

Global Optimization of Neural Network

Chae Young Lee and Yeon Jun Lim*, Taeseon Yoon**

* Department of Natural Science, Hankuk Academy of Foreign Studies, 50, Oedae-ro 54beon-gil, Mohyeon-myeon, Cheoin-gu, Yongin-si, Gyeonggi-do, Republic of Korea

**Department of Computer Science and Engineering, Korea University, Seoul, Republic of Korea

limetree377@hafs.hs.kr, yeonjunlim@hafs.hs.kr, tsyoon@hafs.hs.kr

Abstract— The local optimization problem of cost functions has blocked the advancement of not only the traditional mathematics but also the emerging Neural Network (NN). Since most of the cost functions used in the gradient descent of NN is non-convex, there are high possibilities that the algorithm may converge at the local optimum. In order to eliminate the substantial error caused from this problem, deep learning has been introduced. Yet, standard computing environment cannot possibly power calculation that deep learning requires. The fake alpha algorithm, newly introduced in this paper, shares the same goal with the deep learning algorithm, but can successfully be operated in standard computers. Additionally, the heuristic approach that the fake alpha algorithm takes in the gradient descent process enhances the efficiency of NN.

Keywords— Artificial Neural Network, Fake Alpha Algorithm, Global Optimization

I. INTRODUCTION

Most Neural Network (NN) algorithms adjust their weight values through a process called Gradient Descent (GD) where a loss function is defined. [10] The purpose of GD is minimizing this usually non-convex loss function. [13] As GD is based on a direct search method – measuring previous against current calculated values – like greedy criterion, GD easily falls in the trap of local minima during iterations. This factor of GD lowers the accuracy and efficiency of NN. [2][6][7]

Several solutions to escape local minimum were suggested so far. The most widely known method is i) lengthening the iteration and ii) deepening the NN – deep learning. Both rely on a simple principle of increasing the amount of calculation. This enables the greedy criterion to compare most of the values and thus provides the overview of the loss function. [9] However, both possess critical problems: the former may eventually struggle with overfitting of training data where the latter cannot be operated in standard computing environments. [3][12][7]

Surprisingly, the problem of local minimum in NN can be easily solved without performing hard core computation by fake alpha (f-alpha) algorithm. As its name suggests, NN undergoes a fake gradient descent with a f-alpha value to update the weight values close to the global minimum. In other words, GD of a compressed form of loss function is held to find the approximate value of the global minimum. Therefore, the possibility that NN converges at local minima

drastically decreases. Figure 1 suggests the overall idea of this algorithm.

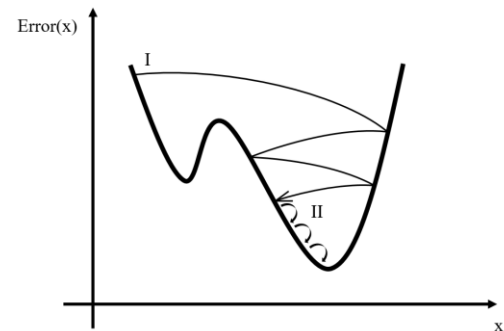


Figure 1. The Visual Description of f-alpha Algorithm where step I directs the prior step of gradient descent with normal alpha and step II, with f-alpha.

The efficiency of the f-alpha algorithm is confirmed through a series of controlled experiments with the famous Iris flower data set.

II. BACKGROUND

Neural Network (NN) is an algorithm that applies data-processing structure of neurons of the brain to logic circuit. Early NN focused on the implementation of biological neural network theory to a logic gate. However, NN has progressed while proposing the solution of XOR problem. Today, it became the representative algorithm of sub symbolic learning; it can be applied universally to classification problems. The standard type NN is composed of three sections: the input nodes, the hidden nodes, and the output nodes. Input nodes are multiplied by changeable floating number to generate hidden nodes. Hidden nodes repeat the same process to build the output nodes. These outputs are compared with the aimed values. These errors calculated are revised by the process of back propagation. The repetition of this process yields accurate results. [1][7]

The intermediate goal of NN is accurately calculating the minimum value of the cost function. The most common method used to achieve this is the gradient descent (GD). [8][4] GD is based on a delta rule that optimizes the objective function – the least mean square of the error which is

$$C = \frac{1}{2} (y_j - \hat{y}_j)^2$$

where $\hat{y}_{ij}$ is the calculated output value and y is the target output value. The delta rule is a method that approaches the global optimum by gradually correcting it as much as delta, based on the gradient of random point on the quadratic curve or quadratic surface. The delta rule is given by

$$\Delta w_{ji} = \alpha(y_j - \hat{y}_j)g'(h_j)x_i$$

where w_{ji} is the j 's i th weight, α is the learning rate, $g(x)$ is the activation function, h_j is the weighted sum of the input values, and x_i is the i th input. [11][2]

III. TEST CASES AND RESULTS

The Iris flower data set is well known for its usage as a test case for machine learning models. Introduced by the British statistician Ronald Fisher, the data set is comprised of 50 samples from three species of Iris: Iris setosa, Iris virginica, and Iris versicolor. [9] The data set used in this experiment is from the UCI Machine Learning Repository.

The algorithm was trained to recognize the species of the flower among three options, given the value of its four features: sepal length, sepal width, petal length, and petal width. The experiment is conducted to compare this training accuracy of the algorithm with that of the standard NN. Table 1 shows the environment setting that does not change throughout the whole experiment.

TABLE 1. ENVIRONMENT SETTING

Variable	Value
Input Unit	6
Hidden Unit	6
Output Unit	2
Error Limit	0.0001
Divergence Prevention	200,000

The number of input, hidden, and output units are graphically represented in Figure 1. The error limit lets the algorithm to escape the iteration when the error between the actual and aimed output values becomes smaller. Finally, divergence prevention is set in order to prevent the algorithm to diverge during gradient descent. [1]

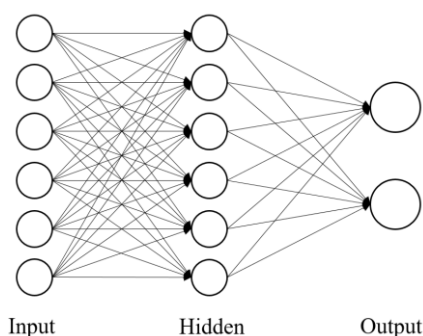


Figure 1. Diagram of both standard NN and f-alpha algorithm

As the goal of the experiment is confirming the efficiency of the f-alpha algorithm, the experiment is conducted with a test group (f-alpha algorithm) and a control group (standard NN algorithm). The test group has nine cases with different alpha. Each case is again divided into three subcases by the iteration number (10, 50, 100) – f-alpha iteration number for the test group and minimum alpha iteration number for the control group. Ten trials were done for each subcase. This specification of the experiment is introduced in Table 2.

TABLE 2. EXPERIMENT SPECIFICATION

Test Group		Control Group
<i>F-Alpha</i>	Alpha	Alpha
1	0.01	0.01
5		
10		
1	0.1	0.1
5		
10		
1	0.5	0.5
5		
10		

(sec)

The results of the experiment are shown as the amount of time required to finish the calculation. Since every case has ten trials, its resulting time is recorded as the mean value, the variance, and the standard deviation of ten trials as shown in Table 3.

TABLE 3. EXPERIMENT RESULT

	Test Group		Control Group	
	<i>f-alpha</i>	Value	<i>alpha</i>	Value
Mean	1	5.616	0.01	5.378
	5	1.364	0.1	0.455
	10	1.173	0.5	0.158
	Average	2.718	Average	1.996
S.D.	1	1.167	0.01	0.441
	5	0.156	0.1	0.816
	10	0.127	0.5	1.134
	Average	0.484	Average	0.797

(sec)

The mean values and the standard deviations of both groups in Table 3 are compared in Figure 3 and Figure 4, respectively.

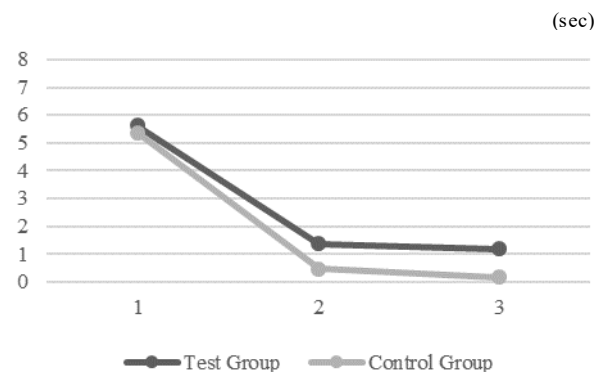


Figure 3. Mean Value of Operating Time

The mean value of the algorithm's operational time generally decreases as the alpha increases, in both the test and the control group. That is, the efficiency of the algorithm increases proportionally to the alpha value. The standard deviation, however, showed relatively various aspects. The data gradually descended in the test group, while that of the control group increased in proportion to the alpha value.

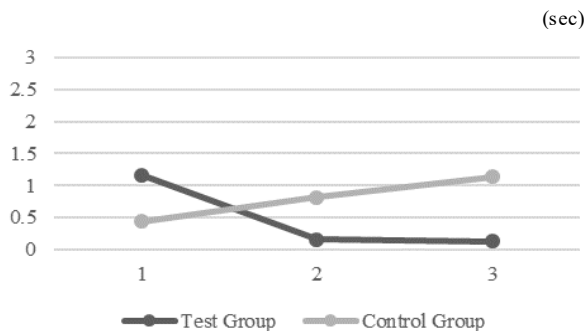


Figure 4. Standard Value of Operating Time

The average of all values in Figure 3 and 4 is represented in Figure 5. The mean value of the test group is larger than that of the control group where the standard deviation is the opposite. This proves the accuracy of the f -alpha algorithm. The mean value indicates that the test group needs more time to complete its calculation. This is because the gradient descent of the f -alpha algorithm is done in a smaller range. The lower value of the standard deviation suggests that the algorithm is more credible and accurate than the standard NN. The results, overall, prove that the f -alpha algorithm deals with smaller numbers and thus, result in more accurate outputs.

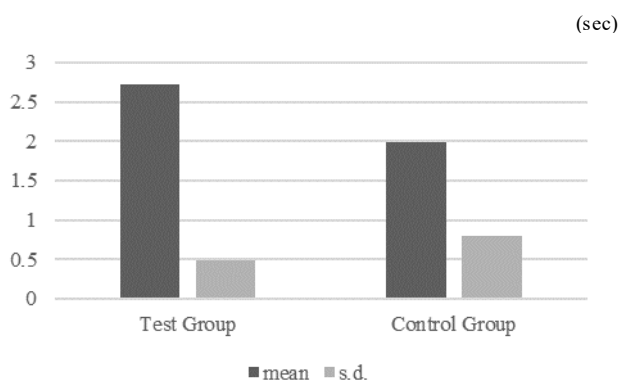


Figure 5. Average Experiment Results

IV. CONCLUSION

The experiment held to test the efficiency of the f -alpha algorithm achieved its purpose; the f -alpha algorithm enhances the accuracy of the outputs. Though the experiment was conducted with relatively small amounts of dataset, it has apparently shown the potential of the f -alpha algorithm. That is, f -alpha algorithm may contribute in solving the notorious deep learning problem as well as the global

optimization problem. The deep learning algorithm, notwithstanding its accuracy, has been criticized for its inefficiency as normal computing environment cannot endure the heavy calculation that it requires. [12] Although it is impossible for f -alpha algorithm alone to completely substitute deep learning, the implementation in each unit of deep learning may lighten the amount of calculation and therefore, improve the efficiency.

Still, improvements must be sought. Further research is required to find the ideal relationship of f -alpha values, f -alpha iterations, and the datasets. Without it, the amount of trials needed to achieve the utmost results outnumbers that of a standard neural network.

REFERENCES

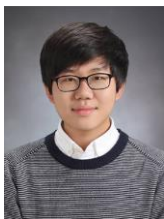
- [1] Basheer, I., and M. Hajmeer. "Artificial neural networks: fundamentals, computing, design, and application." *Journal of Microbiological Methods* 43 (2000): 6-7. Web.
- [2] Bengio, Y., P. Simard, and P. Frasconi. "Learning long-term dependencies with gradient descent is difficult." *IEEE Transactions on Neural Networks* 5.2 (1994): 157-66. Web.
- [3] Choromanska, A., M. Henaff, M. Mathieu, G. B. Arous, and Y. LeCun. "The Loss Surfaces of Multilayer Networks." *Appearing in Proceedings of the 18th International Conference on Artificial Intelligence and Statistics* 38 (2015): 1-8. Web.
- [4] Dreyfus, Stuart E. "Artificial neural networks, back propagation, and the Kelley-Bryson gradient procedure." *Journal of Guidance, Control, and Dynamics* 13.5 (1990): 926-28. Web.
- [5] Fisher, R. A. "The Use of Multiple Measurements in Taxonomic Problems." *Annals of Eugenics* 7.2 (1936): 179-88. Web.
- [6] Glorot, X., and Y. Bengio. "Understanding the difficulty of training deep feedforward neural networks." *Proceedings of Machine Learning Research* 9 (2010): 249-56. Web.
- [7] Hinton, Geoffrey E., Simon Osindero, and Yee-Whye Teh. "A Fast Learning Algorithm for Deep Belief Nets." *Neural Computation* 18.7 (2006): 1527-554. Web.
- [8] Kiwiel, Krzysztof C. "Convergence and efficiency of subgradient methods for quasiconvex minimization." *Mathematical Programming* 90.1 (2001): 1-25. Web.
- [9] Mobahi, Hossein, and John W. Fisher. "On the Link between Gaussian Homotopy Continuation and Convex Envelopes." *Lecture Notes in Computer Science Energy Minimization Methods in Computer Vision and Pattern Recognition* (2015): 43-56. Web.
- [10] Qian, Ning. "On the momentum term in gradient descent learning algorithms." *Neural Networks* 12.1 (1999): 145-51. Web.
- [11] Rumelhart, David E., and James L. McClelland.

Parallel distributed processing. Explorations in the microstructure of cognition. vol. 1: Foundations. MIT: Cambridge, 1986. Print.

- [12] Shankar, Sukrit, Duncan Robertson, Yani Ioannou, Antonio Criminisi, and Roberto Cipolla. "Refining Architectures of Deep Convolutional Neural Networks." *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016): n. pag. Web.
- [13] Strongin, Roman G., and Yaroslav D. Sergeyev. "Multidimensional Parallel Algorithms." *Global Optimization with Non-Convex Constraints Nonconvex Optimization and Its Applications* (2000): 551-610. Web.



Chae Young Lee was born in Iksan, Korea, in 2000. Since March 2016, she has been studying in the Department of International, Hankuk Academy of Foreign Studies. She is currently with the Artificial Intelligence Research Institute (AIRI) as a research intern. She was the recipient of the Best Researcher Award of the Future Scientist Conference, Seoul, Korea, 2017.



Yeon Jun Lim was born in Seoul, Korea, in 2000. Since March 2016, he has been studying in the Department of Natural Science, Hankuk Academy of Foreign Studies. His research interest is in artificial intelligence, neuroscience, and biology.



Taeseon Yoon was born in Seoul, Korea, in 1972. He received the Ph.D. candidate degree in computer education from the Korea University, Seoul, Korea, in 2003. From 1998 to 2003, he was with EJB analyst and SCJP. From 2003 to 2004, he joined the Department of Computer Education, University of Korea, as a lecturer and Ansan University, as an adjunct professor. Since December 2004, he has been with the Hankuk Academy of Foreign Studies, where he was a computer science and statistics teacher. He was the recipient of the Best Teacher Award of the Science Conference, Gyeonggi-Do, Korea, 2013.