Comparative analysis of various new activation functions based on convolutional neural network

Jianhao Zhu*1, Zong Chen1

1Key Laboratory of Marine Power Engineering & Technology (Ministry of Transport), Wuhan University of Technology Wuhan, China

Email address: jhzhu@whut.edu.cn;

Abstract—With the rapid development of artificial intelligence, deep learning is developing rapidly. As the main framework of in-depth learning, convolution neural network achieved much in semantic segmentation and image classification. Based on the low speed engineering vehicle tracking system project, this paper studies the activation function, which is an important component of convolution neural network. Firstly, it summarizes the advantages and disadvantages of four traditional activation functions such as Sigmoid, and then introduces three new activation functions: Swish, FTS, and Relu-Softplus. Finally, using the TensorFlow framework, a single new activation function and a hybrid new activation function are applied to the convolution layer of the four neural networks constructed, and the image recognition comparison is performed in the data set CIFAR-10. It is concluded that the recognition precision of mixing the new activation functions is higher than that of using a new activation function alone, and the average time is shorter and the convergence speed is faster.

1. BACKGROUND

We are now living in a sci-fi movie-like world: AlphaGo has defeated many of the world's top Go players; artificial intelligence can automatically translates one language into another; mobile phones can recognize our voices. These all show that artificial intelligence is playing a vital role in our lives. At the same time, deep learning also contributed a lot. Many scholars believe that it is an innovative technology and even a breakthrough in the past ten years. Activation functions play an important role in deep learning. Various traditional activation functions proposed in the early days are used widely but still in a limited manner. Therefore, it is necessary to combine several traditional activation functions to construct a new activation function to make up for the shortcomings.

2. THE ROLE OF APPLYING ACTIVATION FUNCTIONS

The activation function is the link between the perception and the neural network. It is the core of every neural network and plays an important role in the network training. Its function is to convert input signals into output ones. The neural network uses the activation function to introduce non-linear factors to better classify the data. If a linear activation function is adopted, despite the layer number of the network, it is still a linear mapping, which does not lead to any big change in the data and results we expect. Therefore, it is a must to use non-linear functions to solve the problems that cannot be handle by linear models, and to improve the expression ability of the neural network [1]. The output of
each convolutional layer through the activation function will become much more complicated, so the key to building a good neural network is to choose an appropriate activation function.

3. TRADITIONAL ACTIVATION FUNCTION

3.1. Sigmoid
The Sigmoid function is a very smooth activation function that is often used in neural networks. It is suitable for forward propagation and data compression, without causing large amplitude deformation. The image is S-shaped, which is similar to the biological S-shaped growth curve. Therefore, it is also called S-shaped growth curve.

Many people nowadays no longer choose to use it, which is because that when it is back-propagated, the deeper the layers, the smaller the propagation value. The value eventually approaches 0, resulting in gradient dispersion and training failure. In addition, its average value is not 0, which will cause the back propagation gradient to be updated in the positive direction when the input is positive, and vice versa. It seems to be a kind of "bundled transaction", and is not conducive to convergence. In addition, the Sigmoid function contains a complicated power operation, which slows down the training process. Its mathematical expression is:

\[ \text{Sigmoid}(x) = \frac{1}{1 + e^{-x}} \]

3.2. Relu
In the history of neural network development, the Sigmoid function has been discovered and used for a long time. But the Relu function is the activation function recognized to be the most effective and widely used one nowadays [2]. Since it has a strong convergence rate during gradient descent, it is considered to be linear (in fact, Rule is still nonlinear) and unsaturated. In the SGD algorithm, the Relu function is faster and simpler than other activation functions, which can greatly improve the efficiency of machine operation. It also solves the problem of gradient dispersion and slow operation, and is suitable for back propagation.

However, problems still exist. When \( x < 0 \), the function value is always 0. Some neurons will never be activated, and the back-propagation gradient will never be updated, as if Relu has lost its activity. The average output value of the Relu function is not 0, which is not conducive to convergence (similar to the Sigmoid function). The Relu function cannot compress the data, and the data amplitude will continue to expand as the number of layers increases. Its expression is:

\[ \text{Relu}(x) = \begin{cases} x, & x > 0 \\ 0, & x < 0 \end{cases} \]

3.3. Tanh
The Tanh function is closely related to the Sigmoid function. The expression can be written as \( \text{Tanh} (x) = 2\text{sigmoid}(2x) - 1 \). In this way, there will also be problems of gradient disappearance and heavy calculation. The Tanh function is usually applied in the case of large feature differences, which can be
continuously increased during the training process. The mean value of Tanh is 0. So its effect is better than Sigmoid in practice, with less iteration times and faster convergence speed. Its expression is:

\[
Tanh (x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}
\]

3.4. Softplus

The image of the Softplus function is similar to that of the Rule function, but is smoother. Because the Softplus function requires logarithmic operation, it is not as simple as the Rule function in use. In addition, it is experimentally proven that the Rule function performs better than the Softplus function. Its expression is:

\[
Softplus (x) = \ln (1 + e^x)
\]

4.NEW ACTIVATION FUNCTION

4.1. Swish

The Swish function [3] is a new type of activation function proposed by Google in 2017, which has caused a huge sensation in its presence. Swish function expressions are not obtained through theoretical reasoning, but through experimental application of small-scale exhaustive search and large-scale RNN controller application. A large number of experiments prove that its effect is much better than the Relu function. Its expression is:

\[
Swish(x) = \frac{x}{1 + e^{-\beta x}}
\]

Many experiments have confirmed that when the \( \beta \) value is 1, the gradient is consistent with Relu. This is the most suitable for reinforcement learning training. Swish derivative expression is:

\[
Swish'(x) = \frac{1 - Swish(x)}{1 + e^{-x}} + Swish(x)
\]

4.2. FTS

The full name of FTS function [4] is Flatten-T Swish. Hock Hung Chieng proposed the FTS function in 2018 based on the Swish function. Its expression is:

\[
FTS(x) = \begin{cases} 
\frac{x}{1 + e^{-\beta x}} + T, & x > 0, \ \beta = 1 \\
T, & x < 0
\end{cases}
\]

The proposition of FTS function was inspired by Swish. At first, the expression was Relu \((x)\)-Sigmoid \((x)\). In this case, when \( x < 0 \), the value of the function is still 0, as well as the reverse gradient. The inactivity problem is still not solved. Therefore, we decided to add a constant \( T \) to Relu \((x)\)-Sigmoid \((x)\). Generally, the value of \( T \) is negative, and the neural network benefits from the negative form.

We can obtain the FTS derivative expression:

\[
FTS'(x) = \begin{cases} 
\sigma(x)(1 - g(x)) + g(x), & x > 0 \\
0, & x < 0
\end{cases}
\]

In the formula,

\[
\sigma(x) = s-sigmoid(x) \\
g(x) = x \sigma(x)
\]

4.3. Relus-softplus

The Relus-softplus function [5] was proposed by Qu in 2017 by analyzing the advantages and disadvantages of various traditional activation functions and combining the Relu function with the Softplus function (which can suppress the excessive sparsity of Relu). Its expression is:

\[
Relus-softplus (x) = \begin{cases} 
\ln (1 + e^x) - \ln 2, & x < 0 \\
x, & x \geq 0
\end{cases}
\]
The derivative expression of the function is:

$$\text{Relu-softplus}'(x) = \begin{cases} 1, & x > 0 \\ \frac{1}{1 + e^{-x}}, & x < 0 \end{cases}$$

The derivative structure is simple in structure and is easy to calculate, which greatly reduces the time required for derivation of back propagation and enhances the calculation speed.

It can be seen from Fig. 2 that the Swish function image is smoother and non-monotonic. When applied to local corresponding normalization, it can deliver better effect than Relu. The T in the FTS function is taken as -0.2 and -0.4 respectively. When x > 0, the image is parallel to Swish. The Softplus function in the Relu-softplus image well suppresses the sparseness of Relu while maintaining its characteristic of fast convergence. When x < 0, the function value is no longer always 0, and so is the reverse gradient value, which solves the problem of Relu inactivation.

![Figure 2 three new activation function images](image)

From Fig. 3, we can see that the derivative of FTS is consistent and shares the same properties with the Swish derivative when x > 0. FTS has a strong sparseness in the back-propagation process, and only responds to a very small number of input signals while ignoring most of them deliberately. This helps to greatly improve the operating efficiency and accuracy.

![Figure 3 derivative images of three new activation functions](image)

5. EXPERMINT

5.1. Model selection and environment configuration

In the experiment, we selected Ubuntu 16.04 operating system and used NVIDIA Quadro P1000 to speed up training. We wrote programs using Python and built a deep learning model by using the Tensorflow framework.

The convolutional neural network structure adopts a traditional convolutional neural network—VGGNet19 [6-7]. It has 19 layers, with a simple structure, strong expandability, and good generalization ability. VGGNet19 first trains a simple network B, and then initializes the subsequent complex network using the weight of B, which greatly accelerates the convergence speed. Due to that VGGNet has a very deep network and a small convolution for achieving implicit regularization, although it requires many model parameters, it only needs a small number of iterations to realize
convergence. Transforming a large convolutional layer into two small layers in series can help to speed up the training process and strengthen the ability of feature learning. VGGNet19 uses the method of Multi-Scale to enhance data, increase data amount, and prevent the model from overfitting. VGGNet19 boasts excellent classification performance and is not complicated; therefore it is one of the classic convolutional neural networks.

The data set adopts the classic CIFAR-10 [8-9]. It contains 50,000 training sets and 10,000 test sets. A total of 10 types of pictures are marked without overlapping. No two types of objects will appear in the same picture at the same time. This data set is universal, so it was chosen for this experiment.

Training and verification were performed using Swish, FTS, Relu-softplus, and a combination of the three. The combination of the three (hereinafter referred to as mix) means to evenly allocate three new functions on the convolutional layers of the entire neural network. Batch size is set to 32 (Since the structure of the VGGNet convolution model is very large, if the batch size used is also large, then the GPU video memory will be insufficient). Each type is trained for 100 times, and the recognition accuracy rate is counted from the 11th time. The first 10 rounds can be regarded as the warm-up of the program to avoid problems such as memory loading and cache hits.

**TABLE 1. Average back propagation time of each network**

| Function name | FTS  | Swish | Relu-Softplus | mix   |
|---------------|------|-------|---------------|-------|
| Average time consuming | 0.665 | 0.713 | 0.756         | 0.698 |

![Figure 4 change of recognition accuracy of different activation functions with training times](image)

5.2. Experimental result and analysis

From Table 1 and Fig.4, it can be seen that Relu-softplus has the lowest recognition accuracy rate (98.4%), longer average back propagation time (0.756s), and slow convergence speed. Compared with Relu-softplus, Swish has higher recognition accuracy rate (98.6%), less average time consumption (0.713s), and faster convergence speed. The recognition rate of FTS is 98.7%, higher than the first two. It is also faster in back propagation (0.665s) and convergence. The new activation function with the combination of the three has the highest recognition rate (99.2%), shorter average time for back propagation (0.698s), and faster convergence speed. Therefore, by evenly allocating the three new activation functions, we can have higher accuracy, shorter average time for back propagation, and faster convergence. This also suggests that using different activation functions at different layers for constructing a convolutional neural network may bring better results.
6. CONCLUSION
Deep learning is now being applied in all walks of life, contributing to the development of society.
Neural networks have various types, and the most widely used one is convolutional neural network.
Activation function is an important part of the convolutional neural network, and its related research
will also help to further optimize the operation rate. This paper, by comparing the performance of the
three new activation functions, namely, Relu-softplus, Swish, and FTS, and their mix, shows that the
recognition accuracy is 98.4%, 98.6%, 98.7%, 99.2%, and the average time is 0.756s, 0.713s, 0.665s,
0.698s, respectively. The data shows that the mix of three functions can deliver higher recognition
accuracy and efficiency with faster convergence speed.

References
[1] Li, M.W. (2016) Research on convolution neural network method in image classification. Nanjing
    University of Posts and Telecommunications, Nanjing.
[2] KRIZHEVSKY A, SUTSKEVER I, HINTON G E. (2012) ImageNet classification with deep
    convolutional neural networks [C] // International Conference on Neural Information
    Processing Systems. Lake Tahoe: Springer: 1097–1105.
[3] Prajit Ramachandran , Barret Zoph, Quoc V. Le. (2017) SEARCHINGFOR ACTIVATION
    FUNCTIONS. arXiv :1710.05941:2-5.
[4] Hock Hung Chieng, Noorhaniza Wahid, Ong Pauline, Sai Raj Kishore Perla. Flatten-T Swish: a
    thresholded ReLU-Swish-like activation function for deep learning. arXiv :1812.06247,2018:76-82.
[5] Qu, Z.L., Hu, X.F. (2017) Research on convolution neural network based on improved activation
    function. Comput. Technol. Dev.,27(12):79-79.
[6] Huang, W.J., Tang, Y. (2017) Tensor flow practice. Electronic Industry Press.
[7] Wang, Y., Zhang, H.J., Huang, H.X. (2019) A survey of image semantic segmentation algorithms
    based on deep learning. Appl. Electron.Technol.,45(06):24-24.
[8] Yu, P., Zhao, J.S., Zhang, J. (2015) Image recognition of convolutional neural network based on
    nonlinear correction function. Science Technology and Engineering.,15(34).
[9] Xu, S.W., Chen, S.Y. (2018) Image classification method based on deep learning. Appl.
    Electron.Technol.,44(06).