An Optimization Strategy Based on Hybrid Algorithm of Adam and SGD

Yijun Wang[1], Pengyu Zhou[2,a] and Wenya Zhong[3]

[1,2]Software Engineering, Central South University, 410075 Changsha, China
[3] Biological Engineering, Nanyang Normal University, 473000 Nanyang,China

Abstract: Despite superior training outcomes, adaptive optimization methods such as Adam, Adagrad or RMSprop have been found to generalize poorly compared to stochastic gradient descent (SGD). So scholars (Nitish Shirish Keskar et al.,2017) proposed a hybrid strategy to start training with Adam and switch to SGD at the right time. In the learning task with a large output space, it was observed that Adam could not converge to an optimal solution (or could not converge to an extreme point in a non-convex scene) [1]. Therefore, this paper proposes a new variant of the ADAM algorithm (AMSGRAD), which not only solves the convergence problem, but also improves the empirical performance.

1 Introduction

Stochastic gradient descent (SGD) [2] has emerged as one of the most used training algorithms for deep neural networks. Despite its simplicity, SGD performs well empirically across a variety of applications but also has strong theoretical foundations. One disadvantage of SGD is that it scales the gradient uniformly in all directions; this can be particularly detrimental for ill-scaled problems. This also makes the process of tuning the learning rate α circumstantially laborious. To correct for these shortcomings, several adaptive methods have been proposed which diagonally scale the gradient via estimates of the function's curvature. Examples of such methods include Adam [3], Adagrad [4] and RMSprop [5]. These methods can be interpreted as methods that use a vector of learning rates, one for each parameter, which are adapted as the training algorithm progresses. Interestingly however, in these and other instances, Adam outperforms SGD in both training and generalization metrics in the initial portion of the training, but then the performance stagnates. To investigate this further, [6] propose SWATS, a simple strategy that combines the best of both worlds by Switching from Adam to SGD. So this paper proposes an optimization strategy based on Adam and SGD hybrid algorithm to guarantee the convergence of Adam.

2 Introduction to basic SGD and Adam algorithms

Training neural networks is equivalent to solving the following non-convex optimization problem,

$$\min_{w \in R^n} f(w),$$

where f is a loss function. The iterations of SGD can be described as:

$$w_k = w_{k-1} - \alpha_{k-1}\, \widehat{\nabla} f(w_{k-1}),$$

where w_k denotes the k^{th} iterate, α_k is a (tuned) step size sequence, also called the learning rate, and $\widehat{\nabla} f(w_k)$ denotes the stochastic gradient computed at w_k.

And Math ematically, the Adam update equation can be represented as:

$$w_k = w_{k-1} - \alpha_{k-1} \cdot \frac{\sqrt{1-\beta_2^k}}{\sqrt{1-\beta_1^k}} \cdot \frac{m_{k-1}}{\sqrt{v_{k-1}}+\epsilon}, \text{ where} \quad (1)$$

$$m_{k-1} = \beta_1 m_{k-2} + (1-\beta_1)\widehat{\nabla} f(w_{k-1}), \quad (2)$$

$$v_{k-1} = \beta_2 v_{k-2} + (1-\beta_2)\widehat{\nabla} f(w_{k-1})^2, \quad (3)$$

Among them,β_1, β_2 are two hyperparameters. The former controls first-order momentum and the latter controls second-order momentum.

3 SWATS strategy

Given the insights of [7] which suggest that the lack of generalization performance of adaptive methods stems from the non-uniform scaling of the gradient, a natural hybrid strategy would begin the training process with Adam and switch to SGD when appropriate. To investigate this further, Nitish Shirish Keskar propose SWATS, a simple strategy that combines the best of both worlds by Switching from Adam to SGD.

3.1 Learning rate for SGD after the switch

Adam's descending direction is:

$$\eta_t^{Adam} = ({}^{\alpha}/_{\sqrt{V_t}}) \cdot m_t$$

The downward direction of SGD is:

[a] Corresponding author:pyzhoucsu@163.com

$$\eta_t^{SGD} = \alpha^{SGD} \cdot g_t$$

η_t^{SGD} must be decomposed into the sum of the direction of η_t^{Adam} and the two directions in the orthogonal direction, Then its projection in the direction of η_t^{Adam} means the distance that SGD advances in the direction of decline determined by the Adam algorithm, The projection in the orthogonal direction of η_t^{Adam} is the distance that SGD advances in its own selected correction direction. Figure 1:

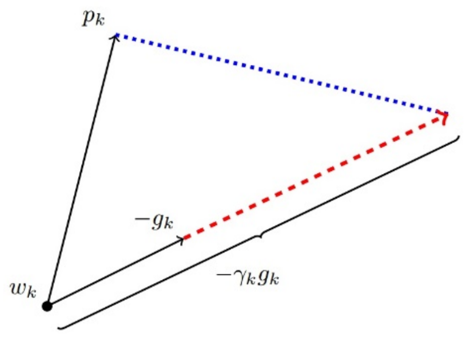

Figure 1: Illustrating the learning rate for SGD (γ_k) estimated by our proposed projection given an iterate w_k , a stochastic gradient g_k and the Adam step p_k.

You can get an orthogonal projection of SGD in the direction of Adam's descent, which should be exactly equal to Adam's descent direction (with step size).

$$proj_{\eta^{SGD}} = \eta_t^{Adam}$$

It also gets the converted SGD learning rate:

$$\alpha_t^{SGD} = ((\eta_t^{Adam})^T \eta_t^{Adam})/((\eta_t^{Adam})^T g_t)$$

Since γ_k is a noisy estimate of the scaling needed, we main tain an exponential average initialized at 0, denoted by λ_k such that

$$\lambda_k = \beta_2 \lambda_{k-1} + (1-\beta_2)\gamma_k$$

We use β_2 of Adam, see (3), as the averaging coefficient since this reuse avoids another hyperparameter and also because the performance is relatively invariant to fine-grained specification of this parameter.

3.2 Switchover point

Having answered the question of what learning rate λ_k to choose for SGD after the switch, we now discuss when to switch to SGD. We propose checking a simple, yet powerful, criterion:

$$\left|\frac{\lambda_k}{1-\beta_2^k} - \gamma_k\right| < \epsilon, \qquad (4)$$

at every iteration with k > 1. The condition compares the bias-corrected exponential averaged value and the current value (γ_k). The bias correction is necessary to prevent the influence of the zero initialization during the initial portion of training. Once this condition is true, we switch over to SGD with learning rate $\Lambda := \frac{\lambda_k}{1-\beta_2^k}$. Nitish Shirish Keskar also experimented with more complex criteria including those involving monitoring of gradient norms. However, we found that this simple un-normalized criterion works well across a variety of different applications.

4 The non-convergence of Adam

[8] discuss fundamental flaw in the current exponential moving average methods like ADAM. They show that ADAM can fail to converge to an optimal solution even in simple one-dimensional convex settings. These examples of non-convergence contradict the claim of convergence in (Kingma & Ba, 2015), and the main issue lies in the following quantity of interest:

$$\Gamma_{t+1} = \left(\frac{\sqrt{V_{t+1}}}{\alpha_{t+1}} - \frac{\sqrt{V_t}}{\alpha_t}\right). \qquad (5)$$

This quantity essentially measures the change in the inverse of learning rate of the adaptive method with respect to time. One key observation is that for SGD and ADAGRAD, $\Gamma_t \succcurlyeq 0$ for all $t \in [T]$. In particular, update rules for these algorithms lead to "non-increasing" learning rates. However, this is not necessarily the case for exponential moving average variants like ADAM, Γ_t can potentially be indefinite for $t \in [T]$. Sashank

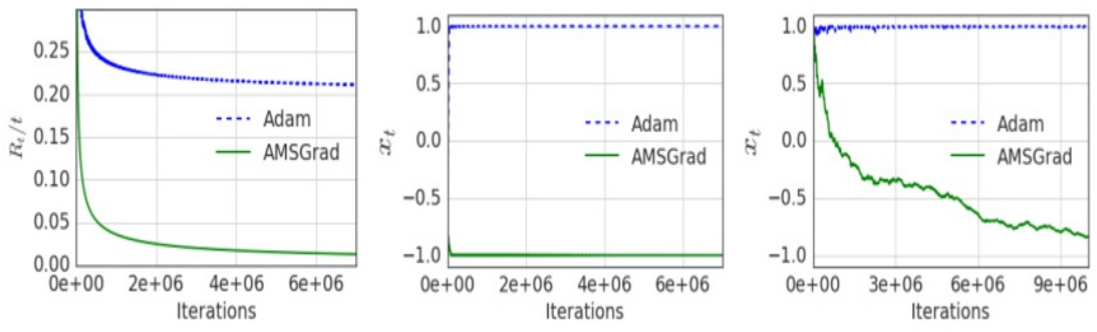

Figure 2: Performance comparison of A DAM and AMSG RAD on synthetic example on a simple one dimensional convex problem inspired by our examples of non-convergence. The first two plots (left and center) are for the online setting and the last one (right) is for the stochastic setting.

show that this violation of positive definiteness can lead to undesirable convergence behavior for ADAM. Consider the following simple sequence of linear functions for $\mathcal{F} = [-1,1]$:

$$f_t(x) = \begin{cases} Cx, & for\ t\ mod\ 3 = 1 \\ -x, & otherwise, \end{cases}$$

where C > 2. For this function sequence, it is easy to see that the point x = −1 provides the minimum regret. Suppose $\beta_1 = 0$ and $\beta_2 = 1/(1 + C^2)$. We show that ADAM converges to a highly suboptimal solution of x = +1 for this setting. Intuitively, the reasoning is as follows. The algorithm obtains the large gradient C once every 3 steps, and while the other 2 steps it observes the gradient -1, which moves the algorithm in the wrong direction. The large gradient C is unable to counteract this effect since it is scaled down by a factor of almost C for the given value of β_2 , and hence the algorithm converges to 1 rather than -1.

5 A new type of exponential moving average method: AMSGRAD

In this section, we develop a new principled exponential moving average variant and provide its convergence analysis. Our aim is to devise a new strategy with guaranteed convergence while preserving the practical benefits of ADAM. To understand the design of our algorithms, let us revisit the quantity Γ_t in (5). For ADAM, this quantity can potentially be negative. The proof in the original paper of ADAM erroneously assumes that Γ_t is positive semi-definite and is hence, incorrect. For the first part, we modify these algorithms to satisfy this additional constraint. Later on, we also explore an alternative approach where Γ_t can be made positive semi-definite by using values of β_1 and β_2 that change with t.

Algorithm 1 AMSGRAD

```
Input: $x_1 \in \mathcal{F}$, step size $\{\alpha_t\}_{t=1}^{T}$, $\{\beta_{1t}\}_{t=1}^{T}$, $\beta_2$
Set $m_0 = 0$, $v_0 = 0$ and $\hat{v}_0 = 0$
for $t = 1$ to $T$ do
    $g_t = \nabla f_t(x_t)$
    $m_t = \beta_{1t} m_{t-1} + (1 - \beta_{1t}) g_t$
    $v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2$
    $\hat{v}_t = \max(\hat{v}_{t-1}, v_t)$ and $\hat{V}_t = \mathrm{diag}(\hat{v}_t)$
    $x_{t+1} = \Pi_{\mathcal{F}, \sqrt{\hat{V}_t}}(x_t - \alpha_t m_t / \sqrt{\hat{v}_t})$
end for
```

AMSGRAD uses a smaller learning rate in comparison to ADAM and yet incorporates the intuition of slowly decaying the effect of past gradients on the learning rate as long as Γ t is positive semi-definite. Algorithm 1 presents the pseudocode for the algorithm. The key difference of AMSGRAD with ADAM is that it maintains the maximum of all v_t until the present time step and uses this maximum value for normalizing the running average of the gradient instead of v_t in ADAM. By doing this, AMSGRAD results in a non-increasing step size and avoids the pitfalls of ADAM, $\Gamma_t \geqslant 0$ for all $t \in [T]$ even with constant β_2. Also, in Algorithm 1, one typically uses a constant β_{1t} in practice (although, the proof requires a decreasing schedule for proving convergence of the algorithm).

To gain more intuition for the updates of AMSGRAD, it is instructive to compare its update with ADAM and ADAGRAD. Suppose at particular time step t and coordinate $i \in [d]$, we have $v_{t-1,i} > g_{t,i}^2 > 0$, then ADAM aggressively increases the learning rate, however, as we have seen in the previous section, this can be detrimental to the overall performance of the algorithm. On the other hand, ADAGRAD slightly decreases the learning rate, which often leads to poor performance in practice since such an accumulation of gradients over a large time period can significantly decrease the learning rate. In contrast, AMSGRAD neither increases nor decreases the learning rate and furthermore, decreases v_t which can potentially lead to non-decreasing learning rate even if gradient is large in the future iterations.

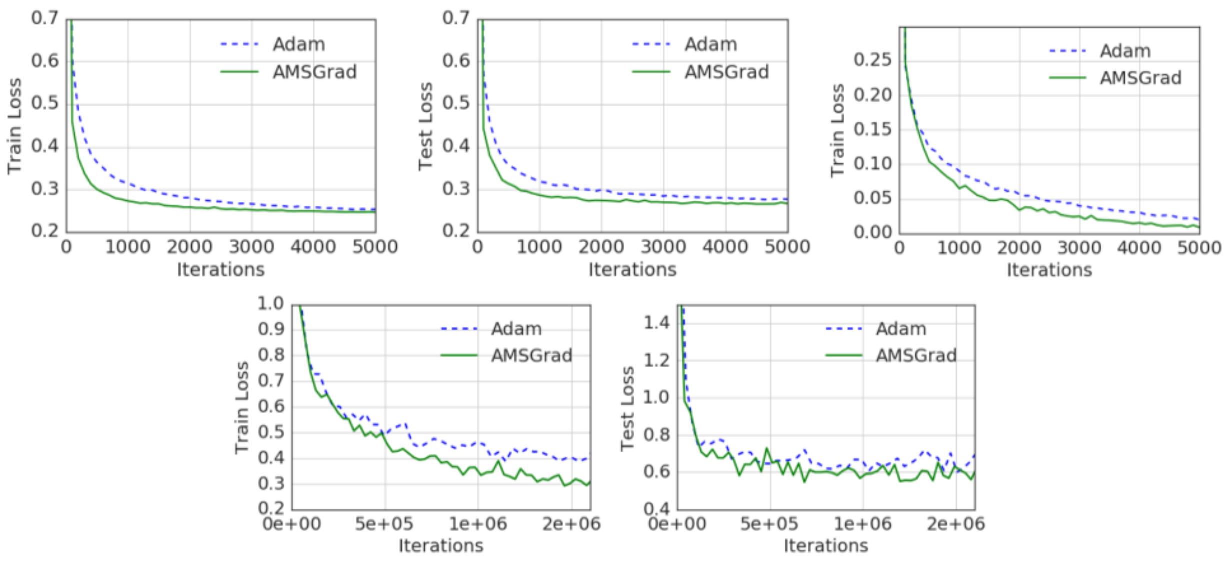

Figure 3: Performance comparison of ADAM and AMSGRAD for logistic regression, feed forward neural network and CIFARNET. The top row shows performance of ADAM and AMSGRAD on logistic regression (left and center) and 1-hidden layer feedforward neural network (right) on MNIST. In the bottom row, the two plots compare the training and test loss of ADAM and AMSGRAD with respect to the iterations for CIFARNE

6 Experiments

To demonstrate the convergence issue of ADAM, we first consider the following simple convex setting inspired from our examples of non-convergence:

$$f_t(\mathrm{x}) = \begin{cases} 1010x\,, for\ t\ mod\ 101 = 1 \\ -10x, \quad otherwise, \end{cases}$$

with the constraint set $\mathcal{F}$ = [−1,1]. We first observe that, similar to the examples of non-convergence we have considered, the optimal solution is x = −1; thus, for convergence, we expect the algorithms to converge to x = −1. For this sequence of functions, we investigate the regret and the value of the iterate x_t for ADAM and AMSGRAD. To enable fair comparison, we set $\beta_1 = 0.9$ and $\beta_2 = 0.99$ for ADAM and AMSGRAD algorithm, which are typically the parameters settings used for ADAM in practice. Figure 2 shows the average regret (R_t /t) and value of the iterate (x_t) for this problem. We first note that the average regret of ADAM does not converge to 0 with increasing t. Furthermore, its iterates x_t converge to x = 1, which unfortunately has the largest regret amongst all points in the domain. On the other hand, the average regret of AMSGRAD converges to 0 and its iterate converges to the optimal solution. Figure 2 also shows the stochastic optimization setting:

$$f_t(\mathrm{x}) = \begin{cases} 1010x\,, with\ probability\ 0.01 \\ -10x, \quad otherwise, \end{cases}$$

Similar to the aforementioned online setting, the optimal solution for this problem is x = −1. Again, we see that the iterate x t of ADAM converges to the highly suboptimal solution x = 1.

Finally, we consider the multiclass classification problem on the standard CIFAR-10 dataset, which consists of 60,000 labeled examples of 32 × 32 images. We use CIFARNET, a convolutional neural network (CNN) with several layers of convolution, pooling and non-linear units, for training a multiclass classifier for this problem. In particular, this architecture has 2convolutional layers with 64 channels and kernel size of 6 × 6 followed by 2 fully connected layers of size 384 and 192. The network uses 2 × 2 max pooling and layer response normalization between the convolutional layers. A dropout layer with keep probability of 0.5 is applied in between the fully connected layers [9]. The minibatch size is also set to 128 similar to previous experiments. The results for this problem are reported in Figure 3. The parameters for ADAM and AMSGRAD are selected in a way similar to the previous experiments. We can see that AMSGRAD performs considerably better than ADAM on train loss and accuracy. Furthermore, this performance gain also translates into good performance on test loss.

7 Conclusion

This article first explained the SWTHS strategy proposed by Nitish Shirish Keskar, and proposed a new optimization model ADAGRAD in combination with Adam's non-convergence. This new method essentially endows the algorithm with a long-term memory of past gradients. These fixes retain the good practical performance of the original algorithms, and in some cases actually show improvements.

Acknowledgment

Note: this paper is supported by the attachment project, project name such as attachment.

References

1. Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In Advances in Neural Information Processing Systems **25**, pp. 1097–1105, 2012.
2. Robbins, Herbert and Monro, Sutton. A stochastic approximation method. The annals of mathematical statistics,pp. 400–407, 1951.
3. Kingma, D. and Ba, J. Adam: A method for stochastic optimization. In International Conference on Learning Representations (ICLR 2015), 2015
4. Duchi, J., Hazan, E., and Singer, Y. Adaptive subgradient methods for online learning and stochastic optimization.The Journal of Machine Learning Research, 12:2121–2159, 2011.
5. Tieleman, T. and Hinton, G. Lecture 6.5-RMSProp: Divide the gradient by a running average of its recent magni-tude. COURSERA:Neural Networks for Machine Learning, **4**, 2012.
6. Nitish Shirish Keskar ,Richard Socher . Improving Generalization Performance by Switching from Adam to SGD. In International conference on machine learning, pp. 200–210, 2017.
7. Wilson, A. C., Roelofs, R., Stern, M., Srebro, N., and Recht, B. The Marginal Value of Adaptive Gradient Methods in Machine Learning. ArXiv e-prints, May 2017.
8. Sashank J. Reddi, Satyen Kale & Sanjiv Kumar. On the Convergence of Adam and beyond. Published as a conference paper at ICLR 2018, 2018.
9. Sutskever, I., Martens, J., Dahl, G., and Hinton, G. Onthe importance of initialization and momentum in deeplearning. In International conference on machine learning, pp. 1139–1147,2013.