The Role of Activation Function in CNN

Wang Hao
College of Computer Science and Information Engineering
Shanghai Institute of Technology
ShangHai, China
e-mail:wh18895623027@163.com

Wang Yizhou
College of Computer Science and Information Engineering
Shanghai Institute of Technology
ShangHai, China
e-mail: 1025190165@qq.com

Lou Yaqin
College of Chemistry and Chemical EngineeringSoutheast University
NanJing, China
e-mail: 1850131001@qq.com

Song Zhili
College of Computer Science and Information Engineering
Shanghai Institute of Technology
ShangHai, China
* Corresponding author e-mail: ZLSONG@SIT.edu.cn

***Abstract*—We all know that the purpose of introducing activation function is to give neural network nonlinear expression ability, so that it can better fit the results, so as to improve the accuracy. However, different activation functions have different performance in different neural networks. In this paper, several activation functions commonly used by researchers are compared one by one, and qualitative comparison results are given by combining with specific neural network models. For example, when using the MNIST dataset in LeNet, PReLU achieved the highest accuracy of 98.724%, followed by Swish at 98.708%. When cifar-10 data set was used, the highest accuracy rate of ELU was 64.580%, followed by Mish at 64.455%. When Using VGG16, ReLU reached the highest accuracy of 90.226%, followed by PReLU at 90.197%. When using ResNet50, ELU achieved the highest accuracy of 89.943%, followed by Mish at 89.780%.**

Keywords: activation function; convolutional neural network; accuracy;

I. INTRODUCTION

How to balance accuracy and speed is always the goal of researchers. Training a high accuracy model requires a lot of training data. This is not only a high demand for data but also a challenge to the computational power of researchers. Low-configuration Graphics Processing Units (GPU) can only use a small mini-batch-size, which can result in a large amount of time required to train a model. Researchers then turn their attention to how to get more accurate models without increasing the data set. There are many methods, such as improving data enhancement algorithm, adjusting network structure, improving activation function and so on.

Since the advent of neural networks, many researchers have tried to improve their performance in specific tasks. For example, the Sigmoid function as a commonly used activation function in the basic neural network has a lot of room for use in logistic regression problems. In the image classification task, since the network structure with a deeper number of layers is used, we usually use the ReLU function to improve the non-linear ability of the network and avoid the gradient disappearance or the gradient explosion after multiple iterations of the network. phenomenon. The recent Mish activation function has good performance in many network structures, but it is not necessarily the best activation function. It is necessary to do a quantitative study of accuracy according to different network structures and different data sets.

II. RELATED WORK

Each neuron node in the neural network accepts the output value of the neuron of the previous layer as the input value of this neuron, and passes the input value to the next layer. The input layer neuron node will directly pass the input attribute value to the next layer. layer. In a multilayer neural network, there is a functional relationship between the output of the upper node and the input of the lower node. This function is called the activation function.

The ideal activation function should have the following characteristics:

(1) It can prevent the gradient from disappearing when outputting to the data at both ends.

(2) Take the coordinate (0,0) as the center of symmetry, so that the gradient will not move in a specific direction.

(3) Since each layer of the network needs to use an activation function, its computational cost should be very low.

(4) The neural network uses the gradient descent method for iterative training, and the activation function used in each layer should be differentiable.

In the research of deep learning, some scholars pay more attention to finding a good activation function. The purpose of adding activation functions to the neural network is to introduce nonlinear capabilities, and different activation functions have different effects on the nonlinear fitting capabilities of the model. Generally, the properties that the activation function should have are:

(1) Non-linearity: the derivative is not a constant. This can ensure that the multilayer network does not degenerate into a single-layer linear network.

(2) Differentiability: corresponds to the computability of the gradient in optimization.

978-0-7381-1141-4/20/$31.00 ©2020 IEEE

DOI 10.1109/ITCA52113.2020.00096

(3) Simple: A complex activation function will reduce the calculation speed.

(4) Saturation: Saturation refers to the problem that the gradient is close to zero in certain intervals (that is, the gradient disappears), making it impossible to update the parameters.

(5) Monotonic: The sign of the derivative does not change. When the activation function is monotonic, the single-layer network can be guaranteed to be a convex function.

(6) Fewer parameters: Most activation functions have no parameters.

Early researchers used Sigmoid and Tanh more frequently. When the variable takes a large positive value or a small negative value, saturation will occur, and it is no longer sensitive to small changes in the input data. In back propagation, when the gradient is close to 0, the weight will not be updated basically, and the gradient will disappear easily, so that the training of the deep network cannot be completed. In order to solve this problem, the ReLu[1] activation function was proposed by Nair and Hinton in 2010. It has low computational complexity and does not require exponential calculations. The activation value can be obtained as long as a threshold value. Due to these advantages, the ReLu activation function has been studied They are widely used, and the feedforward neural network model is used as the default activation function. The disadvantage is that the ReLu function can only solve the problem that the gradient disappears when the variable value is positive. Then came some activation functions, such as leakyReLu[2], PReLu[3], ReLu6[4], SELU[5], Swish[6], hard-Swish[7] and Mish[8], which were also used to solve the problem of gradient disappearance when the variable value was negative.

III. OUR WORK

In order to compare the properties of several commonly used activation functions, we draw some of the images of the activation functions and analyze them. We can see that the mathematical properties of different activation functions are quite different. The activation function with arctan(x) as the composite has more obvious gradient changes than the activation function with tanh(x)[10] as the composite, so it can converge faster during network training.

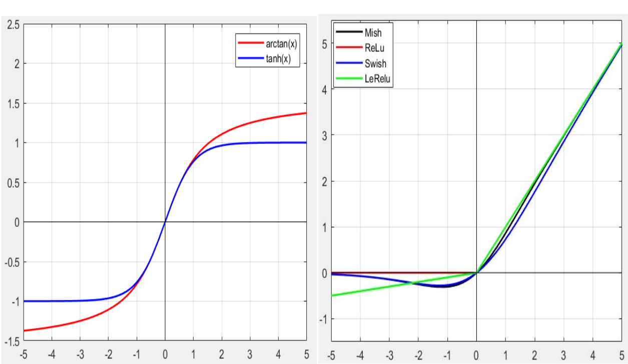

Figure 1. Function graphs of several activation functions

As can be seen from the figure above, arctan(x) has a more pronounced gradient change in the positive X-axis, while tanh(x) reaches saturation faster. From the activation function image on the left, it can be seen that the positive gradient change of ReLU function image in the X-axis is constant, but the negative gradient change in the X-axis is 0, which will cause the gradient disappearance of the network model during training. Addressing this situation, LeakyReLU can be better solved. When LeakyReLU's input reaches the negative direction of the X-axis, it still has a certain gradient, so that gradient disappearance does not occur in the network model during training. Both Swish and Mish belong to the same form of activation function, which are non-monotonic and self-regularized activation functions. Among them, compared with Swish, Mish has a trend of higher gradient change, especially in the change of the second derivative.

IV. EXPERIMENTAL RESULTS

The platform of this experiment: the operating system is Ubuntu16.04 LTS, 64-bit; the processor model is i5-9600K, 3.70GHz×6; the graphics card model is GeForce RTX 2060 SUPER 8G; the memory size is 16G; the Python version is 3.6; Pytorch version It is 1.2; the data set uses MNIST, CIFAR-10.

A. Performance on the LeNet network

This experiment uses the LeNet[11] network with 2 convolutional layers, and the data set uses the handwritten letter MNIST data set. The Batchsize is 64, the initial learning rate is 0.001, the loss function uses the cross-entropy loss function, and the learning rate optimization uses the Adam[12] optimizer. It can be seen that PReLU achieves the highest accuracy after 40,000 iterations of training, which is 98.724%. Mish is followed by 98.708%, which has better performance. Among them, the Sigmoid activation function is prone to the problem of gradient disappearance during training, so it performs very poorly in this experiment. It can be seen that different activation functions can have great performance differences under the same network structure.

TABLE I. COMPARISON OF 8 ACTIVATION FUNCTIONS IN LENET

epoch	ReLU	LReLU	Tanh	Swish	ELU	PReLU	Mish	Sigmoid
5000	97.620	97.662	96.058	97.402	96.972	97.146	97.368	10.454
10000	98.140	98.160	97.312	98.128	97.702	98.192	97.884	12.760
15000	98.250	98.224	97.600	98.236	97.826	98.290	98.172	17.248
20000	98.320	98.278	97.710	98.296	97.900	98.398	98.246	21.082
25000	98.368	98.356	97.792	98.368	98.026	98.456	98.322	25.300
30000	98.450	98.426	97.876	98.490	98.116	98.496	98.516	28.052
35000	98.534	98.462	98.014	98.568	98.214	98.558	98.636	34.114
40000	98.628	98.602	98.204	98.674	98.318	**98.724**	98.708	43.864

This experiment uses a LeNet network with 2 convolutional layers, and the color image CIFAR-10 data set used in the data set. The batchsize is 64, the learning rate is optimized using the SGD[13] optimizer, the initial learning rate is 0.001, and the momentum is 0.9. The loss function uses a cross-entropy loss function. It can be seen that after 10,000 iterations of training ELU, the highest accuracy rate is 64.580%. Mish is closely followed with 64.455%. At the same time, it can be seen that when the number of neural network layers is small, the ability to classify color images is poor. Therefore, increasing the number of layers of the network model is an important part of improving the performance of the model.

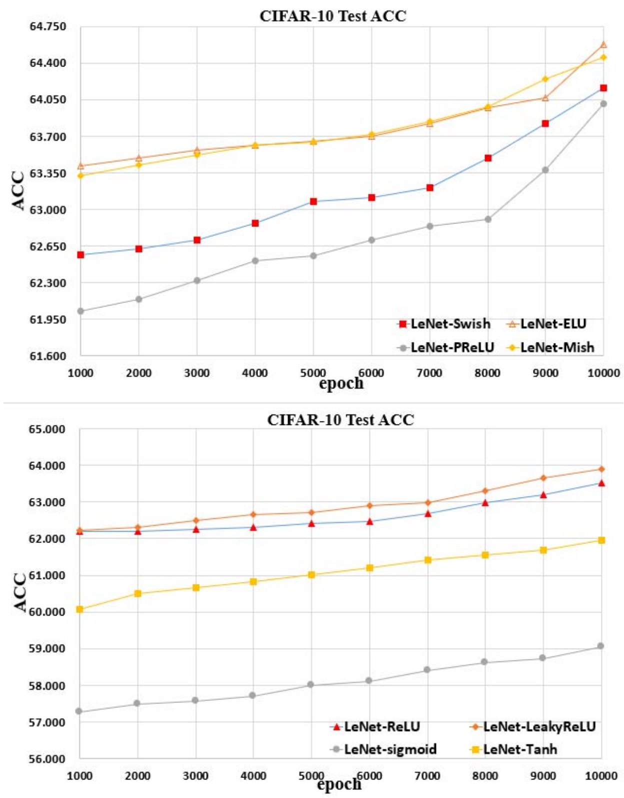

Figure 2. Comparison of 8 activation functions in LeNet

B. Performance in the VGG16 network

This experiment uses the VGG[14] network with 16 convolutional layers, and the color image CIFAR-10 data set used in the data set. The Batchsize is 64, the learning rate is optimized using the SGD[13] optimizer, the initial learning rate is 0.01, the momentum is 0.8, and the weight decay parameter is 0.001. The loss function uses a cross-entropy loss function. It can be seen that after 36,000 iterations of training ReLU, the highest accuracy rate is 90.226%. Among them, Sigmoid[9] performed poorly in this experiment.

TABLE II. COMPARISON OF 8 ACTIVATION FUNCTIONS IN VGG16

epoch	ReLU	LReLU	Tanh	Swish	ELU	PReLU	Mish	Sigmoid
4000	80.387	79.879	73.871	82.087	79.260	80.273	81.790	23.505
8000	83.492	83.482	78.547	84.353	81.436	82.213	83.967	24.148
12000	86.328	86.413	80.795	86.636	84.266	86.057	86.450	33.209
16000	86.945	86.471	80.961	86.770	84.488	86.682	86.513	34.098
20000	88.596	88.464	82.200	88.321	86.044	89.397	88.326	38.171
24000	88.854	88.718	82.858	88.918	86.305	89.568	88.670	45.378
28000	89.954	90.001	84.082	89.628	87.303	90.099	89.598	46.055
32000	90.029	91.071	84.151	89.640	87.345	90.162	89.642	48.307
36000	**90.226**	90.192	84.394	89.711	87.367	90.197	89.677	57.176

C. Performance in the ResNet50 network

This experiment uses a ResNet[15] network with 50 convolutional layers, and the color image CIFAR-10 data set used in the data set. The Batchsize is 128, the learning rate is optimized using the SGD optimizer, the initial learning rate is 0.1, the momentum is 0.9, and the weight decay parameter is 5e-4. The loss function uses a cross-entropy loss function. It can be seen that after 84,000 iterations of training ELU, the highest accuracy rate is 89.943%. Followed by Mish at 89.780%.At the same time, it can be seen that Sigmoid performed better in this experiment.

TABLE III. COMPARISON OF 8 ACTIVATION FUNCTIONS IN RESNET50

epoch	ReLU	LReLU	Tanh	Swish	ELU	PReLU	Mish	Sigmoid
12000	65.287	65.371	66.090	65.400	65.986	65.763	65.850	65.941
24000	84.269	84.611	84.370	84.050	84.702	84.387	84.590	84.010
36000	88.059	88.633	88.800	87.870	88.871	87.967	88.670	88.130
48000	88.952	89.286	89.270	88.900	89.693	89.242	89.500	89.030
60000	89.027	89.387	89.330	89.030	89.846	89.393	89.700	89.080
72000	89.050	89.439	89.350	89.060	89.883	89.425	89.740	89.110
84000	89.170	89.474	89.440	89.110	**89.943**	89.468	89.780	89.190

ACKNOWLEDGMENT

This thesis is completed under the kind care and careful guidance of my tutor, Mr. Song Zhili. His serious scientific attitude, rigorous academic spirit, and excelsior working style deeply infected and inspired me. Mr. Song not only gave me careful guidance in my studies but also gave me meticulous care in my thoughts and life. Here I would like to express my sincere gratitude and high respect to Mr. Song. It is better to teach a man to fish than to teach him to fish. I learned to accept new ideas and to think independently. I would also like to thank my girlfriend(Lou Yaqin) who encouraged me silently behind my back. Thanks to your help and support, I was able to overcome difficulties and doubts one by one until the successful completion of this article.

REFERENCES

[1] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In Proceedings of International Conference on Machine Learning (ICML), pages 807–814, 2010.

[2] AndrewLMaas,AwniYHannun,andAndrewYNg. Rectifier nonlinearities improve neural network acoustic models. In Proceedings of International Conference on Machine Learning (ICML), volume 30, page 3, 2013.

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), pages 1026–1034, 2015.

[4] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. MobileNets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861, 2017.

[5] G¨unter Klambauer, Thomas Unterthiner, Andreas Mayr, and Sepp Hochreiter. Self-normalizing neural networks. In Advances in Neural Information Processing Systems (NIPS), pages 971–980, 2017.

[6] Prajit Ramachandran, Barret Zoph, and Quoc V Le. Searching for activation functions. arXiv preprint arXiv:1710.05941, 2017.

[7] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, et al. Searching for MobileNetV3. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2019.

[8] Diganta Misra. Mish: A self regularized nonmonotonic neural activation function. arXiv preprint arXiv:1908.08681, 2019.

[9] Lin H T , Lin C J . A Study on Sigmoid Kernels for SVM and the Training of non-PSD Kernels by SMO-type Methods[J]. Submitted to Neural Computation, 2003.

[10] Fan E . Extended tanh-function method and its applications to nonlinear equations[J]. Physics Letters A, 2000, 277(4-5):212-218.

[11] Al-Jawfi R . Handwriting Arabic Character Recognition LeNet Using Neural Network[J]. International Arab Journal of Information Technology (IAJIT), 2009, 6(3):304-309.

[12] Kingma D , Ba J . Adam: A Method for Stochastic Optimization[J]. Computer ence, 2014.

[13] Paras. Stochastic Gradient Descent[J]. Optimization, 2014.

[14] Dongxin G , Kaiyan C , Yang Z , et al. New template attack method for encryption chip based on VGGNet convolutional neural network[J]. Application Research of Computers, 2019.

[15] Szegedy C , Ioffe S , Vanhoucke V , et al. Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning[J]. 2016.