Analysis of Accuracy and Epoch on Back-propagation BFGS Quasi-Newton

Herlan Silaban¹, Muhammad Zarlis², Sawaluddin³

¹,²,³Department of Computer Sciences, Faculty of Computer Sciences and Information Technology, Universitas Sumatera Utara, Medan, Indonesia. 21055

E-mail : herlan76ban@gmail.com, m.zarlis@yahoo.com, sawal@usu.ac.id

Abstract. Back-propagation is one of the learning algorithms on artificial neural networks that have been widely used to solve various problems, such as pattern recognition, prediction and classification. The Back-propagation architecture will affect the outcome of learning processed. BFGS Quasi-Newton is one of the functions that can be used to change the weight of back-propagation. This research tested some back-propagation architectures using classical back-propagation and back-propagation with BFGS. There are 7 architectures that have been tested on glass dataset with various numbers of neurons, 6 architectures with 1 hidden layer and 1 architecture with 2 hidden layers. BP with BFGS improves the convergence of the learning process. The average improvement convergence is 98.34%. BP with BFGS is more optimal on architectures with smaller number of neurons with decreased epoch number is 94.37% with the increase of accuracy about 0.5%.

1. Introduction

Back-propagation is one of the training algorithms on artificial neural networks. The network architecture on back-propagation is multilayer, it makes the output of the model more accurate. Architecture will greatly affect the time of training and the accuracy of network training results. If the artificial neural network more complex, it will take a long time of training time [1]. Suciati [2] classify batik by using back-propagation, and the number of epoch is highest for about 10000. Back-propagation optimization has been done to improve the performance of artificial neural network back-propagation either to decrease the convergent level of training or even improve the accuracy of back-propagation. Quasi-Newton BFGS is one of the optimization methods that can be used in the optimization of back-propagation training by making changes in the weight contained in the network [3]. This research uses Glass Identification dataset from UCI Machine Learning repository [4]. Glass dataset is a classification glass based on the oxide content of the glass to identify the process of making the glass whether using the float method or not. The glass type that exist on the dataset are : Float Processed (Building_windows_float_processed; 70 rows & Vehicle_windows_float_processed;17 rows) and Non Float Processed (Building_windows_non_float_processed; 76 rows, Vehicle_windows_non_float_processed, Containers; 13 rows, Tableware; 9 rows & Headlamps; 29 rows). The purpose of this research is to analyze the number of epoch and accuracy on backpropagation network training with BFGS Quasi-Newton on some network architectures so that will be find the suitable architecture for classification problem.

2. Backpropagation

Back-propagation is one of the most popular training on artificial neural networks for predictive and pattern recognition cases. Back-propagation is widely used because of its multilayer network so that the resulting output is more accurate. Back-propagation was first introduced by Rumelhart and McClelland in 1986. BP was formed by generalizing the training rules in the network model by adding a hidden layer. One of the BP networks that will be used in this research is a network with architecture...
With 1 input layer (X) consisting of 9 input neurons, 1 hidden layer (Z) with 7 neurons and 1 output layer (Y) with 2 output neurons float processed (Y1) or non float processed (Y2).

Base on the study, the number of input neurons is 9, the output neuron is 2 and the number of neurons in the hidden layer are using the formula [5].

\[
\#1\text{st hidden layer} = \sqrt{(m + 2)N} + 2 \frac{\sqrt{N}}{m+2}
\]

\[
\#2\text{nd hidden layer} = m \frac{\sqrt{N}}{m+2}
\]

N is the number of datasets whereas m is the number of outputs. Number of datasets used N = 214 and output m = 2.

In this study, the determination of the number of neurons also concerns the rule-of-thumb number of neurons [6].

- The number of hidden neurons should be between the size of the input layer and the size of the output layer.
- The number of hidden neurons should be \(2/3\) the size of the input layer, plus the size of the output layer.
- The number of hidden neurons should be less than twice the size of the input layer.

The BP architectures that will be used in this research can be seen on Table 1.

| Architecture | Description |
|--------------|-------------|
| 9-59-2       | X = 9, Z = 59 & Y = 2 |
| 9-43-2       | X = 9, Z = 43 & Y = 2 |
| 9-44-2       | X = 9, Z = 44 & Y = 2 |
| 9-8-2        | X = 9, Z = 8 & Y = 2 |
| 9-7-2        | X = 9, Z = 7 & Y = 2 |
| 9-4-2        | X = 9, Z = 4 & Y = 2 |
| 9-44-15-2    | X = 9, Z1 = 44, Z2 = 15 & Y = 2 |

3. BFGS Quasi-Newton

The BFGS algorithm (Broyden-Fletcher-Goldfarb-Shanno) is an algorithm that developed from the Quasi-Newton algorithm. BFGS algorithm is one of the algorithms that is used to solve nonlinear optimization problems without any constraints. Quasi-Newton equation [7] [8].

The BFGS Algorithm:

Step 0. Given \(x_1 \in \mathbb{R}^n\); \(B_1 \in \mathbb{R}^{nxn}\) positive definite;

Compute \(g_1 = \nabla f(x_1)\). If \(g_1 = 0\), stop; otherwise, set \(k := 1\).

Step 1. Set \(d_k = -g_k\).

Step 2. Carry out a line search along \(d_k\), getting \(\alpha_k > 0\),

\(x_{k+1} = x_k + \alpha_k d_k\), and \(g_{k+1} = \nabla f(x_{k+1})\); If \(g_{k+1} = 0\), stop.

Step 3. Set

\[
B_{k+1} = B - \frac{B_k S_k S_k B_k}{(S_k)^t B_k S_k} + \frac{Y_k Y_k}{(S_k)^t Y_k}
\]
where

\[ s_k = \alpha_k d_k, \]
\[ y_k = g_{k+1} - g_k. \]

Step 4. \( k := k+1; \) go to Step 1.

4. Experiments & Results

The test will be conducted first on the backpropagation standard with learning rate 0.5. Based on the test result, the smallest number of epoch that is obtained from 9-4-2 architecture is 231 with MSE 0.0619 and 91.6% accuracy. While the largest number of epoch is 10000 on architecture 9-59-2, 9-43-2, 9-8-2 and 9-7-2. The highest accuracy in standard BP testing is 98.6% with MSE 0.0203 while lowest accuracy is 91.6% with MSE 0.0619.

![Table 2. Standard Backpropagation Testing](image)

In BP testing with Quasi-Newton BFGS the number of epochs on the BP network with Quasi-Newton BFGS decreased significantly. The highest epoch value is 31 and the lowest epoch value is 13. While the accuracy on BP with BFGS decreases, the accuracy decrease in BFGS is not significant from the accuracy that had been produced by BP Standard. The lowest accuracy on BFGS is 92.1% with MSE 0.0483 and the highest is 95.8% with MSE 0.0335.

![Table 3. Backpropagation Testing with BFGS Quasi-Newton](image)

Based on the test, there was a significant decrease in epoch value. The greatest decrease in epoch value occurred on architecture 9-59-2 from 10000 (BP Standard) to 16 (BP BFGS) with 98.1% accuracy (BP Standard) and 95.8% (BP BFGS), 99.84% decrease epoch. The average decrease in the epoch number is 98.34%. The presentation of the decrease in epoch number from standard BP and BP with BFGS can be seen in Table 4.
Based on the research that has been done by testing 7 network architecture using BP standard and BP with BFGS Quasi-Newton weight changes, there is a decrease in the number of epoch in all network architecture accompanied by the increase of accuracy value on 1 architecture while on 6 others decreased accuracy. From the 7 networks tested there was a decrease in the epoch value in all the architectures. The comparison graph of the epoch value in standard BP with BP BFGS can be seen in Figure 2.

5. Conclusion
BP using BFGS method will be optimal with small number of neurons with decreased epoch 94.37% and accuracy improved by 0.5%. BP artificial neural networks can be used to classify glass dataset with proper architectural election. In a network architecture that has a smaller number of neurons, there is an increase in accuracy when using BP with BFGS. Based on the test results, it can be concluded that there is a significant decrease in the epoch number when using BFGS on back-propagation. Future task, the accuracy on BP with BFGS is still in reduction, so further research is needed to avoid the decreasing of accuracy.

References
[1] Panchal, G., Ganatra, A., Shah, P., & Panchal, D. 2011. Determination of over-learning and over-fitting problem in backpropagation neural network. International Journal on Soft Computing (IJSC) 2, pp. 40-51.
[2] Suciati, N. 2014. Batik Motif Classification using Color-Texture-Based Feature Extraction and Backpropagation Neural Network. Proceeding of IIAI 3rd International Conference on Advanced Applied Informatics, pp 517-512.
[3] Nazri, M.N., Meghana, R., & Rajesh, R. 2006. An Improved Learning Algorithm Based on The Broyden-Fletcher-Goldfarb-Shanno (BFGS) Method For Back Propagation Neural Networks. Proceedings of the Sixth International Conference on Intelligent Systems Design and Applications (ISDA’06).
[4] UCI Machine Learning Repository. 2017. Glass Identification Dataset. (Online) https://archive.ics.uci.edu/ml/datasets/glass+identification (5 Juni 2017).
[5] Stathakis, D. 2009. “How many hidden layers and nodes?”, International Journal of Remote
Sensing, vol. 30, pp. 2133–2147.
[6] Heaton, J. Introduction to Neural Networks for Java, Second Edition. Heaton Research: St. Louis.
[7] Dai, Y. H. 2002. Convergence Properties of the BFGS Algorithm. Proceeding of Society for Industrial and Applied Mathematics (SIAM), pp. 693–701.
[8] Dai, Y. H. 2012. A perfect example for the BFGS method. Proceeding of Mathematical Programming, pp. 501–530.
[9] Caruana, R. 2001. Overfitting in Neural Nets: Backpropagation, Conjugate Gradient, and Early Stopping. (Online) https://papers.nips.cc/paper/1895-overfitting-in-neural-nets-backpropagation-conjugate-gradient-and-early-stopping.pdf (18 Mei 2017).
[10] Cui, Z., Yuan, G., Sheng, Z., Liu, W., Wang, X., & Duan, X. 2015. A Modified BFGS Formula Using a Trust Region Model for Nonsmooth Convex Minimizations. (Online) http://journals.plos.org/plosone/article/authors?id=10.1371/journal.pone.0140606 (15 Juni 2017).
[11] Lawrence, S. & Lee, C. 2000. Overfitting and Neural Networks: Conjugate Gradient and Backpropagation. Proceeding of International Conference on Neural Networks, pp. 114-119.
[12] Heaton, J. 2015. Artificial Intelligence for Humans, Volume 3: Deep Learning and Neural Networks. Heaton Research: St. Louis.
[13] Koscak, J. & Jaksa, R. 2010. Stochastic Weight Update in The Backpropagation Algorithm on Feed-Forward Neural Networks. IEEE.
[14] Liu, S., L. Xu & D. Li. 2016. Multi-scale prediction of water temperature using empirical mode decomposition with back-propagation neural networks. Journal Computer and Electrical Engineering 49: 1-8.
[15] Rich, L., Y. Zeng & T. Chen. 2015. Backpropagation neural network with adaptive differential evolution algorithm for time series forecasting. Journal Expert Systems with Applications, pp. 855-863.