Research on Approximate Fitting Method of Activation Function Based on Gradient Equalization in Deep Learning

Zhichen Wang¹²*, Hongliang Li¹

¹Jiangnan Institute of Computing Technology, Wuxi, Jiangsu, 214083, CHINA
²Officers College of PAP, Chengdu, Sichuan, 610213, CHINA
wangzhch@mail.ustc.edu.cn

Abstract. Activation function plays an important role in neural network model. Neural networks can express more complex situations by introducing nonlinearity due to activation functions. In this paper, an approximate fitting method of activation function based on gradient bisection is proposed. The algorithm can fit the activation function curve well. Through experiments, the storage capacity can be reduced by one time compared with the traditional approximate fitting method, but the accuracy of the model is not affected.

1. Introduction

Activation function plays a very important role in neural network model to learn and understand very complex and nonlinear functions. It introduces the nonlinear characteristic into the neural network, and its main purpose is to convert the input signal of a node in the neural network model into an output signal. The output signal is now used as the input of the next layer of the neural network model, namely the sum of the product of the input (X) and their corresponding weight (W), and the activation function is applied to it to obtain the output of this layer and feed it as the input to the next layer.[1][2][3]

If the activation function is not used in the neural network, the output signal will be only a simple linear function. Linear function is a polynomial with limited complexity and less ability to learn complex function maps from data. A neural network without activation functions would be just a linear regression model, with limited power, and in most cases not perform well. We want neural networks to be able to learn and compute situations that are much more complex than linear functions. Without activation functions, the neural network will not be able to learn and simulate other complex types of data, such as images, video, audio and voice. That's why we use artificial neural networks, like deep learning, to understand complex things, nonlinear problems that have a lot of hidden layers between them, and that can also help us understand complex data. Because nonlinear functions have curvature. A neural network model that can learn and represent almost anything, and arbitrarily complex functions that map inputs to outputs, means they can compute and learn any function. [4][5] Almost any process we can think of can be represented as a function computation in a neural network. All of this requires the application of activation functions in order to make the network more powerful, to increase its ability to learn complex things, complex form data, and complex mapping of arbitrary functions that represent nonlinearity between inputs and outputs. [6][7] Therefore, using nonlinear activation functions, it is possible to generate nonlinear mappings between inputs and outputs. Another important feature of activation functions is that they should be distinguishable. We need to do this in
order to perform a reverse optimization strategy when advancing backwards in the network to calculate the error (missing) gradient relative to the weight, and then optimize the weight accordingly using gradient descent or any other optimization technique to reduce the error.[8][9]

In practice, due to the complexity of the activation function, it is generally necessary to adopt some approximate means for approximation fitting, and use multi-segment lines to represent the activation function, so as to reduce the computational complexity. Generally, the table lookup method and Taylor expansion method are used for approximate calculation. Among them, with the expansion of Taylor's formula, the operation time will be longer and longer, and the precision is not so good. In the case of high time requirement, table lookup method has great advantages.

The table lookup method generally divides the definition domain or the value domain of the function evenly and calculates the corresponding value of the function. These values are then stored in a data table. When used, the corresponding interval is obtained by looking up the table according to the input value, and the corresponding function value is calculated by taking the line within this interval. The actual fitting effect is shown in Fig.1(a), where a part of the Sigmoid function is selected for fitting. In the method of uniform segmentation of the defined domain (called Method 1 in the table below), we can see that the variation of the function value is greatly different in different defined domain intervals. In the interval where the function value changes rapidly, the fitting effect is poor. As the function value changes slowly, the fitting effect gradually improves.

![Fig.1(a) the method of uniform segmentation of the defined domain, (b) the method of uniform range segmentation](image)

When the method of uniform range segmentation (method 2 in the following table) is adopted, as shown in Fig.1(b), the fitting effect is better in the interval where the range of function changes rapidly, and the fitting effect gradually deteriorates as the change of function value becomes slower.

From the actual performance, when the table generated by the above two methods is used for approximate fitting of the activation function, the activation function needs to be segmented quite intensively in order to achieve convergence. Taking AlexNet network as an example, CIFAR-10 data set was used for the experiment. The results are shown in the Table1.

| activate function segmentation precision | original function | 1/32          | 1/64          | 1/128         | 1/256         |
|-----------------------------------------|------------------|--------------|--------------|--------------|--------------|
| uniform segmentation of the domain      |                  |              |              |              |              |
| (Method 1)                              | 81.7%            | no convergence | no convergence | 81.8%        | 81.5%        |
| uniform segmentation of range           |                  |              |              |              |              |
| (Method 2)                              | 81.7%            | no convergence | no convergence | 81.7%        | 81.5%        |

As can be seen from the table, when the approximate fitting broken line of the activation function curve is relatively small, the activation function cannot achieve the original performance, resulting in the failure of convergence during the neural network training. With the increase of approximate fitting broken lines, the neural network can converge again during training. At the same time, the training precision of the neural network can be quite accurate as that of the original network. It can be seen that the actual performance of the activation function is directly related to the fitting precision of the approximate fitting broken line.
But it can also be seen that with the increase of approximate fitting broken lines. Although the performance of the activation function can be improved, the storage capacity is also improving. Therefore, the approximate fitting method based on gradient equalization proposed in this paper can better balance the contradiction between activation function performance and storage capacity.

2. Approximate Fitting Method of Bisector Broken Line

In the method of bisector fitting, the curve is bisector along the x-coordinate and the y-coordinate value of the corresponding segmentation point is obtained. According to the coordinate values of the two points, the expression of the line segment between the two points is calculated. Connect each line segment, that is, the approximate fitting broken line of the curve. For example, Sigmoid function is divided into 10 broken lines in the range from 0 to 10, and each broken line has 1000 sampling points for approximate fitting. The algorithm is as follows:

```plaintext
for x ← 0 to 10 do
    y[i] ← sigmoid(x)
end
for x1 ← 0 to 10000 do
    j ← int(x1/1000)
    y1[j] ← (y[j+1]-y[j])/(x1/1000-j)
end
```

![Fig.2(a)](image)

**Fig.2(a)** approximate broken line, **(b) The Sigmoid function defines the domain bisecting absolute error**, **(c) Sigmoid function range bisected absolute error**

The approximate broken line is shown in Fig2(a). In this fitting, only 10 straight lines participate in the fitting, and the curve has obvious edges and corners, so it can be clearly seen that there is an error. The specific absolute error curve and relative error curve of the error are shown in Fig2(b).

It can be seen from Fig2 that there is an error between the fitted multi-segment broken line and the original curve. The error reaches its maximum in the middle of the two interval points. At the same time, when the input value is small, the fitting error is large. With the increase of input value, the error generally decreases. It can be seen that this fitting method can maintain good accuracy and better approximate the original curve when the input value is large. However, when the input value is small, the curve cannot be well fitted due to the large error. The approximation obtained by fitting has a large error with the original curve. This error has a great impact on the neural network model, because most of the input values as activation functions in the neural network are concentrated in the area with small input values, and only a small part of the input values will be relatively large. As a result, the activation function curve approximated by this method has a large error in the actual neural network.

3. Approximation Algorithm Based on Gradient Bisection

Aiming at the defect of using the traditional method of equipartition approximation to fit the activation function, this paper improves the algorithm and proposes a new approximation algorithm based on gradient bisection. The main problem of this algorithm is that when the input value of the activation function is small, the performance of the activation function degrades due to the large error of the output value, which leads to the non-convergence of the model. In this paper, the Sigmoid function is similarly divided into 10 broken lines within the range of 0 to 10, and each broken line has 1000 sampling points for approximate fitting. The algorithm is as follows:
\[ y \leftarrow f(x) \leftarrow \text{sigmoid}(x) \]
\[ y' \leftarrow f'(x) \leftarrow (\text{sigmoid}(x))' \]
\[ y_1 \leftarrow f^{-1}(y') \]

for \( x_1 \leftarrow 1 \) to \( 10 \) do
\[ y_1(x_1) \leftarrow f^{-1}((x_1/10)*0.25) \]
end

for \( x_2 \leftarrow 1 \) to \( 10 \) do
\[ y_2(x_2) \leftarrow \text{sigmoid}(y_1(x_1)) \]
end

for \( x_3 \leftarrow 0 \) to \( 10000 \) do
\[ i \leftarrow \text{int}(x_3/1000) \]
\[ y_1[i] \leftarrow (y[i+1]-y[i])/(x_3/1000-i) \]
end

The curve fitted by this algorithm is shown in Fig3(a). It can be seen from Fig3(a) that when the input value is small, the fitted broken line has been significantly improved compared with the original broken line, which can better represent the actual activation function curve. To solve the original broken line fitting method in the input value is small when the larger error defects.

Fig.3(a) the curve fitted by our algorithm, (b) the gradient of the Sigmoid function

This algorithm does not divide the curve according to the input value, but first calculates the gradient curve of the function, as shown in Fig3(b). As can be seen from Fig3(b), when the input value of the Sigmoid function is small, the gradient of the function is large, that is, the change of the function in this area changes rapidly with the input value. As the input increases, the gradient descends quickly. As can be seen from Fig3(b), after the input value exceeds 6, the gradient of the function is close to 0, with little change. Therefore, the segmentation method based on gradient change can be used to allocate more linear segments for fitting when the input value is small, so as to improve the final fitting accuracy.

The following Fig4 shows the absolute error and relative error curves of the broken line fitted by the algorithm and the actual activation function curve. It can be seen from Fig4 that when the input value is small (<4), both the absolute error and the relative error are significantly reduced from the original bisectic line approximate fitting method. It can better reduce the actual output error of activation function when the input value is small. Of course, when the input value is large, there will be a large error peak in the output of the activation function. This is the last segment of the line involved in the approximate fitting, which spans a large range of input values. So there will be an error peak. However, this error peak has little impact on the performance of the activation function, because in the neural network, it is rare to have a large input value, and most of the input value is concentrated in a small area. In general, the error of output value is greatly reduced.
4. Experiments

In the existing neural network model, the main activation functions are Sigmoid, Softmax, Tanh and Relu. The Relu function is a linear function when the input value is positive, and 0 when the input value is negative, so the calculation is quite convenient and no fitting operation is required. While Sigmoid, Softmax and TANH are curves, which need to be fitted with line segments in actual use to reduce computation.

4.1. Different fitting intervals of the same activation function

The approximate fitting of the activation function mentioned above was carried out within the range of 0 to 10. For different approximate fitting intervals, precision experiments were carried out in combination with the neural network model. This paper takes AlexNet network as an example, adopts CIFAR10 data set and Sigmoid activation function, and conducts experiments for different approximate fitting intervals and different fitting accuracy. Results are shown in Table 2.

| Interval | ours method 1 | ours method 1 | ours method 1 | ours method 1 |
|----------|---------------|---------------|---------------|---------------|
| 0-10     | no convergence | no convergence | 81.5%         | 81.6%         |
| 0-8      | no convergence | no convergence | 81.6%         | 81.7%         |
| 0-6      | 81.5%          | no convergence | 81.6%         | 81.5%         |
| 0-10     | no convergence | no convergence | 81.5%         | 81.4%         |
| 0-8      | no convergence | no convergence | 81.6%         | 81.5%         |
| 0-6      | 81.5%          | no convergence | 81.6%         | 81.4%         |

It can be seen from the data in the table that when using methods 1 and 2 to approximate the Sigmoid function, more broken lines are selected to participate in the fitting, and the original function can be well fitted in different fitting intervals, so that the neural network training can converge to a certain accuracy. Neural network training cannot converge when the broken lines involved in fitting are reduced. It can be seen that the neural network model can be converged when the polylines are small (64 polylines). At the same time, when the fitting interval is 0-6, using 32 broken lines to participate in the fitting can make the neural network converge. It can be seen that due to the particularity of the function Sigmoid, when the input value of the function is above 6, the transformation of the function value is very small. Therefore, even if only 32 segments of broken lines participate in approximate fitting, the fitting accuracy of the 0-6 interval can meet the training requirements of the neural network, and the neural network can still achieve convergence.

4.2. Influence of different function fitting accuracy in the same network

The influence of different fitting intervals and different fitting accuracy on the accuracy of neural network model when CIFAR10 dataset and Sigmoid activation function are used in AlexNet network is discussed above. This section discusses different activation functions and different fitting accuracy
in the same network. BiLSTM network and Conll2003 data set were used in the experiment. The results are shown in Table 3.

| method 1 | original function | 1/32 | 1/64 | 1/128 | 1/256 | 1/512 |
|----------|-------------------|------|------|-------|-------|-------|
| sigmoid  | 91.05%            |      | no convergence |     |       | 91.11% | 91.07% | 91.06% |
| tanh     | 91.05%            |      | no convergence |     |       | 91.03% | 91.05% | 91.06% |
| method 2 | original function | 1/32 | 1/64 | 1/128 | 1/256 | 1/512 |
| sigmoid  | 91.05%            |      | no convergence |     |       | 91.1%  | 91.04% | 91.05% |
| tanh     | 91.05%            |      | no convergence |     |       | 91.01% | 91.06% | 91.05% |
| ours     | original function | 1/32 | 1/64 | 1/128 | 1/256 | 1/512 |
| sigmoid  | 91.05%            |      | no convergence | 91.11% | 91.07% | 91.06% |
| tanh     | 91.05%            |      | no convergence | 91.03% | 91.05% | 91.06% |

As can be seen from the experimental results in the table, when the approximate fitting method of methods 1 and 2 is adopted, when there are more broken lines involved in the approximate fitting, the selection of Sigmoid function and Tanh function can achieve convergence state of the neural network model. As the number of broken lines involved in approximate fitting decreases, the neural network model fails to converge in both methods 1 and 2. When the method presented in this paper is adopted, it can be seen from the third table that the neural network can still converge to the original accuracy when 64 broken lines are used for fitting. Compared with the traditional methods 1 and 2, the size of the table can be reduced by half when the activation function is approximated by the table lookup method.

5. Summary
This paper analyzes the approximate fitting operation in the use of activation function and finds out that the error mainly concentrates in the area with small input value. Because of these errors, the performance of the activation function is reduced and the neural network model cannot converge in the training. In order to achieve the convergence of the model, more line segments must be used in the approximate fitting operation to increase the storage capacity of the system. In order to reduce the storage capacity, this paper adopts gradient equalization method to determine the starting and ending points of each broken line, so that the input value is small and the output value error of the activation function is small. Although there is a relatively large error in the output value of the activation function when the input value is large, it has little impact on the performance of the activation function in general. The activation function curve approximated by this algorithm can guarantee the accuracy of neural network model and reduce the storage capacity significantly. The contradiction between activation function performance and storage capacity is resolved. Through experiments, it can be seen that when different functions are selected in different networks, the table lookup method can be reduced by more than half compared with the traditional method. When Sigmoid function is selected, when the fitting interval is 0-6, the size of table pair can be reduced to 1/4. This shows that the method in this paper can indeed use smaller tables to approximate the table fitting activation function, and the fitting accuracy can ensure the convergence of the neural network during training.

References
[1] Ko J H, Mudassar B, Na T, et al. Design of an Energy-Efficient Accelerator for Training of Convolutional Neural Networks using Frequency-Domain Computation[C]/ Acm/edac/ieee Design Automation Conference. IEEE, 2017:1-6.
[2] Xu K, Qin M, Sun F, et al. Learning in the Frequency Domain[J]. 2020.
[3] Rippel O, Snoek J, Adams R P. Spectral Representations for Convolutional Neural Networks[J]. 2015.
[4] Maini P, Kolluru K, Pruthi D, et al. Why and when should you pool? Analyzing Pooling in Recurrent Architectures[J]. 2020.
[5] Merity S, Keskar N S, Socher R. Regularizing and Optimizing LSTM Language Models[J]. 2017.
[6] Khandelwal U, He H, Qi P, et al. Sharp Nearby, Fuzzy Far Away: How Neural Language Models Use Context[J]. 2018.
[7] Yang Z, Dai Z, Yang Y, et al. XLNet: Generalized Autoregressive Pretraining for Language Understanding[J]. 2019.
[8] Yang Z, Yang D, Dyer C, et al. Hierarchical Attention Networks for Document Classification[C]// Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. 2017.
[9] Guo Q, Yu Z, Wu Y, et al. Dynamic Recursive Neural Network[C]// 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2020.