Comparison of Cyclical and Fixed Learning Rates with different Optimizers using CNN

Tarushi Jat
Department of Information Technology
National Institute of Technology, Surathkal
Karnataka, India
tarushijat.202it029@nitk.edu.in

Divyanshi Bhojak
Department of Information Technology
National Institute of Technology, Surathkal
Karnataka, India
divyanshib.202it007@nitk.edu.in

Pragnesh Thaker
Department of Information Technology
National Institute of Technology, Surathkal
Karnataka, India
pragnesh.187it001@nitk.edu.in

Biju R. Mohan
Department of Information Technology
National Institute of Technology, Surathkal
Karnataka, India
biju@nitk.edu.in

***Abstract*—The learning rates and the optimizers are the two most important concepts in training neural networks. The traditional techniques of the learning rate have the drawback of the precariousness of accuracy. Aiming at these issues, we worked on a new technique that instead of using a fixed value to tweak learning rate during training, we used a cyclical method that selects a span of values for the learning rate. Cyclical Learning Rate (CLR) that proposed the strategy to increase and decrease the learning rate back and forth and eliminate the need to perform multiple investigations to achieve higher accuracy in fewer iterations. On the other hand, optimization algorithms like Adam accomplish much higher optimization performance in comparison to stochastic gradient descent. But present-day studies have demonstrated that the Adam fails to give a generalized neural network model as opposed to SGD. Consequently in this paper, we worked on Adam with weight decay(AdamW) which aims to give as good a generalization as SGD in fewer iterations. In this paper, we have implemented cyclical learning rates with SGD, Adam and AdamW optimizers and experimented with CIFAR-10, CIFAR-100, and SVHN datasets.**

***Index Terms*—Cyclical Learning Rate, Convolutional Neural Networks, Image Classificastion, Optimizers, Adam, Stochastic Gradient Descent, Weight Decay.**

I. Introduction

Neural Networks are no longer rare phrases in the Computer Science community. The main reason that makes them crucial are the solutions that they provide to a wide range of real-world problems and their variations from one another. Of course, the entire future of neural networks does not reside in attempts to simulate consciousness. Indeed, at the present moment, there is a need to improve this system. Focusing on the recent developments in neural networks, we have proposed a few techniques to enhance the performance of training neural networks.

Learning rate is a crucial hyper-parameter to tune for training neural networks that find out the size of the step in each and every iteration while minimizing the loss function. In the deep neural network, the parameters θ (weights) are updated by

$$\theta^t = \theta^{t-1} - \epsilon_t(\delta L/\delta\theta) \quad (1)$$

where L is the loss function and ϵ_t is the learning rate.

Setting an optimal learning rate is eventually a rigid trade because an immense value of learning rate will lead to undesirable divergent behavior in the loss function and if the learning rate is chosen small then the model will be trained very slowly as the updates in the network will be very minute. Cyclical learning rate will cut down the efforts explicitly in tuning learning rate as it allows the learning rate to tweak in a cyclic fashion between defined values during the complete run. The Cyclical Learning rate will keep on increasing and decreasing, forming a cycle during the entire training process between a fixed boundary range provided explicitly during the model training.

Along with learning rates, optimizers shape and mold the model into its most accurate possible form by futzing with the weights. They are of prime importance in training neural networks because they are used to tune the parameters of neural networks in order to minimize the cost function. In deep neural networks, the most practical optimization methods are based on stochastic gradient descent (SGD) algorithms, but its convergence time is larger than other optimization algorithms like Adam. Adam optimizer designed precisely for training deep neural networks and it takes advantage of computing individual learning rates for different parameters. Adam has gained huge fame in terms of speed of training the neural network and optimization. But it has been observed that in this case, like with the CIFAR-10 dataset, Adam fails to achieve the state-of-art performance and lacks in providing a better generalization model. Consequently, we worked with AdamW which aims to fill the differences between SGD and Adam in terms of the performance generalization. The objective of using Adam with weight decay is to keep the weights of the network small because in training the model with Adam, the weights of the network get heavier and lead to a degraded generalization.

In this paper, we worked on cyclical learning rate, then compare different optimizers, and worked on different datasets CIFAR-10, and SVHN. Lastly, we will compare these techniques based on performance metrics such as accuracy, recall, and precision.

II. Related Work

Cyclical learning rates are known as competitors for adaptive learning rates because calculations of adaptive learning rates involve some computational cost where cyclical learning rate sets a global learning rate and does not possess this computational cost, so can be used freely. Adaptive learning rates can also be combined with cyclical learning rates.

Leslie N. Smith [1] initially introduced the idea of a cyclical learning rate for training neural networks. They experimented with various policies of implementing cyclical learning rates like triangular policy, triangular-2 policy, and exp_range policy. And their results have shown that training a model with a cyclical learning rate gives a little better accuracy than using a fixed learning rate. However, they have implemented their work with learning rates using adaptive optimization algorithms like Adam, RMSprop, AdaGrad, AdaDelta using CIFAR-10 and CIFAR-100 datasets.

Ilya Loshchilov and Frank Hutter [2] have recognized the problem of generalization with Adam and proposed the solution in the research paper ”Decoupled Weight Decay Regularization”, where they worked towards fixing weight decay regularization in Adam. They did a simple modification by decoupling the weight decay from the optimization steps taken with respect to the loss function and called it AdamW. They have also shown that this improved version of Adam is capable of competing with SGD with momentum on image classification datasets like CIFAR-10.

Jinia Konar, Prerit Khandelwal and Rishabh Tripathi [3] did a comparison of various learning rate scheduling techniques on a convolutional neural network where they worked with constant learning rate, step decay learning rate, exponential decay learning rate, differential learning rate, cyclical learning rate and stochastic gradient descent with warm restarts [4]. After performing experimentation with different image classification datasets, they concluded that the cyclical learning rate performs better as compared with the rest.

III. Methodology

In this paper, we are considering cyclical learning rate implementation with three optimizers separately which are Stochastic Gradient Descent (SGD), Adam and Adam with Weight Decay (AdamW), and compared it with the Fixed Learning Rate (FLR) implementation based on various performance metrics.

A. Cyclical Learning Rates

In the cyclical learning rate technique, the values of the learning rate keep on varying between the two boundary values during the entire training process of the model. We fixed the two extreme boundary values for learning rate as lower and upper bound values within which learning rate values will tweak [**?**]. There are various techniques through which the learning rate will keep on updating and some of the highly used techniques are triangular-1 policy, triangular-2 policy, and exp-range policy. In our experiment, we have used the cyclical learning rate implementation with triangular-1 policy. The triangular-1 update policy can be understood from Fig. 1.

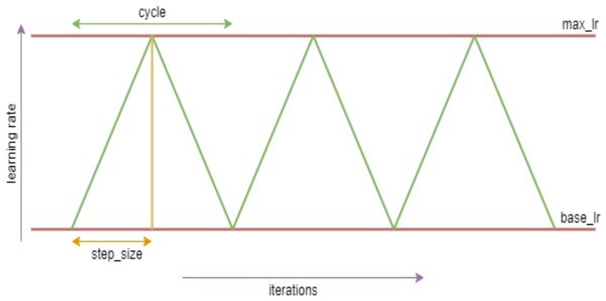

Fig. 1. Cyclical Learning Rate - Triangular Policy.

The learning rate will increase from the lower bound that is base_lr to the upper bound that is max_lr for the defined step size and then will decrease from upper bound to lower bound completing one cycle of cyclical learning rate. The calculation of learning rate in every step following triangular-1 policy is computed as follows:

$$lr = base\_lr + (max\_lr - base\_lr)max(0, 1 - x) \tag{2}$$

where x is defined as,

$$x = abs(iterations/(2stepsize)) \tag{3}$$

And cycle is calculated as,

$$cycle = floor(1 + iterations/(2stepsize)) \tag{4}$$

In above calculation, base_lr is the lower bound and max_lr is the upper bound of cyclical learning rate, step size is defined as (½ * cycle length), iterations are the number of mini-batches completed till yet.

The advantage of using cyclical learning rate over any other technique of updating learning rate is that it never gets stuck in the plateau region as the learning rate keeps on increasing for one half of the cycle length and therefore this will bring it out from such a region. Also, the narrower parts in the loss function can be explored with decreasing values of learning rate for the remaining half of the cycle.

B. Optimization Algorithms

During the training phase, we tweak and modify our parameters to optimize and make our predictions as accurate as possible. But the main concern is what is the basis on which we can judge what, when, where, and how to make a modification so that we can acquire the best possible results. And this is where optimization algorithms come to the rescue. Optimizers bind together the parameters along with the loss function to output the best predictions for our models.

In this paper, we have experimented with two popular optimizers Adam and Stochastic Gradient Descent with momentum and up to the minute optimizer the Adam with weight decay. Adam [2] after its origination in 2015 gained huge fame since then and became very prominent in terms of optimization algorithms. But a while, researchers observe that notwithstanding speed in training time in some areas it does not converge to optimal solutions. Then the other optimization algorithm we experimented with was stochastic gradient descent with momentum.

Stochastics gradient descent a single learning rate names alpha which does not change during the entire training phase and is maintained for all the Weights. And later a simple addition to this algorithm with momentum named SGD with momentum proved to work faster and better in comparison to the exemplary SGD algorithm. SGD with momentum in this momentum helps to escalate gradient vectors in accurate directions. But the problem is that it was not adaptive in the sense that it does not update the parameters when to slow down and when to speed up depending on their importance. Consequently, we lastly experiment with Adam with weight decay.

C. Adam with Weight Decay

At the closing of the year 2017, appeared to get a new tenure of survival with the discovery of the concept of weight decay. AdamW [6] introduces the concept of weight decay which is a regulatory regularization approach that appends a small penalty to the loss function. In AdamW, the foremost notion is that the hypothesis of weight decay in training neural networks attempts to keep the value of weights small, as larger weights are likely to overfit the model during training. AdamW modifies the classic implementation of weight decay regularization, by decoupling it from the updates of gradient In contrast to Adam, AdamW with this modification in implementation also solves the generalization problem observed in Adam. Through experiments in our paper, we will show some more insights into this discussion in subsequent sections.

IV. Experiments

A. Datasets

All the experiments of cyclical learning rate with three different optimizers as described in the methodology are performed on convolutional neural network architecture with the help of two datasets : CIFAR-10 and SVHN.

- CIFAR-10: It is the one of the standard dataset used for image classification problems. It is claimed that for this dataset, we get state-of-art results with stochastic gradient descent optimizer whereas Adam does not give generalized results [5]. This dataset contains a total of 60,000 images which are distributed among 10 classes (for example, airplane, frog, deer, horse, automobile, etc). For training the convolutional neural network architecture, we took 50,000 images, and for testing 10,000 images were used for the experiments.
- SVHN: It is the Street View House Number dataset which contains images of real world house numbers and it is distributed among 10 classes. For training the convolutional neural network architecture, we used 73,257 images and for testing 26,032 images were utilized.

B. Architecture for Convolutional Neural Network

We have designed a convolutional neural network architecture that consists of six 2D convolutional layers of filter size 32 for the first two convolutional layers then 64 for the next two layers and lastly 128 for the last two layers. Each convolutional layer has a filter size of 3X3. Batch Normalization was added after every convolutional layer and dropout was used after every two convolutional layers. Max pooling with a pool size of 2X2 is used after every two convolutional layers. Lastly, we have used two dense layers. In all the convolutional and dense layers, we have used relu activation function except in the last layer used softmax as the activation function. The architecture is also shown in Fig. 2.

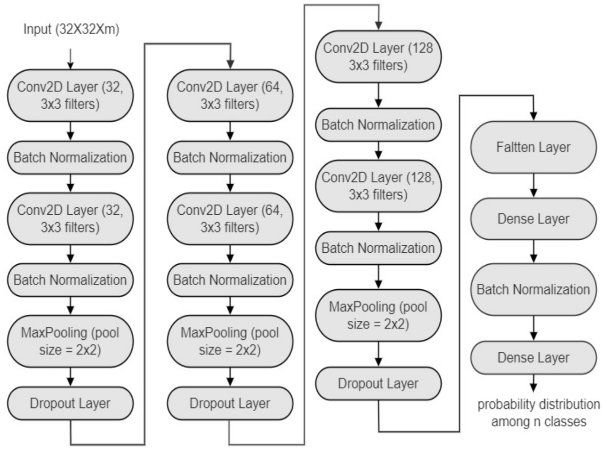
Input (32X32Xm)
Conv2D Layer (32, 3x3 filters)
Batch Normalization
Conv2D Layer (32, 3x3 filters)
Batch Normalization
MaxPooling (pool size = 2x2)
Dropout Layer
Conv2D Layer (64, 3x3 filters)
Batch Normalization
Conv2D Layer (64, 3x3 filters)
Batch Normalization
MaxPooling (pool size = 2x2)
Dropout Layer
Conv2D Layer (128 3x3 filters)
Batch Normalization
Conv2D Layer (128, 3x3 filters)
Batch Normalization
MaxPooling (pool size = 2x2)
Dropout Layer
Faltten Layer
Dense Layer
Batch Normalization
Dense Layer
probability distribution among n classes

Fig. 2. Convolutional Neural Network Architecture.

C. Implementation Details

We investigate the comparative performance of the three optimization algorithms mentioned in the above sections across two major categories of image classification datasets are CIFAR-10 of image size 32 x 32 and SVHN of image size 32 x 32. For implementation, we trained the above-mentioned self-architecture convolutional neural network on Adam, SGD with momentum and AdamW optimization algorithms. To generate more training data and to deal with the issues of overfitting, we augmented the images by horizontally and vertically shifting, rotating and horizontally flipping the trained images.

In the recent research studies as we observed the SGD with momentum somewhat takes a bit longer in comparison with Adam and AdamW. Therefore, we trained SGD with momentum for 100 epochs in contrast with Adam and AdamW with 50 epochs. For a fixed learning rate we set the learning rate value to 0.001. For cyclical learning implemented in "triangular" mode and to cyclically oscillate the learning rate value, we specified the base_lr of 0.001 and max_lr of 0.005 Performance and results in an analysis of the experiments are mentioned in subsequent sections.

D. Analysis and Results

After training the convolutional neural network model on the datasets CIFAR-10 and SVHN with cyclical learning rates and fixed learning rates with the help of three optimization algorithms Adam, SGD and AdamW, we have collected the following results as described in Table I and Table II. We have compared the performance of our experiments based on different performances metrics which are train accuracy, test accuracy, precision and recall.

TABLE I
FIXED LEARNING RATES

Dataset	Optimizer	Accuracy	Precision	Recall
CIFAR-10	Adam	76.65%	0.78	0.77
CIFAR-10	SGD	71.64%	0.72	0.72
CIFAR-10	AdamW	78.63%	0.79	0.79
SVHN	Adam	92.14%	0.91	0.92
SVHN	SGD	91.28%	0.91	0.90
SVHN	AdamW	92.64%	0.92	0.92

TABLE II
CYCLICAL LEARNING RATES

Dataset	Optimizer	Accuracy	Precision	Recall
CIFAR-10	Adam	89.05%	0.89	0.89
CIFAR-10	SGD	87.50%	0.88	0.88
CIFAR-10	AdamW	89.52%	0.89	0.90
SVHN	Adam	94.75%	0.95	0.95
SVHN	SGD	95.13%	0.95	0.94
SVHN	AdamW	95.37%	0.95	0.95

In the Table 1 and Table 2, we can see that cyclical leaning rate gave far better results than the fixed learning rates. Also, the performance of AdamW was slightly better in every case than the other two optimization algorithms. The model accuracy graph for CIFAR -10 implemented with fixed learning rates using three optimization algorithms is plotted in Fig. 3. The model accuracy graph for CIFAR -10 implemented with cyclical learning rates using three optimization algorithms is plotted in Fig. 4. The model accuracy graph for SVHN dataset implemented with fixed learning rates using three optimization algorithms is plotted in Fig. 5. The model accuracy graph for SVHN dataset implemented with cyclical learning rates using three optimization algorithms is plotted in Fig. 6. In every case, model training using the optimization algorithms Adam and AdamW was done for 50 epochs and model training using the optimization algorithm SGD was done for 100 epoch since SGD optimizer has higher convergence than the other two optimization algorithms.

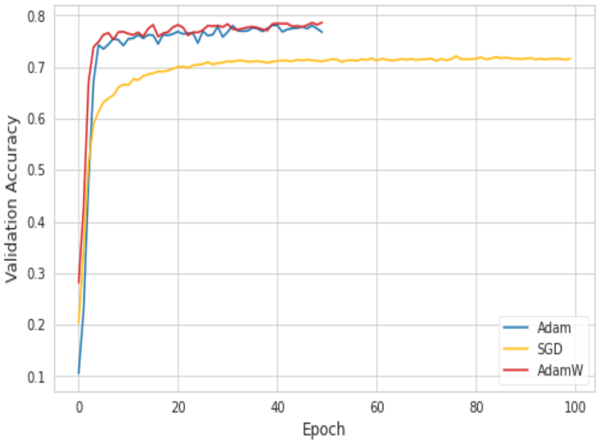

Fig. 3. Model Accuracy for CIFAR-10 dataset using FLR.

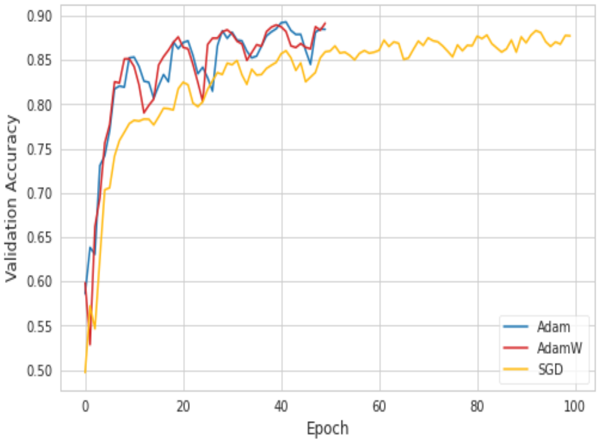

Fig. 4. Model Accuracy for CIFAR-10 dataset using CLR.

CONCLUSION

Cyclical learning rate when compared to fixed learning rate in terms of accuracy is a better technique for setting and controlling learning rates for training a neural network. The results acquired from this project demonstrated that cyclical learning rate implementation with all the three optimization algorithms performed way better in comparison to the fixed learning rate on CIFAR-10 and SVHN datasets. The cyclic policy for learning rate is easy to implement and unlike adaptive learning rate methods, incurs essentially no additional computational expense. After analyzing the results, we have concluded that the AdamW optimization algorithm implemented on convolution neural network architecture with cyclical learning rates performed better than the rest for

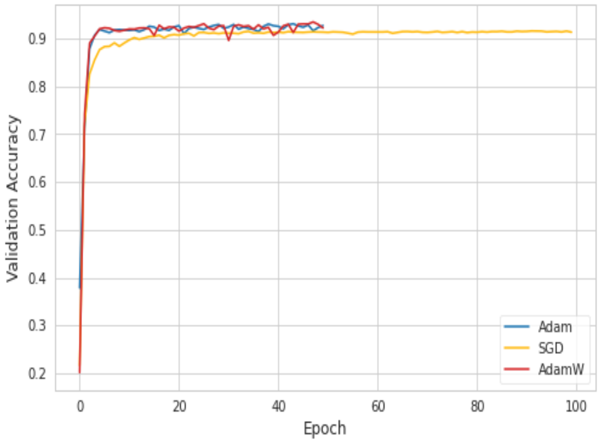

Fig. 5. Model Accuracy for SVHN dataset using FLR.

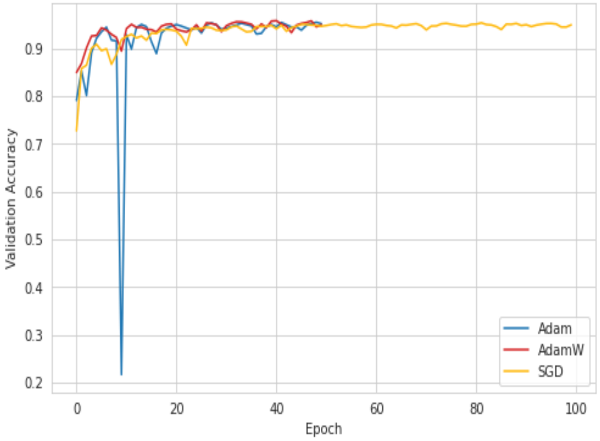

Fig. 6. Model Accuracy for SVHN dataset using CLR.

CIFAR-10 and SVHN datasets. Also, the Adam and AdamW optimization algorithms gave results with half the number of epochs in comparison with SGD who took longer to converge to optimal results. Adam with weight decay when implemented with cyclical learning rates outperformed with an accuracy of 89.52% for the CIFAR-10 dataset and with an accuracy of 95.37% for the SVHN dataset. Further future work required to enhance the overall performance of training neural networks with other advanced techniques and other architectures.

References

[1] Leslie N. Smith, “Cyclical Learning Rates for Training Neural Networks,” 2017 IEEE Winter Conference on Applications of Computer Vision (WACV),Santa Rosa, CA, USA, 2017, 10.1109/WACV.2017.58.

[2] J. Diederik P. Kingma and Jimmy Ba , Adam: A Method for Stochastic Optimization, arXiv preprint arXiv:1412.6980v9, 2015.

[3] Jinia Konar, Prerit Khandelwal and Rishabh Tripathi, “Comparison of Various Learning Rate Scheduling Techniques on Convolutional Neural Network,” 2020 IEEE International Students’ Conference on Electrical,Electronics and Computer Science (SCEECS), Bhopal, India, 2020, 10.1109/SCEECS48394.2020.94.

[4] Ilya Loshchilov and Frank Hutter, “SGDR: Stochastic Gradient Descent with Warm Restarts,”

[5] Nitish Shirish Keskar and Richard Socher, “Improving Generalization Performance by Switching from Adam to SGD,” arXiv preprint, 2017

[6] Ilya Loshchilov and Frank Hutter, “Decoupled Weight Decay Regularization,” 2019.

[7] Guodong Zhang, Chaoqi Wang, Et al. ”Three Mechanisms of Weight Decay Regularization”, Computer Science, Mathematics, Published 2019

[8] Liyuan Liu, Haoming Jiang Et al., ”On the Variance of the Adaptive Learning Rate and Beyond”, ublished 2020

[9] C. Lee, Jianfeng Liu, Wei Peng, ”Applying Cyclical Learning Rate to Neural Machine Translation”, Published 2020

[10] Andreas Søeborg Kirkedal, Yeon-Jun Kim, ”Multilingual Deep Neural Network Training Using Cyclical Learning Rate”, Published in INTERSPEECH 2018

[11] Dami Choi, Christopher J. Shallue et al., ”On Empirical Comparisons of Optimizers for Deep Learning”, Published 2019

[12] Chaoyue Liu, Mikhail Belkin, ”Accelerating SGD with momentum for over-parameterized learning”, published at ICLR 2020.

[13] Ashok Cutkosky, Harsh Mehta, ”Momentum Improves Normalized SGD”, Published in ICML 2020.

[14] Diederik P. Kingma, Jimmy Ba, ”Adam: A Method for Stochastic Optimization”, Published 2015, Computer Science, Mathematics.

[15] J. Zhang, Sai Praneeth Karimireddy et al, ”Why ADAM Beats SGD for Attention Models”, Published 2019.