A Review on Adam Stochastic Optimization

Vladislav Trukhin [*1] **Uzair Mirza** [*1]

Abstract

Adaptive moment estimation, Adam, is a popular stochastic optimization algorithm used in neural networks. It belongs to a class of adaptive learning rate algorithms, such as RMSprop, while also combining momentum techniques from SGD Momentum to achieve better performance. We analyze the main theorem of Adam and its assumptions and test Adam's performance relative to other common SGD algorithms.

1. Background

With the increasing growing complexity of neural networks, a demand is created for faster and more efficient optimization techniques. Gradient descent is one of the most commonly used techniques for optimization with the prevalent variant being stochastic gradient descent, SGD. The vanilla SGD update rule is the following.

$$\theta_{t+1} = \theta_t - \alpha \nabla_{\theta_t} f_t(\theta_t)$$

θ_i is the parameter vector at timestep i and α is the stepsize, which is a hyper-parameter that dictates the magnitude of progress made in the negative $\nabla_{\theta_t} f_t(\theta_t)$ direction for next itteration.

A thing to note here that, it is not without challenges, as choosing and updating a learning rate correctly is difficult but necessary to optimize the algorithm further and optimizing over non-convex functions can stall the algorithm. In response to these issues, other derivations of SGD have been proposed.

Momentum SGD attempts to alleviate the later. SGD has difficulty in navigating ravine surface curves where it will oscillate without making significant progress. It achieves this by adding a momentum component to the update rule in the form of reusing the past update vector.

$$v_t = \gamma v_{t-1} + \alpha \nabla_{\theta_t} f_t(\theta_t)$$

$$\theta_{t+1} = \theta_t - v_t$$

The new parameter v_i, is the moment component at timestep i. γ another hyper-parameter here, dictates the magnitude of previous moments on the next update.

This allows Momentum SGD to overcome ravines similar to an analogy of a rolling a ball down a hill.

RMSprop is an adaptive learning rates method which uses separate learning rates for each parameter, effective for sparse data where there is mismatch in frequency between different features. It achieves this by modifying the standard learning rate by an exponentially decaying average of squared gradients.

$$E((\nabla_{\theta_t} f_t(\theta_t))^2) = 0.9E((\nabla_{\theta_{t-1}} f_t(\theta_{t-1}))^2) + (\nabla_{\theta_{t-1}} f_t(\theta_{t-1}))^2$$

$$\theta_{t+1} = \theta_t - \frac{\alpha}{\sqrt{E((\nabla_{\theta_t} f_t(\theta_t))^2) + \epsilon}} \nabla_{\theta_{t-1}} f_t(\theta_{t-1})$$

The new added parameter ϵ, is another hyper-parameter added to deal with the problem of update explosion in the case of vanishing gradients $\nabla_{\theta_t} f_t(\theta_t)$.

Through this deviation, RMSprop helps to alleviate the former issue with updating the learning rate.

With these two techniques each solving a specific issue with SGD, it was only a matter of time for a new deviation to develop that combines the two. Such a technique arose in a 2014 paper, known as Adam (**?**).

2. Adam

The paper proposes an algorithm to combine the strengths of both momentum SGD and RMSprop. Adam, name derived from adaptive moment estimation, utilizes estimates of both

[*]Equal contribution [1]Department of Computer and Mathematical Sciences, University of Toronto, Toronto, Canada. Correspondence to: Vladislav Trukhin <vladislav.trukhin@mail.utoronto.ca>, Uzair Mirza <uzair.mirza@mail.utoronto.ca>.

Proceedings of the 39^{th} International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022.

the first and the second moment for its parameter updates. It begins by computing the decaying average of past gradients and past squared gradients, m_t and v_t.

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

m_t and v_t correspond to the first and second moment. The terms are afterwards bias-corrected to $\hat{m}_t$ and $\hat{v}_t$

$$\hat{m}_t = \frac{m_t}{1-\beta_1^t}$$
$$\hat{v}_t = \frac{v_t}{1-\beta_2^t}$$

Finally the update rule becomes:

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t}+\epsilon}\hat{m}_t$$

The final algorithm is summarised as

Algorithm 1 Adam

Require: α
Require: $\beta_1, \beta_2 \in [0, 1)$
Require: ϵ
Require: θ_0
$m_0 \leftarrow 0$
$v_0 \leftarrow 0$
$t \leftarrow 0$
while θ_t does not converge **do**
  $t \leftarrow t + 1$
  $g_t \leftarrow \nabla_\theta f_t(\theta_{t-1})$
  $m_t \leftarrow \beta_1 \cdot m_{t-1} + (1-\beta_1) \cdot g_t$
  $v_t \leftarrow \beta_2 \cdot v_{t-1} + (1-\beta_2) \cdot g_t^2$
  $\hat{m}_t \leftarrow m_t/(1-\beta_1^t)$
  $\hat{v}_t \leftarrow v_t/(1-\beta_2^t)$
  $\theta_t \leftarrow \theta_{t-1} - \alpha \cdot \hat{m}_t/(\sqrt{\hat{v}_t}+\epsilon))$ (Parameter update)
end while
return θ_t

The breakdown of the components involved in the algorithms is given in Table 1

The paper provides a theoretical guarantee for the regret bound $R(T)$ for Adam as the following corollary.

Corollary 2.1. *Assume that the function f_t has bounded gradients, $||\nabla f_t(\theta)||_2 \leq G$, $||\nabla f_t(\theta)||_\infty \leq G_\infty$ for all $\theta \in R^d$ and distance between any θ_t generated by Adam is bounded, $||\theta_n - \theta_m||_2 \leq D$ and $||\theta_n - \theta_m||_\infty \leq D_\infty$ for any $m, n \in \{1, ...T\}$. Along with $\beta_1, \beta_2 \in [0, 1)$ satisfying $\beta_1^2 < \sqrt{\beta_2}$. Adam achieves the following guarantee, for all $T \geq 1$.*

$$\frac{R(T)}{T} = O(\frac{1}{\sqrt{T}})$$

where $R(T) = \Sigma_{t=1}^{T}[f_t(\theta_t) - f_t(\theta^*)]$, $f_1(\theta), ..., f_T(\theta)$ are an arbitrary, unknown sequence of convex cost functions at each time t, θ_t is the parameter of interest in predicting, and $\theta^* = arg\min_\theta \Sigma_{t=1}^{T} f_t(\theta)$.

Table 1. Variables with suggested values for hyper-parameter

Variables	Details
α	step size for each update. *(Good default value $\alpha = 0.001$.)*
β_1	Exponential decay rate for 1^{st} moment estimate. *(Good default value $\beta_1 = 0.9$.)*
β_2	Exponential decay rate for 2^{st} moment estimate. *(Good default value $\beta_2 = 0.999$.)*
ϵ	Correction term for vanishing gradients. *(Good values $\epsilon = 10^{-8}$)*
t	time step
θ_i	Parameter vector at step i
m_i	1^{st} moment estimate vector at step i
v_i	2^{nd} moment estimate vector at step i
$g_t/\nabla_{\theta_t} f_t(\theta_t)$	Gradients w.r.t. stochastic objective at step t
$\hat{m}_t$	Bias-corrected 1^{st} moment estimate
$\hat{v}_t$	Bias-corrected 2^{nd} raw moment estimate

Using this corollary, the paper concludes that the Regret of Adam converges.

$$\lim_{T\to\infty} \frac{R(T)}{T} \to 0$$

3. Analysis of Adam

3.1. Assumptions

The assumption of f_t having bounded gradients for each time step t is a standard assumption in convergence proofs for SGD. This can be seen by intuition as the standard SGD update rule relies on a gradient to update the parameter, ensuring it does not diverge.

The assumption of bounded distance between parameters of different time steps is necessary to likewise ensure $R(T)$ does not diverge.

Convexity in the cost function $f_t(\theta)$ is also assumed. This is due to non-convex convergence being difficult to prove and in most settings a smooth non-convex function can be lower

bounded by a convex function. Hence a bound guarantee or estimations based on this convex function will be sufficient for non-convex convergence.

3.2. Regret Analysis

The regret for Adam is given as,

$$R(T) \leq \frac{D^2}{2\alpha(1-\beta_1)}\Sigma_{i=1}^{d}\sqrt{T\hat{v}_{T,i}} + \frac{\alpha(1+\beta_1)G_\infty}{(1-\beta_1)\sqrt{1-\beta_2}(1-\gamma)^2}\Sigma_{i=1}^{d}\|g_{1:T,i}\|_2 + \Sigma_{i=1}^{d}\frac{D_\infty^2 G_\infty\sqrt{1-\beta_2}}{2\alpha\beta_1(1-\lambda)^2} \quad (1)$$

Here we have purposely broken down the regret in 3 line-breaks to see the meaning and effect each term has.

The first term,

$$\frac{D^2}{2\alpha(1-\beta_1)}\Sigma_{i=1}^{d}\sqrt{T\hat{v}_{T,i}}$$

can be further broken down in 2 terms as $D^2/2\alpha(1-\beta_1)$ and $\Sigma_{i=1}^{d}\sqrt{T\hat{v}_{T,i}}$ and analysed. Here the first term says that a large distance between the parameters discovered by the algorithm will contribute to a larger regret however note this can be controlled by the step size and the exponential decay of the 1st moment. The second term here accounts for the amount of variance encountered scaled by the total time. Meaning that we get heavily penalised for noisy updates as a later stage.

Now the second term,

$$\frac{\alpha(1+\beta_1)G_\infty}{(1-\beta_1)\sqrt{1-\beta_2}(1-\gamma)^2}\Sigma_{i=1}^{d}\|g_{1:T,i}\|_2$$

which again can be broken down in 2 terms as $\alpha(1+\beta_1)G_\infty/(1-\beta_1)\sqrt{1-\beta_2}$ and into $(1-\gamma)^2\Sigma_{i=1}^{d}\|g_{1:T,i}\|_2$. Here we can look at the first term as a ratio. The numerator which consists of the step size, exponential decay of the first moment and the upper bound for the gradients all contribute to the higher regret. The denominator consists both exponential the decay terms but in negative forms. Hence meaning in order to minimize the regret this ratio between them should all be minimized. The second term here $(1-\gamma)^2\Sigma_{i=1}^{d}\|g_{1:T,i}\|_2$ accounts for the magnitude of all the previous gradients encountered is scaled by the first ratio term hence highlighting the important of convergence as T increases.

The final term

$$\Sigma_{i=1}^{d}D_\infty^2 G_\infty\sqrt{1-\beta_2}/(2\alpha\beta_1(1-\lambda)^2)$$

This term relies on the bound of the distance and the gradient to be in relationship with the step size and exponential decay of the first moment. Here we can observe that as the bounds increase the regret penalty will increase. The magnitude of effect these bounds have is determined by the ratio of $\sqrt{1-\beta_2}/(2\alpha\beta_1(1-\lambda)^2)$

3.3. Regret, Convergence Comparisons

Comparisons of convergence and the regret guarantee under similar assumptions gives us a good idea about the performance between the algorithms. As Adam builds up on Momentum and RMSprop. We compare and investigate the bounds between them.
The Regret guarantee for Momentum derived by(**?**) under similar assumptions is given by,

$$R(T) \leq \frac{\gamma}{(1-\gamma)}(f(\theta_1) - f(\theta^*)) + \|\theta_1 - \theta^*\|^2$$

Setting the max bound on the norm to be D we get the convergence of the regret bound as,

$$R(T) = O(D^2)$$

Here we can see that the final guarantee is dependant on the norm rather that T and hence fails to converge to 0 as T increases.
Hence in general we can expect Adam to achieve better convergence than Momentum.

We now look at the other algorithm on which Adam is based. RMSprop which incorporates having adaptive learning rate for each parameter. In a similar setting (**?**) were able to achieve the regret bound given by,

$$R(T) \leq (\frac{D_\infty^2}{2\alpha} + \frac{\alpha(2-\gamma)}{\gamma})\Sigma_{i=1}^{d}(\sqrt{Tv_{T,i}}) + \sqrt{T}\epsilon_T$$

Hence as stated by (**?**) the regret bound converges to,

$$R(T) = O(log(T)/\sqrt{T})$$

Note, although better, it is still not the best $O(\sqrt{T})$ *sub-linear* convergence achieved by Adam.

3.4. Regret Failure under Assumptions

We now show how even in the optimal setting and with following all the core assumptions we are still not able to achieve the generalised regret guarantee.
The regret bound derived in Corollary 1.1 (**?**) rely on setting the $\epsilon = 0$ hence assuming to never have a problem of $\hat{vt}$ the raw moment vanishing. Even in the case of non-vanishing moment (**?**) were able to show that in a setting where with the constraint $\beta_1 < \sqrt{\beta_2}$, $\theta_i \in [-1, 1]$ and set $\epsilon \neq 0$ and $f_t(\theta)$ as:

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

where $C \geq 2$ and defining the problem for one dimensional case and setting a starting value of $\theta_0 = 1$ and setting $\beta_1 = 0$, $\beta_2 = \frac{2}{(1+c^2)C^2}$ and $\alpha_t = \frac{\alpha}{\sqrt{t}}$ where $\alpha < \sqrt{1-\beta_2}$. Through induction they first established that all the updates were non-negative and that $\theta_{3t} = 1$. Then by calculating the general bound for each possible θ_{3t+i} for $i \in \{0, 1, 2, 3\}$, updates the authors showed that each 3 step iteration of Adam achieves a regret of at-least $2C - 4$. Resulting in a regret bound of;

$$R(T) \geq \frac{(2C-4)T}{3} \implies \lim_{T\to\infty} R(T) \nrightarrow 0$$

Keeping all the previous assumptions constant but for the **general case** of $\epsilon > 0$ and $f_t(\theta)$ being defined as:

$$f_t(\theta) = \begin{cases} C\theta\sqrt{\epsilon}, & \text{for } t \bmod 3 = 1 \\ -\theta\sqrt{\epsilon}, & \text{otherwise} \end{cases}$$

a similar scaled regret bound of $R(T) \geq (2C-4)T\epsilon/3$ was achieved. Hence showing that in this fixed setting while following all the assumptions the problem is not *learnable* using Adam.
Although slightly unlikely but we have now demonstrated how in a controlled setting we're unable to achieve the generalised results.

3.5. Gradients not being Bounded

The main consequence of the gradient not being bounded leads to the failure of Lemma A.3 and Lemma A.4 required to form upper bounds for the regret guarantee. Not having these bounds will result not only in failure of absolute convergence but also for updates to not remain stable.
Also note for the upper bound for the regret on each iteration in the current setting which is,

$$f_t(\theta t) - f_t(\theta^*) \leq g_t^T(\theta_t - \theta^*)$$

when g_t is not bounded we cannot guarantee on this upper bound hence fail to achieve the convergence of regret bound to $O(\sqrt{T})$.

3.6. Norm not being Bounded

In the general setting where the norms between the parameters is bounded with D and D_∞ respectively. When deriving the regret guarantee this allows us to constraint on $(\theta_{t,i} - \theta^*_{,i})^2$ and come up with an upper bound for the guarantee.
The standard consequence of this is the divergence of parameters the after the updates. The best guarantee available would have these norms in the final answer. Hence the regret being achieved would be

$$R(T) = O(\sqrt{T} \cdot (\theta_{t,i} - \theta^*_{,i})^2) \implies O((\theta_{t,i} - \theta^*_{,i})^2)$$

Which no longer converges to 0 as T increases.

4. Empirical Studies on Adam

Table 2. Training Accuracy on MNIST Neural Network Classification

Algorithm	Avg Accuracy (5)	Avg Accuracy (300)
SGD	73.21%	81.59% .
SGD Momentum	61.61%	73.55% .
RMSprop	46.59%	89.29% .
Adam	76.61%	86.65% .

We ran a 64 layered MNIST digit classification neural network under 4 different optimizer algorithms. Each optimizer was run 5 times with batch size of 32, where the average prediction accuracy was taken after 5 and 300 epochs.

We found Adam and RMSprop outperformed SGD which in turn outperformed SGD Momentum. Adam achieved the fastest convergence speed, achieving a very high prediction accuracy after only 5 epochs however the algorithm was narrowly beaten by RMSprop in accuracy by epoch 300. SGD achieved good results for a vanilla algorithm, trailing not far behind Adam and RMSprop. SGD Momentum achieved the worst performance.

This result showcases that Adam does converge, and faster than the other algorithms, while achieving a respectable final accuracy after running through all 300 epochs. These results show Adam has the ability of increasing convergence speed while maintaining a good accuracy.

5. Limitations of Adam

There are several limitations of Adam that prevent it from being objectively better than SGD and other optimization algorithms. The proof for convergence of Adam requires the assumptions of convexity and bounded gradients, an assumption that often is difficult to meet as models become ever more complex. The failure of the theorem showcases that Adam is not yet fully understood, presenting another limitation.

Additionally, Adam and adaptive methods have been found to struggle in generalization in comparison to SGD and SGD with momentum for the same amount of hyperparameter tuning (**?**). This means vanilla optimization methods continue to be a strong candidate for models, making the decision between using Adam versus other methods less clear-cut.

6. Conclusion

Adam is a rather simple algorithm in that it combines the advantages of both SGD Momentum and RMSprop to achieve faster optimization. Despite being intuitive, analysis of its convergence properties have been difficult to derive even under convex settings, as seen in the example of the theorem failing. In our test, Adam manages to converge faster than other SGD algorithms however careful hyper-parameter tuning might be necessary to generalize this performance to all neural network models. While Adam may not be fully understood mathematically, it has managed to secure its position as one of the most popular optimization algorithms for neural networks.

7. Future Directions

With our research we were able to identify the consequences of failure of the assumptions and highlight cases where we fail to generalise even when following the key assumptions. A good place to continue in this direction would be to conduct an ablation study and compare the generalised behaviour in a non-convex setting. This is because, as algorithms become more complex, it becomes more mathematically difficult to find which assumptions are necessary to guarantee convergence.

A. Appendix

A.1. Regret bound and convergence for Adam

Keeping all the assumptions from **Corollary 1.1** we first declare some definitions and lemmas to use for our proof. For simplicity we declare some notations to represent $g_t \triangleq \nabla_t(\theta_t)$, $g_{t,i}$ as the i^{th} element. $g_{1:t,i} = [g_{1,i}, g_{2,i}, \dots g_{t,i}]$ (**?**)

Definition A.1. A function $f : R^d \to R$ is convex if for all $x, y \in R^d$ for all $\lambda \in [0, 1]$

$$\lambda f(x) + (1 - \lambda) f(y) \geq f(\lambda x + (1 - \lambda) y)$$

Lemma A.2. *If a function $f : R^d \to R$ is convex if for all $x, y \in R^d$,*

$$f(y) \geq f(x) + \nabla f(x)^T (y - x)$$

Lemma A.3. *let $g_t = \nabla f_t(\theta_t)$ and $g_{1:t}$ be defined as earlier and bounded by, $\|g_t\|_2 \leq G, \|g_t\|_\infty \leq G_\infty$. Then,*

$$\Sigma_{t=1}^{T} \sqrt{\frac{g_{t,i}^2}{t}} \leq 2 G_\infty \|g_{1:T,i}\|_2$$

Lemma A.4. *Let $\gamma \triangleq \frac{\beta_1^2}{\sqrt{\beta_2}}, \beta_i \in [0, 1)$ satisfying $\frac{\beta_1^2}{\sqrt{\beta_2}} < 1$ and $\|g_t\|_2 \leq G, \|g_t\|_\infty \leq G_\infty$ then this inequality holds*

$$\Sigma_{t=1}^{T} \frac{\hat{m}_{t,i}^2}{\sqrt{t \hat{v}_{t,i}}} \leq \frac{2}{1 - \gamma} \frac{1}{\sqrt{1 - \beta_2}} \|g_{1:T,i}\|_2$$

Proof. Lemma A.2 was rearranged and used here to initiate the regret argument and get the bound for each step

$$f_t(\theta t) - f_t(\theta^*) \leq g_t^T (\theta_t - \theta^*) = \Sigma_{i=1}^{d} g_{t,i} (\theta_{t,i} - \theta_i^*) \qquad d \text{ is the dimension of the parameter space}$$

Using the original update rule from the algorithm, we expand the $t + 1$ parameter update and end up with,

$$\theta_{t+1} = \theta_t - \frac{\alpha_t}{1 - \beta_1^t} \left(\frac{\beta_{1,t}}{\sqrt{\hat{v}_t}} m_{t-1} + \frac{(1 - \beta_{1,t})}{\sqrt{\hat{v}_t}} g_t \right)$$

We then would like to look at the bound for $g_{t,i}(\theta_{t,i} - \theta_{,i}^*)$. This was derived by looking at the expansion of the regret square of single update at θ_{t+1}. As $t + 1$ update depends on t then in the expression for $g_{t,i}(\theta_{t,i} - \theta_{,i}^*)$ we can use this derivation to simplify and get upper bound for our gradient update at each step,

$$g_{t,i}(\theta_{t,i} - \theta_{,i}^*) \leq \frac{1}{2\alpha_t(1 - \beta_1)} \left((\theta_{t,i} - \theta_{,i}^*)^2 - (\theta_{t+1,i} - \theta_{,i}^*)^2 \right) \sqrt{\hat{v}_{t,i}} + \frac{\beta_{1,t}}{2\alpha_{t-1}(1 - \beta_{1,t})} (\theta_{t,i} - \theta_{,i}^{*2}) \sqrt{\hat{v}_{t-1,i}}$$

$$+ \frac{\beta_1 \alpha_{t-1}}{2(1 - \beta_1)} \frac{m_{t-1,i}^2}{\sqrt{\hat{v}_{t-1}, i}} + \frac{\alpha_t}{2(1 - \beta_1)} \frac{\hat{m}_{t,i}^2}{\sqrt{\hat{v}_{t,i}}}$$

With this we now have an upper bound for the gradient at each step. Lemma A.4 was further used to control and get an upper bound for $\frac{\hat{m}_{t,i}^2}{\sqrt{\hat{v}_{t,i}}}$ in the previous relation. Next we derive the regret bound by taking the summation across every dimension of the parameter space and get,

$$R(T) \leq \Sigma_{i=1}^{d} \frac{1}{2\alpha_1(1 - \beta_1)} (\theta_{t,i} - \theta_{,i}^{*2}) \sqrt{\hat{v}_{1,i}} + \Sigma_{i=1}^{d} \Sigma_{t=2}^{T} \frac{1}{2(1 - \beta_1)} (\theta_{t,i} - \theta_{,i}^*)^2 \left(\frac{\sqrt{\hat{v}_{t,i}}}{\alpha_t} - \frac{\sqrt{\hat{v}_{t-1,i}}}{\alpha_{t-1}} \right)$$

$$+ \frac{\beta_1 \alpha G_\infty}{(1 - \beta_1)\sqrt{1 - \beta_2}(1 - \gamma)^2} \Sigma_{i=1}^{d} \|g_{1:T,i}\|_2 + \frac{\alpha G_\infty}{(1 - \beta_1)\sqrt{1 - \beta_2}(1 - \gamma)^2} \Sigma \|g_{1:T,i}\|_2$$

$$+ \Sigma_{i=1}^{d} \Sigma_{t=2}^{T} \frac{1}{2(1 - \beta_1)} (\theta_{t,i} - \theta_{,i}^*)^2 \sqrt{\hat{v}_{t,i}}$$

Now with the assumption about the bounded norm ie. $\|\theta_t - \theta^*\|_2 \leq D, \|\theta_m - \theta_n\|_\infty \leq D_\infty$ and using the upper bound for arithmetic series we end up with the final bound for regret as,

$$R(T) \leq \frac{D^2}{2\alpha(1-\beta_1)}\Sigma_{i=1}^{d}\sqrt{T\hat{v}_{T,i}} + \frac{\alpha(1+\beta_1)G_\infty}{(1-\beta_1)\sqrt{1-\beta_2}(1-\gamma)^2}\Sigma_{i=1}^{d}\|g_{1:T,i}\|_2|\Sigma_{i=1}^{d}\frac{D_\infty^2 G_\infty\sqrt{1-\beta_2}}{2\alpha\beta_1(1-\lambda)^2}$$

Note this regret converges to $O(\sqrt{T})$ hence we can rewrite it as,

$$R(T) = O(\sqrt{T}) \rightarrow \frac{R(T)}{T} = O(\frac{1}{\sqrt{T}}) \Longrightarrow \lim_{T\rightarrow\infty}\frac{R(T)}{T} \rightarrow 0$$

□

A.2. Code

The code written to get the results under the experiment section can be found at the repository hosted on github at

https://github.com/vladislavtrukhin/Adamreview