# LENET-5 CONVOLUTION NEURAL NETWORK WITH MISH ACTIVATION FUNCTION AND FIXED MEMORY STEP GRADIENT DESCENT METHOD

**ZHI-HAO ZHANG[1], ZAN YANG[2*],YUAN SUN[1],YANG-FAN WU[3],YI-DAN XING[1]**

[1]Department of Electronics and Information Engineering, Tongji Zhejiang College, Jiaxing, China
[2]Department of Science, Tongji Zhejiang College, Jiaxing, China
[3]Department of Economics and Management, Tongji Zhejiang College, Jiaxing, China
E-MAIL: yangzan953@163.com

**Abstract:**

**Convolutional neural network is the most important algorithm in the field of deep learning. The traditional convolution neural network usually uses Sigmoid or Relu as the activation function, but the two sides of Sigmoid are saturated, and Relu has a dead zone, which is very easy to cause gradient disappearance and gradient explosion. In this paper, Mish activation function is introduced into LENET-5 convolutional neural network, which overcomes the shortcomings of traditional activation function. At the same time, the fixed memory step gradient descent method is used to replace the gradient method of the optimization part, which improves the global convergence of the algorithm.**

**Keywords:**

**LENET-5 convolution neural network; Mish activation function; Fixed memory step gradient descent method**

## 1. Introduction

There are lots of researches about convolutional neural network such as literature [1-3]. It is a deep neural network based on the improvement of biological cerebral cortex structure. The network complexity is reduced by three methods: local receptive field, weight sharing and down sampling. model. The LENET-5 convolutional neural network is a typical convolutional neural network proposed by LeCun et al. It has a wide range of applications in image classification. The LENET-5 neural network generally uses the Relu function or Sigmoid function as its activation function. Refer to literature [4-7], we know that for Sigmoid function, it is easy to cause gradient explosion or gradient dispersion during the iterative update of deep neural network parameters. For Relu function, when a relatively large gradient flows into the neuron, the neuron dies and there is no more activation. appear. The neural network accuracy will be poor when the network parameters are not set properly. At the same time, for the traditional convolutional neural network, the parameter iterative strategy is the neural network error reverse layer-by-layer derivative propagation. The optimization method are general gradient descent method and other optimization methods based on gradient iterative, mostly rely on the current point, lack of globality.

In response to above problems, this paper makes improvements. The innovations of this paper are as follows: In view of the shortcomings of the traditional activation function, this paper is inspired by the literature [8] to introduce a new activation function Mish. Mish activation function has a Sigmoid activation function curve smoothing, and its first and second derivatives are everywhere, and Relu function has lower bounds. The upper bound has good sparsity, prevents over-fitting of the network, improves the computational complexity of Sigmoid function, the disadvantages of gradient explosion and gradient dispersion, and the death of neurons when Relu function faces large gradients. The two activation functions greatly improve the performance of the convolutional neural network.

To solve the problem of the traditional gradient descent method, this paper is inspired by the literature [9], introduces the fractional order descent method with fixed memory step to iterate the neural network parameters, which relies on the gradient iterative process. All the points in the middle increase the memory and globality of the gradient value correction, which increases the accuracy of the neural network classification.

## 2. LENET-5 Convolutional Neural Network Based on Fractional Step Descent Method Optimization

The LENET-5 neural network is a classical convolutional neural network proposed by Y. Wei, Y. Kang, W. Yin, and Y. Wang in Design of generalized fractional order gradient descent method. The LENET-5 convolutional neural network consists of seven layers, including three

978-1-7281-4242-5/19/$31.00 ©2019 IEEE

convolutional layers, two pooling layers, one fully connected layer, and one output layer. Since the operation performed at each neuron node is a linear operation, the nonlinear function cannot be fitted, and nonlinear operation is unable to be performed. For solving this problem, it's generally necessary to add an activation function between the neural network layers, activate the neuron through the function mapping, and improve the ability of generalization and dealing with nonlinear problems for neural network.

In the LENET-5 convolutional neural network, common activation functions are Sigmoid function and Relu function, their definitions are as shown:

$$\upsilon(z)=\frac{1}{1+\mathrm{e}^{-z}},\quad R(x)=x\cdot u\left(x\right),$$

where $u(x)$ is the unit step function and $u\left(x\right)=\begin{cases}1, x\geq 0\\ 0, else\end{cases}$.

Sigmoid function, also called the logic function, returns the result of the neural network operation as the classification probability. When the neuron is mapped by Sigmoid, it will be divided into positive class when the value is bigger than 0.5, and the inverse is divided into anti-class. The advantage of Sigmoid function is that its function and derivative curves are smooth and can be everywhere. However, Sigmoid function tends to be saturated on both sides. As the number of the neural network layers increases, when the initial weight is too large, the gradient value may increase gradually and cause a gradient explosion. When the initial weight is too small, the gradient value may decrease gradually and cause a gradient dispersion, both of which have caused great errors in prediction accuracy of neural networks. The gradient of the Relu function won't decay when the value of the independent variable is bigger than zero, thereby effectively avoid the problem of gradient dispersion. At the same time, when the input value is negative, Relu function will be converted to 0, which will cause the sparseness of the network, and reduce interdependence between parameters, which alleviates the problem of over-fitting. This sparsity also causes drawbacks. When a neuron flows into a relatively large gradient value or the input value is less than 0, the neuron no longer has an activation function after the network parameter is updated, that is, the neuron dies. This will cause too many features to be masked, reduce the effective capacity of the model, and fail to learn effective features.

To solve this problem, this paper is inspired by the literature [8], introducing a new activation function Mish, which can improve the shortcomings such as gradient explosion, gradient dispersion, large calculation and being easy to cause death of the neuron when Relu function flows into large gradient. Mish function has the following properties: lower bound, monotonous, and differentiated everywhere. It is defined as:

$$Mish(x)=x\cdot\tanh(softplus(x)),$$

where

$$softplus(x)=\ln\left(1+e^{x}\right)$$

these functions pair as shown in fig.1:

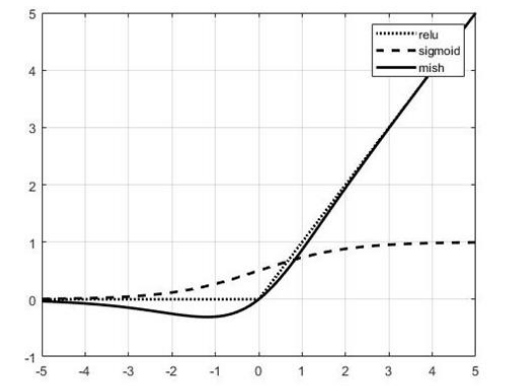

**Fig.1** Comparison of different activation function curves

It can be seen that as the function value increases, Mish function has a linear property similar to Relu in the first quadrant of the coordinate axis, and in the negative half axis of the coordinate axis, it can be clearly seen that the independent variable value of Mish function tends to be negative infinity, its function value approaches zero. At the same time, Mish function will have a buffer zone when the negative half-axis approaches the origin, and the smoothness can be differentiated everywhere, which effectively avoids the defect that neurons die in the negative semi-axed caused by Relu function and makes sure that still has certain sparsity. At the same time, the positive half-axis of Mish function has linear properties, and the first derivative and the second derivative curves are smooth, so the gradient is easy to be calculated, which simplifies the operation cost of the neural network and improves the computational efficiency of the neural network.

The commonly used parameter optimization methods for convolutional neural networks are usually based on gradient descent method or stochastic gradient descent method. Through a large number of studies and tests, gradient method is one of the most effective and fastest methods for obtaining optimal parameter solutions. Because the traditional gradient descent method has the disadvantages of relying only on the current point, lacking of globality, and slow convergence speed as the learning rate is set incorrectly. In response to these shortcomings, this paper is inspired by the literature [9], introducing the Caputo-defined fractional step-down method with fixed memory steps to optimize the neural network loss function for neural network parameter iteration, it has the advantages of long memory characteristics, non-locality, slow convergence rate, and the integral-after-derivative calculation form is easy to apply engineering calculation. Therefore, this method is used to optimize the convolutional neural network parameters, of which Caputo The derivative is as follows:

$$ {}_{c}^{C}\mathrm{D}_{x}^{\alpha} f(x) = \frac{1}{\Gamma(n-\alpha)} \int_{c}^{x} \frac{f^{(n)}(\tau)}{(x-\tau)^{\alpha-n+1}} \mathrm{d}\tau, $$

where $n-1<\alpha<n, n\in ¢_{+}$, $c$ is the lower boundary, $\Gamma(\alpha)=\int_{0}^{+\infty} \mathrm{e}^{-t} t^{\alpha-1} \mathrm{d}t$, $\begin{pmatrix}\alpha \\ \beta\end{pmatrix} = \frac{\Gamma(\alpha+1)}{\Gamma(\beta+1)\Gamma(\alpha-\beta+1)}, \alpha \in {}^{-}, \beta \in$ .

Because the computer is inconvenient to handle the derivative and integral operations, to facilitate computer operations, it is expanded into a series of forms:

$$ {}_{c}^{C}\mathrm{D}_{x}^{\alpha} f(x) = \sum_{i=n}^{+\infty} \begin{pmatrix}\alpha-n \\ i-n\end{pmatrix} \frac{f^{(i)}(x)}{\Gamma(i+1-\alpha)} (x-c)^{i-\alpha}. $$

Its iterative format is given by:

$$ x_{k+1} = x_{k} - \mu_{x_{k-K}} \, {}_{c}^{C}\mathrm{D}_{x}^{\alpha} f(x)\Big|_{x=x_{k}}, \quad K \in ¢_{+}. $$

This paper is inspired by the literature [9]. When the algorithm converges, it will converge to its actual limit point. The main idea of this method is called the fixed memory principle. The fractional gradient descent method iteration format can be expressed as:

$$ x_{k+1} = x_{k} - \mu \sum_{i=1}^{+\infty} \begin{pmatrix}\alpha-1 \\ i-1\end{pmatrix} \frac{f^{(i)}(x_{k})}{\Gamma(i+1-\alpha)} (x_{k}-x_{k-K})^{i-\alpha}. $$

Among them, $\mu$ is the learning rate. The iteration of the fractional step-down method can find the global minimum of the better Loss function faster, thus optimizing the training speed and accuracy of the neural network.

## 3. Simulation Study

In this paper, using Lenet-5 neural network, the Minist dataset in the library is used to complete the numerical simulation experiment in the environment of TensorFlow based on Python language and Ubuntu system. Firstly, this paper adopts the gradient descent method to the traditional Lenet-5 neural network by changing three different activation functions, Sigmoid, Relu, and Mish, performed image classification tests on the Minist dataset. All other parameters remained unchanged. A total of 10,000 trainings were performed on the Minist training set, the accuracy and loss function of the output network were output every 100 times. The output results are shown in fig.2 and fig.3.

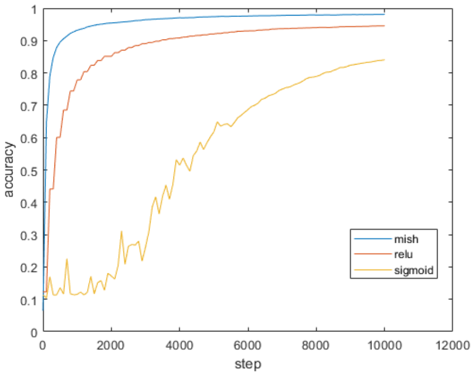

**Fig.2** Accuracy curve for each epoch model

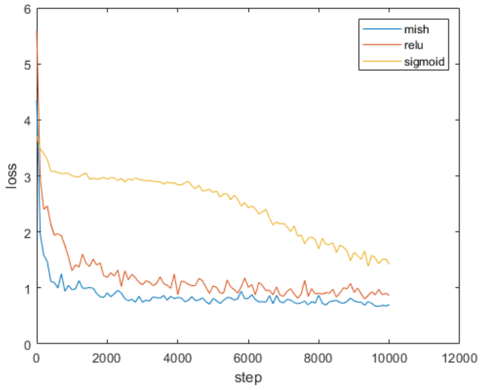

**Fig.3** Loss curve per epoch model

It can be clearly seen that after 10,000 iterations of training, the Sigmoid activation function has the lowest accuracy rate of 84.16%. And the network is difficult to converge; the network using the Relu function as the activation function has higher accuracy and faster convergence. The accuracy rate is 94.55%. The network using Mish as the activation function weakens its shortcomings by combining the advantages of the two activation functions, so the training accuracy is the highest, the maximum accuracy reaches 98.09%, which is 13.93% higher than that of Sigmoid, 3.54% higher than that of Relu. The network convergence speed is the fastest, and the highest accuracy indicates the efficient and accurate characteristics of Mish function.

At the same time, the fractional descent method with fixed memory step is compared with the traditional gradient descent method. In this paper, the gradient descent method is recorded as GD, the gradient descent method with fixed memory step is recorded as FMSGD, and $\rho_{\mathrm{s}}(\tau)$ is from literature [10] leads to the parameters describing the iterative performance of the algorithm, $\tau$ is the number of iterations of the function values. fig.4 and fig.5 show that FMSGD has a better iteration time performance curve than GD. It proves the fractional gradient descent method, which has the advantages of memory, globality and fast iteration.

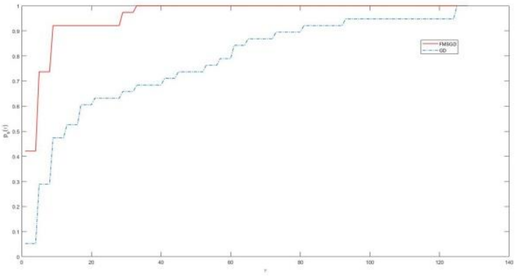

**Fig.4** Performance for CPU Time of GD and FMSGD

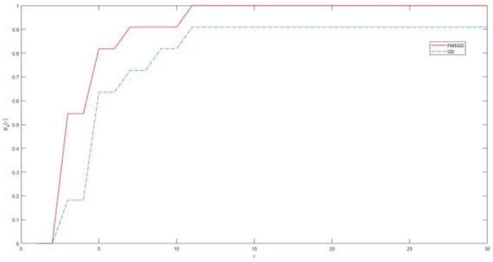

**Fig.5** Performance for NIT of GD and FMSGD

## 4. Conclusions

Combined with numerical simulation and program practice, the traditional acceleration function Sigmoid and Relu function are used to compare the fitting precision with the loss function, and the fractional step-down method of fixed memory step is compared with the traditional gradient descent method, the result shows that Mish function have better activation performance than Relu and Sigmoid functions, and the fractional descent method with fixed memory step has better global convergence and performance than the traditional gradient descent method.

## Acknowledgements

This work is partly supported by The Seventh Student Innovation Activity Project of Tongji Zhejiang College No.0719038, 2019 Zhejiang University Students Science and Technology Innovation Activity Plan Project No.2019R437005 and The Eighth Teaching Reform Project of Tongji Zhejiang College No.0119029.

## References

[1] Y. Wei, Y. Kang, W. Yin, and Y. Wang, “Design of generalized fractional order gradient descent method,” 2018.

[2] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner et al., “Gradient-based learning applied to document recognition,” Proceedings of the IEEE, vol. 86, no. 11, pp. 2278–2324, 1998.

[3] V.Nair andG.E. Hinton,“Rectifiedlinear unitsimproverestricted boltzmann machines,” in Proceedings of the 27th international conference on machine learning (ICML-10), 2010, pp. 807–814.

[4] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, “Backpropagation applied to handwritten zip code recognition,” Neural computation, vol. 1, no. 4, pp. 541–551, 1989.

[5] A. Krizhevsky, I. Sutskever, and G. Hinton, “Imagenet classification with deep convolutional neural networks,” Advances in neural information processing systems, vol. 25, no. 2, 2012.

[6] P. Dolezel, P. Skrabanek, and L. Gago, “Weight initialization possibilities for feedforward neural network with linear saturated activation functions,” IFAC-PapersOnLine, vol. 49, no. 25, pp. 49–54, 2016.

[7] S. Gomar, M. Mirhassani, and M. Ahmadi, “Precise digital implementations of hyperbolic tanh and sigmoid function,” in 2016 50th Asilomar Conference on Signals, Systems and Computers. IEEE, 2016, pp. 1586– 1589.

[8] D. Misra, “Mish: A self regularized non-monotonic neural activation function,” arXiv preprint arXiv:1908.08681, 2019.

[9] Y.Wei, Y.Kang, W.Yin, and Y.Wang, “Design of generalized fractional order gradient descent method,” arXiv preprint arXiv:1901.05294, 2018.

[10] D. Li and D. Zhu, “An affine scaling interior trust-region method combining with nonmonotone line search filter technique for linear inequality constrained minimization,” International Journal of Computer Mathematics, vol. 95, no. 8, pp. 1494–1526, 2018.