Reproducing «On The Convergence of ADAM and Beyond»

Sashank J. Reddi
Google New York
New York, NY 10011, USA
sashank@gmail.com

Satyen Kale
Google New York
New York, NY 10011, USA
satyenkale@gmail.com

Sanjiv Kumar
Google New York
New York, NY 10011, USA
sanjiv@gmail.com

Contents

Submitted as course project for IFT6135 H2018 Representation Learning

1 Paper Summary

In this section we are going to provide a short summary on the issue that the paper is concerned with and the proposed solution provided to deal with that issue. Note that the paper has a rigorous mathematical approach in identifying the case it is studying and the variant or in other words the solution it is proposing. However we are omitting these mathematical analysis and results in our summary since our goal is to give the reader a rough understanding of the original paper. If the reader has more enthusiasm to see the mathematical analysis, we refer them to the original paper.

1.1 Motivation

A class of optimization methods commonly used in practice to train deep networks are based on using gradient updates scaled by the square roots of exponential moving averages of squared past gradients. A few popular methods of this class are:

- RMSPROP,
- ADAM
- ADADELTA
- NADAM

Although this class of optimization methods are very popular in practice, theoretically and empirically they could fail to converge to an optimal point or attain poor convergence. The key characteristic of this class of optimization methods is that they attain slow decay in the learning rate by using gradient updates scaled by the square roots of exponential moving averages of squared past gradients, which leads to the updates being more dependent on the most recent gradients, since the older ones fade away. Practically speaking only the few most recent gradients influence the update.

The mentioned key characteristic of these methods is to blame for their poor convergence, since there could be a scenario where some mini-batches result into large but less frequent gradients, and while these large gradients are very informative and useful for optimal convergence, their influence fades away quickly due to the exponential averaging, therefore leading to poor convergence.

Solving this issue and bringing optimal convergence and robustness to this class of optimization methods is the main motivation behind the idea of this paper. From here on, we can simply take ADAM as a representative of the class of optimization methods we are concerned with. Although the analysis provided in the paper, does extend to any method from this class.But for simplicity in delivering the summary we will only focus on ADAM algorithm. ADAM algorithm is given in Algorithm 1, where :

- $\mathcal{F}$ is the feasible set of points we are optimizing over
- f_t is the loss function
- The notation $\Pi_{\mathcal{F},\sqrt{A}}(y)$ means $\text{argmin}_{x\in\mathcal{F}}\|A^{\frac{1}{2}}(x-y)\|$

Algorithm 1 ADAM

```
procedure ON INPUT $x_1 \in \mathcal{F}$, STEP SIZE $\{\alpha_t > 0\}_{t=1}^{T}$
    Set $t = 1$, $m_0 = 0$, $v_0 = 0$
    while $t \leq 0$ do
        $g_t = \nabla f_t(x_t)$
        $m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t$
        $v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t$ and $V_t = \text{diag}(v_t)$
        $\hat{x}_{t+1} = x_t - \frac{\alpha_t m_t}{\sqrt{v_t}}$
        $x_{t+1} = \Pi_{\mathcal{F},\sqrt{V_t}}(\hat{x}_{t+1})$
        $t \leftarrow t + 1$
```

1.2 Proposed approach

In order to battle with the problem of poor convergence of the methods like ADAM which, as mentioned in the section 1.1 is due to the exponential moving average in the algorithm, the paper suggests that the algorithms should employ a "long-term memory" of gradients. For the case of ADAM we can see this variant (i.e. equipped with long-term memory of gradients) of the algorithm presented in Algorithm 2.

Algorithm 2 AMSGRAD (ADAM with long-term memory of gradients)

```
procedure ON INPUT $x_1 \in \mathcal{F}$, STEP SIZE $\{\alpha_t > 0\}_{t=1}^T, \{\beta_{1t}\}_{t=1}^T, \beta_2$
    Set $t = 1$, $m_0 = 0$, $v_0 = 0$, $\hat{v}_0 = 0$
    while $t \leq 0$ do
        $g_t = \nabla f_t(x_t)$
        $m_t = \beta_{1t} m_{t-1} + (1 - \beta_{1t}) g_t$
        $v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t$
        $\hat{v}_t = \max(\hat{v}_{t-1}, v_t)$        ▷ This is the key deference (i.e. the long-term memory)
        $\hat{V}_t = \text{diag}(\hat{v}_t)$
        $\hat{x}_{t+1} = x_t - \frac{\alpha_t m_t}{\sqrt{\hat{v}_t}}$
        $x_{t+1} = \Pi_{\mathcal{F}, \sqrt{\hat{V}_t}}(\hat{x}_{t+1})$
        $t \leftarrow t + 1$
```

2 Experiments

2.1 Motivation for the experiments

There are three main experiments presented in this paper. These experiments are on the following cases:

1. A synthetic case
2. Logistic Regression case
3. Neural Networks case

The proposed experiments are not concerned with complected architectures of neural networks but rather they are chosen to represent different classes of problems. The synthetic case is presented to underline the sub-optimal convergence of ADAM and optimal convergence of AMSGRAD in a simple convex setting. The motivation behind the logistic regression and the neural networks experiments is to underline the superiority of AMSGRAD to ADAM in the case of convex and non-convex setting respectively. These two experiments are performed on real-world data and the motivation here is to show that AMSGRAD does not only perform well in theoretical synthetic cases but it also performs well in real-world practical setting.

2.2 Reproducing The Main Results

In this section we will attempt to reproduce the main results of the paper. As mentioned in the previous section 2.1 we are aiming to reproduce the results of the paper in 3 experiments. We also performed an experiment on a variational auto encoder (VAE) to investigate the effect of AMSGRAD in a case study that is different for the ones mentioned in the paper. Note that some of the hyper-parameters we used for these experiments are specified by the paper, and the rest are found by grid search. A list of all of our hyper-parameters for this section can be found in the Table 1.

2.2.1 Synthetic Case

In this we are going to use our optimization methods to find the optimal solution in the following case:

$$f_t(x) = \begin{cases} 1010x & \text{with probability } 0.01 \\ -10x & \text{otherwise.} \end{cases} \tag{1}$$

With $\mathcal{F} = [-1, 1]$. We can see that in this simple case the optimal solution is $x = -1$ and hence we expect that our optimization methods converge to $x = -1$. however, as we can see in part (a) of the Figure 2, the ADAM method fails to converge to -1 and instead converges to the suboptimal solution of $x = 1$. As mentioned in

Table 1: Hyper-Parameters

Experiment	α_t	β_1	β_2	**Batch Size**	**Epochs**
Synthetic Case	$\frac{0.001}{\sqrt{t}}$	0.9	0.99	1	1M
Logistic Regression	$\frac{0.01}{\sqrt{t}}$	0.9	0.999	128	100
Neural Networks	$\frac{0.01}{\sqrt{t}}$	0.9	0.999	128	100
VAE	$\frac{0.002}{\sqrt{t}}$	0.5	0.999	128	100

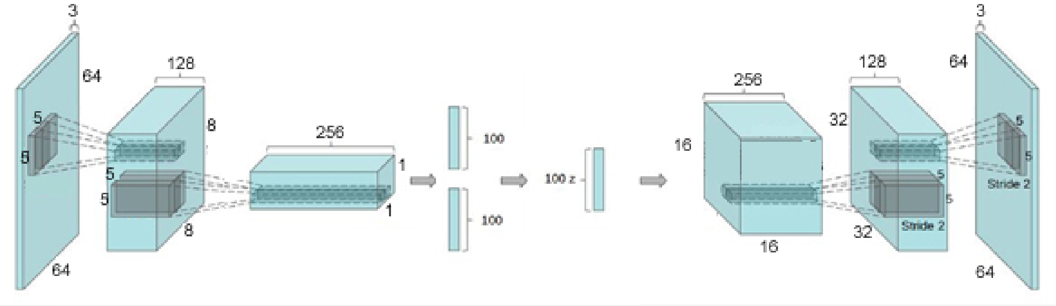

Figure 1: An abstract view of the VAE architecture used on celebA data set

section 1.1 this behavior is due to the exponential moving averages of squared past gradients that causes the rare but informative gradients to die out quickly. In equation 1 the gradient that is useful for optimal convergence will rarely appear (with probability of 0.01) and hence the ADAM method will not be able to effectively use this information. However, AMSGRAD does not suffer from the same problem as it was constructed to, hence reaching the optimal solution.

2.2.2 Logistic Regression

In this experiment we will attempt to produce the result of the logistic regression experiment shown in the paper on MNIST data set. The architecture we used to reproduce the result is the same as the ones presented in the paper. The comparison of loss of the 2 methods shown in part (b) of Figure 2 is compatible with the ones found in the paper, and we can see the better convergence of AMSGRAD in comparison to the ADAM method.

2.2.3 Neural Networks

The simple neural network we used here is based on the structure that was dictated by the paper and we used MNIST data set to investigate the performance of AMSGRAD and ADAM on the loss function. As shown in part (c) of Figure 2, AMSGRAD performs exceptionally better than ADAM. This result is compatible with the ones presented in the paper.

2.2.4 VAE

This experiment is not given by the paper, however we decided to examine the performance of AMSGRAD in an experiment that is not dictated by the paper. An abstract view to the architecture of the VAE we used for this study can be found in the Figure 1. The result of our experiment on the model loss can be seen in part (d) of Figure 2. This experiment indicates that AMSGRAD attains a lower loss in comparison to ADAM. Note that the plot is only over 100 epochs, and although we clearly see that AMSGRAD is doing better than ADAm,it seems like they are both approaching to the same value. However, this is not a contradiction as the paper mention and proofs that AMSGRAD will do better or equal to ADAM in general, hence we can expect that they both converge to the same value in some cases.

3 Discussion

The papers goal was to study and show the poor convergence of exponential moving average methods such as ADAM. They demonstrate a source of problem that could be responsible for the poor convergence and provide a method to fix it. What we can see in the experiments, reflects exactly that. Hence the experiments are aligned and follow the goals of the paper.

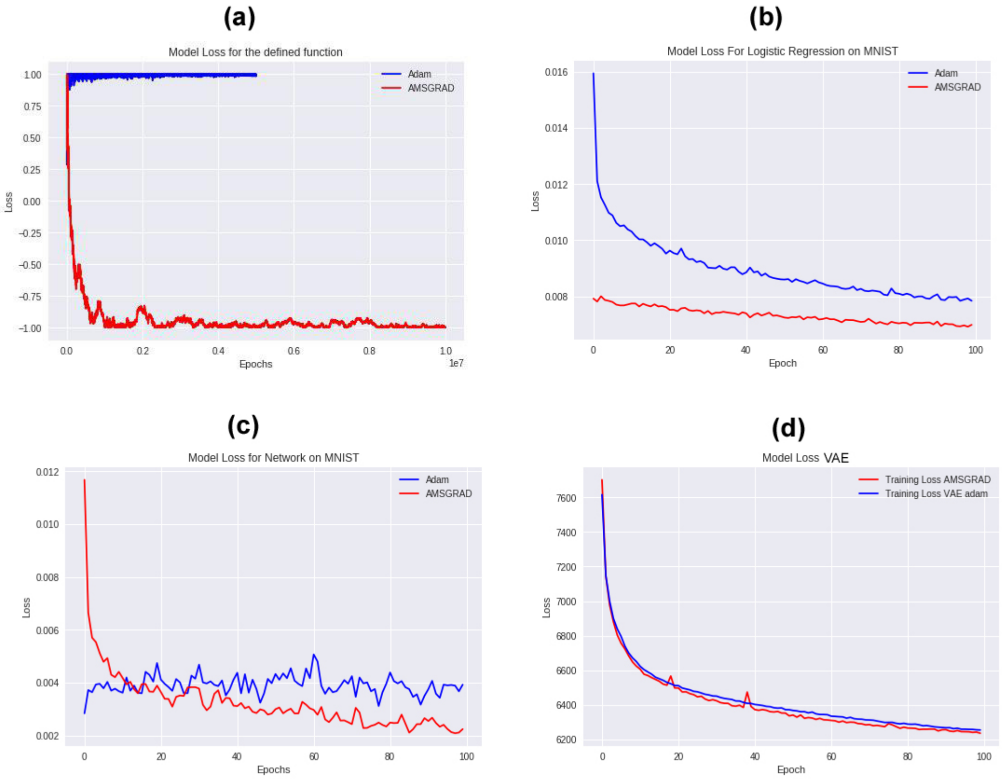

Figure 2: (a): Loss comparison between ADAM and AMSGRAD for the synthetic case (b): Loss comparison between ADAM and AMSGRAD for logistic regression on MNIST (c): Loss comparison between ADAM and AMSGRAD for neural network on MNIST (d): Loss comparison between ADAM and AMSGRAD for VAE on celebA data set

The paper has a mathematical approach to its goal. It provides thorough analysis of the current class of exponential moving average optimization methods; it finds a flaw and provides a solution to fix that flaw. However, this does not mean that there is no other possible sources responsible for or contributing to the poor convergence. The variant of the of the algorithm introduced by the paper, might still suffer from the poor convergence problem in some cases, however the paper does not provide any insight in this matter. Perhaps future studies could be dedicated to uncover if there are any other sources that could possible contribute to the poor convergence of the new variant introduced by the paper, or a proof for the nonexistence of such sources.
The experiments proposed by the paper, although covering both convex and non-convex settings, are performed on simple networks. Although the significance of using AMSGRAD is quite obvious in the proposed experiment, we wonder how significant would the use of AMSGRAD be on the state of the art architectures with deep models. In our independent experiment with a VAE, we found that both methods converge to the same value, although the paper has mentioned that AMSGRAD will perform equally as good or better than ADAM, we wonder in practice how often is that case.