ISSN 2616-5775 Vol. 5, Issue 13: 69-76, DOI: 10.25236/AJCIS.2022.051311

Design of nonlinear segmentation activation functions for object detection

Wenxiao Wei[1], Jieyu Liu[1,*], Qiang Shen[1], Yajing Wang[2]

[1]College of Missile Engineering, Rocket Force University of Engineering, Xi'an, Shaanxi, 710025, China
[2]State Key Laboratory of Complex Electromagnetic Environment Effects on Electronics and Information System, Luoyang, 471003, China
[*]Corresponding author

Abstract: *The existing activation functions ReLU, Tanh, and Mish have problems such as "neuronal death", offset, and poor robustness. Aiming at these problems, the XExp activation function is proposed by combining the advantages of ReLU, Swish, and Mish functions, and the problem of negative half-axis neuronal death is optimized by using the nonlinearity of non-RELU family functions and the non-zero characteristics of negative half-axis functions, and the soft saturation of negative semi-axis is retained. By designing the position of the origin of the function, the problem of positive half-axis offset in the Swish and Mish functions are solved. In terms of convergence speed, the MNIST dataset achieved 93.87% training accuracy during the first batch training on the newly proposed activation function XExp function, which was more than 85% higher in convergence speed compared with the Relu function; In terms of model convergence stability, compared with the accuracy of the Relu function, the XExp function can still achieve 98.05% accuracy when the number of convolutional layers is increased to 25 layers. The two data sets of CIFAR-10 and CIFAR-100 verify their versatility and practicality in the field of object detection.*

Keywords: *deep learning; activation function; robustness; object detection*

1. Introduction

In 2016 Professor Bengio defined the activation function and stated that the activation function is a mapping that is derivable almost everywhere [1]. Convolutional neural networks activate certain data features by activation function mapping, and since the distribution of most data in the target detection algorithm is nonlinear, to increase the nonlinearity of the model and strengthen the learning ability of the network, the neural network will introduce the activation function after certain convolutional layers to change the linear computation of the neural network, so the selection of a suitable nonlinear function for the performance of the network model such as detection accuracy and detection speed, Therefore, choosing the right nonlinear function is important for the performance of the network model such as detection accuracy and detection speed.

The earliest widely used activation function is the "S" type activation function, commonly known as the Sigmoid function and Tanh function, both of which tend to have fixed values as they converge to infinity [2]. the problem of gradient dispersion after the derivatives are multiplied. The Tanh function, on the other hand, takes its value at the center point 0, which slightly alleviates the problem of gradient dispersion of the Sigmoid function due to the range of its derivatives at (0,1) [2], and converges faster than the Sigmoid function. However, the first-order derivatives of both converge to 0 at infinity, making the gradient response of the function close to 0. Therefore, the Rectified Linear Unit (ReLU) function [3] and its optimization function were proposed by researchers to overcome the saturation phenomenon appearing at both ends, but inevitably the newly proposed activation function still has some defects.

To solve the existing problems, the LeakyRelu function proposed by Dubey AK [4] and others, and the ELU activation function proposed by Clevert DA [5] and others were modified for the negative semiaxis of Relu, which effectively alleviated the problem of neuron "death"; however, the computational complexity of their modifications increased. Subsequently, many experts and scholars constructed new activation functions by fusing multiple functions. For example, the Relu-Softplus proposed by Qi Shi [6] and the Relu-Softsign activation function proposed by Hongxia Wang [7] have better results compared with a single function, taking the advantage of complementary approaches to improve performance, but there are still some problems in the training of network models, one is that

ISSN 2616-5775 Vol. 5, Issue 13: 69-76, DOI: 10.25236/AJCIS.2022.051311

there is still a great improvement in speed and accuracy, and the other is that the training difficulty of the model increases.

In this paper, we analyze the roles, advantages, and disadvantages of various types of activation functions in neural networks, improve the existing activation functions for their shortcomings, and propose new and improved activation functions. To verify the effectiveness of the improved activation functions, we also validate the accuracy, speed, and stability of these functions on public datasets such as MNIST, CIFAR-10, and CIFAR-100, and verify their generality using various underlying networks.

2. Design ideas for the activation function

The ReLU function is calculated as:

$$y = \begin{cases} x, & x > 0 \\ 0, & \text{Others} \end{cases} \tag{1}$$

The ReLU function is a popular activation function with simple calculation and good effect, and it is the first proposed segmental activation function [8]. It provides a good solution to the gradient disappearance problem of Sigmoid and Tanh activation functions. In the range of x>0, the value is a linear variation of x, which solves the problem of gradient disappearance of the function in the positive semi-axis interval [9]. However, the ReLU function takes a constant value of 0 in the negative semi-axis interval, which also leads to its first-order derivative taking a value of 0 at x < 0. Compared with the Sigmoid function and the Tanh function, it is in a hard saturation state in the negative semi-axis. This means that when the value is taken to the negative half-axis, the corresponding weight is 0 and the neuron is inactive so that the parameters cannot be updated and training cannot be performed, a state also known as "neuron death". The result of neuron death is irreversible, so neuron death must be avoided in the model training [10,11]. In addition to this, the ReLU function causes a shift in the parameter update during the backpropagation calculation of the function because the values of the negative half-axis are all 0, while the mean value of the function is constantly greater than 0. The center of the range of values is not 0. Therefore, it causes a shift in the parameter update during the backpropagation calculation of the function [12]. In backpropagation, suppose there are two eigenvalues s_1, s_2, and the output y is obtained after the weights w_1, w_2 in the neural network, and the final result L is obtained after softmax, and the bias derivative is calculated for W to update the model parameters, and the calculation process is:

$$\frac{\partial L}{\partial w_1} = \frac{\partial L}{\partial y} * \frac{\partial y}{\partial w_1} = \frac{\partial L}{\partial y} * s_1 \tag{2}$$

$$\frac{\partial L}{\partial w_2} = \frac{\partial L}{\partial y} * \frac{\partial y}{\partial w_2} = \frac{\partial L}{\partial y} * s_2 \tag{3}$$

Since the activation value of the ReLU function must be non-zero, the values of s_1 and s_2 must both be greater than 0. Then the gradient of the weights after the same sign of w_1, w_2 can only move in the direction of the first and third quadrants, and it is impossible to find the updated parameters along the direction of the fastest decreasing gradient.

To solve the problem of dead and irreversible neurons in the ReLU function, experts and scholars have studied and improved the ReLU function [13,14]. The non-ReLU family activation functions are similar to the Sigmoid and Tanh functions, which are non-segmented, while their derivative functions are continuous, discarding the soft saturation of the Sigmoid and Tanh functions in tending to positive and negative infinity, and retaining the advantages of the ReLU family functions. Among them, the most representative ones are the Swish function and the Mish function.

The Swish activation function [15] is an activation function proposed by Google in 2017, and its function expression is

$$y_{Swish} = x \cdot Sigmoid(\beta x) = \frac{x}{1+e^{-\beta x}} \tag{4}$$

Where the function image changes the slope with the change of parameter β. When β is 0, the Swish activation function is linear; when β tends to infinity, the Swish function is the ReLU function. Its function image and the image of the derivative function are shown in Figures 1 and 2. The Swish function can then be regarded as a smooth activation function with linear interpolation between the linear function and the ReLU function.

The expression of the Mish function [16] is

ISSN 2616-5775 Vol. 5, Issue 13: 69-76, DOI: 10.25236/AJCIS.2022.051311

$$y_{mish} = x \cdot Tanh(\ln(1 + e^{x})) \tag{5}$$

In contrast to the Swish function, the functional expressions of both can be unified in the form of $x \cdot f(x)$. This design of the activation function ensures that the distribution of the initial data is hardly changed in the positive half-axis part, while soft saturation on the negative half-axis produces a buffer to promote the mean convergence to zero [17,18]. Thus, the non-ReLU family of activation functions retains the advantages of the ReLU function and enhances the smoothness of the gradient change due to its continuous function [19]. However, both also have disadvantages. First, the non-ReLU family functions are designed with more complex activation function expressions to ensure the continuity of the function, so the complexity is greatly increased in the gradient calculation, which reduces the model training efficiency; second, the gradient of both is smaller at the input of 0, and the convergence speed of the network learning training is reduced [20]; in addition, the advantages of the ReLU family functions are not retained in the positive half-axis, which ensures that the positive half-axis slope is 1 and does not change the advantage of the initial data distribution [21,22]. Therefore, to better solve the current problems of activation functions and combine the advantages of ReLU family activation functions and non-ReLU family activation functions, this paper designs new activation functions to improve the efficiency and robustness of network training tests. The results are shown in figure 1.

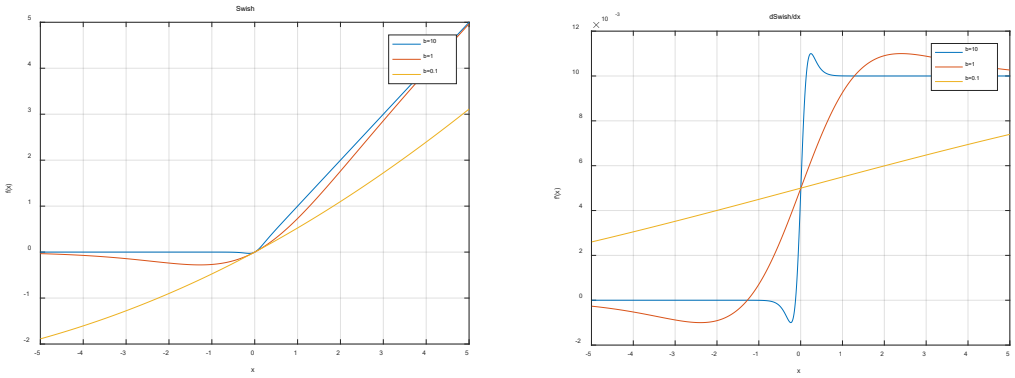

Figure 1: Swish function and First-order derivative of the Swish function

3. XExp activation function

Based on the analysis of the advantages and disadvantages of the activation function, this paper proposes a new and improved activation function, called the XExp function, whose expression is

$$y = \begin{cases} x, x > 0 \\ \dfrac{x}{e^{-x}}, x \le 0 \end{cases} \tag{6}$$

The XExp function achieves a global minimum of -0.3679 around x=-1.00. The image comparing the four activation functions, Swish, Mish, ReLU, and XExp, is shown in Figure 2, which shows the advantages and disadvantages of the four functions visually. Compared with the Swish and Mish functions, the XExp function has a larger rate of change around x=0, i.e., it has a larger gradient, which can provide a larger gradient backpropagation in the backward propagation process of the network model training, which can accelerate the optimization of the network parameters and achieve convergence faster; in addition, compared with these two functions, the XExp activation function takes the value of The same as the ReLU activation function, the positive semi-axis slope is 1, which ensures the initial input invariance of the input data and avoids the problem of network bias on the training and test sets. The results are shown in figure 2.

ISSN 2616-5775 Vol. 5, Issue 13: 69-76, DOI: 10.25236/AJCIS.2022.051311

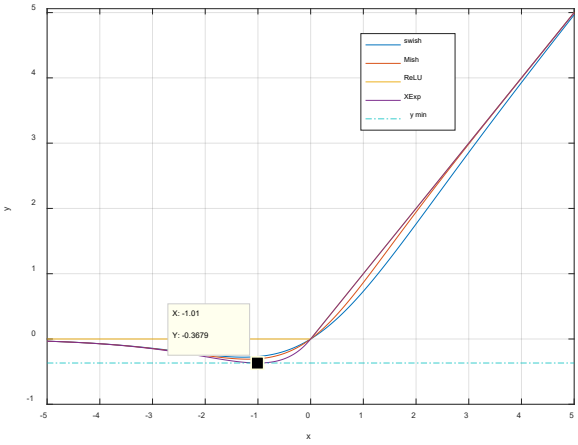

Figure 2: Images of the functions of Relu, Swish, Mish, XExp

The forward propagation process in the network detection target requires the calculation of the activation function, while the backward propagation part will involve the calculation of the first-order derivative function of the activation function, so the first-order derivative function and second-order derivative function of the activation function are analyzed and the graphs are plotted in Figure 3. it can be seen through the visualization in the figure that the XExp function avoids the problem of the disappearance of the negative semi axial gradient and the death of neurons compared to the ReLU function. Compared with the Swish function and Mish function both the first-order derivative function and the second-order derivative function have a larger gradient, which makes the network converge more efficiently in the backpropagation part of the parameter update optimization, and explains the reason for the fast convergence of the XExp activation function in terms of the underlying principle.

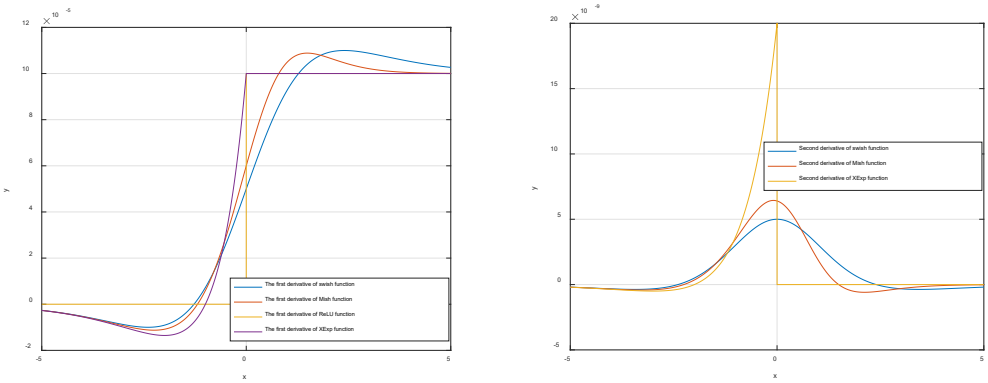

Figure 3: First-order and second derivatives of different activation functions

4. Experimental verification

To further validate the effectiveness of the proposed activation function XExp in deep convolutional networks in this chapter, experimental validation in terms of speed, robustness, and effectiveness is performed on several public image datasets in the following. The experiments are performed on MNIST, CIFAR-10, and CIFAR-100 public image datasets. The experiments were run under the Pytorch framework on Ubuntu 16.04 with CUDA 9.0 and cuDNN 7.1 to accelerate the training. The computer is equipped with a Corei7-8700k CPU, NVIDIA GTX1080Ti graphics card, and 32 G of RAM. all experiments are done with the same GPU, CPU, and other hardware configurations and software configurations, and the network algorithms are kept consistent in all structures functions except for changing the activation function.

4.1 Experiments based on the MNIST dataset

The MNIST dataset is a classical dataset for handwriting recognition, consisting of 10 categories of handwritten digits from 0-9. The training set contains 60,000 and the test set contains 10,000, and the

ISSN 2616-5775 Vol. 5, Issue 13: 69-76, DOI: 10.25236/AJCIS.2022.051311

images are grayscale images with a size of 28 × 28 pixels [23].

The network structure chosen for the experiments is shown in Figure 4. The output of each layer of the network indicates the number of channels, width, and height of the output, while the "layer" structure contains the "Batch Normalization (BN) layer + Activation function layer + Dropout layer + Fully connected layer" [24]. Without changing the network structure, the network has 15 neural network layers, the first 3 layers are convolutional, convolutional, and maximum pooling layers and the last 12 fully connected layers contain 500 neurons in each layer. The MNIST dataset is set to be trained on this network in batches of 10, and as the number of training increases, the loss value gradually decreases and the accuracy keeps increasing. The results are shown in figure 4.

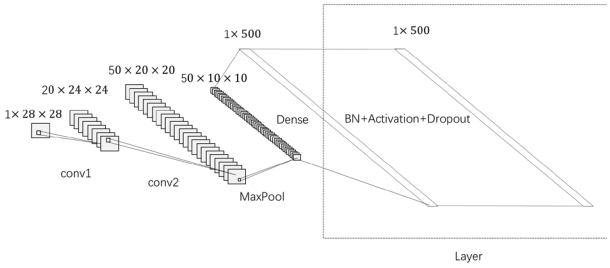

Figure 4: Network structure of the MNIST dataset experiment

As shown in Figure 5, the comparison of the XExp function with the ReLU function and the Swish and Mish functions for testing accuracy under the same network structure shows that the XExp activation function is faster in convergence and can reach an accuracy of 0.9387 at the end of the first batch of training, which enables the network parameters to be updated and optimized most efficiently, fitting a better network model and eventually being able to achieve Better accuracy. The results are shown in figure 5.

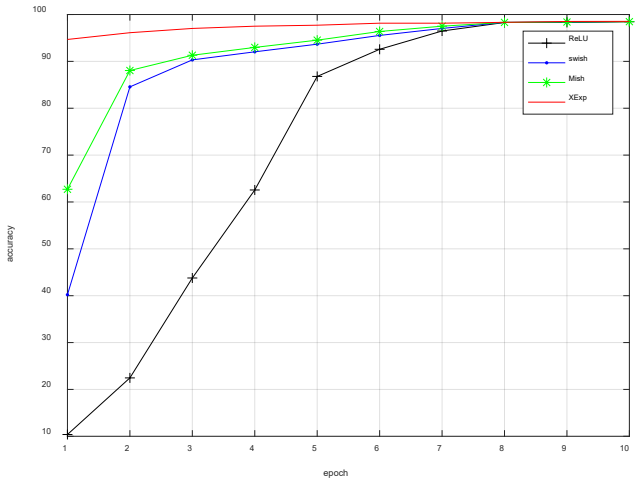

Figure 5: Test accuracy of the MNIST dataset in a 15-layer network

The above experiments verified the rapidity of the XExp function in facilitating the convergence of the network model. To more comprehensively verify the performance of the proposed activation function XExp function, the number of layers of the network structure was increased to 25 layers for the second experiment. The network structure includes the first 3 layers of the convolutional layer, and the maximum pooling layer respectively, and the layer contains 12 layers of the combined structure are increased to 22 layers, and the obtained network accuracy test is shown in Figure 6.

ISSN 2616-5775 Vol. 5, Issue 13: 69-76, DOI: 10.25236/AJCIS.2022.051311

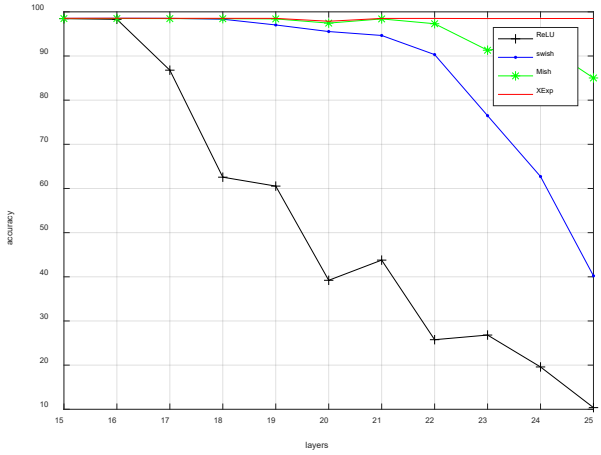

Figure 6: Test accuracy of the MNIST dataset in different layers of the network

As can be seen from Figure 6, with the increase in the number of network layers, the ReLU function, as well as the Swish function and the Mish function, show a trend of gradually decreasing detection accuracy, although the Swish function and the Mish function show more prominent stability compared with the ReLU function, the detection accuracy of the network also shows a decreasing trend after increasing to the 21-layer network, while for the XExp function, the Therefore, the XExp activation function can effectively avoid the overfitting of the network, while the other activation functions cannot maintain it.

4.2 Experiments based on CIFAR-10 and CIFAR-100 datasets

To further verify and analyze the recognition effect of XExp function on different datasets, CIFAR-10 [25] and CIFAR-100 [26] are selected as the experimental datasets, compared with the MNIST handwriting recognition dataset, the CIFAR dataset is more complex and is a color image dataset, i.e., the number of channels of the input data is 3. The sample is shown in Figure 7. Since the detection of color images requires a neural network with stronger learning ability, a relatively complex convolutional neural network model is chosen for the experiments on the CIFAR dataset.

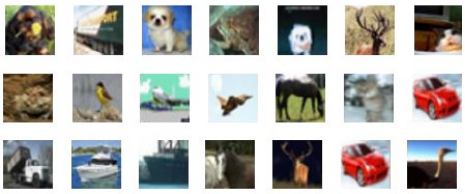

Figure 7: Sample CIFAR dataset

The image categories in the CIFAR-10 dataset are divided into 10 categories in total, and each category contains 6000 images. The training set contains 50,000 images and the test set contains 10,000 images with an image size of 32×32. The training batches of each network model on the dataset are 200 times and the batch_size is 64, and the training results are shown in Table 1.

Table 1: Test accuracy of different network models with different activation functions on the CIFAR-10 dataset (%)

Models	XExp	ReLU	Mish	Swish
LeNet	72.60	70.35	72.17	70.50
AlexNet	76.98	75.89	76.00	73.65
ResNet18	91.90	91.50	91.81	91.67
ResNet32	92.35	91.78	92.29	92.01

As can be seen from Table 1, the XExp function works well in all four of the more classic networks mentioned above. In comparison, the lighter the network is, the more obvious the detection accuracy is improved by the XExp function.

The CIFAR-100 dataset is more diverse and complex compared to the CIFAR-10 dataset, containing

ISSN 2616-5775 Vol. 5, Issue 13: 69-76, DOI: 10.25236/AJCIS.2022.051311

100 categories of images, 600 images per category, where the ratio of training set to test set is 5:1. The testing results are shown in Table 2.

Table 2: Testing accuracy of different network models with different activation functions on the CIFAR-100 dataset (%)

Models	XExp	ReLU	Mish	Swish
LeNet	39.63	37.98	37.43	37.50
AlexNet	42.54	41.91	40.97	41.23
ResNet18	67.39	67.23	67.26	67.18
ResNet32	69.51	68.45	69.44	68.76

The combined results of the experiments in the table and the above experiments in this section can verify that the XExp function proposed in this section has good test results both in simple data sets and simple networks, and in complex data sets and complex neural networks, which shows that the XExp function in the network does accelerate the convergence speed of the network, and its ability to prevent overfitting and its generalization ability is very good.

4.3 Computational Speed Experiment

The above experiments verify that the XExp function can accelerate the convergence speed of the network and at the same time has good overfitting prevention. The experimental design of XExp function, Swish function, and Mish function are compared after 105 calculations each, and the experimental results are shown in Table 3.

Table 3: XExp function, Swish function, Mish function calculation time (ms)

Functions	XExp	Mish	Swish
Original function	0.9311	2.2423	0.7108
First Order Derivative	1.0894	2.4114	1.0682
Second order derivative	1.3755	3.1084	1.3725

The running time of the XExp function in the above table is significantly smaller than that of the Mish function, but the computation time increases slightly compared to the Swish function due to its use as a segmentation function. However, the gradient of the XExp function is larger in the convolutional neural network compared to the other functions, and the parameters can be updated more rapidly toward the optimal values, leading to the convergence of the model after a smaller number of training sessions.

5. Conclusion

The new segmented activation function proposed in this paper, the XExp activation function, has the same positive half-axis choice as the ReLU function, which ensures the original distribution of the input data; the negative half-axis is chosen in a way that retains the advantages of the non-ReLU family function and avoids the neuron that appears in the negative half-axis interval of the ReLU family function The problem of death and gradient disappearance of the function converges the mean value of the function toward 0. By conducting different experiments on the MNIST dataset, CIFAR-10 dataset, and CIFAR-100 dataset, the effectiveness and better generalization ability of XExp activation function are verified, and it also can effectively avoid the overfitting phenomenon. In the next step, the activation function can be applied to a larger dataset to test its effectiveness in the practical application of target detection, and the proposed new activation function still has a relatively large room for improvement in terms of accuracy and speed.

References

[1] Li HW, Wu QX. Implementation scheme of neural network activation function in smart sensors[J]. Sensors and Microsystems,2014,33(1):46-48.
[2] Glorot X, Bengio Y. Understanding the di-fficultyof training deep feedforward neural networks[J]. Journal of Machine Learning Research,2010,9:249-256.
[3] Nair V, Hinton G E. Rectified linear units improve restricted boltzmann machines[C]. International Conference on Machine Learning, Omnipress, 2010:807-814.
[4] Dubey A K, Jain V. A comparative study of relu and leaky-relu activation functions for convolutional neural networks [M]. Applications of computing, automation and wireless systems in electrical

ISSN 2616-5775 Vol. 5, Issue 13: 69-76, DOI: 10.25236/AJCIS.2022.051311

engineering.Springer, Singapore, 2019: 873-880.
[5] He K, Zhang X, Ren S, et al. Delving deep into rectifiers: beyond human-level performance on imagenet classification [C]. Proceedings of the IEEE International Conference on Computer Vision. 2015: 1026-1034.
[6] Shi Q. Research and validation of image classification optimization algorithm based on convolutional neural network[D]. Beijing Jiaotong University,2017.
[7] WANG Hongxia, ZHOU Jiaqi, KU Chenghao, LIN Hong. Design of activation functions in convolutional neural networks for image classification[J]. Journal of Zhejiang University (Engineering Edition),2019,53(07):1363-1373.
[8] Clevert, Djork Arné, Unterthiner T, et al. Fast andaccurate deep network learning by e-xponential linear units (ELU) [J]. Computer S-cience, 2015.
[9] Ramachandran P, Zoph B, Le Q V. Searching for an Activation Functions[C]. ICLR 2018 Conference. 2017-10-27.
[10] BALDI P, SADOWSKI P, LU Zhiqin. Learning in the Machine:Random Backpropagation and the Deep Learning Channel[J]. Artificial Intelligence, 2018, 260: 1-35.
[11] Bu F. Research on small target detection and segmentation algorithm based on convolutional neural network[D]. Xi'an University of Electronic Science and Technology, 2019.
[12] NAIR V, HINTON G E. Rectified Linear Units Improve Restircted Boltzmann Machines[C] /Proceedings of the27th International Conference on Machine Learning.New York:ACM, 2010:807-814.
[13] HOCHREITER S. The Vanishing Gradient Problem during Learning Recurrent Neural Nets and Problem Solutions[J]. International Journal of Uncertainty Fuzziness and Knowledge-Based Systems, 1998, 6 (2) :107-116.
[14] LECUN Y, BENGIO Y, HINTON G. Deep learning [J]. Nature, 2015, 521:436-444.
[15] LIU YC, WANG TH, et al. A novel adaptive activation function for deep learning neural networks[J]. Journal of Jilin University (Science Edition),2019,57(4):857-859.
[16] Lin L. Small target detection based on deep learning [D]. University of Electronic Science and Technology,2020.
[17] ZHOU P, NI B B, GENG C, et al. Scale-transferrable object detection[C]/Proc. of the IEEE Conference on Computer Vision and Pattern Recognition, 2018 : 528-537.
[18] LI Zechao, TANG Jinhui. Weakly Supervised Deep Metric Learning for Community-Contributed Image Retrieval[J]. IEEE Transactions on Multimedia, 2015, 17 (11) : 1989-1999.
[19] He Kaiming, Zhang Xiangyu, Ren Shaoqing, et al. Developing Deep into Rectifiers:Surpassing Human-Level Performance on ImageNet Classification[EB/OL].2015-02-06. https:// arxiv.org/abs/1502.01852.
[20] CLEVERT D A, UNTERTHINER T, HOCHREITER S. Fast and Accurate Deep Network Learning by Exponential Linear Units (ELU) [C/OL]//ICLR 2016.2016-02-22. https://arxiv.org/abs/1511.07289.
[21] Bei Su. Real-time target detection based on multi-scale neural network and self-attentive mechanism [D]. Xi'an University of Electronic Science and Technology,2020.
[22] Li Huihui, Zhou Kangpeng, Han Taichu. Improved SSD ship target detection based on CReLU and FPN[J]. Journal of Instrumentation, 2020,41(04):183-190.
[23] Xu Yankai, Liu Zengmei, Xue Yaru, Cao Siyuan. A seismic random noise suppression method applying two-channel convolutional neural network[J]. Petroleum Geophysical Exploration, 2022,57(04):747-756+735.DOI:10.13810/j.cnki.issn.1000-7210.2022.04.001.
[24] Su Zengzhi, Wang Meiling, Yang Chengzhi, Wu Hongchao. Radar signal modulation mode identification based on improved EfficientNet [J/OL]. Telecommunications Technology: 1-9 [2022-08-28].
[25] Pang C, Jiang Y, Wu T, Liao C-W, Ma W-G. Effect of neural network parameters on earthquake type recognition[J]. Science Technology and Engineering,2022,22(18):7765-7772.
[26] Liang Ruobing, Liu Bo, Sun Yuehong. Advances in deep learning empirical loss function geomorphic analysis [J/OL]. Systems Engineering Theory and Practice:1-14 [2022-08-28].