ICIC Express Letters
Part B: Applications
Volume **12**, Number **3**, March **2021**

ICIC International ©2021 ISSN 2185-2766
pp. 301–306

GRAPHICAL INTERFACE OF GENETIC OPTIMIZATION IN NEURAL NETWORK MODELLING FOR TIME SERIES

Rukun Santoso, Budi Warsito* and Hasbi Yasin

Department of Statistics
Faculty of Sciences and Mathematics
Diponegoro University
Prof. Soedarto, SH. Street, Tembalang, Semarang 50275, Indonesia
{ rukunsantoso25; hasbiyasin17 }@gmail.com; *Corresponding author: budiwarsito1975@gmail.com

Received August 2020; accepted October 2020

Abstract. *Various gradient based optimization techniques have been developed for optimizing neural network weights. However, the stability of results in obtaining the optimal weights has become an interesting open problem. In recent developments, the utilizing of non-gradient optimization has been widely practiced. Various heuristic optimization methods have been used for obtaining the optimal weights of neural network. Simulated annealing, ant colony and particle swarm optimization are part of the popular techniques. One other technique that is also often used is genetic algorithm. The advantage of this algorithm is its generality in using various types of activation function. It is a logical operation inspired by the properties of creatures; therefore, the use of an algorithm that imitates the behavior of creatures is expected to be appropriate for obtaining the optimal weights. However, the use of genetic optimization needs a complicated computation especially in running programs related to the selection of genetic components. In this research, the using of graphical user interface (GUI) for the operation of genetic algorithm in optimizing neural network has been developed. It is done to facilitate the process of running the computing program of genetic algorithm in neural network modeling. The developed computing program has been applied to the time series data.*
Keywords: Genetic algorithm, Neural network, Time series, GUI

1. **Introduction.** Neural network modeling in time series field has been developed a lot in recent years [1,2]. This is a modeling algorithm inspired by the working system of biological nerves. A main class of neural network is Feed Forward Neural Network (FFNN). In this class, the connections between neurons in input layer, hidden layer & output layer represent the strength of relationship between each unit and are explained as weights. More complicated network will require more weights that should be obtained. Due to the nonlinearity and many more of weights, the specific optimization method is needed for developing neural network model. Gradient based optimization methods, the most widely used, often reach unstable results [3]. Meanwhile, it is not guaranteed that neural network will have a global optimum after training, instead of becoming stuck around an arbitrarily poor local minimum [4]. On the other hand, the non-gradient based optimization methods have also been developed and used in neural network model [5]. Some of them are particle swarm optimization [6], ant colony [7], adaptive moment (Adam) gradient descent as used by Wang et al. [8], bee colony [9] and genetic algorithm [10]. However, metaheuristic optimization methods as GA need many evolution operators for updating generation [11]. As the consequence, it needs more time and encounters more computational problem to reach the local optimum which is near the global optimum. It needs a simple and easy executable computing program.

DOI: 10.24507/icicelb.12.03.301

A lot of computer programs have been developed for genetic algorithm in optimizing neural network. However, preparing input data for coupled neural network models by using genetic optimization is often times a slow, tedious, and error-prone process. To save time and avoid errors, some researchers have developed GUI as a model simulation. Bailey et al. [12] have developed SWATMOD-Prep, a stand-alone Python-based GUI that facilitates preparing linkage files for a SWAT-MODFLOW model simulation. Park et al. [13] have presented QSWATMOD, a QGIS-based GUI plug-in that links a SWAT model with a MODFLOW model within a geographical information system (GIS) setting. In neural network field, GUI for modeling computation has been developed as in [14-16]. On the other hand, GUI for genetic optimizations also has been built [17,18].

The main purpose of this paper is to develop the neural network model that fits into intuitive method for estimating time series data. The input variables are considered as the lagged time of the series. As in time series modeling, the current value is depending on past values of the series. The detailed step-by-step genetic optimization on the neural network model of the expert driven estimation approach has been developed. The in-sample and out-sample predictions are made as the base of selecting the best model. The model can be further used to predict the future. A GUI by using MATLAB routine also has been developed so that the neural network model with genetic optimization can be used without any prior knowledge of neural network. The organization of this paper is summarized as follows. Section 2 describes the genetic optimization in FFNN. Section 3 deals with results and discussions of the developed GUI. Finally, Section 4 is devoted to conclusions.

2. Genetic Optimization in Neural Network.

2.1. Neural network. FFNN, the main class of neural network model, is used in this research. This processing contains three layers, i.e., input layer, hidden layer and output layer. In neural network for time series modeling, the input contains lagged time from past series. The weighted sum of the input is then sent to the hidden layer. A nonlinear activation function in the hidden layer processes the incoming signal from the input. The network then carries out the weighted sum of the signals coming out from the hidden layer. A linear activation function at the output layer then works to produce an output, which is expected to be as close as possible to the intended target. Connection weights between layers in the neural network should be estimated to get the output as expected. Network architecture as described can be seen in Figure 1.

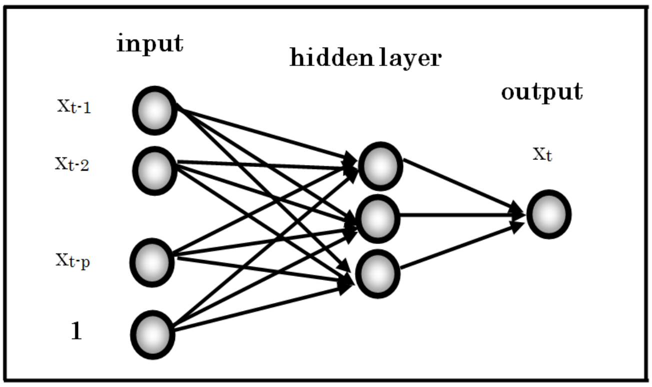

FIGURE 1. Neural network architecture for predicting time series

The mathematical model of the architecture can be seen in Equation (1).

$$x_t = f^o\left(w^b + \sum_{j=1}^{l} w_j^o f^h\left(w_j^b + \sum_{i=1}^{p} w_{ji}^h x_{t-1}\right)\right) \quad (1)$$

The symbols w^b, w_j^o, w_j^b, w_{ji}^h represent the weights from bias to output, the weights from hidden unit j to output, the weights from bias to hidden unit j and the weights from input i to hidden unit j, respectively. The numbers of hidden unit l and lags p are specified by using a certain technique. The activation function in hidden layer is symbolized by f^h and in this case the sigmoid logistic is chosen. Whereas, the linear activation function in output layer is symbolized by f^o. The weights vector $w = \left(w^b, w_j^o, w_j^b, w_{ji}^h\right)$ should be estimated by using an optimization method. In this case the genetic algorithm is used as searching method.

2.2. **Genetic optimization.** Genetic algorithm is a probabilistic search technique that resembles the principle of genetics. The basic idea of genetic algorithms is started with an initial set of points in Ω called initial population, denoted by $P(0)$. Furthermore, the objective function at points in $P(0)$ is evaluated, and a new set of points $P(1)$ is created, based on the evaluation of $P(0)$ through certain operations. Operating on a genetic algorithm aimed to create a new population with an average objective function value that is higher than the previous. Iteration procedure is performed in order to produce a population in the next generation, namely, $P(2), P(3), \ldots$, until an appropriate stopping criterion is reached. The selection process is performed by crossover and mutation operations. In every stage of the process of iteration k, the fitness $f(x(k))$ is evaluated for each member $x(k)$ of the population $P(k)$. After all fitness is evaluated, a new population $P(k+1)$ is formed by selection and evolution processes. Selection is performed by forming a mating pool set $M(k)$ from $P(k)$. The number of elements of $M(k)$ is the same as with $P(k)$, and is called the population size. The selection process ensures that individuals or chromosomes with higher fitness are preferable to be selected in crossover operation rather than the lower. After applying the crossover and mutation operations to the mating pool $M(k)$, the new population $P(k+1)$ is obtained. The procedure is then repeated iteratively until a desired number of generations.

3. **Graphical Interface Development.** In developing GUIs, the GUIDE (MATLAB's Graphical User Interface Development Environment) has been used. GUIDE stores GUIs in two files, i.e., NN-AG.fig as MATLAB Figure file and NN-AG.m as MATLAB M file. The two files are generated when the GUI is saved or run for the first time. NN-AG.fig contains the GUI figure layout and the components of the GUI. The figure can be seen in Figure 2. There are three main parts including the input, process and output. Input contains the data type part and genetic operator part. In the data type part, the Pop-Up menu is used to open and choose the data that will be analyzed. One of the chosen data can be selected from the list in this menu. The display of data type menu options is shown in Figure 3. In the genetic operator part, three objects are created: Static Text, Edit Text and Pop-Up. For the smartness, several inputs of the same type are placed in a frame using Panel menu. Static text object contains label of the input including the number of looping and the number of maximum generations. Looping is intended to count how many repetitions are done. The desired values of the input can be entered in Edit text object.

The operator of genetic algorithm consists of probability of crossover, probability of mutation and the Tolerance as stopping criteria which are made with Static text and the desired values of each component could be written in the box provided by Edit text. They are placed in a frame by using Panel menu. There are three components in genetic optimization developed in this research: Crossover, Selection and Mutation. All of the components are made by Static text and the list of each component is provided by Pop-Up menu. In Crossover component, the Single Point Crossover or Two Point Crossover could be selected whereas Roulette Wheel or Tournament could be chosen as Selection method. The two mutation techniques are also offered namely Adapt Feasible and Gaussian. The

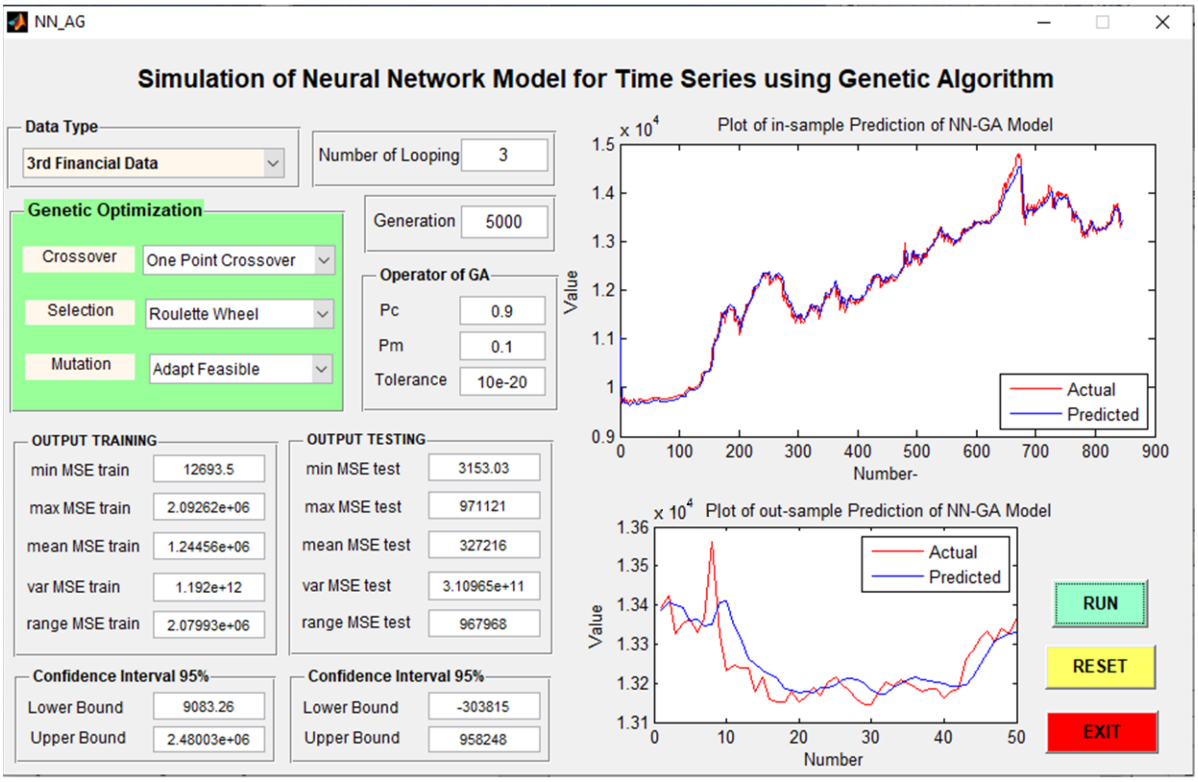

FIGURE 2. Graphical user interface of neural network – genetic algorithm

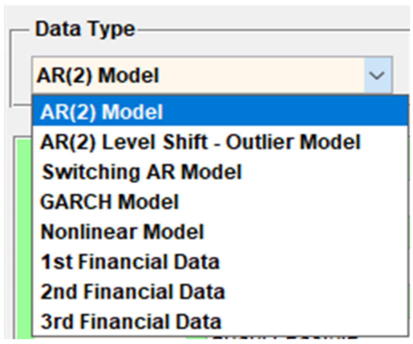

FIGURE 3. Display of data type menu

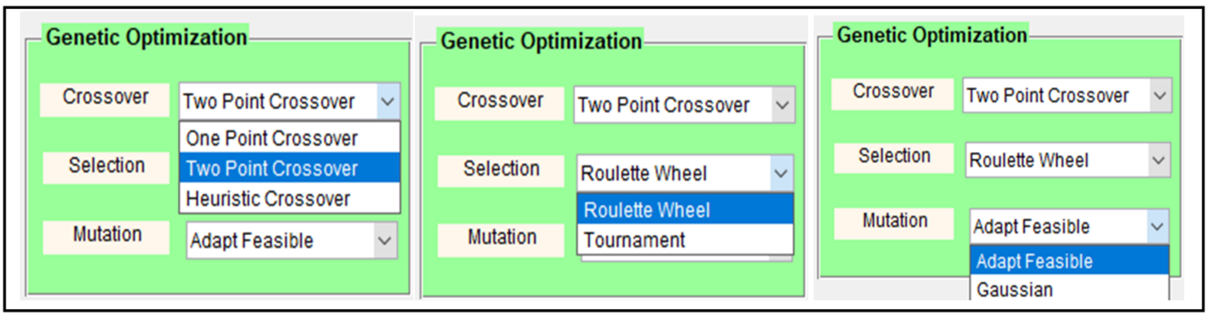

FIGURE 4. Display of operator menu on genetic algorithm

three components are combined in one Panel box. This flexibility to choose various components basically allows the user to choose the items of their own choice [19]. The detail of display of the operator menu on genetic algorithm is shown in Figure 4.

Processing of the proposed procedure is done by RUN box. It is provided by Push button. The output of the process is divided into two parts. The first part is the calculation values consisting of outputs from training data, outputs from testing data and the confidence interval from a desired looping process. In the output of training and testing data, the calculation results displayed include the minimum, maximum, mean, variance and range of the mean square error. These values indicate the goodness and stability result of each process with a desired number of looping. As in the input, each values group of this part is made in a box by using Panel menu. This is done both for training, testing and the confidence interval. The second part of the output is plotting in-sample and out-sample prediction. The first figure contains plot of actual versus prediction of training data whereas the second contains plot of actual versus prediction of testing data. The two figures are built by Axes menu. When the RUN button is clicked, all calculation will appear on the display screen together with the desired plots. On the other hand, when the RESET button is pressed all displays will be clean again. The EXIT button is used to quit from the GUIs. With these two output parts, the interface developed meets the requirements as a complete system that should have two modules [20]. The three-push button menus are built in bold and highlighted points to make this work eye catchy [21].

GUI Code file or MATLAB M file which is built together with the establishment of GUIs contains the code that controls the GUI, including the call-backs for its components. This is referred as GUI M file. Complete coding is done, such that the GUI will not run the process and display an error message when input type does not match to the certainty or the inputs are not in range. A call-back is assigned to analyze the push button. The GUI type built is well suited for modeling a time series data by using complex systems as the hybrid GA optimization and NN model. This has relevance to [22] which made a lot of simulation and also presented several learning opportunities. In case of teaching and learning of solving a set of complex formulas, this is also appropriate to [23], even this GUI is more applicable in real case instead of the theoretical one. A similar result is the GUI development for the environmental quality prediction system [24]. It has also used neural network for making predictions, and also has been executed quite well. The drawback is the unused plots so that the accuracy of predictions is not well described. This part is refined through completeness in the plot section.

4. **Conclusions.** In this paper, the using of genetic algorithm in obtaining weights of neural network for time series in the realm of computing is studied. Neural network model with genetic optimization can be applied in various fields and data type of time series, both generated data from specific distribution and real data. The using of graphical interface for executing the computational program makes it easier for user without prior knowledge of neural network. The GUI developed would prove to be efficient to be used. However, this research has a limitation about the GUI that can be run by MATLAB's GUIDE tool only. An independent software tool which does not need MATLAB or a stand-alone open source-based GUI like Python or R can be developed to run the network. The GUI can also be developed for the other heuristic optimization methods and the hybrid with gradient methods in obtaining weights of neural network.

Acknowledgments. This research is funded by Directorate of Research and Community Service, Ministry of Research and Higher Education with the scheme of Basic Research No: 257-21/UN7.P4.3/PP/2019.

REFERENCES

[1] N. R. Sari, W. F. Mahmudy and A. P. Wibawa, Backpropagation on neural network method for inflation rate forecasting in Indonesia, *Int. J. Soft Comput. Its Appl.*, 2016.
[2] Y. Hasbi, W. Budi and S. Rukun, Feed forward neural network modeling for rainfall prediction, *E3S Web of Conferences*, vol.73, no.9, 05017, 2018.

[3] X. H. Le, H. V. Ho, G. Lee and S. Jung, Application of long short-term memory (LSTM) neural network for flood forecasting, *Water*, vol.11, no.7, 2019.
[4] K. Kawaguchi and L. P. Kaelbling, Elimination of all bad local minima in deep learning, *arXiv Preprint arXiv:1901.00279*, 2019.
[5] V. K. Ojha, A. Abraham and V. Snášel, Metaheuristic design of feedforward neural networks: A review of two decades of research, *Engineering Applications of Artificial Intelligence*, vol.60, pp.97-116, 2017.
[6] B. Warsito, H. Yasin and A. Prahutama, Particle swarm optimization versus gradient based methods in optimizing neural network, *Journal of Physics: Conference Series*, vol.1217, no.1, 012101, 2019.
[7] M. Mavrovouniotis and S. Yang, Training neural networks with ant colony optimization algorithms for pattern classification, *Soft Computing*, vol.19, no.6, pp.1511-1522, 2015.
[8] Y. Wang, X. Chen and X. Ye, Prediction of network protocol data flow based on a recurrent neural network, *International Journal of Innovative Computing, Information and Control*, vol.15, no.4, pp.1381-1395, 2019.
[9] J. A. Bullinaria and K. Al Yahya, Artificial bee colony training of neural networks, *Nature Inspired Cooperative Strategies for Optimization (NICSO2013)*, pp.191-201, 2014.
[10] M. A. J. Idrissi, H. Ramchoun, Y. Ghanouand and M. Ettaouil, Genetic algorithm for neural network architecture optimization, *2016 the 3rd International Conference on Logistics Operations Management (GOL)*, pp.1-4, 2016.
[11] M. Čepinšek, S. H. Liuand and M. Mernik, Exploration and exploitation in evolutionary algorithms: A survey, *ACM Computing Surveys (CSUR)*, vol.45, no.3, 2013.
[12] R. Bailey, H. Rathjens, K. Bieger, I. Chaubey and J. Arnold, SWATMOD-Prep: Graphical user interface for preparing coupled SWAT-MODFLOW simulations, *Journal of the American Water Resources Association (JAWRA)*, vol.53, no.2, pp.400-410, 2017.
[13] S. Park, A. Nielsen, R. T. Bailey, D. Trolle and K. Bieger, A QGIS-based graphical user interface for application and evaluation of SWAT-MODFLOW models, *Environmental Modelling & Software*, vol.111, pp.493-497, 2019.
[14] S. Klemm, A. Scherzinger, D. Drees and X. Jiang, Barista – A graphical tool for designing and training deep neural networks, *arXiv Preprint arXiv:1802.04626*, 2018.
[15] A. H. Makwana, D. H. Patel and A. A. Mehta, Artificial neural network based graphical user interface for estimation of fabrication time in rig construction project, *International Journal for Scientific Research & Development*, vol.1, no.4, pp.1011-1014, 2013.
[16] H. Nakahara, H. Yonekawa, T. Fujii, M. Shimoda and S. Sato, A demonstration of the GUINNESS: A GUI based neural NEtwork SyntheSizer for an FPGA, *2017 the 27th International Conference on Field Programmable Logic and Applications (FPL)*, p.1, 2017.
[17] J. C. Quiroz, S. J. Louis, A. Shankar and S. M. Dascalu, Interactive genetic algorithms for user interface design, *2007 IEEE Congress on Evolutionary Computation*, pp.1366-1373, 2007.
[18] P. Salem, User interface optimization using genetic programming with an application to landing pages, *Proc. of the ACM on Human-Computer Interaction*, vol.1, no.1, pp.1-17, 2017.
[19] S. P. Reiss, Y. Miao and Q. Xin, Seeking the user interface, *Automated Software Engineering*, vol.25, no.1, pp.157-193, 2018.
[20] M. Sharma and G. Sahu, MATLAB GUI based signal processing for various function, *International Journal of Science and Research (IJSR)*, vol.5, no.10, pp.1815-1821, 2016.
[21] W. Iftikhar, M. S. A. Malik, S. Tariq, M. S. Sultan, J. Ahmad and F. Shareef, User interface design issues in HCI, *International Journal of Computer Science and Network Security*, vol.18, no.8, pp.153-157, 2018.
[22] J. T. Scheele, P. R. Ohmann and A. R. Green, A Matlab user-interface tool for modelling herds, *European Journal of Physics*, vol.39, no.6, 064004, 2018.
[23] S. Syaharuddin, H. R. Negara, V. Mandailina and L. Sucipto, Calculus problem solution and simulation using GUI of Matlab, *International Journal of Scientific & Technology Research*, vol.6, no.9, pp.275-279, 2017.
[24] N. A. Rahim and Z. Ahmad, Graphical user interface application in Matlab environment for water and air quality process monitoring, *Chemical Engineering Transactions*, vol.56, pp.97-102, 2017.