A Combined Activation Function for Learning Performance Improvement of CNN Image Classification

Guangliang Pan[1, a], Jun Li[2, b, *], Fei Lin[2, c], Tingting Sun[3, d], Yulin Sun[1, e]
[1]*School of Electrical Engineering and Automation, Qilu University of Technology(Shandong Academy of Sciences), Jinan 250353, China*
[2]*School of Electronic and Information Engineering (Department of Physics), Qilu University of Technology (Shandong Academy of Sciences),Jinan 250353,China*
[3]*Weihai Ocean Vocational College, Weihai 264300, China*

Keywords: Convolutional neural network, LeNet-5, sigmoid, tanh-relu, tanh, relu,

Abstract: With the rise of artificial intelligence, it has unlimited possibilities for machines to replace human work. Aiming at how to improve the learning performance of convolutional neural network (CNN) image classification by changing the activation function, a combined Tanh-relu activation function is proposed based on the single Sigmoid, Tanh and Relu activation functions. Based on CNN-LeNet-5, the size of the convolution kernel and sampling window is changed and the number of layers of the convolutional neural network is reduced. At the same time, the network structure of the LeNet-5 model is improved. On the Mnist handwritten digital dataset, the combined Tanh-relu activation function was compared with a single activation function. The experimental results show that the CNN model with combined Tanh-relu activation function has faster accuracy fitting speed and higher accuracy, improves the convergence speed of loss and enhances the convergence performance of CNN model.

1 INTRODUCTION

In the information age, the explosive growth of data volume makes the display of deep learning (DL) extraordinarily important (Meyer P, Noblet V , Mazzara C , et al, 2018). As a common model of deep learning, convolutional neural networks have achieved great success in image processing (Arena P , Basile A , Bucolo M , et al, 2003; Al-Ajlan Amani, El Allali Achraf, 2018). In order to make the neural network learn more complex data, the activation function introduces the nonlinear input into the neural network by converting the input signal of the node into the output signal, enhancing the learning ability of the neural network model and improving the classification performance.

At the beginning of the introduction of the activation function, mainly the Sigmoid and Tanh activation functions are mainly applied to various neural network models. Both of these activation functions are saturated S-type activation functions, which are prone to gradient dispersion during neural network training. For the study of activation functions, people have never stopped. Kr-izhevsky used Relu (corrected linear unit) as the activation function for the first time in the ImageNet ILSVRC competition (Krizhevsky, Alex, I. Sutskever, and G. Hinton, 2012). All of the above studies have studied a single activation function, without considering the use of a single activation function. Improving the convolutional neural network structure is also an important way to optimize the learning performance of the model (Horn Z C, Auret L, Mccoy J T, et al, 2017).

Inspired by the literature (Qian S , Liu H , Liu C , et al, 2017, Yao G , Lei T , Zhong J, 2018), this paper combines the advantages and disadvantages of the three activation functions, and reconstructs the LeNet-5 model of the convolutional neural network structure, reduces the layer of the fully connected layer and changes the size of the convolution kernel. A combined activation function is applied to convolutional neural network image classification to improve the image classification performance of convolutional neural networks.

Pan, G., Li, J., Lin, F., Sun, T. and Sun, Y.
A Combined Activation Function for Learning Performance Improvement of CNN Image Classification.
DOI: 10.5220/0008851103600366
In *Proceedings of 5th International Conference on Vehicle, Mechanical and Electrical Engineering (ICVMEE 2019)*, pages 360-366
ISBN: 978-989-758-412-1

2 USED ACTIVATION FUNCTION

The activation function is an indispensable part of the convolutional neural network. In the early stage of deep learning, Sigmoid is a widely used activation function (A. Uncini, L. Vecci, S. Member, and F. Piazza, 1999). It is a saturated S-type activation function with mathematical expressions:

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

According to the formula analysis, the function input value is from negative infinity to positive infinity, and the function output value is always positive. When the input value is positive or negative infinity, the gradient gradually disappears to 0, causing the gradient to diffuse and the convergence to be slow. The output value is not centered on 0, and model optimization is more difficult.

Aiming at some shortcomings of the sigmoid activation function, a new activation function Tanh activation function is also developed, which is also a saturated S-type activation function. The definition formula is:

$$f(x)=\frac{1-e^{-2x}}{1+e^{-2x}} \quad (2)$$

It can be seen from the formula that the output value of the Tanh activation function becomes -1 to 1, and the model optimization is easier with 0 as the center, but there is still a gradient dispersion problem.

The Relu activation function is a new activation function that is proposed after deep learning research. Relative to the previous two activation functions is simpler, the mathematical expression is:

$$f(x)=\max(x,0) \quad (3)$$

According to the formula analysis, the new Relu activation function does not disappear when the input value tends to be infinite, which effectively solves the gradient dispersion problem, the convergence speed is faster, and the model optimization is better. However, some gradients may be weak or even death during training, which is called death neuron. Secondly, it can only be applied to the hidden layer of neural network model, which has certain limitations.

3 COMBINED ACTIVATION FUNCTION APPLIED TO CONVOLUTIONAL NEURAL NETWORK

Convolutional neural network is a model of deep learning (Philipp P, Felix V, 2018). It has the characteristics of local connection and parameter sharing (Sharma N, Jain V, Mishra A, 2018). It has strong advantages in image processing. LeNet-5 model is a classic model of convolutional neural network (Fu R, Li B, Gao Y, et al, 2018; Albelwi S, Mahmood A, 2016). Change the size of the convolution kernel and sampling window based on the original LeNet-5 model. The improved LeNet-5 model is shown in Fig. 1.

The convolutional neural network based on the improved LeNet-5 model is trained by BP algorithm (Shuchao P, Anan D, Orgun M A, et al, 2018). The training process of BP algorithm is divided into forward calculation of data, back propagation of error and update of weight. Defining the error of the current output layer to the partial derivative of the input as, it is called sensitivity.

For forward calculation of data, the hidden layer output values are defined as follows:

$$\begin{cases} a_h^{H1}=W_h^{H1}\times X_i \\ b_h^{H1}=f(a_h^{H1}) \end{cases} \quad (4)$$

Where X_i is the current node input, W_h^{H1} is the weight, and $f(\bullet)$ is the current layer activation function.

The output value of the output layer is defined as follows:

$$a_k=\sum W_{hk}\times b_h^{H1} \quad (5)$$

In Tensorflow, the one-hot method is usually used for error back propagation and weight update. The cross entropy function is defined as follows:

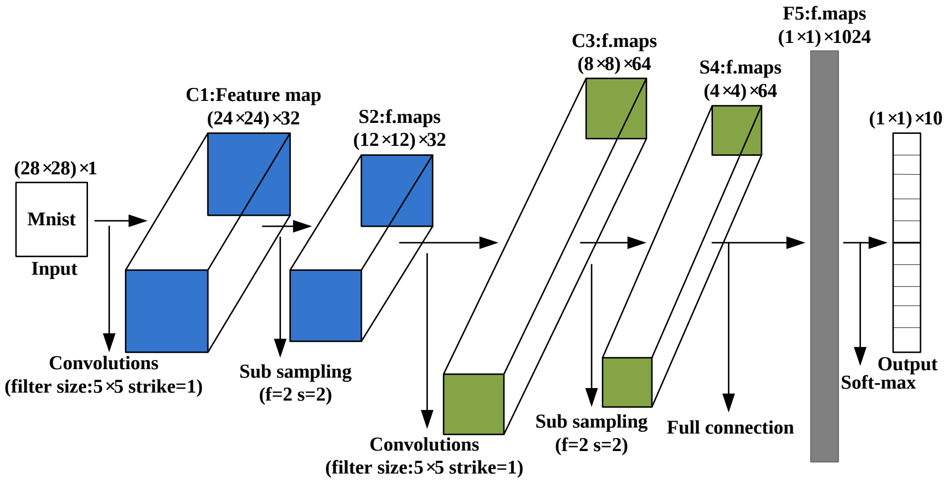

Figure 1. Improved structural model of CNN-LeNet-5.

$$loss = -y\log(f(x)) \tag{6}$$

The output layer feeds back to the reverse derivation of the fully connected layer. According to the one-hot method, only one value is "1" and the rest is "0". The cross entropy is defined as:

$$\begin{aligned} loss(f(x), y) &= -\sum y\log(f(x)) \\ &= -(0\times\log(f(x_1)) + \cdots + 1\times\log(f(x_n))) \\ &= -\log(f(x_n)) \end{aligned} \tag{7}$$

The loss value is as follows:

$$loss = -(y - \log(f(x))) \tag{8}$$

Let $y = 1$ get:

$$loss = -(1 - \log(f(x))) \tag{9}$$

The output layer uses the Soft-max classification, and the formula for the full connection layer weight update is as follows:

$$\frac{\delta loss}{\delta W} = -\frac{1}{m}\times(1 - f(x))f'(x) + \lambda W \tag{10}$$

The pooled layer feeds back to the reverse derivation of the convolutional layer, and its convolutional layer sensitivity is as follows:

$$\delta_j^l = pool(\delta_j^{l+1}) * h'(a_j^l) \tag{11}$$

Among them, * is the dot multiplication. The convolutional layer feeds back to the reverse derivation of the pooling layer, assuming that l is a pooling layer, $l+1$ is a convolutional layer, and the convolutional layer has m features, and the sum of the pooling layer sensitivity is as follows:

$$\delta_j^l = \sum\nolimits_j^m (\delta_j^{l+1}) \odot K_{ij} \tag{12}$$

Among them, $\odot$ is a convolution operation. The sensitivity is obtained by the above calculation, and then the weights and offsets in the convolutional neural network are as follows:

$$\frac{\delta loss}{\delta W_{ij}} = X_i \delta_j^{i+1} \tag{13}$$

$$\frac{\delta loss}{\delta b_{ij}} = \sum(\delta_j^{i+1}) \tag{14}$$

Through the above theoretical analysis of the error propagation between the convolutional neural network convolutional layer and the pooled layer, from the first convolutional layer to the pooled layer uses the Relu activation function. From the pooled layer to the second convolutional layer uses the Tanh activation function. From the pooling layer to the fully connected layer uses the Relu activation function. Finally the fully connected layer to the output layer uses the Soft-max classification.

This combination of a single activation function in the hidden layer, first of all, the Relu activation function is currently the single-function activation function of the strongest performance, and it has been pointed out in the analysis of the second section that it is faster than the first two activation functions. The convergence speed, so the convolutional neural network model will soon reach a relatively stable state range in the early stage. At this point, the intervention of the Tanh activation function makes the model enter a better optimization state, instead of using the Sigmoid activation function, because the Tanh activation function itself has stronger optimization ability than the Sigmoid activation function. After extracting the higher-level features, the optimization performance enters the bottleneck period, and the Relu activation function is used again to improve the convergence performance, so that the final output layer has higher classification accuracy.

4 EXPERIMENTAL RESULTS AND PERFORMANCE EVALUATION

In order to verify the feasibility of the proposed combined activation function, an experimental analysis was carried out. First, set two evaluation indicators: loss and accuracy. This paper solves the classification problem of images, and the full connection layer to output uses Soft-max classification, so loss is the log loss corresponding to the calculation. The classification probability of each class is calculated by the Soft-max layer, and then the loss is calculated by cross_entropy. The formula is defined as:

$$P(x=i)=e^{a_i}\sum_{k=1}^{100}e^{a_k} \tag{15}$$

$$l=-\sum_{k=1}^{100}x_k \log P(x=k) \tag{16}$$

Where a_i is the i-th neuron output value in the Soft-max classification; x_k is the label value of the k-th class (this is the k-th class, which is 1, otherwise 0); $P(x=i)$ is the classification probability of the i-th class; l is the output value of the cross entropy loss.

Accuracy is the ratio of the correct number of samples to the total number of samples (the classification accuracy), the formula is as follows:

$$accuracy=\frac{n_r}{N} \tag{17}$$

Where n_r is the correct number of samples and N is the total number of samples.

Since the convolutional neural network model is in a convergent state at the end of training, the two performance indicators loss and accuracy fluctuate within a small range. Therefore, with the average of the last three rounds as the final loss and accuracy, the formula is as follows:

$$Average_{lo}=\sum_{18}^{20}\frac{loss_i}{5} \tag{18}$$

$$Average_{ac}=\sum_{18}^{20}\frac{accuracy_i}{5} \tag{19}$$

Where $loss_i$ is the loss of each round, $Average_{lo}$ is the mean loss, $accuracy_i$ is the classification accuracy of each round, and $Average_{ac}$ is the mean accuracy. The number of training rounds of the convolutional neural network model is 20 rounds. On the MNIST handwritten digit recognition data set, the evaluation indexes $loss_t$, $accuracy_t$ of the training set is respectively counted.

Mnist is a data set for handwritten digit recognition, which is 10 numbers handwritten from 0 to 9. The convolutional neural network is trained according to the LeNet-5 network structure in Section 3. The model parameters are set as shown in Table 1.

Table 1. Model parameter.

Parameter	Parameter value
Training set	60000(36×36)
Test set	10000(36×36)
Number of network layers(N)	5
Learning efficiency(α)	0.0001
Number of training rounds	20
Batch-size	50

On the Mnist dataset, the evaluation indicators of the convolutional neural network model test results are shown in Table 2. It can be seen from Table 2 that the Tanh-relu combined activation function has the lowest loss and the highest accuracy on the training set. On the training set, the final classification accuracy of the Tanh-relu activation function is 11.6% higher than the sigmoid activation function, 2.1% higher than the Tanh activation function, and 1.6% higher than the Relu activation function. Due to the increased complexity of the combined activation function, there is a slight increase in training time compared to a single activation function. The experimental results on the MNIST dataset show that the average training time per round of the Tanh-relu activation function is only 2.48s, 1.37s, and 0.44s higher than that of Sigmoid, Tanh, and Relu.

Table 2. Experimental results of different activation functions on MNIST.

Activation function	$loss_t$	$accu_t$
Sigmoid	1.5943	0.881
Tanh	1.5103	0.961
Relu	1.5003	0.966
Tanh-relu	1.4936	0.981

Fig. 2 shows the accuracy fit of different activation functions on a convolutional neural network model. It can be seen from Fig. 2 that during the whole training process, the four activation functions are quickly fitted to a higher accuracy, but the sigmoid activation function has a significant difference compared with the other three. The accuracy of the Tanh-relu activation function is the fastest, and the Tanh activation function is not much different from the Relu activation function. Finally, the classification accuracy of the Tanh-relu activation function is the highest, indicating that the CNN model with the combined Tanh-relu activation function has a faster accuracy rate and higher accuracy.

Fig. 3 shows the loss convergence graph for different activation functions on a convolutional neural network model. As can be seen from Fig. 3, the loss of the four activation functions quickly converges below 1.6 during the entire training process. Except for the sigmoid activation function, the other three activation function curves are very close. The loss convergence of the Tanh-relu activation function is the fastest, and the convergence speed of the Tanh activation function and the Relu activation function is equivalent. Finally, the loss of the Tanh-relu activation function is the lowest, indicating that the combined Tanh-relu activation function improves the convergence performance of the convolutional neural network model, there by enhancing the model learning performance.

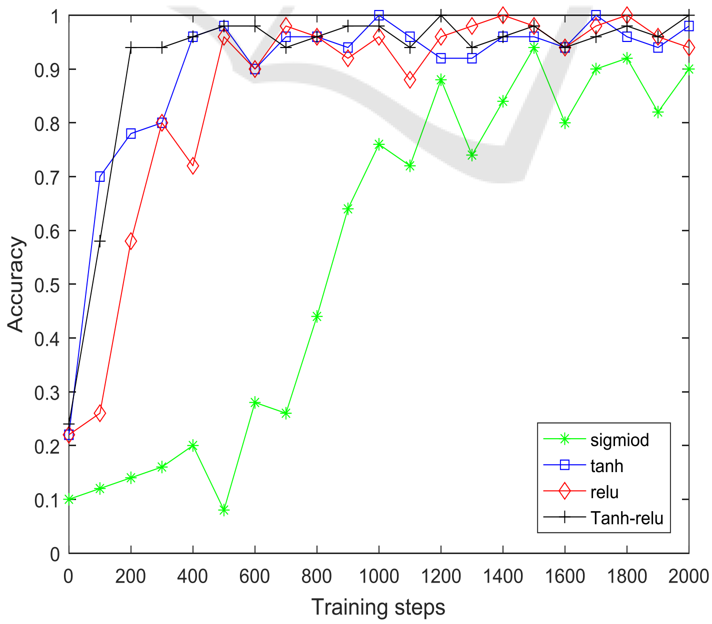

Figure 2. Accuracy of different activation functions.

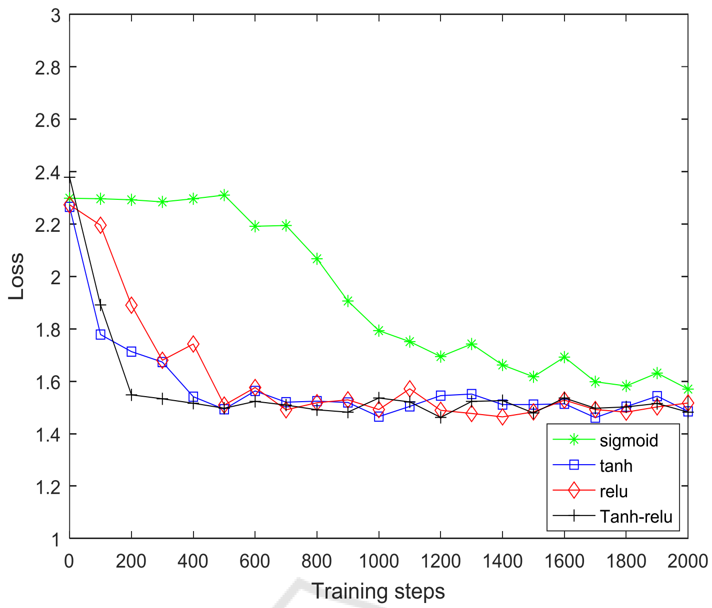

Figure 3. Loss of different activation functions.

5 CONCLUSIONS

By analyzing the three commonly used activation functions and the propagation characteristics of the convolutional neural network model, we propose a combined activation function Tanh-relu. The Tanh and Relu activation functions alternate in the hidden layer of the convolutional neural network, giving full play to the advantages of each activation function and effectively improving the learning ability of the model. The experimental results on the common Mnist dataset show that under the same learning rate, the Tanh-relu combined activation function has the highest classification accuracy and the best convergence effect compared to the single activation function. In this paper, the activation function improves the learning ability of the model, but there is still room for improvement. Deep learning is developing towards low-precision data, and we can study how to develop a new activation function to prevent loss of precision.

ACKNOWLEDGEMENTS

Shandong Ship Control Engineering and Intelligent System Engineering Research Center 2019 Open Special Fund Project SSCC-2019-0007.

REFERENCES

Al-Ajlan Amani,El Allali Achraf. CNN-MGP: Convolutional Neural Networks for Metagenomics Gene Prediction. [J]. Interdisciplinary sciences, computational life sciences, 2018.

Albelwi S, Mahmood A. Automated Optimal Architecture of Deep Convolutional Neural Networks for Image Recognition[C]// 2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA). IEEE, 2016.

Arena P, Basile A, Bucolo M, et al. Image processing for medical diagnosis using CNN [J]. NUCLEAR INSTRUMENTS AND METHODS IN PHYSICS RESEARCH SECTION A, 2003.

A. Uncini, L. Vecci, S. Member, and F. Piazza, “Complex-Valued Neural Networks with Adaptive Spline Activation Function for Digital Radio Links Nonlinear Equalization,” vol. 47, no. 2, pp. 505–514, 1999.

Fu R , Li B , Gao Y , et al. Visualizing and Analyzing Convolution Neural Networks with Gradient Information [J]. Neurocomputing, 2018: S0925231218302479.

Horn Z C, Auret L, Mccoy J T, et al. Performance of Convolutional Neural Networks for Feature Extraction in Froth Flotation Sensing [J]. IFAC-PapersOnLine, 2017, 50(2):13-18.

Krizhevsky, Alex, I. Sutskever, and G. Hinton. "ImageNet Classification with Deep Convolutional Neural Networks." NIPS Curran Associates Inc. 2012.

Meyer P, Noblet V, Mazzara C, et al. Survey on deep learning for radiotherapy [J]. Computers in Biology & Medicine, 2018, 98:126.

Philipp P, Felix V. Optimal approximation of piecewise smooth functions using deep ReLU neural networks [J]. Neural Networks, 2018:S0893608018302454-.

Qian S , Liu H , Liu C , et al. Adaptive activation functions in convolutional neural networks[J]. Neurocomputing, 2017:S0925231217311980.

Sharma N, Jain V, Mishra A. An Analysis of Convolutional Neural Networks for Image Classification [J]. Procedia Computer Science, 2018, 132:377-384.

Shuchao P, Anan D, Orgun M A, et al. A novel fused convolutional neural network for biomedical image classification [J]. Medical & Biological Engineering & Computing, 2018.

Yao G, Lei T, Zhong J. A Review of Convolutional-Neural-Network-Based Action Recognition [J]. Pattern Recognition Letters, 2018:S0167865518302058.