Adapting Regularization with Learning Rate For Better Generalization in Neural Networks

Abstract. When using gradient based methods for training neural networks, the learning rate or step size is commonly adapted as the training progresses. Usually, either a manual annealing scheme or an automatic learning rate adaptation approach is used. Prior work has shown that the learning rate itself has an implicit regularization effect on learning, with higher learning rates providing a higher degree of regularization. Typically an additional explicit regularization scheme is employed as well, with the level of this regularizer kept fixed throughout the learning process. We argue that balancing the extent of explicit regularization with the implicit regularization coming from the finite step sizes is helpful at ameliorating overfitting, particularly in the low learning rate phases of the training process. The *preliminary* results presented here should serve as an impetus to explore theoretical motivations and mechanisms for adaptive regularization in neural networks and other optimization approaches. An interesting observation is that the common practice of warm 'restarting' the learning-rate while training similarly improves generalization, and any differences between fixed and annealed regularization disappear.

Keywords: neural network, learning rate, regularization

1 Introduction

Using gradient descent with a finite step size for training neural networks (and other optimization problems) offers a degree of implicit regularization that is linked to the chosen step size (learning rate), with a higher learning rate offering more regularization. This view is also espoused in [1], where it is noted that the height above floor of the valley in the loss landscape (taken by SGD) is directly related to the learning rate. Xing et al. [1] note that maintaining a certain height above the valley floor, coupled with noise injected from mini-batches is what enables exploration of the loss landscape.

Our experiments and observations concern the other end where the learning rate is annealed down to move from the exploration phase to convergence. Dialing down the learning rate dials down the implicit regularization offered by it, and we argue that the explicit regularizer being used should be dialled up in response.

2 Background For Experiments

2.1 Learning Rate Adaptation and Annealing

There are multiple learning rate annealing [2–4] and adaptation [5] schemes that are commonly used and available in the popular Deep Learning libraries.

Adaptive learning schemes are often even used in conjunction with learning-rate annealing, with stepwise, cosine [2] and cyclical [3] annealing being the common choices.

2.2 Regularization Methods

Weight decay [6] and dropout [7] are examples of typical regularizers employed in neural network training. Other regularizers have been proposed recently which impose priors on each layer's activations [8, 9]. The noise introduced by the mini-batch size can also be used as a regularizer [10], with larger mini-batch sizes offering less regularization.

Neural networks can be regularized through constraints on learning capacity as well, either by enforcing sparsity in the learned weights [11], or through sparsity in weight updates as we and others before us [12] have observed. Our regularization approach, which we refer to as *Sparse Gradients* differs from that of [12] in that our sparsity pattern remains fixed across epochs while that of [12] updates every epoch. We resample the sparsity pattern only when changing the extent of sparsity.

We will use weight decay and *Sparse Gradients* as regularizers in our experiments.

Adaptive Regularization Since our experiments pertain to adapting the regularization with the learning rate, it would be germane to point out some of the work on adaptive regularization [13, 14], as well as some common regularization schemes interpreted through the lens of adaptive regularization [15].

3 Experiments

We conduct our experiments in the context of image classification on CIFAR10 and CIFAR100 [16]. We use ResNets [17] for CIFAR10/100 as the base architecture, evaluating for two different depths. In our experiments we will refer to the architecture with '6n+2=14' layers as *Shallow Resnet* and the architecture with '6n+2=32' layers as *Resnet*. The architecture specifics are as described in Sec 4.2 of [17]. Cross entropy is used as the training objective, and top-1 error is used as the test error.

Mini-batch SGD is used for training, with a mini-batch size of 50. Each experiment is run 5 times, and the graphs show the average, best and worst runs. The parameter and learning rate annealing schemes are shown in Table 1. The learning rate is annealed stepwise from 1e-1 to 1e-5 exponentially in 5 steps, and is repeats twice with warm 'restarts' [4] giving 3 annealing phases. The first phase has 50 epochs per learning-rate-step, and the two subsequent phases have 20 epochs per learning-rate-step.

Table 1. Learning rate and regularization taper schemes for the experiments. For experiments with varying gradient sparsity, the weight decay is held constant at 0.0001

		Weight Decay		Gradient Sparsity		
Epochs	**Learning Rate**	**Sch. 01**	**Sch. 02**	**Sch. 21**	**Sch. 22**	**Sch. 23**
1 – 50	0.1	0.0001	0.0001	0%	0%	0%
51 – 100	0.01	0.0001	0.0001	0%	0%	0%
101 – 150	0.001	0.0008	0.0004	50%	80%	30%
151 – 200	0.0001	0.0016	0.0008	50%	80%	30%
201 – 250	0.00001	0.0016	0.0008	50%	80%	30%
251 – 270	0.1	0.0001	0.0001	0%	0%	0%
271 – 290	0.01	0.0001	0.0001	0%	0%	0%
291 – 310	0.001	0.0008	0.0004	0%	0%	0%
311 - 330	0.0001	0.0016	0.0008	0%	0%	0%
331 - 350	0.00001	0.0016	0.0008	0%	0%	0%
351 – 370	0.1	0.0001	0.0001	0%	0%	0%
371 – 390	0.01	0.0001	0.0001	0%	0%	0%
391 – 410	0.001	0.0008	0.0004	0%	0%	0%
411 – 430	0.0001	0.0016	0.0008	0%	0%	0%
431 – 450	0.00001	0.0016	0.0008	0%	0%	0%

3.1 Weight Decay as Regularizer

We compare weight decay adaptation schemes described in Table 1 against constant weight decay values of 1e-3, 2e-3, 4e-3 and 8e-3. Weight decay adaptation is applied in all three phases. Scheme 01 has stronger regularization than Scheme 02.

On CIFAR10, in the first phase, Scheme 01 performs the best in terms of Test Loss (the proxy objective being minimized), with a fixed weight decay of 1e-3 (red) performing the worst for both *Resnet* (Fig 2) and *Shallow Resnet* (Fig 1) in the first phase.

In terms of the test error, the strong fixed weight decay of 8e-3 (magenta) performs the worst, while both Scheme 01 (yellow) and 02 (cyan) which use a weight decay of 8e-3 at various stages perform comparably as other fixed weight decay values.

In the subsequent phases with learning rate restarts and annealing, a fixed weight decay of 1e-3 (red) comes to perform as well as Schemes 01 and 02 on the test loss, while Scheme 01 and 02 are worse in terms of the test error.

Similar trends hold for CIFAR100 (Fig 3 and Fig 4) without worsening of the test error performance of Schemes 01 and 02 in the later phases of training.

3.2 Sparse Gradient as Regularizer

For these expriments, plots with a fixed level of gradient sparsity are not shown, because as expected, using a constant fixed gradient sparsity acts to regularize

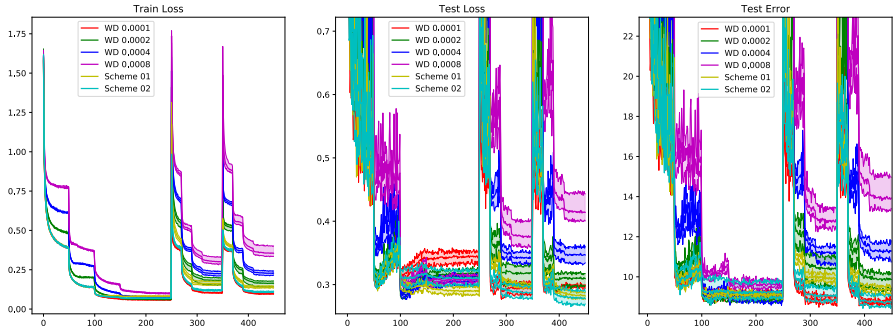

Fig. 1. Shallow Resnet CIFAR10 Weight Decay Experiments

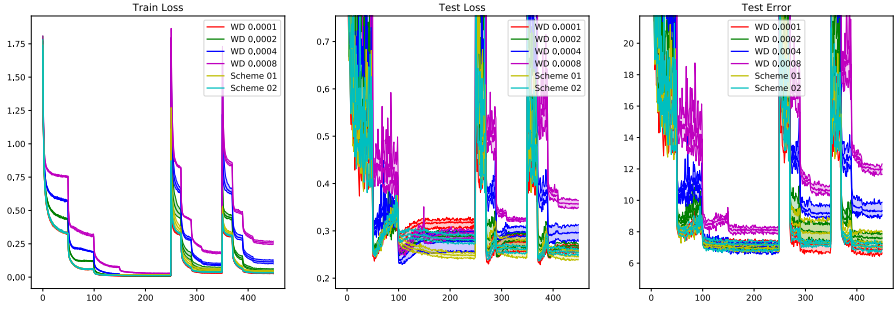

Fig. 2. Resnet CIFAR10 Weight Decay Experiments

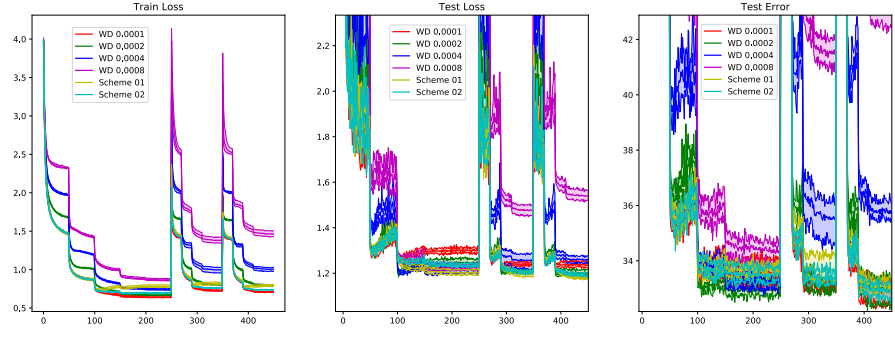

Fig. 3. Shallow Resnet CIFAR100 Weight Decay Experiments

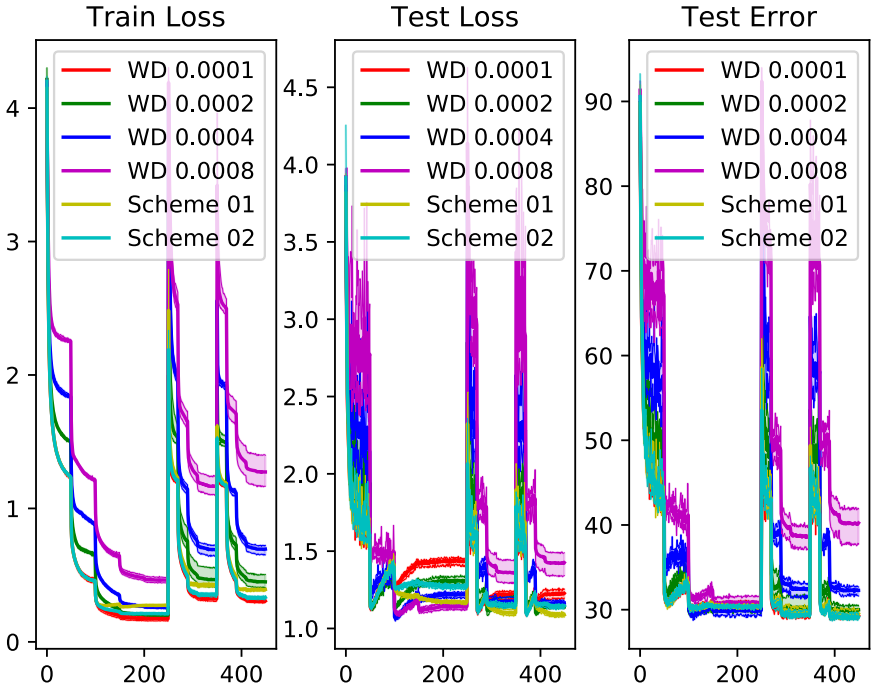

Fig. 4. Resnet CIFAR 100 Weight Decay Experiments

the effective capacity of the network, leading to a worse training and test performance. Also, there is no gradient sparsity in the last two phases of Schemes 21, 22 and 23 to emulate in some ways the re-densification step of Dense-Sparse-Dense type approaches [18, 19].

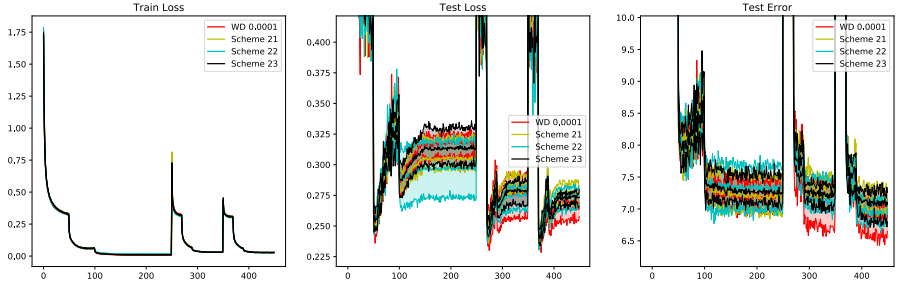

Fig. 5. Resnet CIFAR10 Gradient Sparsity Experiments

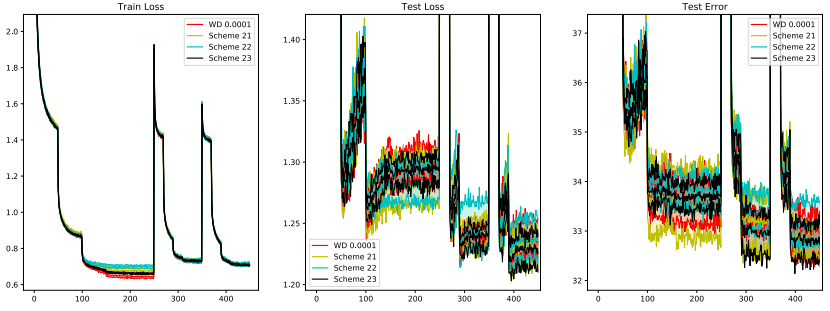

Fig. 6. Shallow Resnet CIFAR100 Gradient Sparsity Experiments

4 Conclusion

Considering the weight decay experiments, for cases where the objective being optimized is not a proxy objective, the benefit of adapting weight decay with learning rate is clear from looking at the test loss. In cases where the objective is a proxy, such as the case considered in the experiments, the test error is for the most part comparable in the first phase. It is interesting to note that learning rate spiking/restarting improves the performance of fixed low weight decay scheme sufficiently to close the test loss gap with the weight decay taper schemes.

No definite conclusions can be drawn from the gradient sparsity[1] tapering experiments, apart from that learning rate restarting/spiking and subsequent annealing should be considered for improved generalization.

5 Discussion and Future Directions

Increasing regularization with a decreasing learning rate could be used with other forms of regularization as well, such as dropout and even limited mini-batch sizes. The latter has the additional advantage of speeding up training with smaller mini-batch sizes in low learning rate regime. There is anecdotal evidence from the authors' work of the efficacy of the mini-batch tapering approach (seen for 3D body pose estimation), but extensions of the paper shall make it more formal. The proposed approach remains to be evaluated on larger datasets such as ImageNet [20], and on other problems beyond image classification.

6 On A Related Note

There is some work on using sparsification and consequent densification [18, 19] of the learned weights as a means to achieve better performance. Implicit in [18], however is the spiking of the learning rate. It is very much possible that the improvement seen in [18] may in a significant part be due to learning rate spiking and subsequent annealing, which the baseline 'LLR' does not incorporate.

[1] The *Sparse Gradient* case is included for completeness. The original motivation behind looking into gradient sparsity was to see if it speeds up the training convergence. Gradient sparsity may also be useful in distributed training settings due to the smaller number of gradients that need to be communicated.

References

1. Xing, C., Arpit, D., Tsirigotis, C., Bengio, Y.: A walk with sgd. arXiv preprint arXiv:1802.08770 (2018)
2. Huang, G., Li, Y., Pleiss, G., Liu, Z., Hopcroft, J.E., Weinberger, K.Q.: Snapshot ensembles: Train 1, get m for free. arXiv preprint arXiv:1704.00109 (2017)
3. Smith, L.N.: Cyclical learning rates for training neural networks. In: Applications of Computer Vision (WACV), 2017 IEEE Winter Conference on, IEEE (2017) 464–472
4. Loshchilov, I., Hutter, F.: Sgdr: Stochastic gradient descent with warm restarts. (2016)
5. Ruder, S.: An overview of gradient descent optimization algorithms. arXiv preprint arXiv:1609.04747 (2016)
6. Krogh, A., Hertz, J.A.: A simple weight decay can improve generalization. In: Advances in neural information processing systems. (1992) 950–957
7. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: A simple way to prevent neural networks from overfitting. The Journal of Machine Learning Research **15**(1) (2014) 1929–1958
8. Cogswell, M., Ahmed, F., Girshick, R., Zitnick, L., Batra, D.: Reducing overfitting in deep networks by decorrelating representations. arXiv preprint arXiv:1511.06068 (2015)
9. Liao, R., Schwing, A., Zemel, R., Urtasun, R.: Learning deep parsimonious representations. In: Advances in Neural Information Processing Systems. (2016) 5076–5084
10. Smith, S.L., Kindermans, P.J., Le, Q.V.: Don't decay the learning rate, increase the batch size. arXiv preprint arXiv:1711.00489 (2017)
11. Li, H., Kadav, A., Durdanovic, I., Samet, H., Graf, H.P.: Pruning filters for efficient convnets. arXiv preprint arXiv:1608.08710 (2016)
12. Skubalska-Rafajłowicz, E.: Training neural networks by optimizing random subspaces of the weight space. In: International Conference on Artificial Intelligence and Soft Computing, Springer (2016) 148–157
13. Goutte, C., Larsen, J.: Adaptive regularization of neural networks using conjugate gradient. In: Acoustics, Speech and Signal Processing, 1998. Proceedings of the 1998 IEEE International Conference on. Volume 2., IEEE (1998) 1201–1204
14. Gupta, V., Koren, T., Singer, Y.: A unified approach to adaptive regularization in online and stochastic optimization. arXiv preprint arXiv:1706.06569 (2017)
15. Wager, S., Wang, S., Liang, P.S.: Dropout training as adaptive regularization. In: Advances in neural information processing systems. (2013) 351–359
16. Krizhevsky, A., Hinton, G.: Learning multiple layers of features from tiny images. (2009)
17. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2016) 770–778
18. Han, S., Pool, J., Narang, S., Mao, H., Tang, S., Elsen, E., Catanzaro, B., Tran, J., Dally, W.J.: Dsd: Regularizing deep neural networks with dense-sparse-dense training flow. arXiv preprint arXiv:1607.04381 (2016)
19. Jin, X., Yuan, X., Feng, J., Yan, S.: Training skinny deep neural networks with iterative hard thresholding methods. arXiv preprint arXiv:1607.05423 (2016)
20. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems. (2012) 1097–1105