Estimation of Computational Complexity of Sensor Accuracy Improvement Algorithm Based on Neural Networks

Volodymyr Turchenko, Volodymyr Kochan, Anatoly Sachenko

Institute of Computer Information Technologies
Ternopil Academy of National Economy
3 Peremoga Square, 46004, Ternopil, UKRAINE
vtu@tanet.edu.te.ua

Abstract. The estimation method of computational complexity of sensor data acquisition and processing algorithm based on neural networks is considered in this paper. An application of this method allows to propose a three-level structure of distributed sensor network with improved accuracy.

1 Introduction

The problem of accuracy improvement of sensor data acquisition and processing [1] is relevant in distributed sensor networks (DSN) [2]. Thus, sensor drift is the predominant compound of common error [3]. Using structure-algorithmic methods of accuracy improvement [4], for example testing and calibration, requires the interruption of data acquisition process that limits the usage of these methods. A prediction method [3] does not have this lack and allows correcting sensors instability during the system's operation. However, the specific features of sensor drift don't provide necessary prediction accuracy in the usage non-adaptive mathematical models. Therefore, it is expedient to combine testing or calibration methods with a prediction method [5]. The authors propose to use artificial neural networks (NN) for sensor drift prediction and accuracy improvement of sensor data acquisition and processing in DSN [6]. However, the use of neural networks requires the corresponding computational resources. Therefore, it is necessary to evaluate computational complexity of sensor data acquisition and processing algorithm based on neural networks in order to develop DSN hardware structure.

2 Sensor Data Acquisition and Processing Algorithm Based on Neural Networks

Sensor data acquisition and processing algorithm based on neural networks uses three kinds of neural networks: (i) integrating "historical" data NN, (ii) approximating NN and (iii) predicting NN. These neural networks were mainly considered in other previous publications of the same team of authors [7-9].

The model of heterogeneous neural network (Fig. 1) is used as predicting NN (PNN), which consists of N input neurons, M neurons of the hidden layer with the logistic activation function and one output neuron with the linear activation function [7]. The output neuron value

G. Dorffner, H. Bischof, and K. Hornik (Eds): ICANN 2001, LNCS 2130, pp. 743-748, 2001

$$y = \sum_{j=1}^{M} w_{j3} h_j - T_o \ , \qquad (1)$$

where w_{j3} is the synapse from j-neuron of hidden layer to output neuron; T_o is the threshold of the output neuron; $h_j = F(\sum_{i=1}^{N} w_{ij} x_i + \sum_{k=1}^{M} w_{kj} h_k(t-1) + w_{3j} y(t-1) - T_j)$ is the output value of j-neuron of hidden layer in t moment of time; w_{ij} is the synapse from i-input neuron to j-neuron of the hidden layer; x_i is the i-element of input vector p; w_{kj} is the synapse from k-neuron of the hidden layer to j-neuron of the same layer; $h_k(t-1)$ is the output value of k-neuron of hidden layer in the previous moment of time $t-1$; w_{3j} is the synapse from output neuron to j-neuron of the hidden layer; $y(t-1)$ is the value of output neuron in the previous moment of time $t-1$; T_j is the threshold of j-neuron of the hidden layer.

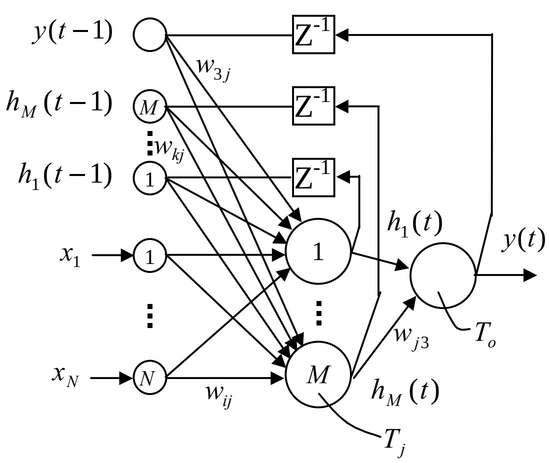

Fig. 1. Structure of Predicting Neural Network

The back propagation error algorithm [7, 9] is used for PNN training. The sum-squared error $E^p(t)$ for each training vector p and common training error $E(t)$ is calculated as

$$E^p(t) = 0.5(y^p(t) - d^p)^2 , \ E(t) = \sum_{p=1}^{P} E^p(t) , \qquad (2)$$

where $y^p(t)$ is current and d^p is desired PNN output value for training iteration t for each training vector p, the error of output neuron is $b_3^p(t) = y^p(t) - d^p$.

The adaptive learning rate [7, 9] for output neuron with linear activation function

$$\alpha_3^p(t) = 1/(1 + \sum_{j=1}^{M} h_j^p(t)) \qquad (3)$$

and for neurons of the hidden layer with the logistic activation function

$$\alpha_2^p(t) = 4/(R^2 \sum_{j=1}^{M} (w_{j3}^p(t))^2 h_j^p(t)(1 - h_j^p(t)) , \qquad (4)$$

where $R = 1 + \sum_{i=1}^{N} (x_i^p(t))^2 + (y(t-1))^2 + \sum_{j=1}^{M} (h_j^p(t-1))^2$.

The synapses and neuron's thresholds are changed during the training process taking into account (3) and (4)

$$w_{ij}^p(t+1) = w_{ij}^p(t) - \alpha_2^p(t) b_3^p(t) w_{j3}^p(t) h_j^p(t)(1-h_j^p(t)) x_i \, , \tag{5}$$

$$w_{j3}^p(t+1) = w_{j3}^p(t) - \alpha_3^p(t) b_3^p(t) h_j^p(t) \, , \tag{6}$$

$$w_{kj}^p(t+1) = w_{kj}^p(t) - \alpha_2^p(t) b_3^p(t) w_{j3}^p(t) h_j^p(t)(1-h_j^p(t)) h_k^p(t-1) \, , \tag{7}$$

$$w_{3j}^p(t+1) = w_{3j}^p(t) - \alpha_2^p(t) b_3^p(t) w_{j3}^p(t) h_j^p(t)(1-h_j^p(t)) y^p(t-1) \, , \tag{8}$$

$$T_o^p(t+1) = T_o^p(t) + \alpha_3^p(t) b_3^p(t) \, , \tag{9}$$

$$T_j^p(t+1) = T_j^p(t-1) + \alpha_2^p(t) b_3^p(t) w_{j3}^p(t) h_j^p(t)(1-h_j^p(t)) \, . \tag{10}$$

The training process of PNN according to the expressions (1-10) is executed while the common training error $E(t)$ does not become less than the required value.

The experimental researches (by simulation modeling) of the proposed algorithms have been done using generalized mathematical models of sensors drift "with acceleration" (Fig. 2). The data from 0-30 calibrations with step 5 on each curve are used for NN training. The prediction interval is 30-60 calibrations with step 1. The average and the maximum percentage prediction errors do not exceed 14% and 27% (Fig. 3) for sensors drift "with acceleration" using integrating "historical" data NN, approximating NN and predicting NN. It allows the improvement of sensor data acquisition and processing accuracy in 3-5 times with a simultaneous increase of interesting interval in 6-12 times [6, 9].

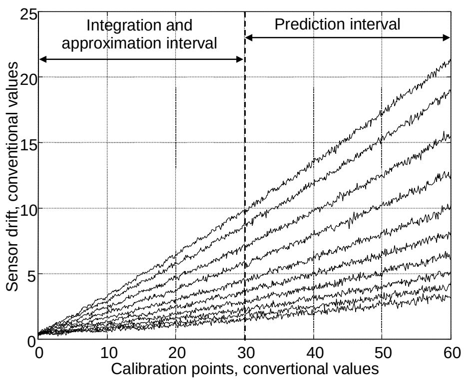

Fig. 2. Sensor drift "with acceleration"

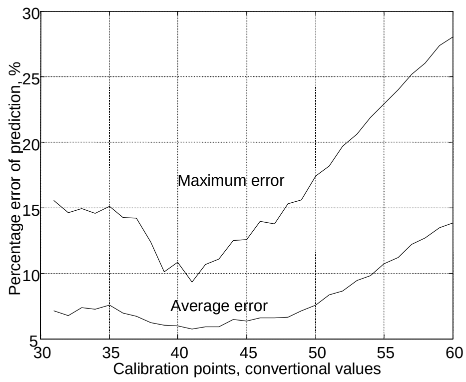

Fig. 3. Prediction results of sensor drift

3 An Approach to Estimation of Computational Complexity of Algorithm

Using the proposed NN-based algorithms of sensor data acquisition and processing defines the corresponding requirements to DSN structure and the necessary computing components. The analysis of time parameters of sensors and known DSN structures [1, 10] allows defining three scales of real time, which is necessary to consider during the analysis of computational productivity of the DSN components:

- The required scanning period for the majority of sensors is not less than 50 µs. Therefore, the data acquisition frequency is $2 \cdot 10^4$ Hz;

- The necessary period of sensor drift correction factor calculation is not less than 20 min. Therefore the frequency is $8.3 \cdot 10^{-4}$ Hz;
- The necessary period of replacement of the sensor drift mathematical model is not less than 20 hours. Therefore, the frequency is $1.4 \cdot 10^{-5}$ Hz.

All the above procedures should run in the correspondent real time scale.

Let us consider a technique of computational complexity (and the necessary productivity of computing components) estimation for the algorithm of sensor drift mathematical model replacement using the PNN structure.

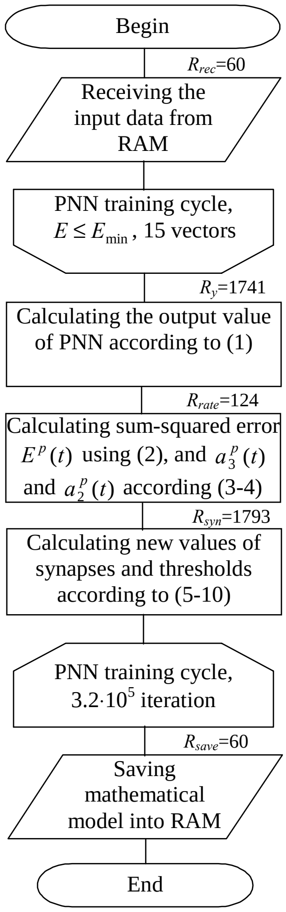

Fig. 4. General algorithm of sensor drift mathematical model replacement

In a general case, productivity P of the computing device [11]

$$P = K \cdot F \cdot R / N , \tag{11}$$

where K is the number of input data channels, F is the frequency of the data entrance on the any of K input channels, $R = R(N)$ is the computational complexity of the algorithm (number of addition/multiplication operations (float-point operations) per second) and N is the number of input data elements. Furthermore, the analysis of computational complexity will be provided for the situation, when only one value from input arrays (one value from each appropriate data array) goes to the computing device on one input channel in order to simplify the calculation of operations.

The productivity of the necessary computing components for the algorithm of sensor drift mathematical model replacement according to Fig. 4 and (11)

$$P_{DRFMODEL} = K \cdot F \cdot R_{DRFMODEL} / N = 2.5 \cdot 10^5 \text{ op/sec},$$

where K =1; F =$1.4 \cdot 10^{-5}$ Hz is the frequency of the mathematical model of correction factor replacement;

$$R_{DRFMODEL} = R_{rec} + 3.2 \cdot 10^5 \cdot 15 \cdot (R_y + R_{rate} + R_{syn}) + R_{save} = 1.8 \cdot 10^{10} \tag{12}$$

operations - is the computational complexity of algorithm ($3.2 \cdot 10^5$ is the number of required iterations; 15 is the number of training vectors; $R_y = 1741$ operations is the computational complexity of PNN output value calculation according to (1); $R_{rate} = 124$ operations is the computational complexity of sum-squared error calculation (2) and the adaptive learning rate for neurons of output (3) and hidden (4) layers; $R_{syn} = 1793$ operations is the computational complexity of the synapses and the thresholds modification for all layers according to (5-10)); N =1.

The estimation of computational complexity of algorithms is necessary in order to choose the optimum DSN structure, which executes NN-based algorithms in order to improve the accuracy of sensor data acquisition and processing. The preliminary analysis has shown that the data acquisition algorithm (*A*), the algorithm of correction factor of sensor drift mathematical model replacement (*B*), the algorithms of PNN training set formation using additional approximating NN (*C*) [7, 9] and the use of a set of integrating "historical" data neural networks (*D*) [6, 8] are required the largest computational recourses. The results of productivity P_s estimation for the above algorithms for single data acquisition channel of DSN are presented in Table 1.

4 Experimental Researches for DSN Structure Design

A personal computer (PC) is the main computing DSN component [10, 12]. Therefore, it is necessary to evaluate productivity of modern PCs on the basis of different processors using the proposed approach. The fragment of PNN calculation routine according to (1-10) is used as the test routine for the estimation of computational complexity and productivity of the algorithm. The computational complexity of the test routine

$$R_{TEST} = 1 \cdot 10^6 \cdot (R_{y1} + R_{rate} + R_{syn}) = 3.7 \cdot 10^9 \text{ operations},$$

where $1 \cdot 10^6$ is the iteration number of the test routine, R_y, R_{rate}, R_{syn} are operations according to (12). The productivity of the tested PCs $P_{TEST} = R_{TEST} / t_{TEST}$, where t_{TEST} is the execution time of the test routine in seconds (Table 2).

Table 1. Computational complexity for single data acquisition channel

Algorithm	P_s, op	R_s, op/s
A	27	$5.4 \cdot 10^5$
B	$1.8 \cdot 10^{10}$	$2.5 \cdot 10^5$
C	$8.6 \cdot 10^8$	$1.2 \cdot 10^4$
D	$1.3 \cdot 10^{10}$	$1.8 \cdot 10^5$

Table 2. Productivity estimation of modern PCs

PC	t_{TEST}, s	P_{TEST}, op/s
Intel Pentium	412	$8.9 \cdot 10^6$
AMD Pro	282	$1.3 \cdot 10^7$
Intel Celeron	69	$5.4 \cdot 10^7$
Intel P-III	52	$7.1 \cdot 10^7$

The productivity analysis of modern PCs (see Table 2) has shown that a single PC doesn't provide the fulfillment of all the considered above NN-based algorithms in real time scale for a multi-channel DSN, especially when it is together with user interaction. The known data acquisition modules [12, 13] fulfill sensor signal processing using only permanent algorithms stored into ROM. Therefore, it is expedient to add the third (middle) level of sensor signal processing [1] into a two-level structure of known DSN. This level should fulfill the algorithms of the current error correction. The elements of the middle level should execute the algorithms, which change during the DSN operation. For this purpose, they should provide a remote reprogramming mode that allows them to be recognized as intelligent nodes [14].

5 Conclusions

The proposed approach for a computational complexity estimation of neural network based algorithms and an appropriate productivity of necessary computing components allows to propose three-level DSN structure with the improved accuracy and effectiveness of sensor data processing in 3-5 times.

Acknowledgments

The authors wish to acknowledge Prof. A. Melnyk from National University “Lvivska Politechnika”, Ukraine for the fruitful discussions during the preparation of this paper. The authors also would like to thank the European organization INTAS for its financial support, grant reference number INTAS-OPEN-97-0606.

References

1. S.Iyengar: Distributed Sensor Network - Introduction to the Special Section, Transaction on Systems, Man, and Cybernetics, Vol. 21, No.5 (1991) 1027-1031
2. A.Sachenko, V.Kochan, V.Turchenko: Intelligent Distributed Sensor Network, Proceedings of the 15th IEEE Instrument. and Measurement Tech. Conf. St. Paul, USA (1998) 60-66
3. A.Sachenko, J.Tverdyj: Perfection of Temperature Measurement Methods. Technika, Kiev Ukraine (1983)
4. Tuz Y: Structural Methods of Accuracy Improvement of Measuring Devices. Vyshcha Shkola, Kiev Ukraine (1976)
5. A. Sachenko: Development of Measurement Precision Algorithm for Intelligent Sensor System, 41 Internationalen Wissenschaft Kolloquium, Ilmenau, Germany (1996) 645-650
6. V.Turchenko, R.Kochan, A.Karachka: Prediction of Sensor Drift Conversion Characteristics Using Neural Networks, Journal of Ternopil State Technical University, Vol 5, No 4. Ternopil, Ukraine (2000) 102-108
7. V.Golovko, J.Savitsky, A.Sachenko, V.Kochan, V.Turchenko, T.Laopoulos, L.Grandinetti: Intelligent System for Prediction of Sensor Drift, Proceedings of the International Conference on Neural Networks and Artificial Intelligence. Brest, Belarus (1999) 126-135
8. A.Sachenko, V.Kochan, V.Turchenko: Sensor Drift Prediction Using Neural Networks, Proceedings of the International Workshop on Virtual and Intelligent Measurement Systems. Annapolis, USA (2000) 88-92
9. A.Sachenko, V.Kochan, V.Turchenko, V.Golovko, J.Savitsky, A.Dunets, T. Laopoulos: Sensor Errors Prediction Using Neural Networks, Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks, Vol. IV. Como, Italy (2000) 441-446
10. V.Golovko, L.Grandinetti, V.Kochan, T.Laopoulos, A.Sachenko, V.Turchenko: Sensor Signal Processing Using Neural Networks, Proceedings of the IEEE Region 8 International Conference Africon’99. Cape Town, South Africa (1999) 339-344
11. A. Melnyk: Real-Time Specialized Computer Systems: Lectures notes, State University “Lvivska Politechnika” (1996)
12. http://content.honeywell.com/sensing/prodinfo/sds/prodsheets/micro.pdf
13. http://www.fluke.com/products/home.asp?SID=7&AGID=6&PID=5308
14. A.Sachenko, V.Kochan, V.Turchenko, V.Tymchyshyn, N.Vasylkiv: Intelligent Nodes for Distributed Sensor Network, Proceedings of the 16th IEEE Instrumentation and Measurement Technology Conference. Venice, Italy (1999) 1479-1484