A simple strategy to prune neural networks with an application to economic time series

by

Johan F. Kaashoek and Herman K. van Dijk

Econometric Institute and Tinbergen Institute
Erasmus University Rotterdam
P.O.Box 1738, 3000 DR Rotterdam
The Netherlands

Abstract

A major problem in applying neural networks is specifying the size of the network. Even for moderately sized networks the number of parameters may become large compared to the number of data. In this paper network performance is examined while 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 fame to handle complex data may have contributed considerably to the diffusion and implementation of neural network models, also in economics and econometrics; see e.g. Hecht-Nielsen [1990], Hertz, Krogh and Palmer [1991], Gallant and White [1988] and White [1989].

In this paper only one type of neural network is considered: a feed-forward 3-layer network with an input layer of I cells (nodes), a hidden layer with H cells (nodes) and an output layer with O cells; the network will be denoted as $nn(I, H, O)$. If

x_{ti} is the input of input cell i at time index t, then the output O_{th} of hidden layer cell h is given as:

$$O_{th} = g(\sum_{i=1}^{I} a_{hi} x_{ti} + b_h) \tag{1}$$

where g is a non-linear activation function, e.g.

$$g(x) = \frac{1}{1+e^{-x}}. \tag{2}$$

Then the value y_{to} of output cell o is given as:

$$y_{to} = c_h O_{th} + d. \tag{3}$$

Hence, the functional form of this network is

$$y_t = \sum_{h=1}^{H} c_h g(\sum_{i=1}^{I} x_{ti} a_{ih} + b_h) + d \tag{4}$$

If, for notation convenience the time index t is dropped, this neural network can be written in matrix form as:

$$y = C\, G(A\, x + b) + d, \tag{5}$$

with A an $H \times I$ matrix, X (for each t) an I dimensional vector of input data, b an H vector, C an $O \times H$ matrix and d an O vector; $G : \mathbb{R}^H \rightarrow \mathbb{R}^H$ is a multi-valued non-linear (activation) function.

It is assumed that this neural network is an approximation of the data generating process:

$$x_t = F(x_{t-1}, \cdots, x_{t-N}) + \epsilon_{t-1}, \tag{6}$$

where $x_t \in \mathbb{R}$, F the data generating function and ϵ_{t-1} represents an unknown noise term. Hence the size O (dimension) of the output layer is a priori given and equal to 1. An upperbound N on the size of the input layer is given by nonlinear data analysis (e.g. embedding dimension, see Takens [1981]) but the size H of the hidden layer is unknown and has to be determined.

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, with unlimited number of hidden layer cells H, the network output encompasses the spectrum of y.

A three fold procedure is applied in reducing the network size. The procedures are not restricted to the specific neural network used in this paper.
The starting-point is what is called by Theil Theil [1971]: the incremental contribution of variables. That is, how much more of the variance of the dependent variable y is explained by inclusion of e.g. the hth explanory variable given that all other variables are used. In Theil (o.c) the incremental contribution is measured in terms of the multiple correlation coefficients. A variable with a low incremental contribution will be a candidate to be excluded from the model.

Secondly, we calculate the principal components of the set of residuals obtained by omitting successively one network node. The vector representing the first principal component, may reveal which node can be excluded from the network.
Thirdly, graphical analysis of network output with exclusion of network nodes is used as justification for in- or exclusion of variables with low incremental contribution.
As a node pruning method, this approach is similar to the one proposed by Mozer and Smolensky [1989]. Our approach although conservative has the advantage that the quantities used are based on the outcome of only one optimization procedure with all variables included.

The paper is organized as follows. In the first part the term incremental contribution and multiple correlation coefficients are briefly explained. The second part explains the principal component analysis of residuals. In the third part pruning of a network based on the incremental contribution of variables and principal component analysis is introduced. Also the possibility of graphical analysis is mentioned shortly.

In the fourth part the procedures are applied to three examples; in the first one the data are generated by a completely deterministic process; in the second one the data are generated by a neural network ("a true neural network") itself while the third example concerns actual economic data: the logarithm of Yen-US Dollar real exchange rates.

2 Incremental contribution of variables

Consider a simple neural network, the linear model (no hidden layer and g, the activation function, be the identity function):

$$y = X\,a + b + \epsilon, \tag{7}$$

with $y \in \mathbb{R}$, a a vector of unknown parameters with length H, b a constant term and ϵ some stochastic process. The multiple correlation coefficient R associated with the least squares regression of y on $(X\,1)$ is given as:

$$R^2 = 1 - \frac{e'e}{(y-\overline{y})'(y-\overline{y})} \tag{8}$$

where $\overline{y}$ is the mean of y and e represents the residuals of the least squares regression:

$$\begin{aligned} \hat{y} &= X\,\hat{a} + \hat{b}, \\ e &= y - \hat{y} \end{aligned} \tag{9}$$

where $\hat{a}$ and $\hat{b}$ are the estimates of a resp. b.
Now regress again with the hth variable not included. The residuals obtained in this way are denoted as e_h and the multiple correlation coefficient R_h is defined by:

$$R_h^2 = 1 - \frac{e_h'e_h}{(y-\overline{y})'(y-\overline{y})}. \tag{10}$$

The quantity $R^2 - R_h^2$ is called the incremental contribution of the variable h in explaining the variance of y; see Theil [1971].

Now, suppose again $\hat{a}$ and $\hat{b}$ are estimates with all variables included, then by putting the hth variable to zero, one gets residuals e_{-h}:

$$\begin{aligned} \hat{y}_{-h} &= X_{-h}\hat{a}_{-h} + \hat{b} \\ e_{-h} &= y - \hat{y}_{-h} \end{aligned} \tag{11}$$

where X_{-h} is data set X without the hth variable, and $\hat{a}_{-h}$ is equal to the vector of estimates $\hat{a}$ minus the hth parameter $\hat{a}_h$; only the constant term $\hat{b}$ is "re-estimated" such that the vector e_{-h} has zero mean.

The multiple correlation coefficient R_{-h} is now calculated as in equation (10).
The contribution of the hth variable is measured as:

$$R_{inc}^2 = R^2 - R_{-h}^2 = \frac{e_{-h}'e_{-h} - e'e}{(y-\overline{y})'(y-\overline{y})} \tag{12}$$

The exclusion of variables from the model is now based on their incremental contributions $R^2 - R_{-h}^2$. Variables with low contributions are the first candidates to be excluded.
Note that the contribution measure (12) is *conservative* because

$$e_{-h}'e_{-h} \geq e_h'e_h \tag{13}$$

where e_h are residuals of a regression of the model without the hth variable. Hence, the R^2_{-h} will be larger than the one based on an additional regression with all variables included except the hth variable.

Finally, instead of the definition of R^2 given in equation (8) and that of R^2_{-h} in equation (10), we will use the square of the correlation coefficient of y and $\hat{y}$, respectively $\hat{y}_h$; hence:

$$R^2 = \frac{(\hat{y}'y)^2}{(y'y)(\hat{y}'\hat{y})} \tag{14}$$

$$R^2_{-h} = \frac{(\hat{y}'_{-h}y)^2}{(y'y)(\hat{y}'_{-h}\hat{y}_{-h})}. \tag{15}$$

So the incremental contribution of a variable is measured as

$$R^2_{inc} = \frac{(\hat{y}'y)^2}{(y'y)(\hat{y}'\hat{y})} - \frac{(\hat{y}'_{-h}y)^2}{(y'y)(\hat{y}'_{-h}\hat{y}_{-h})}. \tag{16}$$

3 Principal component analysis of residuals

Starting point is again the linear model - see equation (7) - and more specific the residuals e_{-h} defined in equation (11):

$$e_{-h} = y - X_{-h}\hat{a}_{-h} - \hat{b} \tag{17}$$

where, as before, $\hat{A}_{-h}$ are the estimates of A without hth variable. All those residuals e_{-h} are put together in one $T \times H$ matrix E_{-H},

$$E_{-H} = (e_{-1}, e_{-2}, \cdots, e_{-H}), \tag{18}$$

where H is dimension of A: the number of parameters without constant term.
Now one can calculate the principal components of E_{-H}.
A principal component p is a $T \times 1$ vector which is a linear combination with factors $\sim v$ of the columns of E_{-H}. Especially, the so-called first principal component gives the *best* linear combination of columns of E_{-H} in least squares sense; see Theil [1971].
To be more specific, the first principal component is the solution of the following quadratic optimalisation problem (subject to some normalisation of p, i.e. $p'p = 1$):

$$\min_{v,p} \sum_{i=1}^{T} \sum_{j=1}^{-H} ([E_{-H} - pv']_{ij})^2 \tag{19}$$

which is equivalent to:

$$\min_{v,p} \operatorname{tr}(E_{-H} - pv')'(E_{-H} - pv') \tag{20}$$

The first order condition of (20) in the vector of factors v is:

$$v = E'_{-H}p, \text{ for every } p \neq 0. \tag{21}$$

Hence, the original minimalisation problem (20) becomes the maximalisation problem:

$$\max_{p} \; p'E_{-H}E'_{-H}p, \text{ subject to } p'p = 1. \tag{22}$$

With condition $p'p = 1$, the first order condition of (22) is the solution of the eigenvalue problem

$$(E_{-H}E'_{-H} - \lambda I)p = 0. \tag{23}$$

with eigenvector λ maximal.
Now, using equation (21), it follows from (23) that

$$(E'_{-H}E_{-H} - \lambda I)v = 0. \tag{24}$$

So, one finds the $H \times 1$ vector v of factors constituting the vector p as the eigenvector of $E'_{-H}E_{-H}$ at largest eigenvalue λ. Moreover, by using equation (21) again, it follows that

$$p = E_{-H}\, v/\lambda \tag{25}$$

where λ is the largest eigenvalue of the symmetric, positive semi-definite matrix $E'_{-H}E_{-H}$.

Based on equation (25) all other principal components can be calculated by getting all other the orthogonal eigenvectors v at eigenvalues λ of the symmetric, positive semi-definite matrix $E'_{-H}E_{-H}$.
All principal components are ordered according the eigenvalues λ of $E'_{-H}E_{-H}$ in descending order.

The use of principal components is motivated by the following statement: the relative 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} 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 variables. And moreover, the elements of this vector v_{max} reveal

which variable may be omitted and which variable must be included in any case. The variable x_h with index h for which the corresponding element in the first principal component is minimal in absolute sense, can be excluded: its exclusion of the model does not contribute very much to the worst case!
And reverse, the variable x_h corresponding to the element in the principal component with maximal absolute value, must not be excluded from the model because its exclusion has the highest contribution 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:

$$w_k = \lambda_k / \sum_{k=1}^{H} \lambda_k. \tag{26}$$

A similar (but reverse) reasoning and interpretation can be applied to the last principal component at the smallest eigenvalue λ_{min} of $E'_{-H}E_{-H}$ (with smallest variance). But the first principal component gives directly the relative importance of the elements of e_{-h} and consequently the relative importance of the elements of X with respect to the explanation of y. This relative ordering is given by the absolute value of the elements of v_{max}, the factors of the *first* principal component.

The advantage of principal component analysis of residuals is given by the fact that each component, from first to last, gives the relative importance of each variable as its contributes to the component. So, principal components analysis give an overall picture in contrast to incremental contributions in R^2.
The decision whether a variable can be excluded, must be based not only on the relative weight of variables in the first (or, inversely, in the last) principal component, but also the condition number of the matrix $E'_{-H}E_{-H}$, e.g. the relative weight of each component given by its corresponding eigenvalue.

4 Network pruning

The functional form of the network used, see equation (4), can be written as:

$$y = h'\,c + d,\ \ c, h \in \mathbb{R}^H;\ d \in \mathbb{R}, \tag{27}$$

$$h_k = g((A\,x)_k + b_k), k = 1, \cdots, H. \tag{28}$$

with $h = (h_1, \cdots, h_H)'$ is a vector denoting the output of hidden layer cells and where g is an activation function; e.g. $g(x) = \frac{1}{1+e^{-x}}$.

In the following three sections, the procedures explained in the fore going sections and graphical analysis are implemented as neural network pruning mechanism.

4.1 Incremental contributions of network layer cells

Suppose some time series $\{y(t), x(t)\}$ are to be fitted to the network given by (27). After some optimization procedure, for each hidden layer cell the incremental contribution to R^2 can be calculated (given A, b and d). Let $\hat{y}$ be the network output with inclusion of all cells, $\hat{y}_{-h}$ network output with hidden layer cell h excluded, then the incremental contribution of hidden cell h is again:

$$R^2_{inc} = R^2 - R^2_{-h} = \frac{(\hat{y}'y)^2}{(y'y)(\hat{y}'\hat{y})} - \frac{(\hat{y}'_{-h}y)^2}{(y'y)(\hat{y}'_{-h}\hat{y}_{-h})}. \tag{29}$$

If the value R^2_{inc} in (29) is low for some h compared to all other values, then this cell is a candidate for exclusion from the network. This decision can be confirmed by the graphical analysis explained above.

If some cell h is to be excluded from the network and corresponding parameters are to be deleted, the optimization procedure is prolonged with all other parameters unchanged except the parameter d which is adjusted for mean differences between original output data $\{y_t\}$ and network output $\{\hat{y}_{-h}(t)\}$[1].

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, with exclusion of input cell i. Now

$$R^2_{inc} = R^2 - R^2_{-i} = \frac{(\hat{y}'y)^2}{(y'y)(\hat{y}'\hat{y})} - \frac{(\hat{y}'_{-i}y)^2}{(y'y)(\hat{y}'_{-i}\hat{y}_{-i})}. \tag{30}$$

And again, the relative value of incremental contributions R^2_{inc} in R^2, can be used in evaluating whether an input cell van be omitted. However, in general,

[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.

economic considerations will determine which variables are to be included; in the case of a time series model as in (6) the number of input cells equals the number of lagged variables used. For instance in the case of pure deterministic time series, the number of input cells is bounded from above by the embedding dimension of the observed series $\{x_t\}$; see e.g. Takens [1981]. However, even if this embedding dimension can be determined, it is only an upperbound and further econometric analysis can be applied which can lead to a reduction of the number of input cells.

4.2 Principal components analysis of network residuals

The application of principal component analysis as exposed in section 3 is straightforward.
For the hidden layer cells we define the matrix:

$$\begin{aligned} E_{-H} &= (e_{-1}, e_{-2}, \cdots, e_{-H}), \\ e_{-h} &= y - \hat{y}_{-h}. \end{aligned} \tag{31}$$

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. The first principal component corresponding with maximal eigenvalue, will have maximal variance.
The relative weight of each component is calculated according to equation (26). The absolute value of the elements of this first principal component may decide whether a hidden layer cell can be excluded: the lower this value, the more probable the cell can be omitted. Whether this decision can be based on the factors (eigenvector) of the first principal component only, will depend on the relative weight of this component.

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

$$\begin{aligned} E_{-I} &= (e_{-1}, e_{-2}, \cdots, e_{-I}), \\ e_{-i} &= y - \hat{y}_{i}. \end{aligned} \tag{32}$$

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

4.3 Graphical analysis

The statistical procedures explained above has a straightforward graphical counterpart. An obvious way to look at neural network performance is to compare the graphs of original output data and neural network estimates.

In a similar way, the graphs of $\{t, \hat{y}_{-h}(t)\}$ compared to the graph of $\{t, y(t)\}$ will give evidence of the contribution of hidden cell h in explaining the variance of $y(t)$. The best way for comparison is to adjust all series $\{y_t\}$ and $\{\hat{y}_{-h}(t)\}, h = 1, \cdots, H$ for mean. With the network configuration given by equation (4) this is no problem as the constant d can always take care of mean differences.

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 of in- or exclusion of input cell i.

5 Three examples

In this section two examples are given which illustrate the pruning method explained above. In both cases the aim is to reconstruct a multi dimensional data generating process while only one component is observed.
If $\{x_t\}$ is the only observed (univariate) component then a I dimensional delay vector $x_{t-1}^I = (x_{t-1}, x_{t-2}, \cdots, x_{t-I})$ is constructed where I is the so-called embedding dimension; see Takens [1981]. Now, the original data generating process may be reconstructed by the dynamical system

$$x_t^I = F(x_{t-1}^I) \tag{33}$$

where $F = (F_1, F_2, \cdots, F_I) : \mathbb{R}^I \to \mathbb{R}^I$.
Since for $1 < k \leq I$ holds:

$$x_t^I[k] = x_{t-k} = F_{k+1}(x_{t-1}^I) \tag{34}$$

hence

$$F_{k+1}(x_{t-1}^I) = x_{t-k} \tag{35}$$

the reconstruction process involves only the estimation of the unknown first component of F; i.e.

$$x_t^I[1] = F_1(x_{t-1}^I). \tag{36}$$

Or written in the original observed data:

$$x_t = F_1(x_{t-1}, x_{t-2}, \cdots, x_{t-I}). \tag{37}$$

The function F_1 is now to be approximated by a neural network with input dimension I.

5.1 Deterministic data

The first example has to do with data generated by the following deterministic process:

$$\begin{aligned} x_t &= 0.95\, x_{t-1} + (\epsilon_{t-1} - 0.5) \\ \epsilon_t &= 4\, \epsilon_{t-1}(1 - \epsilon_{t-1}) \end{aligned} \tag{38}$$

So, the series x_t is an AR-process with disturbances $\epsilon_{t-1} - 0.5$ generated by the logistic map; as is well-known, the series ϵ_t is chaotic.
The observed data are the series $\{x_t\}$ only. The number of data used is 200. The data set is called by AR_{CH}.

The process (38) is to be reconstructed using a neural network. From non-linear data analysis - see Takens [1981] - it follows that an upperbound on the embedding (number of lags) is given by $2\,N+1$ where N is the dimension of the original data generating process; in our case $N=2$.
So, starting point is the unknown relation:

$$x_t = F_1(x_{t-1}, \cdots, x_{t-I}) \tag{39}$$

where F_1 is to be approximated by a neural network with input dimension I equal to 4, one less than the upperbound 5, and a constant input variable representing the constant term b.

The dimension of the hidden layer is taken to be 5; hence $H=5$.

As "learning" method a non-linear optimization procedure is applied; to be more specific, a variable metric method known as Davidon-Fletcher-Powell - see Press et al. [1988] - with object function the sum of squared residuals. The results of the optimization can be found in table 1. The first column of the table gives the cells which are excluded: Hj means cell j of the hidden layer; Ij similar but now for the input layer. Apart from the multiple correlation coefficients and their incremental contribution, the so-called information criterion, SIC, are reported in the tables.

$$SIC = \ln(MSSR) + \frac{n_p}{2T}\ln(T), \tag{40}$$

where n_p is the number of parameters, T is the length of data set and $MSSR$ is the mean sum of squared residuals; see Schwartz [1978]. The SIC value measures abundance of parameters.
Because the relation between input cells and output cell are non linear, and the correlation coefficients are calculated according to equation (14) or equation (15), the SIC values may be not consistent with the correlation coefficients. This is not the case for hidden layer cells since the relation between those cells and the output cell is linear.
In the same table the principal components are presented, or to be correct, the eigenvectors of $E'_{-H}E_{-H}$ and their proportional weights based on the eigenvalues; see equation (26). As we start our reduction procedure at the hidden layer, the principal components of residuals with one input cell excluded, are not reported here.

Table 1: Multiple correlation coefficient, incremental contribution and principal components

Network $(4, 5, 1)$ and Data: AR_{CH}

Cell excluded	Multiple correlation	Incremental contribution	SIC
None	0.9999	.	-6.1688
H1	0.9956	0.0043	-1.7003
H2	0.2023	0.7976	0.4258
H3	0.0001	0.9999	0.1248
H4	0.9972	0.0028	-2.4759
H5	0.9953	0.0047	-2.1077
I1 (x_{t-4})	0.9995	0.0004	-5.1765
I2 (x_{t-3})	0.9943	0.0057	-2.7150
I3 (x_{t-2})	0.7142	0.2857	0.9669
I4 (x_{t-1})	0.4838	0.5161	0.7808

Principal Components on hidden layer cells:
Eigenvectors of $E'_{-H}E_{-H}$

H1	0.2104	0.7783	-0.5720	-0.1512	-0.0077
H2	-0.0111	0.0151	0.1987	-0.7234	0.6610
H3	-0.0004	0.0027	0.1816	-0.6357	-0.7503
H4	-0.8612	0.4336	0.2513	0.0838	-0.0082
H5	0.4625	0.4539	0.7330	0.2069	0.0035
Proportional weights :					
	0.00%	0.05%	0.05%	8.12%	91.72%

From table 1 it is obvious that hidden layer cells 1, 4 and 5 can be excluded; see the first and also the last principal components in table 1. The first principal component (relative importance 91.72%) shows that the hidden cells 2 and 3 have almost equal weights. No graphs are supplied because the values of the incremental contribution and first principal component (and also the relative SIC-values) don't need further illustration. So hidden cells 1, 4 and 5 are excluded from the net; the constant term d is adjusted for mean difference between original output data and network output. A further optimization run is applied to the reduced network. The results are reported in table 2.

Table 2: Multiple correlation coefficient, incremental contribution and principal components

Network $(4, 2, 1)$ and Data: AR_{CH}

Cell excluded	Multiple correlation	Incremental contribution	SIC
None	0.9999	.	-6.5892
H1	0.1410	0.8588	-0.8593
H2	0.0009	0.9997	-0.7833
I1 (x_{t-4})	0.9998	0.0001	-5.8892
I2 (x_{t-3})	0.9987	0.0011	-4.9207
I3 (x_{t-2})	0.8532	0.1465	-0.1782
I4 (x_{t-1})	0.4121	0.5877	-1.8157

Principal Components on hidden layer cells:
Eigenvectors of $E'_{-H}E_{-H}$:

H1	0.7356	-0.6775
H2	0.6775	0.7356

Proportional weights

	3.82%	96.18%

Principal Components on input layer cells:
Eigenvectors of $E'_{-I}E_{-I}$:

I1 (x_{t-4})	0.9983	0.0164	-0.0084	-0.0007
I2 (x_{t-3})	0.0165	-0.9998	0.0077	0.0025
I3 (x_{t-2})	0.0012	0.0020	-0.0652	0.9979
I4 (x_{t-1})	0.0083	0.0080	0.9978	0.0652

Proportional weights:

	0.00%	0.00%	3.24%	96.76%

As the contribution of each hidden layer cell and the elements of the eigenvector at proportional weight 96.18% are (almost) equal, no further reduction in hidden layer cells is applied.

However, input cells 1 and 2 are obvious candidates for further exclusion; see also the principal component analysis applied on input layer cells in table 2 and figure (1). So the network is reduced to 2 input cells and 2 hidden layer cells.

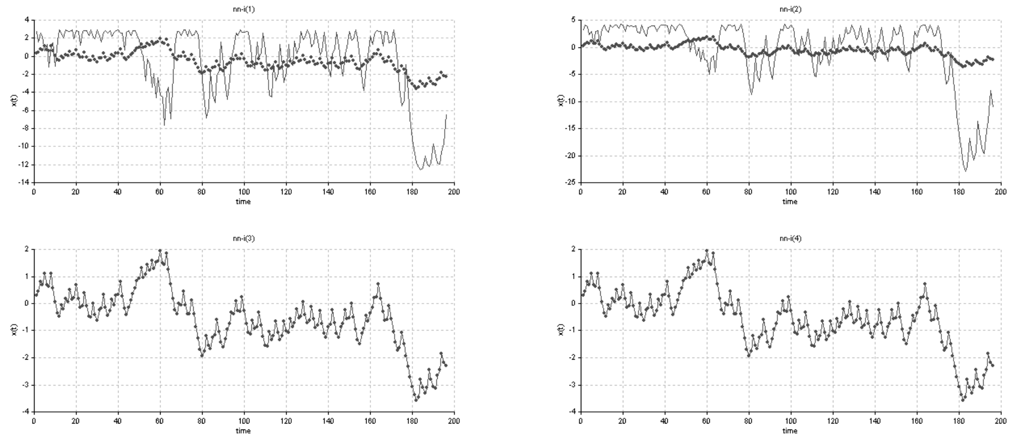

Figure 1: AR_{CH} data (thick dots) and $nn(4, 2, 1)$ network outputs without input cell 1, 2, 3 and 4 (top left to bottom right.

After optimization the results are those reported in table 3.

No further attempts of reduction are applied: the contribution of each cell in a layer is similar and the first principal component analysis on residuals minus one input cell, see table 3, does not show a component with a very small value.

Two graphs show the performance of the final network $nn(2, 2, 1)$. The first graph gives the time series of the actual data $\{x_t\}$ and the neural network estimates $\{\hat{x}_t\}$, the one period ahead prediction, see figure (2).

The second graph compares the series $\{x_t\}$ with a series nn_t which is generated as follows: the initial value is given as $nn_t = x_t, t = 1, \cdots 2$, and from that all other values $nn_t, t > 2$ are generated by the neural network. This time series is denoted as "a neural network generated orbit". Since such a "neural network generated orbit" may start at any value (= time index) and can be prolonged for any time period, it shows the prediction capability of the network function over any period at any time; see figure (3).
The orbit generated by the network function deviates from the actual data after 5 time steps. Note that the "error" $\epsilon_{t-1} - 0.5$ in equation (38) is generated by a

Table 3: Multiple correlation coefficient, incremental contribution and principal components

Network $(2,2,1)$ and Data: AR_{CH}

Cell excluded	Multiple correlation	Incremental contribution	SIC
None	0.9999	.	−6.5892
H1	0.1169	0.8830	1.6399
H2	0.0004	0.9995	0.1402
I1 (x_{t-2})	0.8641	0.1359	0.9877
I2 (x_{t-1})	0.7540	0.2460	−0.8832

Principal Components on input layer cells:
Eigenvectors of $E'_{-I}E_{-I}$:

I1 (x_{t-2})	0.4113	−0.9115
I2 (x_{t-1})	−0.9115	0.4113

Proportional weights:

	5.30%	94.70%

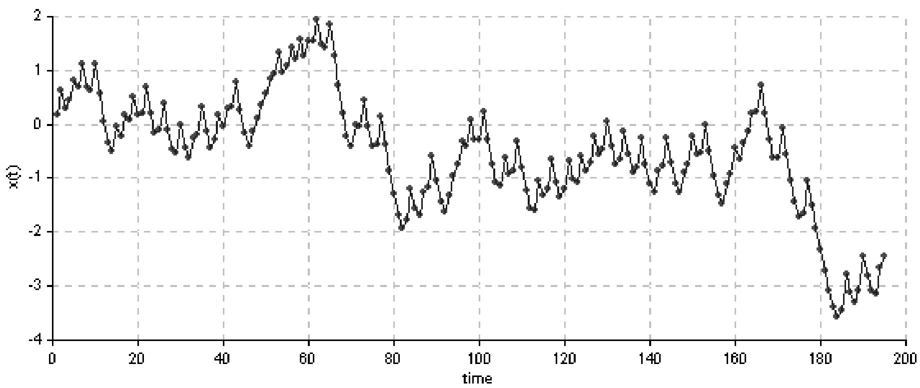

Figure 2: AR_{CH} data (dots) and neural network estimates (continuous curve)

structural unstable (and chaotic) system: small deviations in the original function form of ϵ_t can result in considerable and essential deviations in orbits; see e.g. Devaney [1989]. As the neural network approximates the original system, such deviations in prediction are not surprising.

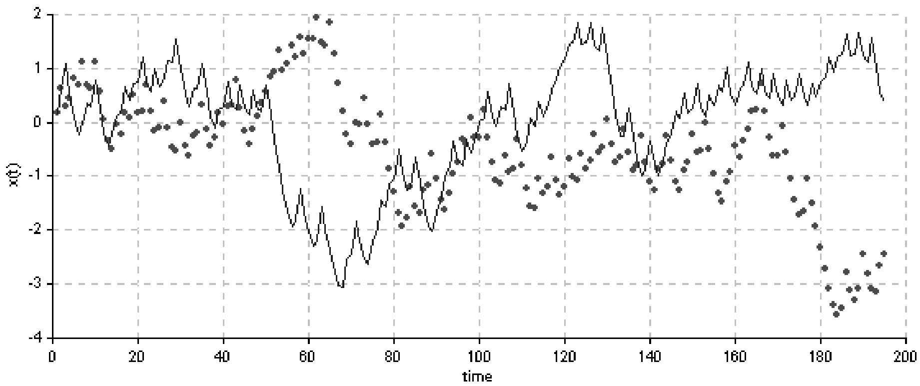

Figure 3: AR_{CH} data (thick dots) and orbit generated by neural network (continuous curve)

5.2 A true neural network

The data used in this section are according to a two dimensional model:

$$\begin{aligned} x_t[2] &= x_{t-1}[1] \\ x_t[1] &= F_1(x_{t-1}[1], x_{t-1}[2]). \end{aligned} \tag{41}$$

with F_1 is the function $\mathbb{R}^2 \rightarrow \mathbb{R}$ given by the above found neural network $nn(2, 2, 1)$. The observed data, denoted as $NN0202$, are again only one dimensional: $\{x_t(1)\}$; the length of the data is 500. So the first two hundred data of this set are just the data plotted in figure (3). Applying the same procedure as explained above - see page 11- the original (true) neural network is to be found again starting with a neural network with 4 inputs (embedding dimension 4) and 6 hidden layer cells. The result of an optimization are reported in table 4.

Table 4: Multiple correlation coefficient, incremental contribution and principal components

Network $(4, 6, 1)$ and Data: $NN0202$

Cell excluded	Multiple correlation	Incremental contribution	SIC
None	0.9999	.	-8.7876
H1	0.9999	0.0000	-8.4503
H2	0.8235	0.1764	0.3966
H3	0.4064	0.5934	0.9796
H4	0.9087	0.0912	1.1715
H5	0.9954	0.0045	1.2118
H6	0.9698	0.0112	2.4198
I1 (x_{t-4})	0.9999	0.0000	-8.7232
I2 (x_{t-3})	0.9999	0.0000	-8.1238
I3 (x_{t-2})	0.0420	0.9579	1.9043
I4 (x_{t-1})	0.0065	0.9934	1.9105

Principal Components on hidden layer cells:

Eigenvectors of $E'_{-H}E_{-H}$:

H1	1.00	0.00	0.00	0.00	0.00	-0.00
H2	0.00	-0.12	-0.01	-0.85	0.02	0.50
H3	-0.00	0.01	-0.01	-0.51	-0.02	-0.86
H4	-0.00	-0.00	0.70	-0.01	0.71	-0.02
H5	-0.00	0.99	-0.00	-0.10	0.00	0.07
H6	0.00	-0.00	-0.71	0.01	0.70	-0.02
Proportional weights :						
	0.00%	0.00%	0.21%	2.18%	27.31%	70.30%

Principal Components on input layer cells:

Eigenvectors of $E'_{-I}E_{-I}$:

I1 (x_{t-4})	0.75	-0.66	0.00	-0.00
I2 (x_{t-3})	-0.66	-0.75	-0.00	0.00
I3 (x_{t-2})	-0.00	-0.00	-0.72	-0.69
I4 (x_{t-1})	-0.00	-0.00	0.69	-0.72
Proportional weights :				
	0.00%	0.00%	7.81%	92.19%

From table (4) it is obvious that input cells 1, x_{t-4}, and 2, x_{t-3}, can be excluded. 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. This gives a network with 2 inputs and 4 hidden layer cells. In table 5 the results of an further optimization run are reported.

Table 5: Multiple correlation coefficient, incremental contribution and principal components

Network $(2,4,1)$ and Data: $NN0202$

Cell excluded	Multiple correlation	Incremental contribution	SIC
None	0.9999	.	−5.5245
H2	0.4408	0.5592	2.2549
H3	0.0435	0.9565	2.3557
H4	0.9321	0.0678	2.0238
H6	0.9365	0.0635	3.3106
I1 (x_{t-2})	0.0140	0.9860	2.2048
I2 (x_{t-1})	0.0052	0.9948	2.1429

Principal Components on hidden layer cells:

Eigenvectors of $E'_{-H}E_{-H}$:

H1	0.00	0.74	0.04	0.68
H2	0.00	0.68	−0.02	−0.74
H3	−0.70	−0.01	0.71	−0.03
H4	−0.71	−0.01	0.70	−0.03
Proportional weights :				
	0.01%	0.58%	5.84%	93.58%

Principal Components on input layer cells:

Eigenvectors of $E'_{-I}E_{-I}$:

I1 (x_{t-2})	−0.68	0.73
I2 (x_{t-1})	0.73	−0.68
Proportional weights :		
	7.20%	92.80%

Table 5 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.

5.3 Economic data

The data used in this section are the logarithm of Yen-US dollar real exchange rates, period December 1972 to June 1988, denoted as $JPUS$; for more details on the data, see Schotman and van Dijk [1991] who fitted a linear auto-regressive model of order one, $AR1$. The data are shown in figure 4.

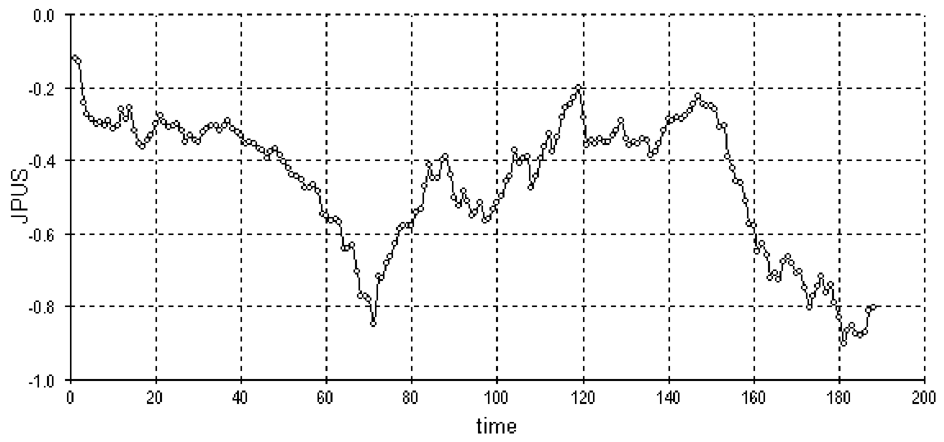

Figure 4: $JPUS$ data

The initial network is rather large: $nn(5,7,1)$. In Kaashoek and van Dijk [1994] it is shown that the series $JPUS$ has an embedding dimension of 5 at the most; this means that only variables with a lag of 5 or less needs to be included.

The results of optimization are summarized in table 6 where only the principal components (eigenvectors of $E'_{-H}E_{-H}$ and their proportional weights) of hidden layer residuals are given.

From table 6 one can conclude that hidden layer cells 3 and 6 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 99.65%) are for those cells again very low. This is further illustrated in figure 5 and figure 6. In this two figures, the neural network output with exclusion of one hidden layer cell each time is compared with actual data $JPUS$. In each figure the neural network output is always adjusted for mean differences.
With respect to the input layer cells, according to the results reported in table 6 all input variables has to be included (so far). Especially, input layer cell 1, variable x_{t-1} and input layer cell 5, variable x_{t-5}, have in this configuration a large contribution, and can not be excluded.

Table 6: Multiple correlation coefficient, incremental contribution and principal components
Network $(5,7,1)$ and Data: $JPUS$

Cell excluded	Multiple correlation	Incremental contribution	SIC
None	0.9810	.	-2.9997
H1	0.0075	0.9735	1.7740
H2	0.1722	0.8088	-0.3818
H3	0.9377	0.0433	-1.3314
H4	0.6236	0.3574	-0.2242
H5	0.2578	0.7232	0.0876
H6	0.9698	0.0112	-2.5207
H7	0.0028	0.9781	1.7461
I1 (x_{t-5})	0.0108	0.9702	-0.9798
I2 (x_{t-4})	0.5728	0.4082	-1.0869
I3 (x_{t-3})	0.4727	0.5083	-1.3049
I4 (x_{t-2})	0.4512	0.5298	-0.9664
I5 (x_{t-1})	0.3641	0.6169	-1.2264

Principal Components on hidden layer cells:
Eigenvectors of $E'_{-H}E_{-H}$:

H1	-0.21	0.25	0.13	0.50	0.35	-0.03	-0.71
H2	0.14	0.10	-0.05	-0.23	0.48	0.82	0.02
H3	0.09	-0.85	0.42	0.20	0.22	0.04	-0.00
H4	-0.37	0.20	0.72	-0.54	0.04	-0.09	-0.04
H5	-0.20	0.05	0.26	0.38	-0.68	0.54	0.04
H6	0.83	0.31	0.43	0.12	-0.06	-0.08	-0.00
H7	-0.23	0.25	0.16	0.46	0.38	-0.09	0.70
Proportional weights :							
	0.00%	0.01%	0.03%	0.04%	0.05%	0.22%	99.65%

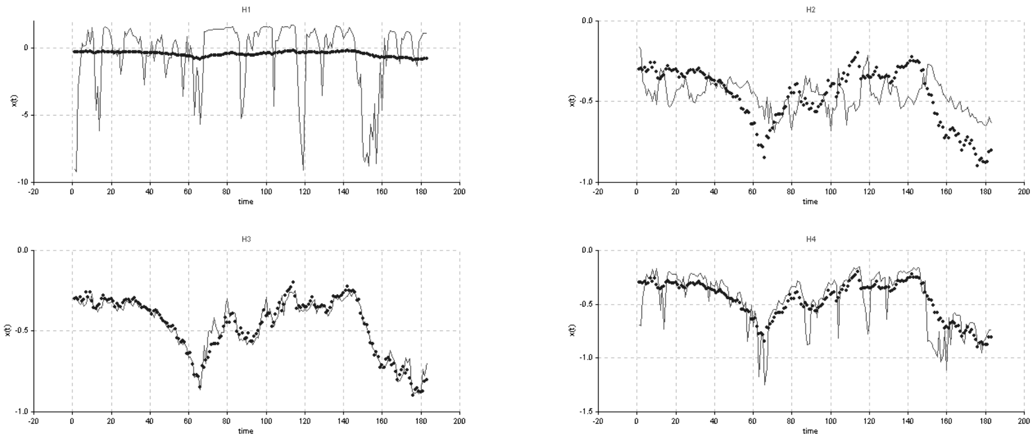

Figure 5: $JPUS$ data (thick dots) and $nn(5,7,1)$ network output without hidden layer cell 1, 2, 3 and 4.

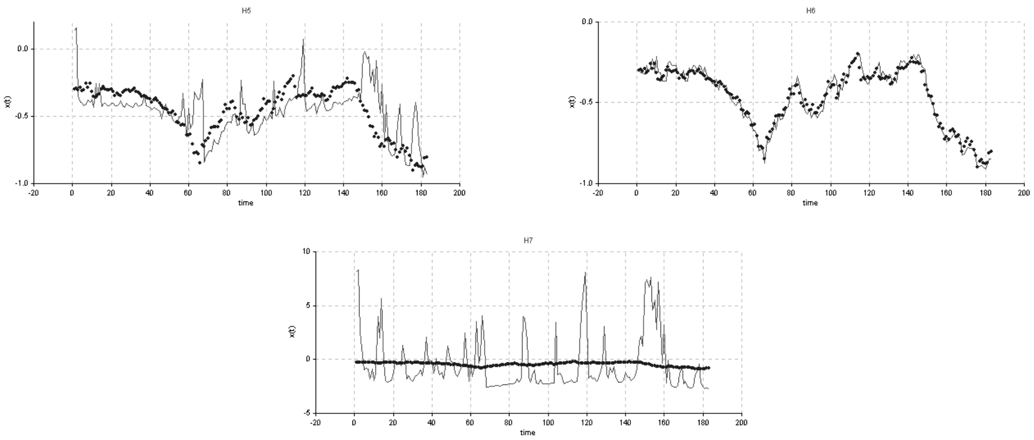

Figure 6: $JPUS$ data (thick dots) and $nn(5,7,1)$ network output without hidden layer cell 5, 6 and 7.

After exclusion of hidden layer cell 3 and 6, another optimization round is applied. In table 7 the results of the network $nn(5,5,1)$ are summarized.

Table 7: Multiple correlation coefficient, incremental contribution and principal components
Network $(5,5,1)$ and Data: $JPUS$

Cell excluded	Multiple correlation	Incremental contribution	SIC
None	0.9745	.	-3.0521
H1	0.0119	0.9626	0.7563
H2	0.1873	0.7872	-0.6145
H3	0.9189	0.0556	-1.1956
H4	0.8718	0.1027	-1.0324
H5	0.0495	0.9250	0.6569
I1 (x_{t-5})	0.0113	0.9632	-1.2504
I2 (x_{t-4})	0.2371	0.7374	-1.0855
I3 (x_{t-3})	0.6999	0.2746	-1.8611
I4 (x_{t-2})	0.7731	0.2014	-1.4533
I5 (x_{t-1})	0.4070	0.5675	-1.2454

Principal Components on hidden layer cells:
Eigenvectors of $E'_{-H}E_{-H}$:

H1	-0.50	0.13	0.44	-0.03	-0.74
H2	-0.05	0.25	-0.12	-0.96	0.05
H3	0.68	0.29	0.67	-0.05	-0.01
H4	0.04	-0.90	0.33	-0.27	0.02
H5	-0.53	0.15	0.49	0.04	0.68

Proportional weights :

	0.04%	0.09%	0.18%	1.03%	98.67%

The results in table 7 are confirmed by examination of the corresponding graphs; in this case, again only the graphs of network output minus one hidden layer cell are shown, see figure 7.
The only candidates for exclusion based on small incremental correlation contributions, are hidden layer cell 3 and 4. A similar conclusion can be conceived from the principal components analysis; see table 7. Whereas the first principal components shows that at least cells 1 and 5 must not be excluded, the second principal component, although with a rather low weight, shows that the hidden cell 2 has still some considerable relevance.

Table 8: Multiple correlation coefficient, incremental contribution and principal components

Network $(5,3,1)$ and Data: $JPUS$

Cell excluded	Multiple correlation	Incremental contribution	SIC
None	0.9705	.	-3.1786
H1	0.1704	0.8001	1.2493
H2	0.1636	0.8069	-0.4931
H3	0.0199	0.9506	-1.1974
I1 (x_{t-5})	0.1946	0.7759	-0.9046
I2 (x_{t-4})	0.8672	0.1033	-2.1275
I3 (x_{t-3})	0.9048	0.0657	-2.5196
I4 (x_{t-2})	0.8609	0.1096	-2.2962
I5 (x_{t-1})	0.8318	0.1387	-1.1524

Principal Components on hidden layer cells:

Eigenvectors of $E'_{-H}E_{-H}$:

H1	-0.70	-0.04	-0.72
H2	-0.00	-1.00	-0.05
H3	-0.72	0.03	0.69

Proportional weights :

	0.07%	1.13%	98.80%

Principal Components on input layer cells:

Eigenvectors of $E'_{-I}E_{-I}$:

I1 (x_{t-5})	-0.07	0.02	0.39	-0.06	0.92
I2 (x_{t-4})	-0.13	-0.87	-0.43	0.07	0.19
I3 (x_{t-3})	-0.70	0.42	-0.54	0.15	0.17
I4 (x_{t-2})	0.70	0.26	-0.57	0.16	0.30
I5 (x_{t-1})	0.00	0.04	-0.23	-0.97	0.03

Proportional weights :

	1.57%	5.69%	6.34%	11.56%	74.84%

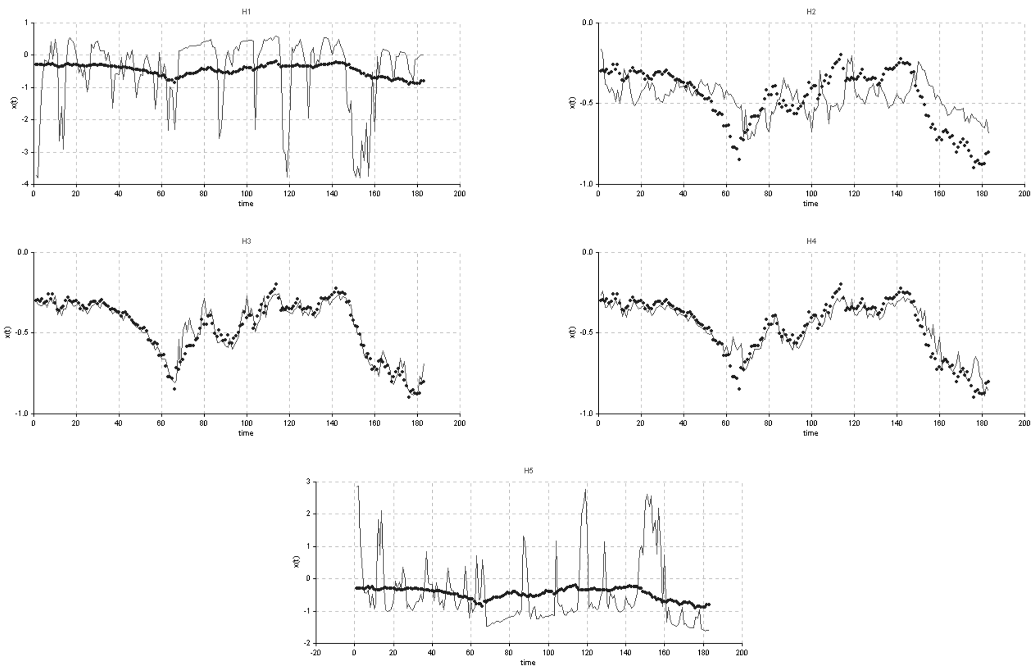

Figure 7: $JPUS$ data (thick dots) and $nn(5,5,1)$ network output without hidden layer cell 1, 2, 3, 4 and 5.

The graphs of figure 7 confirm the conclusion to exclude only hidden cells 3 and 4. Hence the process is continued after exclusion of those two hidden layer cells; the results of optimization are reported in table 8.

With respect to incremental contributions, see table 8, no convincing evidence is given for further exclusion of hidden layer cells. From the principal components analysis, especially the first component, hidden layer cell 2 would be a possible candidate for exclusion. However, the second component is almost equal to e_2, neural network output minus hidden cell 2. So no further reduction in hidden layer cells is attempted and from now on, only input cells are considered.
We start with the graph of network output minus one input layer cell; see figure 8.

Figure 8 is conform table 8 in the sense that the incremental contributions of input cells 2, 3, 4 and 5) are low; so a network $nn(2,3,1)$ with only two input cells, the variables x_{t-1} and x_{t-2} should be a proper guess.

The results after optimization are reported in table 9; it shows a low contribution of the variable x_{t-1}, input cell 2. Figure 9 shows actual data and network estimates. As a performance test of this network, the graphs of network prediction, called network orbit, see page 15, and actual data is shown in figure 10.

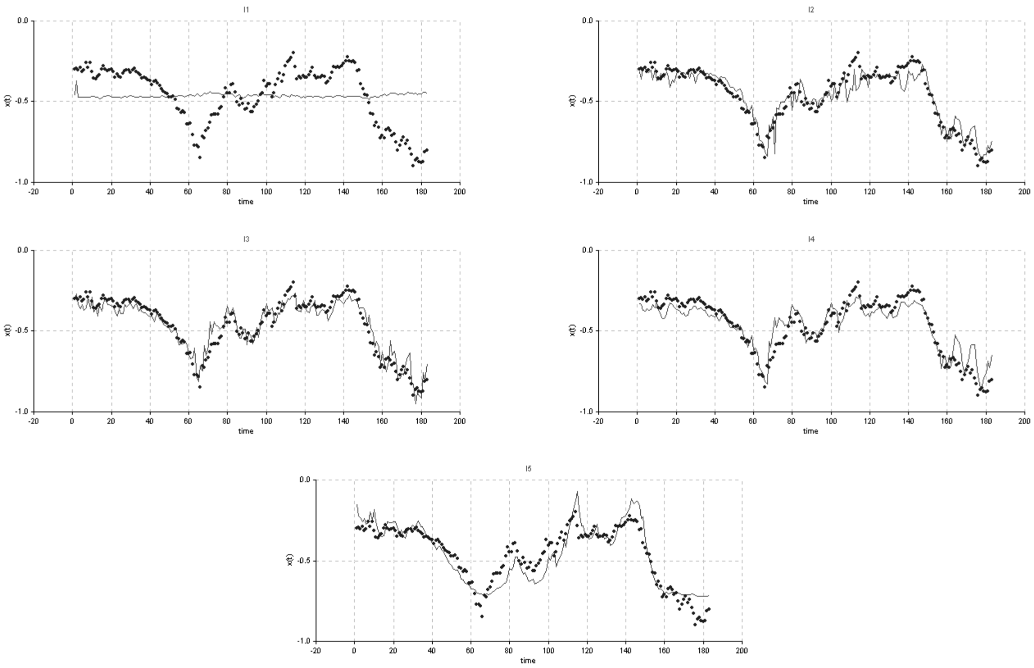

Figure 8: *JPUS* data (thick dots) and $nn(5,3,1)$ network output without input layer cell 1, 2, 3, 4 and 5 respectively.

The performances of networks $nn(2,3,1)$ and $nn(5,3,1)$ differ not much in prediction power; see figure 10. Based on *SIC* value the choice would be the network $nn(2,3,1)$. Although in this case the input variable x_{t-1} has a very low contribution in explaining the variance of x_t, no further reduction is applied.

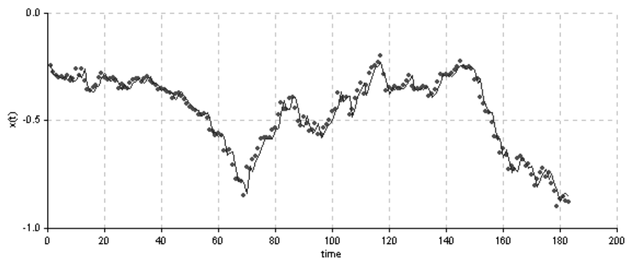

Figure 9: *JPUS* data (thick dots) and network $nn(2,3,1)$ estimates.

Table 9: Multiple correlation coefficient, incremental contribution and principal components

Network $(2, 3, 1)$ and Data: $JPUS$

Cell excluded	Multiple correlation	Incremental contribution	SIC
None	0.9673	.	-3.2547
H1	0.8525	0.1148	2.7181
H2	0.3826	0.5847	-0.8744
H3	0.7030	0.2643	1.2156
I1 (x_{t-2})	0.1156	0.8516	2.4523
I2 (x_{t-1})	0.9636	0.0036	-2.5416

Principal Components on hidden layer cells:

Eigenvectors of $E'_{-H}E_{-H}$:

H1	-0.68	-0.68	-0.28
H2	0.09	-0.45	0.89
H3	-0.73	0.58	0.37

Proportional weights :

4.08%	23.07%	72.84%

Principal Components on input layer cells:

Eigenvectors of $E'_{-I}E_{-I}$:

I1 (x_{t-2})	-0.03	-1.00
I2 (x_{t-1})	-1.00	0.03

Proportional weights :

3.77%	96.23%

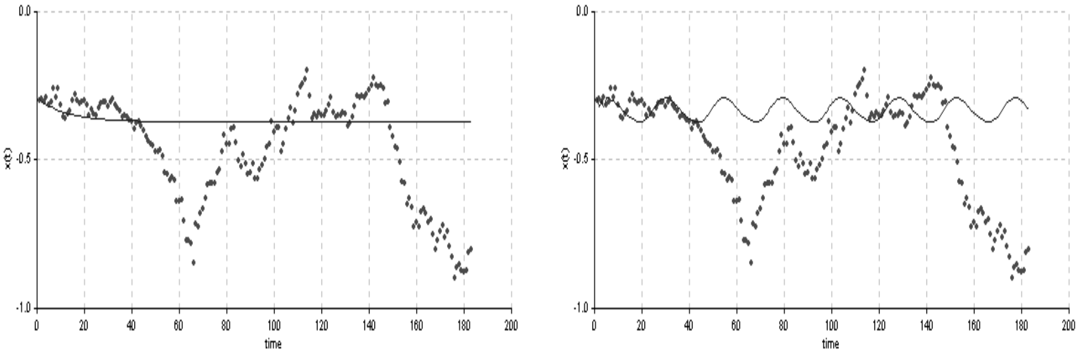

Figure 10: $JPUS$ data (thick dots) and orbit (prediction) generated by network $nn(2,3,1)$ (left) and $nn(5,3,1)$ (right).

6 Summary

In this paper a well known procedure in reducing the number of parameters of a linear model is applied as pruning tool for neural networks. The procedure calculates the incremental contribution of variables, in the case of neural networks: the hidden layer nodes and input layer nodes.
The applied neural network configuration where output is linearly connected to hidden layer cells, seems to be appropriate for applying this procedure on reducing the number of hidden layer cells as is shown in the two examples exposed above.
Another descriptive measure, the principal component analysis of residuals with one node 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.

Inclusion of input variables is based on economic analysis and/or non-linear data analysis; however in most cases one has only an upperbound (and maybe a lowerbound) on the number of variables to be included. So reduction of input layer cells may also be possible in this case and, also in this case with nonlinear connections between input and output layer, incremental contribution of correlation coefficients and graphical analysis provide an easy tool to examining neural network performance and reduction of parameters.

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.

Devaney, R.L., *An Introduction to Chaotic Dynamical Systems*, second ed., Addison-Wesley Publ. Co., Reading, 1989.

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.

Hecht-Nielsen, R., *Neurocomputing*, Addison-Wesley Publ. Co., Menlo Park, CA, 1990.

Hertz, J., A. Krogh & R.G. Palmer, *Introduction to the theory of neural computation*, Addison-Wesley Publishing Company, Reading Massachusetts, 1991.

Kaashoek, J.F. & H.K. van Dijk, Evaluation and Application of numerical procedures to calculate Lyapunov exponents, *Econometric Reviews*, special issue, Vol. 13, No.1, 1994.

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

Schwartz, G., Estimating the Dimension of a Model, *The Annals of Statistics*, **6**, 1978.

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.