The Unreasonable Effectiveness of Adam on Cycles

Ian Gemp∗
DeepMind
London, UK
imgemp@google.com

Brian McWilliams∗
DeepMind
London, UK
bmwc@google.com

Abstract

Generative adversarial networks (GANs) are state of the art generative models for images and other domains. Training GANs is difficult, although not nearly as difficult as expected given theoretical results on finding a Nash (PPAD complete) and our understanding of dynamical systems. Several new algorithms and techniques have been proposed to stabilize GAN training, but nearly all employ Adam or RMSProp. In fact, training a GAN with SGD instead of Adam often fails. Here, we aim to understand how Adam circumvents some of the difficulties associated with GAN training. To this end, we study Adam in the context of a cycle problem. The cycle problem is a canonical equilibrium problem for which naive optimization approaches, e.g., simultaneous SGD, fail. Understanding how Adam works in this context helps reveal reasons for its unexpected success.

1 Introduction

In their seminal work on generative adversarial networks (GANs), Goodfellow et al. [8] proposed a modified minimax objective which was optimized using SGD with momentum. Since this original work, GANs have achieved state of the art performance on a variety of generative modeling tasks, most notably in high-resolution image generation [5, 10]. Notably, nearly every major new GAN model and training algorithm has employed either Adam [12] or RMSProp [20] (see Table 1). For reference, the TensorFlow implementation of Adam proceeds (coordinate-wise) as follows:

\[
\begin{align*}
m_t &= \beta_1 m_{t-1} + (1 - \beta_1) g_t \\
v_t &= \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \\
w_t &= w_t - \alpha \frac{m_t}{\sqrt{v_t} + \epsilon}
\end{align*}
\]

At iteration \(t\), \(g_t\) is the (stochastic) gradient, \(m_t\) is the exponentially averaged gradient, \(v_t\) is the exponentially averaged squared gradient, \(w_t\) are the model parameters and \(\epsilon\) is a small constant (e.g., \(10^{-8}\)). \(\beta_1\) and \(\beta_2\) are hyper-parameters which control the rate of forgetting in the exponentially weighted averages. The ubiquitous default hyper-parameter choices of \(\beta_1 = 0.9\) and \(\beta_2 = 0.999\) have proven themselves empirically on a wide range of supervised learning problems. One possible explanation of the success of Adam in this setting is its tendency to better explore non-smooth optimization landscapes [3].

However, of the GAN works that employ Adam, most choose \(\beta_1 \in \{0, 0.5\}\). No rationale is given for the choice of these values besides the fact that they provided the best performance over a hyperparameter sweep. This is surprising given the suggested default value works so well for deep learning in classification and regression settings. In fact, \(\beta_1 = 0\) represents zero gradient averaging, an extreme that one would expect would negatively impact minibatch training.

* denotes equal contribution.

Adam (β1=0.0)  |  Progressive Growing [10], BigGAN [5], StyleGAN [11], Wasserstein GAN [2]
Adam (β1=0.5)  |  DCGAN [18], Improved Techniques [19], Conditional GAN [16], ExtraAdam [14], CycleGAN [22], Pix2Pix [9], StackGAN [21], UnrolledGAN [15]
RMSProp  |  Numerics of GANs [14], SGA [4], Crossing-the-Curl [6]

Table 1: Adam (β1=0.0 or 0.5) and RMSProp are the algorithms of choice for training GANs. Why?

Why then do such low β1 values work well for training GANs even though these values are rarely shown to be performant on supervised deep learning problems? One major difference is that a GAN is a two player (minimax) game while classification and regression represent one player games—the learning dynamics of games present issues not dealt with in classical optimization settings.

The cycle problem, \( \min_x \max_y \{ V(x, y) = xy \} \), has been proposed as a canonical equilibrium problem for gaining a better understanding of GANs [4, 6]. Cycles cannot appear in the continuous-time gradient descent dynamics of deterministic (full batch training) optimization problems. However, they can (and do [6]) appear in GANs. Our hypothesis is that Adam with low β1 is somehow better poised to cope with the cycle problem and this leads to its better performance on GANs.

In the next section, we explore reasons for how Adam might reach the equilibrium of the cycle problem where gradient descent always fails. We then perform experiments on the cycle problem to tease out the relationships between β1, β2, batch size, convergence rate, and limit sets. Finally, we show empirically that the relationships hold beyond the simple cycle problem setting and extend to GANs trained on real data.

2 The Cycle Problem

![Figure 1: Behaviour of Adam when used to train two models whose learning trajectories trace a cycle. The red arc denotes the approximate effective window over which Adam averages gradients and squared-gradients. The red vector denotes the coupled gradient direction of both players ten iterations prior: \( g_{t-10} = [\nabla_x V|_{t-10}, -\nabla_y V|_{t-10}] \). The blue vector denotes the current coupled gradient direction. The green arrow illustrates the effect of Adam run with β1 = 0 and a β2 value that induces the effective window size matching the red arc. Overbars denote averages over the red arc. The equilibrium is at the origin.](image)

Adam’s effect on the cycle problem can be decomposed into three separate processes. First, a nonzero value for β1 makes Adam average over historical gradients (producing the *1st moment vector*, eq. 1). In the cycle setting, historical gradients direct learning outside the cycle leading to divergence (see \( g_{t-10} \) in Figure 1a), therefore, setting β1 = 0 in this setting is reasonable. Moreover, this is supported by variational inequality theory in which the algorithm used for solving equilibrium problems is extragradient [13]: extragradient performs updates using “future gradients”, \( g_{t+1} \), which will take learning inside the cycle towards the equilibrium. In other words, historical gradients are in direct opposition to theory in this setting.

Second, the cycle problem as presented, assumes deterministic updates. GANs are trained with minibatches, and therefore, must cope with noise in the gradients. Furthermore, GAN dynamics are not purely cyclical. **Nonzero β1 values help to smooth out gradients** and accelerate convergence. So although β1 = 0 seems ideal for the cycle, there is a tradeoff that must be considered depending on minibatch size.

1Authors mention Adam with β1 > 0 is unstable.
Lastly, Adam averages over historical squared gradients as well (producing the 2nd moment vector, eq. 2). As illustrated in Figure 1a, if $\beta_2$ is chosen well, Adam averages squared-gradients over the effective window highlighted by the red arc. The squared gradients over this arc are small in the $x$-direction and large in the $y$-direction. Therefore, when Adam readjusts gradients by dividing by the root of the squared gradient, $\nabla_x V|_t$ gets amplified and $\nabla_y V|_t$ gets attenuated. The effect is the original gradient $g_t$ in blue is transformed to the one in green, now pointing inside the cycle.

The effect of $\beta_2$ is not consistent across all arcs of the cycle despite symmetry of the trajectory. In Figure 1b, the historical gradients are an equal mix of small and large $\nabla_x$ and $\nabla_y$. This results in a 2nd moment vector that is equal across dimensions, and so dividing by the 2nd moment does not change the update direction. Therefore, Adam will appear to diverge at some points along the cycle (similar to simultaneous gradient descent).

To summarize these points, an optimal $\beta_2$ specifies a constant historical window in terms of angle, not arc length for the cycle problem. Alternatively, as Adam approaches the equilibrium, the effective historical window size over which the squared gradients are averaged must shrink to reflect the shrinking radius of the cycle. Therefore, a smaller $\beta_2$ may be ideal at the end of training.

These statements motivate the following GAN hypotheses tested empirically in the next section:

1. The existence of cycles suggests $\beta_1$ less than the default 0.9 will help avoid divergence.
2. The existence of noise and non-cyclical dynamics suggests $\beta_1 > 0$ to filter gradient noise.
3. The existence of cycles suggests $\beta_2$ less than the default 0.999 may help lead iterates closer to an equilibrium at the end of training.

### 3 Cycle Experiments

We first study the effect of $\beta_1$ on convergence to the equilibrium of the cycle. We fix $\beta_2$ to its default value 0.999 (also the value commonly used in the GAN literature) and run Adam for 100 thousand iterations. Figure 2 shows Adam’s trajectory as $\beta_1$ is increased from 0 to 0.9. We first observe that Adam does not converge to the equilibrium at the origin. Instead, it appears to converge to a limit cycle. Let $D(x_{\beta_1}, x^\ast)$ denote the distance of the final iterate to the origin for Adam run with a given value of $\beta_1$. We approximate the radius of the limit cycle with this value. The convergence of Adam to a limit cycle corroborates the preceding discussion regarding the effective window size controlled by $\beta_2$. As the radius of the limit cycle shrinks, so must the window to average over the optimal historical gradients. Secondly, the radius of this limit cycle grows as $\beta_1$ grows. This corroborates the first process explained in the preceding section. A nonzero $\beta_1$ uses historical gradients which contributes to divergence. In the right plot, we see that introducing noise appears to decrease the sensitivity of convergence to choice of $\beta_1$, however, a lower $\beta_1$ is still optimal. These findings agree with current literature: BigGAN trains with $\beta_1 = 0$ and a minibatch size of 2048 ($\gg$ standard 64).

Figure 3 examines the effect of $\beta_2$ on convergence to the equilibrium of the cycle. We fix $\beta_1 = 0$ because it was optimal in the previous experiment. In the extreme case where $\beta_2 = 0$, Adam divides each element of the gradient vector by the square root of the same element squared (plus $\epsilon$). In other
words, \( g_t^{Adam} \approx g_t / |g_t| = \text{sign}(g_t) \). This explains the piecewise linear trajectory of Adam in the first plot. As \( \beta_2 \) is increased, distance to the equilibrium decreases until \( \beta_2 \approx 0.642 \), at which point, a sharp increase in distance is seen. The plot with \( \beta_2 = 0.999 \) shows a limit cycle has formed.

### 4 CIFAR-10 Experiments

We ran experiments with DCGAN [18] trained using Goodfellow’s modified loss [8] on CIFAR-10 to see how the relationships uncovered above translate beyond the cycle problem to neural-network based GANs. Figure 4 reveals, for a batch size of 128, \( \beta_1 \in [0.5, 0.7] \) achieves lower Fréchet inception distance (FID) score than \( \beta_1 = \{0.4, 0.8\} \). For a smaller batch size of 64, gradients are noisier, and so \( \beta_1 = 0.8 \) joins the group of top performers. This agrees with results above where additional noise suggests larger optimal \( \beta_1 \) is possible. Small values of \( \beta_1 \), e.g., 0.1, performed poorly for both batch sizes, although marginally better for the larger batches. We also experimented with varying \( \beta_2 \) according to a cosine decay but found this did not appreciably affect results.

### 5 Conclusion

Adam’s success in training GANs is fortuitous. By examining Adam on a cycle problem, we illustrate relationships between convergence rate, limit sets, \( \beta_1, \beta_2 \), and batch size. Our initial investigations have shown that larger batch sizes can allow for lower \( \beta_1 \) values. In future work, we aim to explore the effects of \( \beta_1 \) and \( \beta_2 \) when much larger batch sizes are used, for example in BigGAN. Also important is the fact that studying such a simple equilibrium problem is able to provide insights to the performance of Adam on GANs.

In future work, we will explore Adam’s theoretical convergence on bilinear saddle point problems. We expect insights gleaned there to help us improve Adam and push empirical GAN performance even further.
Acknowledgments. We thank Tom Anthony, David Balduzzi, Ali Eslami, Marta Garnelo, Mihaela Rosca and Michael Tschannen for helpful discussions.

References


