Paper

The Development of Kalman Filter Learning Technique for Artificial Neural Networks

Agnieszka Krok

Tadeusz Kościuszko Cracow University of Technology, Cracow, Poland

Abstract—The paper presents an idea of using the Kalman Filtering (KF) for learning the Artificial Neural Networks (ANN). It is shown that KF can be fully competitive or more beneficial method with comparison standard Artificial Neural Networks learning techniques. The development of the method is presented respecting selective learning of chosen part of ANN. Another issue presented in this paper is the author's concept of automatic selection of architecture of ANN learned by means of KF based on removing unnecessary connection inside the network. The effectiveness of presented ideas is illustrated on the examples of time series modeling and prediction. Considered data came from the experiments and situ measurements in the field of structural mechanics and materials.

Keywords—Artificial Neural Networks, Kalman Filter.

1. Introduction

The growing popularity and increasing use of the Artificial Neural Networks in virtually all fields of engineering and technical sciences lead to their fast development. From among the learning algorithms gradient descent methods, conjugate gradient methods, the Levenberg-Marquardt algorithm (LM), and the resilient back propagation algorithm (Rprop) were developed predominantly. Their advantages and disadvantages are thoroughly understood [1]. In the paper the alternative learning algorithm was exploited. The nonlinear Kalman Filter can be considered as a simple dynamic Bayesian networks [2]. It calculates a minimum mean-square error estimator for the underlying process and it is adopted into estimating weights and biases of ANN. After development the KF learning method enabled to control the learning process strongly that resulted in very efficient simulations and predictions made by ANN and may be alternative for traditional ANN learning methods [2]–[5].

2. Multilayered Feed Forward Artificial Neural Networks Learned by Kalman Filtering

2.1. Artificial Neural Networks

Standard Multilayered Feed Forward ANN with the same activation function in the each node was considered. Assuming that N, M, K, is the number of inputs, outputs and hidden neurons respectively the answer of the network for the input vector

$$\mathbf{x} = [x_1, \ldots, x_N] \tag{1}$$

for the m-th ($m = 1, \ldots, M$) output is

$$y_m = F\left(\sum_{i=0}^{K} x_i^u w_{m,i}^2\right), \tag{2}$$

where

$$x_i^u = F\left(\sum_{j=0}^{N} x_j w_{i,j}^1\right). \tag{3}$$

Finally

$$y_m = F\left(\sum_{i=0}^{K} F(\sum_{j=0}^{N} x_j w_{i,j}^1) w_{m,i}^2\right), \tag{4}$$

where: F is the activation functions for first and second layer and $w_{i,m}^2$ is the connection between i-th ($i = 1, 2, \ldots, K$) hidden neuron with m-th output neuron ($m = 1, 2, \ldots, M$). Similarly for $w_{i,j}^1$, where $j = 1, 2, \ldots, N$. Vector $[x_1^u, \ldots, x_K^u]$ is created by signals coming from K hidden neurons. Lets assume that $w_{m,0}^1$ and $w_{i,0}^2$ are ANN biases. For real ϕ, the activation functions in the form

$$F(\phi) = \frac{1}{1 + e^{-a\phi}} \tag{5}$$

or

$$F(\phi) = tanh(\phi/2) = \frac{1 - e^{-a\phi}}{1 + e^{-a\phi}} \tag{6}$$

were considered.

The ANN described above is learned by means of KF using the teacher - standard procedure was used: the pth input vector $\mathbf{x}^p$ should generate the desirable target $\mathbf{t}^p$ were p from 1 to P are known learning patterns. The magnitude between target and the ANN output Eq. (4) gives the level of ANN changes that are necessary to obtain its goal. The performance of the network and its generalization abilities were tested on separate testing set of patterns like in every standard method of ANN learning and testing [2]. In the paper the change of ANN weight vector is calculated using estimation inside KF model.

2.2. Algorithm Node Decoupled Extended Kalman Filter

The Kalman Filter is an algorithm that uses a series of measurements observed over time, containing noise and produces estimates of unknown variables. It operates recursively on streams of noisy input data to produce a statistically optimal estimate of the underlying system state. The basic Kalman Filter is linear [6]. In the Extended Kalman Filter (EKF), the state transition and observation models may be differentiable functions. This procedure can be adopted into the process of ANN learning, assuming that the main function in the model represent ANN itself [2]. Among a number possibilities for decoupling the single ANN into several Kalman filtering models, it was proposed to use new KF model for each neuron. It resulted in smaller dimensions of matrix being proceed and additional advantages in the field of controlling learning process.

Let I be the number of all neurons in ANN ($I = M + K$) and let $\mathbf{w}^i$ be the connection weight associated with the ith neuron $i = 1, 2, \ldots, I$. For each neuron the Kalman Filter model is built including process Eq. (7) and measurement Eq. (8):

$$\mathbf{w}_{k+1}^i = \mathbf{w}_k^i + \omega_k^i \,, \tag{7}$$

$$\mathbf{t}_k = \mathbf{h}(\mathbf{w}_k, \mathbf{x}_k) + \nu_k \,, \tag{8}$$

where: k is discrete pseudo-time parameter (for ANN marking the number of learning pattern), $\mathbf{w}$ is the state vector of KF model (here corresponding to the set of synaptic weights and biases), $\mathbf{h}$ is non-linear function that is representing ANN and takes the Formula (4), $\mathbf{x}/\mathbf{t}$ is input/output vectors that corresponds to input of ANN and known target, ω_k^i, ν_k are Gaussian process and measurement noises with mean and covariance matrices defined by:

$$\mathbf{E}(\nu_k) = \mathbf{E}(\omega_k^i) = 0 \,, \tag{9}$$

$$\mathbf{E}\big((\omega_k^i)(\omega_l^i)^T\big) = (\mathbf{Q}_k^i)\delta_{kl} \,, \tag{10}$$

$$\mathbf{E}(\nu_k \nu_l^T) = \mathbf{R}_k \delta_{kl} \,, \tag{11}$$

where: E is the expected value of a random variable, δ_{kl} is equal to 1 if $l = k$ or 0 if not.

The change of ANN weights in i-th node ($\mathbf{w}^i$) during the presentation of k-th learning pattern takes than the following form, see Fig. 1:

$$\mathbf{K}_k^i = \mathbf{P}_k \mathbf{H}_k^i \left[\sum_{j=1}^{g} (\mathbf{H}_k^j)^T \mathbf{P}_k^j \mathbf{H}_k^j + \mathbf{R}_k \right]^{-1} , \tag{12}$$

$$\mathbf{w}_{k+1}^i = \mathbf{w}_k^i + \mathbf{K}_k^i \xi_k \,, \tag{13}$$

$$\mathbf{P}_{k+1}^i = \left(\mathbf{I} - \mathbf{K}_k^i (\mathbf{H}_k^i)^T \right) \mathbf{P}_k^i + \mathbf{Q}_k^i \,, \tag{14}$$

where: $\mathbf{K}_k^i$ is Kalman gain matrix, $\mathbf{P}_k^i$ is approximate error covariance matrix, $\xi_k = \mathbf{t}_k - \mathbf{y}_k$ is error vector, with the target vector $\mathbf{t}_k$ for the k-th presentation of a training pattern, $\mathbf{y}_k$ is output vector given by Eq. (4).

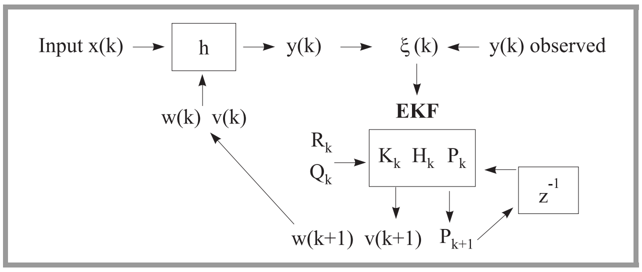

Fig. 1. Kalman Filtering learning algorithm.

$\mathbf{H}$ is the matrix of current linearization of Eq. (8):

$$\mathbf{H}_k^i = \frac{\partial \mathbf{h}}{\partial \mathbf{w}^i} \,. \tag{15}$$

The considered parameters for the Gaussian noise may be for example in the form:

$$\mathbf{Q}_k^i = a \cdot e^{\frac{s-1}{b}} \cdot I \tag{16}$$

$$\mathbf{R}_k = c \cdot e^{\frac{s-1}{d}} \cdot I \,, \tag{17}$$

where: a, b, c, d real, positive numbers, I is identity matrix which dimension depends on the dimension of the model that is the number of connection weights for the particular neuron g, s is the number of learning epoch [7].

In this approach the typical problem of multidimensional optimization of gradient descent type [8] is changed into prediction – correction problem. The prediction phase uses the state estimate from the previous time step to produce an estimate of the state at the current time step – Eq. (13). In the correction phase the quality of estimation is included, see Eq. (14).

3. Development of Kalman Filter Learning Technique

The choosing of the proper (according to considered numerical problem) architecture of ANN is very important. The ANN too small may be insufficient to model the data, too large ANN may be very time consuming and the over fitting may occur. It leads to the very poor results as far as testing set is considered [1]. The choosing of the proper architecture of ANN may be done automatically without involving a priori guess from the network constructor.

In standard ANN learning techniques the whole network is modified during learning process. The magnitude of networks weights correction is the issue that differs from each pattern to another. The KF model gives us also the equipment to measure the quality of estimation, made during learning for the each single weight. The areas (particular nodes or chosen parts of the ANN) can be selected were the quality of estimation is insufficient. Then the network (due to decoupling) may be than learned selectively. Only the areas that are under the average quality of estimation level are changed. In the paper this technique is presented as far as the particular nodes are concerned, but the decoupling into single weights may be applied also.

The techniques presented in this paragraph were proposed by the author and their effectiveness was examined by a lot of tests presented at the end of the paper.

3.1. ANN Pruning

There are two main approaches to the automatically driven ANN architecture setting. In the paper the pruning method was chosen [1]. It assumes starting from large ANN and making it smaller during learning process by erasing the particular weights between neurons. After initial learning the weights are examined as far as their impact into the quality of learning is considered. Ineffective weights are erased. Then the smaller network is learned and the procedure is repeated until it is beneficial for the learning and testing process.

The initial test showed that methods dedicated for traditional ANN pruning cannot be applied for KF learning. Due to the fact that in KF model estimation is made – the statistically based methods of pruning were adopted [9], but the scheme had to be adjusted into KF technique.

Let

$$\mathbf{w} = [w^1, w^2, ..., w^I] \tag{18}$$

be the vector made of all ANN weights. The statistics Λ is calculated:

$$\Lambda(w^i) = \ln \frac{\left| \sum_{p=1}^{L} w^i - \eta \frac{\partial E(p)}{\partial w^i} \right|}{\eta \sqrt{\sum_{p=1}^{L} \left(\frac{\partial E(p)}{\partial w^i} - mean_{\{p=1,2,...,L\}} \frac{\partial E(p)}{\partial w^i} \right)^2}} \tag{19}$$

there the summation is made according to p from 1 to the L-th (last element of the learning set), η is a constant number (learning rate), $mean_{\{p=1,2,...,L\}}$ – is the arithmetical mean in the learning set, $E(p)$ is the learning error for input x^p, that is for p-th learning pattern:

$$E(p) = ||\mathbf{h}(\mathbf{t}_p) - \mathbf{y}_p||^2 \tag{20}$$

where t_p is the pth target whereas h represents the ANN structure, see Eq. (4), $||.||$ is the standard Euclidean norm.

The larger values of Λ statistics for the particular node means that these weights are significant for the learning process. There are the following parameters that were stated to be the most important during the process of pruning:

- the initial ANN is learned until S epoch is reached;
- from $S_{start} = S + 1$ epoch network is changed;
- setting the parameters of white noise in the Eqs. (7)–(8) to enable smaller network learn as fast as at the beginning of the learning process; after cutting the parameters of white noise is shifted back to the value from $S_{reset} < S_{start}$ epoch;
- choosing of k the number of epochs between consecutive cuts;

  setting the Λ_{edge} limit value for the Λ; weights that are below Λ_{egde} are erased;
- after each cutting the testing error is examined. If it is occurred to be growing, the procedure is stopped and the network is shifted back to the last better architecture.

For the particular problem without the previous knowledge of ANN behavior the genetic algorithm may be responsible for the optimal parameter selection.

3.2. Selective Learning of ANN Based on Approximate Covariance Error Matrix

Using the KF learning method it is enable to build separate model for each node of the network. Due to the possible highly asymmetric values of the inputs of the ANN and different values of the initial weights – the values of the weights may differ much. Moreover in KF model their values are estimated with the known quality. By the examining of the diagonal values of the matrix (13), it is possible to select those weights for which the estimation quality is unsatisfactory.

The proposed algorithm is based on dynamic changes of the areas of ANN for the current learning.

Calculating matrix $\mathbf{P}^i_{k+1}$ that approximates

$$E(\mathbf{w}^i_{k+1} - \widehat{\mathbf{w}}^i_{k+1})(\mathbf{w}^i_{k+1} - \widehat{\mathbf{w}}^i_{k+1})^T, \tag{21}$$

where $\widehat{\mathbf{w}}^i_{k+1}$ is the estimator for $\mathbf{w}^i_{k+1}$ calculated after presenting k-th learning pattern, on the main diagonal of the matrix the level of estimation quality for the single weight can be obtained:

$$P^i_{k+1}(m) \approx E\left(w^i_{k+1}(m) - \widehat{w}^i_{k+1}(m)\right)\left(w^i_{k+1}(m) - \widehat{w}^i_{k+1}(m)\right)^T, \tag{22}$$

for m from 1 to the numer of weights in the considered neuron. These values represents the approximation of errors of estimation for the single m-th weight of the i-th neuron in the selected node.

After initial examination the quality of estimation represented by Eq. (22) was treated separately inside each layer. The following algorithm was adopted (presented here for first layer):

- initial learning for $s = 1, 2, \ldots, S_0$ epochs, and then inside layer,
- calculating the mean level of error of estimation inside the layer

  $$MEAN = mean_{m,i} P^i_{k+1}(m), \tag{23}$$

  for i taken from 1 to the range of first layer, m taken from 1 to the number of weights inside layer,

- calculating the mean level of error of estimation inside the i-th neuron

$$Mean(i) = mean_m P^i_{k+1}(m), \tag{24}$$

m taken from 1 to the number of weights inside layer

- selecting those neurons that were learned very good, by finding i such that

$$Mean(i) < \alpha MEAN, \tag{25}$$

where α is assumed percentage constant,

- for those neurons for S_{freeze} epochs the values remains constant,
- the rest of the neurons were learned for S_{learn} epochs,
- calculating $MEAN$, $Mean(i)$ and repeating the procedure until they differ much.

4. Numerical Testing of the Proposed Solutions

Many numerical tests proving the effectiveness of the basic KF method were conducted. Considered numerical problems came from experiments and situ measurements considering: mining tremors, cyclic loading of steel and concrete specimens, hysteresis loops for superconductor. The tests were designed to develop a methodology that could be helpful during modeling and predicting time series and other time dependent phenomenon. The comparison between KF solution and standard neural solution (Back Propagation learning technique, Rprop) was investigated. In the absence of neural solution the results were compared to the available classical methods of mechanics of materials. Considered numerical problems were stated to be very difficult to solve either by ANN or for classical methods [7].

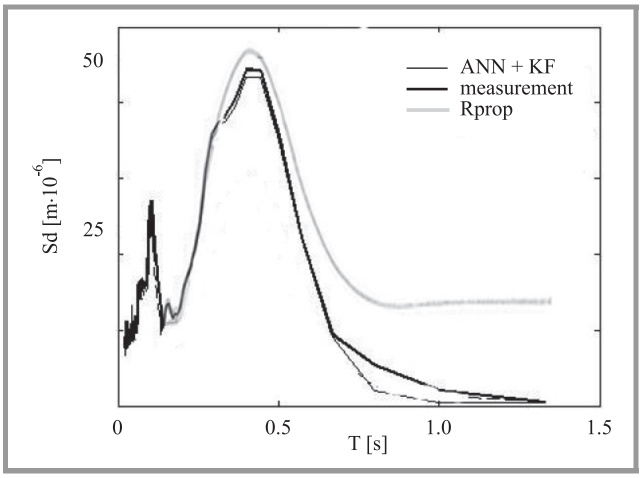

Fig. 2. Neural prediction of response spectra from mining tremors.

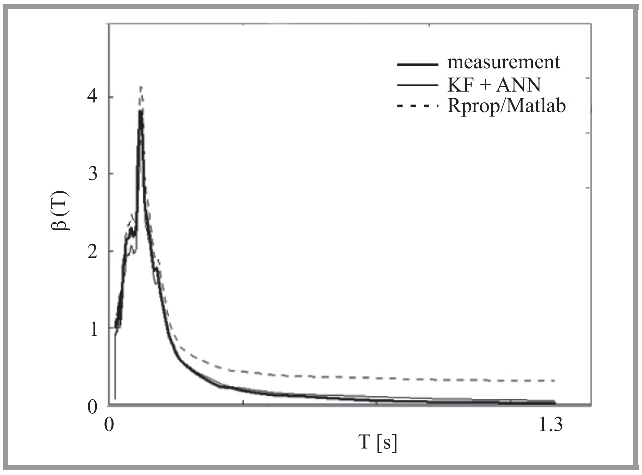

Fig. 3. Neural prediction of response spectra from mining tremors.

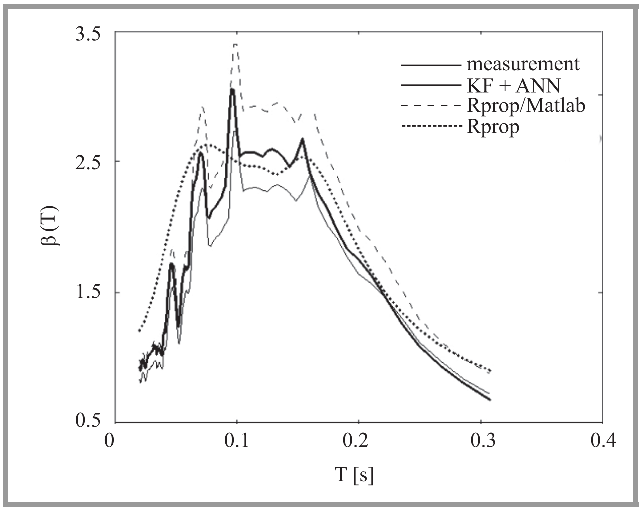

Fig. 4. Neural prediction of response spectra from mining tremors.

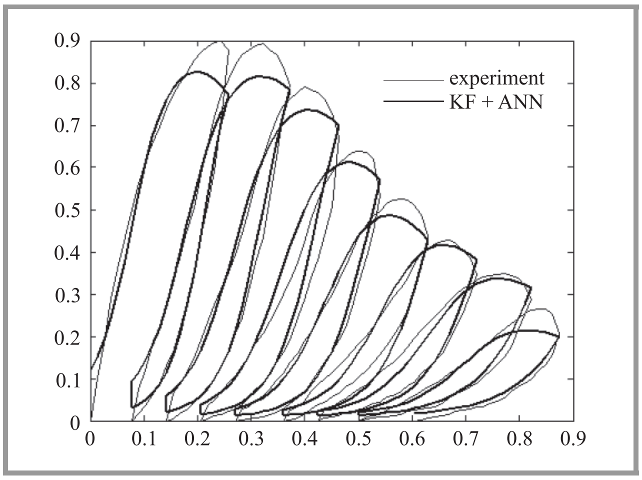

Fig. 5. The analysis of cyclic behavior of concrete specimens.

The chosen results are presented in graphical demonstrative form below, see Figs. 2–9 [10]–[12]. The chosen results for Selective learning of ANN based on Approximate

Covariance Error Matrix and pruning are presented below for the selected numerical problem, see Figs. 10–11 [7].

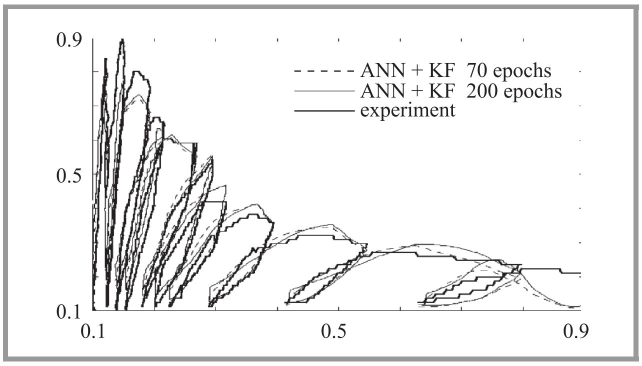

Fig. 6. The analysis of cyclic behavior of concrete specimens.

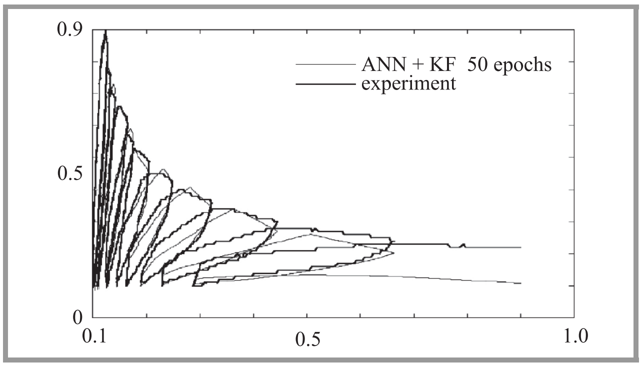

Fig. 7. The analysis of cyclic behavior of concrete specimens.

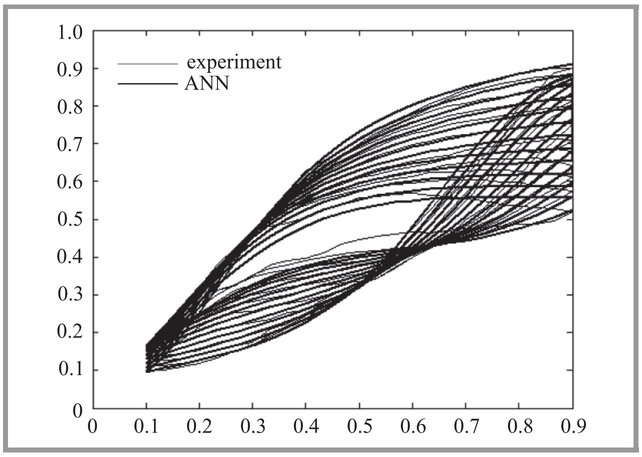

Fig. 8. Simulation and prediction of cyclic loading of steel specimens.

All the algorithms were implemented using Matlab. It is a high-level language and interactive environment for numerical computation, visualization, and programming. Is provides very efficient libraries for matrices operations. KF algorithm was vectored. Neural Network Toolbox including gradient descent methods, conjugate gradient methods, the Levenberg-Marquardt algorithm, and the resilient backpropagation algorithm was used as the reference for investigating the efficiency of KF learning method. The set of good practices were found for applying KF as a learning technique for ANN. It is presented in Table 1.

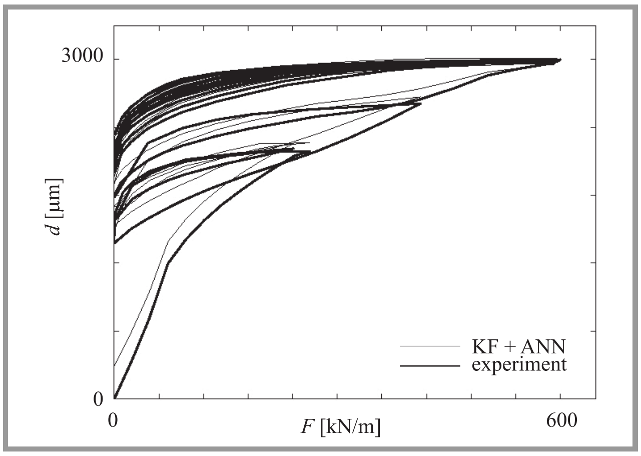

Fig. 9. Simulation of hysteresis loops for superconductor.

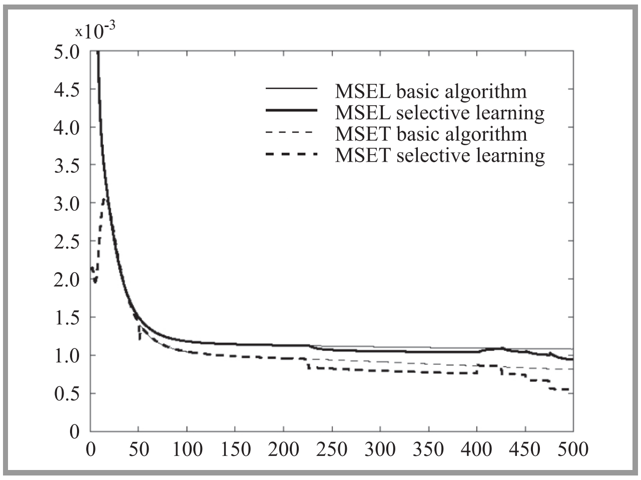

Fig. 10. The pruning of KF neural network for simulation of hysteresis loops for superconductor, MSEL=Mean Square Errors in the Learning Set, MSET=Mean Square Errors in the testing set.

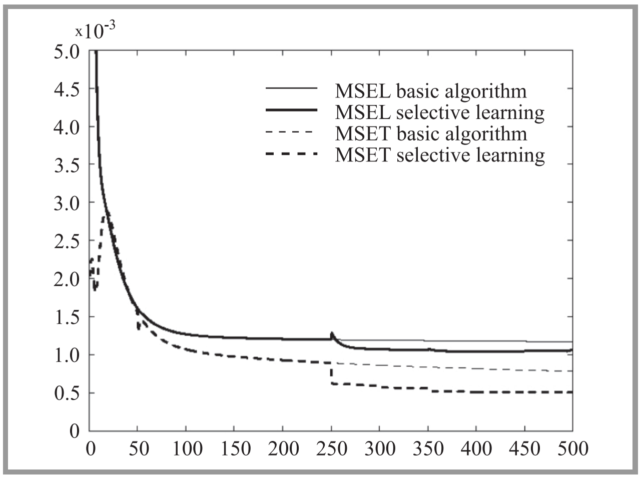

Fig. 11. The pruning of KF neural network for simulation of hysteresis loops for superconductor.

Table 1
Good practices for applying KF as a learning technique for ANN

Problem	KF+ANN good practice
Number of epochs	About 10^2
Input vector	Time window include
Input vector magnitude	[0.2, 0.8]
Number of layers	2
Activation function	Nonlinear (5) or (6)
Noise function for Eqs. (6)–(7)	(16)–(17)
Number of weights in ANN	K About 10
Starting ANN for pruning	Input-50-50-output
Parameter Λ_{edge} for pruning	0.75
Parameter α for Eq. (25)	1

5. Conclusions

Kalman filtering method was stated to be very beneficial during learning of ANN solving numerical problems of real data simulation and prediction. Data for time series with correlation are modeled very precisely. For those type of data the obtained results shown that this method of learning seemed to be matched to the nature of considered numerical problem. For this reason it is beneficial to apply KF instead of traditional ANN learning methods like resilient propagation or Levenberg-Marquardt algorithm.

The effective method for automatic pruning of ANN learned by KF was proposed. It enable to shorten the time of calculations by learning the network containing smaller number of parameters.

The efficient method for expanding the basic KF learning method was developed. The analysis of approximate error covariance matrix values shown that this is the successful way to control the quality of learning process. It may result in avoiding the retarding of the learning process manifested by negligible changes of error of the ANN during learning process.

Both proposed methods are using the specific properties of KF learning method and enable to obtain better numerical solution for the considered problem of time series modeling and predicting.

Acknowledgements

The author is deeply indebted to Professor Zenon Waszczyszyn the supervisor of PhD thesis in which the main results of this paper are presented.

References

[1] S. Haykin, *Neural Networks: A Comprehensive Foundation*. 2nd ed. New Jersey: Prentice-Hall, 1999.

[2] S. Haykin (Ed.), *Kalman Filtering and Neural Networks*. New York: Wiley, 2001.

[3] R. M. García-Gimeno, C. Hervás-Martínez, M. I. de Silóniz, "Improving artificial neural networks with a pruning methodology and genetic algorithms for their application in microbial growth prediction in food", *Int. J. Food Microbiol.*, vol. 72, iss. 1–2, pp. 19–30, 2002.

[4] P. Trebatický, J. Pospichal, "Neural Network training with extended Kalman Filter using graphics processing unit", in *Proc. 18th Int. Conf. Artif. Neural Netw. ICANN 2008*, Prague, Czech Republic, 2008, LNCS 5164. Berlin-Heidelberg: Springer, 2008, pp. 198–207.

[5] M. Aparecido de Oliveira, "An application of Neural Networks trained with Kalman Filter variants (EKF and UKF) to heteroscedastic time series forecasting", *Appl. Mathem. Sci.*, vol. 6, no. 74, pp. 3675–3686, 2012.

[6] R. E. Kalman, "A new approach to linear filtering and prediction problems", *Trans. ASME J. Basic Engin.*, vol. 82, no. 1, pp. 35–45, 1960.

[7] A. Krok, "Analysis of selected problems of structural mechanics and materials by using Artificial Neural Networks and Kalman filters", Ph.D. Thesis, Cracow University of Technology, Cracow, Poland, 2007 (in Polish).

[8] C. M. Bishop, *Neural Networks for Pattern Recognition.*, New York: Oxford University Press, 1995.

[9] L. Prechelt, "Connection pruning with static and adaptive pruning schedules", *Neurocomputing*, vol. 16, no. 1, pp. 49–61, 1997.

[10] A. Krok and Z. Waszczyszyn, "Neural prediction of response spectra from mining tremors using recurrent layered networks and Kalman filtering", in *Proc. 3rd MIT Conf. Comput. Fluid and Solid Mechanics*, Cambridge, USA, 2005, J.-K. Bathe, Ed., Elsevier, 2005, pp. 302–305.

[11] A. Krok and Z. Waszczyszyn, "Kalman filtering for neural prediction of response spectra from mining tremors", *Computers and Structures*, vol. 85, iss. 15–16, pp. 1257–1263, 2007.

[12] A. Krok, "An improved Neural Kalman Filtering Algorithm in the analysis of cyclic behavior of concrete specimens", *Comp. Assist. Mechan. Engin. Sciences*, vol. 18, pp. 275–282, 2011.

[13] A. Krok, "Enhancing NDEKF Algorithm of Artificial Neural Network learning for simulation o hysteresis loops for superconductor", in *Computational Intelligence: Methods and Applications*, L. Rutkowski, L. Zadeh, R. Tadeusiewicz, and J. Żurada, Eds. Warszawa: Akademicka Oficyna Wydawnicza EXIT, 2008.

Agnieszka Krok received her M.Sc. in the field of stochastic processes at the Jagiellonian University, Cracow, Poland and Ph.D. degree in the field of neural networks at Tadeusz Kościuszko Cracow University of Technology, Poland, in 2003 and 2007, respectively. From 2009 she is an Assistant Professor at Faculty of Physics, Mathematics and Computer Science, Tadeusz Kościuszko Cracow University of Technology. Her main scientific and didactics interests are focused mainly on Artificial Intelligence: Artificial Neural Networks, Genetic Algorithms, and additionally on Parallel Processing and Cryptography.

E-mail: agakrok@poczta.fm
Faculty of Physics, Mathematics and Computer Science
Tadeusz Kościuszko Cracow University of Technology
Warszawska st 24
31-155 Cracow, Poland