A Comparison of Optimizer

1.Introduction

With the development of society and the advancement of science and technology, artificial intelligence has gradually emerged, and machine learning has also been applied to more and more fields. Deep learning is a hot research direction for machine learning. It is currently widely used in the fields of picture recognition, natural language processing, robotics and control. In deep learning algorithms, the model structure and optimization method are two main factors that affect the performance of the model. In particular, optimization directly affects the speed and performance of deep learning algorithms, so it is necessary to study the optimization of deep learning algorithms[1]. This project mainly does research on the algorithms and performance of different optimizers in different deep learning tasks.

2.Problem definition

The main problem of the project research is the different performance of four optimizers, SGD, Adam, RMSprop and Nadam, on two data sets, MNIST and CIFAR10, based on convolutional neural networks(CNNs).

3.Methodology

3.1 SGD

SGD uses the gradient descent method to calculate the derivative of errors with respect to weights layer by layer and update the weight and bias of the network layer by layer. Its formula is:

$$\theta = \theta - \frac{\alpha}{B}\sum_{k=1}^{B} \nabla_{\theta} J^{(jk)}(\theta)$$

It is simple, and its gradient requirement is very low and the gradient calculation is fast. But its convergence rate is slow, and it is easy to converge to the local optimal, in some cases it may be trapped in the saddle point.

3.2 RMSprop

RMSprop accelerates gradient descent by eliminating oscillations during gradient descent. Its formula is:

$$E[g^2]_t \leftarrow \rho * E[g^2]_{t-1} + (1 - \rho) * g_t^2$$

$$\Delta\theta \leftarrow \frac{\delta}{\sqrt{E[g^2]_t + \epsilon}} * g_t$$

$$\theta_{t+1} \leftarrow \theta_t - \Delta\theta_t$$

RMSprop still depends on the global learning rate, and it is suitable for handling non-smooth targets.

3.3 Adam

Adam is essentially RMSprop with momentum term. It uses the first-moment estimation and second-moment estimation of gradient to dynamically adjust the learning rate of each parameter. Its formula is:

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t + \epsilon}} \hat{m}_t$$

$$\hat{m}_t = \frac{m_t}{1-\beta_1^t}$$

$$\hat{v}_t = \frac{v_t}{1-\beta_2^t}$$

$$m_t = \beta_1 m_{t-1} + (1-\beta_1) g_t$$

$$v_t = \beta_2 v_{t-1} + (1-\beta_2) g_t^2$$

Adam combines the advantages that Adagrad is good at dealing with sparse gradient and RMSprop is good at dealing with non-smooth targets, and calculates different adaptive learning rates for different parameters. It is suitable for most non-convex optimization problems.

3.4 Nadam

Nadam is similar to Adam with Nesterov momentum term. Its formula is:

$$\hat{g}_t = \frac{g_t}{1-\Pi_{i=1}^{t} \mu_i}$$

$$m_t = \mu_t * m_{t-1} + (1-\mu_t) * g_t$$

$$\hat{m}_t = \frac{m_t}{1-\Pi_{i=1}^{t+1} \mu_i}$$

$$n_t = \nu * n_{t-1} + (1-\nu) * g_t^2$$

$$\hat{n}_t = \frac{n_t}{1-\nu^t} \bar{m}_t = (1-\mu_t) * \hat{g}_t + \mu_{t+1} * \hat{m}_t$$

$$\Delta\theta_t = -\eta * \frac{\bar{m}_t}{\sqrt{\hat{n}_t + \epsilon}}$$

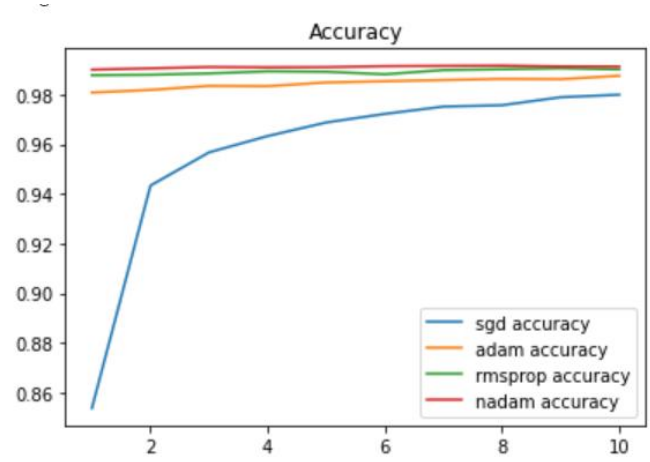
Nadam has a stronger constraint on the learning rate, and at the same time, it also has a more direct impact on the update of gradient. In general, where we want to use a volume-driven RMSprop or Adam, we can use Nadam for better results.

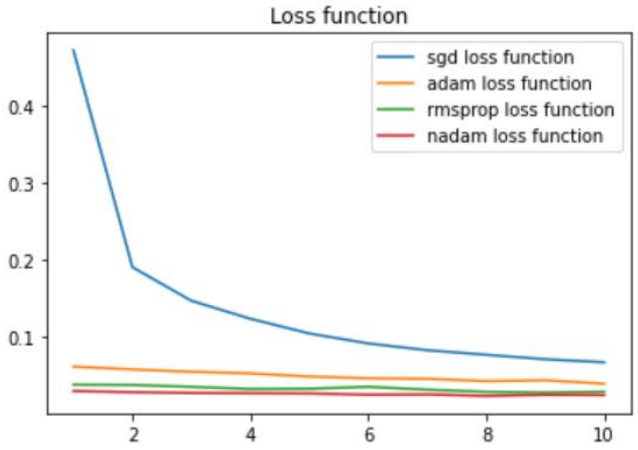
4.Experimental result

In the process of the experiment, the required dataset is downloaded from the keras.dataset module and import first, then the data is preprocessed, and then the network model is constructed and compiled. Then I start training and testing, and finally visualize the results.

4.1 MNIST dataset

For the MNIST dataset, I choose to use SGD, Adam, RMSprop and Nadam optimizers, and train 60,000 pictures and test 10,000 pictures. The results of the visualization are as follows:





In these two graphs, the lines of different colors represent different optimizers. We can see that the accuracy of the Nadam optimizer is the highest and more stable, and the accuracy of the SGD optimizer is gradually increased, but it is not as good as the other three. And in the end, their accuracy is all greater than 96%.

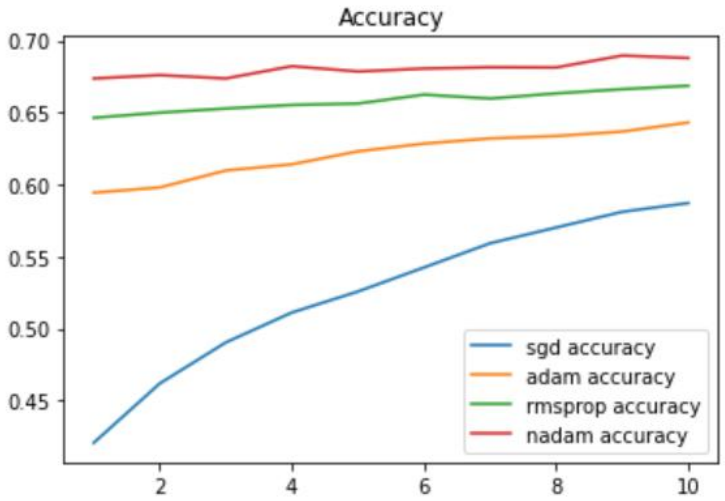
In addition, similarly, the loss function of SGD optimizer declines gradually and eventually becomes stable. The Adam and RMSprop optimizers' are always stable around 0.04, while the Nadam optimizr is the one with the lowest loss function value, around 0.03.

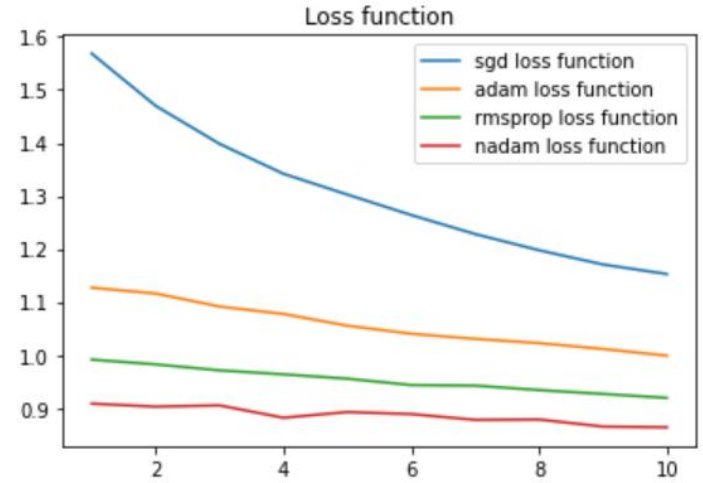
Finally, the results of the test are:

	Loss function	Accuracy	Running time
SGD	0.03	0.9932	2s
Adam	0.03	0.9932	2s
RMSprop	0.03	0.9932	2s
Nadam	0.03	0.9932	2s

4.2 CIFAR10 dataset

For the CIFAR10 dataset, I still choose to use SGD, Adam, RMSprop and Nadam optimizers, and train 50,000 pictures and test 10,000 pictures. The results of the visualization are as follows:





In these two graphs, we can see that, similar to the results of the previous task, the SGD optimizer has a stable increase in accuracy, and the Nadam optimizer has the highest accuracy, but their accuracy is not as high as that of the previous task, with the highest accuracy less than 70% and the lowest just over 55%.

Moreover, when processing the CIFAR10 dataset , among the four optimizers, the SGD optimizer has the highest loss function, which reaches about 1.2, and the nadam has the lowest loss function, which is still about 0.9. The overall situation is not as good as the performance of the previous task.

Finally, the results of the test are:

	Loss function	Accuracy	Running time
SGD	0.8410	0.7130	3s
Adam	0.8410	0.7130	3s
RMSprop	0.8410	0.7130	4s
Nadam	0.8410	0.7130	3s

Conclusion

There are also many problems in the process of doing the project. It is worth mentioning that one of the problems is:

```
ValueError: Input 0 of layer "sequential" is incompatible with the layer: expected shape=(None, 28, 28, 1), found shape=(None, 784)
```

This represents that the shape of my data input is not consistent with what it expected, and later I removed the reshaping statement, but then it had another error. I tried to remove the model constructor from the function, and it worked.

In conclusion, for MNIST dataset and CIFAR10 dataset, the loss function of SGD optimizer decreases the fastest, but the accuracy is not high. RMSprop optimizer has the characteristics of adaptive learning and strong practicability. Adam optimizer performs well in the tasks, while Nadam optimizer has higher accuracy and lower loss function.

References

[1] Li Ming, Lai Guohong, Chang Yanming, Feng Zhiqiang. Performance analysis of different optimizers in deep learning algorithm. Information Technology and Informatization , 2022,(03),206-209.