Published in Towards Data Science

nachiket tanksale Follow

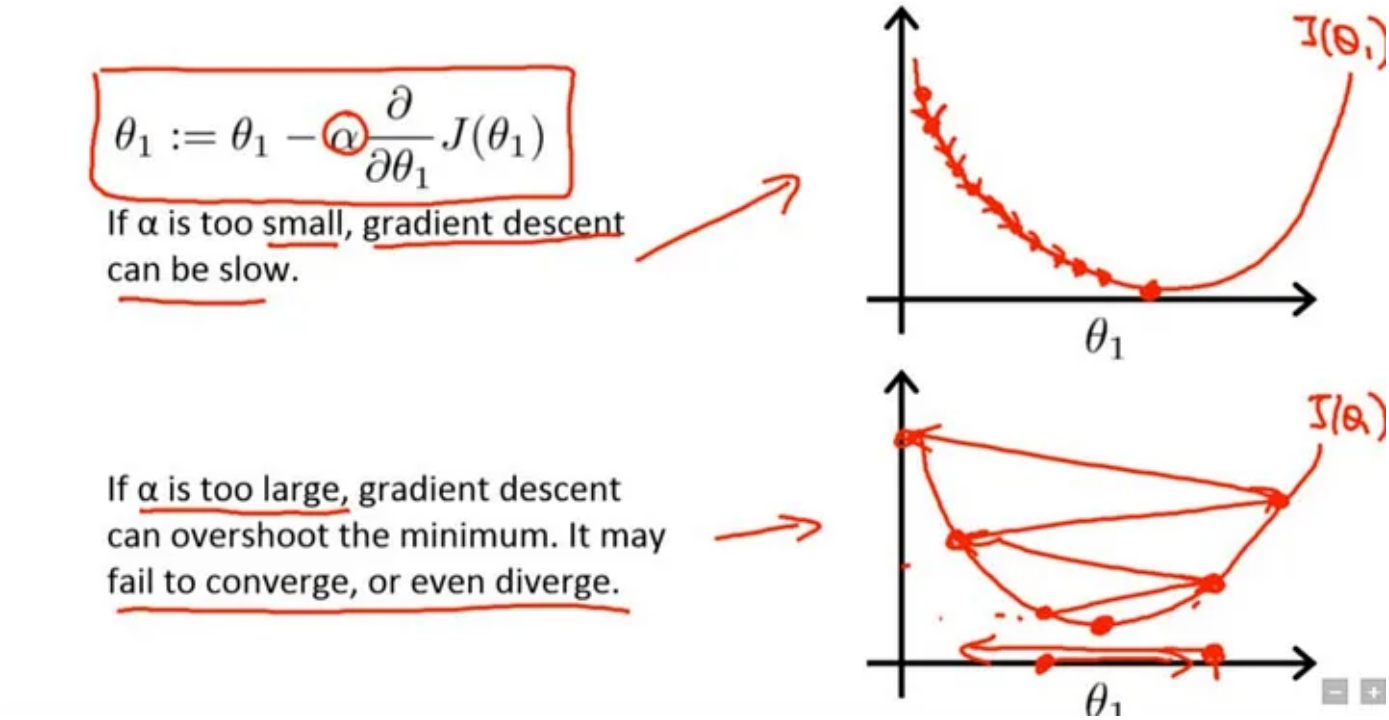
Jun 24, 2018 · 8 min read · Listen

Save

Finding Good Learning Rate and The One Cycle Policy.

Introduction

Learning rate might be the most important hyper parameter in deep learning, as learning rate decides how much gradient to be back propagated. This in turn decides by how much we move towards minima. The small learning rate makes model converge slowly, while the large learning rate makes model diverge. So, the learning rate needs to be just correct.



Gradient descent with small(top) and large (bottom) learning rates. Source: Andrew Ng's Machine Learning course

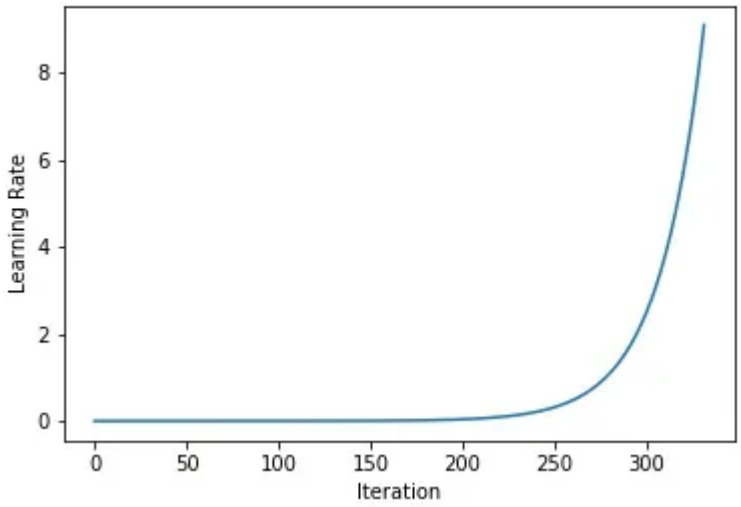
Still finding and setting the correct learning rate is more of trial-and-error. The naive way is to try different learning rates and choose the one which gives smallest loss value without sacrificing the learning speed.(Validation loss also matters for underfitting/overfitting).

This article gives short summary of 2 papers which describes method to set different hyper-parameters. This article assumes that the reader knows back-propagation , gradient descent and hyper-parameters.

Is there a better way?

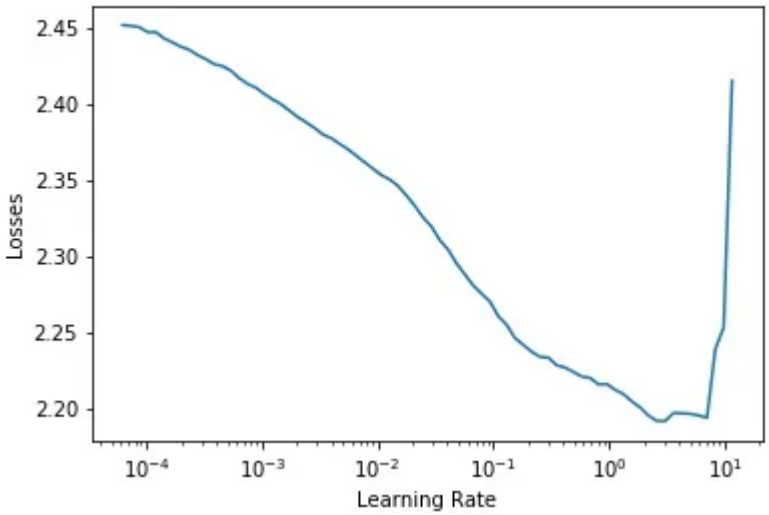
While going through **Practical Deep Learning For Coders, Part 1** mooc, there was mention of paper **Cyclical Learning Rates for Training Neural Networks** by Leslie N. Smith.

The paper mentions the range test run for few epochs to find out good learning rate, where we train from some low learning rate and increase the learning rate after each mini-batch till the loss value starts to explode.



Learning Rate Increase After Every Mini-Batch

The idea is to start with small learning rate (like 1e-4, 1e-3) and increase the learning rate after each mini-batch till loss starts exploding. Once loss starts exploding stop the range test run. Plot the learning rate vs loss plot. Choose the learning rate one order lower than the learning rate where loss is minimum(if loss is low at 0.1, good value to start is 0.01). This is the value where loss is still decreasing. Paper suggests this to be good learning rate value for model.



Test run on CIFAR-10 with batch size 512, resnet 56 , momentum=0.9 and weight decay=1e-4. The learning rate ~10^0 i.e. somewhere around 1 can be used.

So, this is how we'll update the learning rate after each mini-batch:

> *n = number of iterations*
> *max_lr = maximum learning rate to be used. Usually we use higher values like 10, 100. Note that we may not reach this lr value during range test.*
> *init_lr = lower learning rate. We'll start range test from this value. We use very small value like 1e-3, 1e-4.*
> *Let, q be the factor by which we increase learning rate after every mini batch.*
> *Below image shows the equation to find the learning rate after i-th mini-batch.*

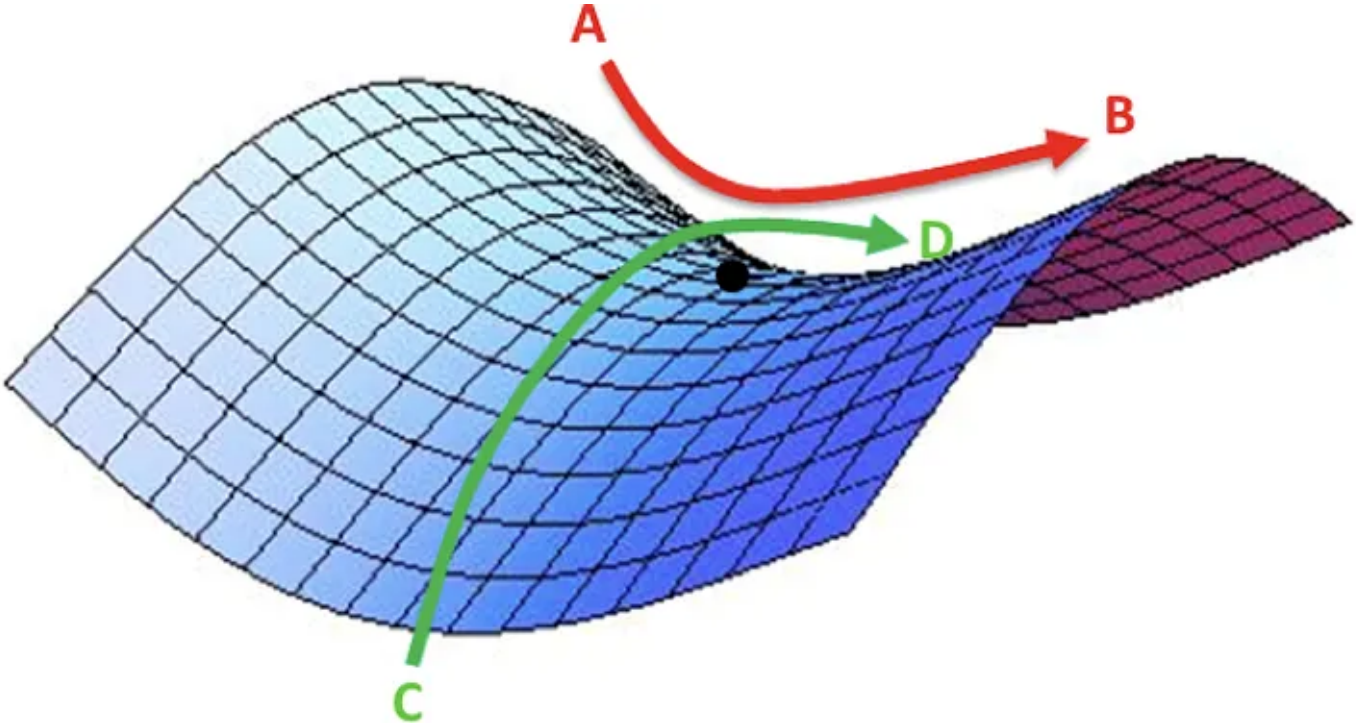
$$max\_lr = init\_lr * q^n$$
$$q = (\frac{max\_lr}{init\_lr})^{\frac{1}{n}}$$
$$lr_i = init\_lr * q^i$$
$$lr_i = init\_lr * (\frac{max\_lr}{init\_lr})^{\frac{i}{n}}$$

once we find optimal learning rate we use it for training the model. Range test is very useful tool as it provides a way to find good learning rate with small number of epoch runs.

Cyclic Learning Rates:

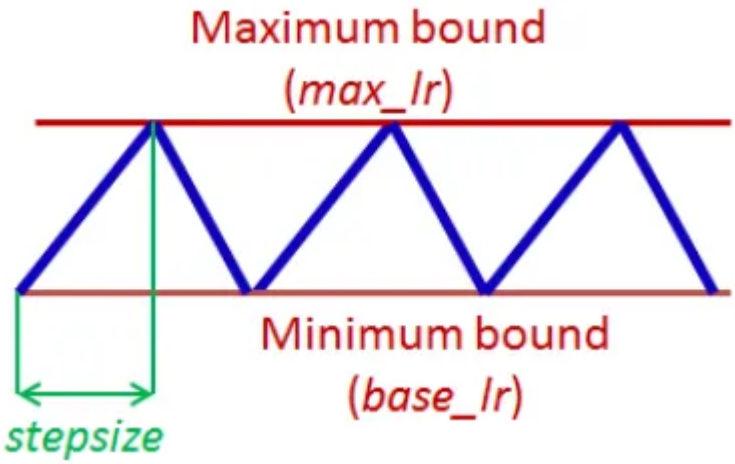
The paper further suggests to cycle the learning rate between lower bound and upper bound during complete run. Conventionally , the learning rate is decreased as the learning starts converging with time. So what is the motivation behind cyclic learning rate?

Intuitively, it is helpful to oscillate the learning rate towards higher learning rate. As the higher learning rate may help to get out of saddle points. If saddle point is elaborate plateau, the lower learning rates might not be able get gradient out of saddle point.



A saddle point in the error surface (Img Credit: safaribooksonline)

Cycle is number of iterations where we go from lower bound learning rate to higher bound and back to lower bound. Cycle may not have boundary on epoch, but in practice it usually does. Stepsize is half of cycle. So Stepsize is number of iterations where we want learning rate to go from one bound to the other.



Cyclic Learning Rate(Image: https://arxiv.org/pdf/1506.01186.pdf)

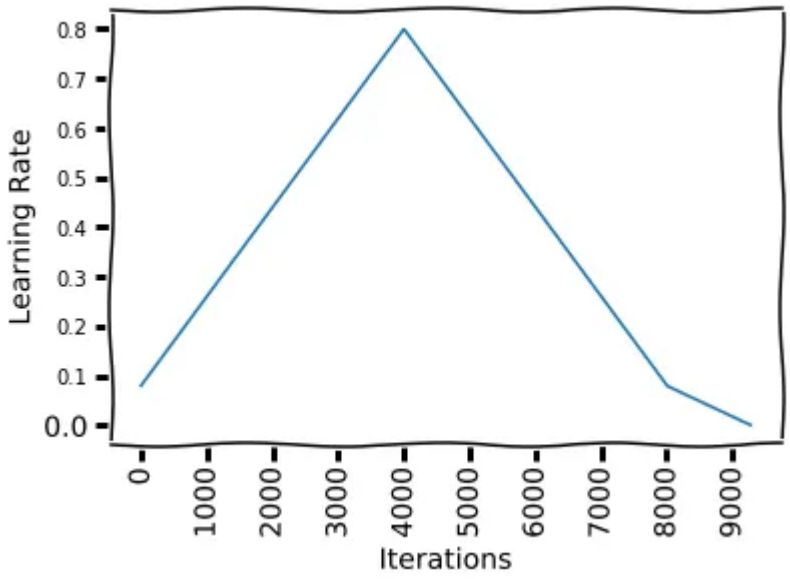
The One Cycle Policy

In the paper "A disciplined approach to neural network hyper-parameters: Part 1 — learning rate, batch size, momentum, and weight decay" , Leslie Smith describes

approach to set hyper-parameters (namely learning rate, momentum and weight decay) and batch size. In particular, he suggests 1 Cycle policy to apply learning rates.

Author recommends to do one cycle of learning rate of 2 steps of equal length. We choose maximum learning rate using range test. We use lower learning rate as 1/5th or 1/10th of maximum learning rate. We go from lower learning rate to higher learning rate in step 1 and back to lower learning rate in step 2. We pick this cycle length slightly lesser than total number of epochs to be trained. And in last remaining iterations, we annihilate learning rate way below lower learning rate value(1/10 th or 1/100 th).

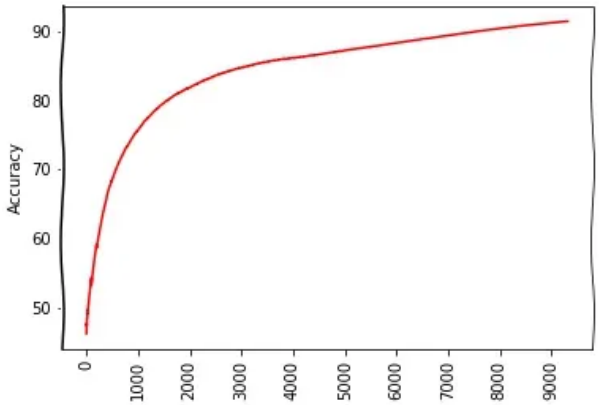
The motivation behind this is that, during the middle of learning when learning rate is higher, the learning rate works as regularisation method and keep network from overfitting. This helps the network to avoid steep areas of loss and land better flatter minima.

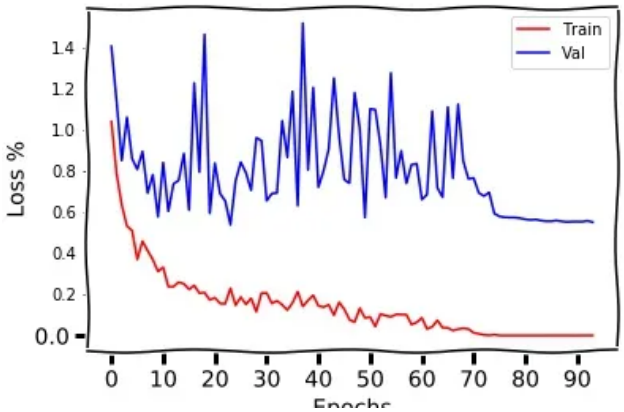


CIFAR -10: One Cycle for learning rate = 0.08–0.8 , batch size 512, weight decay = 1e-4 , resnet-56

As in figure , We start at learning rate 0.08 and make step of 41 epochs to reach learning rate of 0.8, then make another step of 41 epochs where we go back to learning rate 0.08. Then we make another 13 epochs to reach 1/10th of lower learning rate bound(0.08).

With CLR 0.08–0.8 , batch size 512, momentum 0.9 and Resnet-56 , we got ~91.30% accuracy in 95 epochs on CIFAR-10.





Cyclic Momentum

Momentum and learning rate are closely related. It can be seen in the weight update equation for SGD that the momentum has similar impact as learning rate on weight updates.

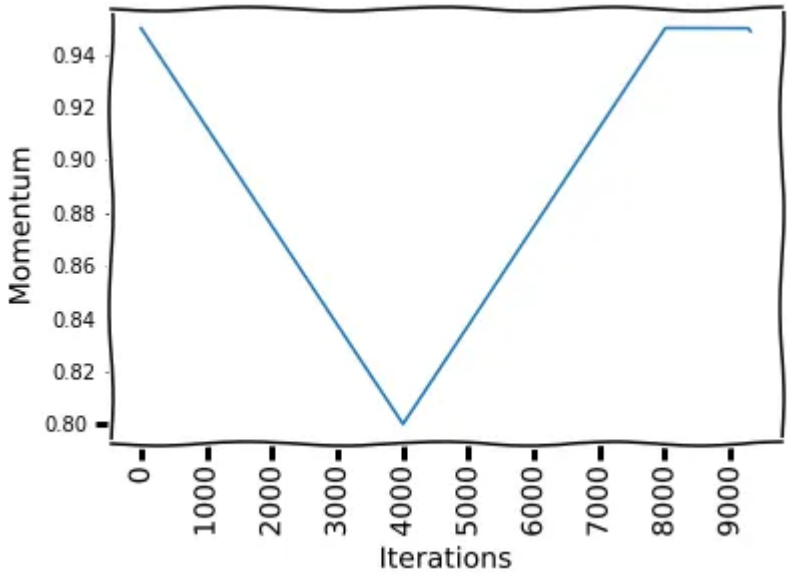
In SGD, weights are updated as:

$$\Theta_{iter+1} = \Theta_{iter} - \epsilon * \partial L * (F(x, \Theta), \Theta)$$

With momentum term, weight update in SGD becomes:

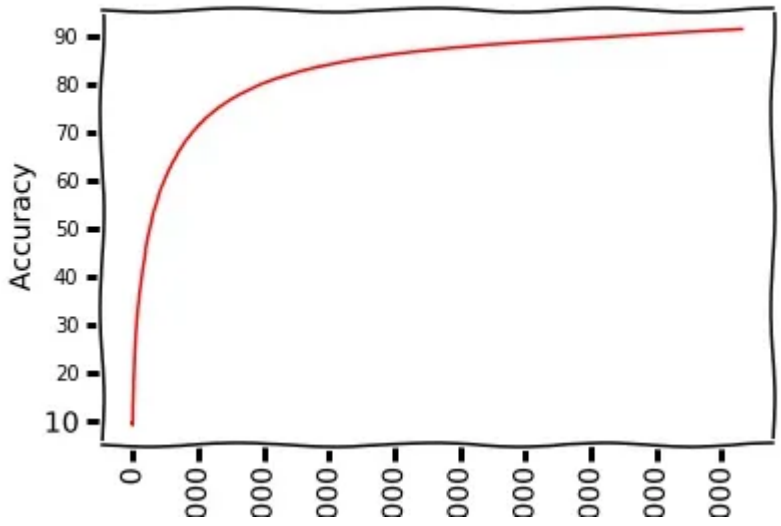
$$v_{iter+1} = \alpha * v_{iter} - \epsilon * \partial L * (F(x, \Theta), \Theta)$$
$$\Theta_{iter+1} = \Theta_{iter} + v$$

Author found in their experiments that reducing the momentum when learning rate is increasing gives better results. This supports the intuition that in that part of the training, we want the SGD to quickly go in new directions to find a better minima, so the new gradients need to be given more weight.

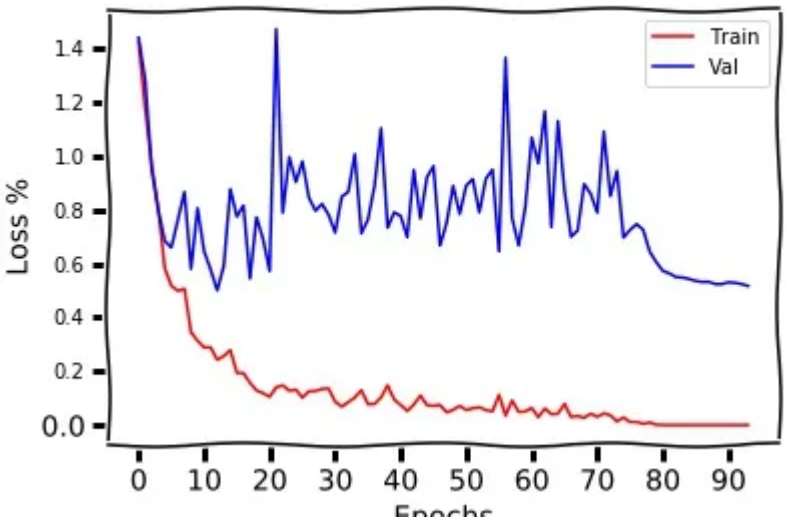


In practice we choose 2 values for momentum. As in One Cycle , we do 2 step cycle of momentum, where in step 1 we reduce momentum from higher to lower bound and in step 2 we increase momentum from lower to higher bound. According to paper, this cyclic momentum gives same final results, but this saves time and efforts of running multiple full cycles with different momentum values.

With One Cycle Policy and cyclic momentum , I could replicate the results mentioned in paper. Where the model achieved 91.54% accuracy in 9310 iterations, while using one cycle with learning rates 0.08–0.8 and momentum 0.95–0.80 with resnet-56 and batch size of 512, while without CLR it requires around 64k iterations to achieve this accuracy.(Paper achieved 92.0 ± 0.2 accuracy) .

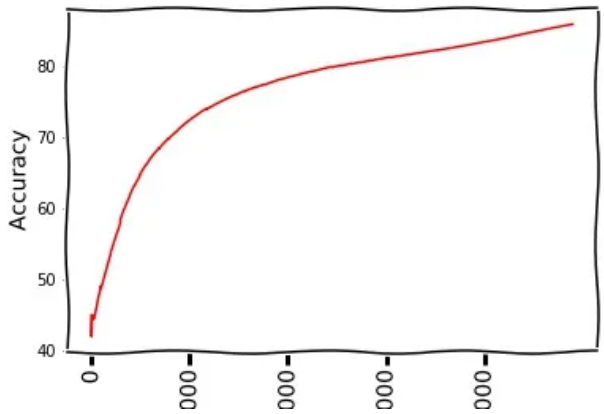


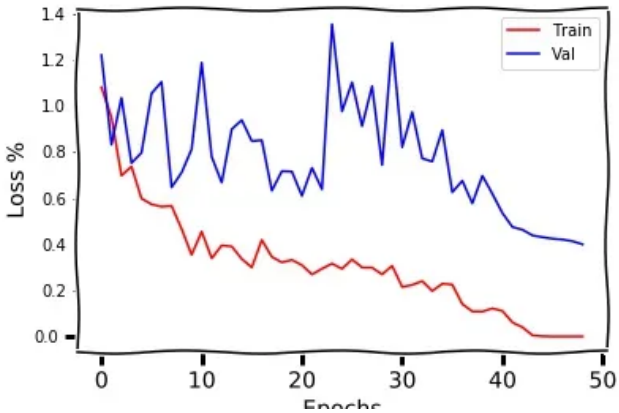
Accuracy vs Iterations



Training and Validation loss vs epochs

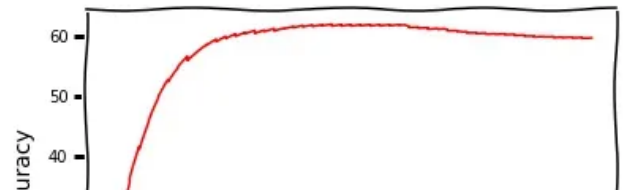
This allows us to train models at higher learning rates. We could get 85.97% training accuracy at learning rate 0.3–3 by training resnet-56 for just 50 epochs.

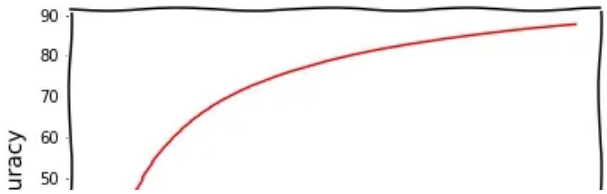




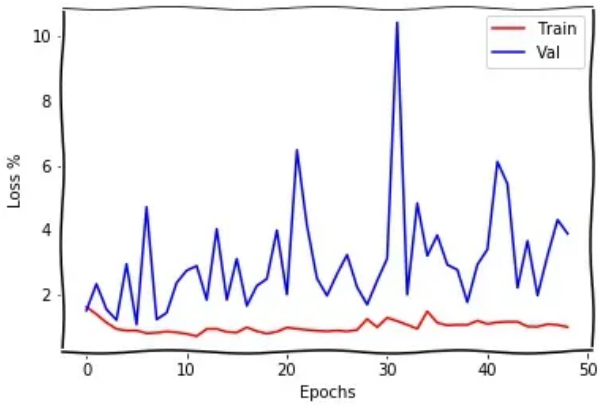
Weight Decay Value matters too.

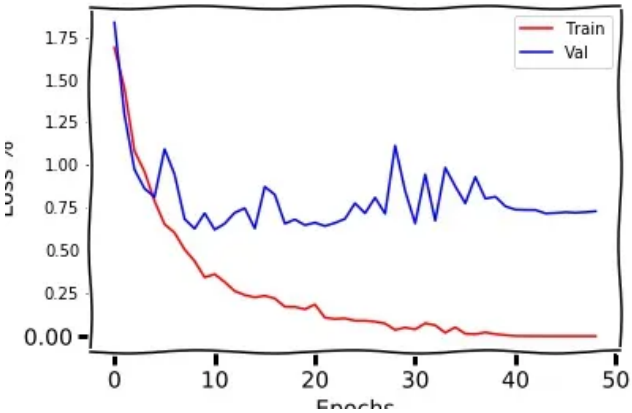
Weight decay is also an important hyper parameter. Weight decay also works as regularisation factor. But its quite different from learning rate or momentum, as author found out the optimum value should remain constant throughout the training.





Resnet 56 Accuracy after 50 epoch on CIFAR-10 with weight decay = 1e-3(left) vs 1e-5(right)





Resnet 56 training and validation losses after 50 epoch on CIFAR-10 with weight decay = 1e-3(left) vs 1e-5(right)

As can be seen from above figures 1e-3 is pretty bad weight decay value, as the accuracy barely reached 60% after 50 epochs,whereas with accuracy of 85.97% and 87.78% with weight decay 1e-4 and 1e-5 respectively.(CLR range 0.3–3 and momentum range 0.95–0.8 , batch size 512)

The author suggests , its reasonable to make combined run with CLR and Cyclic momentum with different values of weight decay to determine learning rate, momentum range and weigh decay simultaneously. The paper suggests to use values like 1e-3, 1e-4, 1e-5 and 0 to start with, if there is no notion of what is correct weight decay value. On the other hand, if we know , say 1e-4 is correct value, paper suggests to try 3 values bisecting the exponent(3e-4, 1e-4 and 3e-5).

Total Batch Size

The paper suggests the highest batch size value that can be fit into memory to be used as batch size.

Conclusion

The range test method provides a [illegible] out a good learning rate value with few iterations of run. The o[illegible]omentum seems to allow model to be trained on higher learning rates and converge faster. The one cycle policy provides some form of regularisation. So, other form of regularisation needs to be adjusted accordingly.

Few of the experiments mentioned above can be found out in notebook here.

References:

1. Cyclical Learning Rates for Training Neural Networks
2. A disciplined approach to neural network hyper-parameters: Part 1 — learning rate, batch size, momentum, and weight decay
3. Andrew Ng's Coursera Course
4. Fastai library
5. Practical Deep Learning For Coders, Part 1
6. https://sgugger.github.io/how-do-you-find-a-good-learning-rate.html
7. http://teleported.in/posts/cyclic-learning-rate/
8. https://sgugger.github.io/the-1cycle-policy.html

If you liked this article, please be sure to give me a clap and follow me to get updates on my future articles.

Also, feel free to connect me on LinkedIn or follow me on Twitter.

If you like my work do consider sponsoring me, it'll help me put out more such work.

Deep Learning Pytorch Learning Rate Cyclic Learning Rate One Cycle