Hindawi
Security and Communication Networks
Volume 2023, Article ID 9761609, 1 page
https://doi.org/10.1155/2023/9761609

Retraction

Retracted: Improving Convolutional Neural Networks with Competitive Activation Function

Security and Communication Networks

Received 26 December 2023; Accepted 26 December 2023; Published 29 December 2023

This article has been retracted by Hindawi, as publisher, following an investigation undertaken by the publisher [1]. This investigation has uncovered evidence of systematic manipulation of the publication and peer-review process. We cannot, therefore, vouch for the reliability or integrity of this article.

Please note that this notice is intended solely to alert readers that the peer-review process of this article has been compromised.

Wiley and Hindawi regret that the usual quality checks did not identify these issues before publication and have since put additional measures in place to safeguard research integrity.

We wish to credit our Research Integrity and Research Publishing teams and anonymous and named external researchers and research integrity experts for contributing to this investigation.

The corresponding author, as the representative of all authors, has been given the opportunity to register their agreement or disagreement to this retraction. We have kept a record of any response received.

References

[1] Y. Ying, N. Zhang, P. He, and S. Peng, "Improving Convolutional Neural Networks with Competitive Activation Function," *Security and Communication Networks*, vol. 2021, Article ID 1933490, 9 pages, 2021.

Hindawi
Security and Communication Networks
Volume 2021, Article ID 1933490, 9 pages
https://doi.org/10.1155/2021/1933490

Research Article

Improving Convolutional Neural Networks with Competitive Activation Function

Yao Ying[1], **Nengbo Zhang**[2], **Ping He**[3,4] and **Silong Peng**[5]

[1]*College of Information Science and Engineering, Northeastern University, Shenyang 110819, China*
[2]*College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China*
[3]*School of Intelligent Systems Science and Engineering, Jinan University, Guangzhou, Guangdong 519070, China*
[4]*Artificial Intelligence Key Laboratory of Sichuan Province, Sichuan University of Science and Engineering, Zigong, Sichuan 643000, China*
[5]*Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China*

Correspondence should be addressed to Ping He; pinghecn@qq.com

Received 9 April 2021; Revised 23 April 2021; Accepted 1 May 2021; Published 13 May 2021

Academic Editor: Chi-Hua Chen

The activation function is the basic component of the convolutional neural network (CNN), which provides the nonlinear transformation capability required by the network. Many activation functions make the original input compete with different linear or nonlinear mapping terms to obtain different nonlinear transformation capabilities. Until recently, the original input of funnel activation (FReLU) competed with the spatial conditions, so FReLU not only has the ability of nonlinear transformation but also has the ability of pixelwise modeling. We summarize the competition mechanism in the activation function and then propose a novel activation function design template: competitive activation function (CAF), which promotes competition among different elements. CAF generalizes all activation functions that use competition mechanisms. According to CAF, we propose a parametric funnel rectified exponential unit (PFREU). PFREU promotes competition among linear mapping, nonlinear mapping, and spatial conditions. We conduct experiments on four datasets of different sizes, and the experimental results of three classical convolutional neural networks proved the superiority of our method.

1. Introduction

Since the convolutional neural network (CNN) [1] was proposed, activation function has always been an important part of CNN. Traditional activation functions such as sigmoid and tanh bring the gradient vanishing problem, which makes the deep convolutional neural network (DCNN) difficult to optimize. Rectified linear unit (ReLU) [2, 3] alleviates the problem of vanishing gradient, which is one of the major factors in the recent renaissance of CNN [4].

Compared with the traditional activation functions, ReLU has a stable improvement but still has its own shortcomings. One of the main disadvantages of ReLU is the dead neurons. ReLU makes the original input compete with a constant term 0, thus obtaining the nonlinear transformation ability, resulting in some neurons being untrained during the whole training process. Many subsequent modifications were proposed to avoid the problem of neuron death during training. LReLU [5] and PReLU [6] make the original input compete with the linear mapping term to obtain the ability of nonlinear transformation while also solves the problem of neuron death. ELU [7] makes the nonlinear mapping term compete with the original input for nonlinear transformation capability. Maxout [8] enables multiple linear mapping terms to compete with each other for the capability of nonlinear transformation. The number of linear mapping terms participating in the competition in Maxout is not fixed, which depends on the demand. In order to further enhance the nonlinear transformation ability, Ramachandran et al. [9–11] propose a nonmonotonic activation function: Swish. Compared with the monotonous activation functions, Swish performs better in many tasks

and gradually replaces ReLU as the default activation function in CNN. Mish [12] is another monotonic activation function after Swish. Recently, funnel activation (FReLU) [13] makes the original input compete with the spatial condition term. The spatial condition is a simple spatial context feature extractor. After using this condition, FReLU not only has the nonlinear transformation ability like the previous activation functions but also has the pixelwise modeling capacity to grasp the context information. In summary, competition mechanisms are ubiquitous in the activation function, and the number and types of elements participating in the competition are not restricted.

In this paper, we summarize the competition mechanism in the activation function and propose a novel activation function design template: competitive activation function (CAF). CAF promotes competition among different elements. The number and types of competing elements in CAF are not fixed; they vary according to demand. CAF generalizes most of the current activation functions. Based on CAF, we propose a concrete instance: parametric funnel rectified exponential unit (PFREU). PFREU promotes competition among linear mapping, nonlinear mapping, and spatial conditions. We conduct experiments using Fashion-MNIST [14], CIFAR-10/100 [15], and Tiny ImageNet [16] datasets to evaluate the effectiveness of our method.

The rest of this paper contains the following sections. Section 2 presents related works. Section 3 describes our method. In Section 4, we detail our experimentations. Section 5 gives a detailed analysis of PFREU. In Section 6, we conclude this paper.

2. Related Work

2.1. Conventional Activation Functions. The activation function provides the nonlinear transformation capability required by CNN. As shown in Figure 1, the conventional activation function focuses on different nonlinear transformations. ReLU uses identity linear mapping in the positive quadrant, which alleviates the problem of gradient disappearance and makes it possible to train DCNN. However, the constant zero of the negative part of ReLU causes the problem of the zero gradient. Zero gradient will cause some neurons to fail to be trained during training. LReLU uses a small fixed slope value in the negative part of the ReLU to avoid the problem of the zero gradient. However, the performance of LReLU will be greatly affected by the predefined initial value of the slope. In order to avoid the predefined fixed slope values affecting the performance of CNN, PReLU makes the slope value learnable. Different from the above work, linear mapping is used in the negative quadrant of ReLU, while ELU uses nonlinear mapping in the negative quadrant of ReLU to avoid the problem of zero gradient. The exponential term in the negative quadrant of ELU makes the activation mean close to 0. This feature makes the gradient closer to the nature gradient [17] and also speeds up the learning process of the network. The exponential term is saturated on the negative part so that the ELU can learn a more robust and stable representation. Scaled exponential linear unit (SELU) [18] is a modification of ELU.

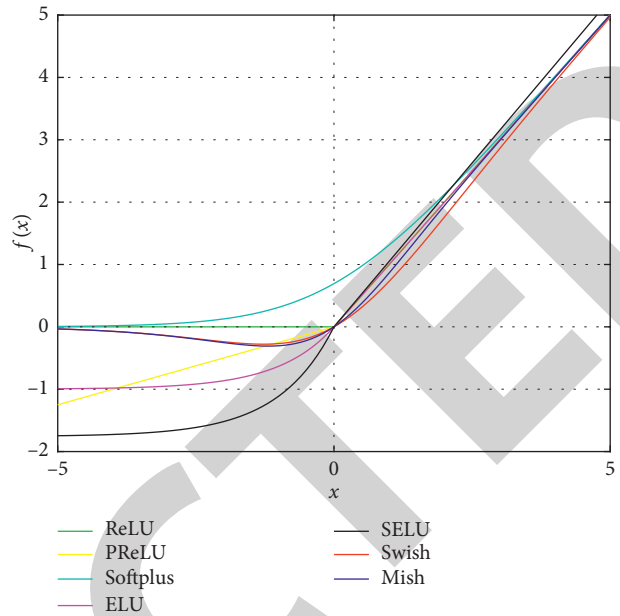

Figure 1: Conventional activation functions.

SELU induces self-normalizing properties to normalize its own output. Swish is a recently proposed nonmonotonic activation function, which has a stronger nonlinear transformation capability than the previous monotonic activation functions. As can be seen from Figure 1, Mish is another nonmonotonic activation function similar to Swish. Both Tanh and Softplus [19] are used in Mish, so the computational cost of Mish is higher than that of other activation functions. In short, conventional activation functions by itself bring nonlinear transformation capabilities to the neural network.

2.2. Context Conditional Activation Functions. Different from the conventional activation functions, the context conditional activation function brings contextual information into the activation function. As shown in Table 1, Maxout [8] expands the input to multiple branches and selects the maximum value in an element-wise way. Maxout generalizes ReLU and does not have the problem of dead neuron. Compared with ReLU in the classification task, Maxout shows a clear improvement. However, multiple branches significantly increase the number of parameters and computational cost. Probout [20] is a modification of Maxout. Probout uses a probabilistic sampling procedure to replace the maximum operation in Maxout to improve its invariance property. The model combined with Probout unit achieves competitive performance on multiple datasets. However, using Probout unit is computationally expensive when testing. FReLU is another activation function integrating the context information. Compared with the conventional activation functions, FReLU adds a spatial condition. The spatial condition provides pixelwise modeling capability, so FReLU can capture contextual information. Similar to the conventional activation functions, FReLU uses $\max(\cdot)$ to obtain the nonlinear transformation

Table 1: Common activation functions in a competitive manner.

Method	Definition
ReLU	$f(x) = \max(x, 0)$
LReLU	$f(x) = \max(x, 0.01x)$
PReLU	$f(x_i) = \max(x_i, p_i x_i)$
ELU	$f(x) = \max(x, \alpha(e^{-\lvert x\rvert} - 1))$
SELU	$f(x) = \lambda \begin{cases} x, & \text{if } x > 0, \\ \alpha e^x - \alpha, & \text{if } x \le 0 \end{cases}$
Softplus	$f(x) = \log(1 + e^x)$
Swish	$f(x) = x \cdot \text{sigmoid}(x)$
Mish	$f(x) = x \cdot \tanh(\text{softplus}(x))$
Maxout	$f(x) = \max_k(w_k^T x + b_k)$
FReLU	$f(x) = \max(x, T(x))$

capability. FReLU solves the long-standing spatial insensitivity problem in conventional activation functions and only increases the negligible computational cost. Although FRELU compared with the conventional activation functions has the above advantages, FReLU's nonlinear transformation ability is weaker.

2.3. Competition Mechanism in CNN. There are many competitive mechanisms in CNN. The widely used max-pooling operation is a typical case of the competition mechanism: the max value in the pooling region is selected. In Table 1, we write common activation functions in a competitive manner. Many activation functions contain competition mechanisms. ReLU makes the original input compete with constant 0 to obtain nonlinear transformation capability. LReLU and PReLU make the original input compete with a linear mapping to obtain nonlinear transformation capability. ELU makes the original input compete with nonlinear mapping to obtain nonlinear transformation capability. Unlike the previous activation functions, which had only two terms, Maxout makes multiple linear mapping terms compete with each other. The number of competing terms in Maxout is not fixed. Local winner take all (LWTA) [21, 22] is a work proposed at the same time as Maxout. The difference between LWTA and Maxout is that, in addition to the maximum output value, LWTA sets the remaining values to 0. FReLU makes the linear mapping term compete with the spatial condition term. Liao et al. [23] propose a novel CNN module that promotes competition among different sizes of convolutional filters. This module is used in the classic CNN models to produce state-of-the-art results on the most commonly used datasets.

3. Methodology

In this section, we first introduce the definition of CAF and then derive the PFREU.

3.1. Competitive Activation Function. As mentioned above, the competition mechanism is widely used in the activation functions. Most of the activation functions compete between two terms or multiple identical types of terms. We summarize the competition mechanism in the activation function and propose a novel activation function design template: CAF. The definition of CAF can be formulated as follows:

$$f(x) = \max(L(x), N(x), T(x), C, \ldots), \tag{1}$$

where $L(x)$ is the linear mapping term, $N(x)$ is the nonlinear mapping term, $T(x)$ is the spatial condition term, and C is a constant term. In equation (1), we can see four types of elements, but CAF does not limit the number or types of terms that participate in the competition. Adding elements beyond these four types to the CAF also complies with the definition of CAF. CAF generalizes all activation functions that use $\max(\cdot)$. We use CAF-α to denote CAF with α terms (it should be noted that the original input x (in equation (1)) is a special linear mapping multiplied by 1).

As we can see from Table 1, most activation functions can be seen as an instance of CAF-2. ReLU makes the linear mapping term compete with the constant term. LReLU and PReLU make competition between two linear mapping terms. ELU makes the linear mapping term compete with the nonlinear mapping term. FReLU makes the linear mapping term compete with the spatial condition term. Maxout can be seen as an instance of CAF-k. We believe that Maxout and FReLU are two representative CAF. Maxout shows that the number of elements participating in the competition is not fixed. FReLU indicates that the types of elements participating in the competition are not fixed, and new element types can be continuously added to enhance CAF. All in all, most current activation functions are constructed by competing among the four types of elements.

We propose a simplified version of CAF-3:

$$f(x) = \max(L(x), N(x), T(x)). \tag{2}$$

It is worth noting that CAF-3 represents all situations where the three elements compete with each other, and here is just one of them. For convenience, we call it CAF-3. It can be seen from Figure 2 that the biggest difference between CAF-3 and conventional activation functions is that CAF-3 adds additional spatial conditions, so CAF has the ability of pixelwise modeling to grasp contextual information. The main difference between CAF-3 and FReLU is that CAF-3 adds a nonlinear mapping term, so compared with FReLU, CAF-3 has a stronger nonlinear transformation capability. CAF-3 adopts a competition mechanism among linear mapping, nonlinear mapping, and spatial conditions to achieve a balance between nonlinear transformation capability and spatial information acquisition capability.

3.2. Parametric Funnel Rectified Exponential Unit. In this section, based on the concept of CAF-3, we propose a new type of activation function: PFREU. We present three variations of PFREU.

Figure 2: Graphical description of the corresponding activation function. (a) ReLU. (b) ELU. (c) PReLU. (d) FReLU. (e) CAF-3.

3.2.1. PFREU-A.

$$f(x_c) = \max\left(\alpha_c x_c, \beta_c x_c \cdot e^{-|x_c|}, T(x_c)\right), \tag{3}$$

$$T(x_c) = \mathrm{BN}(\mathrm{DWConv}(x_c)). \tag{4}$$

Here x_c is the input of PFREU-A on the cth channel, α_c is the coefficient that controls the linear mapping, and β_c is the coefficient that controls the nonlinear mapping. The subscript c in α_c and β_c indicates that we allow linear mapping and nonlinear mapping to vary on different channels. Function $T(\cdot)$ represents the spatial context feature

extractor. In equation (4), DWConv [24, 25] represents the depthwise separable convolutional layer, and BN [26] is an abbreviation for batch normalization operation. We use the Xavier [27] initialization strategy to initialization the depthwise separable convolutional layer. We set the initial values of α_c and β_c to be 1.

3.2.2. PFREU-B

$$f(x_c) = \max\left(\alpha_c x_c, x_c \cdot e^{-|\gamma_c x_c|}, T(x_c)\right). \quad (5)$$

Comparing equations (3) and (5), we can see that the difference between PFREU-B and PFREU-A lies in the nonlinear mapping term. In PFREU-A, the learnable parameter β_c is the coefficient of the exponent. In PFREU-B, the learnable parameter γ_c is a power exponent. The subscript c in γ_c indicates that we allow nonlinear mapping to vary on different channels. We set the initial values of α_c and γ_c to be 1.

3.2.3. PFREU-C

$$f(x_c) = \max\left(\alpha_c x_c, \beta_c x_c \cdot e^{-|\gamma_c x_c|}, T(x_c)\right). \quad (6)$$

PFREU-C combines the learnable parameters in PFREU-A and PFREU-B. Therefore, PFREU-C is the activation function with the most parameters among all PFREU variants. Considering that the weight decay tends to push the parameter values to 0, we do not use the weight decay for the learnable parameters in all PFREU variants. It should be noted that all activation functions that conform to equation (2) belong to CAF-3. We propose PFREU to verify the effectiveness of the CAF algorithm. PFREU is just one instance of the CAF-3 algorithm.

4. Experiments

To verify the effectiveness of our method. We use three CNN models to conduct experiments on four commonly used datasets. In order to exclude the situation where complex data expansion and parameter settings affect the final result, we only use conventional settings. For all models, we choose the Xavier [27] initialization strategy and the classification cross-entropy loss function. All of the results listed in this section are the median for five different tests.

4.1. Fashion-MNIST. Fashion-MNIST (F-MNIST) [14] is a fashion product dataset released by Zalando Research. In 10 classes, it contains 70000 images. These images are grayscale images, and the image size is 28×28 pixels. In the training set, it contains 60000 images. In the test set, it contains 10000 images. The training set and test set are evenly distributed in each class. The format of F-MNIST is the same as MNIST.

We use LeNet-5 [28] to evaluate the performance of different activation functions. For data preprocessing, we only divided the original images by 255. The batch size is set to be 128. We train the network for a total of 20 epochs. We use the cosine shape strategy [29, 30] to set the learning rate. The initial learning rate is 0.01. The weight decay and momentum are set to 0.0005 and 0.9, respectively.

The experimental results are shown in Table 2. Compared with ReLU, FReLU improves the accuracy from 90.34% to 90.98%. FReLU has a better performance compared with the previous nonlinear activation functions. The result shows that the added spatial condition enables FReLU to have the ability of pixelwise modeling, which is not available in the conventional activation functions, so that the performance of LeNet-5 with FReLU is better. All the results of PFREU variants are better than those of FReLU. We think this is because the added nonlinear mapping term makes PFREU have a stronger nonlinear transformation capability than FReLU. LeNet-5 with PFEU-C obtains the best result in all experiments.

4.2. CIFAR. CIFAR [15] is one of the most widely used color image datasets. CIFAR consists of two subsets: CIFAR-10 and CIFAR-100. CIFAR-10 consists of 10 classes. 50000 training images and 10000 testing images are equally distributed in each class. This dataset contains 32×32 pixel images. The size and format of CIFAR-100 are the same as CIFAR-10, but the number of classes has increased tenfold. Therefore, CIFAR-100 is a much more complex task than CIFAR-10.

We used the Network In Network (NIN) [31] and the Residual Network (ResNet) [32] to evaluate the performance of different activation functions. For the NIN model, we use simple data preprocessing: divide the image by 255 and then randomly flip it horizontally. The batch size is set to be 128. We train the network for a total of 200 epochs. The initial learning rate is set to 0.01 and divided by 2 at epoch 80 and then divided by 5 at epoch 140. The weight decay and momentum are set to 0.0001 and 0.9, respectively. For the experiment on ResNet-110, we use three epochs to warm up [33] and the other settings are the same to the original settings.

As shown in Table 3, the NIN model and the ResNet model with the Softplus always fail to converge. With SELU, these models can converge on CIFAR-10 but fail to converge on CIFAR-100. These activation functions with spatial condition behave differently on the NIN model and the ResNet model. In the NIN model, the performance of FReLU is much better than that of the conventional activation functions. We think this is because the spatial conditions are composed of convolutional layers and BN layers, while NIN is a relatively shallow network with only 9 layers. Using FReLU will significantly increase the number of layers of the NIN model. As we all know, depth [34–38] is very important for the expressive ability of CNN. Therefore, the NIN model with FReLU shows a clear advantage over the conventional activation functions. The ResNet model with 110 layers is much deeper than the NIN model. The layer added in FReLU occupies a much smaller proportion in the ResNet model. Therefore, the performance improvement is relatively small. In addition to the depth of the network, the characteristics of the ResNet model itself also affect the performance of the activation function with spatial

Table 2: Classification results (%) with LeNet-5 on Fashion-MNIST.

Method	Accuracy rate (%)
ReLU	90.34
LReLU	90.37
PReLU	90.43
ELU	90.80
SELU	90.84
Softplus	88.87
Swish	89.93
Mish	90.25
FReLU	90.98
PFREU-A	91.09
PFREU-B	91.04
PFREU-C	91.21

Table 3: Classification results (%) with NIN and ResNet-110 on CIFAR.

Model	NIN		ResNet-110	
Dataset	CIFAR-10	CIFAR-100	CIFAR-10	CIFAR-100
ReLU	86.93	59.84	92.64	67.27
LReLU	87.74	59.70	92.89	68.16
PReLU	88.34	62.71	91.88	69.05
ELU	88.18	61.48	92.58	69.31
SELU	*	63.10	*	66.36
Softplus	*	*	*	*
Swish	84.88	*	92.90	68.80
Mish	87.20	59.32	93.20	68.88
FReLU	90.79	67.36	92.98	68.91
PFREU-A	91.11	67.43	93.36	69.73
PFREU-B	91.24	67.67	93.38	70.34
PFREU-C	90.98	67.48	93.28	70.18

conditions. The biggest difference between ResNet and traditional CNN is the shortcut connection [38]. The shortcut connection transfers the shallow feature map directly to the deeper layer. This feature weakens the advantage of FReLU over the conventional activation functions to a certain extent. Therefore, for the ResNet model, the nonlinear transformation ability is more necessary than the pixelwise modeling ability. Therefore, on the ResNet model, the performance of FReLU is worse than that of some conventional activation functions with a stronger nonlinear conversion capability. In CIFAR-100, there are more conventional activation functions that perform better than FReLU than in CIFAR-10. We think this is because the number of classes of CIFAR-100 is 10 times that of CIFAR-10, so an activation function with the stronger nonlinear transformation capability is needed to distinguish different classes.

From Table 3, we can see that, in the NIN model, PFREU outperforms both the conventional activation functions and FReLU. In the ResNet model, PFREU is still superior to the conventional activation functions and FReLU. In the ResNet model, the gap between PFREU and FReLU is larger than that in the NIN model. As mentioned above, the ResNet model requires the nonlinear transformation capability more than the pixelwise modeling capability. The biggest difference between PFREU and FReLU is that PFREU has a nonlinear mapping term. So PFREU has strong nonlinear transformation ability than FReLU. From Table 3, we can see that PFREU has a good balance between nonlinear transformation capability and pixelwise modeling capability. On CIFAR-10, the performance of PFREU-A is better than that of PFREU-C. In contrast, the performance of PFREU-C is better than that of PFREU-A on CIFAR-100. We think the reason is that the CIFAR-10 dataset is a simple recognition task, while the CIFAR-100 dataset is much more complicated. PFREU-C tends to overfit on CIFAR-10. Among all PFREU variants, PFREU-B achieves the best results. We believe that this shows that more parameters do not mean better performance. The core issue of the activation function is design.

4.3. Tiny ImageNet. Tiny ImageNet [16] is a subset of the ImageNet [39] dataset. Tiny ImageNet consists of 200 classes. 100000 training images and 10000 validation images are equally distributed in each class. This dataset contains 64×64 pixel images. We use the ResNet-110 model used on CIFAR to evaluate the performance of different activation functions on Tiny ImageNet. In order to match the size of the image and model, we extend the stride of the first convolution layer to 2. Other settings are the same as CIFAR.

As we can see from Table 4, FReLU is better than most conventional activation functions. As mentioned in the previous section, ResNet weakens the advantages of FReLU over conventional activation functions. The number of classes of Tiny ImageNet is twice the number of classes of CIFAR-100. Therefore, the model needs an activation function with a stronger nonlinear transformation capability to distinguish different classes. The performance of all PFREU variants is better than that of FReLU, we think this shows that PFREU has a stronger nonlinear transformation capability than FReLU. The performance of PFREU and conventional activation functions is comparable. Among all PFREU variants, PFREU-B achieves the best result.

5. Analysis

In this section, we first analyze the two most important abilities in the activation function: nonlinear transformation ability and pixelwise modeling ability. Then, we explore the design factors that led to the performance difference among the PFREU variants. Finally, we analyze the parameter computation of PFREU.

5.1. Nonlinear Transformation vs. Pixelwise Modeling. Activation function is the source of the nonlinear transformation ability of the neural network. All activation functions have different degrees of nonlinear transformation capability. Recently, FReLU has introduced the ability of pixelwise modeling in the activation function. Two questions emerged naturally:

(1) Which ability is more important to the activation function?

Table 4: Classification results (%) with ResNet-110 on Tiny ImageNet.

Method	Accuracy rate (%)
ReLU	52.04
LReLU	51.25
PReLU	52.75
ELU	50.65
SELU	49.55
Softplus	49.91
Swish	51.41
Mish	51.40
FReLU	51.92
PFREU-A	52.27
PFREU-B	52.53
PFREU-C	52.43

(2) How to balance these two abilities in the activation function?

Experiments conducted on CIFAR and Tiny ImageNet provide some observations. Different models behave differently. As the previous analysis, ResNet has a more powerful spatial information acquisition capability. Therefore, compared with NIN, the activation function with spatial conditions has less impact on ResNet. It also depends on the specific task. CIFAR-10 and CIFAR-100 have different levels of difficulty. Therefore, on the CIFAR-100 dataset, the model needs an activation function with a stronger nonlinear transformation capability. Therefore, the network requires both nonlinear transformation capability and pixelwise modeling capability, which is more important depending on the specific model and task.

Conventional activation functions do not have the pixelwise modeling capability. Compared with the traditional activation functions, FReLU's nonlinear transformation ability is relatively weaker. PFREU adds a nonlinear mapping term to enhance the nonlinear transformation ability. The experimental results in Section 4 prove the effectiveness of PFREU. Therefore, our CAF-3 method can construct an activation function that balances nonlinear transformation and pixelwise modeling capabilities.

5.2. The Design of PFREU. The difference among the PFREU variants is the nonlinear mapping term. In particular, the number of parameters of the nonlinear term in PFREU-A and PFREU-B is also the same. We constructed two simple exponential functions to explore how different parameter positions affect the change of the exponential function [40–42].

$$f_1(x) = \alpha \cdot e^x, \quad f_2(x) = e^{\beta \cdot x}. \tag{7}$$

The difference between f_1 and f_2 is whether the learnable parameter is a coefficient or a power exponent.

It can be seen from Figure 3 that, under the same parameter amplitude change, the change of function f_2 is greater than that of f_1. Another major difference between f_1 and f_2 is that f_2 always passes through the origin. As shown in Figure 1, most conventional activation functions pass through the origin, so we think this feature helps improve performance.

We calculate the gradient of f_1 and f_2 with respect to input x:

$$\frac{\partial f_1(x)}{\partial x} = \alpha \cdot e^x, \tag{8}$$

$$\frac{\partial f_2(x)}{\partial x} = \beta \cdot e^{\beta \cdot x}. \tag{9}$$

From equations (8) and (9), we can see that if the learnable parameter is used as a power exponent, it will have a greater impact on the input during backpropagation. A previous study [43] has shown that the amplitude change of the learnable parameters in the activation function is very small during the training process. The learnable parameters as power exponents are more beneficial to network optimization than as coefficients. The experimental results in Section 4 show that the performance of PFREU-B is better than that of PFREU-A in most cases. Therefore, if a learnable parameter is added to the exponential term, we think it should be a power exponent.

5.3. Parameter Computation. We assume a convolutional network layer and the size of the input feature map is $C \times H \times W$, the convolution kernel receptive field is $K_h' \times K_w'$, and the size of the output feature map is $C \times H' \times W'$. The number of convolutional parameters is $CCK_h'K_w'$ and the FLOP (floating-point operation) is $CCK_h'K_w'HW$. Taking PFREU-A as an example, PFREU-A has three terms. We assume that the receptive field of the depthwise separable convolutional layer in the spatial condition is $K_h \times K_w$, the number of parameters for depthwise separable convolution is CK_hK_w, and the FLOP of depthwise separable convolution is CK_hK_wHW. The number of parameters of the linear mapping term is C, and the FLOP of the linear mapping term is CHW. The linear mapping term and the nonlinear mapping term have the same number of parameters. For simplification, we assume $K = K_h = K_w$ and $K' = K_h' = K_w'$.

So the parameter complexity of the convolutional layer is $O(C^2K'^2)$, and after adopting PFREU-A, the parameter complexity becomes $O(C^2K'^2 + CK^2 + 2C)$. The FLOP of the convolutional layer is $O(C^2K'^2HW)$, and after adopting PFREU-A, it becomes $O(C^2K'^2HW + CK^2HW + 2CHW)$. Since C is much larger than K, K', and 2, the additional complexity of PFREU can be negligible.

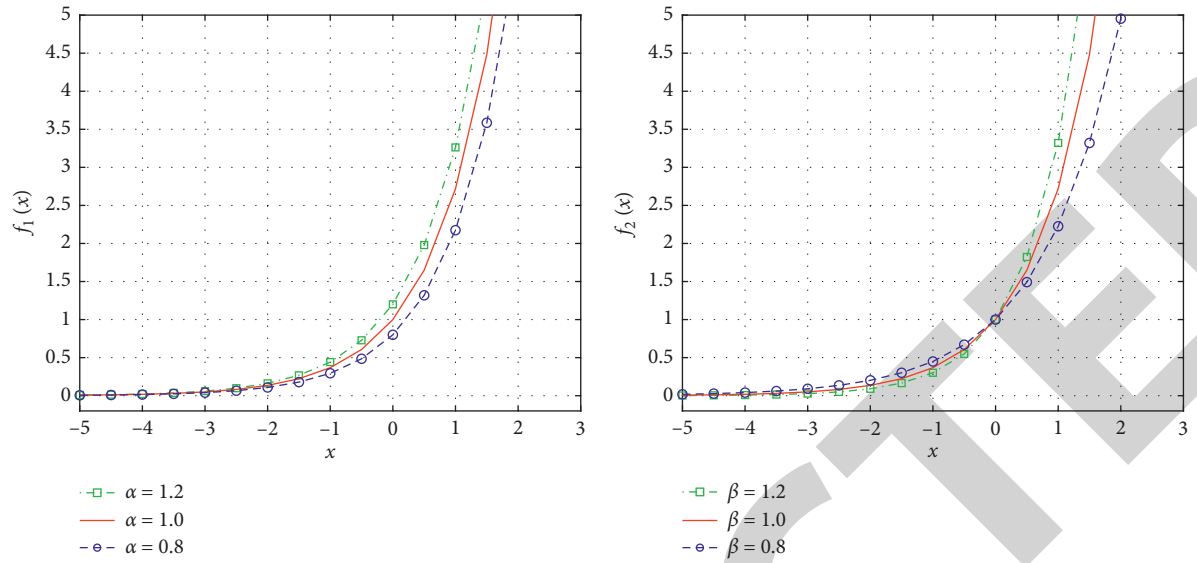

FIGURE 3: (a) The shape of f_1 with different α values. (b) The shape of f_2 with different β values.

6. Conclusion and Future Works

In this paper, we introduce an activation function design template: CAF. CAF summarizes all current activation functions that use $\max(\cdot)$ and provide a direction for future design of new competitive activation functions. In order to verify the effectiveness of CAF, we present an instance that conforms to CAF: PFREU. The performance of PFREU is better than that of other activation functions. Experimental results show that, based on the CAF-3 method, an activation function can be constructed that balances the nonlinear transformation capability and the pixelwise modeling capability. In the future, we will use the Neural Architecture Search technique to explore more activation functions that conform to the CAF template and design new modules to enhance the pixelwise modeling capability of CAF.

Data Availability

The data used to support the findings of this study are available online.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The authors thank Peng Shan, Feng Jia, Jianlin Su, Chunpeng Ma, Jinpeng Zhang, Yang Li, and Shenqi Lai for helpful discussions. This work was supported by the National Natural Science Foundation of China under Grant 11705122, Science and Technology Program of Sichuan under Grant 2020YFH0124, Guangdong Basic and Applied Basic Research Foundation (2021A1515011342), and Zigong Key Science and Technology Project of China under Grant 2020YGJC01.

References

[1] Y. LeCun, B. Boser, J. S. Denker et al., "Backpropagation applied to handwritten zip code recognition," *Neural Computation*, vol. 1, no. 4, pp. 541–551, 1989.

[2] V. Nair, E. Geoffrey, and Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proceedings of the 27th International Conference on Machine Learning (ICML)*, pp. 807–814, Haifa, Israel, June 2010.

[3] X. Glorot, B. Antoine, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics (AISTATS)*, vol. 15, pp. 315–323, Ft. Lauderdale, FL, USA, January 2011.

[4] A. Krizhevsky, I. Sutskever, E. Geoffrey, and Hinton, "ImageNet classification with deep convolutional neural networks," in *Proceedings of the Advances in Neural Information Processing Systems (NIPS)*, vol. 25, pp. 1097–1105, Lake Tahoe, Nevada, USA, January 2012.

[5] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proceedings of the 30th International Conference on Machine Learning Workshop on Deep Learning for Audio*, pp. 1–6, Atlanta, GA, USA, 2013.

[6] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: surpassing human-level performance on ImageNet classification," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 1026–1034, Santiago, Chile, Feburary 2015.

[7] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and accurate deep network learning by exponential linear units (ELUs)," in *Proceedings of the 4rd International Conference on Learning Representations (ICLR)*, pp. 1–14, San Juan, Puerto Rico, Argentina, January 2016.

[8] I. Goodfellow, D. Warde-Farley, M. Mirza, A. Courville, and Y. Bengio, "Maxout networks," in *Proceedings of the 30th International Conference on Machine Learning (ICML)*, vol. 28, pp. 1319–1327, Atlanta, GA, USA, Feburary 2013.

[9] P. Ramachandran, B. Zoph, and V. Le Quoc, "Searching for activation functions," in *Proceedings of the 6rd International*

Conference on Learning Representations Workshop (ICLRW), pp. 1–13, Vancouver, BC, Canada, Feburary 2018.

[10] D. Hendrycks and K. Gimpel, "Gaussian error linear units (GELUs)," 2016, https://arxiv.org/abs/1606.08415.

[11] S. Elfwing, E. Uchibe, and K. Doya, "Sigmoid-weighted linear units for neural network function approximation in reinforcement learning," *Neural Networks*, vol. 107, pp. 3–11, 2018.

[12] D. Misra, "Mish: a self regularized non-monotonic activation function," in *Proceedings of the 31th British Machine Vision Conference (BMVC)*, pp. 1–14, Manchester, UK, September 2020.

[13] N. Ma, X. Zhang, and J. Sun, "Funnel activation for visual recognition," in *Proceedings of the European Conference on Computer Vision (ECCV)*, vol. 12356, pp. 351–368, Glasgow, UK, November 2020.

[14] X. Han, K. Rasul, and V. Roland, "Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms," 2017, https://arxiv.org/abs/1708.07747.

[15] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from Tiny images," Master's thesis, University of Toronto, Toronto, Canada, 2009.

[16] H. Pouransari and S. Ghili, "Tiny imagenet visual recognition challenge," 2013, https://tinyimagenet.herokuapp.com.

[17] S. I. Amari, "Natural gradient works efficiently in learning," *Neural Computation*, vol. 10, no. 2, pp. 177–202, 1998.

[18] G. Klambauer, T. Unterthiner, A. Mayr, and S. Hochreiter, "Self-normalizing neural networks," in *Proceedings of the Advances in Neural Information Processing Systems (NIPS)*, vol. 30, pp. 971–980, Long Beach, CA, USA, December 2017.

[19] C. Dugas, Y. Bengio, F. Bélisle, C. Nadeau, and R. Garcia, "Incorporating second-order functional knowledge for better option pricing," in *Proceedings of the Advances in Neural Information Processing Systems (NIPS)*, Vancouver, BC, Canada, March 2001.

[20] J. Tobias Springenberg and M. Riedmiller, "Improving deep neural networks with probabilistic maxout units," in *Proceedings of the 1rd International Conference on Learning Representations Workshop (ICLRW)*, pp. 1–10, Banff, Canada, Feburary 2014.

[21] R. K. Srivastava, J. Masci, S. Kazerounian, F. Gomez, and J. Schmidhuber, "Compete to compute," in *Proceedings of the Advances in Neural Information Processing Systems (NIPS)*, vol. 26, pp. 2310–2318, Lake Tahoe, Nevada, USA, January 2013.

[22] R. K. Srivastava, J. Masci, F. Gomez, and J. Schmidhuber, "Understanding locally competitive networks," in *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*, pp. 1–11, San Diego, CA, USA, April 2015.

[23] Z. Liao and G. Carneiro, "A deep convolutional neural network module that promotes competition of multiple-size filters," *Pattern Recognition*, vol. 71, pp. 94–105, 2017.

[24] F. Chollet, "Xception: deep learning with depthwise separable convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1800–1807, Honolulu, Hawaii, USA, July 2017.

[25] A. G. Howard, M. Zhu, B. Chen et al., "MobileNets: efficient convolutional neural networks for mobile vision applications," 2017, https://arxiv.org/abs/1704.04861.

[26] S. Ioffe and C. Szegedy, "Batch normalization: accelerating deep network training by reducing internal covariate shift," in *Proceedings of The 32nd International Conference on Machine Learning (ICML)*, pp. 448–456, Lille, France, March 2015.

[27] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 249–256, Sardinia, Italy, January 2010.

[28] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," in *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, November 1998.

[29] I. Loshchilov and F. Hutter, "SGDR: stochastic gradient descent with warm restarts," in *Proceedings of the 4rd International Conference on Learning Representations (ICLR)*, pp. 1–16, San Juan, Puerto Rico, Argentina, November 2016.

[30] G. Huang, S. Liu, L. van der Maaten, K. Q. Weinberger, and Weinberger, "CondenseNet: an efficient DenseNet using learned group convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2752–2761, Salt Lake City, Utah, USA, June 2018.

[31] M. Lin, Q. Chen, and S. Yan, "Network in network," in *Proceedings of the 2rd International Conference on Learning Representations (ICLR)*, pp. 1–10, Banff, Canada, March 2014.

[32] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, Las Vegas, NV, USA, June 2016.

[33] P. Goyal, P. Dollár, B. Ross, Girshick et al., "Accurate, large minibatch SGD: training ImageNet in 1 hour," 2017, https://arxiv.org/abs/1706.02677.

[34] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*, pp. 1–12, San Diego, CA, USA, November 2015.

[35] C. Szegedy, W. Wei Liu, Y. Jia et al., "Going deeper with convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–9, Boston, MA, USA, January 2015.

[36] M. Bianchini and F. Scarselli, "On the complexity of neural network classifiers: a comparison between shallow and deep architectures," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 8, pp. 1553–1565, 2014.

[37] G. Huang, Y. Sun, Z. Liu, D. Sedra, and Q. Kilian, "Deep networks with stochastic depth," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 646–661, Amsterdam, Netherlands, Octomber 2016.

[38] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 630–645, Amsterdam, Netherlands, Octomber 2016.

[39] O. Russakovsky, J. Deng, H. Su et al., "ImageNet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[40] J. T. Barron, "Continuously differentiable exponential linear units," 2017, https://arxiv.org/abs/1704.07483.

[41] L. Trottier, P. Giguere, and B. Chaib-draa, "Parametric exponential linear unit for deep convolutional neural networks," in *Proceedings of the 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pp. 207–214, Cancun, Mexico, December 2017.

[42] Li Yang, C. Fan, Y. Li, Q. Wu, and M. Yue, "Improving deep neural network with multiple parametric exponential linear units," *Neurocomputing*, vol. 301, pp. 11–24, 2018.

[43] A. Gupta and R. Duggal, "P-TELU: parametric tan hyperbolic linear unit activation for deep neural networks," in *Proceedings of the IEEE International Conference on Computer Vision Workshops (ICCVW)*, pp. 974–978, Venice, Italy, Octomber 2017.