Multistep forecasting in electronics based on reduced information

Jelena Milojković and Vančo Litovski

Abstract – New results are reported related to the extension of our prediction methods to the case of multistep forecasting using Artificial Neural Networks (ANNs) based on reduced information. Examples will be given related to prediction of quantities of obsolete computers.

Keywords – Artificial neural networks, prediction, time series, obsolete computers.

I. INTRODUCTION

Prediction of short time series is a topical problem [1]. Cases where the sample length N is too small for generating statistically reliable variants of prediction are encountered every so often. This is a characteristic of many applied problems of prediction in technology development, marketing, politology, investment planning, and other fields. According to statistical analysis, in order to take into account all components, the prediction base period should contain at least several hundreds of units. For periods of several tens of units, satisfactory predictions can be constructed only for the time series representable as the sum of the trend, seasonal, and random components. These models, however, must have a very limited number of parameters. Series made up by the sum of the trend and the random component sometimes may be predicted for even a smaller base period. Finally, as stated in [1] for a prediction base period smaller than some calculated value $N_{\min}$, a more or less satisfactory prediction on the basis of observations is impossible at all, and additional data are required.

All that is valid for the more difficult problem: multi-step ahead prediction. Namely, as the interval in future, for which prediction is made, becomes comparable with the prediction base period, prediction does not make sense no matter how long both series are. Consequently, if the prediction base period is short, the look in future must be limited.

Among the fields not mentioned in [1], dealing with really small set of data or "prediction base period", we will comment here is the environmental impact of electronics which became an important issue nowadays [2]. As a matter of fact, the eco-design of electrical and electronic products is already a legislative matter [3],[4]). Electronic waste (EW) is considered hazardous while, in the same time, in enormous quantities. Prediction in this area is of paramount importance for planning and installing equipments, plants, and facilities for recycling and end-of-life management of electronic products, while short term data are available only.

Jelena Milojković, Vančo Litovski are with the Department of Electronics, Faculty of Electronic Engineering, University of Niš, Aleksandra Medvedeva 14, 18000 Niš, Serbia, E-mail: jelena@venus.elfak.ni.ac.rs, and vanco.litovski@elfak.ni.ac.rs.

In a set of recent studies [5], [6], [7], [8] dealing with the quantities of EW, attempts were made to make prediction based on hunches and rules of thumb. In fact, some presumptions were made and predictions based on them published. Later, having missed the target, the presumptions were corrected, and so on. On the other side, there is a large number of publications dealing with prediction of time series as such, and with prediction related to environmental data based on artificial neural networks (ANNs), to mention only [9].

Having all that in mind we undertook a project of developing an ANN based method that will be convenient for systematic implementation in stationary time series prediction with reduced set of data. Our first results were published in [10], [11], [12]. The main idea implemented was the following. If one wants to create neural network that may be used for forecasting one should enable this property during ANN's training. In addition, the ANN used has to have such a structure to accommodate to the training process for prediction. Following these considerations new forecasting architectures were developed.

The goal of this paper is to present extensions of the methods already published with the goal to implement them in multistep prediction.

The structure of the paper is as follows. After general definitions and statement of the problem we will give a short background related to ANNs application to forecasting. Then we will describe two solutions for possible applications of ANNs aimed to the same forecasting task. Methods of application of these structures and extensions of the ANN structures will be proposed next, aimed to multistep ahead prediction. Finally short discussion of the results and consideration related to future work will be given.

II. PROBLEM FORMULATION AND SOLUTION

A time series is a number of observations that are taken consecutively in time. A time series that can be predicted precisely is called deterministic. A time series that has future elements which can be partly determined using previous values, while the exact values cannot be predicted, is said to be stochastic [13]. The stochastic models provide the forecast as the expectation of the identified stochastic process. They allow calculations on statistical properties of the forecasting error (which of course rely on the assumptions made on the model). The deterministic models, on the other hand, provide only the forecast values, not a measure for the forecasting error [14].

We are here addressing deterministic type of time series, only. It is our task to find a functional expression that captures the complex interwoven deterministic relationships that exists between the phenomenon under consideration and the independent variables.

Consider a scalar time series denoted by y^i, i=1,2, …, m-1. It represents a set of observables of an unknown function Y=$F(t)$, taken at equidistant time instants separated by the interval Δt i.e. $t^{i+1}= t^i+\Delta t$. One step ahead forecasting means to find such a function f that will perform the mapping

$$y^m = f(t^m) = Y^m + \varepsilon \qquad (1)$$

where Y^m is the desired response, with an acceptable error ε.

The prediction of a time series is synonymous with modelling of the underlying physical process responsible for its generation [15]. This is the reason of the difficulty of the task. There have been many attempts to find solution to the problem. Among the classical deterministic methods we may mention the k-nearest-neighbour [16], in which the data series is searched for situations similar to the current one each time a forecast needs to be made. This method asks for kind of periodicity to function that is not the case in the situation considered in our proceedings.

In the past decades ANNs have emerged as a technology with a great promise for identifying and modelling data patterns that are not easily discernible by traditional methods. A comprehensive review of ANN use in forecasting may be found in [17]. Among the many successful implementations we may mention [18] [19] [20]. A common feature, however, of the existing application is that they ask for a relatively long time series to become effective. Typically it should be not shorter than 50 data points [17]. This is due to the fact that they all look for periodicity within the data what can be easily seen from the typical forecasting competition data [19]. Very short time series were treated [20], [21]. Here additional "nonsample information" was added to the time series in order to get statistical estimation from deterministic data.

That is why we went for a search for topological structures of ANN that promise prediction based on short time series. In the next, we will first briefly introduce the feed-forward neural networks that will be used as a basic structure for prediction throughout this paper.

The network is depicted in (Fig.1). It has only one hidden layer, which has been proven sufficient for this kind of problem [22]. Indices: "in", "h", and "o", in this figure, stand for input, hidden, and output, respectively. For the set of weights, $w(k, l)$, connecting the input and the hidden layer we have: k=1,2,..., m_{in}, l=1,2,..., m_{h}, while for the set connecting the hidden and output layer we have: k=1,2, ...m_{h}, l=1,2,..., m_{o}. The thresholds are here denoted as $\theta_{x,r}$, r=1,2, …, m_{h} or m_{o}, with x standing for "h" or "o", depending on the layer. The neurons in the input layer are simply distributing the signals, while those in the hidden layer are activated by a sigmoidal (logistic) function. Finally, the neurons in the output layer are activated by a linear function.

Creation of a feed forward ANN that performs a given task consists of several steps. First, one should decide on the number of inputs and outputs, m_{in} and m_{o}, respectively. That usually comes with the nature of the problem under consideration.

Next, according to the input-output structure the training data are to be organized. Pairs of input-output vectors are taken from the known data and a list is created intended to be presented to the ANN during training. Generally speaking, part of the input data is kept for validation of the training process but when prediction is considered there is no such data. We simply look to the unknown future. Verification of the prediction may be done only after time passed. Here the importance of the dependability onto the whole prediction algorithm and software comes into fore. The algorithm should be organized in that way to perform automatically and give no chance to mistakes.

The internal structure of the network i.e. the number of hidden neurons (m_{h}) is of paramount importance for successful prediction. It defines the number of free parameters that are available for optimization (training). Of course, one would prefer as simple the ANN as possible. That not only makes the solution faster to run but also facilitates the training process in: choice of the initial values of the parameters, reaching convergence, and speeding up the training. To get the value of m_{h} we applied a procedure that is based on proceedings given in [23].

We solve the initial value problem for the weights and thresholds by creating small random numbers with uniform distribution such that $\nu_k \in (-\alpha, \alpha)$, where ν_k stands for the kth parameter (weight or threshold), while α is a properly chosen small number.

When considering the training algorithm one must have in mind that it represents a search of the parameters space for the global minimum of the predefined error function. Any learning algorithm, being it in nature Newthon-Raphson, steepest descent, back propagation, annealing based (Metropolis), or genetic, which, for a given procedure creating initial solution, leads repetitively and systematically to a global minimum, is good enough for application. Of course, some of the algorithms will lead to the solution with different velocity (expressed in number of iterations and elapsed time) but that is of secondary importance. Among the successful algorithms one chooses the one that is fastest, simplest to program, needs less computer memory, etc. The learning algorithm we used for training in these proceedings is a version of the steepest-descent minimization algorithm [24]. It is our experience (almost twenty years, now) that for the problem under consideration it performs the best.

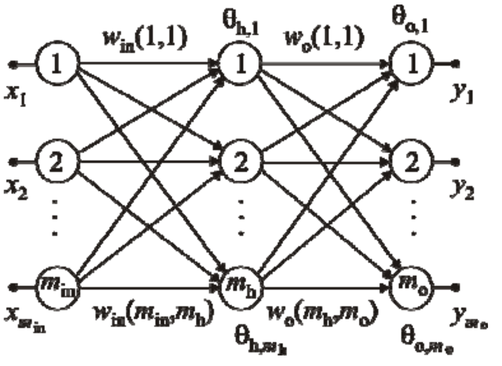

Figure 1. Fully connected feed-forward neural network with one hidden layer and multiple outputs

In prediction of time series, in our case, a set of observables (samples) is given (per year) meaning that only one input signal is available, the discretized time. We are predicting one quantity at a time meaning one output is needed, too. The values of the output are numbers (millions of pieces or weight of obsolete computer units). To make the forecasting problem numerically feasible we performed transformation in both the time variable and the response. The time was reduced by t_0-1 so that

$$t=t^*-(t_0-1). \quad (2)$$

Having in mind that t^* stands for the year, this reduction gives the value of 1 to the year (t_0) related to the first sample. The samples are normalized in the following way

$$y=y^* / M \quad (3)$$

where y^* stands for the current value of the target function, M is a constant which will be chosen according to the problem at hand (for example, M=10^6 cubic feet for the volume of obsolete computers).

If the architecture depicted in (Fig. 1) was to be implemented the following series would be learned: (t^i, $f(t^i)$), i=1,...,m-1. m-1 is here the number of samples available i.e. the number of observables.

Starting with the basic architecture of (Fig. 1.), the possible solutions were investigated in [10] [11] and two new architectures were suggested to be the most convenient for the solution of the forecasting problem based on short prediction base period.

The first one, named *time controlled recurrent* (TCR) was inspired by the time delayed recurrent ANN [15]. It is a recurrent and time delayed architecture but, in the same time, insists on the time variable to control the predicted value as depicted in (Figure 2). Our intention was to benefit from both: the generalization property of the ANNs and the success of the recurrent architecture. Here in fact, the network is learning a set in which the output value is controlled by its own previous instances and the present time. The version of this network, intended to be implemented for one-step-ahead prediction may be analytically expressed as:

$$y^{i+1} = f(t^i, y^i, y^{i-1}, ..., y^{i-q}),\ i=q+1, ..., m-2, \quad (5a)$$

where q stands for the number of previous values of the function used for training. q>0 and, obviously, $q+1<m-1$. After training, the predicted value in the first next step is obtained as

$$y^m = f(t^{m-1}, y^m, y^{m-1}, ..., y^{m-q}). \quad (5b)$$

Note that the learning procedure here was implemented exactly in the same way as in [25].

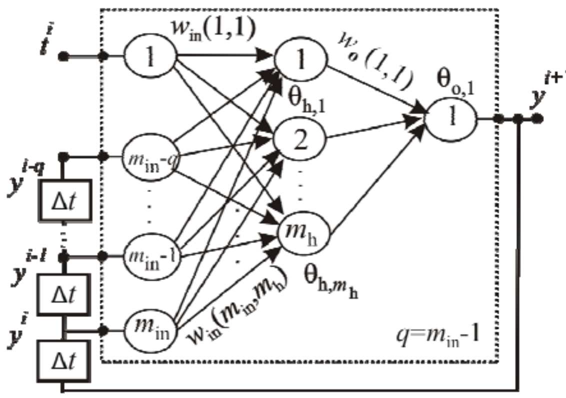

Figure 2. TCR. Time controlled recurrent ANN

The second architecture is named *feed forward accommodated for prediction* (FFAP) and depicted in (Fig. 3). Our idea was here to force the neural network to simultaneously learn the same mapping several times but shifted in time. In that way, we presume, the previous responses of the function will have larger influence on the $f(t)$ mapping.

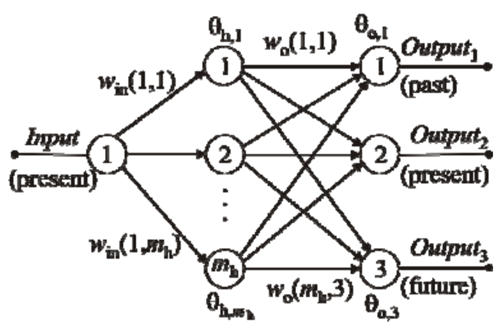

Figure 3. FFAP. Feed forward ANN structure accommodated for prediction

There is one input terminal that, in our case, is t^i. The *Output*$_3$ terminal, or the future terminal, in our case, is to be forced to approximate y^{i+1}. *Output*$_2$ should learn the present value i.e. y^i. Finally, *Output*$_1$ should learn the past value i.e. y^{i-1}. Again, if one wants to control the mapping by a set of previous values, *Output*$_1$ may be seen as a vector such as $\{ y^{i-1}, y^{i-2}, \ldots, y^{i-q}\}$. We may express the functionality of the network, for the case of one-step-ahead prediction, as

$$\{y^{i+1}, y^i, y^{i-1}, \ldots, y^{i-q}\} = \mathbf{f}(t^i)\ , \ i=q+1, \ldots, m-2, \quad (6a)$$

where *Output*$_1$=$\{y^{i-1}, \ldots, y^{i-q}\}$, meaning that: one future, one present and q previous responses are to be learned. After training the predicted and the approximated values of the output are obtained by running the ANN as:

$$\{y^{m+1}, y^m, y^{m-1}, \ldots, y^{m-q}\} = \mathbf{f}(t^m)\ . \quad (6b)$$

The presumption of the mutual interrelation between the output responses of the FFAP network comes from the fact that they all depend on the parameters (weights and thresholds) of the hidden neurons. By adjusting the parameters to learn y^i, for example, one simultaneously changes the y^{i+1} response, and vice versa. In that way, during training, the values of the response from previous time instants indirectly control the prediction.

III. Implementation example

An examples will be given here demonstrating the properties of the solutions proposed with q=2.

We will consider the prediction of the quantities of obsolete computers in the USA based on data given in [5]. According to [5], putting t_0= 1991, after normalization, we get (Table 1) as the set of observables representing the quantities of obsolete computers in the USA. Here M=10^6 cubic feet. The same data are visualized in (Fig. 4). It may be seen that the function that governs the phenomenon is not monotonic giving rise to the difficulty of prediction. If, for example, periodicity is to be exploited in this example (what would be done if the *k*-nearest neighbor method was implemented) then $f(9)$ would be less than 14, since after three points of positive increments (as for the interval {1,3}) comes a negative one. $f(4)<f(3)$ would lead to $f(9)<f(8)$ which is not the case.

The first eight samples will be used as training data while the last one i.e. t=9 and $f(t)$=18.4, will be compared with the predictions obtained, in order to validate the method.

In the following, two experiments will be described based ANN architectures emanated from (Figure 2) and (Figure 3).

The results obtained after learning are expressed in (Table 2). It contains information on both the structure of the networks and the values obtained by prediction.

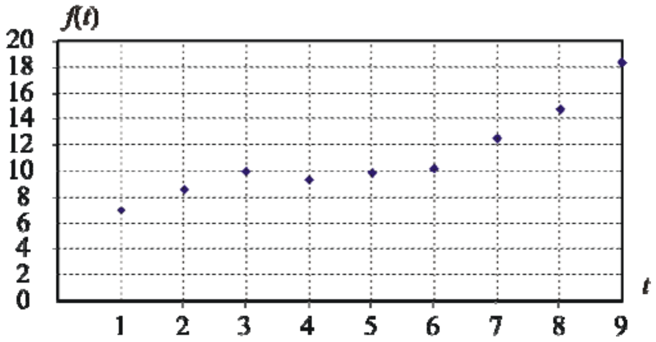

Figure 4. Visualization of Table 1

TABLE 1.
QUANTITIES OF OBSOLETE COMPUTERS IN TIME

t	1	2	3	4	5	6	7	8	9
$f(t)$	7.03	8.67	10.0	9.33	9.85	10.18	12.54	14.76	18.4

TABLE 2.
PREDICTION OF QUANTITIES OF OBSOLETE COMPUTERS. NOTE: $F(9)$=18.4.

Solution type	No. of hidden neurons	No. of output neurons	$f(9)$	Error %
TCR	10	1	17.2114	6.46
FFAP	4	4	18.2274	0.93

By examining the results depicted in (Table 2) we may conclude that satisfactory prediction was obtained with both architectures. Nevertheless, it is to be mentioned that the FFAP is considerably nearer to the solution needed. What is not expressed in the table is the fact that the FFAP solution is much more sensitive to the initial solution for the weights and thresholds, making the training process more difficult and uncertain.

It is not shown here, for the sake of simplicity, but it is worth mentioning that both TCR and FFAP approximate excellent. That means that except for the [9, $f(9)$] point, all previous points on the curve $f(t)$ overlap exactly with the ones depicted in (Figure 4). One should not substitute "approximation" with "prediction", however. Namely, approximation is achieved *within* a given interval. Here, for this example, that is $t \in \{1,8\}$. The ability of the ANN to successfully calculate the values of the function for any value of the independent variable within that interval is referred to as *generalization*. In our case we are looking for the *extrapolation* i.e. the value of the function outside of the given interval. That is what we consider forecasting or prediction.

There is no recommendation as to which of these solutions is to be accepted or discarded. Namely, prediction is a search in the dark and one always needs some reference for the solution offered. Here, since the FFAP solution offers better results, one should keep the TCR solution as a confirmation that the FFAP is not a complete miss what is, of course, possible since the training of an ANN is iterative process that may be stuck in local minimum.

IV. IMPLEMENTATION TO MULTI-STEP PREDICTION

The main goal of our research was to develop a method for one-step-ahead prediction based on reduced set of data. Implementation to long term prediction was always a temptation while we are aware that it is difficult to believe that one may predict for a period in future as long as the prediction base period is. Instead, here we will give the results of an attempt to apply our method to prediction for a somewhat longer period than one-step-ahead.

There are, in our opinion, two ways of how our method may be applied for longer term prediction. First, one may use the predicted results for the time instant t^{i+1}, namely y^{i+1}, and to concatenate the input set with them. Now, the prediction may start for t^{i+2} as if one has longer prediction base period. This may be repeated as long as wanted. The problem with this idea is related to the fact that the error in prediction contained in y^{i+1} will be accumulated in the next prediction, and so on. At the end, one may have no confidence in the final long term prediction. Example of implementation of this idea to the problem of forecasting quantities of obsolete computers is given in Table 3. Both TCR and FFAP ANNs were implemented to get prediction for t=9, based on samples for t=1,2,3,…,7. The idea is to predict two intervals ahead. The value of y for t=8 was predicted first. Then, it was used as if it was part of the input file to predict $y(9)$. We can see that the results are worse than the previous ones with the ones obtained with FFAP ANN being absolutely deteriorated.

TABLE 3.
TWO-STEP-AHEAD PREDICTION BY CONCATENATION

	Actual	Predicted	Error in %
TCR	18.4	16.8616	8.36
FFAP	18.4	26.2071	-66.3

Alternatively, one may predict two (or more) steps ahead directly by skipping the intermediate intervals. In such a case, for the TCR ANN, for instance, one would perform the following

(7a) $$y^{i+k} = f(t^i, y^i, y^{i-1}, ..., y^{i-q}),\ i=q+1, ..., m-1-k,$$

while for the FFAP case we have

(7b) $$\{y^{i+k}, y^i, y^{i-1}, ..., y^{i-q}\} = \mathbf{f}(t^i),\quad i=q+1, ..., m-1-k.$$

In this expressions k stands for the number of intervals in future after the prediction base period.

Looking to them we find that for the one-step-ahead prediction (k=1) we had m-1 samples to be used for training and $b=m-2-q$ "training lessons". On the other side, for multistep prediction, the number of training lessons may be stated as $b=m-1-k-q$, as depicted in Fig. 5 for q=2 and k=2. If the number of intervals in future, k, rises, b is diminished. It is equivalent to reduction of the prediction base period what should lead to reduction of the quality of the forecast.

	q=2		b=4				k=2		
t	1	2	3	4	5	6	7	8	9
$f(t)$	7.03	8.67	10.0	9.33	9.85	10.18	12.54	14.76	18.4

Figure 5. The reduction of the number of training lessons in multistep ahead prediction

This method of prediction was checked by an experiment related to prediction of the number of obsolete computers as above. Again both TCR and FFAP ANNs were implemented. The forecasting results are given in Table 4. Comparing these result with the ones depicted in Table 2, we may conclude that the expected deterioration does not becomes apparent at once (the number of training lessons was reduced by 1 only). In the TCR case we got even an improvement. That makes this approach promising in general and especially in cases when a bit larger prediction base period is available.

TABLE 4.
TWO-STEP-AHEAD PREDICTION WITH SKIPPING

	Actual	Predicted	Error in %
TCR	18.4	18.7458	-1.879
FFAP	18.4	17.5698	4.512

Finally, for the FFAP ANN only, in cases where multiple-step prediction is planned $Output_3$ may be seen as a vector. In this situation, referring to Fig. 3, $Output_1$ has q terminals, $Output_2$ has 1 terminal, and $Output_3$ has k terminals. Analytically, this may be expressed as

(8) $$\{y^{i+k}, y^{i+k-1}, ..., y^i, y^{i-1}, ..., y^{i-q}\} = \mathbf{f}(t^i),\ i=q+1, ..., m-1-k.$$

The appropriate structure of the FFAP ANN is depicted in Fig. 6 for q=2 and k=2. It is important to notice that no skipping is present now. The network is presented by all the future values of the function.

Applying this method to the case of obsolete computers with two-steps-ahead prediction, with q=2 and k=2, produced a solution of $f(9)$=18.5106 what is a miss of only **0,6**%. It is an excellent result comparing with the results presented here earlier. The number of training lessons was now reduced to t∈ {3,6}.

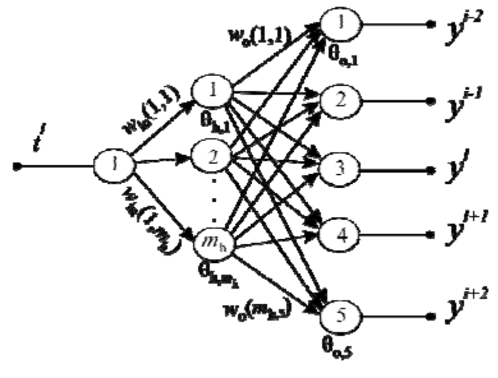

Figure 6. FFAP structure for two step ahead prediction without skipping

TABLE 5.
DATA STRUCTURE FOR THE FFAP NETWORK PREDICTING TWO-STEPS-AHEAD WITHOUT SKIPPING.

	i	y^{i-2}	y^{i-1}	y^{i}	y^{i+1}	y^{i+2}
Training	3	7.03	8.67	10.0	9.33	9.85
Training	4	8.67	10.0	9.33	9.85	10.18
Training	5	10.0	9.33	9.85	10.18	12.54
Training	6	9.33	9.85	10.18	12.54	14.76
Prediction	7	6.447	11.78	10.59	18.40	18.51
Expected valueas		9.85	10.18	12.54	14.76	18.4

The data structure for training and running the FFAP network predicting two-steps-ahead without skipping is given in Table 5. Here, for convenience, in the row i=7, the responses of all outputs of the network are presented while only the last one is usable for prediction. The first four outputs are trained to approximate. To go further, in the last row, the expected values for every output are listed. Comparing the last two rows of Table 5 one easily concludes that no output except the one intended to, is predicting successfully. That is in accordance with our previous results discussed in [11]: "No ANN trained for interpolation can predict (extrapolate) successfully".

Finally, the response of the predicting output (y^5) as a function of time together with the target values is depicted in Fig. 7. One may see that this response not only extrapolates but interpolates excellent as well.

To make the results completely reproducible, Table 6 contains the initial values of the synaptic weigths and thresholds used for the training process of the FFAP network predicting two-steps-ahead without skipping. Table 7 contains the final values obtained after training and used to get the prediction.

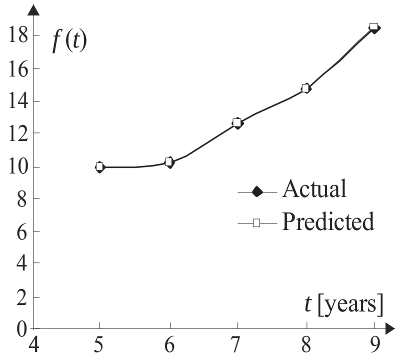

Figure 7. Response of the FFAP structure for two-steps- ahead prediction without skipping

TABLE 6.
INITIAL VALUES OF THE SYNAPTIC WEIGHTS AND THRESHOLDS FOR THE FFAP NETWORK PREDICTING TWO- STEPS-AHEAD WITHOUT SKIPPING

j	$w_{in}(1, j)$	$\theta_{h,j}$	$w_o(j,1)$	$w_o(j,2)$	$w_o(j,3)$	$w_o(j,4)$	$w_o(j,5)$	$\theta_{o,j}$
1	-.122	-.102	-.132	-.132	-.32	-.32	-.132	.135
2	.131	.211	.1243	.131	.124	.124	.240	-.21
3	-.212	-.131	-.214	-.14	-.140	-.140	-.124	.123
4	.120	.121	.124	.240	.24	.324	.324	-.141
5								.121

TABLE 7.
VALUES OF THE SYNAPTIC WEIGHTS AND THRESHOLDS FOR THE FFAP NETWORK PREDICTING TWO- STEPS-AHEAD WITHOUT SKIPPING

j	$w_{in}(1, j)$	$\theta_{h,j}$	$w_o(j,1)$	$w_o(j,2)$	$w_o(j,3)$	$w_o(j,4)$	$w_o(j,5)$	$\theta_{o,j}$
1	-4.45698	-0.584837	-5.72051	6.86385	-5.77336	1.01748	-4.08986	5.81894
2	6.19043	-6.12091	-3.14972	4.6372	1.23713	5.53971	2.14366	-17.4828
3	-5.39005	-6.21706	-2.41941	-3.80742	-4.04059	-7.0992	-6.93629	15.4904
4	42.7411	1.85431	-4.34902	17.3764	-14.6085	2.31862	-5.84657	-2.36684
5								6.52768

V CONCLUSION

New results were reported related to multistep ahead prediction using ANNs based on reduced information. Several solutions were proposed and experimental results were given for one implementation. In general, encouraging results were obtained for two-step prediction except for the method using concatenation. These results will be a basis for further research and implementation to different contexts such the ones presented in [10], [12], [26] and [27].

REFERENCES

[1] Mandel', A.S., "*Method of Analogs in Prediction of Short Time Series: An Expert-statistical Approach*", Automation and Remote Control, Vol. 65, No. 4, 2004, pp. 634-641.

[2] Karn, B., and Matthews, H.S., "*Nanotechnology: Emerging Challenges for Electronics and the Environment*", IEEE Spectrum, Vol. 44, No. 9, 2007, pp. 54-58.

[3] -, *Directive 2005/32/ec of the European parliament and of the council of 6 July 2005*, EN Official Journal of the European Union L 191/29.

[4] Bandyopadhyay, A. , "*A regulatory approach for e-waste management: a cross-national review of current*

practice and policy with an assessment and policy recommendation for the Indian perspective", International Journal of Environment and Waste Management, Vol. 2, No. 1-2, 2008, pp. 139–186.

[5] Matthews, H.S., McMichael, F.C., Hendrickson, C.T., and Hart, D., "*Disposition and End-of-Life Options for Personal Computers*", Green Design Initiative Technical Report #97-10, Carnegie Mellon University, 1997.

[6] Feszty, K., Colin, M., and Baird, J., "*Assessment of the quantities of waste electrical and electronic equipment (WEEE) in Scotland*", Waste management & research, Vol.21, No. 3, 2003, pp. 207-217.

[7] Matthews H.S., and Matthews, D., "*Computers and the Environment: Understanding and Managing Their Impacts*", in Computers and the Environment, eds. E. Williams and R. Kuher, Dordrecht, Kluwer Academic Publications, 2004.

[8] Litovski, V., Milojković, J., Petrović, S., Džipković, D., Šimurina, M., and Krstić, B., "*Program of establishment of the recycling system of waste electronic of computers*", Belgrade, Serbian Agency for Recycling 2006, (in Serbian).

[9] Hawkins, T., Hendrickson, C., Higgins, C., Matthews, H. S., and Suh S., "*A mixed-unit input-output model for environmental life-cycle assessment and material flow analysis*", Environmental Science & Technology, Vol. 41, No. 3, 2007, pp. 1024-1031.

[10] Milojković, J., and Litovski, V. B., "*New methods of prediction implemented for sustainable development*", in 51th Conference of ETRAN, EL1.8, 2007, (in Serbian).

[11] Milojković, J., and Litovski, V. B.,"*Comparison of Some ANN Based Forecasting Methods Implemented on Short Time Series*", in 9th Symposium on Neural Network Applications in Electrical Engineering, NEUREL-2008, pp. 175-178.

[12] Milojković, J, and Litovski, V.B., „*Dynamic Short-Term Forecasting of Electricity Load Using Feed-Forward ANNs*“, Engineering Intelligent Systems for Electrical Engineering and Communication, ISSN 1472-8915, accepted for publication

[13] Hussain, A., (2002), "*Physical time-series prediction using second order pipelined recurrent neural network*", in Proc. of the 2002 IEEE Int. Conf. on Artificial Intelligence Systems, 219-223.

[14] Murto P., "*Neural Network Models for Short-Term Load Forecasting*". MS Thesis, Helsinki University of Technology, Finland, 1998.

[15] Haykin, S., "*Neural Networks, A Comprehensive Foundation*", New York, Macmillan College Publishing Company, 1994.

[16] Plummer, E.A., "*Time series forecasting with feed-forward neural networks: guidelines and limitations*", unpublished M.S. Thesis, University of Wyoming, The Graduate School, 2000.

[17] Zhang, B. G., Patuwo, E., and Hu, M. Y, "*Forecasting with artificial neural networks: The state of the art*", International Journal of Forecasting, Vol. 14, No. 1, 1998, pp. 35-62.

[18] Connor, J., and Douglas Martin, R., "*Recurrent neural networks and robust time series prediction*", *IEEE Trans. on Neural Networks*, Vol. 5, No. 2, 1994, pp. 240-254.

[19] -, *NN3 Competition*, http://www.neural-forecasting-competition.com/

[20] Brännäs, K., and Hellström, J., "*Forecasting based on Very Small Samples and Additional Non-Sample Information*", Umeå Economic Studies 472, Umeå University, Sweden, 1998.

[21] Navone, H. D., and Ceccatto, H. A., "*Forecasting chaos from small data sets: a comparison of different nonlinear algorithms*", Journal of Physics A: Mathematical and Theoretical, Vol. 28, No. 12, 1995, pp. 3381-3388.

[22] Masters, T., "*Practical Neural Network Recipes in C++*", San Diego, Academic Press, 1993.

[23] Baum, E.B., and Haussler, D., "*What size net gives valid generalization*", Neural Computing (MIT Press), Vol. 1, No.1, 1989, pp. 151-160.

[24] Zografski, Z., "*A novel machine learning algorithm and its use in modeling and simulation of dynamical systems*", in Proc. of 5th Annual European Computer Conference, COMPEURO '91, 1991, pp. 860-864.

[25] Bernieri, A., D'Apuzzo, M., Sansone, L., and Savastano, M, "*A neural network approach for identification and fault diagnosis on dynamic systems*", IEEE Transactions on Instrumentation and Measurements, Vol. 43, No. 6, 1994, pp. 867-873.

[26] Milojković, J.B., and Litovski, V.B., "*Short-term forecasting of electricity load using recurrent ANNs*", 15th Int. Symposium On Power Electronics – Ee2009, Novi Sad, Serbia, October 2009, Paper No. T1-1.7, pp. 1-5.

[27] Milojković, J. And Litovski, V.B., „*Short-term forecasting of the electricity load at suburban level*“, Ninth National Conference with International Participation, ETAI 2009, Ohrid, Macedonia, September 2009, Proc. on disc, Paper No. A2-1.