Introduction to Cyclical Learning Rates for training Neural Nets

By

Sayak Paul

Project Instructor at **DataCamp**

(**Google DevFest**, Kolkata, India)

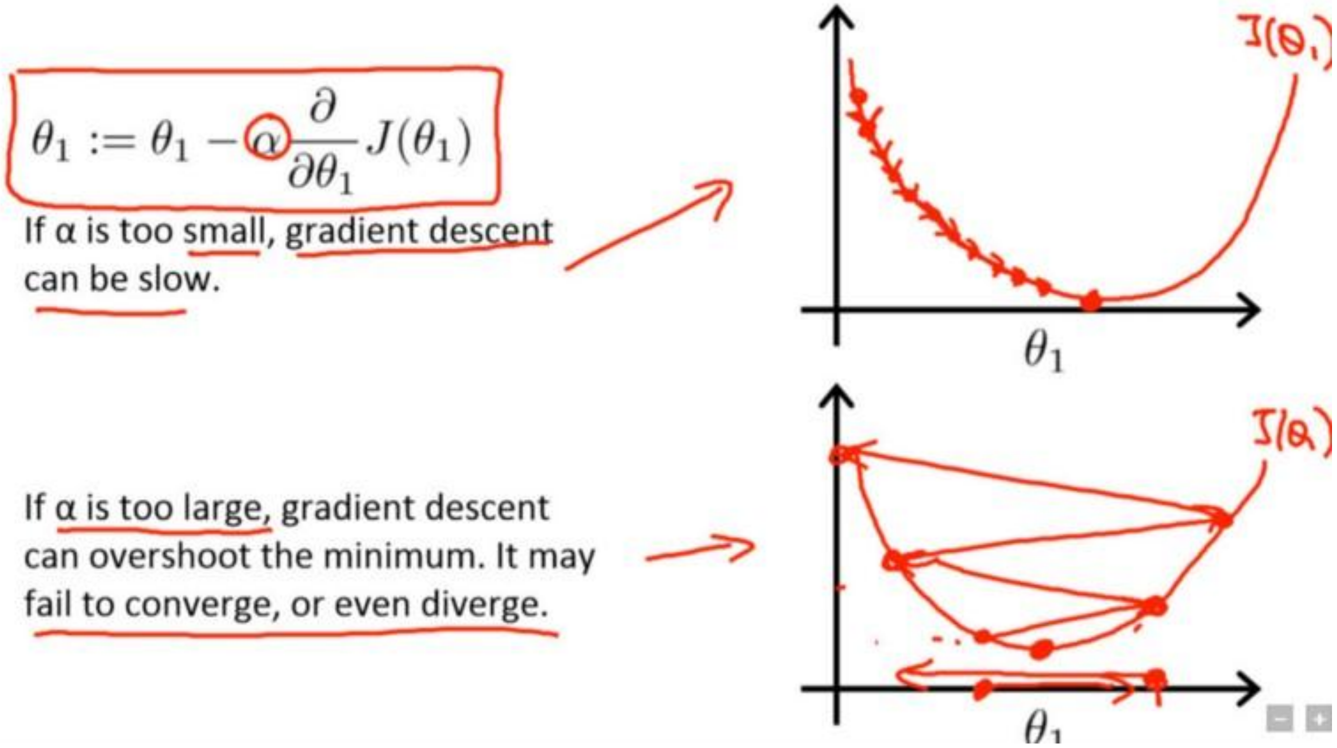
(3 – 4th November, 2018)

Overview of the talk

- Why are ***learning rates*** used?
- Some existing approaches for choosing the right learning rate
- What are the ***shortcomings of these approaches***?
- Need of a systematic approach for setting the learning rate – ***Cyclical Learning Rates*** (CLR)
- What is CLR?
- Some ***amazing results*** shown by CLR
- Conclusion

Why are *learning rates* used?

Learning is an important ***hyperparameter*** for adjusting the weights of a network with respect to the loss gradient.



$$\theta_1 := \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_1)$$

If α is too small, gradient descent can be slow.

If α is too large, gradient descent can overshoot the minimum. It may fail to converge, or even diverge.

Source: Andrew Ng's lecture notes from Coursera

Some existing approaches for choosing the right learning rate

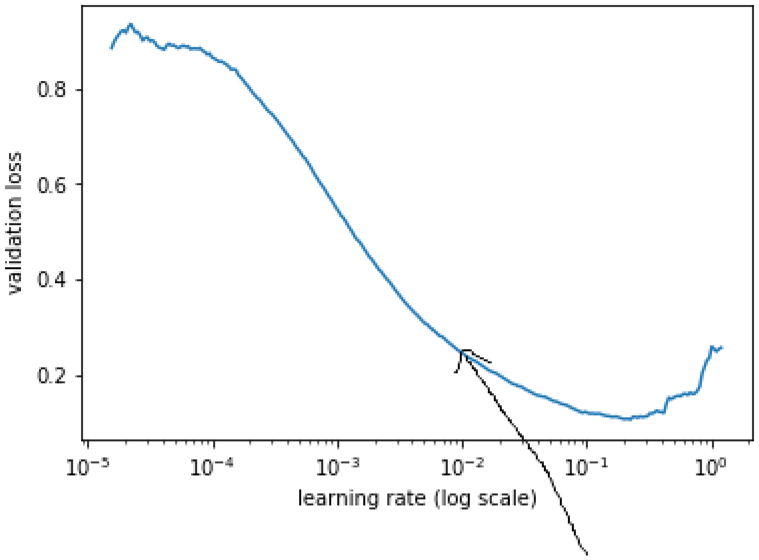
- *Trying out different learning rates* for a problem.
- ***Grid-searching/Random-searching*** over a pre-defined range of learning rates.
- ***Adaptive Learning Rates.***

Problems with the previous approaches:

- Computationally costly.
- Gives no early clue if at all the result would get better.

Cyclical Learning Rates*

- Proposed by ***Leslie N. Smith*** in his paper entitled “*Cyclical Learning Rates for Training Neural Networks*” in 2015.
- The idea is to simply keep increasing the learning rate from a very small value, until the loss stops decreasing.



The sweet spot!

Source

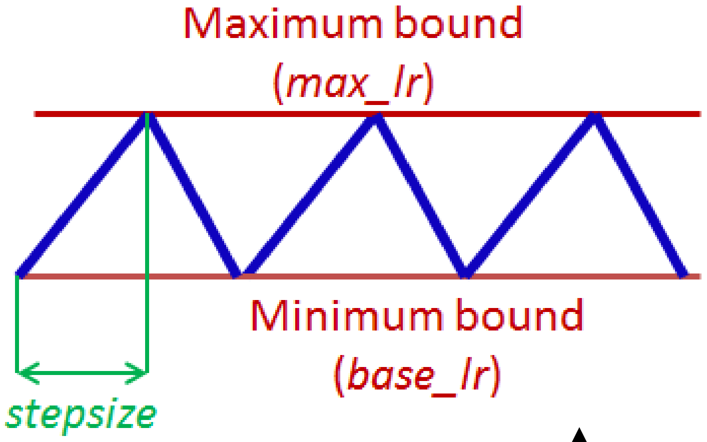
* **Cyclical Learning Rates for Training Neural Networks – Leslie N. Smith**

How are Cyclical Learning Rates (CLR) *systematic*?

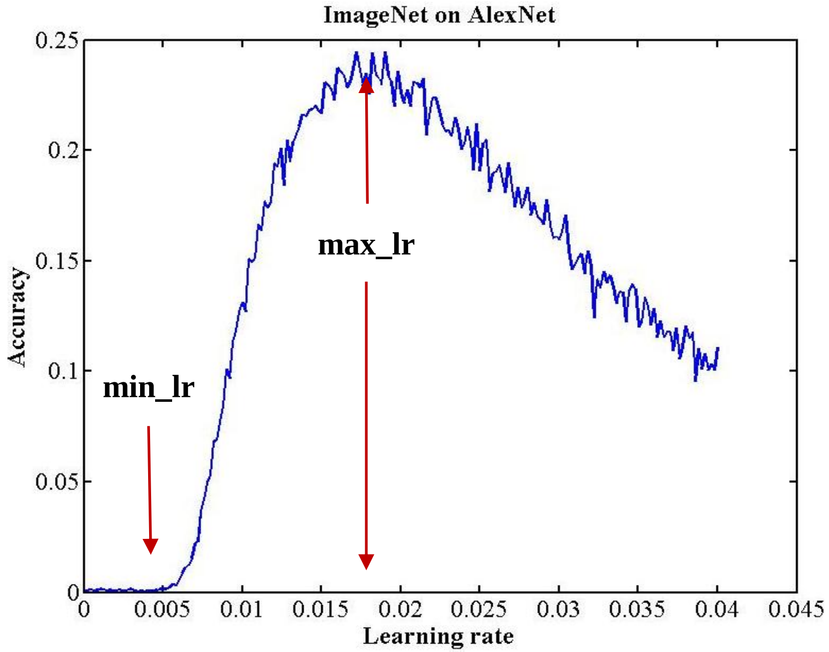
- The main idea behind CLR ***is varying learning rates*** between min and max values.
- ***LR_Range_Test()*** is conducted for fixing the min and max values of learning rate.

LR_Range_Test()

- One step of increasing learning rate.



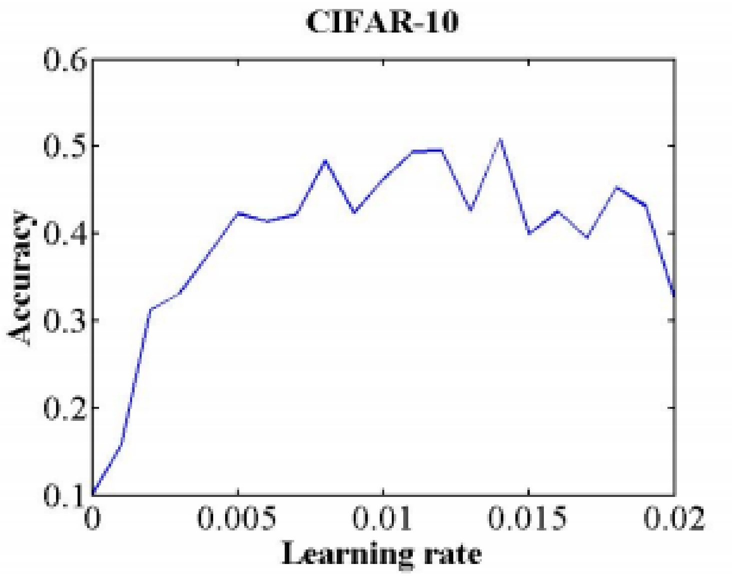
Also called **Triangular Learning Rate Policy**



Source: Cyclical Learning Rates for Training Neural Networks **– Leslie N. Smith**

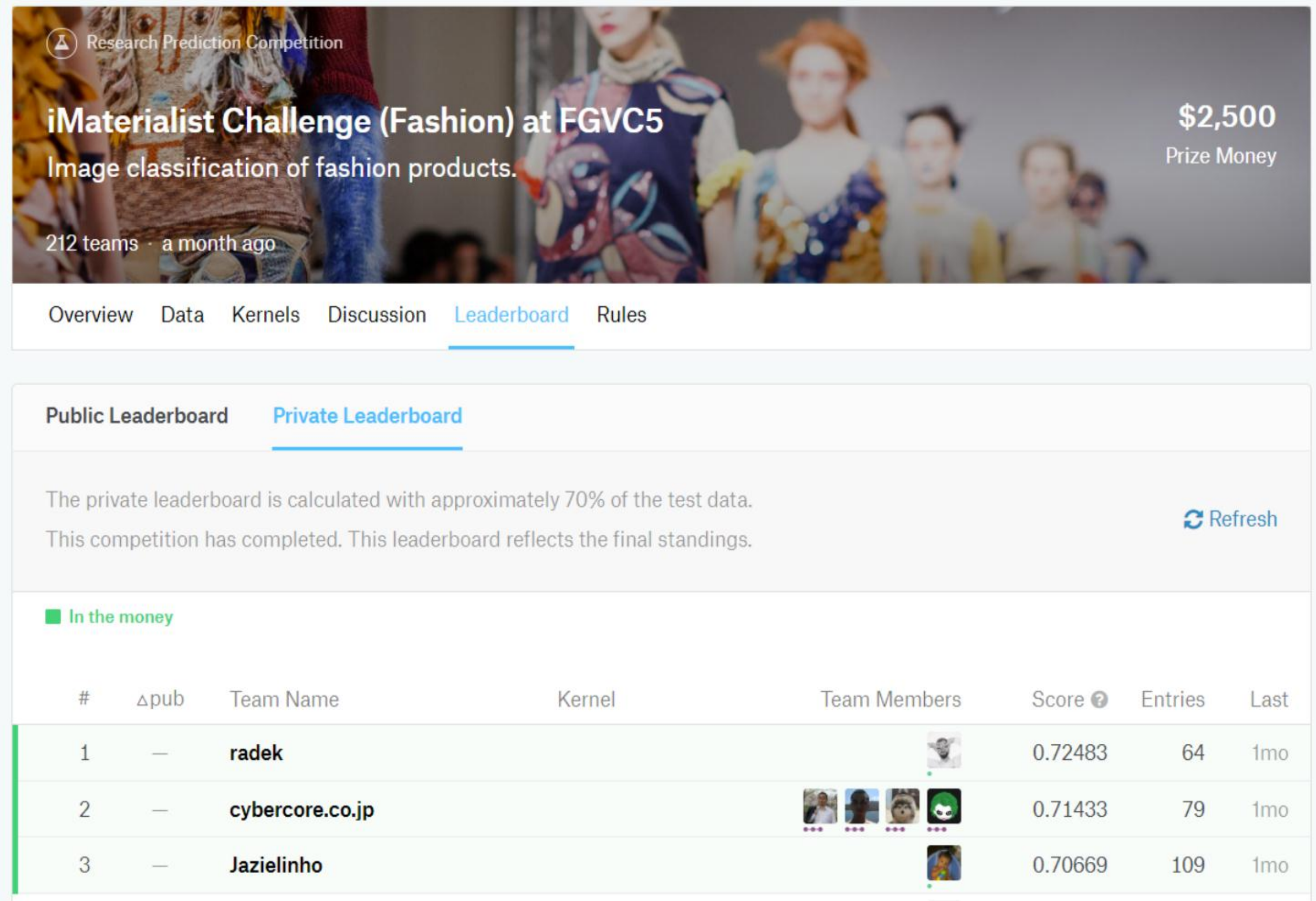
Choosing max_lr and min_lr

- Run the model for several epochs while letting the learning rate increase linearly (use triangular learning rate policy) between low and high learning rate values.
- Next, plot the **accuracy versus learning rate** curve.
- Note the learning rate value when the accuracy starts to increase and when the accuracy slows, becomes ragged, or starts to fall. These two learning rates are good choices for defining the range of the learning rates.



Source: Cyclical Learning Rates for Training Neural Networks – **Leslie N. Smith**

Some *amazing results* shown by CLR



iMaterialist Challenge (Fashion) at FGVC5

Image classification of fashion products.

212 teams · a month ago

The private leaderboard is calculated with approximately 70% of the test data.

This competition has completed. This leaderboard reflects the final standings.

🟩 In the money

#	∆pub	Team Name	Kernel	Team Members	Score	Entries	Last
1	—	radek			0.72483	64	1mo
2	—	cybercore.co.jp			0.71433	79	1mo
3	—	Jazielinho			0.70669	109	1mo

Kaggle iMaterialist Challenge (Fashion) Leaderboard

Some *amazing results* shown by CLR (contd.)

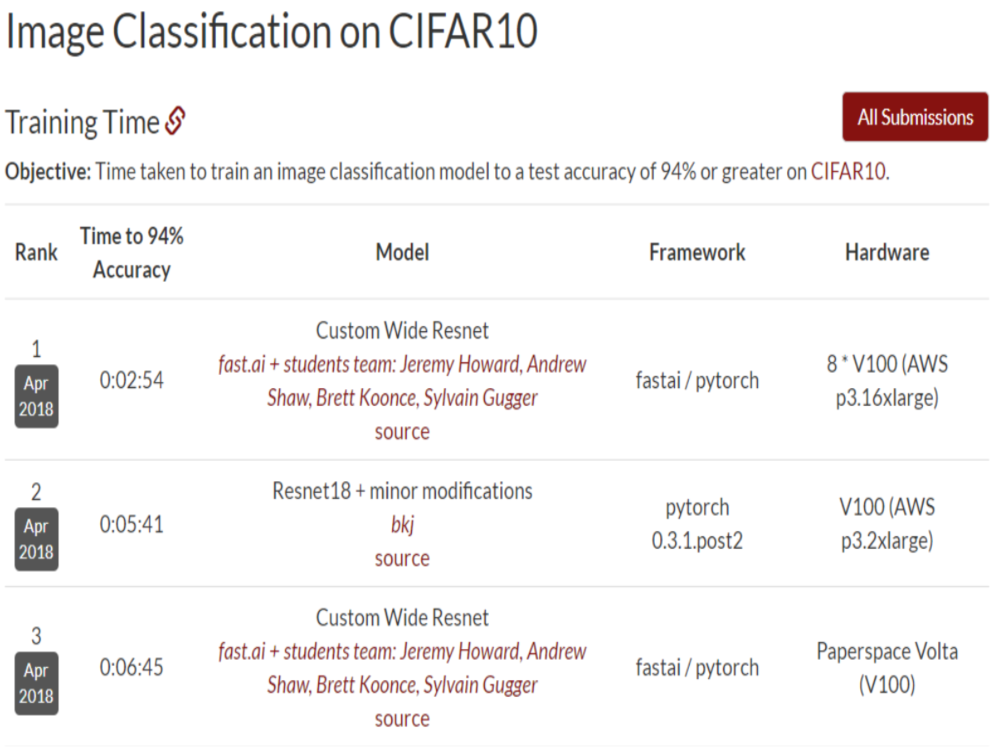


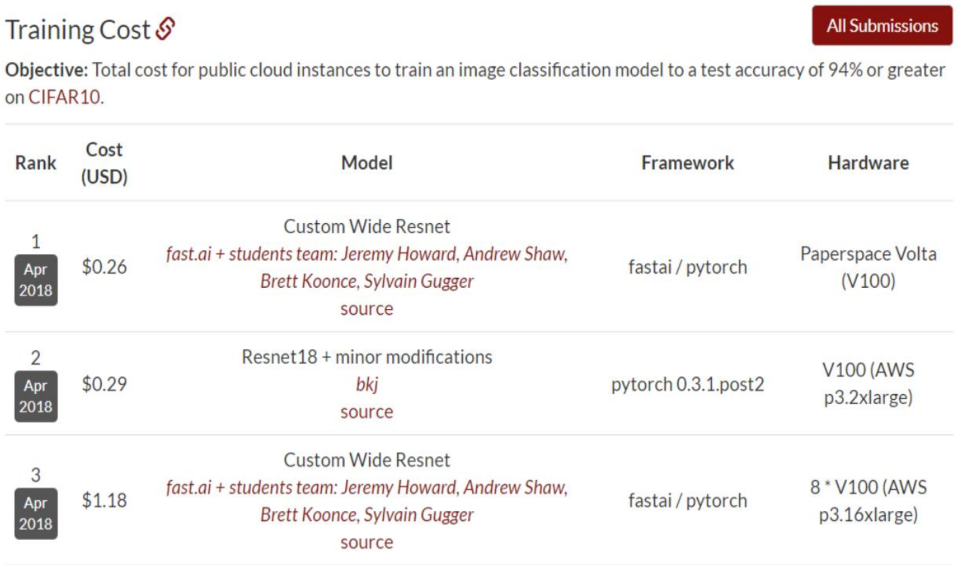
Image Classification on CIFAR10

Training Time

All Submissions

Objective: Time taken to train an image classification model to a test accuracy of 94% or greater on CIFAR10.

Rank	Time to 94% Accuracy	Model	Framework	Hardware
1 Apr 2018	0:02:54	Custom Wide Resnet *fast.ai + students team: Jeremy Howard, Andrew Shaw, Brett Koonce, Sylvain Gugger* source	fastai / pytorch	8 * V100 (AWS p3.16xlarge)
2 Apr 2018	0:05:41	Resnet18 + minor modifications *bkj* source	pytorch 0.3.1.post2	V100 (AWS p3.2xlarge)
3 Apr 2018	0:06:45	Custom Wide Resnet *fast.ai + students team: Jeremy Howard, Andrew Shaw, Brett Koonce, Sylvain Gugger* source	fastai / pytorch	Paperspace Volta (V100)



Training Cost

All Submissions

Objective: Total cost for public cloud instances to train an image classification model to a test accuracy of 94% or greater on CIFAR10.

Rank	Cost (USD)	Model	Framework	Hardware
1 Apr 2018	$0.26	Custom Wide Resnet *fast.ai + students team: Jeremy Howard, Andrew Shaw, Brett Koonce, Sylvain Gugger* source	fastai / pytorch	Paperspace Volta (V100)
2 Apr 2018	$0.29	Resnet18 + minor modifications *bkj* source	pytorch 0.3.1.post2	V100 (AWS p3.2xlarge)
3 Apr 2018	$1.18	Custom Wide Resnet *fast.ai + students team: Jeremy Howard, Andrew Shaw, Brett Koonce, Sylvain Gugger* source	fastai / pytorch	8 * V100 (AWS p3.16xlarge)

DAWNBench Challenge Leaderboard and Leader’s specs

Limitations of CLR

- Limited applicability.
- Seems to work only for **Cifar-10** and **resnets**.
- But definitely provides a more systematic way for choosing learning rate than the earlier approaches.

Notable byproducts of CLR

- Learning rate annealing (SDGR).
- Differential Learning Rates.

Some Wealth of Wisdom

- Cyclical Learning Rates for Training Neural Networks – Paper link
- Link to access the slides – https://github.com/sayakpaul/GoogleDevFestKol2018
- DataCamp tutorial covering CLR - https://goo.gl/2fpkQQ

Thank you!

Sayak Paul – spsayakpaul@gmail.com