ADAMP: SLOWING DOWN THE SLOWDOWN FOR MOMENTUM OPTIMIZERS ON SCALE-IN Variant WEIGHTS

Byeongho Heo,¹ Sanghyuk Chun,¹ Seong Joon Oh,¹ Dongyoon Han,¹ Sangdoo Yun,¹ Gyuwan Kim,¹ Youngjung Uh,² and Jung-Woo Ha¹

1) Naver AI Lab
2) Applied Information Engineering, Yonsei University (Works done at Naver AI Lab)

Byeongho Heo and Sanghyuk Chun contribute equally.

ABSTRACT
Normalization techniques, such as batch normalization (BN) [2], have become standard tools for training deep neural network models. Originally proposed to reduce the internal covariate shift [2], normalization methods have proven to encourage several desirable properties in deep neural networks, such as better generalization and the scale invariance [3]. Prior studies have observed that the normalization-induced scale invariance of weights stabilizes the convergence
for the neural network training [3,4]. We provide a sketch of the argument here. Given weights \( w \) and an input \( x \), we observe that the normalization makes the weights become scale-invariant:

\[
\text{Norm}(w^\top x) = \text{Norm}(cw^\top x) \quad \forall c > 0.
\] (1)

The resulting equivalence relation among the weights lets us consider the weights only in terms of their \( \ell_2 \)-normalized vectors \( \hat{w} := \frac{w}{\|w\|_2} \) on the sphere \( S^{d-1} = \{ v \in \mathbb{R}^d : \|v\|_2 = 1 \} \). We refer to \( S^{d-1} \) as the effective space, as opposed to the nominal space \( \mathbb{R}^d \) where the actual optimization algorithms operate. The mismatch between these spaces results in the discrepancy between the gradient descent steps on \( \mathbb{R}^d \) and their effective steps on \( S^{d-1} \). Specifically, for the gradient descent updates, the effective step sizes \( \|\Delta \hat{w}_{t+1}\|_2 := \|\hat{w}_{t+1} - \hat{w}_t\|_2 \) are the scaled versions of the nominal step sizes \( \|\Delta w_{t+1}\|_2 := \|w_{t+1} - w_t\|_2 \) by the factor \( \frac{1}{\|w_t\|_2} \) [3]. Since \( \|w_t\|_2 \) increases during training [4], the effective step sizes \( \|\Delta \hat{w}_t\|_2 \) decrease as the optimization progresses. The automatic decrease in step sizes stabilizes the convergence of gradient descent algorithms applied on models with normalization layers: even if the nominal learning rate is set to a constant, the theoretically optimal convergence rate is guaranteed [4].

In this work, we show that the widely used momentum-based gradient descent optimizers decreases the effective step size \( \Delta \hat{w}_t \) even more rapidly than the momentum-less counterparts considered in [4]. This leads to a slower convergence for \( \hat{w}_t \) and potentially sub-optimal model performances. This phenomenon is not confined to the toy setup, for example, 95.5% and 91.8% of the parameters of the widely-used ResNet18 and ResNet50 [5] are scale-invariant due to BN.

We propose a simple solution to slow down the decay of effective step sizes while maintaining the step directions of the original optimizer in the effective space. At each iteration of a momentum-based gradient descent optimizer, we propose to project out the radial component (i.e. component parallel to \( w \)) from the update, thereby reducing the increase in the weight norm over time. Because of the scale invariance, the procedure does not alter the update direction in the effective space; it only changes the effective step sizes. We apply this technique on SGD and Adam [6] (SGDP and AdamP, respectively) and verify the resulting performance boosts over a diverse set of practical machine learning tasks.

REFERENCES


