On Bock’s Conjecture Regarding the Adam Optimizer

Mohamed Akrout 1  Douglas Tweed 2

Abstract

In 2014, Kingma and Ba published their Adam optimizer algorithm (Kingma & Ba, 2014), together with a mathematical argument that was meant to help justify it. In 2018, Bock and colleagues (Bock et al., 2018) reported that a key piece was missing from that argument — an unproven lemma which we will call Bock’s conjecture. Here we show that this conjecture is false, but we prove a modified version of it which can take its place in analyses of Adam.

1. Introduction

Kingma and Ba (Kingma & Ba, 2014) tried to prove that their Adam optimizer zeroed the error-measure known as average regret, in a learning task called online convex optimization (Zinkevich, 2003). Rubio (Rubio, 2017) and Bock et al. (Bock et al., 2018) found mistakes in the proof, and Bock et al. managed to repair most of them, but they could not verify one key statement, called Lemma 10.4 in Kingma and Ba’s paper and Conjecture 4.2 in Bock’s. We will show that this conjecture is in fact false, but that a modified version of it is true, though no version can provide a general proof of convergence for Adam in the setting of online convex optimization.

For tractability, most analyses of Adam have used variants of the algorithm that are slightly different from the one generally employed in deep learning. Here, we will use the version laid out in Algorithm 1, which differs from that of Kingma, Ba, and Bock et al. only in that they set \( \lambda_m = \lambda_g \in (0, 1) \). We will explain the significance of this difference where it becomes relevant.

In this algorithm, each of the variables \( g_t \), \( m_t \), \( v_t \), \( \hat{m}_t \), \( \hat{v}_t \), and \( \theta_t \) is a real-valued vector; for instance \( g_t \) is the \( g \)-vector at time \( t \). But the vector operations in Adam are all element-wise, except possibly in line 2, and therefore we can analyse the parts after that line element-wise, i.e. we can assume throughout this paper that the vectors \( g_t \) etc. have one element each (except in Section 4, where calculations of regret depend on non-element-wise operations outside Adam). We will write \( g_{1:T} \) for the \( T \)-element vector \([g_1, g_2, ..., g_T]\). We also define

\[
x_1 \triangleq 1 - \beta_1, \quad x_2 \triangleq 1 - \beta_2,
\]

and

\[
s_T \triangleq \sum_{t=1}^{T} \frac{\hat{m}_t^2}{\sqrt{\hat{v}_t}}
\]

which is central to Bock’s conjecture. We assume \( g_1 \neq 0 \) because otherwise \( s_T \) is undefined.

We can now state

**Bock’s conjecture.** In Algorithm 1, if \( \lambda_m = \lambda_g \in (0, 1) \) and \( \gamma = \beta_1^2 / \sqrt{\beta_2} < 1 \) then for any \( g_{1:T} \) we have

\[
s_T \leq \frac{2}{(1 - \gamma)} \frac{1}{\sqrt{1 - \beta_2}} \|g_{1:T}\|_2.
\]

We will call the right-hand side of this inequality the Kingma-Ba or K-B bound.

---

1AIP Labs, Budapest, Hungary 2Department of Physiology, University of Toronto, Toronto, Canada. Correspondence to: Douglas Tweed <douglas.tweed@utoronto.ca>.
On Bock’s Conjecture Regarding the Adam Optimizer

2. Counterexample to Bock’s conjecture

Consider vectors $g_{t:T}$ where $g_t > 0 \forall t$. Set $\beta_1$ and $\beta_2$ equal, i.e., $\beta_1 = \beta_2 = \beta$, and observe that $s_{T}$ and the K-B bound are right-continuous functions of $\beta$ at $\beta = 0$. So if we can find a counterexample where $\beta = 0$ then there also exist counterexamples where $\beta \in (0, 1)$.

Letting $\beta \to 0$, we get $\hat{w}_t = g_t$ and $\hat{v}_t = g_t^2$ (from lines 3–6 of Algorithm 1) and $\gamma = \beta^{3/2} = 0$. Bock’s conjecture then takes the form

$$\sum_{t=1}^{T} \frac{g_t}{\sqrt{t}} \leq 2 \|g_{1:T}\|_2.$$ 

If we choose $g_t = 1/\sqrt{t}$ then this inequality becomes

$$\sum_{t=1}^{T} \frac{1}{t} \leq 2 \sqrt{\sum_{t=1}^{T} \frac{1}{t}},$$

which is false when the left-hand side $> 4$, as happens when $T > 30$.

By continuity, (3) is also violated in cases where $\beta \in (0, 1)$. This plot shows an example:

![Figure 1. We ran Algorithm 1 for 200 time steps using $\beta_1 = \beta_2 = 0.1$, $\lambda_m = \lambda_g = 1 - 10^{-8}$, and $g_t = 1/\sqrt{t}$, and we computed the K-B bound and $s_t$ at each step. $s_t$ surpassed the bound at $t = 59$.](image)

3. Modifying Bock’s conjecture

We want to replace the K-B bound on the right-hand side of (3) with a different bound that we can verify, at least for values of $\beta_1$ and $\beta_2$ that are typically used in AI applications of Adam.

Lemma 3.1. In Algorithm 1, if $\lambda_m = \lambda_g = 1$, $\rho = \beta_2/\beta_1^2 \in (1, 2)$, and $K = \rho/(\rho - 1)$ then $\forall t \in [1, \infty)$

$$\frac{m_t^2}{v_t} < K \frac{x_t^2}{x_2},$$

(4)

Proof. By induction:

(i) At $t = 1$, we have

$$\frac{m_1^2}{v_1} = \frac{x_1^2 g_1^2}{x_2 g_1^2} = \frac{x_1^2}{x_2},$$

and so

$$\frac{m_1^2}{v_1} < K \frac{x_1^2}{x_2}$$

because $K > 1$.

(ii) Next we show that if (4) holds at any time $t$ then it still holds at $t + 1$, i.e.:  

$$\frac{m_{t+1}^2}{v_{t+1}} - K \frac{x_{t+1}^2}{x_2} < 0,$$

(5)

or equivalently,

$$m_{t+1}^2 - K \frac{x_{t+1}^2}{x_2} v_{t+1} < 0.$$  

If we substitute the formulas for $m_{t+1}$ and $v_{t+1}$ from lines 3 and 4 of Algorithm 1, and use the definitions in (1), the left-hand side becomes

$$(\beta_1 m_t + x_1 g_{t+1})^2 - K \frac{x_t^2}{x_2} (\beta_2 v_t + x_2 g_{t+1}^2).$$

(6)

We expand the squared sum, rearrange, and apply (4) to see that (6) is less than

$$\beta_1^2 m_t^2 + 2 \beta_1 m_t x_1 g_{t+1} - (K - 1) x_1^2 g_{t+1}^2 - \beta_2 m_t^2.$$  

We break up the first addend into a sum of two terms to get

$$\frac{K}{K-1} \beta_1^2 m_t^2 - 1 \beta_1^2 m_t^2 + 2 \beta_1 m_t x_1 g_{t+1} - (K - 1) x_1^2 g_{t+1}^2 - \beta_2 m_t^2,$$

which is

$$\left( \frac{K}{K-1} \beta_1^2 - \beta_2 \right) m_t^2 - \left( \frac{1}{\sqrt{K-1}} \beta_1 m_t - \sqrt{K-1} x_1 g_{t+1} \right)^2.$$  

By the definition of $K$, the top line here equals 0, and the quantity as a whole $\leq 0$, proving (5).
Lemma 3.2. In Algorithm 1, if $\lambda_m = \lambda_g = 1$ and $\beta_2 \geq 2 \beta_1 - \beta_1^2$ then \( \forall t \in [1, \infty) \)

\[
\frac{\hat{m}_t^2}{v_t} \leq \frac{m_t^2}{v_t}.
\]

Proof. From lines 5 and 6 of Algorithm 1 we have

\[
\frac{\hat{m}_t^2}{v_t} = c_t \frac{m_t^2}{v_t} \text{ where } c_t \triangleq \frac{1 - \beta_1^2}{(1 - \beta_1)^2}.\]

Note that $c_t = x_2/x_1^2$, which is $\leq 1$ when $\beta_2 \geq 2 \beta_1 - \beta_1^2$. To prove that $c_t \leq 1 \ \forall t \in [1, \infty)$, we define this function of continuous time:

\[
h(t) \triangleq (1 - \beta_1^2) - (1 - \beta_1)^2. \quad (7)
\]

We will show that \( \forall t \in [1, \infty) \)

\[
\frac{dh}{dt} \leq 0 \text{ whenever } h(t) = 0,
\]

because that means $h(t)$, starting at $h(1) = x_2 - x_1^2 \leq 0$, can never cross over to any positive value, and therefore $c_t$ stays $\leq 1$.

We have

\[
\frac{dh}{dt} = -\log \beta_2 + (1 - \beta_1^2) \log \beta_2 + 2 \beta_1 (1 - \beta_1) \log \beta_1,
\]

and if $h(t) = 0$,

\[
\frac{dh}{dt} = -\log \beta_2 + (1 - \beta_1^2) \log \beta_2 + 2 \beta_1 (1 - \beta_1) \log \beta_1,
\]

because then $(1 - \beta_1^2) = (1 - \beta_1)^2$ by the definition in (7).

We define $\alpha \triangleq 1 - \beta_1^2$ to get, $\forall \alpha \in [\alpha_1, 1]$,

\[
\frac{dh}{dt} = -\log \beta_2 + \log \beta_2 \alpha^2 + 2 \log \beta_1 \alpha (1 - \alpha)
\]

\[
= \log \beta_2 \left((1 - r) \alpha^2 + r \alpha - 1\right), \quad (8)
\]

where $r \triangleq 2 \log \beta_1 / \log \beta_2$, which $> 1$ when $\beta_2 \geq 2 \beta_1 - \beta_1^2$. Because $r > 1$, the polynomial $P(\alpha)$ is concave down. It follows that $P(\alpha) \geq 0$ on $[\alpha_1, 1]$, because $P(1) = 0$ and $P(x_1) \geq 0$ by the conditions on $\beta_1$ and $\beta_2$ (see the Appendix). Therefore by (8), $dh/dt \leq 0$ at any $t \in [1, \infty)$ where $h(t) = 0$, which means $h$ can never cross 0 and $c_t$ cannot exceed 1.

\[\square\]

Result 3.3. In Algorithm 1, if $\lambda_g = 1$, $\beta_2 < 2 \beta_1^2$, $\beta_2 \geq 2 \beta_1 - \beta_1^2$, $K = \beta_2 / (\beta_2 - \beta_1^2)$ as in Lemma 3.1, and $\tau = [-\log(2)/\log(\beta_1)]$ then $\forall T \in [1, \infty)$

\[
s_T < (2 + \sqrt{\tau}) \sqrt{1 + K \frac{x_1^2}{x_2} \log T} \|g_{1:T}\|_2. \quad (9)
\]

Proof. We may assume that $\|g_{1:T}\|_2 = 1$, as $s_T$ is a homogeneous function of degree 1 of $g_{1:T}$; that is, if we multiply every element of $g_{1:T}$ by a constant, $\zeta$, then the effect on $s_T$ is to multiply it by $\zeta$ as well. We can also say that $g_i \geq 0 \ \forall t \in [1, \infty)$, as we are seeking an upper bound for $s_T$, and given any $g_{1:T}$ with negative elements, we could always increase $s_T$ by flipping the signs of those negative $g_{i}$. And we can assume that $\lambda_m = 1$, because any $\lambda_m \in (0, 1)$ would only shrink $s_T$, as is clear from (2), lines 3 and 5 of Algorithm 1, and the non-negativity of all the $g_{i}$.

The definition of $s_T$ in (2) shows that it is the dot product of two vectors:

\[
s_T = \hat{m}_{1:T} \cdot \mu_{1:T}, \quad (10)
\]

where $\mu_{1:T}$ is the vector with elements $\mu_t = \hat{m}_t / \sqrt{v_t}$.

The first vector in this dot product has a bounded 2-norm. First of all, $\|m_{1:T}\|_2 \leq \|g_{1:T}\|_2 = 1$ because $m_{1:T}$ is an exponential moving average of $g_{1:T}$, and the 2-norm of such an average cannot exceed the 2-norm of its input. Then we get each $\hat{m}_t$ by multiplying $m_t$ by the factor $1/(1 - \beta_1^2)$. For all $t > \tau$, those factors are $< 2$, so $\|\hat{m}_{1:T+1}\|_2 < 2 \|m_{1:T+1}\|_2 < 2$. And $\|\hat{m}_{1:T}\|_2 \leq \sqrt{T}$ because $\hat{m}_t \leq 1 \ \forall t \in [1, \infty)$. Therefore

\[
\|\hat{m}_{1:T}\|_2 < 2 + \sqrt{T}. \quad (11)
\]

The second vector in the dot product (10) also has a bounded 2-norm. Using the definition of the norm and then Lemmas 3.1 and 3.2, we get

\[
\|\mu_{1:T}\|_2^2 = \sum_{t=1}^{T} \frac{\hat{m}_t^2}{v_t} \leq 1 + \sum_{t=2}^{T} \frac{1}{v_t} \left(\frac{K x_1^2}{x_2^2}\right)
\]

\[
\leq 1 + \left(\frac{K x_1^2}{x_2^2}\right) \int_{1}^{T} \frac{1}{t} \, dt
\]

\[
= 1 + K \frac{x_1^2}{x_2} \log T,
\]

and

\[
\|\mu_{1:T}\|_2 \leq \sqrt{1 + K \frac{x_1^2}{x_2} \log T}. \quad (12)
\]

Therefore by (10), (11), (12), and the Cauchy-Schwarz inequality, we have (9).

\[\square\]

The range of $\beta$ values that is permissible, given the conditions in Result 3.3, is shown in green in the next picture. For instance if $\beta_1 = 0.9$ then we must have $\beta_2 \in [0.99, 1)$. This range includes the $\beta$ values most commonly used in deep learning.
4. Convergence of Adam

Bock and colleagues, like Kingma and Ba before them, analysed Adam in the setting of online convex optimization, by looking at regret (Zinkevich, 2003). In their Theorem 4.4, Bock et al. tried to show that, under certain conditions, Adam would make average regret converge to 0. Our upper bound on \( s_T \) in (9) can replace the K-B bound in Bock’s attempted proof of Theorem 4.4, but the proof still fails because of a mistake in the final inequality involving the term \( 2 \), near the end of page 3.

And there is no point trying to fix that mistake without also adding new conditions to the theorem, because as it stands, Theorem 4.4’s guarantee of convergence is false. Reddi and colleagues (Reddi et al., 2019) have presented a task where Adam fails to drive average regret to 0. In their demonstration, Reddi et al. optimized using Adam and projection rather than Adam alone, which meant they violated the conditions of Bock’s Theorem 4.4. But by adding weight decay it is easy to create a modified version of Reddi’s task where the conditions of Theorem 4.4 are fulfilled but Adam nonetheless fails to zero the average regret.

5. Conclusion

Our upper bound on \( s_T \) in (9) can replace the Kingma-Ba bound in analyses of the Adam optimizer. Our findings also resolve the clash between the attempted proof of convergence for Adam in (Bock et al., 2018) and the failure of convergence demonstrated in (Reddi et al., 2019).

References

Bock, S., Goppold, J., and Weiß, M. An improvement of the convergence proof of the adam-optimizer. arXiv preprint arXiv:1804.10587, 2018.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980, 2014.

Reddi, S. J., Kale, S., and Kumar, S. On the convergence of adam and beyond. arXiv preprint arXiv:1904.09237, 2019.

Rubio, D. M. Convergence analysis of an adaptive method of gradient descent. University of Oxford, Oxford, M. Sc. thesis, 2017.

Zinkevich, M. Online convex programming and generalized infinitesimal gradient ascent. In Proceedings of the 20th international conference on machine learning (ICML-03), pp. 928–936, 2003.

Appendix

In our proof of Lemma 3.2, we said that \( P(x_1) \geq 0 \), where \( P \) was the polynomial in (8), i.e.

\[
(1 - r)x_1^2 + rx - 1 \geq 0,
\]

where \( r \triangleq 2 \log \beta_1 / \log \beta_2 \). To verify (13), we observe that it is equivalent to

\[
r \geq \frac{1 - x_1^2}{x_1 - x_1^2},
\]

which, by the definition of \( r \), is in turn equivalent to

\[
\beta_2 \geq \beta_1 (1 - \frac{x_1^2}{1 - x_1^2}),
\]

Now given that \( \beta_2 \geq 2 \beta_1 - \beta_1^2 \) in Lemma 3.2, it will suffice to show that

\[
2 \beta_1 - \beta_1^2 \geq \beta_1 (1 - \frac{\gamma}{\beta_1}),
\]

i.e.

\[
y(\beta_1) \triangleq \beta_1^{\frac{\beta_1}{1 - \beta_1}} + \beta_1 - 2 \leq 0,
\]

for \( \beta_1 \in (0, 1) \).

Straightforward calculations show that

\[
y(1) = 0, \quad \frac{dy}{d\beta_1}(1) = 0, \quad \frac{d^2y}{d\beta_1^2}(1) = -2,
\]

i.e. \( y(1) \) is a strict local maximum.

To see that \( y \leq 0 \) in \((0, 1)\), we compute

\[
\frac{dy}{d\beta_1} = \beta_1^{\frac{\beta_1}{1 - \beta_1}} \left( \frac{-2 \log \beta_1 + \beta_1 - 2}{(2 - \beta_1)^2} \right) + 1
\]

(15)
and observe that if $y$ were 0 at any $\beta_1 \in (0, 1)$, then by (14) the term
\[
\frac{\beta_1^{\delta_1}}{\beta_1^{\gamma_1 - 2}}
\]
would $= 2 - \beta_1$, and (15) would become
\[
\frac{dy}{d\beta_1} = \frac{-2 \log \beta_1}{2 - \beta_1} > 0.
\]
So if $y$ were $\geq 0$ at any $\beta_1' \in (0, 1)$ then it would stay $\geq 0$ on $(\beta_1', 1)$, contradicting the fact that $y(1)$ is a strict local maximum. Therefore $y$ must remain $\leq 0$ in $(0, 1)$, confirming (14) and (13).