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

Lou Yaqin
College of Chemistry and Chemical Engineering
Southeast University
Nanjing, China
e-mail: 1850131001@qq.com

Wang Yizhou
College of Computer Science and Information Engineering
Shanghai Institute of Technology
ShangHai, China
e-mail: 1025190165@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 nonlinear 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. 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.
(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 widely used, and the feedforward neural network model is used as the default activation function. The activation function with tanh(x)[10] composite has more obvious gradient changes than the activation function with arctan(x)[10] as the composite, so it can converge faster during network training.

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

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.

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

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


