Neural network analysis of varying trends in real exchange rates

Johan F. Kaashoek (kaashoek@few.eur.nl)
Econometric Institute, Erasmus University Rotterdam, P.O. Box 1738, 3000 DR Rotterdam, The Netherlands

Herman K. van Dijk (hkvdijk@few.eur.nl)
Econometric Institute and Tinbergen Institute, Erasmus University Rotterdam, P.O.Box 1738, 3000 DR Rotterdam, The Netherlands

Report EI9915/A Econometric Institute Rotterdam

Abstract. In this paper neural networks are fitted to the real exchange rates of seven industrialized countries. The size and topology of the used networks is found by reducing the size of the network 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.

1. Introduction

The flexibility of neural networks to handle complex patterns in the data has lead to the diffusion and implementation of neural network models in economics and econometrics; see e.g. Gallant and White (1988) and White (1989). In this paper we consider one type of neural network: a feed-forward 3-layer network with an input layer of I cells, a hidden layer with H cells and an output layer with O cells. We assume that the neural network considered is an approximation of the data generating process:

$$y_t = F(y_{t-1}, \cdots, y_{t-I}) + \epsilon_t, \tag{1}$$

where y_t is a continuous real-valued variable, F is the data generating function and ϵ_t represents an unknown noise term. Hence the dimension O of the output layer is *a priori* given and equal to 1. The network will be denoted as $nn(I, H)$. An upper bound on the size of the input layer I is given by nonlinear data analysis (e.g. embedding dimension, see Takens (1981)). A crucial point is the unknown size H of the hidden layer. A strategy to determine the size H needs to be chosen and this strategy can also be applied to determine the size of the input layer I, since the upper bound may be too large for practical applications.

The flexibility of a neural network makes that overfitting, i.e. fitting the noise process, and consequently bad prediction behaviour can easily

occur; see Bishop (1995). In simple terms: given an unlimited number of hidden layer cells H, the network output encompasses the spectrum of y_t. In this paper, a three fold procedure in reducing the size of the network is proposed. The basic idea is what is called by Theil (1971), in the context of linear regression, the incremental contribution of explanatory variables. That is, how much is the reduction of the explained variance of the dependent variable y 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. Thirdly, 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 cell pruning method, this approach is similar to the one proposed by Mozer and Smolensky (1989). However, our approach has the advantage that the quantities used are based on the outcome of only one optimization procedure with all variables included. A more elaborate exposure of the procedure mentioned above, can be found in Kaashoek and van Dijk (1998).

The particular network, which results from the cell-pruning procedure described above, may be used for 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 nonlinear trends in the data. As such, our analysis may be considered as a first step of exploratory data analysis to varying trends in long economic time series. As actual data we use the logarithm of monthly real exchange rates, against the dollar, of five industrialized countries for the period 1957-1998.

The paper is organized as follows. In the first part, the graphical analysis, the incremental contribution of cells and the principal component analysis of residuals are explained in the context of a standard feed forward neural network.
In the second part the procedures are applied to two examples. First, data are generated by a neural network ("a true neural network"), The second example concerns actual economic data: the logarithm of real exchange rate of the Yen against the dollar 1957-1998. Finally,

results on real exchange rates of four other industrialized countries are reported.

2. Network pruning

The functional form of the network used can be summarized as:

$$y = h' c + d, \tag{2}$$
$$h = G(A x + b), \tag{3}$$

where h, c, b are $H \times 1$ vectors, A is an $H \times I$ matrix and d and y are scalars. The vector function $G = (g_1, g_2, \cdots, g_H)'$ has as typical element $g_h(x) = \frac{1}{1+e^{-x}}$. We note that in the present paper the input x is given as $(y_{t-1}, y_{t-2}, \cdots, y_{t-I})$. Network output will be denoted as $\hat{y}$.

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:

$$y_t = d + \frac{c_1}{1 + e^{-a_1 y_{t-1} - b_1}} + \frac{c_2}{1 + e^{-a_2 y_{t-1} - b_2}} \tag{4}$$

In order to determine the parameters of the network we minimize the sum of the squared differences of y_t and $\hat{y}_t$, $t = 1, \cdots, T$. As an optimization procedure we apply a two-step method as follows. First, the criterion function is adjusted by means of a linear regression of $(\tilde{h}_1, \tilde{h}_2, 1)$ on y_t where $\tilde{h}_i = g_i(\tilde{A} x + \tilde{b})$. A nonlinear optimization, i.e. the variable metric method of Davidon-Fletcher-Powell - see Press et al. (1988) - is applied to adjust the parameters A and b according to the criterion function.

2.1. Graphical analysis

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

Consider now the network (2) 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 $y_{t-1}, \cdots, y_{t-I}$ can be examined. Let $\{\hat{y}_{-i}(t)\}, i = 1, \cdots, I$ 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 in- or exclusion of input cell i.

2.2. Incremental contributions of cells

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

$$R^2 = \frac{(\hat{y}'y)^2}{(y'y)(\hat{y}'\hat{y})} \tag{5}$$

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

$$e_{-h} = y - \hat{y}_{-h} \tag{6}$$

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

$$R^2_{-h} = \frac{(\hat{y}'_{-h}y)^2}{(y'y)(\hat{y}'_{-h}\hat{y}_{-h})} \tag{7}$$

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:

$$R^2 - R^2_{-h}. \tag{8}$$

If the value in (8) is low for some h compared to all other values, then this cell is a candidate for exclusion from the network.

[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.

Note that for a linear model with constant term (see e.g. equation (2)), the R^2 of equation (5) equals to

$$R^2_{lin} = 1 - \frac{e'e}{y'y} \tag{9}$$

with

$$e = y - \hat{y} \tag{10}$$

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:

$$\frac{\hat{e}'_{-h}\hat{e}_{-h} - e'e}{y'y}. \tag{11}$$

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 (6), in the linear case, would be simply the result of putting a parameter h to zero. Since equation (11) 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 (8) is *conservative* with respect to the one which is based on the value given in (11). However, this approach has the obvious advantage that the quantities used are based on one nonlinear regression of a nonlinear 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 input cell i is put to zero ($A_{hi} = 0, h = 1, \cdots H$), then the reduced network can be quantified by the square of the correlation coefficient R^2_{-i} between y and $\hat{y}_{-i}$ with

$$R^2_{-i} = \frac{(\hat{y}'_{-i}y)^2}{(y'y)(\hat{y}'_{-i}\hat{y}_{-i})} \tag{12}$$

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

$$R^2 - R^2_{-i}. \tag{13}$$

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

2.3. Principal components analysis of network residuals

For the hidden layer cells we define the matrix:

$$E_{-H} = (e_{-1}, e_{-2}, \cdots, e_{-H}), \tag{14}$$

with e_{-h}, $h = 1, \cdots, H$ defined in equation (6). 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. And 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:

$$w_k = \lambda_k / \sum_{k=1}^{H} \lambda_k. \tag{15}$$

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

$$E_{-I} = (e_{-1}, e_{-2}, \cdots, e_{-I}), \tag{16}$$

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

$$e_{-i} = y - \hat{y}_{-i}. \tag{17}$$

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.

3. 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:

$$\begin{aligned} y_{2,t} &= y_{1,t-1} \\ y_{1,t} &= F_1(y_{1,t-1}, y_{2,t-1}). \end{aligned} \tag{18}$$

with F_1 is the function $\mathbb{R}^2 \to \mathbb{R}$ 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 length of the data is 500.
Applying the procedures as explained above - see section 2- the original (true) neural network is to be found again starting with a neural network with 4 inputs $(y_{t-1}, y_{t-2}, y_{t-3}, y_{t-4})$, and 6 hidden layer cells. The result 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		

With respect to hidden layer cells, comparing the incremental contributions and the eigenvectors of $E'_{-H}E_{-H}$, hidden layer cell 1 and 5 may be excluded also. 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 2 inputs and 4 hidden layer cells. In Table II the results of an further optimization run are reported.

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		

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.

4. Varying trends in real exchange rates

The data used in this section are the logarithm of Yen-US dollar real exchange rates, period January 1957 to March 1998, denoted as $JPUS$. 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 data are shown in figure 1.

Figure 1. $JPUS$ data

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 .

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		

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. So, optimization is continued after removing hidden layer cell 3 and 9. In Table IV the results are summarized. Again, only results on hidden layer cells are reported.

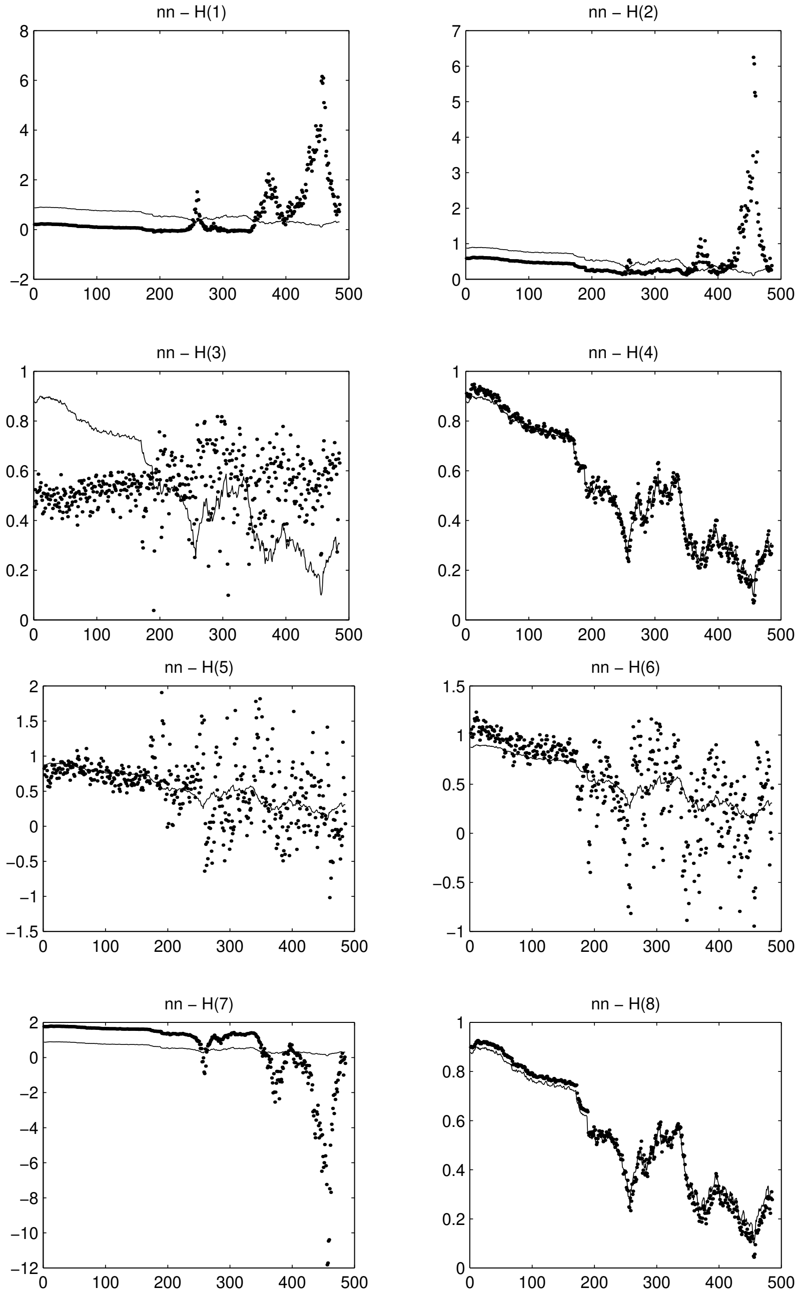

Figure 2. $JPUS$ data and $nn(5,8)$ network output (thick dots) without hidden layer cell 1, 2, 3, 4, 5, 6, 7 and 8 respectively.

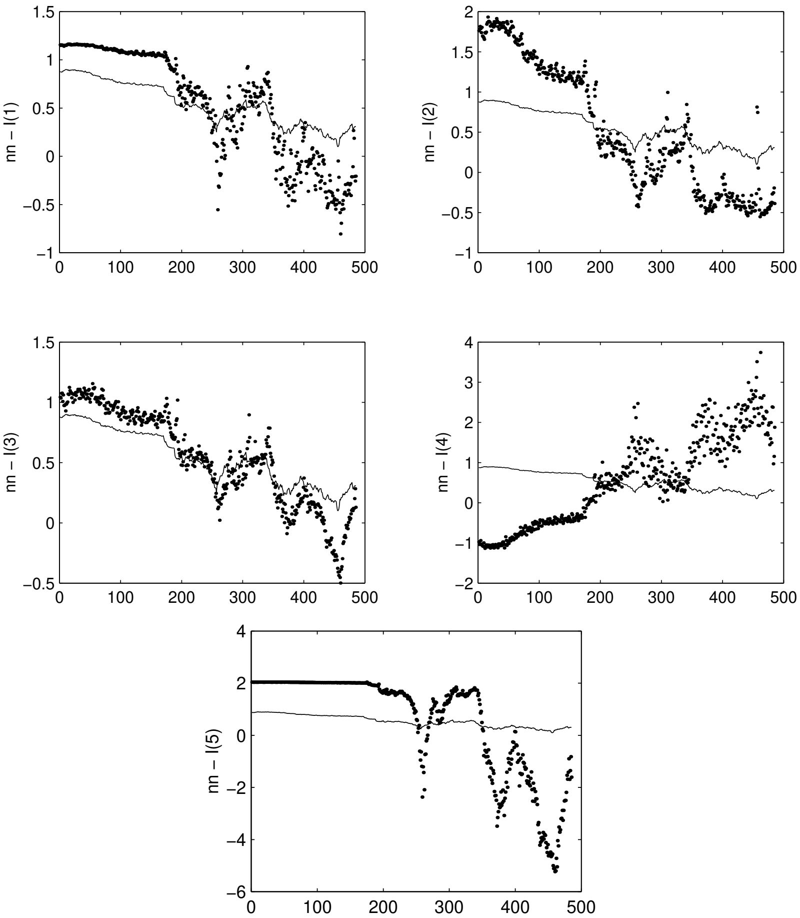

Figure 3. $JPUS$ data and $nn(5,8)$ network output (thick dots) without input layer cell 2, 2, 4, 5, and 6 respectively (input cell 1 represents constant term).

Table IV, and figure (4) 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 3 hidden layer cells was obtained. The results are reported in Table V.

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 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. 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 3 hidden layer cells.

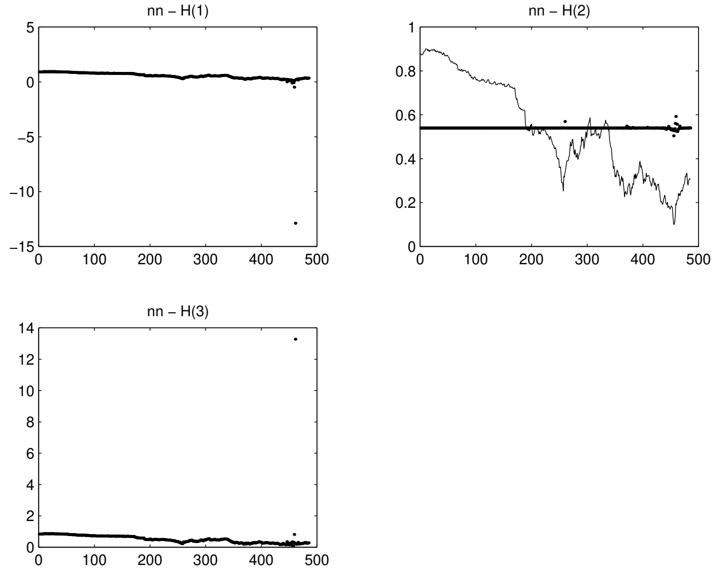

Figure 4. $nn(5,3)$ network output without one hidden layer cell ($H1$, $H2$ and $H3$ respectively) compared with actual data.

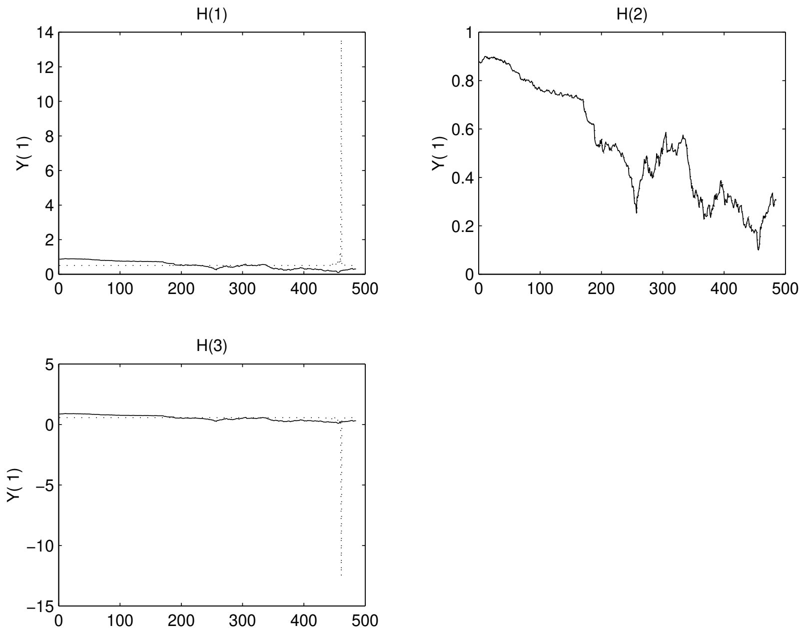

Figure 5. $nn(5,3)$ network output of only one hidden layer cell ($H1$, $H2$ and $H3$ respectively) compared with actual data.

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 just this variable y_{t-1} is important; all other variables do hardly contribute! 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.

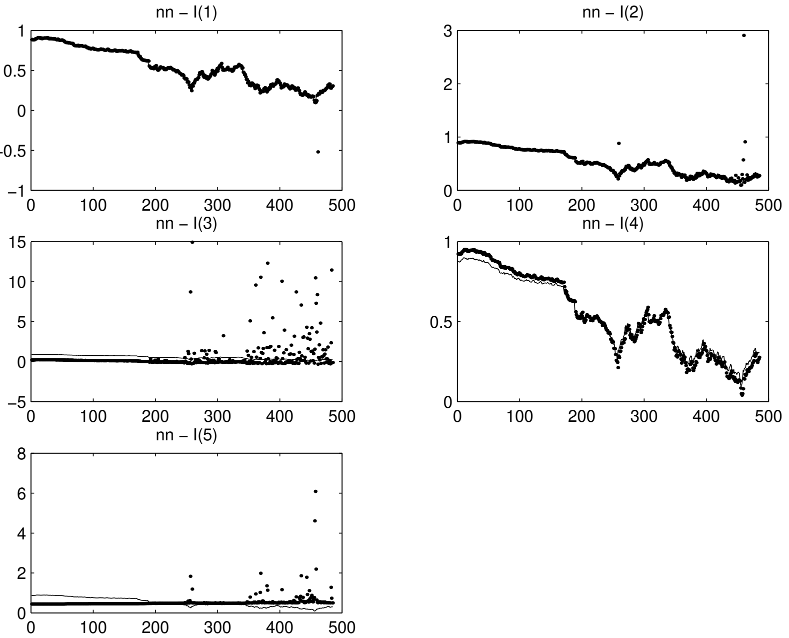

Figure 6. $nn(5,3)$ network output minus input of one input layer cell ($I1$ till $I5$) compared with actual data.

So, 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, such an 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:

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

or written as switching model:

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

where

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

$$\tag{22}$$

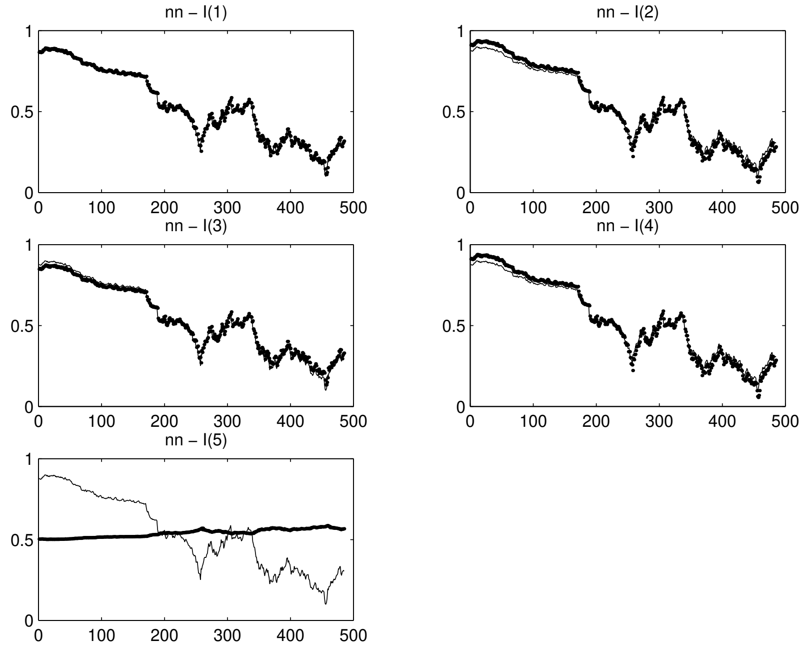

Figure 7. $nn(5, 1)$ network output minus input of one input layer cell ($I1$ till $I5$) compared with actual data.

The small $nn(1, 1)$ seems to work properly, so this type of neural network is used to fit 5 other US dollar real exchange rates: namely the Canadian dollar, the French franc, the UK pound, the German (west) mark and the Dutch guilder.

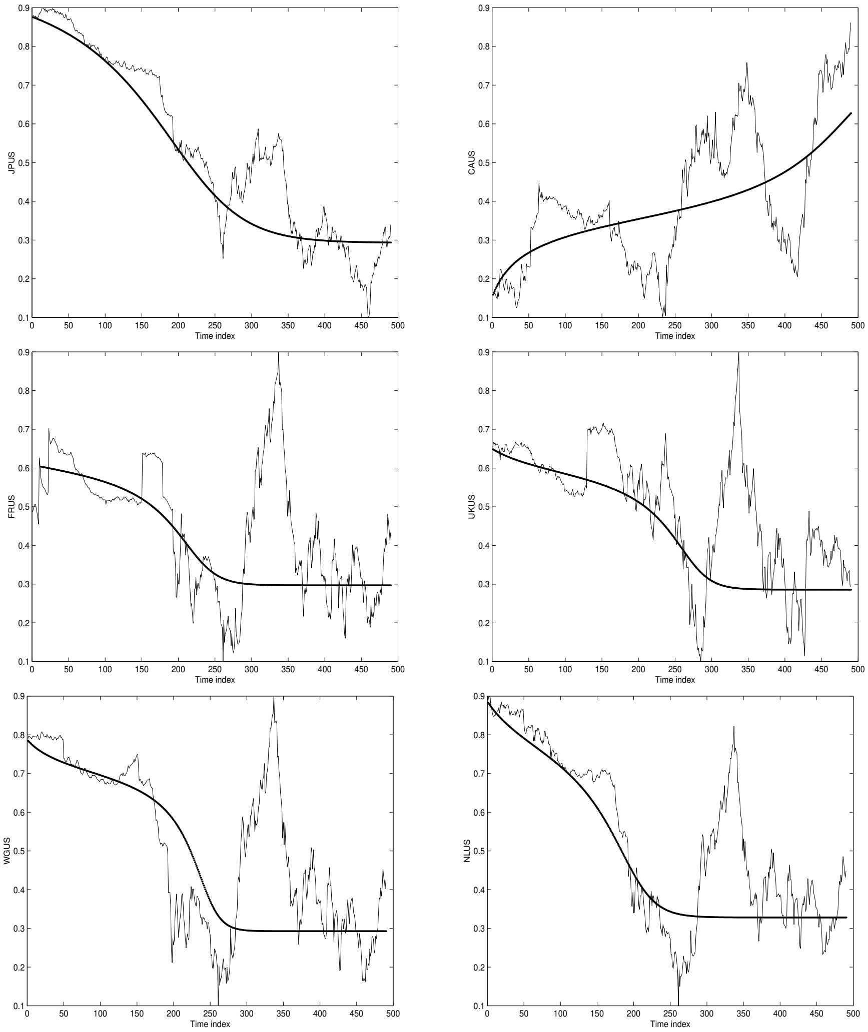

Figure 8. Logarithm of US real exchange rates (Japan, Canada, France, United Kingdom, Germany (West), Netherlands); real data and neural network $nn(1,1)$ orbit (thick lines) 1957-1998

The resulting networks can be used for prediction. As an example, the network is used to generate a data series, called orbit, in the following way. In the case of $nn(1,1)$ the orbit data $\hat{y}_t$ are generated by the system equation:

$$\hat{y}_t = nn(\hat{y}_{t-1}) \tag{23}$$

where nn represents the neural network function; see equation (19). Only the starting value y_1 is taken from actual data (except for orbit $FRUS$ where the starting value is equal to y_{12}). In figure (8) the resulting neural network orbit with the actual data are exposed.

The results indicate that in all cases a nonlinear 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).

5. Summary

In this paper the number of parameters in a neural network is reduced by applying elementary procedures. The procedure calculates the incremental contribution of variables, in the case of neural networks: the hidden layer cells and input layer cells. Another descriptive measure, the principal component analysis of residuals with one cell omitted, confirms the in- or exclusion reasoning based on incremental contributions. The advantage of our proposed principal component procedure is that in 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.

Our empirical analysis shows substantial evidence on nonlinear trends in the real exchange rates. As such, these results may give rise to the search for parsimonious models that give an adequate description of the observed data; see e.g. Granger and Teräsvirta (1993).

Acknowledgements

The authors wish to thank Timo Teräsvirta, Stockholm School of Economics, for helpful discussions.

References

Bishop, C.M., *Neural Networks for Pattern Recognition*, Clarendon Press Oxford, 1995.

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

Granger, C.W.J. & T. Teräsvirta, *Modelling Nonlinear Economic Relationships*, Oxford University Press Inc., New York, 1993.

Kaashoek, J.F. & H.K. van Dijk, A simple strategy to prune neural networks with an application to economic time series, Report 9854/A, Econometric Institute Erasmus University Rotterdam, 1998.

Malinvaud, E., *Statistical Methods of Econometrics*, North-Holland Publ. Co., Amsterdam London, 1970.

Mozer, M.C. & P. Smolensky, Skeletonization: a technique for trimming the fat from a network via relevance assessment, in D.S. Touretzky (Ed.) *Advances in Neural Information Processing Systems*, vol. 1, San Mateo, CA., 1989.

Press, W.H., B.P. Flannery, S.A. Teukolsky & W.T. Vetterling, *Numerical Recipes*, Cambridge University Press, Cambridge, 1988.

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

Takens, F., Detecting strange attractors in turbulence, in D.A. Rand and L.S. Young (eds.), *Dynamical systems and turbulence*, Springer-Verlag, Berlin, 1981.

Theil, H., *Principle of Econometrics*, Wiley & Sons, 1971

White, H., Some Asymptotic Results for Learning on Single Hidden Layer Feedforward Network Models, *Journal of the American Statistical Association*,vol.**84**, no.408, 1989.