Bi-activation function : an Enhanced Version of an Activation function in Convolutional Neural Network

1st Seoung-Ho Choi
Dept. of Electronic Information Engineering
Hansung University
Seoul, Republic of South Korea
jcn99250@naver.com

2nd Kyoungyeon Kim
Dept. of Electronic Information Engineering
Hansung University
Seoul, Republic of South Korea
tkakrdudndi@gmail.com

dj

***Abstract*— An activation function has been mostly used for training a convolutional neural network (CNN). The activation function consists of two parts. One is a blocking area when x < 0 and the other is a linear output area when x >= 0. We found that the activation function does not efficiently reflect information about parts of x >= 0 and x < 0. From this observation, we propose a bi-activation function, consisting of a pos-activation function and a neg-activation function. The pos-activation function is the same as the existing activation function, and the neg-activation function has a blocking area when x >= 0 and a linear output area when x < 0. We experimented with CNN and typical datasets such as MNIST, Fashion MNIST, CIFAR-10, and CIFAR-100. The bi-activation function performed better than RELU and eLU.**

Keywords*— *The bi-activation function, Nonlinear feature, Convolutional Neural Networks

I. INTRODUCTION

RELU, a function frequently used as an activation function, can be simply expressed : $f(x) = \max(0, x)$, when x is the input of a neuron. That is, the output is zero when x < 0 (called a blocking area) and the output is the same as the input when x >= 0 (called a linear output area). Those properties indicate that the only positive part of the existing activation function is reflected and it takes more time to train the model. In addition, it is a disadvantage that it does not reflect generalized characteristics in the model because it does not reflect negative partial information.

To improve this problem, we propose a bi-activation function as an improved version of a activation function. To verify the performance of the bi-activation function, we extensively experimented on CNN with typical datasets such as MNIST, Fashion MNIST, CIFAR-10, and CIFAR-100 and compared the results with those of RELU and eLU. As a result, the bi-activation function shows better performance than RELU and eLU in all experiments.

II. RELATED WORK

The RELU and eLU experimented have been used as the activation function. RELU has the characteristic of reflecting through a linear filter in positive information. eLU has characteristics that reflect nonlinear characteristics in positive information. Unlike previous research, the newly proposed a bi-activation function is as follows.

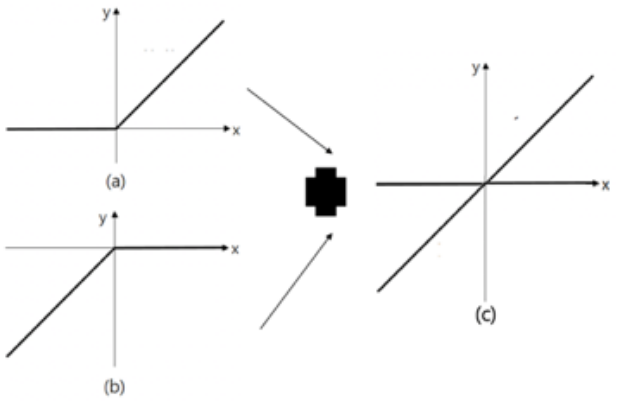

Fig. 1 Process of bi-activation function: (a) Pos-activation function, (b) Neg-activation function, (c) bi-activation function.

III. BI ACTIVATION FUNCTION

Bi-activation function consists of two types: pos-activation function and neg-activation function. In terms of RELU, the pos-activation function is the same as the existing RELU. That is, there is a blocking area when x < 0 and a linear output area when x >= 0. Neg-activation function has a blocking area when x >= 0 and a linear output area when x < 0. Simply neg-activation function is expressed: $f_x = \max(x, 0)$. Fig 1.(c) show bi-activation function, which consists of pos-activation function and neg-activation function. Pos-activation function has a blocking area, when x < 0 and a linear output area, when x >= 0 and neg-activation function has vice versa. CNN reflect the nonlinearity of inputs and outputs of training data using the activation function by properly selecting the linear output area.

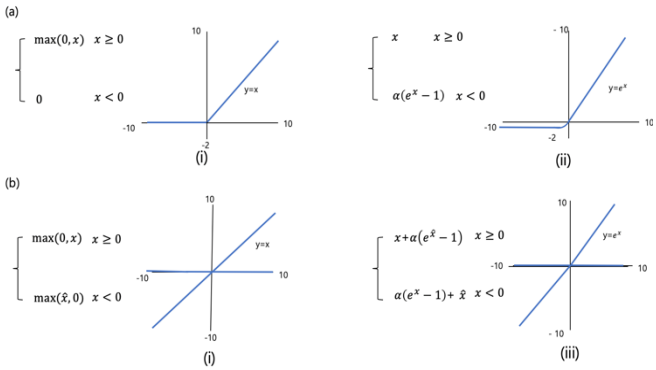

Fig. 2 Experiment of activation function: (a) Existing method, (b) Proposal method, (a).i RELU, (a).ii eLU, (b).i bi-RELU, (b).ii bi-eLU.

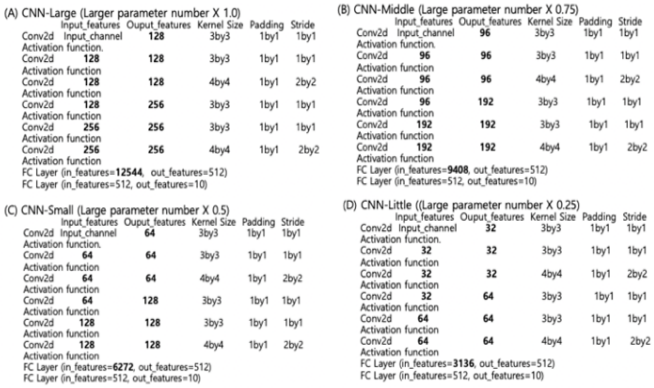

Fig. 3 Experiment of model sample: (a) CNN-Large, (b) CNN-Middle, (c) CNN-Small, (d) CNN-Little.

IV. EXPERIMENTAL RESULTS

We tested the novel bi-activation function. The proposed method applied to existing functions. Fig. 2 show the result of the experiments. Firstly, Fig 2.(a) show RELU, and eLU and Fig 2.(c)-(d) shows the result of applying the bi-activation function method to RELU and eLU, respectively. The CNN model used in this paper is intended to see the effect of a large-margin model prediction effect using Log SoftMax. When applying this log SoftMax, NLL Loss is generally applied. However, there is a problem that a non-convex effect causes the NLL loss. Therefore, cross-entropy is applied to generate the convex effect of the model. This is because the convex function creates a unique solution and can be easily solved using the gradient method. Therefore, studies have been conducted to apply convex after applying cross-entropy to log SoftMax [3]. The CNN model is defined as a validation model to verify that the activation function experiment group presented. Fig. 2 is efficient even when there are few parameters. Four models are composed of Fig 3.(a) CNN-Large, Fig 3.(b) CNN-Middle, Fig 3.(c) CNN-Small, and Fig 3.(d) CNN-Little. The number of CNN-Large filter maps based. The model was designed based on the number of filter maps with CNN-Middle 0.75 times, the number of filter maps with CNN-Small 0.5 times, and the number of filters with CNN-Little 0.25 times. The loss used for the CNN model uses cross-entropy, and the optimization method is experimentally verified using the Adam optimization. We experimented with training datasets such as MNIST, Fashion MNIST, CIFAR-10, and CIFAR-100. When the proposed method is better than the activation function, it is indicated in bold.

We compare the experiments using the MNIST dataset with Seed 999, 500, and 1 to analyze the influence of each filter number. The experimental results are to verify results obtained from Table 1. That is different depending on the number of filters. First of all, linear activation showed a decrease in test value when there were a large number of filters. However, eLU with nonlinear features exhibited a performance increase and decreased as the number of filters decreased, which confirms that the appropriate number of filters should found when inferring through the Activation function in the model. Also, there is a problem of finding an appropriate number of filters, even when the bi-activation function is applied. When the bi-RELU is applied, the accuracy increases linearly and decreases as the number of filters decreases. The bi-eLU exhibits a nonlinear phenomenon in which the accuracy increases as the number of features decreases, then decreases and then increases. The performance varies depending on the number of filters and the nonlinearity of the activation function.

The proposal method receives both positive and negative information from the activation function and outputs less error value and improved performance than activation that seems to improve performance by making the feature a little clearer by processing both positive and negative information at the same time. The comparison of the same number of filters showed that most of them improved over a activation function.

Table 1. Comparison of influence according to the number of CNN model feature maps (a) CNN-Large, (b) CNN-Middle, (c) CNN-Small, (d) CNN-Little.

	RELU [1]		**eLU [2]**		**bi-RELU**		**bi-eLU**	
(a)	**Loss**	**Acc**	**Loss**	**Acc**	**Loss**	**Acc**	**Loss**	**Acc**
Train	2.303	0.110	2.822	0.139	**1.961**	**0.353**	**2.503**	**0.260**
Test	X	0.062	X	0.125	X	**0.312**	X	**0.260**
(b)	Loss	Acc	Loss	Acc	Loss	Acc	Loss	Acc
Train	2.302	0.110	2.828	0.137	**1.912**	**0.361**	2.368	**0.291**
Test	X	0.125	X	0.031	X	**0.166**	X	**0.078**
(c)	Loss	Acc	Loss	Acc	Loss	Acc	Loss	Acc
Train	2.303	0.110	2.826	0.138	**1.897**	**0.371**	2.548	**0.235**
Test	X	0.125	X	0.0625	X	**0.5**	X	**0.1875**
(d)	Loss	Acc	Loss	Acc	Loss	Acc	Loss	Acc
Train	2.302	0.110	2.828	0.137	**1.912**	**0.361**	2.368	**0.291**
Test	X	0.125	X	0.031	X	**0.166**	X	**0.078**

We test the proposal method in this paper with the CNN-Small model and 2 layer CNN to verify the effect of the proposal method.

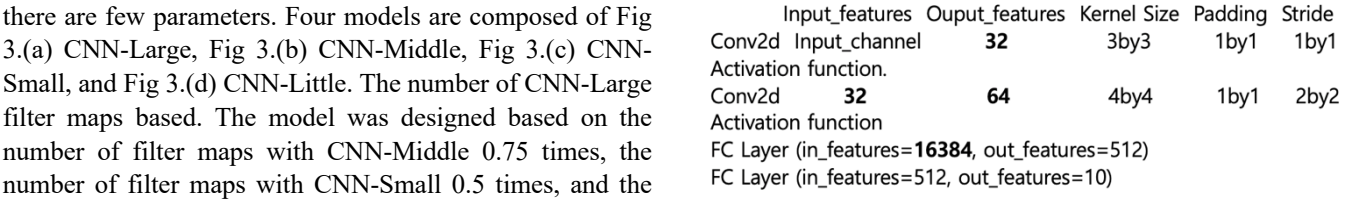

Fig. 4 Experiment of two layer CNN

To analyze the initial random influences of the proposed method, we conducted experiments about three seed values, Seed 999, Seed 500, and Seed 1.

Table 2. Train / Test accuracy and loss in CNN-Small according to Seed 999 on CNN small.

(a)	MNIST		Fashion MNIST		Cifar10		Cifar100	
Train	**Loss**	**Acc**	**Loss**	**Acc**	**Loss**	**Acc**	**Loss**	**Acc**
RELU[1]	2.303	0.110	2.304	0.101	2.304	0.103	4.611	0.01
eLU[2]	2.826	0.138	2.820	0.139	2.841	0.136	6.918	0.035
bi-RELU	**1.897**	**0.371**	**1.737**	**0.397**	2.431	**0.161**	5.136	**0.041**
bi-eLU	**2.548**	**0.235**	**2.445**	**0.256**	**2.766**	**0.150**	**6.611**	**0.041**
Test	**Acc**		**Acc**		**Acc**		**Acc**	
RELU[1]	0.125		0.1875		0.1875		0	
eLU[2]	0.0625		0.125		0		0	
bi-RELU	**0.5**		**0.3125**		0.041		0	
bi-eLU	**0.1875**		**0.3125**		0		0	

Table 3. . Train / Test accuracy and loss error in CNN-Small according to Seed 500 on CNN small.

(b)	MNIST		Fashion MNIST		Cifar10		Cifar100	
Train	**Loss**	**Acc**	**Loss**	**Acc**	**Loss**	**Acc**	**Loss**	**Acc**
RELU[1]	2.303	0.111	2.304	0.102	2.304	0.102	4.610	0.009
eLU[2]	2.834	0.139	2.839	0.139	2.84	0.137	6.902	0.035
bi-RELU	**1.836**	**0.395**	**1.621**	**0.437**	2.407	**0.164**	5.071	**0.044**
bi-eLU	**2.524**	**0.241**	**2.296**	**0.301**	**2.807**	**0.154**	**6.705**	**0.04**
Test	**Acc**		**Acc**		**Acc**		**Acc**	
RELU	0.125		0.125		0.1875		0	
eLU	0.0625		0.0625		0.0625		0	
bi-RELU	**0.3125**		**0.4375**		0.0625		0	
bi-eLU	**0.1875**		**0.1875**		0.03125		0	

Table 4. Train / Test accuracy and loss error in CNN-Small according to Seed 1 on CNN small.

(c)	MNIST		Fashion MNIST		Cifar10		Cifar100	
Train	**Loss**	**Acc**	**Loss**	**Acc**	**Loss**	**Acc**	**Loss**	**Acc**
RELU[1]	0.142	0.973	0.41	0.863	2.304	0.102	4.61	0.010
eLU[2]	2.843	0.139	2.834	0.14	2.84	0.137	6.923	0.035
bi-RELU	1.931	0.357	1.693	0.405	2.407	**0.164**	5.109	**0.041**
bi-eLU	**2.508**	**0.254**	**2.361**	**0.289**	**2.807**	**0.154**	**6.766**	**0.042**
Test	**Acc**		**Acc**		**Acc**		**Acc**	
RELU	0.875		0.875		0.1875		0	
eLU	0.0625		0.125		0		0	
bi-RELU	0.437		0.4375		0.0416		0	
bi-eLU	**0.125**		**0.25**		0		0	

We perform a quantitative analysis of the experimental results using the average of three seed test results. In Table 2., the proposed method in this paper obtained an increase of 0.305 on average in the case of Train in bi-RELU in the MNIST dataset. The accuracy was reduced by 2.3%. The test resulted in a 4.1% improvement. In bi-eLU, the average loss was reduced by 0.307 during the train. Accuracy increased by 10.4%. W improved by 10.4%. The average of both methods loss results reduces by 0.001, a 4.05% improvement in the train, and a 7.29% accuracy increase in the test. In Fashion-MNIST, On average loss increased 0.011, accuracy increased 5.766% on the test, and the same on test in bi-RELU. On average, bi-eLU showed a loss reduction of 0.463 in loss, Improve accuracy 14.2% in the train, and 14.58% in the test. The average of two methods reduce loss 0.226 in the train, improves the accuracy by 10.0% in the train, and improves the accuracy by 7.29% in the test. In CIFAR-10, in the case of bi-RELU, Loss shows a 0.0345 increase in a train, 6.96% accuracy improvement in a train and a 13.9% inaccuracy reduction in the test. In bi-eLU, loss decreases by 0.036, 1.46% on average accuracy improve, and the same in the test. The average of the two methods is 0.023 increase in loss, 4.2% accuracy improvement in a train, and a 6.9% accuracy reduction in the test. In the case of CIFAR-100, bi-RELU shows an average loss 0.486 increase in accuracy, 3.23% accuracy in trains, and the same result in the test. In bi-eLU, Loss decreases by 0.172 on average, improves accuracy by 0.67%, and is the same when testing. The average of both methods is 0.156 increase in loss, 1.95% increase of accuracy, and the same in the test. Finally, the average result of improvement and reduction of the four data sets shows that the bi-activation function method shows 0.018 reductions in train, 5.06% accuracy improves, and 1.8975% accuracy improves in test. In the case of MNIST and Fashion MNIST, the proposed method shows the improved results in Train and Test, which shows that the performance is improved by learning more effectively the positive information and negative information model and obtaining a more precise decision boundary. However, the complexity of data from CIFAR-10 increased as the number of image sizes, and image channels increased compared to the MNIST series. Nevertheless, the method proposed in this paper considers both positive information and negative information at the same time so that the average accuracy increased by classifying through clear boundaries in the train, but the generalized boundary not found in the test accuracy due to the poor performance. We can see that the decision boundary that led to the data found. This cause is seen to occur as the data size and data channel increase. In the case of data of CIFAR-100, the complexity increases as the number of classes in this model increases, so it inferred that the data not adequately learned. That shows that the loss value is significantly higher than the data set result. Also, we can see that the method proposed in this paper decreases when the performance increases when tested with various seed values. This method is more affected by the influence of the initial value because the method proposed in this paper reflects bi-directional information confirm. Finally, the method proposed in this paper influenced by data and seed value, but it confirmed that reflecting positive and negative information helps to improve model learning and performance by maximizing the margin between model information.

Fig. 5 is composed of 2 parts. The upper scatter plot is the result of the variance of the activation function. The lower histogram is the result of the performance of the activation function. We include the results of experiments with five seeds and two CNN models in Figure 5. The above experiment calculates the average for each activation function. The notation is written as (μ, σ). μ is mean, σ is the standard deviation. RELU is (0.54227, 0.2271). eLU is (0.543914, 0.29004). bi-RELU is performance analysis of proposal methods in Fig. 5. x) small of CNN, y) two layer of CNN, a) Using the MNIST dataset, b) Using Fashion MNIST, A) Activation function of RELU, B) Activation function of eLU, C) Activation function of bi-RELU, and D) Activation function of bi-eLU. The results of Figure 5 show more

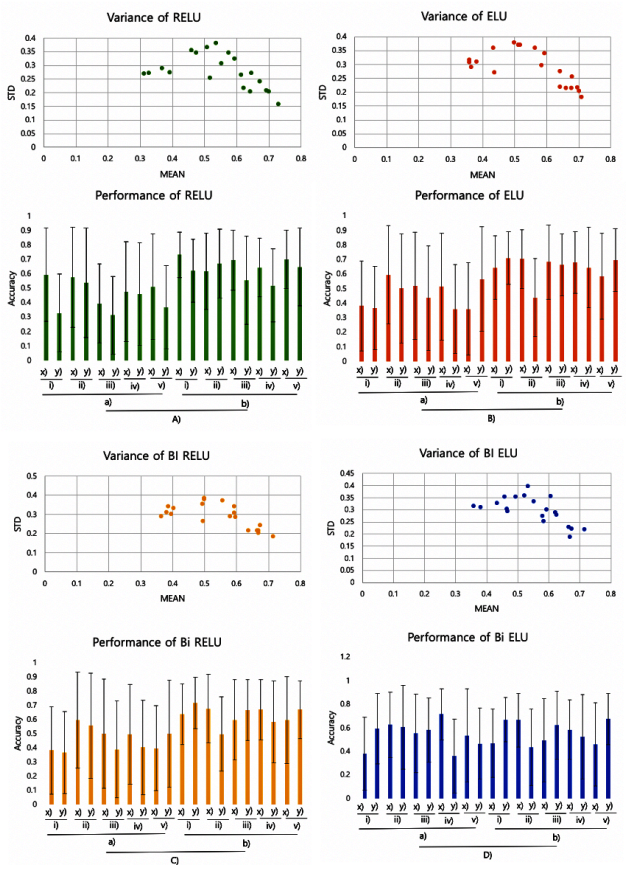

Fig. 5 Performance analysis of proposal methods. x) small of CNN, y) two layer of CNN, a) Using MNIST dataset, b) Using Fashion MNIST, A) Activation function of RELU, B) Activation function of eLU, C) Activation function of bi-RELU, D) Activation function of bi-eLU, i) Seed 1, ii) Seed 250, iii) Seed 500, iv) Seed 750, v) Seed 999

clustered plots of ELU variance compared to RELU. This is because the nonlinear characteristics reflect more clustered results. In Figure 5, the results of experiments applied to each activation function for each of the two models show that some variation in performance occurs for each seed.

V. Conclusion

We have demonstrated an improved performance by the bi-activation function. Bi-activation function combines pos-activation function and neg-activation function in a small number of parameter spaces. Compared with the existing activation function, the bi-activation function considers bi-directional information to reflect generalization characteristics in the model through bi-directional information. As this reflects the bi-directional characteristic, it can be seen that the convergence speed is faster in the learning process than when reflecting on the existing single characteristic. Because reflecting the information with the existing activation function has a high complexity in processing the information while processing through the bi-directional information, the complexity is somewhat lower, so the convergence speed seems to be faster. To show the advantage of the proposed activation function, we verified the effect of the initialization method on the input of the bi-directional information in a few parameters that show that bi-directional information is a little bit better when it is nonlinear. Since the weight of the model of the existing deep learning has a value between 0 and 1, the weight information of the deep learning model between 0 and 1 more effectively reflects the nonlinear characteristics through the activation function having a nonlinear characteristic. This property of the proposed transform effectively used for optimization or edge device deep learning.

References

[1] V. Nair and G. E. Hinton, “Rectified Linear Units Improve Restricted Boltzmann Machines,” *In ICML*, 2010.

[2] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, “Fast and Accurate Deep Network Learning by Exponential Linear Units,” *arXiv:1511.07289*, 2015.

[3] W. Liu, Y. Wen, Z. Yu, and M. Yang, “Large-Margin Softmax Loss for Convolutional Neural Networks,” *arXiv:1612.022954v4*, 2017