Summary of Results for Learning Rate and Group Normalization Experiments for MGNet

Jacob McLain

The learning rate controls how much the existing weights in the model are updated in response to the gradient error function. This is one of the most important hyper-parameters in configuring and tuning neural networks, and can have significant effects on the performance of the model. Learning rate schedules are used to update the learning rate as the model is trained, often resulting in large increases in test accuracy. The following learning rate schedules were used as the basis for this experiment. An initial learning rate of 0.1 was used for all tests. A number of networks were tested, but only results for MgNet are shown below.

A standard step-decay schedule:

$$lr = lr_0(0.1^{\lfloor \frac{x_i}{30} \rfloor}) \tag{1}$$

Adjusted step-decay schedule:

$$\begin{aligned} lr &= lr_0 && x \leq 20 \\ lr &= lr_0(0.2^{\lfloor \frac{x_i - 15}{15} \rfloor}) && 20 < x \leq 60 \\ lr &= lr_0(0.1^{\lfloor \frac{x_i - 45}{15} \rfloor}) && 60 < x \leq 120 \end{aligned}$$

Exponential decay scheduling:

$$lr = lr_0(e^{-kx_i}), k = \frac{1}{10} \tag{2}$$

Cosine decay scheduling:

$$lr = \frac{lr_0}{2}(1 + \cos(\pi \frac{x_i}{x_n})) \tag{3}$$

Proposed sigmoidal decay scheduling:

$$lr = lr_0 \frac{1}{(1 + e^y)} \tag{4}$$

where the adjustment to the sigmoid curve is made through

$$y = (\frac{x_i - \frac{x_n}{2}}{\frac{x_n}{10} - c}) \tag{5}$$

LR Schedule	MgNet Test Accuracy 120 Epochs (64 channel)	MgNet Test Accuracy 60 Epochs (64 channel)	ResNet-18 Test Accuracy 120 Epochs
Std Step-Decay	92.7	92.3	94.39
Custom Step-Decay	93.29	92.3	94.88
Exponential Decay	91.71	91.73	93.29
Cosine Decay	93.43	93.12	95.05
Inverse Sigmoidal Decay	93.57	92.89	95.25

Table 1: Learning Rate Scheduling for MGNet and ResNets

The above table shows the results using the LR schedules described. Batch nomalization was applied for all tests. Based on the results, it seems that, in the case of MgNet and ResNet without soft-restart, using a sigmoidal based learning rate schedule may slightly out-compete the other LR schedules, including the cosine learning rate [1]. Additionally, using a sigmoidal learning rate may offer more customization in comparison to the cosine decay learning rate, and it is possible that a sigmoidal decay learning rate schedule could perform notably better with some fine-tuning based on the model and optimizer. It can be seen that the cosine learning rate performs better than the base sigmoidal learning rate for 60 epochs, but it is likely that the sigmoidal LR can improve upon this with some variations to the sigmoid curve. Additional tests could be conducted to see if a sigmoidal learning rate would prove effective in a soft restart scenario.

After testing some variations of the sigmoidal learning rate, it was found that the test accuracy could be improved even more by changing the value of c in equation (5), which adjusts the scale of the inverse sigmoid curve. Notable increases in test accuracy were also observed by combining and inverse sigmoidal lr schedule with a step decay. One disadvantage of the sigmoid learning rate in comparison to the cosine schedule is that the sigmoid function will asymptotically approach zero as the number of iterations increase, while the cosine rate does indeed reach zero when the wave is at its maximum amplitude. To attempt to compensate for this, a function was utilized that took the sigmoid learning rate for the first 100 epochs, and then swapped to a step decay scheduling for the final few epochs using very small learning rates (1e-4 and 5e-5). This ammendment resulted in a consistent 0.2 - 0.3 percent increase in test accuracy. Further experiments could be conducted to further verify these results, and to optimize inverse sigmoidal scheduling.

Inverese sigmoidal scheduling shows promise for its ability to accurately train models very quickly, and may actually outperform other methods of Superconvergence [2] based on limited results. Because the learning rate scales with

the number of epochs, the model is able to converge to its final accuracy more quickly, and can achieve over 93 percent accuracy in both MgNet and ResNet in only 60 epochs, when using a smaller batch size of 16. Finding a learning rate that is able to give reasonably high test accuracy in a small number of epochs would be very valuable for researchers, as it allows them to run reasonably accurate tests in a short amount of time, which could increase the rate at which research can be done in a given time period.

After establishing a promising learning rate in the form of the inverse sigmoidal scheduling, the next set of experiments focused on testing variations of Group Normalization for MgNet. Group Normalization [3] is an alternative to Batch Normalization that normalizes the data over different dimensions of the 4D-image tensor. The method has been shown to give similar results to Batch Normalization for high batch sizes on ImageNet, while also maintaining high test accuracy even when the batch size is dropped significantly. Tests using an alternative general normalization were compared with standard group normalization, with the benchmark test accuracy of 93.57 in mind from batch normalization (Table 1). All tests were run for 120 epochs using MgNet for CIFAR-10 with 64 channels and 2 iterations, using a sigmoidal learning rate.

GeneralNorm2d					
	mb 128	mb 16			mb 128
groups =	test acc %	test acc %		groups =	test acc %
[1,1,1,1]	91.34	84.68		[1,1,4,4]	91.86
[1,4,1,1]	92.11	84.95		[1,4,4,4]	92.66
[1,16,1,1]	92.65	85.51		[1,16,4,4]	91.69
[1,32,1,1]	92.87	87.28		[1,32,4,4]	91.76
[1,64,1,1]	93.43	89.03		[1,64,4,4]	91.12

Table 2: General Normalization For MgNet

The experiments show noticebly lower test accuracy than batch normalization, which regularly produces well over 93 percent test accuracy. To find if the diminished test accuracy was due to incompatibility with MGNet, an ad-

torch.nn.GroupNorm( )			
	mb 128	mb 16	mb 8
parameters	test acc %		
(1,64)	90.72	90.64	89.91
(4,64)	90.78	90.39	90.12
(16,64)	91.22	90.04	90.44
(32,64)	91.23	90.11	90.39
(64,64)	90.58	89.20	89.57

Table 3: Group Normalization for MgNet

	Batch Norm	Group Norm (4)	Group Norm (16)
Minibatch Size	Test Accuracy		
128	92.73	84.36	90.00
64	93.08	88.73	90.69
32	93.45	90.34	93.02
16	93.30	90.80	93.48
8	93.48	90.74	93.94

Table 4: Group Normalization for ResNet-56

ditional experiment was conducted in attempt to replicate published results of group normalization on ResNet [3]. ResNet-56, which is optimized for CIFAR-10, was used for this experiment. Group normalization with both 4 groups and 16 groups was tested for a range of minibatch sizes, and compared to batch normalization. Using groups of 4 shows significantly worse test accuracy than batch norm, but lower minibatch sizes of 16 or 8 work quite well with 16 groups, even outperforming batch normalization.

Attempting to replicate these same parameters in MgNet shows no increase in test accuracy (Table 3). It therefore appears that MGNet's architecture may be less suited to group normalization than ResNet. However, it is still possible that specific set of undiscovered parameters for minibatch size, learning rate, and number of groups could facilitate high test accuracy with MGNet using group normalization.

References

[1] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

[2] Leslie N Smith and Nicholay Topin. Super-convergence: Very fast training of neural networks using large learning rates. In *Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications*, volume 11006, page 1100612. International Society for Optics and Photonics, 2019.

[3] Yuxin Wu and Kaiming He. Group normalization. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19, 2018.