USING THE POLYAK STEP SIZE IN TRAINING CONVOLUTIONAL NEURAL NETWORKS

Daragh King
School of Computer Science and Statistics
Trinity College Dublin
College Green, Dublin 2, D02 PN40, Ireland
kingd6@tcd.ie

ABSTRACT

The Polyak step size (PSS) is an adaptive learning rate that has yet to see much prominence in deep learning (DL). This paper investigates using the PSS in training Convolutional Neural Networks (CNN) for image classification (IC). We show that by introducing two upper bounds for the PSS, we can train accurate CNNs without the need for calculating a learning rate apriori. Additionally, we compare the upper-bounded PSS rates against other adaptive learning rate methods and show that they achieve competitive performance.

1 INTRODUCTION AND RELATED WORKS

The PSS is an adaptive learning rate used within gradient descent and its stochastic variants (Polyak, 1987). Since gradient descent features as the predominant method for optimising DL models, it's worth exploring how learning rates such as the PSS can affect this procedure. According to the PSS, updates to the learning rate α_t are calculated as follows.

$$\alpha_t = \frac{f(\boldsymbol{x_t}) - f^*}{\nabla f(\boldsymbol{x_t})^{\mathbf{T}} \nabla f(\boldsymbol{x_t})} \tag{1}$$

In the context of DL, f in the above equation is a cost function. The PSS assumes knowing the minimum of this function (f^*), but can be assumed to be 0 for many standard cost functions in DL (e.g the cross-entropy loss for multi-class classification). It is also worth noting that the gradient values, $\nabla f(\boldsymbol{x_t})$, and the total cost, $f(\boldsymbol{x_t})$, are already known from back-propagation (Rumelhart et al., 1986). This means that calculating α_t introduces minimal computational overhead.

The PSS remains largely unstudied in DL, though recent papers such as Ren et al. (2022) explore its mathematical complexities, whilst Loizou et al. (2021) begin to study its use for training DL models. Our work reinforces the utility of the PSS in DL, showing that an upper-bounded PSS can train accurate CNNs without an apriori learning rate choice.

2 METHOD AND RESULTS

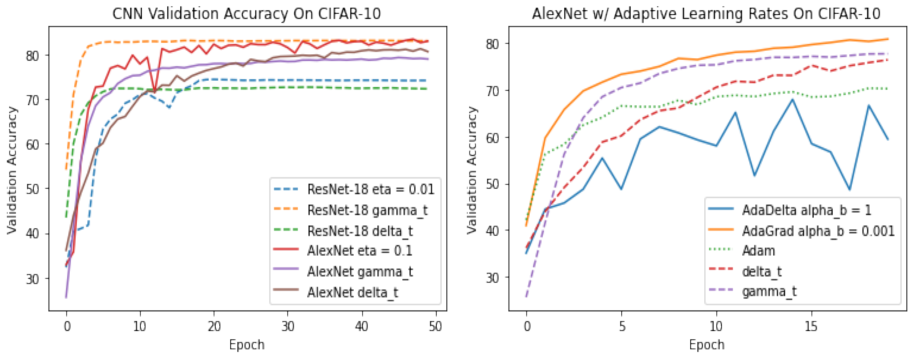
We introduce the following two distinct methods of placing an upper-bound on α_t.

$$\gamma_t = \min\{\alpha_t, 1\} \tag{2}$$

$$\delta_t = \min\{\gamma_t, \delta_{t-1}\} \tag{3}$$

The need for these upper-bounds arose from our initial empirical investigations. In the early stages of CNN training, large losses and comparatively small gradients resulted in values of $\alpha_t >> 1$. This resulted in excessively large model-parameter updates and poor predictive performance. Using equation 2 allows the learning rate γ_t to remain within the range $(0, 1]$, and equation 3 ensures that δ_t progresses in a descending, step-like manner.

In order to have a baseline to compare the above methods against, we trained AlexNet (Krizhevsky et al., 2012) and ResNet-18 (He et al., 2016) on the CIFAR-10 dataset (Krizhevsky et al., 2009). The



(a) Using γ_t, δ_t and η

(b) Using γ_t, δ_t, AdaGrad, AdaDelta, Adam

Figure 1: Validation Accuracies obtained using different learning rates.

constant η learning rates used for these base models were 0.1 and 0.01 respectively, these acted as preferred initial learning rate choices and were determined using grid-search. The aforementioned set ups (and their equivalent using γ_t and δ_t) were trained for 50 epochs, using a batch size of 64.

Fig. 1(a) clearly shows that the CNNs using γ_t and δ_t achieved competitive performance relative to those that used η rates. In particular, ResNet-18 using the γ_t learning rate achieved the highest validation accuracy in the least number of epochs. This confirms that using an upper-bounded PSS rate is a viable alternative to choosing a learning rate apriori, this is most useful when the time and computational resources to perform extensive hyper-parameter optimisation (e.g grid-search for η) are unavailable.

Following the above, we must ask how γ_t and δ_t fare against other adaptive learning rates? To answer this question we compare our proposed methods against AdaGrad (Duchi et al., 2011) with a base learning rate α_b that was determined using grid search, AdaDelta (Zeiler, 2012) with an α_b of 1 (as per the original paper) and Adam (Kingma & Ba, 2017) using the default hyper-parameters. Fig. 1(b) depicts the validation accuracies obtained by all methods whilst training AlexNet on CIFAR-10. Though AdaGrad outperforms γ_t and δ_t in this instance, it's worth re-emphasising the initial time cost needed for calculating AdaGrad's α_b. Again, we note that γ_t and δ_t provide competitive performance (surpassing Adam and AdaDelta) whilst doing away with this upfront cost.

3 Conclusions and Future Directions

The introduced learning rates γ_t and δ_t resulted in varied performance characteristics while training different CNN architectures (see those trained with δ_t in Fig. 1(a)). This poses questions for further study (e.g when should δ_t increase?). Despite this, the introduced rates proved to be a competitive option for training CNNs - one that discards the need for deciding a learning rate apriori. Additional future works would apply γ_t and δ_t to training larger CNN architectures on more complex datasets.

URM Statement

The author acknowledges that they meet the URM criteria of the ICLR 2024 Tiny Papers Track.

References

John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(61):2121–2159, 2011. URL http://jmlr.org/papers/v12/duchi11a.html.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger (eds.), *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012. URL `https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf`.

Nicolas Loizou, Sharan Vaswani, Issam Hadj Laradji, and Simon Lacoste-Julien. Stochastic polyak step-size for sgd: An adaptive learning rate for fast convergence. In *International Conference on Artificial Intelligence and Statistics*, pp. 1306–1314. PMLR, 2021.

Boris T Polyak. Introduction to optimization. 1987.

Tongzheng Ren, Fuheng Cui, Alexia Atsidakou, Sujay Sanghavi, and Nhat Ho. Towards statistical and computational complexities of polyak step size gradient descent. In *International Conference on Artificial Intelligence and Statistics*, pp. 3930–3961. PMLR, 2022.

David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.

Matthew D Zeiler. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012.

4 APPENDIX

4.1 GRID-SEARCH DETAILS

The following array of learning rates was used when determining the η values (used as constant learning rates in training the baseline AlexNet and ResNet-18 models) and α_b (for AdaGrad usage when training AlexNet), using 10-fold, cross-validated grid-search.

$$\boldsymbol{\alpha} = [0.5, 0.1, 0.01, 0.001, 0.0001] \tag{4}$$

4.2 CNN TRAINING SETUPS

The following describes the setup for the baseline models trained using constant η learning rates.

Dataset	Model	η	Batch size	Epochs
CIFAR-10	AlexNet	0.1	64	50
	ResNet-18	0.01	64	50

The following describes the setup for the models trained using our proposed γ_t and δ_t learning rates.

Dataset	Model	γ_t	δ_t	Batch size	Epochs
CIFAR-10	AlexNet	X	✓	64	50
	AlexNet	✓	X	64	50
	ResNet-18	X	✓	64	50
	ResNet-18	✓	X	64	50

The following describes the setup for the models trained using AdaGrad and AdaDelta methods.

Dataset	Model	AdaGrad	AdaDelta	α_b	Batch size	Epochs
CIFAR-10	AlexNet	X	✓	1	64	20
	AlexNet	✓	X	0.001	64	20

4.3 UNBOUNDED POLYAK STEP SIZE VALUES

The following table describes the setup for a model trained using the unbound PSS. The learning rates derived in this training process are depicted below in Fig. 2.

Dataset	Model	Batch size	Epochs
CIFAR-10	AlexNet	64	20

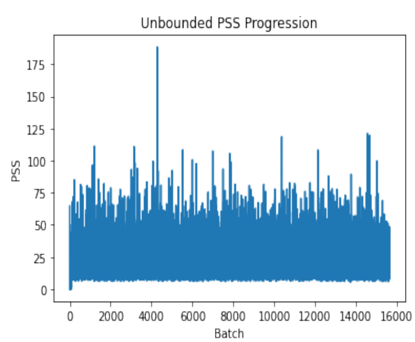
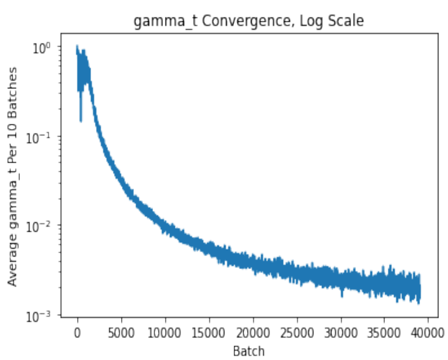
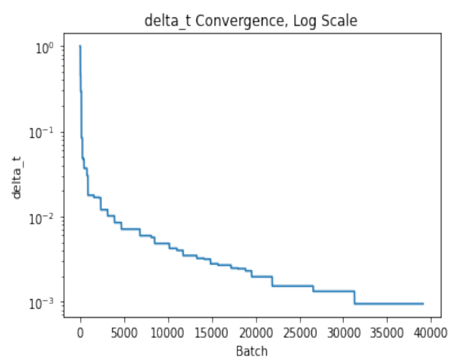


Figure 2: Without an upper-bound the PSS values far exceed 1.

The following graph depicts how the γ_t and δ_t rates progress during training.



(a) γ_t progression



(b) δ_t progression

Figure 3: γ_t and δ_t progressions whilst training AlexNet on CIFAR-10.