Optimization of Gradient Vanishing Problem in Deep Neural Networks Based on Attention Mechanism

Wu Nannan

The Open University of Putian, Putian, Fujian Province, 351100, China

Keywords: Deep Neural Network; Gradient Vanishing; Attention Mechanism; Optimization Algorithm

Abstract: With the wide application of Deep Neural Network (DNN) in many fields, the problem of gradient vanishing has become a key obstacle to its performance improvement. This article focuses on how to optimize DNN by attention mechanism to solve the problem of gradient vanishing. In this study, the theoretical basis of DNN, gradient vanishing and attention mechanism is analyzed, and an optimization algorithm model based on attention mechanism is constructed. The model skillfully integrates the attention mechanism module in the hidden layer, and expounds the model architecture design and algorithm flow in detail. The experimental results show that compared with the traditional DNN model on the self-built image data set, the optimization model based on attention mechanism effectively alleviates the problem of gradient vanishing, and significantly improves the performance indexes such as gradient stability, accuracy, recall and F1 value, and has a faster classification speed. This shows that introducing attention mechanism into DNN can effectively optimize gradient propagation and provide an effective way to solve the problem of gradient vanishing.

1. Introduction

With the rapid development of artificial intelligence, DNN has made breakthrough achievements in many fields, such as image recognition, natural language processing, speech recognition and so on, with its excellent feature extraction and complex pattern recognition capabilities [1-2]. However, with the increasing number of network layers, the problem of gradient vanishing gradually becomes prominent, which seriously restricts the performance improvement and wide application of DNN [3]. The problem of gradient vanishing is mainly caused by the continuous attenuation of gradient in the back propagation algorithm when it is transmitted between network layers, which makes it difficult for the network layer near the input layer to update parameters effectively, which makes the model training in trouble and unable to fully learn the complex features in the data [4]. Many scholars have tried to alleviate this problem by improving the network structure, such as introducing residual connection or adjusting activation function, such as using ReLU function, but the effect is still unsatisfactory [5].

Attention mechanism, as a technology that imitates human attention distribution, has been widely used in DNN in recent years [6]. It can make the model automatically focus on key parts when processing information, and dynamically allocate weights, thus enhancing the transmission of important information. Introducing attention mechanism into DNN to solve the problem of gradient vanishing is expected to open up a new way [7]. On the one hand, attention mechanism can enhance the effectiveness of information transmission between different layers and avoid excessive attenuation of gradient in the process of communication [8]. On the other hand, it can adaptively adjust the attention of the network to different features and optimize the reverse propagation path of the gradient.

This article focuses on "optimization research on DNN gradient vanishing based on attention mechanism" The purpose is to analyze the optimization effect of attention mechanism on the problem of DNN gradient vanishing, and to construct a more effective DNN model through theoretical analysis and experimental verification.

DOI: 10.25236/icceme.2025.002

2. Theoretical basis

DNN consists of multiple neuron layers, including input layer, hidden layer and output layer. Each layer of neurons is connected by weight, and weighted sum and nonlinear transformation are carried out according to the input data. In the task of image recognition, the input layer receives image data, the hidden layer extracts different levels of features such as edges and textures, and the output layer obtains classification results [9]. Back-propagation algorithm is the core of DNN training. According to the error between the prediction result and the real label, it back-propagates the gradient from the output layer to the input layer to update the weight of each layer and reduce the error gradually.

In the process of DNN backward propagation, the problem of gradient vanishing is easy to appear. When some activation functions are used, such as Sigmoid function, their derivatives tend to be zero when the input value is large or small. With the increase of the number of layers in the network, the gradient is multiplied by the derivative of the activation function, which leads to the gradient near the input layer becoming extremely small, which makes the weight update of these layers slow or even stagnant. It is difficult for the model to learn effective features, which shows that the training accuracy cannot be improved and the convergence speed is extremely slow.

Attention mechanism simulates the process of human attention selection, so that the model can automatically focus on the key parts when processing input information. Its core principle is to calculate attention weights for different parts of the input, and then sum the information according to the weights [10]. Taking machine translation of natural language processing as an example, the model can allocate different attention to different positions of the source language sentences according to the currently generated translation words, focusing on the parts related to the current translation, thus generating more accurate translations. Attention mechanism can enhance the model's ability to capture and use important information, and provide a new perspective for solving the problem of DNN gradient vanishing.

3. Optimization algorithm based on attention mechanism

The DNN optimization model based on attention mechanism keeps the basic framework of traditional DNN, which is composed of input layer, multiple hidden layers and output layers. The input layer is responsible for receiving the original data, and the output layer produces the final prediction result. However, in order to effectively solve the problem of gradient vanishing, the attention mechanism module is integrated into the hidden layer. The purpose of this design is to make the model dynamically allocate attention weight and highlight key information in the process of data processing, and then optimize the reverse propagation path of gradient.

The attention mechanism module is embedded in each hidden layer. Before the input data of each hidden layer enters the conventional neural network calculation, it is input to the attention mechanism module. The module will calculate the corresponding attention weight according to the characteristics of the input data. These weights represent the importance of different parts of the input data to the current layer processing. In this way, the attention mechanism can guide the model to pay more attention to the information that is more critical to gradient propagation and feature learning, thus avoiding the problem of gradient vanishing aggravated by the loss in the process of information transmission.

Initialization: randomly initialize the weights W_i and offset b_i of each layer of DNN, and at the same time initialize the parameters in the attention mechanism module.

Forward propagation: Input data X first enters the first hidden layer. In the hidden layer, the data first passes through the attention mechanism module. Let the input data be x, and the attention mechanism module obtains the attention weight α through calculation:

$$\alpha = \mathrm{Attention}(x) (1)$$

The function here can take many forms, such as the attention calculation method based on dot product. Take simple dot product attention as an example. Firstly, input x is linearly transformed to get q,k,v (representing query, key and value respectively), namely:

$$(q=W_qx)，(k=W_kx)，(v=W_vx)(2)$$

Then calculate the attention weight α:

$$\alpha=\text{softmax}(\frac{qk^T}{\sqrt{d_k}})(3)$$

Where: d_k is the dimension of k. Then, the weighted input is input into $\alpha\cdot x$ to the conventional neural network layer for calculation, namely:

$$h=\sigma(W(\alpha\cdot x)+b)(4)$$

Where σ is the activation function -ReLU function. Repeat the above process, and the data passes through each hidden layer in turn until the output layer gets the prediction result $\hat{y}$.

Back propagation: First, calculate the loss function $L(\hat{y},y)$ of the prediction result $\hat{y}$ and the real label y. Then, according to the loss function, the gradient $\frac{\partial L}{\partial \hat{y}}$ of the output layer is calculated and propagated back to the hidden layer. In the process of hidden layer back propagation, because the attention mechanism module changes the information transmission path, the influence of attention weight should be considered in gradient calculation. For the gradient $\frac{\partial L}{\partial(\alpha\cdot x)}$ output by the attention mechanism module, the gradient $\frac{\partial L}{\partial x}$ of the input x is calculated by the chain rule, and the parameters in the attention mechanism module are updated. Finally, the back propagation gradient is continued, and the weight W_i and offset b_i of each layer are updated, so that the loss function is gradually reduced and the training and optimization process of the model is completed.

With the help of the above optimization algorithm model based on attention mechanism, it is expected to effectively alleviate the problem of gradient vanishing and improve the learning ability and performance of the model during DNN training.

4. Experimental verification

In order to verify the effectiveness of DNN optimization model based on attention mechanism (hereinafter referred to as optimization model) in solving the problem of gradient vanishing, the classic image classification task is selected as the experimental scene. The experimental data set is a self-built image data set, including 10 categories, with a total of 10,000 images, of which 7,000 are used for training, 2,000 for verification and 1,000 for testing.

The experiment is based on Python programming language, and the deep learning framework PyTorch is used to build the model. The hardware environment is a computer equipped with NVIDIA GeForce RTX 3080 GPU to speed up the model training process.

The model construction stage includes traditional DNN model (as a comparison model) and optimization model. The optimization model adopts the architecture design mentioned above, and incorporates the attention mechanism in the hidden layer.

In the parameter setting stage, the same initialization parameters are configured for the two models, in which the learning rate is set to 0.001, and Adam optimizer is selected to update the parameters. The batch size of 64 is adopted in the training process, and the total training rounds are set to 100 epoch.

The model training and evaluation process records the training loss and verification accuracy of each epoch. After the training, the test set is used to evaluate the performance of the two models, and the main evaluation indicators cover accuracy, recall and F1 value.

Table 1 shows the changes of the gradient value of a certain layer near the input layer in the training process of the two models, and the results are shown in Table 1. The average gradient of traditional DNN model decreased sharply in the late training period, and the gradient disappeared obviously. However, the average gradient of the optimized model is always kept at a relatively stable level, which effectively alleviates the problem of gradient vanishing.

Table 1 Gradient Mean Variation Near Input Layer

Epoch	Traditional DNN Gradient Mean	Optimized Model Gradient Mean	Gradient Stability Score (1-10, 10=Best)
10	0.052	0.078	8
20	0.031	0.075	8
30	0.018	0.072	7
40	0.009	0.070	7
50	0.004	0.068	7
60	0.002	0.066	7
70	0.001	0.064	6
80	0.0005	0.062	6
90	0.0002	0.060	6
100	0.0001	0.058	6

The performance of the two models is evaluated on the test set, and the results are shown in Table 2. The optimized model is significantly superior to the traditional DNN model in accuracy, recall and F1 value. This shows that the optimized model not only effectively solves the problem of gradient vanishing, but also improves the overall performance of the model, so that it can identify image categories more accurately in image classification tasks.

Table 2 Performance Comparison

Model	Accuracy (%)	Recall (%)	F1 Score	Misclassification Rate (%)	Classification Speed (images/sec)
Traditional DNN	72.5	70.8	71.6	27.5	500
Optimized Model	85.3	83.7	84.5	14.7	600

To sum up, the effectiveness and superiority of the optimization model based on attention mechanism in solving the problem of DNN gradient vanishing are verified by experiments, which provides strong support for the performance improvement of DNN in practical applications.

5. Conclusions

This article focuses on the optimization of DNN gradient vanishing based on attention mechanism, and has achieved a series of valuable results. On the theoretical level, this article discusses the principle of DNN, the origin of gradient vanishing, and the characteristics and potential advantages of attention mechanism.

Based on theoretical analysis, a DNN optimization model with attention mechanism is constructed. The model embeds the attention mechanism module in the hidden layer, dynamically allocates the weights of different parts of the input data, guides the model to focus on key information, and optimizes the reverse propagation path of the gradient. In the algorithm design, the steps of initialization, forward propagation and backward propagation are planned to ensure the effective operation of the model.

The results verify the effectiveness of the model. From the perspective of gradient change, compared with the traditional DNN model, the average gradient near the input layer of the optimized model is more stable in the training process, which significantly alleviates the problem of gradient vanishing. In terms of performance indicators, the optimized model is far superior to the traditional model in accuracy, recall and F1 value, with lower misclassification rate and faster classification speed. This shows that the optimized model not only solves the problem of gradient vanishing, but also improves the performance of the model in image classification tasks.

However, there are still some limitations in this study. Although the self-built data set can meet the basic needs of the experiment, compared with the large-scale public data set, the diversity and scale are insufficient. In the future, we can consider further verifying the universality of the model on more challenging data sets. In addition, the introduction of attention mechanism increases the computational complexity of the model, and subsequent research can explore more efficient implementation methods or optimization strategies of attention mechanism to balance the performance of the model and the consumption of computing resources.

References

[1] Ma Yuge, Su Chaoguang, Ding Renwei. A multi-attribute identification method for low-order faults based on LOFUnet deep convolutional neural network[J]. Computing Techniques for Geophysical and Geochemical Exploration, 2024, 46(3):272-283.

[2] Wang Ziwei, Lu Jiwen, Zhou Jie. Binary neural network based on adaptive gradient optimization[J]. Acta Electronica Sinica, 2023, 51(02):257-266.

[3] Gai Jianxin, Xue Xianfeng, Wu Jingyi. Cooperative spectrum sensing method based on deep convolutional neural network[J]. Journal of Electronics & Information Technology, 2021, 43(10): 2911-2919.

[4] Liang Yongqi, Bai Shuangcheng, Zhang Zhiyi. Research progress of neural networks combining Hamiltonian mechanics in deep learning[J]. Computer Engineering and Applications, 2025, 61(14): 20-36.

[5] Xiao Yu, Wang Jingzhong, Wang Baocheng. Chinese text classification method based on deep learning[J]. Computer Engineering and Design, 2021, 42(04):1014-1019.

[6] Qian Xiaomei, Liu Jiayong, Cheng Pengsen. Distant supervision relation extraction based on densely connected convolutional neural network[J]. Computer Science, 2020, 47(02):157-162.

[7] Zheng Feifan. Anomaly detection model based on ResNet deep neural network[J]. Journal of Network New Media Technology, 2020, 9(02):16-22.

[8] Meng Jinlong, Tang Shihua, Zhang Yan. GNSS height anomaly fitting method based on MVO optimized neural network[J]. Journal of Geodesy and Geodynamics, 2022, 42(12):1233-1238.

[9] Jiang Zichao, Jiang Junyang, Yao Qinghe. A fast solving method for differential equations based on neural network[J]. Chinese Journal of Theoretical and Applied Mechanics, 2021, 53(07): 1912-1921.

[10] Huo Aiqing, Zhang Wenle, Li Haoping. Traffic sign recognition based on SqueezeNet model combining deep residual network and GRU[J]. Computer Engineering and Science, 2020, 42(11): 2030-2036.