MLP Coursework 1: Learning Algorithms and Regularization

s1502810

Abstract

Optimization has always been an important topic in the field of neural networks. It is precisely because of continuous optimization that neural networks can be applied to various fields in reality. Optimization is also the main content of this report. By comparing various gradient descent algorithms, different optimization methods are found to optimize them.

1. Introduction

Learning rate is a very important hyperparameter of neural network. For its debugging, it can affect the fitting speed and fitting degree of a neural network. A good learning rate can greatly improve the accuracy of neural networks. Similarly, the gradient descent algorithm is also very important for the optimization of the neural network. The quality of the gradient descent algorithm also affects the fitting speed and accuracy of the neural network. So this report will focus on these two aspects. In the first part, we will use SGD to train the neural network. In the second part, we will compare the traditional SGD algorithm with the later improved RMSProp algorithm and Adam algorithm. Part III We will use the scheduler to optimize the choice of learning rate. In the last part we will focus on the ge of the Adam algorithm.The experimental method is mainly to train the best neural network by adjusting the learning rate, and then verify by the test data. This data set is taken from Balanced EMINST, a total of three: training data (100000), valid data (15800), test data (15800).

2. Baseline systems

In this section, in order to get the best performance of stochastic gradient descent (SGD) in 2-5 hidden layers as the baseline, I debugged the learning rate hyperparameter. Firstly I will compare these systems using the valid data set and find the best learning rate.

Then I use the test data to compare the accuracy of these best systems.

The accuracy rate is the highest and 3 hidden layers and 4 hidden layers, the accuracy rate is about 0.83, and the accuracy of 2 hidden layers and 5 hidden layers is only about 0.82. Compared with the valid data, these systems have a 1

In the following experiments, I will use this part of the best 3 hidden SGD as the baseline to compare the performance

Table 1. accuracy(valid) for different layers and different learning rate

learning_rate	2 Hidden	3 Hidden	4 Hidden	5 Hidden
0.1	8.15E-01	8.15E-01	8.17E-01	8.17E-01
0.05	8.28E-01	8.28E-01	8.27E-01	8.29E-01
0.01	8.35E-01	8.35E-01	8.39E-01	8.32E-01
0.0075	8.30E-01	8.42E-01	8.42E-01	8.32E-01
0.005	8.22E-01	8.22E-01	8.28E-01	8.33E-01

Table 2. the best learning rate for 2-5 Hidden Layers using test set

2 Hidden layers	1.00E-02	8.23E-01
3 Hidden layers	7.50E-03	8.34E-01
4 Hidden layers	7.50E-03	8.31E-01
5 Hidden layers	5.00E-03	8.22E-01

of subsequent algorithms.

3. Learning algorithms – RMSProp and Adam

In this section, we will compare the differences between SGD, RMSProp and Adam algorithms. We first use the RMSProp and Adam algorithms to train the neural networks of 3 hidden layers . By adjusting the respective parameters (learning rate), we use the training data and the verification data to train the best system. We will then test the two systems with test data and compare them to SGD.

3.1. Algorithm

SGD algorithm equation:

$$\begin{aligned} g_t &\leftarrow \nabla J_i\left(\theta_{t-1}\right) \\ \theta_t &\leftarrow \ \theta_{t-1} \ - \ \eta g_t \end{aligned} \tag{1}$$

SGD is a variant of the gradient descent algorithm because the original gradient descent algorithm is too inefficient. SGD extracts small batches (independently distributed samples) and calculates their average gradient values, so the speed is faster and the efficiency is higher. However,Because SGD is updated frequently, the cost function will be severely oscillated

RMSProp algorithm equation:

$$\begin{aligned} g_t &\leftarrow \nabla J_i\left(\theta_{t-1}\right) \\ G_t &\leftarrow \gamma G_t + (1-\gamma)\, g_t \odot g_t \\ \theta_t &\leftarrow \theta_{t-1} - \tfrac{\eta}{\sqrt{G_t + \epsilon}} \odot g_t \end{aligned} \tag{2}$$

According to the formula2, we can know that in the direction of the parameter space is more gradual, more progress will be made (because it is gentle, so the sum of the squares of the historical gradient is smaller, the magnitude of the corresponding learning decline is smaller), and the steep direction can be made gentle. , thus speeding up the training.

Adam algorithm equation:

$$\begin{aligned} & g_t \leftarrow \nabla J_i\left(\theta_{t-1}\right) \\ & m_t \leftarrow \ \beta_1 m_{t-1} \ + \ (1-\beta_1)\, g_t \\ & v_t \leftarrow \ \beta_2 v_{t-1} \ + \ (1-\beta_2)\, g_t^2 \\ & \widehat{m}_t \leftarrow \tfrac{m_t}{1-\beta_1^t} \\ & \widehat{v}_t \leftarrow \tfrac{v_t}{1-\beta_2^t} \\ & x_t \leftarrow \ x_{t-1} \ - \eta \left(\tfrac{\alpha \widehat{m}_t}{\sqrt{\widehat{v}_t + \epsilon}} \right) \end{aligned} \tag{3}$$

The Adam algorithm also takes advantage of the AdaGrad and RMSProp algorithms. Adam not only calculates the adaptive parameter learning rate based on the first-order moment mean as the RMSProp algorithm, but also makes full use of the second-order moment mean of the gradient (ie, the unbiased variance). Specifically, the algorithm calculates the exponential moving average of the gradient, and the hyperparameters beta1 and beta2 control the decay rate of these moving averages.

3.2. Experiment

Training:I used five different learning rates to train the neural network, then validated the data using valid data, and found the learning rate that made the accuracy of valid data the highest.

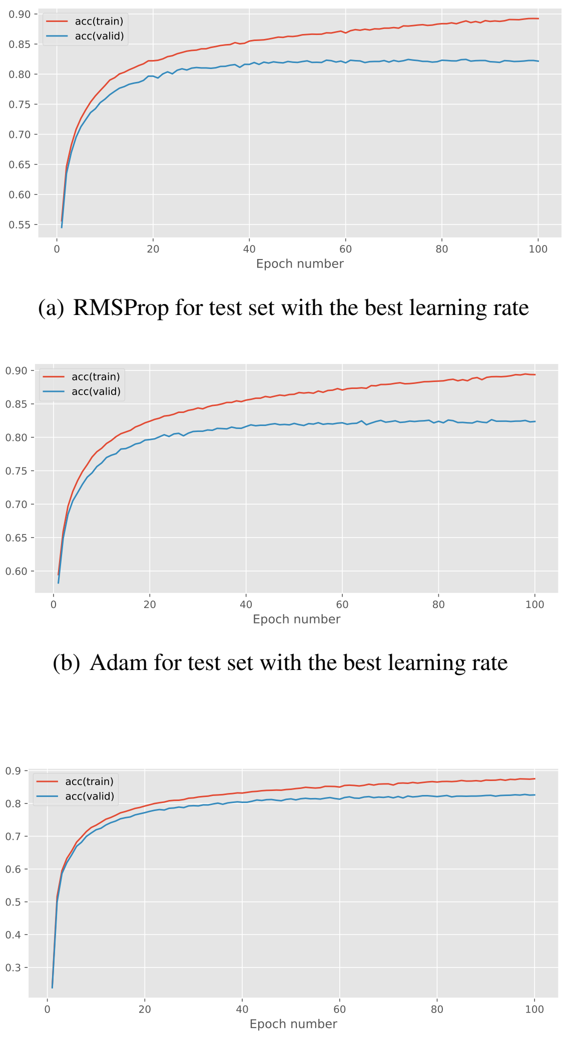
accuray(valid) of 3 Hidden layer		
learning_rate	RMSProp	Adam
1.00E-03	8.03E-01	8.15E-01
5.00E-04	8.06E-01	8.17E-01
1.00E-04	8.35E-01	8.33E-01
5.00E-05	8.34E-01	8.32E-01
1.00E-05	7.73E-01	7.27E-01

Table 3. the best learning rate for each system

algorithm	best learning rate
RMSProp	0.0001
Adam	0.0001
SGD	0.0075

This table3.2 shows the training results, and the second table3.2 shows the best learning rate for each system.I found the best learning rate by comparing the accuracy of valid set.

Test:Use test data to test the best trained system above and compare the effects of the three systems,and with the same initial weights. Because I want to know how well they fit, but also want to know their fitting speed.



(a) RMSProp for test set with the best learning rate

(b) Adam for test set with the best learning rate

(c) SGD for test set with the best learning rate

Table 4. accuracy for test set

algorithm	accuracy(test)
RMSProp	8.22E-01
Adam	8.24E-01
SGD	8.26E-01

According to the table4, SGD has a slight advantage in accuracy compared to the other two, but from the figure3.2and the figure3.2 RMSProp and Adam are faster than SGD in terms of fitting speed, and Adam is the fastest.

4. Cosine annealing learning rate scheduler

In this section, I will first introduce the cosine annealing scheduler and explain how it works. Then I will adjust the super parameters, train the system I think is the best, and then compare the performance of SGD and Adam in "no scheduler", "scheduler with no restarts", "special restarts" through test data.

4.1. Algorithm

cosine annealing scheduler update learning equation:

$$\eta_t = \eta^{(i)}_{min} + 0.5(\eta^{(i)}_{max} - \eta^{(i)}_{min})(i + \cos(\frac{\pi \mathrm{T_{cur}}}{T_i})) \quad (4)$$

In deep learning optimization, an important hyperparameter is the learning rate η. A good learning rate allows the neural network to achieve better fitting and faster fitting.

The cosine annealing scheduler is an algorithm that optimizes the learning rate. It can help the neural network adjust the learning rate. We can know by equation**??** that we give a range of learning rate, the number of iterations of the cycle, and then the scheduler initializes the learning rate to a certain value and then gradually reduces it each time it restarts. . The key is to adjust the learning rate through a positive cosine annealing scheme, which will quickly reduce the learning rate.

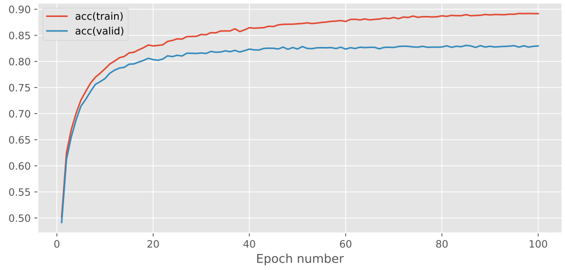
4.2. Experiment

Training:Because in the section, we just compare SGD and Adam with using cosine annealing with no restarts, and with cosine annealing with restarts specified by an initial T_i = 25 and T_{mult} = 3. "no restart" means the T_i is equal to the epoch number(100), and T_{mult} could be any number(I assign 1).So according to the equation4,I only need to adjust the values of η_{max} and η_{min} to get the best performance of the neural network.I introduced a variable range to control the range of learning rates. The specific formula is as follows:

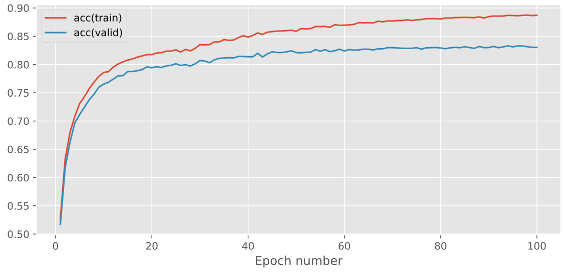
$$\begin{aligned} \eta_{max} &= best\_learning\_rate * range \\ \eta_{min} &= best\_learning\_rate \ / \ range \end{aligned} \quad (5)$$

The best learning rates I have trained for SGD and Adam are 7.5e-3**??** and 1e-4**??** respectively.

for SGD:

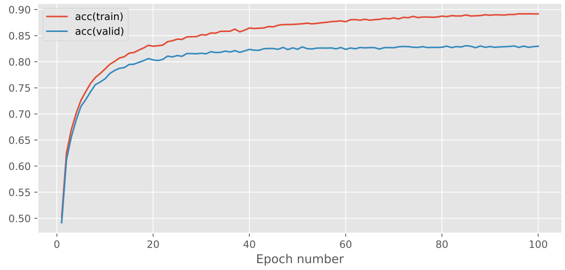


(d) SGD with scheduler no restarts

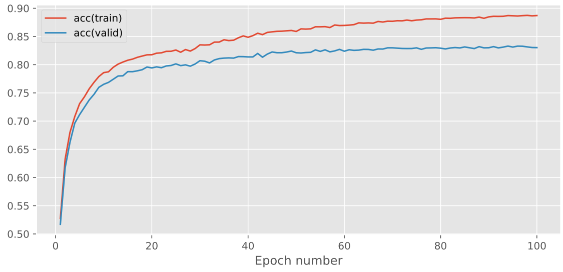


(e) SGD with scheduler special restarts

for SGD:



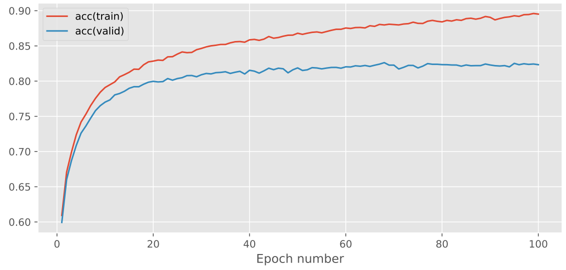
(f) Adam with scheduler no restarts



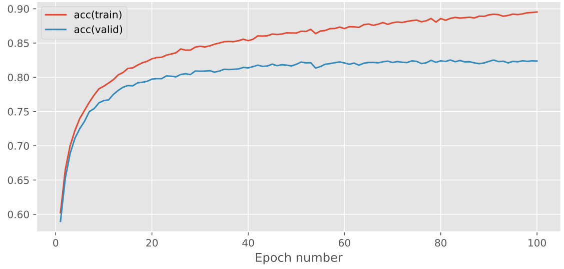
(g) Adam with scheduler special restarts

From the figure, the accuracy of SGD With scheduler is not much improved, basically the same as before, but the fitting speed of SGD has improved a lot.

for Adam:



(h) RMSProp for test set with the best learning rate



(i) Adam for test set with the best learning rate

Table 5. SGD for scheduler

range	no restart (acc(valid))	special restart(acc(valid))
1.05	8.36E-01	8.36E-01
1.1	8.38E-01	8.36E-01
1.15	8.39E-01	8.38E-01
1.2	8.40E-01	8.36E-01
2	8.40E-01	8.42E-01
3	8.38E-01	8.41E-01
5	8.37E-01	8.43E-01
10	8.32E-01	8.38E-01

Table 6. Adam for scheduler

range	no restart (acc(valid))	special restart(acc(valid))
1.05	8.39E-01	8.40E-01
1.1	8.36E-01	8.35E-01
1.15	8.36E-01	8.34E-01
1.2	8.37E-01	8.36E-01
2	8.36E-01	8.37E-01
3	8.34E-01	8.39E-01
5	8.28E-01	8.32E-01
10	8.23E-01	8.23E-01

According to table5 and table6, we can find the best range of learning for SGD4.2 and Adam4.2 are 2 and 1.05 respectively.

Test:using test set to test the best SGD with no restarts and special restarts, comparing the no scheduler, the same for Adam.

5. Regularization and weight decay with Adam

5.1. the problem using L2 regularization with Adam

This is the equation of L2:

$$f^{reg}(x_t) = f_t(x_t) + \frac{w}{2}\|x\|_2^2 \qquad (6)$$

L2 regularization means add L2 norm to loss function.According to the equation of Adam3, This means that the larger the value of g_t, the larger the v_t will be. When the weight is updated, an error will occur. The greater the weight, the smaller the penalty will be.

5.2. solution

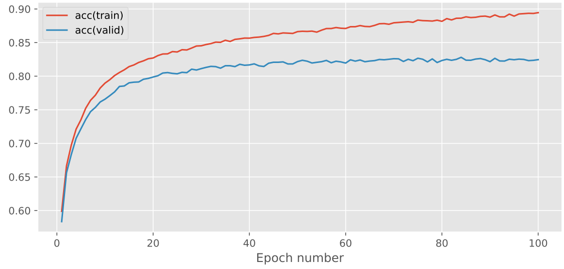
Weight Decay is the solution.This algorithm adds an item when Adam updates the weights.

$$x_t \leftarrow x_{t-1} - \eta \left(\frac{\alpha \widehat{m}_t}{\sqrt{\hat{v}_t}+\epsilon}\right) + w x_{t-1} \qquad (7)$$

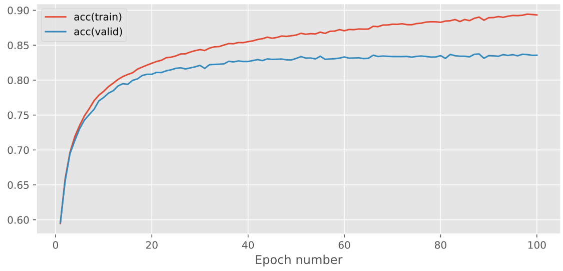
Weight decay is updated with the same coefficients for all weights. The larger the weight, the greater the penalty.

5.3. Experiment

L2 VS Weight decay:



(j) L2



(k) Weight decay

the accuracy of L2 is 0.825 and the data of Weight decay is 0.836,this also shows that weight decay is more suitable for Adam than L2.

constant learning rate vs. cosine annealing schedule:

constant learning rate5.3

no restarts in the scheduler vs. use of a warm restart:

6. Conclusions

In conclusion, Although the fitting speed of SGD is not fast, its accuracy rate is not lower than RMSPorp and Adam. In the third part, after using scheduler with SGD, the fitting speed of SGD is obviously faster than before. The Adam algorithm's fitting speed is very fast, but there is no obvious difference between the accuracy and the SGD. Later, the weight decay optimization is used to prevent the Adam from overfitting. The warm restarts can further accelerate the convergence speed of Adam, and can also improve Adam's convergence. Accuracy The scheduler can help us get rid of the pain of debugging learning rate, although the accuracy of neural network learning by scheduler may not be the highest, but it can save a lot of time and indirectly improve efficiency. The best systems is AdamW with restarts.

References

Langley, P. Crafting papers on machine learning. In Langley, Pat (ed.), *Proceedings of the 17th International Conference on Machine Learning (ICML 2000)*, pp. 1207–1216, Stanford, CA, 2000. Morgan Kaufmann.

Loshchilov, Ilya and Hutter, Frank. Fixing weight decay regularization in Adam. *arXiv preprint arXiv:1711.05101*, 2017. URL https://arxiv.org/abs/1711.05101.