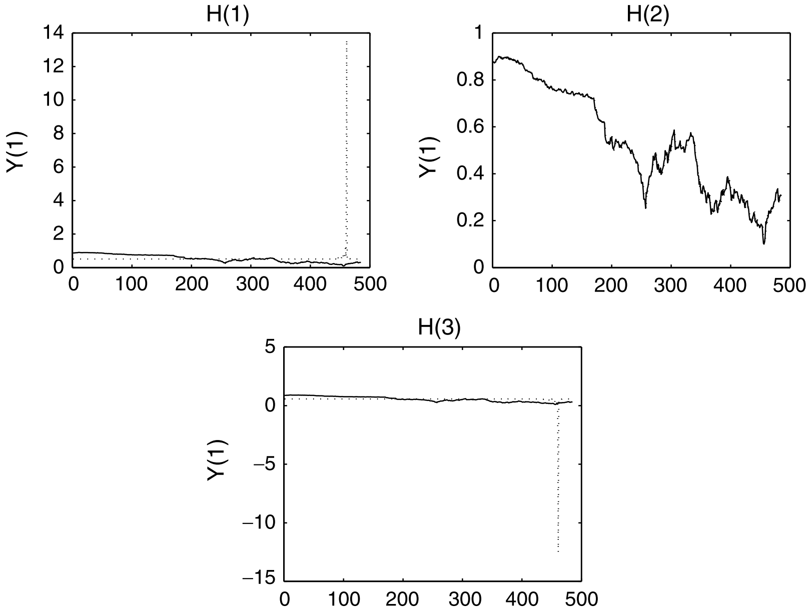

Figure 5. $nn(5, 3)$ network output of only one hidden layer cell ($H1$, $H2$ and $H3$ respectively) compared with actual data

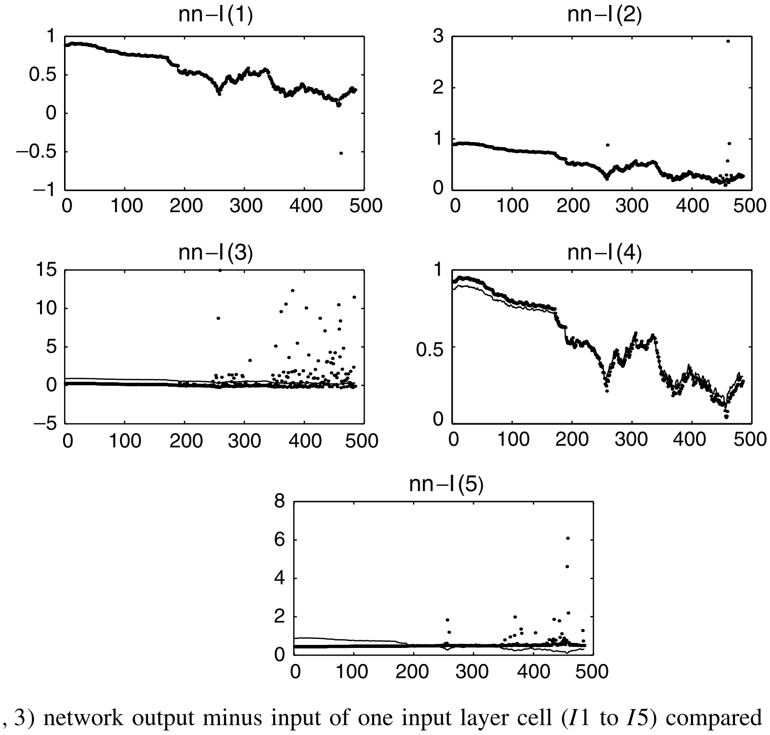

Figure 6. $nn(5, 3)$ network output minus input of one input layer cell ($I1$ to $I5$) compared with actual data

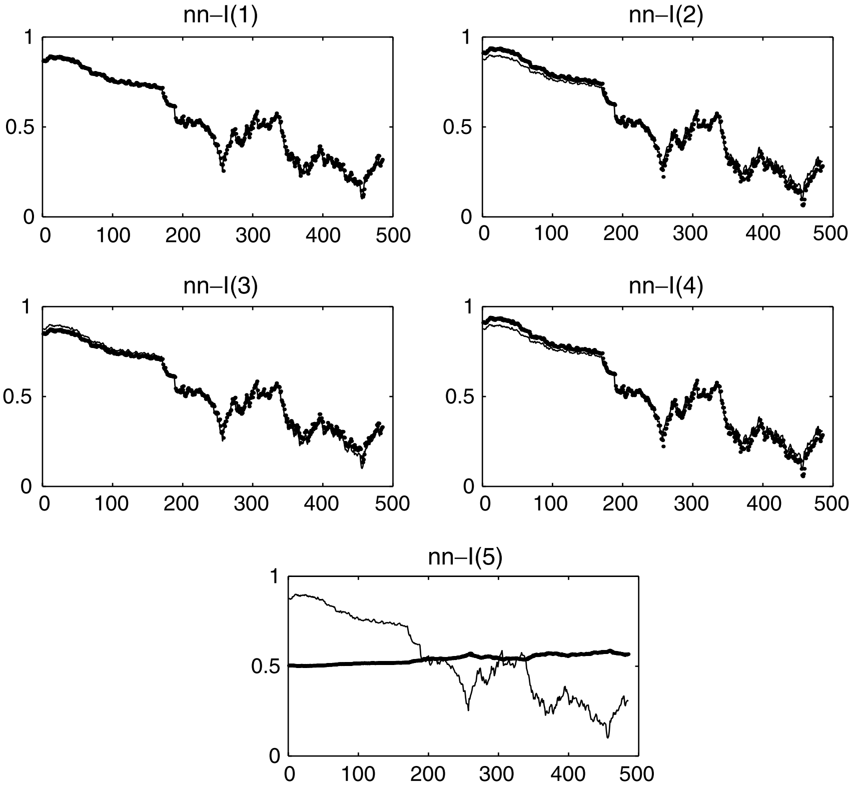

Figure 7. $nn(5, 1)$ network output minus input of one input layer cell ($I1$ to $I5$) compared with actual data

or written as switching model:

$$y_t = d(1 - F(y_{t-1})) + (c + d)F(y_{t-1}) \tag{20}$$

where

$$F(y) = \frac{1}{1 + e^{-ay-b}} \tag{21}$$

The US dollar real exchange rates of five other currencies, namely the Canadian dollar, the French franc, the British pound, the German (west) mark and the Dutch guilder, are fitted also by a neural network model. In all five cases, the reduction process explained above results in a similar neural network configuration: one input cell and one hidden layer cell.

The resulting networks are simple to be analysed. The number of fixpoints for system equation (19) is in all but one case equal to 1. Those fixpoints are stable and even globally attracting: for all starting points, the solution of system equation (19) will tend to that fixpoint. Only the model for the *JPUS*-data has three fixedpoints: two stable ones (0.2910 and 1.2684) and one unstable (0.9773). Note that those values refer to data scaled between 0.1 and 0.9. Hence for all relevant starting points between 0.1 and 0.9, the solution will tend to 0.2910. In Table VI the fixedpoints are summarized.

Table VI. Fixedpoints of model equation (19)

Currency (scaled to [0.1,0.9])	Fixedpoint	Fixedpoint	Fixedpoint
British pound	0.2891		
Dutch guilder	0.3219		
French franc	0.2959		
German mark	0.2926		
Japanese Yen	0.2919	0.9773	1.2684

A first exercise is to analyse the transient behaviour of the $nn(1, 1)$ neural network model. This is done by generating dynamic forecast data $\hat{y}_t$, called *orbits*. Those data are generated in the following way:

$$\hat{y}_t = nn(\hat{y}_{t-1}) \tag{22}$$

where nn is the relevant $nn(1, 1)$ neural network function, and will tend for every initial value $\hat{y}_1$ to the fixpoint of the model.

The orbits (together with the actual data) are depicted in Figure 8. The initial value y_1 is taken from actual data (except for orbit *FRUS* where the starting value is equal to y_{12}). The graphs indicate that in all cases a non-linear trend is a probable model. We conclude that our analysis can be interpreted as a first step to a more detailed analysis of the parametric form of a time series model for real exchanges as, for instance, a threshold model; see Granger and Teräsvirta (1993).

The performance of neural network models compared to ARIMA models

The resulting network $nn(1, 1)$ is compared to a linear ARMA(p, q) model

$$y_t = a + b_1 y_{t-1} + \cdots + b_p y_{t-p} + u_t \tag{23}$$

$$u_t = \varepsilon_t + \theta_1 \varepsilon_{t-1} + \cdots + \theta_q \varepsilon_{t-q} \tag{24}$$

For the *JPUS* data, the results are summarized in Table VII. The R^2 and *SIC*, the Schwartz Information Criterion, are computed for the actual data and the root mean squared errors (RMSE) are computed using dynamical forecasts (orbit-data).

The two classes of models do not differ much with respect to their fit to the actual data, see R^2 and *SIC*-values. However, the neural network model outperforms the ARMA models on sample forecast behaviour with approximately 20%. We note that the use of higher-order ARMA specifications did not improve the results. Moreover, the neural network model shows clearly the transition to a new equilibrium state which is not present in any of the linear models; see Figure 9 and the results presented in the next subsection.

Long-term equilibrium values of exchange rates and the EMU parities

The stable fixedpoints represent long-term equilibrium values of real exchange rates. It is tempting to compare some of those equilibrium values (after rescaling and converting to nominal exchanges rates) with the mutual fixed nominal exchange rates of EMU-currencies and US dollar exchange rate with respect to the Euro as introduced 1 January 1999. In Table VIII the nominal US-dollar exchange rates for all currencies present in the used data set are given based on Consumer Price

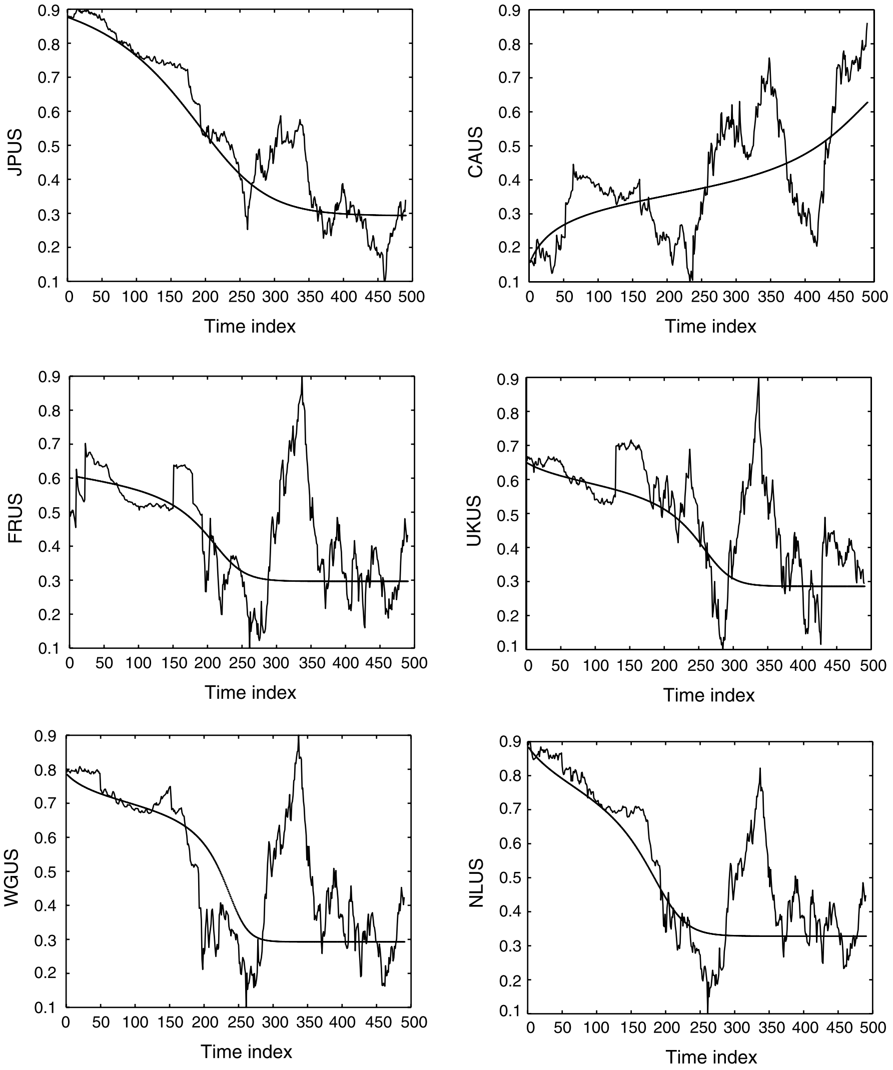

Figure 8. Logarithm of US real exchange rates (Japan, Canada, France, British, Germany (West), Netherlands); real data (scaled to [0.1,0.9]) and neural network $nn(1, 1)$ orbit (thick lines) 1957–1998

Indices (CPI) of 1998. Column 2 of Table VIII gives the real exchange rates calculated from the equilibrium values (fixpoints) of the relevant neural network model; column 3 gives Consumer Price Indices for 1998 with base year 1990 at 100. Using a US Consumer Price Index of 124.5, the nominal exchange rates are calculated; see column 4 of Table VIII.

Using fixed Euro/EMU currency nominal exchange rates one can calculate for all three EMU countries their mutual nominal exchange rates. For instance the German mark/French franc exchange rate is calculated as $1.9558/6.5596 = 0.2981$. In the right half of Table IX, between the vertical lines, in bold, the results of those calculations are summarized.

Table VII. Overview of $nn(1,1)$ and ARMA specification (data scaled to [0.1, 0.9])

Model	Fit actual data		Dynamical forecasts RMSE
	R^2	*SIC*	
nn(1,1)	0.9962	−5.6691	0.0814
ARMA(1,0)	0.9962	−5.6821	0.0993
ARMA(1,1)	0.9962	−5.6811	0.1022
ARMA(2,0)	0.9962	−5.6811	0.1022
ARMA(2,1)	0.9962	−5.6666	0.1027

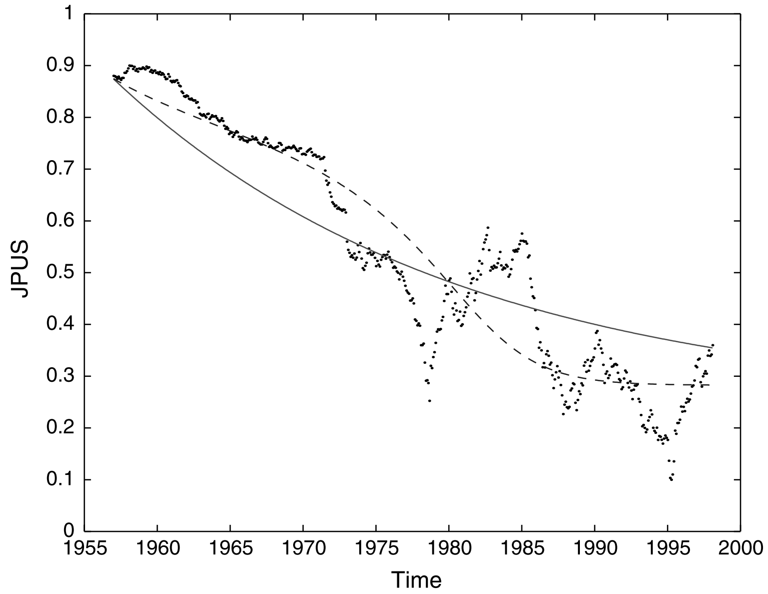

Figure 9. *JPUS* data (scaled, dotted curve), Neural network prediction (dashed curve), and ARMA(1,0) prediction (line)

Table VIII. $nn(1,1)$ model equilibria US-dollar exchange rates

Currency	real	CPI (1998)	nominal
British pound	0.8571	127.5	0.8778
Canadian dollar	1.4480	116.2	1.3515
Dutch guilder	1.7599	121.5	1.7176
French franc	5.5342	115.9	5.1520
German mark	1.5994	119.2	1.5313
Japanese Yen	130.88	109.6	115.22

Table IX. Mutual nominal exchange rates based on fixed Euro conversion rates (bold) and on model US dollar nominal exchange rates

Currency	Euro	US	Euro/US	Dutch	French	German
Dutch guilder	**2.2037**	1.7176	1.2830		**0.3360**	**1.1269**
French franc	**6.5596**	5.1520	1.2732	0.3334		**0.2981**
German mark	**1.9558**	1.5312	1.2773	1.1217	0.2972	

However, instead of using the fixed Euro exchange rates, one can also use the equilibrium values of column 4 in Table VIII (repeated in column 3 of Table IX); e.g. the German mark/French franc exchange rate is now calculated as $1.5312/5.1520 = 0.2972$. The results of those calculations are in the lower-left part of columns 5 and 6 of Table IX.

For the three EMU currencies involved, the neural network gives mutual nominal exchange rates which are comparable with results based on the fixed Euro exchange rates.

The column Euro/US gives Euro/US dollar exchange rates based on the fixed Euro/EMU currency nominal exchange rates, column 2 in Table IX, and the EMU-currency/US dollar nominal equilibrium exchange rates, column 3 of the same table.

Compared to the Euro/US dollar exchange rate of 1.1595 at 1 January 1999, the results in column 4 give a 10% higher rate. We emphasize that this result may heavily depend on the choice of the US Consumer Price Index. In the case of EMU currencies only the relative value of EMU countries' price indices play a role.

CONCLUSIONS

In this paper the number of cells in a neural network is reduced by applying some basic descriptive procedures. The incremental contribution of hidden layer cells and input layer cells is computed using R^2. Another descriptive measure, the principal component analysis of residuals with one cell omitted, confirms the inclusion or exclusion reasoning based on incremental contributions. The advantage of our proposed principal component procedure is that at one stroke two quantities, i.e. the first and last principal component, are obtained which both give evidence which cells can be excluded. Those two quantities are supplemented by graphical analysis of network performance with hidden and/or input cells excluded.

The pruning method is descriptive. As an expert tool it appears to give good results in the sense that the predictive performance and the long-term dynamic properties of the resulting neural network model compare favourably with ARIMA models. For a statistical procedure, such as testing for significance of cells, we refer the reader to White (2000).

We determine the varying trend, the stability and the long-term equilibrium values of several exchange rates. Our results indicate that within the EMU countries the exchange rates were at 1 January 1999 'properly fixed' at their long term values. For the Euro/US dollar exchange rate we find evidence that at 1 January 1999 the value of the Euro compared to the US dollar was fixed at a rate which is 10% lower than the long-term equilibrium value.

We end with listing some topics for further research. An extension of the pruning procedure would be to consider not only the incremental contribution of single cells but of pairs of cells to catch so-called 'anti-symmetric' output of cells. At this stage, in principal, only graphical analysis reveals such behaviour. Also, economic structural interpretation of the empirical results may lead to more restrictions that can be used to reduce the size of the network further. In particular, common non-linear patterns in different countries need to be investigated. Further, descriptive analysis may be used for determining a class of models which is more parsimonious in the number of parameters and still gives a good description of the observed data; see e.g. Granger and Teräsvirta (1993). Finally, more research in econometric methodology, either Classical or Bayesian, may lead to insights on the statistical properties of the proposed procedures; see e.g. White (2000).

ACKNOWLEDGEMENTS

We thank two referees and Timo Teräsvirta, Stockholm School of Economics, for constructive comments which led to a substantial revision of an earlier version of this paper. Responsibility for errors remains, of course, ours.

REFERENCES

Bishop CM. 1995. *Neural Networks for Pattern Recognition*. Clarendon Press: Oxford.

Gallant AR, White H. 1989. There exists a neural network that does not make avoidable mistakes. *Proc. of the International Conference on Neural Networks*, San Diego, 1988 IEEE Press: New York.

Granger CWJ, Teräsvirta T. 1993. *Modelling Nonlinear Economic Relationships*. Oxford University Press: New York.

Kaashoek JF, van Dijk HK. 1998. A simple strategy to prune neural networks with an application to economic time series. Report 9854/A: Econometric Institute Erasmus University Rotterdam.

Kaashoek JF, van Dijk HK. 2002. Neural networks as econometric tool. To appear in *Computer-Aided Econometrics*, Giles D (ed.). Marcel Dekker: New York.

Koopmans TC. 1937. *Linear Regression of Economic Time Series*. De Erven F. Bohn NV: Haarlem.

Malinvaud E. 1970. *Statistical Methods of Econometrics*. North-Holland: Amsterdam.

Mozer MC, Smolensky P. 1989. Skeletonization: a technique for trimming the fat from a network via relevance assessment. In *Advances in Neural Information Processing Systems*, Vol. 1, Touretzky DS (ed.). Morgan Kaufman: San Mateo, CA.

Phillips PCB. 1989. Partially identified econometric models. *Econometric Theory* **5**: 181–240.

Press WH, Flannery BP, Teukolsky SA, Vetterling WT. 1988. *Numerical Recipes*. Cambridge University Press: Cambridge.

Schotman P, van Dijk HK. 1991. On Bayesian routes to unit roots. *Journal of Applied Econometrics*.

Takens F. 1981. Detecting strange attractors in turbulence. In *Dynamical Systems and Turbulence*, Rand DA, Young LS (eds). Springer-Verlag: Berlin.

Theil H. 1971. *Principle of Econometrics*. Wiley: New York.

Tinbergen J. 1939. *Statistical testing of business-cycle theories. Volume 1: A method and its application to investment activity. Volume 2: Business cycles in the United States of America, 1919–1932*. League of Nations: Geneva.

White H. 1989. Some asymptotic results for learning on single hidden layer feedforward network models. *Journal of the American Statistical Association* **84**: No. 408.

White H. 2000. A reality check for data snooping. *Econometrica* **68**: No. 5, 1097–1127.

Authors' biographies:

Johan F. Kaashoek is assistant professor of mathematics of the Econometric Institute, Erasmus University Rotterdam.

Herman K. van Dijk is professor of econometrics and director of the Econometric Institute, Erasmus University Rotterdam.

Authors' address:

Johan F. Kaashoek and **Herman K. van Dijk**, Econometric Institute, Erasmus University Rotterdam PO Box 1738, 3000 DR Rotterdam, The Netherlands.