Validating ADAM and AMSGRAD optimizers

Musluoglu Cem Ates, Novakovic Milica, Van Schreven Cyril
Department of Electrical Engineering, EPFL, Switzerland

***Abstract*—With the popularity of Machine and Deep Learning, the number of optimization algorithms keep increasing. However, we should question their reliability and limits before using them. Our aim in this paper is to present the comparison of the performance of some of these, in the context of a Deep Learning problem. For this, we used the MNIST dataset on popular architectures. More specifically we compared the initial ADAM optimizer with the AMSGRAD.**

I. Introduction

As studied in a recent publication [1], the exponential moving average based method Adam may in certain cases fail to converge. This issue is backed up by general Theorems that tells us that given certain conditions, there exists optimization problems for which Adam does not give the optimal solution. They propose a new algorithm similar to Adam but by correcting an error in the original Adam proposition [2]. In this paper, we aim to test this newly proposed algorithm, named AMSGrad and compare its performance with Adam and SGD and verify the results of [1]. For this we decided to use a known dataset and architecture, namely MNIST.

II. Models and Methods

We first start our analysis by comparing the algorithms. Let us consider our loss function $f_t : F \subseteq \mathbb{R}^d \to \mathbb{R}$ and Π_F the projection function of a point $x \in \mathbb{R}^d$. As pointed out in [1], a general algorithm for adaptive method is given in Algorithm 1.

Algorithm 1 Adaptive Algorithm

for $t = 1$ to T **do**
  $g_t = \nabla f_t(x_t)$
  $m_t = \phi_t(g_1, ..., g_t)$ and $V_t = \psi_t(g_1, ..., g_t)$
  $\hat{x}_{t+1} = x_t - \alpha_t m_t / \sqrt{V_t}$
  $x_{t+1} = \Pi_F(\hat{x}_{t+1})$
end for

In the previous algorithm, α_t is the learning rate. By specifying ϕ_t and ψ_t, we find different optimization algorithms. For example, SGD is given by

$$\phi_t(g_1, ..., g_t) = g_t$$

and

$$\psi_t(g_1, ..., g_t) = \mathbb{I}$$

On the other hand, Adam is given by

$$\phi_t(g_1, ..., g_t) = (1 - \beta_1) \sum_{i=1}^{t} \beta_1^{t-i} g_i$$

and

$$\psi_t(g_1, ..., g_t) = (1 - \beta_2) diag(\sum_{i=1}^{t} \beta_2^{t-i} g_i^2)$$

β_1 and β_2 are parameters that need to be chosen. They observed that the quantity $\Gamma_{t+1} = \frac{\sqrt{V_{t+1}}}{\alpha_{t+1}} - \frac{\sqrt{V_t}}{\alpha_t}$ is positive semi-definite for SGD but in the case of the exponential moving average methods like Adam, this variable can be indefinite. This leads to a problem because the decrease of the learning rate is not guaranteed. The algorithm AMSGrad proposed by [1] resolves this problem by keeping the maximum of v_t such that $V_t = diag(v_t)$ instead of changing it every time. This algorithm is given below.

Algorithm 2 AMSGrad

Set $m_0 = 0$, $v_0 = 0$, $\hat{v}_0 = 0$
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 $\hat{V}_t = diag(\hat{v}_t)$
  $x_{t+1} = \Pi_F(x_t - \alpha_t m_t / \sqrt{\hat{V}_t})$
end for

We can observe that β_{1t} varies with t but [1] tells that a constant value can also be accepted.

Understanding the problem of Adam and looking at a solution to it allows us to better understand both algorithms. We will now concentrate on verifying the results obtained in [1]. For this purpose, we choose the MNIST dataset and try the SGD, Adam and AMSGrad on three architectures; one which is convex (Just a linear layer), a network with a hidden layer and LeNet. We chose widely used architectures on the MNIST dataset.

III. Tests and Results

We present in this section our results. We consider the whole MNIST dataset, which consists of 60000 training and 10000 testing, 28×28 pixels images containing handwritten digits. We focused on accuracy and loss to compare the algorithms. Specifically, validation accuracy was used to select the parameters.

In our tests, the results are obtained with 40 epochs, the learning rate is 10^{-1} for SGD, 10^{-4} for Adam. For the latter, we did a grid search to find $\beta_1 = 0.91$ and $\beta_2 = 0.999$ and decided to use the same parameters for AMSGrad. As a first analysis we take a convex optimization problem; our network only consists of a linear layer which has an input size of 784

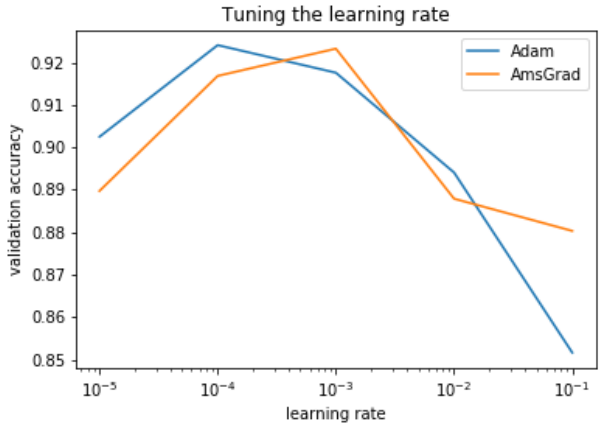

Fig. 1. Learning rate of Adam and AMSGrad optimizer for the convex model.

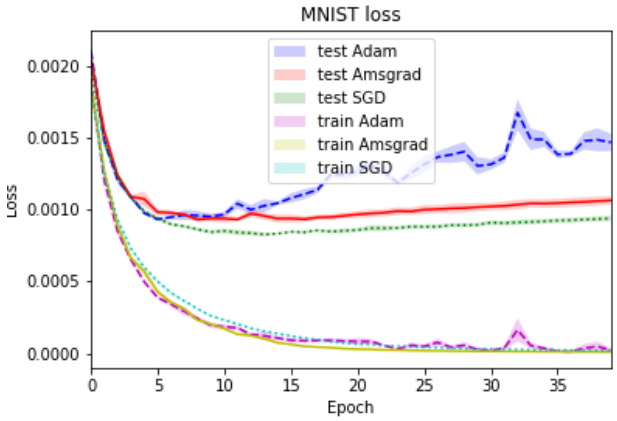

Fig. 3. Loss results for the validation with one hidden layer.

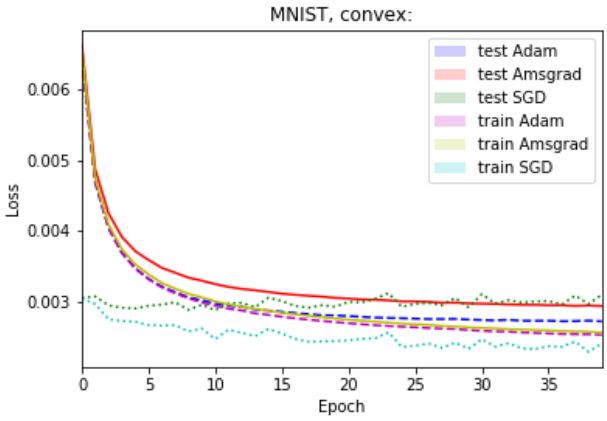

Fig. 2. Loss results for the validation with the convex model.

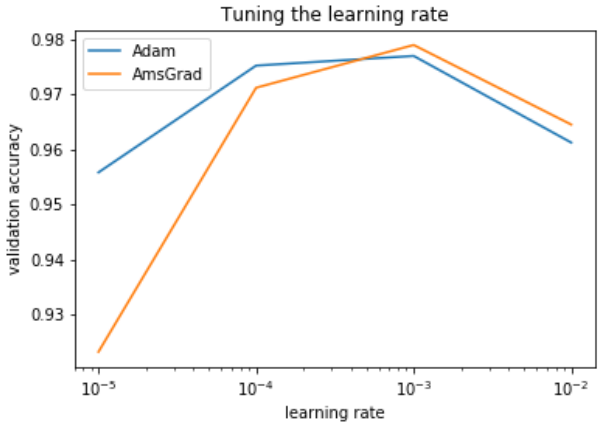

Fig. 4. Learning rate of Adam and AMSGrad optimizer for the single hidden layer network.

units and an output of 10 units. We use a softmax output activation and choose the maximum of the ten to classify the digits.

We are now interested in the architecture with one hidden layer of 100 units activated by a ReLU. We take again 40 epochs, a learning rate of 10^{-1} for SGD, and 10^{-3} for Adam. We also take the same parameters of Adam for AMSGrad. The loss values for the optimizers and for train and test validation can be seen in Fig. 3.

IV. Discussion

We could notice based on figures 1 and 4 that in convex model our optimal parameters changes from the default ones, while in non-convex settings our optimal learning rate is the same as the default ones and we our accuracy is more robust to changes of learning rate around the default ones.

On figure 3 we can see that as claimed in our reference paper, the test loss of ADAM goes up relatively fast, while the test loss of AMSGRAD stays similar to that of the SGD.

References

[1] S. K. Sashank J. Reddi and S. Kumar, "On the convergence of adam and beyond," *ICLR Conference*, 2018.

[2] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *Proceedings of 3rd International Conference on Learning Representations*, 2015.