Performance One-step secant Training Method for Forecasting Cases

N L W S R Ginantra¹*, Gita Widi Bhawika², Achmad Daengs GS³, Pawer Darasa Panjaitan⁴, Mohammad Aryo Arifin⁵, Anjar Want⁶, Muhammad Amin⁷, Harly Okprana⁸, Abdullah Syaffi⁹, Umar Anwar⁹

¹ STMIK STIKOM Indonesia, Denpasar, Bali, Indonesia
² Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia
³ Universitas 45 Surabaya, Surabaya, Indonesia
⁴ Universitas Simalungun, Pematangsiantar, Indonesia
⁵ Universitas PGRI Palembang, Palembang, Indonesia
⁶ STIKOM Tunas Bangsa, Medan, Indonesia
⁷ Universitas Islam Indragiri, Riau, Indonesia
⁸ Universitas Pembangunan Panca Budi, Medan, Indonesia
⁹ Politeknik Ilmu Pemasyarakatan, Indonesia

*wiwik@stiki-indonesia.ac.id

Abstract. The training function used in the ANN method, especially backpropagation, can produce different forecasting accuracy, depending on the method parameters given and the data to be predicted. This paper aims to analyze the ability and performance of one of the training functions in the backpropagation algorithm, namely One-step secant, which can later be used or used as a reference in the case of data forecasting. This method is able to update the values of bias and weights according to the one-step secant method. The analysis process uses a dataset of Foreign Exchange Reserves (US $ Million) in Indonesia 2011-2020. Based on this dataset, the dataset will be divided into two parts. The training data uses the 2011-2014 and 2015 dataset as the training data target. Meanwhile, the test data used 2016-2019 and 2020 as the target test data. The analysis process uses 5 experimental architectures, namely 4-5-1, 4-7-1, 4-9-1, 4-11-1 and 4-13-1. The results of the research based on the analysis obtained the best network architecture 4-11-1 with an MSE Training value of 0.00000012, MSE testing/performance of 0.00115144 (the smallest compared to other architectures) and Epoch 343 Iterations.

1. Introducing

Many training functions and activation functions in the Artificial Neural Network (ANN) algorithm, especially the Backpropagation algorithm, are good for prediction cases and other computational problems [1]–[5]. There are many types of ANN algorithms, including: perceptron, backpropagation, Learning vector quantization (LVQ), probablistic neural networks, hopfield, radial base network [6]–[12], all of which have their own characteristics. The ANN algorithm discussed in this paper uses the One-step secant method training function, which is one of the developments of training functions in the backpropagation algorithm. This method is able to work systematically by training multiplayer networks using mathematical science based on developed network architecture.
architectures. The training function of the One-step secant backpropagation method is able to update the weight and bias values according to the one-step secant method [13]. Data that has been trained properly will provide an appropriate output if it is given input that is different from the architectural pattern used in training. This generalization property makes training more efficient because it does not require a very long time like conventional backpropagation algorithms.

This paper will discuss and analyze the training function of the ANN algorithm using the One-step secant (OSS) backpropagation method to solve the forecast data case, in order to obtain the best architecture and performance that can be used as a reference for obtaining forecasting results. Currently, forecasting methods that use computational, statistical and experiential data are very interesting to research, especially when using soft computing and artificial intelligence tools such as artificial neural networks (ANN) which are very well known [14], one of them is the One-step method of secant backpropagation. One-step secant technique is capable of training any network as long as its input, weight and transfer functions have derived functions [15], [16]. The One-step secant (Trainoss) method provides maximum deviation between all training functions [17]. One-step-secant algorithm (OSS) is an approach to connect the approaching gap between quasi-Newton and the conjugate gradient algorithm [18]. The complete Hessian matrix is not preserved for this approach but is assumed for each iteration. The OSS advantage is in calculating the direction of the new search, not calculating the inverse matrix. However, OSS requires more compute and storage processing per iteration than the conjugate gradient method [19][20].

The One-step secant (OSS) method has been widely used to solve many complex problems. Q H Nguyen, et al (2020) Developed the ML architecture based on the Feedforward Neural Network (FNN) as a substitute for the new hybrid method and the development of the One-step secant (OSS) method to predict the capacity of the Concrete Filled Steel Tube (CFST) column, while the OSS was used to optimize the weight and bias of the FNN to develop a hybrid architecture. (FNN-OSS). The result is that FNN-OSS is a strong and effective algorithm for predicting the load-bearing capacity of CFST [21]. Solikhun, et al (2020) Conducting research to predict school enrollment rates in Indonesia through optimization of standard backpropagation neural networks with One Step Secant (OSS). This study uses 4 architectural architectures (5-4-1, 5-8-1, 5-16-1 and 5-32-1). Architecture 5-16-1 is the best. Based on the research analysis, the accuracy is 96.97% for One Step Secant (OSS) and 100% on standard backpropagation. So the standard backpropagation algorithm is superior in terms of accuracy but loses in terms of iteration and speed [22]. Furthermore, the research conducted by I D Uwanaauka, and P Akpinar (2020) discusses the efficiency of predicting the carbonation depth of targeted concrete using ANN with the optimization of the One-step Secant method, which is an alternative to the conventional Levenberg-Marquardt used. The network training in this paper combines 10 unequal hidden neurons and 11 data distribution ratios. For the prediction of concrete carbonation, the value of the relation coefficient (R) is 0.99 with a variation of the percentage of 30-55%, which means that the R-value increases significantly than the observed 60-80%. Besides, based on the observation that the hidden neuron variation between 5-25 results in a less significant change in predictive accuracy, both the R-value, MSE for the percentage of training data between 60-80% [23].

Based on these previous studies, this paper proposes a training function with the One-step secant backpropagation method combined with the binary sigmoid activation function (logsig) and the linear function (purelin) to perform data forecasting. This method is analyzed by training and testing the time-series data on the Position of Foreign Exchange Reserves in Indonesia, to find the best architecture and performance.

2. **Methodology**

2.1. *Experiment Dataset*

The experimental dataset used in this paper is data on Foreign Exchange Reserves (US $ million) in Indonesia for 2011-2020 which consists of Monetary Gold, Special Drawing Rights (SDRs), Reserve Position in the Fund (RPF), Other Foreign Reserves, Money Foreign Paper (MFP) and
Deposits, Securities and other claims. Data obtained from the Indonesian Central Bureau of Statistics. This data is quantitative data.

Table 1. Foreign Exchange Position (US $ Million)

| Foreign Exchange Reserves | 2011      | 2012      | 2013 ... | 2018   | 2019   | 2020 |
|----------------------------|-----------|-----------|-----------|--------|--------|------|
| Monetary Gold              | 3593      | 3935      | 3023      | 3229.64| 3843.88| 4758 |
| Special Drawing Rights (SDRs) | 2696     | 2715      | 2712      | 1552.90| 1541.95| 1605 |
| Reserve Position in the Fund (RPF) | 223      | 224       | 224       | 1095.83| 1090.05| 1135 |
| Other Foreign Reserves     | 10361     | 10590     | 93428     | 114775.90 | 111748.33 | 117324 |
| Money Foreign Paper (MFP) and Deposits | 12585   | 22044     | 19204     | 12548.67 | 10326.10 | 10385 |
| Securities                 | 90795     | 83299     | 73669     | 101655.78 | 111748.33 | 117324 |
| Other claims               | 231       | 564       | 555       | 571    | 633    | 689  |

Source: Central Bureau of Statistics [24]

2.2. Research Stages

Stages of research carried out to forecast the level of inflation growth in Indonesia based on expenditure groups include:

a. Collect the Research dataset to be used.

b. Preprocessing. The knowledge is then normalized using the following equation [25]–[29]:

\[ x' = \frac{0.8(x-a)}{b-a} + 0.1 \]  

Formula description: \( x' \): is the normalization result, \( x \): is the normalized info, \( a \): is the lowest value, and \( b \): is the highest value.

Then the data is split into two sections, namely training and testing.

c. Determine the concept of the network architecture to be used for the training and testing process.

d. Analyze the architectural architecture used.

e. Choose the best architectural architecture and performance.

3. Results and Discussion

3.1. Normalizing data

The Foreign Exchange Reserves dataset presented in table 1 must first be normalized using the equation formula (1).

Table 2. Normalization of Indonesia's Foreign Exchange Reserves

| Foreign Exchange Reserves | 2011 (X1) | 2012 (X2) | 2013 (X3) | 2014 (X4) | 2015 (Y1) | 2016 (X5) | 2017 (X6) | 2018 (X7) | 2019 (X8) | 2020 (Y2) |
|---------------------------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| 1                         | 0.12562   | 0.12821   | 0.12132   | 0.12135   | 0.11858   | 0.11443   | 0.11736   | 0.11664   | 0.12048   | 0.12620   |
| 2                         | 0.11885   | 0.11899   | 0.11897   | 0.11775   | 0.11693   | 0.10581   | 0.10637   | 0.10615   | 0.10608   | 0.10647   |
| 3                         | 0.10016   | 0.10017   | 0.10017   | 0.10007   | 0.10000   | 0.10304   | 0.10343   | 0.10128   | 0.10325   | 0.10353   |
| 4                         | 0.88139   | 0.89874   | 0.80445   | 0.90000   | 0.85884   | 0.79068   | 0.87337   | 0.81475   | 0.86439   | 0.90000   |
| 5                         | 0.19357   | 0.26505   | 0.24359   | 0.24565   | 0.18980   | 0.16770   | 0.15347   | 0.17496   | 0.16105   | 0.16142   |
| 6                         | 0.78455   | 0.72791   | 0.65514   | 0.74852   | 0.76333   | 0.71572   | 0.81271   | 0.73263   | 0.79580   | 0.83069   |
| 7                         | 0.10022   | 0.10274   | 0.10267   | 0.10277   | 0.10266   | 0.10012   | 0.10005   | 0.10000   | 0.10039   | 0.10074   |

After the data is normalized, the next step is to divide the data into 2 groups. The first group is the training data and the second group is the testing data. For training data using data from 2011 (X1) - 2014 (X4) with a target of 2015 (Y1). While the test data was taken from 2016 (X5) - 2019 (X8) with a target of 2020 (Y2). To help data analysis, the tools used Matlab 2011b and Microsoft Excel with the architecture to be analyzed by One-step secant backpropagation 4-5-1, 4-7-1, 4-9-1, 4-11-1 and 4-13-
1. In this paper, the activation functions used are binary sigmoid (logsig) and linear function (purelin), while the supporting parameters are according to the default parameters of the One-step secant technique in Matlab. The program code can be seen in figure 1.

```matlab
% Entering Training Data
% Entering Input Data
p=[0.12362 0.11855 0.10016 0.85139 0.19357 0.78455 0.10022;
0.12521 0.11899 0.10017 0.89984 0.26505 0.72971 0.10274;
0.12132 0.11897 0.10017 0.80445 0.24359 0.65514 0.10267;
0.12135 0.11775 0.10007 0.90000 0.24565 0.74652 0.10277]
% Entering Target Output Data
t=[0.11858 0.11693 0.10000 0.85884 0.18980 0.76333 0.10266]
% Creating a Multi Layer Neural Network (5,7,9,11,13)
net = newff(minmax(p),[5,1],{'logsig','purelin'},'trainoss');
% Generating weight and bias
net.IW{1,1}
net.LW{2,1}
net.b{1}
net.b{2}
% One-step secant (trainoss) default parameter value
net.trainParam.epochs = 1000;
net.trainParam.show = 25;
net.trainParam.showCommandLine = 0;
net.trainParam.showWindow = 1;
net.trainParam.goal = 0;
net.trainParam.time = inf;
net.trainParam.min_grad = 1e-6;
net.trainParam.max_fail = 5;
net.trainParam.searchFcn = 'srchcha';
% Conducting Training
net = train(net,p,t)
% See the results when the performance is found
[s,F,E,e,perf] = sim(net,p,[],[],[],)
% Entering Input Data (Testing)
p1=[0.11493 0.10581 0.10304 0.79068 0.16770 0.71572 0.10012;
0.11736 0.10637 0.10363 0.87337 0.15347 0.81271 0.10005;
0.11664 0.10615 0.10328 0.81475 0.17496 0.73263 0.10000;
0.12045 0.10605 0.10325 0.86439 0.16105 0.79580 0.10039]
% Entering Output Data (Testing)
t1=[0.12620 0.10647 0.10353 0.90000 0.16142 0.83069 0.10074]
% Simulations using Test data based on the results of the training
[s,F,E,e,perf] = sim(net,p1,[],[],[],t1)
```

**Figure 1. Matlab Program Code**

3.2. Network architecture

In the previous explanation, it has been stated that there is five network architecture used in this study, namely 4-5-1, 4-7-1, 4-9-1, 4-11-1 and 4-13-1. From these five architectures, we will describe only 4-11-1 architecture which is the best architecture among the four other architectures. In the following, figures and tables of the results of training and testing with architecture 4-11-1 will be presented.
Figure 2. Results of Training Architecture 4-11-1

Figure 2 is the result of training with Matlab for architecture 4-11-1. Training from that architecture resulted in an Epoch of 343 iterations.

Table 3. Training Architecture 4-11-1

| No | X1  | X2  | X3  | X4  | Target (Y1) | Epoch 343 | Actual     | Error   | SSE    | Performance |
|----|-----|-----|-----|-----|-------------|-----------|------------|---------|--------|-------------|
| 1  | 0,12562 | 0,12821 | 0,12132 | 0,12135 | 0,11858     | 0,11850   | 0,00008    | 0,00000001 |
| 2  | 0,11885 | 0,11899 | 0,11897 | 0,11775 | 0,11693     | 0,11700   | 0,00007    | 0,00000001 |
| 3  | 0,10016 | 0,10017 | 0,10017 | 0,10007 | 0,10000     | 0,10060   | 0,000060   | 0,00000036 |
| 4  | 0,88139 | 0,89874 | 0,80445 | 0,90000 | 0,85884     | 0,85880   | 0,00004    | 0,00000000 | 0,00000012 |
| 5  | 0,19357 | 0,26505 | 0,24559 | 0,25465 | 0,18980     | 0,18980   | 0,0000000  | 0,00000000 |
| 6  | 0,78455 | 0,72791 | 0,65514 | 0,74852 | 0,76333     | 0,76330   | 0,0000000  | 0,00000000 |
| 7  | 0,10022 | 0,10274 | 0,10267 | 0,10277 | 0,10266     | 0,10200   | 0,000066   | 0,00000044 |

Total SSE 0,00000081
MSE 0,00000012

Table 4. Testing Architecture 4-11-1

| No | X5  | X6  | X7  | X8  | Target (Y2) | Epoch 1 | Actual     | Error   | SSE    | Performance |
|----|-----|-----|-----|-----|-------------|---------|------------|---------|--------|-------------|
| 1  | 0,11443 | 0,11736 | 0,11664 | 0,12048 | 0,12620     | 0,11260  | 0,01360    | 0,00018507 |
| 2  | 0,10581 | 0,10637 | 0,10615 | 0,10608 | 0,10647     | 0,10540  | 0,00107    | 0,00000115 |
| 3  | 0,10304 | 0,10343 | 0,10328 | 0,10325 | 0,10353     | 0,10310  | 0,00043    | 0,00000018 |
| 4  | 0,79068 | 0,87337 | 0,81475 | 0,86439 | 0,90000     | 0,82280  | 0,07720    | 0,00595984 | 0,00115144 |
| 5  | 0,16770 | 0,15347 | 0,17496 | 0,16105 | 0,16142     | 0,17050  | -0,00908   | 0,00008244 |
| 6  | 0,71572 | 0,81271 | 0,73263 | 0,79580 | 0,83069     | 0,78790  | 0,04279    | 0,00183131 |
| 7  | 0,10012 | 0,10005 | 0,10000 | 0,10039 | 0,10074     | 0,10040  | 0,00034    | 0,00000011 |

Total SSE 0,00806011
3.3. Best Architecture and Performance Selection

The following is a comparison table between architectures 4-5-1, 4-7-1, 4-9-1, 4-11-1 and 4-13-1.

| No | X5 | X6 | X7 | X8 | Target (Y2) | Epoch 1 | Actual Error | SSE | Performance |
|----|----|----|----|----|-------------|---------|--------------|-----|-------------|
|    |    |    |    |    |             |         |              |     |             |
|    |    |    |    |    |             |         |              |     |             |
|    |    |    |    |    |             |         |              |     |             |
|    |    |    |    |    |             |         |              |     |             |

Table 5. Network Architecture Comparison

Figure 3. Graph of Comparison of Epoch Value and MSE Testing from Network Architecture

Based on the comparison of five architectures, architecture 4-11-1 is the best architecture with an Epoch of 343 iterations. Actually, the Epoch is quite big and not the best compared to the other four architectures. However, the MSE training / Performance value is the smallest compared to the other four architectures, which is 0.00115144. That is the reason this architecture was chosen as the best architecture because the smaller the MSE value, the better the results when used in forecasting cases.

4. Conclusion

One-step secant (OSS) backpropagation method can be used to solve forecasting problems. This is because, based on the results of the analysis, the error rate is quite low, and the actual results are close to the desired target data. Based on the comparison of the 5 network architectures used (4-5-1, 4-7-1, 4-9-1, 4-11-1 and 4-13-1), the 4-11-1 architecture is the best because the value of MSE testing and performance is better than other architectures.

References

[1] H. K. Ghritlahre and R. K. Prasad, “Prediction of Thermal Performance of Unidirectional Flow Porous Bed Solar Air Heater with Optimal Training Function Using Artificial Neural Network,” Energy Procedia, vol. 109, pp. 369–376, 2017.

[2] E. Siregar, H. Mawengkang, E. B. Nababan, and A. Wanto, “Analysis of Backpropagation Method with Sigmoid Bipolar and Linear Function in Prediction of Population Growth,” J.
1. M. Tyrtaiou, A. Papaleonidas, A. Elenas, and L. Iliadis, “Accomplished Reliability Level for Seismic Structural Damage Prediction Using Artificial Neural Networks,” *Proc. 21st EANN (Engineering Appl. Neural Networks)* 2020 Conf. EANN 2020. Proc. Int. Neural Networks Soc., vol. 2, pp. 85–98, 2020.

2. B. Febradii, Z. Zamzami, Y. Yunefri, and A. Wanto, “Bipolar function in backpropagation algorithm in predicting Indonesia’s coal exports by major destination countries,” *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 420, no. 1, p. 012087, 2018.

3. N. Nasution, A. Zamsuri, L. Lisnawita, and A. Wanto, “Polak-Ribiere updates analysis with binary and linear function in determining coffee exports in Indonesia,” *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 420, no. 1, pp. 1–9, 2018.

4. A. Sagheer, M. Zidan, and M. M. Abdelsamea, “A Novel Autonomous Perceptron Model for Pattern Classification Applications,” *Entropy*, vol. 21, no. 8, pp. 1–24, 2019.

5. T. P. Lillicrap, A. Santoro, L. Marris, C. J. Akerman, and G. Hinton, “Backpropagation and the brain,” *Nat. Rev. Neurosci.*, vol. 21, no. 6, pp. 335–346, 2020.

6. Y. Y. Shen, Y. M. Zhang, X. Y. Zhang, and C. L. Liu, “Online semi-supervised learning with learning vector quantization,” *Neurocomputing*, vol. 399, pp. 467–478, 2020.

7. S. Earp and A. Curtis, “Probabilistic neural network-based 2D travel-time tomography,” *Neural Comput. Appl.*, vol. 32, no. 22, pp. 17077–17095, 2020.

8. X. Xu and N. Gupta, “Application of radial basis neural network to transform viscoelastic to elastic properties for materials with multiple thermal transitions,” *J. Mater. Sci.*, vol. 54, no. 11, pp. 8401–8413, 2019.

9. W. Rahman, P. T. Nguyen, M. Rusliyadi, E. Laxmi Lydia, and K. Shankar, “Network monitoring tools and techniques uses in the network traffic management system,” *Int. J. Recent Technol. Eng.*, vol. 8, no. 2 Special Issue 11, pp. 4182–4188, Sep. 2019.

10. F. A. Ruslan, A. M. Samad, and R. Adnan, “4 Hours NNARX flood prediction model using ‘traingd’ and ‘trainoss’ training function: A comparative study,” 2018 IEEE 14th Int. Colloq. Signal Process. its Appl. CSPA 2018, pp. 77–81, 2018.

11. M. M. Shora, H. Ghassemi, and H. Nowruzzi, “Using computational fluid dynamic and artificial neural networks to predict the performance and cavitation volume of a propeller under different geometrical and physical characteristics,” *J. Mar. Eng. Technol.*, vol. 17, no. 2, pp. 59–84, 2018.

12. Mathworks, “Trainoss,” *One-step secant backpropagation*.

13. E. Susanto, Y. Novitasari, W. Rahman, and A. P. O. Amane, “Designing Software to Introduce the Musical Instruments,” in *Journal of Physics: Conference Series*, 2019, vol. 1364, no. 1.

14. R. P. Singh et al., “Computational studies for the effective electrical conductivity of copper powder filled LDPE/LLDPE composites,” *Indian J. Pure Appl. Phys.*, vol. 58, no. 6, pp. 486–493, 2020.

15. A. Perera, H. Azamathulla, and U. Rathnayake, “Comparison of different artificial neural network ( ANN ) training algorithms to predict the atmospheric temperature in Tabuk, Saudi Arabia,” *J. MAUSAM*, vol. 71, no. 2, pp. 233–244, 2020.

16. R. Constantinescu, V. Lazarescu, and R. Tahboub, “Geometrical Form Recognition Using ‘One-Step-Secant’ Algorithm In Case of Neural Network,” *UPB Sci. Bull. Ser. C Electr. Eng.*, vol. 70, no. 2, pp. 15–28, 2008.

17. D. Upadhyay, “Classification of EEG Signals under Different Mental Tasks Using Wavelet Transform and Neural Network with One Step Secant Algorithm,” *Int. J. Sci. Eng. …*, vol. 2, no. 4, pp. 256–259, 2013.

18. Q. H. Nguyen et al., “A Novel Hybrid Model Based on a Feedforward Neural Network and One Step Secant Algorithm for Prediction of Load-Bearing Capacity of Rectangular Concrete-
[22] Solikhun, M. Wahyudi, M. Safii, and M. Zarlis, “Backpropagation Network Optimization Using One Step Secant (OSS) Algorithm,” *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 769, no. 1, p. 012037, 2020.

[23] I. D. Uwanuakwa and P. Akpinar, “Investigations on the Influence of Variations in Hidden Neurons and Training Data Percentage on the Efficiency of Concrete Carbonation Depth Prediction with ANN,” *Adv. Intell. Syst. Comput.*, vol. 1095, pp. 958–965, 2020.

[24] BPS, “Posisi Cadangan Devisa (Juta US$),” 2021.

[25] A. Wanto and J. T. Hardinata, “Estimations of Indonesian poor people as poverty reduction efforts facing industrial revolution 4.0,” *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 725, no. 1, pp. 1–8, 2020.

[26] A. Wanto et al., “Forecasting the Export and Import Volume of Crude Oil, Oil Products and Gas Using ANN,” *J. Phys. Conf. Ser.*, vol. 1255, no. 1, pp. 1–6, 2019.

[27] A. Wanto et al., “Analysis of the Backpropagation Algorithm in Viewing Import Value Development Levels Based on Main Country of Origin,” *J. Phys. Conf. Ser.*, vol. 1255, no. 1, pp. 1–6, 2019.

[28] G. W. Bhawika et al., “Implementation of ANN for Predicting the Percentage of Illiteracy in Indonesia by Age Group,” *J. Phys. Conf. Ser.*, vol. 1255, no. 1, pp. 1–6, 2019.

[29] N. L. W. S. R. Ginantra, M. A. Hanafiah, A. Wanto, R. Winanajaya, and H. Okprana, “Utilization of the Batch Training Method for Predicting Natural Disasters and Their Impacts,” *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 1071, no. 012022, pp. 1–7, 2021.