UDC 004.891.3 : 004.3

T.O. GOVORUSCHENKO

Khmelnitskiy National University, Ukraine

MODEL OF DECISION MAKER OF REPEATED APPLICATION SOFTWARE TESTING SYSTEM

In article decision maker of repeated application software testing system (RASTS) simulation in Matlab and training of this ANN is considered. Conclusions about dependence results of ANN training (training time and performance) and training algorithm, criterion of training quality evaluation, size of training sample is concluded.

repeated applications software testing system, Decision maker of repeated application software testing system, Model of decision maker (ANN), ANN training, Training algorithm, Criterion of training quality evaluation, Training sample

Introduction

For solving of problem of software testing efficiency increasing repeated application software testing system [1, 2] was developed (fig. 1).

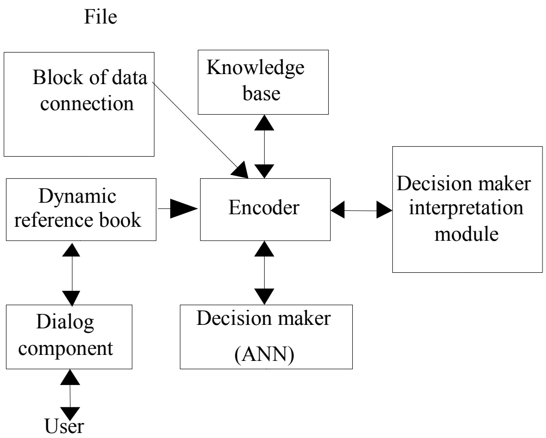

Fig. 1. Structure chart of RASTS

On the block of data connection user file with results of the basic testing, represented as a testing journal «Testing method – Testing operation – Finding mistake type» is fed. The file data are processed by the encoder. Encoder transforms input data from a linguistic form in a quantitative form, fills the knowledge base by input data and forms entrances vectors for decision maker. Knowledge base contains tables with input data of system, auxiliary tables and tables with rules for the forming deduction about a necessity and method(s) of the repeated testing. Solution of tasks of hidden mistakes finding is based on category model of process of repeated testing [3], in which considering of importance of each type mistakes, interference of mistakes types, fuzzy input data about existent mistakes is allowed, and is possible with the artificial neuron network (ANN) using. Therefore by decision maker is used an artificial neuron network, on the entrances of which information about methods and operations of the basic testing and types of finding during the basic testing software mistake(s) is given, and category level of hidden mistakes is decision maker results. Results of decision maker are given to encoder, which fills knowledge base by result data, transforms of resulting vectors in a linguistic form and are transmitted on the decision maker interpretation module. Decision maker interpretation module on the basis of rules [2] generates a deduction about a necessity and method(s) of the repeated testing, which is transmitted through dialog component to the user. Dynamic guide gives to user information about input file format, known basic software testing techniques and operations, finding mistake types and transmits all messages some system components to user. The result of system functioning is deduction about repeated testing necessity and advisable repeated testing method(s).

For simulation in Matlab from ANN (described in [3 – 5]) effector layer output functionals $y_1,...,y_i,...,y_m$

were deleted, only outputs $Y_1, Y_2, ..., Y_h, ...Y_m$ are keeping active, in other words such ANN was simulated [6]:

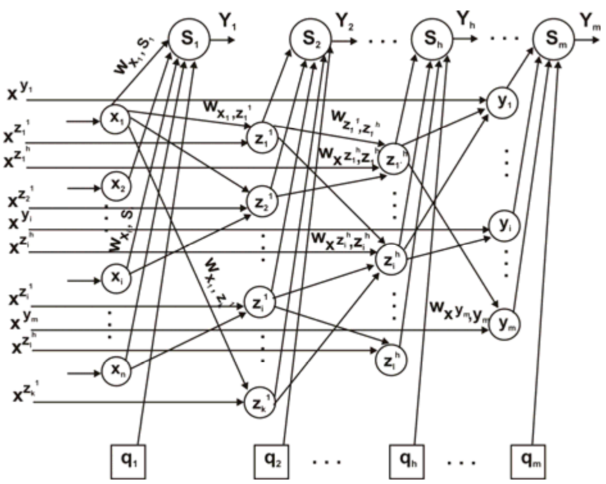

Fig. 2. Decision Maker of RASTS

1. Decision Maker (ANN) Structure in Matlab 6.1

ANN structure chart developed using in Matlab 6.1 is presented on figure 3 [6].

By statistics [7] tester tests program using as a rule four basic software testing techniques, therefore on each of inputs $q_1 - q_4$ ”one” must be transmitted.

Testing is processed using one of testing techniques, which were formed as a result of two testing techniques under the one number unification (table 1).

Table 1

Numbering of application software testing techniques

Number	Software testing techniques
1	Functional testing Correctness testing
2	Elements testing
3	Testing of independent paths (branches) Top-down testing
4	Bottom-up testing Testing of elements conjugation

Tester tests program using no more than four operations of the same basic software testing technique [7], therefore on each of inputs Input2 – Input5 no more than four testing operation numbers [6] can be transmitted. On inputs Input2 (x^{Z_1}), Input3 (x^{Z_2}), Input4 (x^{Z_3}), Input5 (x^{Z_4}) basic software testing operations numbers are transmitted.

On input Input1 (x) were transmitted finding during basic testing mistakes types numbers. Inasmuch as by statistics [7] maximum 14 – 15 mistakes was present in program, therefore on this input no more than 16 finding mistakes types numbers can be transmitted.

Values of ANN inputs lies in the ranges of table 2. Numeration of software testing operations and finding mistakes types is shown in [6].

Table 2

ANN inputs range

Input	Range
q_1	0..1
q_2	0..1
q_3	0..1
q_4	0..1
Input1 (x)	0..22
Input2 (x^{Z_1})	0, 20..32, 50, 51
Input3 (x^{Z_2})	0, 41..46, 52, 53
Input4 (x^{Z_3})	0, 10..19, 33..40
Input5 (x^{Z_4})	0, 1..9, 47..49

Each of outputs Y_i is corellated with i -th cathegory level and is possess the value “one”, if ANN prognosed presence of *i*-th cathegory level mistakes, else output Y_i is possess the value “zero”.

Architecture such ANN in Matlab 6.1 and training vectors sequence example is described in [6]. ANN initialization technique is Nguyen-Widrow technique. Expedient ANN training algorithm and criterion of training quality evaluation were determined by experience.

2. ANN training algorithm and criterion of training quality evaluation choice

For ANN training algorithm and criterion of training quality evaluation choice ANN was investigated during the training with training sample of 66 vectors, 497 vectors, 2250 vectors. Training was realized by different training algorithms using different criterions of training quality evaluation. On the base of investigation results tables 3, 4, 5 were formed.

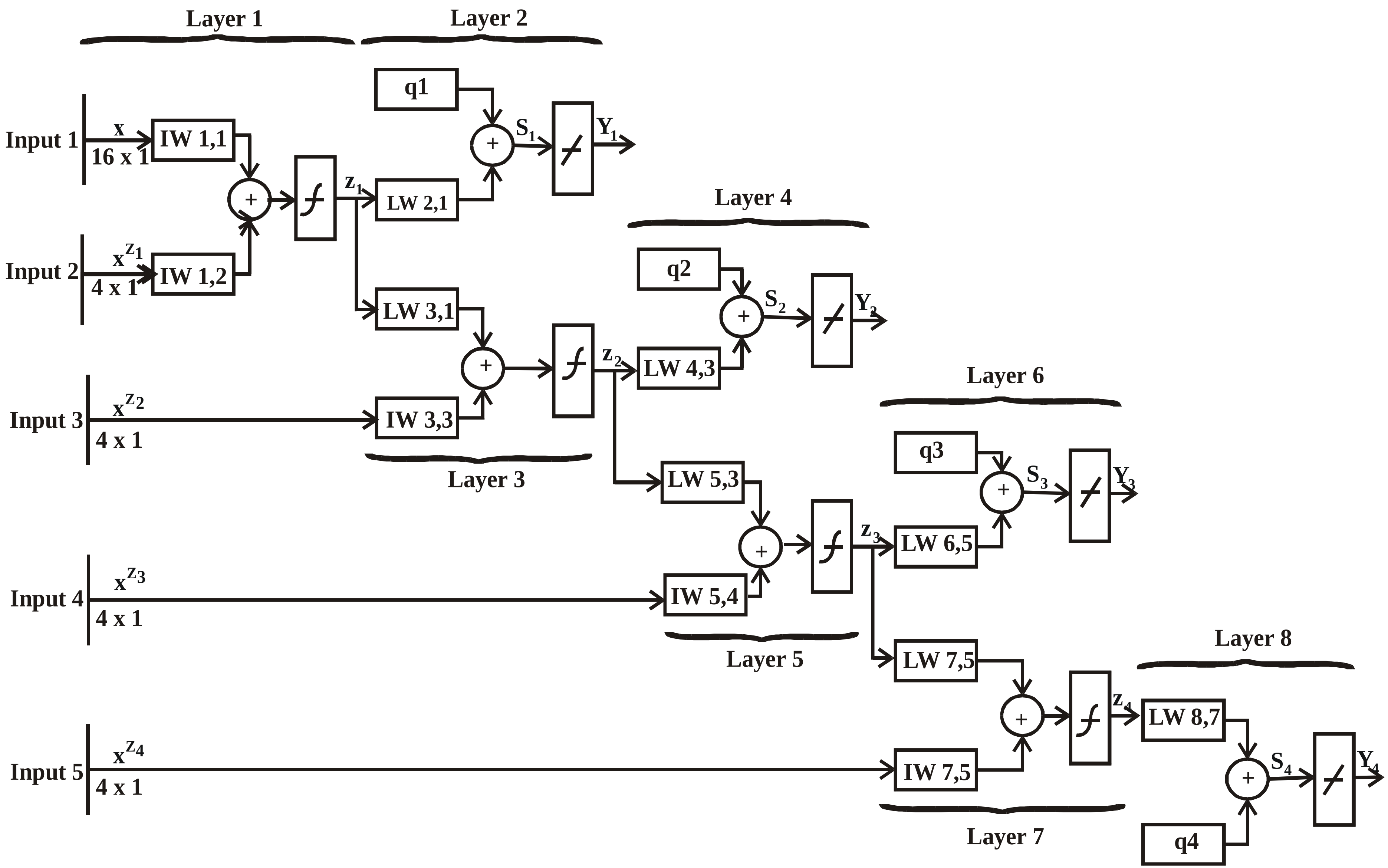

Fig. 3. Structure chart of decision maker (ANN) in Matlab 6.1.

Table 3

Investigation of ANN training algorithms by example training sample of 66 training vectors

Training algorithm	Criterion of training quality evaluation	Perfor-mance	Training time	Epochs quantity
Broyton, Fletcher, Goldfarb and Shano training algorithm (BFGS)	Roof-mean-square deviation (mse)	0,177973	28 sec.	2 epochs
Broyton, Fletcher, Goldfarb and Shano training algorithm (BFGS)	Quality composite test (msereg)	0,160309	28 sec.	2 epochs
Training algorithm CGB on base conjugate-gradient method with back propagation and restarts in modification of Pauel-Biele	Roof-mean-square deviation (mse)	0,177973	16 sec.	1 epoch
Training algorithm CGB on base conjugate-gradient method with back propagation and restarts in modification of Pauel-Biele	Quality composite test (msereg)	0,160309	16 sec.	1 epoch
Fletcher-Reevs algorithm (CGF)	Roof-mean-square deviation (mse)	0,177973	17 sec.	1 epoch
Fletcher-Reevs algorithm (CGF)	Quality composite test (msereg)	0,160309	17 sec.	1 epoch
Polak-Ribeyra algorithm (CGP)	Roof-mean-square deviation (mse)	0,177973	19 sec.	1 epoch
Polak-Ribeyra algorithm (CGP)	Quality composite test (msereg)	0,160309	19 sec.	1 epoch
Gradient escapement algorithm (GD)	Roof-mean-square deviation (mse)	0,177973	7 min. 7 sec.	5850 epochs
Gradient escapement algorithm with choice of debugging speed parameter (GDA)	Roof-mean-square deviation (mse)	0,177973	25 sec.	125 epochs
Gradient escapement algorithm with choice of debugging speed parameter (GDA)	Quality composite test (msereg)	0,160309	25 sec.	125 epochs
Gradient escapement algorithm with dis-turbance (GDM)	Roof-mean-square deviation (mse)	0,177973	7 min. 20 sec.	5775 epochs
Gradient escapement algorithm with dis-turbance (GDM)	Quality composite test (msereg)	0,160309	8 min.	6125 epochs
Gradient escapement algorithm with dis-turbance and adaptation of debugging speed parameter (GDX)	Roof-mean-square deviation (mse)	0,177973	31 sec.	202 epochs
Gradient escapement algorithm with dis-turbance and adaptation of debugging speed parameter (GDX)	Quality composite test (msereg)	0,160309	34 sec.	238 epochs
Levenberg-Marquardt algorithm (LM)	Roof-mean-square deviation (mse)	0,177973	1 min. 15 sec.	3 epochs
Levenberg-Marquardt algorithm (LM)	Quality composite test (msereg)	0,160309	1 min. 10 sec.	2 epochs
One-step algorithm of secant method (OSS)	Roof-mean-square deviation (mse)	0,177973	17 sec.	2 epochs
One-step algorithm of secant method (OSS)	Quality composite test (msereg)	0,160309	17 sec.	2 epochs
Threshold back-propagation mistake algorithm (Rprop)	Roof-mean-square deviation (mse)	0,177973	20 sec.	32 epochs
Threshold back-propagation mistake algorithm (Rprop)	Quality composite test (msereg)	0,160309	20 sec.	34 epochs
Training algorithm SCG	Roof-mean-square deviation (mse)	0,177973	17 sec.	1 epoch
Training algorithm SCG	Quality composite test (msereg)	0,160309	17 sec.	1 epoch

Table 4

Investigation of ANN training algorithms by example training sample of 497 training vectors

Training algorithm	Criterion of training quality evaluation	Perfor-mance	Training time	Epochs quantity
Broyton, Fletcher, Goldfarb and Shano training algorithm (BFGS)	Roof-mean-square deviation (mse)	0,161178	1 min. 25 sec.	2 epochs
Broyton, Fletcher, Goldfarb and Shano training algorithm (BFGS)	Quality composite test (msereg)	0,145232	1 min. 25 sec.	2 epochs
Training algorithm CGB on base conjugate-gradient method with back propagation and restarts in modification of Pauel-Biele	Roof-mean-square deviation (mse)	0,161178	1 min. 13 sec.	1 epoch
Training algorithm CGB on base conjugate-gradient method with back propagation and restarts in modification of Pauel-Biele	Quality composite test (msereg)	0,145232	1 min. 13 sec.	1 epoch
Fletcher-Reevs algorithm (CGF)	Roof-mean-square deviation (mse)	0,161178	1 min. 13 sec.	1 epoch
Fletcher-Reevs algorithm (CGF)	Quality composite test (msereg)	0,145232	1 min. 13 sec.	1 epoch
Polak-Ribeyra algorithm (CGP)	Roof-mean-square deviation (mse)	0,161178	1 min. 13 sec.	1 epoch
Polak-Ribeyra algorithm (CGP)	Quality composite test (msereg)	0,145232	1 min. 13 sec.	1 epoch
Gradient escapement algorithm (GD)	Roof-mean-square deviation (mse)	0,161178	17 min. 30 sec.	5925 epochs
Gradient escapement algorithm with choice of debugging speed parameter (GDA)	Roof-mean-square deviation (mse)	0,161178	1 min. 34 sec.	128 epochs
Gradient escapement algorithm with choice of debugging speed parameter (GDA)	Quality composite test (msereg)	0,145232	1 min. 35 sec.	130 epochs
Gradient escapement algorithm with dis-turbance (GDM)	Roof-mean-square deviation (mse)	0,161178	17 min. 30 sec.	5850 epochs
Gradient escapement algorithm with dis-turbance (GDM)	Quality composite test (msereg)	0,145232	20 min.	6850 epochs
Gradient escapement algorithm with dis-turbance and adaptation of debugging speed parameter (GDX)	Roof-mean-square deviation (mse)	0,161178	1 min. 50 sec.	202 epochs
Gradient escapement algorithm with dis-turbance and adaptation of debugging speed parameter (GDX)	Quality composite test (msereg)	0,145232	1 min. 57 sec.	238 epochs
Levenberg-Marquardt algorithm (LM)	Roof-mean-square deviation (mse)	0,161178	45 min.	3 epochs
Levenberg-Marquardt algorithm (LM)	Quality composite test (msereg)	0,145232	45 min.	2 epochs
One-step algorithm of secant method (OSS)	Roof-mean-square deviation (mse)	0,161178	1 min. 19 sec.	2 epochs
One-step algorithm of secant method (OSS)	Quality composite test (msereg)	0.145232	1 min. 17 sec.	2 epochs
Threshold back-propagation mistake algorithm (Rprop)	Roof-mean-square deviation (mse)	0,161178	1 min. 16 sec.	29 epochs
Threshold back-propagation mistake algorithm (Rprop)	Quality composite test (msereg)	0,145232	1 min. 16 sec.	32 epochs
Training algorithm SCG	Roof-mean-square deviation (mse)	0,161178	1 min. 16 sec.	1 epoch
Training algorithm SCG	Quality composite test (msereg)	0,145232	1 min. 16 sec.	1 epoch

Table 5

Investigation of ANN training algorithms by example training sample of 2250 training vectors

Training algorithm	Criterion of training quality evaluation	Perfor-mance	Training time	Epochs quantity
Broyton, Fletcher, Goldfarb and Shano training algorithm (BFGS)	Roof-mean-square deviation (mse)	0,124273	7 min.	2 epochs
Broyton, Fletcher, Goldfarb and Shano training algorithm (BFGS)	Quality composite test (msereg)	0,11209	6 min. 40 sec.	2 epochs
Training algorithm CGB on base conjugate-gradient method with back propagation and restarts in modification of Pauel-Biele	Roof-mean-square deviation (mse)	0,124273	6 min. 30 sec.	1 epoch
Training algorithm CGB on base conjugate-gradient method with back propagation and restarts in modification of Pauel-Biele	Quality composite test (msereg)	0,11209	6 min. 15 sec.	1 epoch
Fletcher-Reevs algorithm (CGF)	Roof-mean-square deviation (mse)	0,124273	6 min.	1 epoch
Fletcher-Reevs algorithm (CGF)	Quality composite test (msereg)	0,11209	6 min. 38 sec.	1 epoch
Polak-Ribeyra algorithm (CGP)	Roof-mean-square deviation (mse)	0,124273	6 min. 45 sec.	1 epoch
Polak-Ribeyra algorithm (CGP)	Quality composite test (msereg)	0,11209	6 min. 40 sec.	1 epoch
Gradient escapement algorithm (GD)	Roof-mean-square deviation (mse)	0,124273	1 hour 40 min.	5875 epochs
Gradient escapement algorithm with choice of debugging speed parameter (GDA)	Roof-mean-square deviation (mse)	0,124273	9 min.	128 epochs
Gradient escapement algorithm with choice of debugging speed parameter (GDA)	Quality composite test (msereg)	0,11209	8 min. 30 sec.	130 epochs
Gradient escapement algorithm with dis-turbance (GDM)	Roof-mean-square deviation (mse)	0,124273	1 hour 9 min.	5825 epochs
Gradient escapement algorithm with dis-turbance (GDM)	Quality composite test (msereg)	0,11209	1 hour 15 min.	6875 epochs
Gradient escapement algorithm with dis-turbance and adaptation of debugging speed parameter (GDX)	Roof-mean-square deviation (mse)	0,124273	14 min.	202 epochs
Gradient escapement algorithm with dis-turbance and adaptation of debugging speed parameter (GDX)	Quality composite test (msereg)	0,11209	11 min.	238 epochs
Levenberg-Marquardt algorithm (LM)	Roof-mean-square deviation (mse)	0,124273	3 hours	3 epochs
Levenberg-Marquardt algorithm (LM)	Quality composite test (msereg)	0,11209	3 hours	2 epochs
One-step algorithm of secant method (OSS)	Roof-mean-square deviation (mse)	0,124273	7 min. 30 sec.	2 epochs
One-step algorithm of secant method (OSS)	Quality composite test (msereg)	0,11209	8 min. 15 sec.	2 epochs
Threshold back-propagation mistake algorithm (Rprop)	Roof-mean-square deviation (mse)	0,124273	7 min. 15 sec.	31 epochs
Threshold back-propagation mistake algorithm (Rprop)	Quality composite test (msereg)	0,11209	8 min. 15 sec.	32 epochs
Training algorithm SCG	Roof-mean-square deviation (mse)	0,124273	7 min. 15 sec.	1 epoch
Training algorithm SCG	Quality composite test (msereg)	0,11209	6 min. 25 sec.	1 epoch

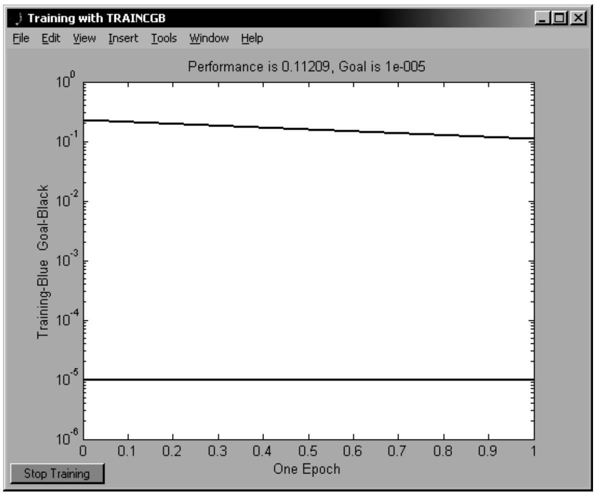

Fig. 4. Training algorithm CGB on base conjugate-gradient method with back propagation and restarts in modification of Pauel-Biele

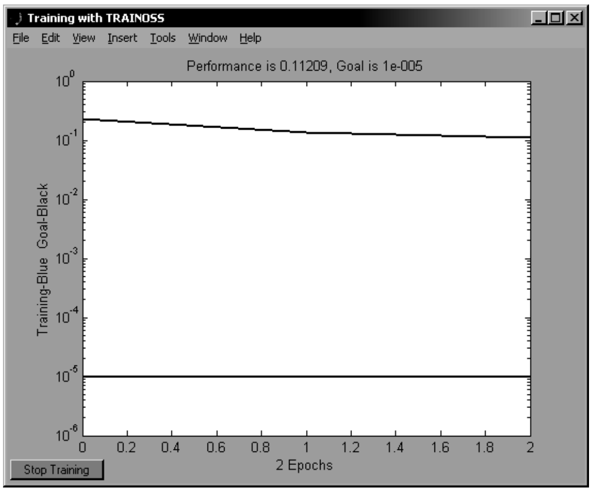

Fig. 5. One-step algorithm of secant method (OSS)

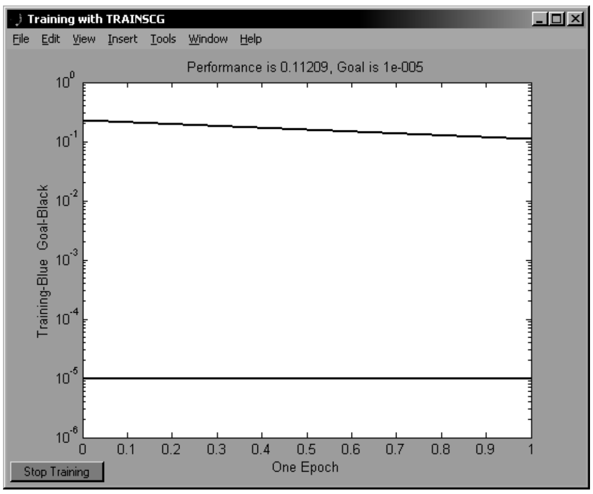

Fig. 6. Training algorithm SCG

As a result of tables 3 – 5 analysis was determined, that at time index the training algorithm CGB on base conjugate-gradient method with back propagation and restarts in modification of Pauel-Biele, Fletcher-Reevs algorithm (CGF), Polak-Ribeyra algorithm (CGP), one-step algorithm of secant method (OSS), threshold back-propagation mistake algorithm (Rprop), training algorithm SCG with using of quality composite test are the best. At index "epochs quantity" training algorithm CGB on base conjugate-gradient method with back propagation and restarts in modification of Pauel-Biele, Fletcher-Reevs algorithm (CGF), Polak-Ribeyra algorithm (CGP), training algorithm SCG, Broyton, Fletcher, Goldfarb and Shano training algorithm (BFGS), one-step algorithm of secant method (OSS), Levenberg-Marquardt algorithm (LM) with using of quality composite test are the best. Training algorithm CGB on base conjugate-gradient method with back propagation and restarts in modification of Pauel-Biele, Fletcher-Reevs algorithm (CGF), Polak-Ribeyra algorithm (CGP) are modified algorithms on base conjugate-gradient method with back propagation mistakes, therefore one of them (training algorithm CGB on base conjugate-gradient method with back propagation and restarts in modification of Pauel-Biele with using of quality composite test – fig. 4) was chosen.

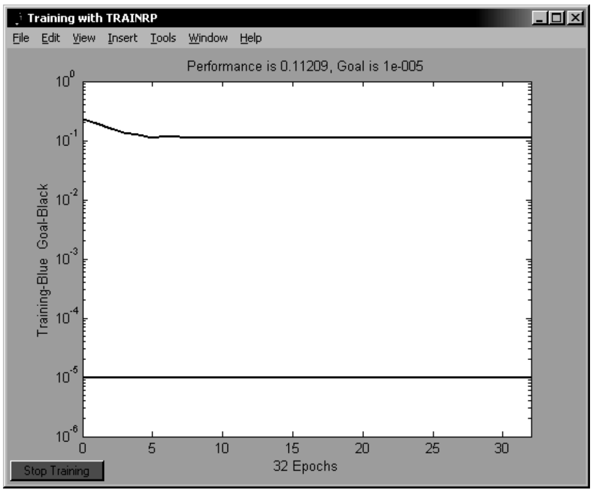

Fig. 7. Threshold back-propagation mistake algorithm (Rprop)

On figure upper curve represents speed and performance of ANN training on the adjusted algorithm. Under straight line represents goal performance.

Also one-step algorithm of secant method (fig. 5), training algorithm SCG (fig. 6) and threshold back-propagation mistake algorithm (fig. 7) with using of quality composite test were chosen.

Conclusion

In this article model of decision maker of repeated application software testing system in Matlab 6.1 was developed, its structure was described. Series of ANN training experiments using different training algorithm and criterions of training quality evaluation were realized. As a result of these experiments ANN training algorithms and criterion of training quality evaluation were specified.

As a result of investigation next was specified: training algorithm CGB on base conjugate-gradient method with back propagation and restarts in modification of Pauel-Biele, one-step algorithm of secant method, training algorithm SCG and threshold back-propagation mistake algorithm with using of quality composite test are the most effective, convenient and expedient . Performance of simulated ANN training doesn't depend on training time and iteration (epochs) quantity. It depends on criterion of training quality evaluation.

Maximum performance, which was reached in training process with training sample of 66 vectors, is 0,160309, with training sample of 497 vectors – 0,145232, with training sample of 2250 vectors – 0.11209.

In other words, with increasing training sample size 7,5 times more performance improves on 0,015077, with increase training sample size 4,5 times more performance improves on 0,033142, as a whole with increase training sample size 34 times more performance improves on 0,048219. So, after analysis of received results the conclusion was drawed: has no sense training sample power (size) to increase.

References

1. Говорущенко Т.О. Система повторного тестування програмного забезпечення // Радіоелектронні і комп'ютерні системи. – 2005. – № 4. – С. 120-126.

2. Govoruschenko T.O. Determination of Neccesity and Advisable Method(s) of Repeated Application Software Testing // Радіоелектронні і комп'ютерні системи. – 2006. – № 6. – С. 136-141.

3. Локазюк В.М., Пантелєєва (Говорущенко) Т.О. Категорійна модель процесу повторного тестування дефектів програмного забезпечення // Вісник Технологічного університету Поділля – Хмельницький: ТУП. – 2004. – Ч. 1, т. 1. – С. 53-58.

4. Lokazyuk V.M., Govoruschenko T.O. Category Model of Process of Repeated Software Testing // Proceedings of the Third IEEE Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications. – Sofia, Bulgaria,: 2005. – P. 241-245.

5. Говорущенко Т.О. Оцінка ефективності виявлення прихованих помилок у програмному забезпеченні // Вісник Хмельницького національного університету. – Хмельницький. – 2005. – Ч. 1, т. 2. – С. 190-195.

6. Локазюк В.М., Поморова О.В., Говорущенко Т.О. Імітаційна модель системи повторного тестування програмного забезпечення // Вісник Хмельницького національного університету. – Хмельницький. – 2006. – № 6. – С. 65-72.

7. Калбертсон Р., Браун К., Кобб Г. Быстрое тестирование. – М.: Вильямс, 2002. – 384 с.

Поступила в редакцию 5.02.2006

Рецензент: канд. техн. наук, доцент А.С. Савенко, Хмельницкий национальный университет, Хмельницкий.