IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received September 4, 2019, accepted October 6, 2019, date of publication October 17, 2019, date of current version October 30, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2948112

Improved Convolutional Neural Network Based on Fast Exponentially Linear Unit Activation Function

ZHENG QIUMEI, TAN DAN, AND WANG FENGHUA
Department of Computer and Communication Engineering, China University of Petroleum, Qingdao 266580, China

Corresponding author: Wang Fenghua (fenghuawang@upc.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61305008, and in part by the Fundamental Research Funds for the Central Universities under Grant 19CX02030A.

ABSTRACT The activation functions play increasingly important roles in deep convolutional neural networks. The traditional activation functions have some problems such as gradient disappearance, neuron death and output offset, and so on. To solve these problems, we propose a new activation function in this paper, Fast Exponentially Linear Unit (FELU), aiming to speed up exponential linear calculations and reduce the time of network running. FELU has the advantages of Rectified Linear Unit (RELU) and Exponential Linear Unit (ELU), leading to have better classification accuracy and faster calculation speed. We test five traditional activation functions such as ReLU, ELU, SLU, MPELU, TReLU, and our new activation function on the cifar10, cifar100 and GTSRB data sets. Experiments show that the proposed activation function FELU not only improves the speed of the exponential calculation, reducing the time of convolutional neural network running, but also effectively enhances the noise robustness of network to improve the accuracy of classification.

INDEX TERMS Activation function, deep learning, exponential function.

I. INTRODUCTION

In recent years, deep learning has achieved remarkable results in the field of computer vision application, such as image classification [1]–[4], object detection [5]–[9], image retrieval [10] and so on [11]. The convolutional neural network is a trainable multi-layer network structure composing of multiple single-layer convolutional neural networks. Each single-layer convolutional neural network consists of three basic stages: feature extraction, nonlinear activation, and downsampling. The activation function can retain the features extracted by the convolutional layer, remove redundant data, and map out the features by using nonlinear functions. At present, many activation functions are applied in convolutional neural networks, such as Sigmoid [12], Tanh [13], ReLU [14], PReLU [15], RReLU [16], Leaky_ReLU [17], ELU [18], etc. [19], [21]–[25]. The Sigmoid and Tanh functions belong to the saturation activation function, which have gradient disappearance problem in deep convolutional neural networks [20]. Using a non-saturated activation function instead of a saturation activation function lays the foundation for solving this problem. The unsaturated activation function can not only solve the gradient disappearance phenomenon, but also accelerate the convergence speed.

Nair *et al.* [14] proposed a simple unsaturated activation function ReLU (Rectified Linear Unit) in 2010 to solve the problem of vanishing gradient of the saturation activation function. It produced profound effect on the development of convolutional neural network. However, since the ReLU function only keeps the positive part, the negative part is always zero. This situation leads to neuron death and output offset, which affects the convergence of the neural network. In order to solve this problem, Anthimopoulos *et al.* [17] proposed a new function LReLU (Leaky ReLU) in 2013, which an improved version of ReLU function. It introduces a function with a variable parameter on the negative part, which can activate negative features, to effectively solve the problem of vanishing gradient, but the experimental process is complicated because the appropriate variable parameter needs to be obtained through multiple training. He *et al.* [15] proposed a Parametric Rectified Linear Unit (PReLU) in 2015, which an improved version of LReLU. In PReLU, the slopes of

The associate editor coordinating the review of this manuscript and approving it for publication was Nikhil Padhi.

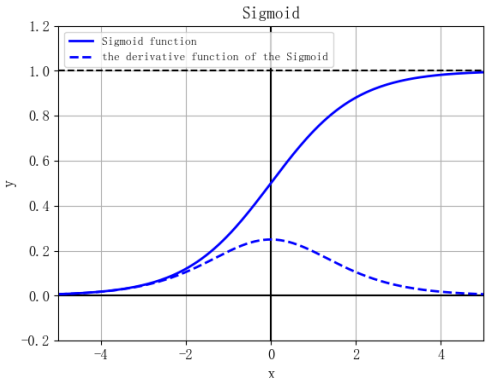


FIGURE 1. Sigmoid function.

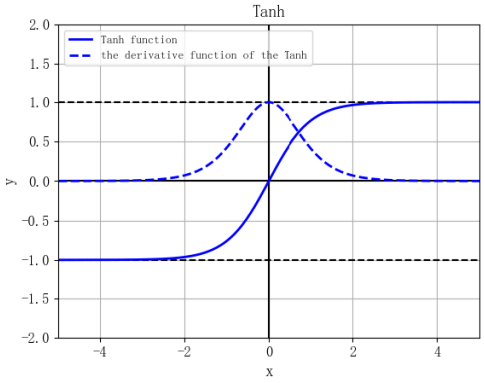


FIGURE 2. Tanh function.

negative part are learned form data rather than predefined. It is the key to transcending human classification. However, the noise robustness of the network in the in-active state is poor, owing to the existence of the linear function of the negative part. Clevert *et al.* [18] proposed an Exponential Linear Unit (ELU) in 2015. Exponential operation on the negative part leads the mean of output close to zero, which makes the gradient closer to the natural gradient and enhances the noise robustness. However, due to the large amount of calculation in the gradient, ELU slows down the speed of the network running. Li *et al.* [20] proposed a Multiple Parametric Exponential Linear Unit (MPELU) in 2018. It can improve the accuracy of classification by introducing multiple parameters to control the exponential function on the negative part, but it is slower in calculation.

Aiming at the current problems such as gradient disappearance, output offset and slow running speed, the main contribution of this work includes: we propose a new activation function FELU to improve the performance of the convolutional neural network by optimizing the exponential function on the negative part. In the negative part, using a new exponential function that can speed up the calculation effectively speeds up the network operation. Experiments show that the network model based on FELU function has faster running speed and better classification accuracy.

This paper is organized as follow. The related work is reviewed in Section 2. The new activation function is proposed in Section 3. The experimental results and analysis are given in Section 4. Finally, Conclusion is given in Section 5.

II. RELATED WORK

A. SATURATION ACTIVATION FUNCTION

The common saturation activation functions include Sigmoid function [12] and Tanh function [13], and their curves are shown in Fig. 1 and Fig. 2, respectively.

The Sigmoid function compresses all real numbers between (0,1). It has a large gain from the middle region and the output is bounded. When the input is large, the output is closer to one, otherwise the output is closer to zero. Its derivative function is simple to solve ant its mathematical form is shown in equation (1).

$$f(x) = \frac{1}{1 + e^{-x}} \tag{1}$$

Sigmoid function is rarely used now. Firstly, the Sigmoid function will occur the gradient disappearance phenomenon when it is backpropagating. When the input is very large or very small, the output enters the saturation region of the function, and the derivative also tends to zero, which is prone to occur gradient disappearance phenomenon, resulting in the neural network unable to complete the training. And the Sigmoid function has a characteristic of output zero mean, which causes the gradient to either be positively updated or negatively updated, which is detrimental to network convergence [26]. Finally, the Sigmoid function involves the power operation, which increases the training time of the network. The appearance of the Tanh solves the problem of non-zero mean of the Sigmoid function. It has better fault tolerance, and it is superior to the Sigmoid function, but it still has problems of vanishing gradient and power operation. The mathematical form of the Tanh function is shown in Equation (2).

$$f(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \tag{2}$$

Therefore, the Sigmoid and Tanh activation functions still have significant limitations in gradient disappearance and the time of training.

B. UNSATURATION ACTIVATION FUNCTION

In order to solve the problem of saturation activation function, the researchers propose a series of unsaturation activation functions, including ReLU function [14], PReLU function [15] and ELU function [18], and so on.

1) RELU ACTIVATION FUNCTION

The ReLU activation function is one of the most widely used activation functions in neural networks [14], [21], [27], which can effectively solve the problem of vanishing gradient and slow training time of saturated S-type activation function.

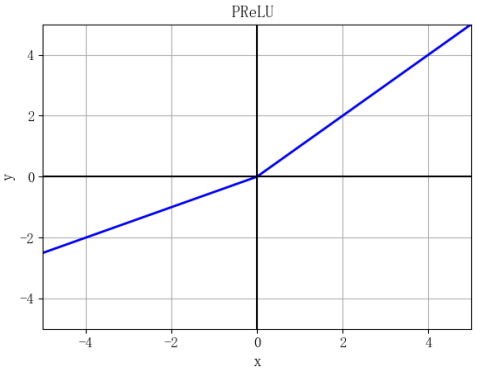


FIGURE 3. PReLU function.

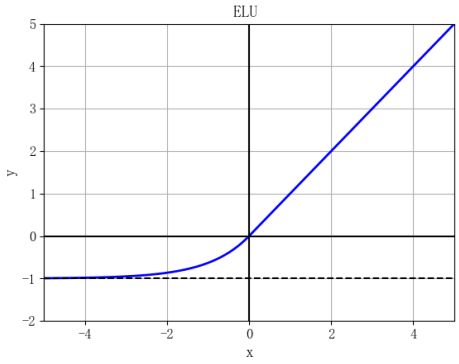


FIGURE 4. ELU function.

Its mathematical expression is as shown in equation (3).

$$f(x) = \begin{cases} x_i, & x_i \geq 0 \\ 0, & x_i < 0 \end{cases} \tag{3}$$

When the input is greater than zero, the output is linearly mapped. Conversely, the output is forced to zero. This enhances the sparseness of the network, reduces the interdependence of parameters, and effectively alleviates the problem of the parameter over-fitting. The ReLU function does not contain a power operation, so it can reduce the amount of calculation and speed up the training time of the network [26]. However, the ReLU function has a hard saturation property because of the negative part of ReLU is always zero, which makes the weight value of the input that input into the hard saturated region to zero during training, and then results in the problem of neuron death. The ReLU function also keeps the output mean value of the function be greater than zero because the negative part is always zero, which makes it difficult for the network to find the direction with the fastest gradient drop in the back propagation process, thus affecting the network convergence [21].

2) PRELU ACTIVATION FUNCTION

The PReLU function was proposed in 2015, and it solved the problem that the negative part of ReLU is always zero, and its curve is shown in Fig. 3.

The negative part of the PReLU has a variable parameter α, which can be learned from the network during the training process, and its value range is between α. The PReLU has values of output on the negative part, which makes function have a feature of the output zero-mean, ensuring the input on the negative part can be activated. However, owing to the parameter α needs to be learned from the training process, the PReLu will increase the amount of computation of the network, and also cause parameter over-fitting on small data sets, affecting the final classification accuracy.

3) ELU ACTIVATION FUNCTION

Inspired by natural gradients [28], Clevert *et al.* [18] proposed an activation function ELU based on exponential operation in 2015. Similar to the PReLU function, there has values of output on the negative part. Unlike PReLU, an exponential function is used on the negative part. Experiments have shown that the gradients conveyed by the ELU are closer to the natural gradient. Its curve is shown in Fig. 4.

The ELU function adopts the positive part of the ReLU function, which can effectively avoid the gradient disappearance phenomenon. Introducing an exponential function on the negative part makes the ELU function have an output zero-mean property, which prevents the occurrence of neuronal death. However, due to the complicated calculation of the exponential operation, the network is still much slower in running time than the ReLU function.

In summary, although ELU solves the problem of neuronal death and output offset of ReLU, and also has the advantages of ReLU, which can prevent the gradient disappearing phenomenon. However, the neural network with ELU function runs much slower than with the ReLU function. Since the running speed plays an important role in the real-time of the application of the convolutional neural network, a new activation function is needed to solve this problem. Therefore, we propose a new activation function to improve the ELU function, which not only maintains the advantages of the ELU function, but also speeds up the network operation.

III. NEW ACTIVATION FUNCTION (FELU)

In this section, we propose a new activation function FELU to improve ELU function. The contribution of FELU is using a fast exponential function that approximates the natural gradient calculation in the negative part, which can speed up the calculation to reduce the running time of network by speeding up the calculation of exponent.

A. FAST EXPONENTIAL CALCULATION

The exponential function is the most typical nonlinear function in the neural network. The application in the neural network often uses most of the time in calculating. The typical exponential functione^x is calculated by using the power series expansion, and the approximate calculation of summing the first n terms is shown in equation (4).

$$e^x = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + L + \frac{x^n}{n!} \tag{4}$$

The standard math library of exp function has a high precision. The e is an infinite loop number, which value is 2.718281828459……, and the exp function takes a long time to calculate, so we need to find a fast exponential function to speed up the calculation. Schraudolph *et al.* [29] proposed a fast exponential calculation method based on floating-point number, and the mathematical formula of EXP macro operation is as shown in equation (5).

$$e^{x+y} = 2^k(1 + x/\ln(2) - k) \text{ among } k = \lfloor x/\ln(2) \rfloor \quad (5)$$

$y \equiv c \cdot \ln(2)/2^{20}$, where $\lfloor u \rfloor$ represents the largest integer less than or equal to u, c is an optimal constant obtained after-multiple calculations. We use simple displacement bits and integer algebra operations to achieve exponential fast approximation based on the representation of IEEE-754 floating-point number calculation [30]. Floating point calculation is divided into single precision and double precision. The single-precision floating-point number is 32 bits, and the double-precision floating-point number is 64 bits, both contain the sign bit S, the exponent bit E, the mantissa bit M, and the exponential offset value Z. If the difference between single precision and double precision is ignored, the format of the floating-point number can be expressed as shown in the equation (6).

$$Y = (-1)^s \cdot (1 + M) \cdot 2^X, \quad X = E - Z \quad (6)$$

The value of Z of the single-precision is 127, and the value of Z of the double-precision is 1023. If the values of S and M both are 0, the equation will become equation $Y = 2^X$. If you just want to obtain the value of 2^x, you only need to store x in the position of the high 32 bits that belongs to the exponent bit. At this time, e^x can be reduced to a power-based operation form of base 2, as shown in equation (7).

$$e^x = 2^{x/\ln 2} = 2^{yH}, \quad H = x/\ln 2 + Z \quad (7)$$

where y represents the number of bits that can be moved. If $x/\ln 2$ is divisible, then H is stored in the exponent bit without causing loss of precision. Conversely, it will generate a decimal place. At this time, H is shifted to the left by y bits and the fractional part is stored in the mantissa M. The specific proof process is cumbersome and it has been stated in the literature [29], we will no further proof here. The literature [29] also proves that the running speed on each machine with the EXP floating-point calculation method is faster than the standard math library, the results as shown in Fig. 5.

The expansion of $2^{x/\ln 2}$ is the same as the expansion of e^x, however, due to the use of the idea of floating-point calculations, the new function not only can accelerate the exponential calculation, but also achieve the purpose of quickly approximating e^x. Inspired by this research, it can be considered that $2^{x/\ln 2}$ is an exponential function that can achieve the fast exponential calculation. Therefore, the value of e^x of the activation function ELU on the negative part is optimized to $2^{x/\ln 2}$, generating a new activation function in this paper,

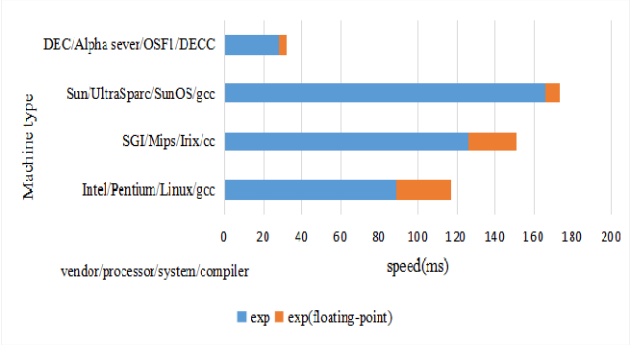


FIGURE 5. Comparison of the speed of the EXP standard function and floating point calculation on each machine.

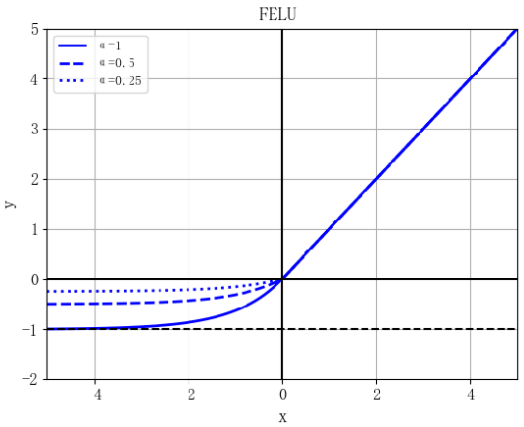


FIGURE 6. FELU function.

in this way, it can achieve the same as the classification accuracy of ELU and achieve the purpose of accelerating calculation.

B. IMPROVED ACTIVATION FUNCTION FELU

In order to speed up the calculation of the exponential function, according to the fast exponential function obtained in Section 3.1, the mathematical form of the new activation function is shown in Equation (8), and its curve is shown in Fig. 6.

$$f(x) = \begin{cases} x & x > 0 \\ \alpha(2^{x/\ln(2)} - 1) & x \leq 0 \end{cases} \quad (8)$$

Due to achieving the fast approximation, the curve trend is basically consistent with the ELU, which ensures that the new activation function does not change the accuracy advantage of the ELU.

It can be seen from Fig. 6 that the function of FELU introduces a new exponential function in the negative part. The curve of this function is smooth respect to the linear function, and it has the advantage of being differentiable everywhere and output zero mean, which can effectively avoid the occurrence of output offset. So you can find the direction in which the gradient drops faster. At the same

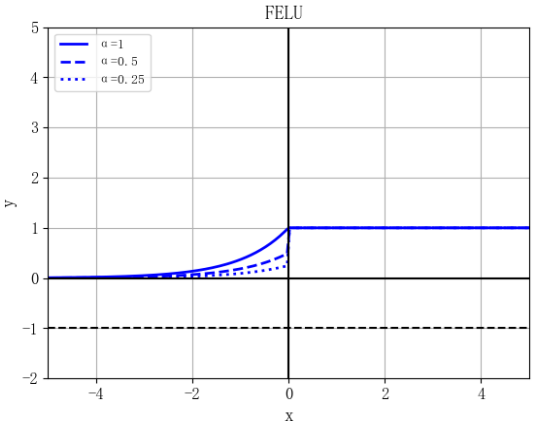


FIGURE 7. The derivative curve of FELU.

time, the parameterα is a variable parameter learned through training, which is used to control the soft saturation region of the negative part to determine the activation degree of the feature in the negative region. Fig. 6 shows the curve of the exponential function corresponding to the case that the value of α is 1, 0.5, 0.25 in the negative part, respectively. It can be seen from the figure that the smaller the activation function is, the closer it is to the ReLU function, and the more likely the neuron death problem occurs, on the contrary, the function is closer to the linear function near the origin, which can speed up the convergence of the network. In order to alleviate the phenomenon of neuronal death and activate more features that fall in the negative part, it is necessary to find a suitable value of α during training.

The derivative of Equation (8) is shown in Equation (9).

$$\frac{df(x)}{dx} = \begin{cases} 1 & x > 0 \\ \alpha \cdot 2^{x/\ln 2} & x \leq 0 \end{cases} \tag{9}$$

Its curve is shown in Fig. 7.

It can be seen from Figure 7 that when the input is greater than zero, the derivative of the function is always equal to one, so that the gradient does not have an attenuation problem, and it can also effectively alleviate the occurrence of vanishing gradient. We introduce a fast exponential function on the negative part, and retain a partial gradient, then the input can be effectively activated after entering the negative region, and the feature information of part of the image can be retained. Gradient calculation is the calculation process of the network in the back propagation. The negative part continuously passes the information obtained by the gradient calculation to the previous layer, and updates the information of the gradient. Therefore, the larger the negative saturation region is, the more the eigenvalues are activated, the more gradient information is forwarded, and the gradient update is more accurate, so we can more effectively alleviate the occurrence of neuron death. When the value of α is 1, it can not only maximize the negative soft-saturated region to activate more negative features, but also make the activation function linearly at the origin to accelerate network convergence, so we think that the value of α is 1 is an optimal value to achieve the optimal result in theory. In the negative part, the output value of the function is kept between [0, 1), which can reduce the influence of the output information of the previous layer on the output information of the next layer, and has good noise robustness to improve the classification accuracy.

It can be seen that FELU can effectively solve the problem of output offset, enhance the noise robustness, speed up the network calculation, improve the classification accuracy, and reduce the network running time.

TABLE 1. Experimental parameter settings.

Learning Rate (base_lr)	0.01
Learning strategy (lr_policy)	inv
Training iterations (max_iter)	40000
Training batch (base_size)	64
Testing iterations (test_iter)	500
Testing batch (base_size)	100
weight_decay	0.0005
Optimization type (type)	SGD

IV. EXPERIMENT AND ANALYSIS

A. EXPERIMENT SETTINGS

This paper uses the caffe framework environment to complete experiments on a Linux machine with Centos Linux release 7.4, NVIDIA Tesla-PCIE-16GB. The test content selects the traffic identification of the current research hotspot image classification field. In this paper, the traditional ReLU [14], ELU [18], SLU [19], MPELU [20], TReLU [21] and the FELU function proposed in this paper are trained and verified on the authoritative data sets such as cifar10, cifar100 and GTSRB. Due to the different activation functions and data sets, there will be differences in network runtime and classification accuracy. The parameter settings after the optimization obtained in this paper are shown in Table 1.

B. NETWORK STRUCTURE

In this paper, we uses a simple neural network that is not easy to cause information loss as our experimental network, which consists of two convolutional layers, two pooling layers and two fully connected layers. Each convolution layer uses a filter which size is 5*5. The first layer of convolutional layer outputs 20 channels, the second layer outputs 50 channels, their strides are 1. The pooling layer uses a filter which size is 2*2, their strides are 2. The first fully connected layer outputs 500 channels, and the second fully connected layer acts as a classifier, which select the corresponding number of output channels according to the data set category (10/100/43). The network structure and parameter settings are shown in Fig. 8.

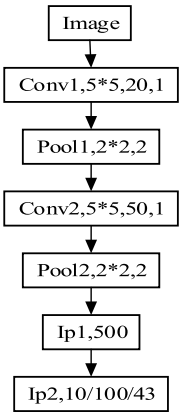


FIGURE 8. Network structure and parameter settings.

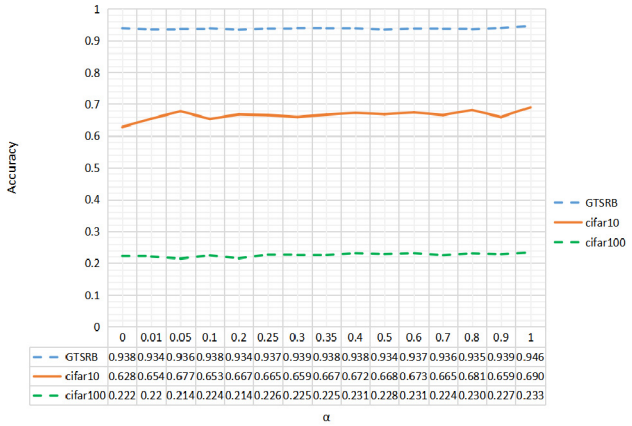


FIGURE 9. The α experimental results on three data sets.

C. PARAMETER α EXPERIMENT AND ANALYSIS

In this paper, we use 15 parameters to test the parameter α, and the 15 parameters are set to 0, 0.01, 0.05, 0.1, 0.2, 0.25, 0.3, 0.35, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, respectively. We use the network structure and training test parameters of this paper to conduct experiments on the data sets of cifar10, cifar100, and GTSRB, and select the optimal value of α that can get better results on three sets of data sets. The experimental results are shown in Fig. 9.

The experimental results show that when the parameter α is 1, the accuracy of the FELU is the highest, which is better than the accuracy of the ReLU under the same data set (when α is 0), no matter which set of data sets. The optimal value of α obtained by experiment is consistent with the optimal α value obtained in theory. Therefore, we used the FELU function with the value of parameter α of 1 to compare with other activation functions in the next experiment.

D. ACTIVATION FUNCTION EXPERIMENTAL ANALYSIS

In this paper, we compare and analyze different activation functions on the cifar10, cifar100 and GTSRB data sets.

TABLE 2. The experiment results of training cifar10 data set.

Activation Function	Accuracy	The running time of Network (ms)	The time of forward propagation (ms)	The time of backward propagation (ms)
ReLU[15]	0.6280	568.28	0.538604	0.472928
ELU[19]	0.6464	738.96	0.733771	0.474207
SLU[20]	0.3888	863.90	0.892081	0.621134
MPELU[21]	0.6492	1483.61	1.174470	0.844734
TReLU[22]	0.6516	1069.39	0.744291	0.576131
FELU(paper)	0.6904	481.54	0.452731	0.408947

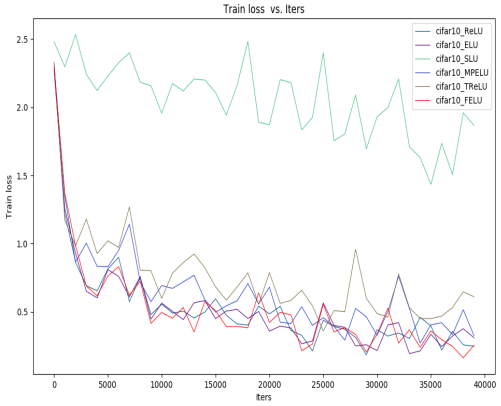


FIGURE 10. The training loss value decline curve of cifar10 data set.

1) CIFAR10 DATA SETS

The cifar10 dataset consists of 60,000 color images, the size of each image is 32*32 and it has 10 categories in total. Of these, 50,000 were used for training and 10,000 were used for testing. The results of the network training on cifar10 data set in this paper are shown in Table 2.

The experimental results show that FELU has better classification accuracy on cifar10 data set than the other activation functions under the same environment and parameter settings, and FELU is faster than other activation functions in time of network running and gradient calculation. The curve of classification accuracy of the test set and the curve of loss function of the training are shown in Fig. 10 and Fig. 11, respectively.

It can be seen from the curve of the accuracy and the loss function that the classification accuracy of FELU on cifar10 data set is better than other activation functions after 40K iterations, and the decline degree of loss value in training is also the largest.

2) CIFAR100 DATA SETS

The cifar100 data set is similar to the cifar10 data set, but the cifar100 data set has 100 categories, and each category has 500 training images and 100 test images. Its results of the network training on cifar100 data set in this paper are shown in Table 3.

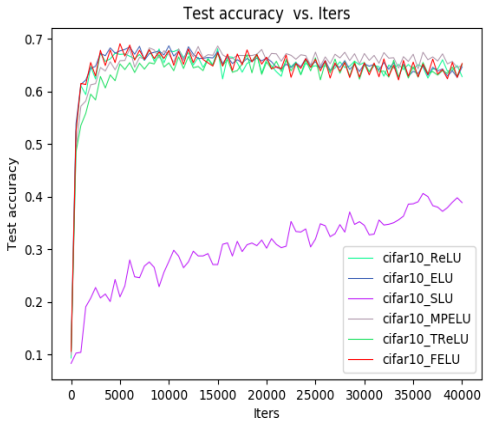


FIGURE 11. The accuracy curve of the cifar10 test data set.

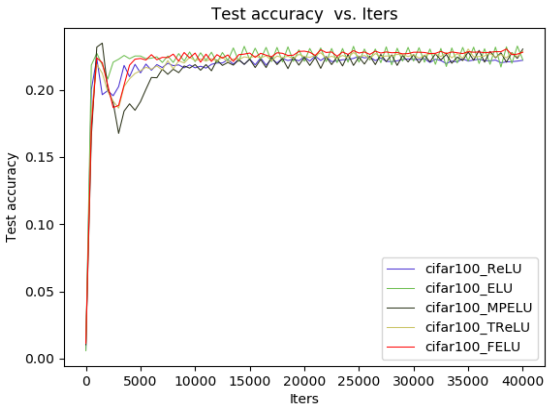


FIGURE 12. The accuracy curve of the cifar100 test data set.

TABLE 3. The experiment results of cifar100 data set.

Activation Function	Accuracy	The running time of Network (ms)	The time of forward propagation (ms)	The time of backward propagation (ms)
ReLU[15]	0.2220	545.40	0.534817	0.413282
ELU[19]	0.2224	770.82	0.614922	0.524191
SLU[20]	-	-	-	-
MPELU[21]	0.2304	1064.66	0.993751	0.713898
TReLU[22]	0.2288	953.62	0.847064	0.600892
FELU(paper)	0.2338	444.93	0.42723	0.373044

The experimental results show that the classification accuracy of FELU on the cifar100 data set is better than the other activation functions. The classification result of FELU is about 1% higher than that of ReLU, ELU and TReLU, and is closer to MPELU. Whether it is in the time of network running or the time of gradient calculation, FELU is faster than other activation functions, especially faster than MPELU. The curve of classification accuracy of test set and the curve of loss function in training are shown in Fig. 12 and Fig. 13, respectively.

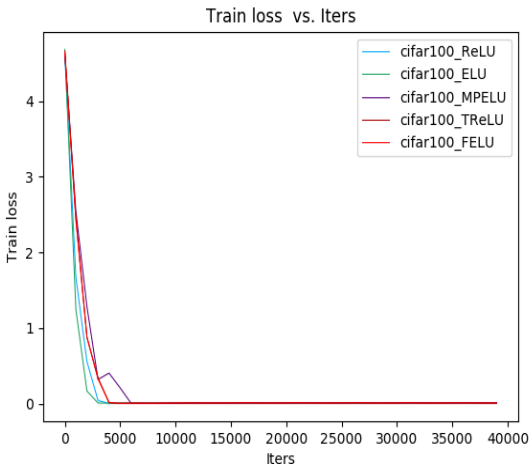


FIGURE 13. The training loss value decline curve of cifar100 data set.

TABLE 4. The experiment results of GTSRB data set.

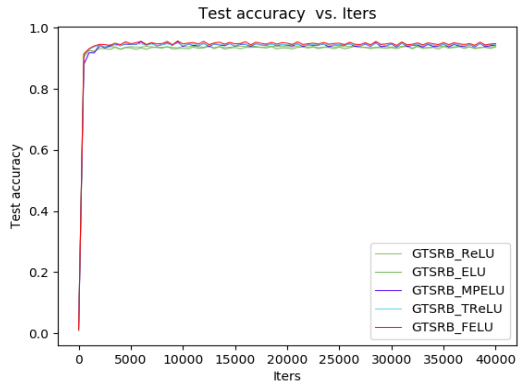
Activation Function	Accuracy	The running time of Network (ms)	The time of forward propagation (ms)	The time of backward propagation (ms)
ReLU[15]	0.9388	625.13	0.429670	0.391889
ELU[19]	0.9374	702.778	0.638701	0.611220
SLU[20]	-	-	-	-
MPELU[21]	0.9438	1106.11	0.882970	0.870629
TReLU[22]	0.9456	920.12	0.818195	0.698743
FELU(paper)	0.9572	507.96	0.522658	0.408434

It can be seen from the curve of accuracy and the loss function that the classification accuracy of FELU on the cifar100 data set increases with the number of iterations, and reaches the optimal after 40K iteration. The loss value in training decreases by a similar magnitude to the TReLU. The classification accuracy of ELU is undulating with increasing number of iterations and it is lower than FELU accuracy after 40K iteration. The classification accuracy of MPELU is similar to that of FELU, but its speed is slower more than 2 times.

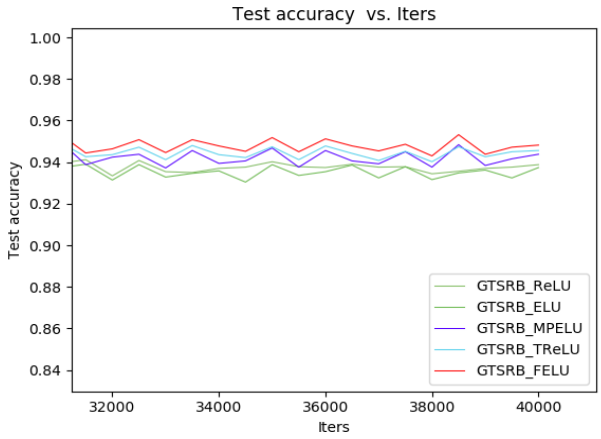
3) GTSRB DATA SETS

The GTSRB data set is a traffic sign data set based on the German standard. There are 43 categories, including 39,209 training images and 12,630 test images. The sample images contain traffic signs and surrounding areas, ranging in size from 15x15 to 250x250. The results of the network training on GTSRB data set in this paper are shown in Table 4.

The experimental results show that FELU has the highest classification accuracy rate on the GTSRB data set, which is about 2% higher than RELU and ELU, and about 1% higher than MPELU and TReLU. The FELU function is the fastest regardless of the time of the network running or computing. Experiments show that this activation function can improve the speed of the network while effectively improving the accuracy of classification. The curve of classification



(a) Global map of accuracy curve



(b) Partial enlargement of the accuracy curve

FIGURE 14. The accuracy curve of the GTSRB test data set.

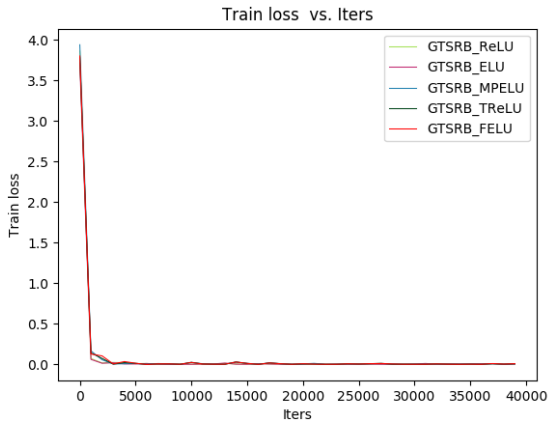


FIGURE 15. The training loss value decline curve of GTSRB data set.

accuracy of the test set is shown in Fig. 14, where (a) shows the global curve of the accuracy, (b) shows the partial enlargement curve of the accuracy. The curve of loss function in training is shown in Fig. 15.

It can be seen from the curve of classification accuracy in testing and the loss decline in training that the classification accuracy of FELU proposed in this paper on the GTSRB data set has been at the leading position with the increase of the number of iterations, and it is consistent with other activation functions in terms of loss value convergence.

TABLE 5. The experiment results of GTSRB data set.

Network	Activation Function	Accuracy	The running time of Network (ms)	The time of forward propagation (ms)	The time of backward propagation (ms)
VGG16	ReLU[15]	0.9366	286.79	0.2901	0.2485
	ELU[19]	0.9372	290.14	0.3904	0.2583
	FELU(paper)	0.9444	286.07	0.2906	0.2462
ResNet 50	ReLU[15]	0.9208	15856.5	15.8191	16.0356
	ELU[19]	0.9424	16138.6	15.9285	16.2907
	FELU(paper)	0.9557	15675.2	15.8505	15.7677

The above experimental results show that the FELU activation function is significantly faster than other activation functions in terms of network running time and computing time, even more than twice as fast as MPELU. In terms of classification accuracy, the FELU activation function has a significant improvement over the other activation functions on the three sets of data sets. Therefore, the experiment proves that FELU proposed in this paper can achieve faster running speed and better classification accuracy when the value of parameter α is 1.

E. THE ANALYSIS ON COMPLEX NETWORK APPLICATION

In order to effectively verify the advantages of the activation function of this paper, the VGG16 and ResNet50 networks are used to train the data set GTSRB, and the classic ReLU function, ELU function and FELU function are compared. The experimental results are shown in Table 5.

The experimental results show that the activation function FELU proposed in this paper also has good experimental results in the complex networks VGG16 and ResNet50. It has faster speed and higher accuracy. Therefore, FELU can be applied to both simple networks and complex networks.

V. CONCLUSION

This paper proposes a new activation function named FELU for improving ELU function to deal with the slow speed of calculation. By using the fast exponential function on the negative part, it effectively alleviates the problems of neuron death and output offset, and effectively reduces the time of network operation and calculation. The experimental results show that the new activation function can achieve the fastest running time and the best classification accuracy on all three data sets, and the new activation function also achieves better results in other network structures (such as VGG, ResNet, etc.). So FELU proposed in this paper has certain practical value in deep convolutional neural networks, and we will plan to further improve the real-time performance in future research.

REFERENCES

[1] A. Krizhevsky, I. Sutskever, and G. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. NIPS*. Curran Associates Inc., 2012.

[2] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
[3] M. Rastegari, V. Ordonez, J. Redmon, and A. Farhadi, "XNOR-NET: ImageNet classification using binary convolutional neural networks," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 525–542.
[4] Y. Tan, P. Tang, Y. Zhou, W. Luo, Y. Kang, and G. Li, "Photograph aesthetical evaluation and classification with deep convolutional neural networks," *Neurocomputing*, vol. 228, pp. 165–175, Mar. 2017.
[5] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. 13th Int. Conf. Artif. Intell. Statist. (AISTATS)*, 2010, pp. 249–256.
[6] Y. Zhu, C. Zhang, D. Zhou, X. Wang, X. Bai, and W. Liu, "Traffic sign detection and recognition using fully convolutional network guided proposals," *Neurocomputing*, vol. 214, pp. 758–766, Nov. 2016.
[7] T. Yang, X. Long, A. K. Sangaiah, Z. Zheng, and C. Tong, "Deep detection network for real-life traffic sign in vehicular networks," *Comput. Netw.*, vol. 136, no. 2018, pp. 95–104.
[8] J. Yu, B. Zhang, Z. Kuang, D. Lin, and J. Fan, "iPrivacy: Image privacy protection by identifying sensitive objects via deep multi-task learning," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 5, pp. 1005–1016, May 2016.
[9] Y. Pang, J. Cao, and X. Li, "Learning sampling distributions for efficient object detection," *IEEE Trans. Cybern.*, vol. 47, no. 1, pp. 117–129, Jan. 2017.
[10] J. Yu, X. Yang, F. Gao, and D. Tao, "Deep multimodal distance metric learning using click constraints for image ranking," *IEEE Trans. Cybern.*, vol. 47, no. 12, pp. 4014–4024, Dec. 2017.
[11] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3431–3440.
[12] A. Iliev, N. Kyurkchiev, and S. Markov, "On the Approximation of the step function by some sigmoid functions," *Math. Comput. Simul.*, vol. 133, pp. 223–234, Mar. 2017.
[13] A. Hamidoğlu, "On general form of the Tanh method and its application to nonlinear partial differential equations," *Numer. Algebra, Control Optim.*, vol. 6, no. 2, pp. 175–181, 2016.
[14] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proc. 27Th Int. Conf. Mach. Learn. (ICML)*, 2010, pp. 1–8.
[15] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1026–1034.
[16] B. Xu, N. Wang, T. Chen, and M. Li, "Empirical evaluation of rectified activations in convolutional network," Nov. 2015, *arXiv:1505.00853*. [Online]. Available: https://arxiv.org/abs/1505.00853
[17] M. Anthimopoulos, S. Christodoulidis, L. Ebner, A. Christe, and S. Mougiakakou, "Lung pattern classification for interstitial lung diseases using a deep convolutional neural network," *IEEE Trans. Med. Imag.*, vol. 35, no. 5, pp. 1207–1216, May 2016.
[18] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and accurate deep network learning by exponential linear units (ELUs)," Nov. 2015, *arXiv:1511.07289*. [Online]. Available: https://arxiv.org/abs/1511.07289
[19] Z. Huizhen, L. Fuxian, and L. Longyue, "A novel softplus linear unit for deep CNN," *J. Harbin Inst. Technol.*, vol. 50, no. 4, pp. 117–123, 2018.
[20] Y. Li, C. Fan, Y. Li, Q. Wu, and Y. Ming, "Improving deep neural network with multiple parametric exponential linear units," *Neurocomputing*, vol. 301, pp. 11–24, Aug. 2018.
[21] Z. Tao, Y. Jian, S. Wen-Ai, and S. Chao-Feng, "Research on improved activation function TReLU," *J. Chin. Comput. Syst.*, vol. 40, no. 1, pp. 58–63, 2019.
[22] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. 30th Int. Conf. Mach. Learn. (ICML)*, vol. 30, 2013, pp. 1–6.
[23] F. Godin, J. Degrave, J. Dambre, and W. De Neve, "Dual rectified linear units (DReLUs): A replacement for tanh activation functions in quasi-recurrent neural networks," *Pattern Recognit. Lett.*, vol. 116, pp. 8–14, Dec. 2018.
[24] S. Qian, H. Liu, C. Liu, S. Wu, and H. S. Wong, "Adaptive activation functions in convolutional neural networks," *Neurocomputing*, vol. 272, pp. 204–212, Jan. 2018.
[25] H. Zhao, F. Liu, L. Li, and C. Lu, "A novel softplus linear unit for deep convolutional neural networks," *Appl. Intell.*, vol. 48, no. 7, pp. 1707–1720, Jul. 2018.
[26] G. Lin and W. Shen, "Research on convolutional neural network based on improved Relu piecewise activation function," *Procedia Comput. Sci.*, vol. 131, pp. 977–984, Jan. 2018.
[27] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proc. 14th Int. Conf. Artif. Intell. Statist.*, 2011, pp. 315–323.
[28] S.-I. Amari, "Natural gradient works efficiently in learning," *Neural Comput.*, vol. 10, no. 2, pp. 251–276, 1998.
[29] N. N. Schraudolph, "A fast, compact approximation of the exponential function," *Neural Comput.*, vol. 11, no. 4, pp. 853–862, 1999.
[30] F. Perini and R. D. Reitz, "Fast approximations of exponential and logarithm functions combined with efficient storage/retrieval for combustion kinetics calculations," *Combustion Flame*, vol. 194, pp. 37–51, Aug. 2018.

ZHENG QIUMEI received the B.S. degree from the East China Petroleum Institute, China, in 1986, and the M.E. degree from the China University of Petroleum (East China), China, in 1999. From 1995 to 1998, she held an academic visiting position with Houston University. She is currently a Professor with the College of Computer Science and Technology, China University of Petroleum (East China). Her research interests include computer vision, image processing, and deep learning.

TAN DAN received the B.S. degree from Qufu Normal University, China, in 2017. She is currently pursuing the M.E. degree with the China University of Petroleum (East China), Qingdao, China. Her research interests include artificial intelligence and image processing.

WANG FENGHUA received the B.S. degree from the Shandong University of Science and Technology, China, in 2002, and the Ph.D. degree from Xi'an Jiaotong University, in 2009. In 2014, he held an academic visiting position with The University of Western Australia. He is currently a Researcher with the College of Computer Science and Technology, China University of Petroleum (East China). His research interests include computer vision, pattern recognition, and biometrics.