NYU CS-GY 6763: Algorithmic Machine Learning and Data Science Spring 2022

On the Convergence of Adam and Beyond...and Beyond!

Names: Richard Fu, Roman Negri, Timothy Merino

Abstract

In this paper we aim to summarize findings about the convergence of ADAM in contrast to other gradient descent algorithms and extend that research by examining specific unexplored forms of data which introduce convergence differences. We detail two specific cases in which the data is formulated to challenges the flaws of ADAM and compare the results to those of AMSGRAD on the same data. Our results suggest that in specific cases AMSGRAD may be a needed alternative in order to efficiently converge to an optimal solution. We then explore the real life consequences of these findings and the extension to scenarios involving life long learning and changing datasets.

1 Introduction

The goal of stochastic optimization in recent machine learning applications is to learn or predict some latent variable within a dataset in which the probability distribution is unknown. In the convex setting, the vanilla gradient descent method provides the most straightforward way to find the global minima; by taking the gradient at every point to iteratively change the search variable until it converges. However, vanilla gradient descent is impractical in the case of large datasets where it becomes computationally taxing, as well as in non-convex situations, where it may converge to a local instead of global minima. To combat this, stochastic gradient descent randomly selects one data sample to calculate the gradient on, and adjusts the variable based on the random gradient. In this case, since the gradient is allowed to 'jump around' the space, we are more likely to capture the global minima in the non-convex case while requiring less computation than the normal gradient descent. The general form of an SGD update is as follows,

$$\hat{x}_{t+1} = \hat{x}_t - \eta g_t$$

where η is the tunable learning rate and g is the gradient at time t.

However, vanilla SGD has problems with regard to tuning the learning rate parameter. In vanilla gradient descent, setting a low learning rate guarantees a convergence to a minima, even if it is time consuming. SGD, because of its random nature, does not necessarily guarantee this. This problem has led to many variants of SGD such as ADAGRAD, RMSPROP, and ADAM [1] that tune the learning rate based on the first and second moments of the historical gradients. The generic form of a SGD variant is as follows,

$$\hat{x}_{t+1} = \hat{x}_t - \frac{\alpha_t m_t}{\sqrt{v_t}}$$

where

$$m_t = \phi(\{g_1, ..., g_t\})$$

$$v_t = \psi(\{g_1^2, ..., g_t^2\})$$

The single gradient term from normal SGD now depends on all the previous gradients, and the learning rate η is replaced with a function of all the squared elements of the historical gradients.

In this paper we aim to replicate and extend the findings related to ADAM non-convergence demonstrated by Reddi et al.[2]. First, we will summarize at a high level the findings of Reddi et al., and how they relate to our experiments. Then, we replicate the counter-examples presented in Reddi et al., comparing the results of ADAM and the improved optimizer they introduce, AMSGRAD. Finally, we aim to expand on their findings by generating a sample dataset that can induce the non-convergent behavior shown in their example. We demonstrate the non-convergence of ADAM on this dataset, and compare this to the convergence of AMSGRAD

In our research, we focus primarily on the ADAGRAD and ADAM optimizer m and v functions in this report, as the original paper, "On the Convergence of ADAM and Beyond"[2] primarily focuses on the non-convergence of the ADAM optimizer under specific conditions. In our experiments, we focus on the one dimensional case for the first and second moment functions, similar to the nonconvergence examples presented in the paper.

2 On the Convergence of Adam and Beyond - Reddi et al.

2.1 Regret Bounds

Regret is often taken as the metric to determine whether an optimization model converges. The regret function provided in this paper is the static regret in online learning, given by

$$R_T = \sum_{i}^{T} f_i(x_i) - min_x \sum_{i}^{T} f_i(x)$$

This measures difference between the cumulative loss of a parameter x_i and the cumulative loss of the optimal solution x in hindsight, over T iterations. Note that it assumes that there is some fairly optimal solution x over all iterations, and that if at some point the optimizer converges, R_T stops growing and $\lim_{T\to\infty} \frac{R_T}{T} = 0$. In their paper, Reddi et al. prove that there exists a case in both online optimization and stochastic optimization in which ADAM fails this requirement and has nonzero average regret [2].

Another way of viewing this, also presented in the paper, is that for an optimizer to converge, its regret must be $o(T)$, that is, its regret bound must be a function strictly smaller than T.

2.2 Adagrad and Adam

The following equations for SGD, ADAGRAD and ADAM, are detailed in the paper directly, but in this report, for ease of comprehension, they are presented in the 1 dimensional case. For example matrix V, typically a diagonal matrix generated by the second moment gradients, will be represented by a constant v since there is only one gradient. Additionally, α is defined to be the learning rate.

As a baseline, the m and v functions for SGD are,

$$m = g_t$$
$$v = 1$$

In ADAGRAD, the learning rate is directly affected by all the historical gradients. The m and v functions are,

$$m = g_t$$
$$v = \frac{\sum_i^T g_t^2}{T}$$

As a result from gradient aggregation, ADAGRAD tends to have the problem of diminishing learning rate, leading to slow convergences.

ADAM rectifies this by introducing two new parameters β_1, β_2, which provide an exponential decay to the older gradients. This is known as the 'exponential moving average' model,

$$m = (1 - \beta_1) \sum_{i=1}^{T} \beta_1^{t-i} g_i$$
$$v = (1 - \beta_2) \sum_{i=1}^{T} \beta_2^{t-i} g_i^2$$

Alternatively, this can be written recursively as,

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t$$
$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2$$

2.3 Non-convergence of Adam

Reddi et al. discuss a flaw in exponential moving average optimizers, especially with regard to learning rates [2]. They identify the following quantity of interest,

$$\Gamma_{t+1} = \frac{\sqrt{V_{t+1}}}{\alpha_{t+1}} - \frac{\sqrt{V_t}}{\alpha_t}$$

This is the rate of change of the inverse of the learning rate, measured between two adjacent time steps. They claim that divergence issues in ADAM and RMSPROP are caused when Γ_t is not positive definite for some $t \in T$. In SGD and ADAGRAD, this value is always positive definite since we assume α remains constant, and v is non-decreasing with time. Thus, learning rates for SGD and ADAGRAD always remain non-increasing. However, for ADAM and other exponential moving average optimizers, the historical gradients are modified, meaning v is no longer non-decreasing for all t. The paper concludes that $\Gamma_t \not\geq 0$ can lead to non-convergence issues.

The significance of this quantity shown by Reddi et al.[2] has lead to new ADAM variants that force $\Gamma_t \succ 0$. However, Zou et al. note that that there is yet no satisfactory explanation for the core reason of the divergence of ADAM and RMSPROP [3].

To demonstrate this point, Reddi et al. present a situation where ADAM does not converge as a result of this exponential decay of the gradients affecting the learning rate[2]. They consider a loss function f_t as follows,

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

where $C \geq 2$. In this case, the optimal regret occurs when $x = -1$. However, for certain choices of ADAM hyper parameters $\beta_1 = 0, \beta_2 = 1/(1 + C^2)$, ADAM converges to $x = 1$. This is due to the fact that the large gradient C occurs infrequently, and is scaled down by β_2. Having such a large gradient should decrease the overall learning rate and push x in the positive direction, however, the effect of all the other smaller gradients overriding the decayed large gradient aggressively increases the learning rate in the opposite direction.

2.4 Introduction of AMSGrad

As a solution for this non-convergence problem the paper introduces a new alternative to ADAM, AMSGRAD [2]. The main change behind it is that since non-convergence happens on the basis that $\Gamma_t \not\succeq 0$, forcing the positive semi-definiteness of this quantity by introducing a non-decreasing v solves any non-convergence issues.

The improved first and second movement gradients follow the equations:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t$$
$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2$$
$$\hat{v}_t = \max(\hat{v}_{t-1}, v_t)$$

Where $\hat{v}_t$ is then used in the gradient descent step:

$$\hat{x}_{t+1} = \hat{x}_t - \frac{\alpha_t m_t}{\sqrt{\hat{v}_t}}$$

Although this means a marginally slower convergence time than ADAM, this guarantees $\Gamma_t \succeq 0$ for all $t \in [T]$ without having to change β_2, and avoids the possible negative consequences of using methods such as ADAM and RMSPROP allowing for a regret bound of $O(\sqrt{T})$.

We illustrate the example of nonconvergence as well as the effect of AMSGRAD through implementation of both optimizers in Python 3.7.13. First we generate sample gradients directly from the example distribution similar to the one in 2.3, with identical choices for β_1, β_2. In the gradient vector, every C^{th} element starting with the second element, has a gradient of C, while all others have a gradient of -1. We then implement ADAM and AMSGRAD update step as written in their respective papers.

The results are as follows,

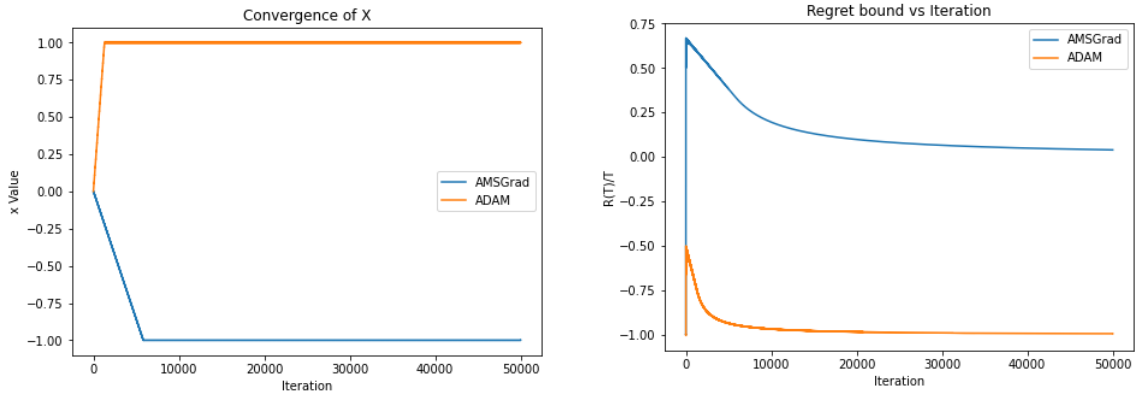

Figure 1: Comparison of AMSGrad and Adam convergence and average regret

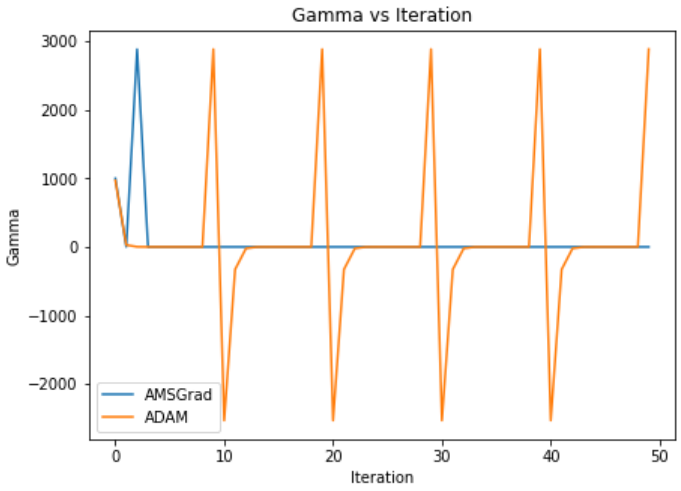

Figure 2: Gamma function for both Adam and AMSGrad

Figure 1 illustrates the convergence and regret bound of Adam and AMSGrad. As the paper suggests, even for a small value $C = 4$, Adam fails to converge to the correct value and its average regret is nonzero. AMSGrad converges to the correct value, although slightly slower than Adam, and its average regret trends to zero as expected.

Figure 2 illustrates how AMSGrad restricts $\Gamma \succeq 0$ by enforcing that the update v is non decreasing. This function is periodic because every C^{th} value has a gradient of C. We can see that in the case of Adam, every time there is a high positive gradient of C, Γ spikes, but over time it decreases due to the effect of the small negative gradients until Γ becomes negative and the learning rate increases for the small negative gradients. In the case of AMSGrad, since v is restricted to v_{max}, once the first C value gradient spikes, Γ never drops below zero and is constant, since v does not change. Note that this is the case because there are no other gradients in this system other than C and -1. With gradients greater than C, we would expect more spikes in the graph for when the learning rate decreases as a result of the sudden gradient.

3 Research Extensions

We intend to extend the toy example to an application in practice by analyzing two situations where the gradients could vary as significantly as they do in the example. Where Reddi et al. [2] emphasizes the loss function, we aim to generate a dataset mimicking real world data that forces the loss function to behave as in the paper. In both situations we introduce ADAM and AMSGRAD to non i.i.d data, the first one which involves non i.i.d noise, and the second which is a dataset that draws from different distributions. An example where the first situation may occur is in communications settings where the noise from data may not have consistent distribution due to interference. The second situation (which is related to the probabilistic example from the paper) may have applications in lifelong learning, where a dataset might change drastically while training. It's important to note that whilst doing research, we found that many of the applications that cause ADAM to fail involve the learning rate, and typically using non-i.i.d data affects learning rate for exponential moving average estimators.

3.1 Variable Noise Datasets

First we consider a situation where the residual in a linear regression problem is not identically distributed. This could lead to large, uneven gradients similar to the example given in the paper, as there may be some error distributions that are skewed or do not center around the mean. For this, we mix a dataset consisting of noise vectors from the gamma, exponential, uniform, χ^2, and normal distributions (Figure 3). We attempt to track the gradient of the mean squared error loss function $f = \frac{1}{n}\sum_i^N (Y - \hat{Y})^2$ and attempt to see if ADAM converges to the correct slope.

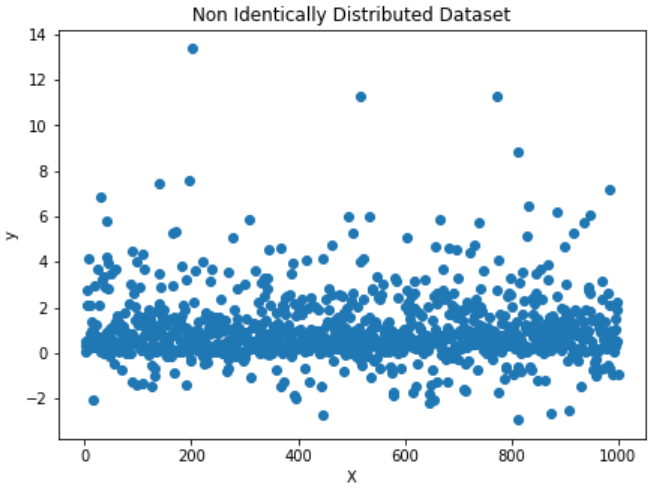

Figure 3: Mixed Noise dataset

Given this dataset, we would expect a convergence to a slope of zero, with the low probability high value datapoints skewing the gradient enough to force non-convergence in the ADAM case.

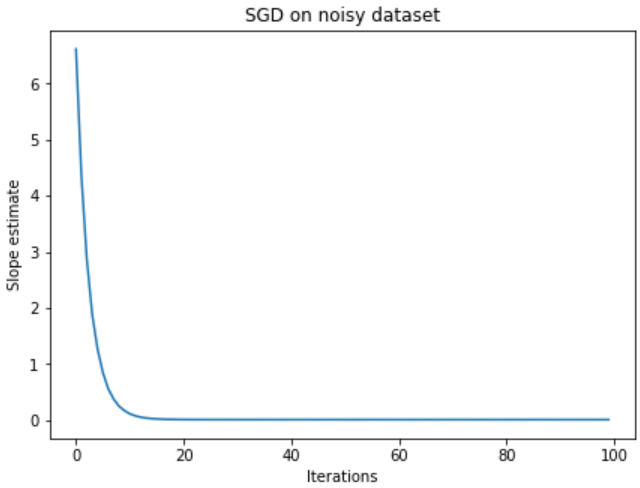

Figure 4: SGD convergence for the noisy data

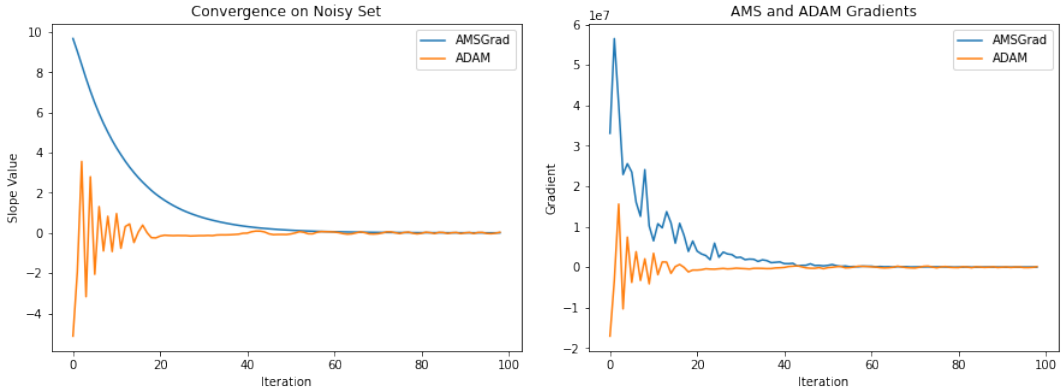

Figure 5: AMSGrad and Adam convergence for the noisy data

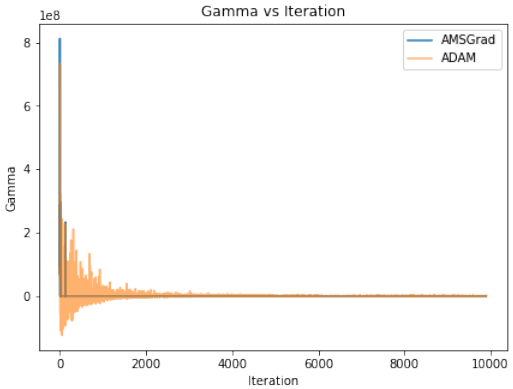

Figure 6: Adam and AMSGrad Gamma values for noisy data

Both Adam and AMSGrad use $\beta_1 = 0.9, \beta_2 = 0.99$, which is standard in practice. Adam and AMSGrad both converge to the proper slope, however, AMSGrad has a significantly smoother

descent - indicative of the non-increasing learning rate. Additionally, in Figure 5, we notice that despite converging to the correct value, the SGD estimate converged far quicker (in roughly 10 epochs), even with a learning rate of 10^{-8}, than the ADAM or AMSGRAD optimizers. We attribute this to the fact that the learning rate does not depend on the gradients, so the algorithm steadily moves towards the optimum.

We also notice that although Γ_t takes negative values for ADAM, ADAM successfully converges to the optimal value of 0. Though Reddi et al. identity this quantity as the main issue leading to non-convergence of ADAM, they only note that this violation of positive semi-definiteness can lead to undesirable convergence behavior[2]. This example demonstrates that negative values of Γ_t do not guarantee non-convergence for ADAM.

3.2 Inconsistent Distribution - Sampling from multiple distributions

Second, we look into a situation where we may sample from one dataset, but with data points introduced from a different population at a very low probability. This concept is introduced in the paper, but we will show a dataset that produces similar gradients as the ones described in the paper. Given this dataset, we would expect a proper linear regression to account for the outliers.

We begin by creating data points with two distributions, one being much rarer than the other. Specifically, generating a set of 1000 points along the line $y = x$ with 20 of those chosen randomly being instead given a slope of $y = -20x$. This creates a realistic dataset where a rare positive gradient changes the direction of the optimization similar to the toy example detailed in the paper.

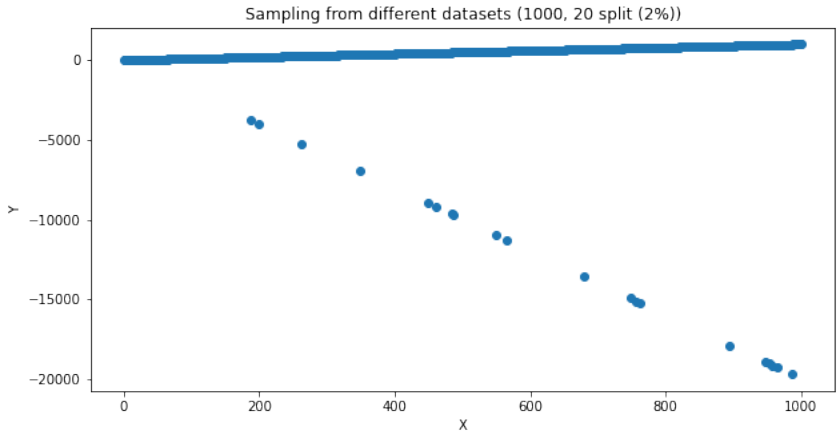

Figure 7: Differing Datasets

Figure 6 illustrates the dataset of size 1000 where most of the points lie with a $y = x$ slope above zero, but 20 samples are drawn that have a negative slope and are distant from the original dataset. The optimal slope for a MSE line was found using the closed-form solution for linear regression, with the np.linalg.lstsq library function. In this scenario, the optimal slope was 0.41.

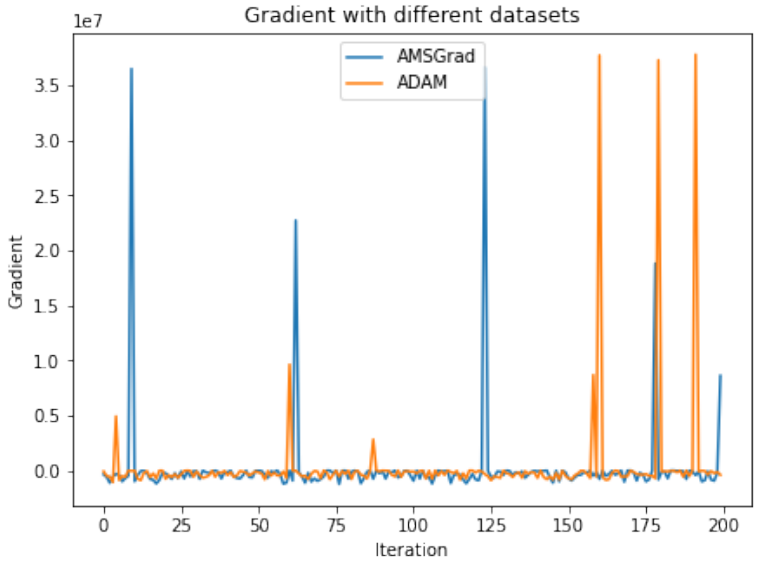

Figure 8: Gradients - Combined Dataset

Figure 7 serves to illustrate the fact that the gradients look very similar to the example given in the paper. We have a baseline of many fairly small negative gradients, which occasionally spike when one of the points in the smaller dataset is processed.

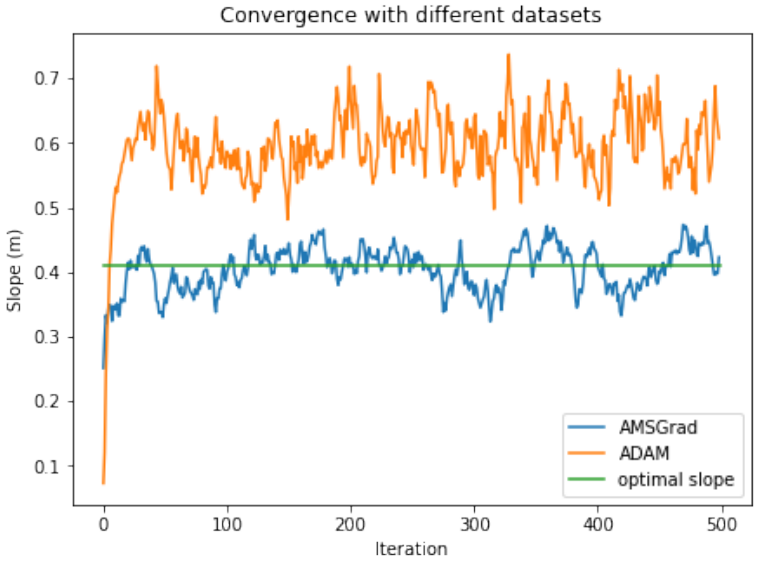

Figure 9: Convergence for ADAM and AMSGRAD - Combined Dataset

After running each optimizer we see in Figure 8 that the ADAM optimizer favors converging closer to slope $m = 1$ specifically at $y = 0.61x$, caused by the more common $y = x$ distribution, whereas AMSGRAD converges to a lower value at $y = 0.42x$, much closer to the true optimal line $y_{\text{optimal}} = 0.41x$. This can be attributed to the gradient decay mentioned in the paper that causes ADAM to converge to the wrong value. Realistically, it is important see some change in the regression as a result of outliers.

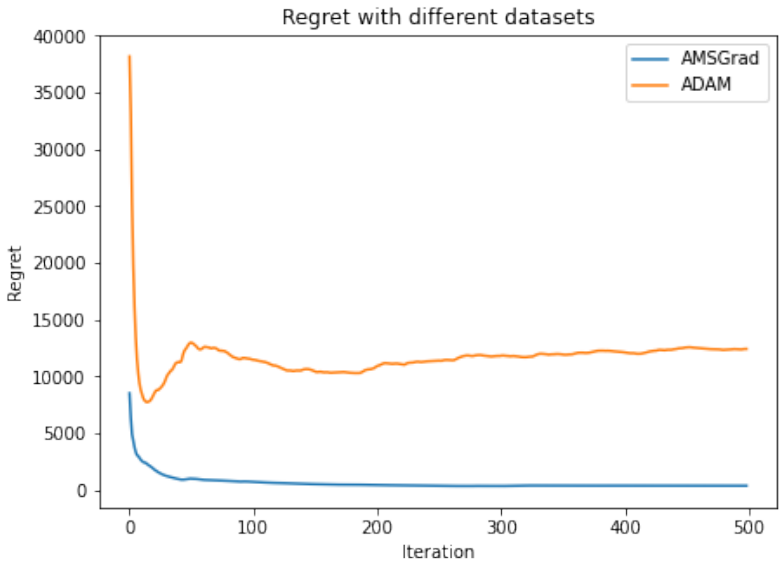

Figure 10: Average Regret for ADAM and AMSGRAD - Combined Dataset

Finally we can use the regret bounds as a method to display the non-converge of ADAM. In Figure 9, while the average regret for the AMSGRAD optimizer approaches 0, the ADAM optimizer not only does not trend towards 0 but is very far off above 0 average regret.

4 Conclusion

Reddi et al. proposes a solution to a theoretical flaw of the ADAM optimizer - AMSGRAD, an algorithm which restricts the value of v and the learning rate so as to account for large historical gradients [2]. Despite having good experimental results, it is unclear whether the AMSGRAD proposed has any notable benefit over ADAM in non contrived cases. In this report, we set out and found a situation in practice where ADAM fails due to historical gradient decay, and which AMSGRAD rectifies, by generating a problematic dataset based on the findings of Reddi et al. [2].

We demonstrate the non-convergence of ADAM on a dataset generated from two distributions. Online learning scenarios where there are multiple distributions are known to cause problems with learning rates, and are common in lifelong learning or federated learning scenarios. Understanding the drawbacks of ADAM in this scope can lead to better optimization and lead future works to uncover distinct tests for these non-convergence cases.

References

[1] Diederik P. Kingma and Jimmy Ba. *Adam: A Method for Stochastic Optimization*. 2014. DOI: 10.48550/ARXIV.1412.6980. URL: https://arxiv.org/abs/1412.6980.

[2] Sashank J. Reddi, Satyen Kale, and Sanjiv Kumar. "On the Convergence of Adam and Beyond". In: *International Conference on Learning Representations*. 2018. URL: https://openreview.net/forum?id=ryQu7f-RZ.

[3] Fangyu Zou et al. "A Sufficient Condition for Convergences of Adam and RMSProp". In: *CoRR* abs/1811.09358 (2018). arXiv: 1811.09358. URL: http://arxiv.org/abs/1811.09358.