॥ सा विद्या या विमुक्तये ॥

भारतीय प्रौद्योगिकी संस्थान धारवाड

Indian Institute of Technology Dharwad

Mathematics for Data science(CS427)

Course Project

Proof of Local Convergence for the Adam Optimizer

GROUP MEMBERS

No.	NAME	ROLL NO.	BRANCH
1	Aduma Rishith Reddy	210010002	CSE
2	Gorantla Nikhil Sai	210010018	CSE
3	A.V.S.Sreenivasu	210020001	CSE

Contents

1 Adam Optimizer

Adam is a type of optimization method that adjusts the learning rate for each parameter individually based on the estimates of the first and second moments of the gradients. It is a combination of two other algorithms, AdaGrad and RMSprop, and it offers several advantages over them. One advantage is that it is invariant to the scaling of the gradient. Another advantage is that it bounds the stepsizes by the stepsize hyperparameter, which prevents them from becoming too large or too small. A third advantage is that it does not require a stationary objective, which means that it can handle non-stationary or noisy objectives. A fourth advantage is that it works well with sparse gradients, which occur when some parameters are updated infrequently or not at all. A fifth advantage is that it naturally performs a form of step size annealing, which means that it gradually decreases the stepsize over time to converge to a local minimum.

2 Pseudocode

Algorithm 1 Adam, our proposed algorithm for stochastic optimization. For a slightly more efficient (but less clear) order of computation,g_t^2indicates the elementwise square (gt $\odot$gt). Good default settings according to the original authors are $\alpha = 0.001$,$\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-8}$. All operations on vectors are element-wise. With β_1^t and β_2^t and we denote β_1 and β_2 to the power t.

```
Require: $\alpha \in \mathbb{R}^+, \epsilon \in \mathbb{R}, (\beta_1, \beta_2) \in (0,1), w_0 \in \mathbb{R}^n and\, the\, function\, f(w) \in C^2(\mathbb{R}^n, \mathbb{R})$
Ensure:
  $m_0 \leftarrow 0$
  $v_0 \leftarrow 0$
  $t \leftarrow 0$
while $\theta_t$ not converged do
    $t \leftarrow t + 1$;
    $g_t \leftarrow \nabla_\theta f_t(\theta_{t-1})$ /*Get gradients w.r.t. stochastic objective at timestep t*/
    $m_t \leftarrow \beta_1 m_{t-1} + (1 - \beta_1) g_t$ /*Update biased first moment estimate*/
    $v_t \leftarrow \beta_2 v_{t-1} + (1 - \beta_2) g_t^2$ /*Update biased second moment estimate*/
    $\hat{m}_t \leftarrow m_t / (1 - \beta_1^t)$ /*Compute bias-corrected first moment estimate*/
    $\hat{v}t \leftarrow v_t / (1 - \beta_2^t)$ /*Compute bias-corrected second moment estimate*/
    $\theta_t \leftarrow \theta t - 1 - \alpha \hat{m}_t / (\sqrt{\hat{v}_t} + \epsilon)$ /*Update parameters*/
end
```

3 Local Convergence Of Adam's Algorithm

3.1 Problem

For an algorithm to be effective we should be able to show that it converges.This was attempted by the authors of [1],under some very restrictive assumptions, like non divergence, however this was later proved wrong in papers by different authors[2],[3].
The Adam Optimizer has become one of the most popular optimization methods for training neural networks in recent years.Even if it appears to be working, there is no convergence proof for Adam to our knowledge.This paper sets out to discover and analyze a method for proving and analyzing local convergence in a methods clear and concise way which only requires simple calculation unlike other lengthy proofs in [1],[4] and [5].

3.2 Solution

The authors of the paper intended to provide a method to analyse the local convergence of the algorithm in batch mode for deterministic fixed training set,which provides us with necessary conditions for hyperparameters(1) of Adam Algorithm to achieve convergence.Due to the local nature of the arguments the objective function can be non-convex but must be at least twice continuously differentiable.

$$\frac{\alpha \max_{i=1}^{n}(\mu_i)}{\sqrt{\epsilon}}(1-\beta_1) < 2\beta_1 + 2 \tag{1}$$

Here μ_i is the i^{th} eigen value of $\nabla^2 f(w_*)$ (Hessian)

3.3 Proof Of Local Convergence

We consider a dynamic system to represent our learning algorithm.We define a common state vector x consisting of the moments –like m and v for Adam– and the weights,so we have $x = (m, v, w)$. Then the optimization process can be written as an iteration $x_{t+1} = T(t, x_t)$ for some function $T : \mathbb{N}_0 \times X \to X\ with\ X \subset \mathbb{R}^p$.We have to show that this condition leads to a fixed point x_* of T, where the moments are all zero. We can check for the convergence using Banach fixed point theorem and check the stability of the algorithm using Lyapunov theorem.Asymptotic stability is equivalent to convergence of the iteration defined T to x for all x_0 sufficiently close to x.The conditions needed for the fixed point analysis and stability results require the learning rate to be sufficiently small.

3.3.1 Lyapunov Theorem

Given $\dot{x} = f(x)$ and define $V(x)$ on Ω such that

- $V(x) = 0$ at $x = x_*$
- $V(x) > 0$ at $all\ x\ except\ x = x_*$

- $$\dot{V}(x) = \nabla V(x).f(x) < 0 \; \forall \, x \in \; \Omega - \{x_*\}$$

If such a function $\dot{V}(x)$ exists then we can say that x_* is Locally Asymptotically Stable.

3.3.2 Banach Fixed Point Theorem

Banach Fixed point Theorem guarantees the existence and uniqueness of fixed points of certain self-maps of metric spaces, and provides a constructive method to find those fixed points.
In Proof:
A point x_* ∈M is called equilibrium or fixed point if T(t,x_*)=x_* for all t ∈ $\mathbb{N}_0$,so the constant function x_t=x_* for all t ∈ $\mathbb{N}_0$ is a solution of x_{t+1} = T(t, x_t).
In the following the asterisk(*) will always denote equilibria or their components.consider a solution x=x(x_0) of above equation ,x is stable if for all ϵ>o there exists δ=$\delta(\epsilon)$ such that $\tilde{x} = \tilde{x}(\tilde{x_0})$of x_{t+1} = T(t, x_t). $||\tilde{x_0} - x_0||<\delta$ fulfills $||\tilde{x_t} - x_t||<\epsilon$for all t ∈ $\mathbb{N}_0$.
x is called attractive if there exists δ>0 such that any solution $\tilde{x}$ with $||\tilde{x_0} - x_0||<\delta$ fulfils $\lim_{x\to\infty}||\tilde{x_t} - x_t|| = 0$.x is called asymptotically stable if it is stable and attractive .
A contraction is a self mapping on some set with Lipschitz constant L<1 ,i.e mapping $\tilde{T} : M \to M$, $M \subset \mathbb{R}^n$ with $||\tilde{T}(x) - \tilde{T}(Y)|| \leq L||x - y||$ for all x,y ∈ M.
If M is complete, i.e. all Cauchy sequences converge, then a unique fixed point x_* ∈M of $\tilde{T}$ exists by the Banach fixed point theorem

3.3.3 Convergence

Let $w \in \mathbb{R}^n$ be weights of function $f(w)$ which has to be minimized.Using previously defined state space $x = (m, v, w)$ and update using previously declared statements,we can write the optimization process as iteration of time-dependent dynamic system

$$x_{t+1} = [m_{t+1}, v_{t+1}, w_{t+1}]$$

We can split this into autonomous(Independent of t) and non autonomous parts(Dependent of t) as follows.

$$
\begin{aligned}
x_{t+1} =& T(t,x_t)\\
& \overline{T}(x_t)+\Theta(t,x_t)\\
where&\\
\overline{T}(x_t) =& \begin{bmatrix} \beta_1 m_t + (1-\beta_1)g(w_t) \\ \beta_2 v_t + (1-\beta_2)g(w_t)\times g(w_t) \\ w_t - \alpha\frac{(m_{t+1})}{\sqrt{v_{t+1}+\epsilon}} \end{bmatrix} \qquad (2)\\
and&\\
\Theta(t,x_t) =& \begin{bmatrix} 0 \\ 0 \\ \alpha(\frac{\sqrt{1-\beta_2^{t+1}}}{1-\beta_1^{t+1}}-1)\frac{(m_{t+1})}{\sqrt{v_{t+1}+\epsilon}} \end{bmatrix} \qquad (3)
\end{aligned}
$$

Here we can consider $\overline{T}$ to be the autonomous part and Θ to be the disturbance.Then we can find the Jacobian of the autonomous part. Consider w_* to be a critical point then $\nabla f(w_*) = 0$.Then because of this x_* becomes a fixed point as $T(x_*) = x_*$ and as a result x_* can only be $x_* = (0,0,w_*)$ Substituting it we get a simplified Jacobian as

$$
J_{\overline{T}}(0,0,w_*) = \begin{pmatrix} \beta_1 I & 0 & (1-\beta_1)\nabla_w g(w_*) \\ 0 & \beta_2 I & 0 \\ -\frac{\alpha\beta_1}{\sqrt{\epsilon}}I & 0 & I-\frac{\alpha(1-\beta_1)}{\sqrt{\epsilon}}\nabla_w g(w_*) \end{pmatrix}
$$

Here $\nabla_w g(w_*)$ is the Hessian of f $\nabla_w^2 f(w_*)$ which is positive definite. Here $\mu_i \forall i \in \{1,2...n\}$ are the eigen values of $\nabla_w g(w_*)$ Then the eigen values are

$$
\begin{aligned}
\lambda_{1,i} &= \beta_2\\
\lambda_{(2,3),i} &= \frac{(\beta_1+1)\pm\sqrt{(\beta_1+1)^2-4(\beta_1-\frac{\alpha\mu_i(\beta_1-1)}{\sqrt{\epsilon}})}}{2}
\end{aligned}
$$

However According to theorem on local stability as stated in the paper, Consider x_* be fixed point and $\overline{T}$ be continuously differentiable in open neighbourhood $B_r(x_*) \subset M$ If $||J_{\overline{T}}(0,0,w_*)|| < 1$ for some norm then there exists $0 < \epsilon \leq r$ and $0 \leq c < 1$ such that for all x_0 with $||x_0 - x_*|| < \epsilon$ then

$$
||x(t,x_0) - x_*|| \leq c^t ||x_0 - x_*||
$$

This states that x_* is locally exponentially and asymptotically stable.
However we know that if $\max_{i=1}^{n} \lambda_i < 1$ then $||J_{\overline{T}}(0,0,w_*) < 1$ Thus if we can prove that all eigen values of the Jacobian are less than 1 then we can conclude the function is asymptotically stable.

3.3.4 Proving condition on eigen values

Let $\phi_i = \frac{\alpha\mu_i}{\sqrt{\epsilon}}(1-\beta_1)$

$$\begin{aligned}
&We\ know\ that\\
&\lambda_1 = \beta_2 < 1\\
&\lambda_{(2,3)} = \frac{1+\beta_1-\phi_i \pm \sqrt{(1+\beta_1-\phi_i)^2-4\beta_1}}{2}\\
&But\\
&\sqrt{(1+\beta_1-\phi_i)^2-4\beta_1} < \sqrt{(1+\beta_1)^2-4\beta_1} = (1-\beta_1)\\
&The\ equation\ turns\ into\\
&< \frac{1}{2}|1+\beta_1-\phi_1 \pm (1-\beta_1)|\\
&|\lambda_{(2,3)}| < 1\ when\ +\ is\ taken\\
&|\lambda_{(2,3)}| < \beta_1 < 1\ when\ -\ is\ taken
\end{aligned}$$

Thereby we can conclude that all eigen values are less than 1, hence we can say that the function is asymptotically and exponentially stable at x_* We now have to prove that the algorithm converges to the point (here exponentially)

3.3.5 Proving Local Convergence

We know from previous statements that $m_* = 0$ and $g(w_*) = 0$. Using this and the Lipschitz continuity over gradient$(g(w_t))$ of function we attempt to prove the convergence.

$$||\Theta(t,x)|| = \alpha|(\frac{\sqrt{1-\beta_2^{t+1}}}{1-\beta_1^{t+1}} - 1)|\frac{||\beta_1 m + (1-\beta_1)g(w)||}{\sqrt{\beta_2 v + (1-\beta_2)g(w) \times g(w) + \epsilon}}$$

$\beta_2 v + (1-\beta_2)g(w) \times g(w)$ *is greater than* 0 *thus we can write it as*

$$\begin{aligned}
&\leq \frac{\alpha}{\sqrt{\epsilon}}|\frac{\sqrt{1-\beta_2^{t+1}} - (1-\beta_1^{t+1})}{1-\beta_1^{t+1}}|||\beta_1 m + (1-\beta_1)g(w)|| \\
&\leq \frac{\alpha}{\sqrt{\epsilon}(1-\beta_1)}|\frac{1-\beta_2^{t+1} - (1-\beta_1^{t+1})^2}{\sqrt{1-\beta_2^{t+1}} + (1-\beta_1^{t+1})}|\ ||\beta_1 m + (1-\beta_1)g(w)|| \\
&\leq \frac{\alpha}{\sqrt{\epsilon}(1-\beta_1)}|\frac{1-\beta_2^{t+1} - (1-\beta_1^{t+1})^2}{\sqrt{1-\beta_2^{t+1}} + (1-\beta_1^{t+1})}|\ (\beta_1||m-m_*|| + (1-\beta_1)||g(w)-g(w_*)||) \\
&\leq \frac{C}{4}| - \beta_2^{t+1} - 2\beta_1^{t+1} + \beta_1^{2(t+1)}|\ (\beta_1||m|| + (1-\beta_1)||g(w)||) \\
&\leq C\beta^{t+1}(\beta_1||m-m_*|| + (1-\beta_1)L||w-w_*||)
\end{aligned}$$

Here $\beta = max\{\beta_1, \beta_2, \beta_1^2\}$ and $C = \frac{4\alpha}{\sqrt{\epsilon}(1-\beta_1)(\sqrt{1-\beta_2}+(1-\beta_1))}$
Here $(\beta_1||m-m_*|| + (1-\beta_1)L||w-w_*||)$ is of the form of a norm.

$$||(\tilde{m},\tilde{w})||_* = (\beta_1||\tilde{m}|| + (1-\beta_1)L||\tilde{w}||)$$

From equivalence of norm we can estimate as $||(\tilde{m},\tilde{w})||_* \leq \tilde{C}||(\tilde{m},\tilde{w})||$

$$\begin{aligned}
&\leq C\beta^{t+1}\tilde{C}||(m-m_*),(w-w_*)|| \\
&\leq (C\beta\tilde{C})\beta^t||x-x_*||
\end{aligned}$$

This estimate can help us to explain the convergence of the algorithm using the following theorem.

3.3.6 Convergence with perturbation

Let $M \subset \mathbb{R}^n$ be a complete set,$\tilde{T} : M \times M$; be Lipschitz continuous with $L < 1, x_* \in M$ the unique fixed point of $\tilde{T}$.Assume $B_r(x_*) \subset M$ for some $r > 0$. From previous results:

$$\tilde{x}_{t+1} = \tilde{T}(\tilde{x}_t) + \Theta(t,\tilde{x}_t)$$

For $||\Theta(t,\tilde{x}_t)|| \leq C\beta^t||\tilde{x}-x_*||$ for all $\tilde{x}_t \in M,\ t \in \mathbb{N}$ for some $C \geq 0$ *and* $0 < \beta < 1$. Then there exists $\epsilon > 0$ such that for all $\tilde{x}_0 \in M$ with $||\tilde{x}_0 - x_*|| < \epsilon$ then the iterations stays in M and converges to x_*

proof:

Let $x = x(\tilde{(x_0)})$ be the solution of the undisturbed iteration $x_{t+1} = \tilde{T}(x_t)$ for all $\tilde{T}(x_t)$ with initial solution $\tilde{x} = \tilde{x}(\tilde{(x_0)})$ .we define $e_t := ||\tilde{x}_t - x_*||$,estimate using assumptions

$$\begin{aligned} e_{t+1} &= ||\tilde{T}(\tilde{x}_t) + \Theta(t, \tilde{x}_t) - x_*|| \\ &= ||\tilde{T}(\tilde{x}_t) - \tilde{T}(x_*) + \Theta(t, \tilde{x}_t)|| \\ &\leq ||\tilde{T}(\tilde{x}_t) - \tilde{T}(x_*)|| + ||\Theta(t, \tilde{x}_t)|| \\ &\leq L||\tilde{x}_t - x_*|| + C\beta^t||\tilde{x}_t - x_*|| \\ &= (L + C\beta^t)e_t \end{aligned}$$

Choosing t large enough ,we get $0 < L + C\beta^t \leq \tilde{(L)} < 1$ for all $t \geq$K because β,L<1.Then

$$e_t \leq (\prod_{k=1}^{K}(L + C\beta^K))\tilde{L}^{t-k}e_0 =: \tilde{C}\tilde{L}^{t-k}e_0$$

with $\tilde{C}$ independent of $\tilde{x}_0$.So e_t converges to 0 exponentially.
The arguments so far have only been valid if $\tilde{x}_t \in$M,i.e.the iteration is well defined.But choosing $\tilde{x}_0$ such that $e_0 = ||\tilde{x}_0 - x_*|| < \frac{r}{c}$ small enough that we can achieve $e_t \leq \tilde{C}e_0 < r$.

3.3.7 Results

From the above theorem we can conclude that thew Algorithm gives local exponential convergence for the iterational dynamic system. Also from our previous result we know that the point x_* is locally exponentially stable.Because we haven't used the conditions for convex this proof holds for all functions as long as $||\nabla f(w_t)|| < \epsilon$ holds.Additionally the inequality (1) can be used to tune the hyper parameters such that it results in converge.

4 Relevance

This paper has enabled us to explore the various testing criteria for an optimizer. We have learned about the subtle differences between Adam's Optimizer and other optimizers that lead up to it. This work has broadened our knowledge of optimizers beyond what was covered in class and has also enhanced our comprehension of other related concepts taught in class.
We were able to comprehend the mathematics in the proof in large part because of the classes on linear algebra and differential calculus. Additionally, it helped us see each step of the process intuitively and comprehend its importance in achieving our goal.

5 Experiments:

Before the Adam optimizer, Several algorithms were used for optimization like Gradient Descent, Momentum Optimizer, Nesterov Accelerated Gradient, AdaGrad and RMSProp. The performance of RMSProp is almost as good as Adam, but Adam is better in some cases. For this experiment we tested the Adam optimizer on several different functions and the results are as follows

The following images show the path taken by the optimizer for the functions tested.

- Paraboloid
  $f(x, y) = (x - 2)^2 + (y + 3)^2$

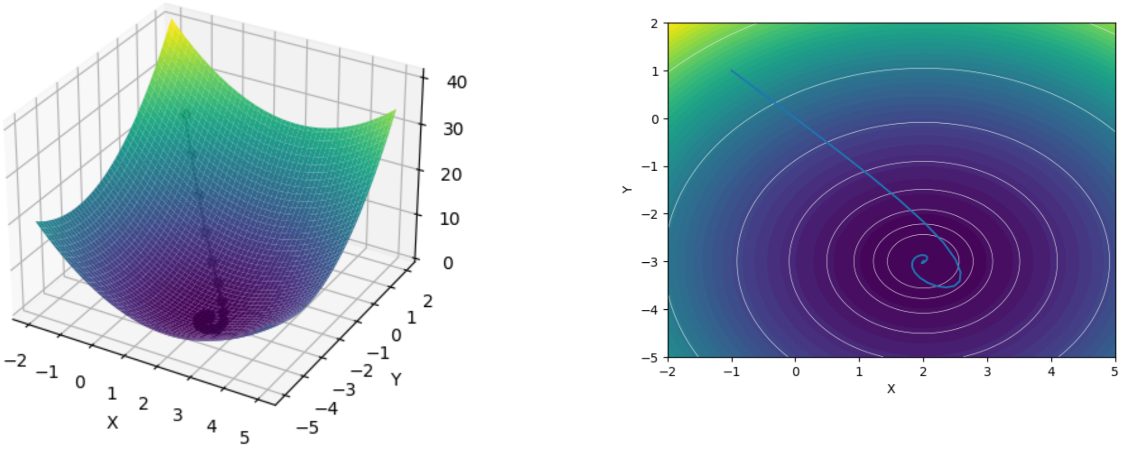


Figure 1: 3D and contour plot of the function along with the path taken by the Adam optimizer to reach the optimum.

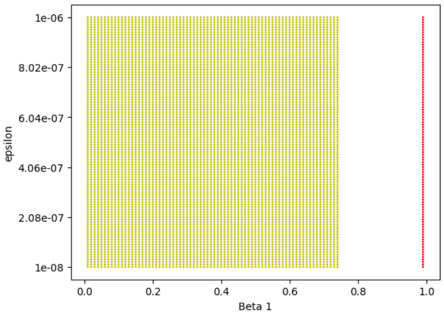


Figure 2: Running the Adam optimizer for different values of β_1 and ϵ

- Rosenbrok Function
  $f(x,y) = (1-x)^2 + 100(y-x^2)^2$

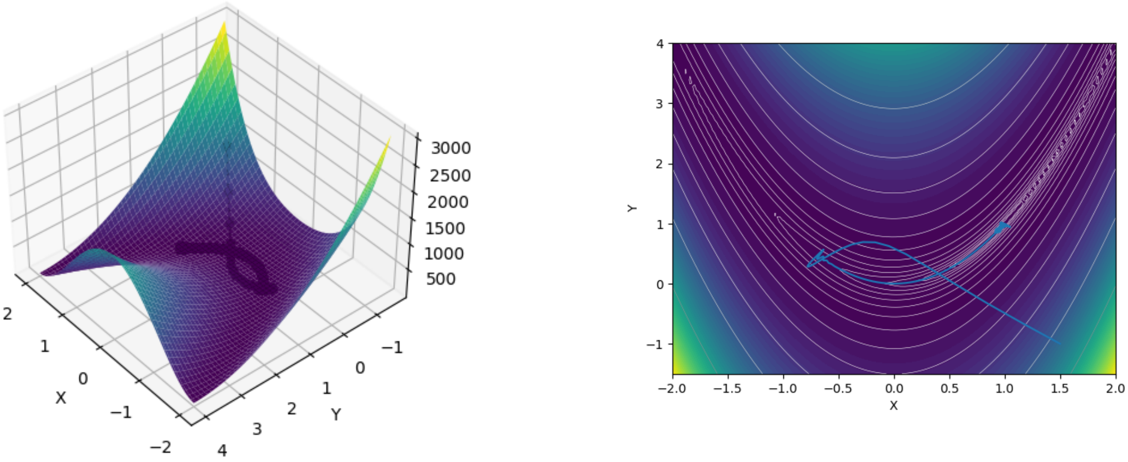


Figure 3: 3D and contour plot of the function along with the path taken by the Adam optimizer to reach the optimum.

- Function from the paper
  $f(x,y) = (x+2)^2(y+1)^2 + (x+2)^2 + 0.1(y+1)^2$

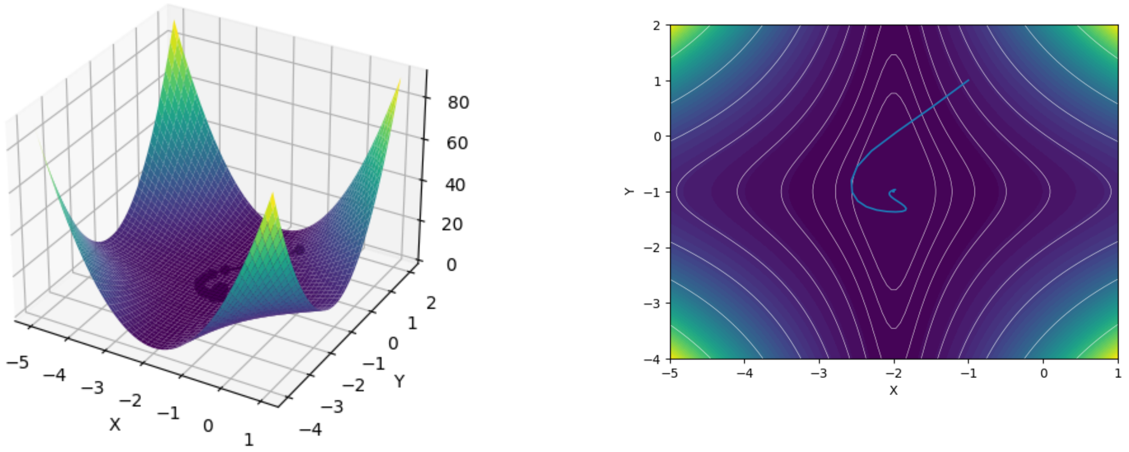


Figure 4: 3D and contour plot of the function along with the path taken by the Adam optimizer to reach the optimum.

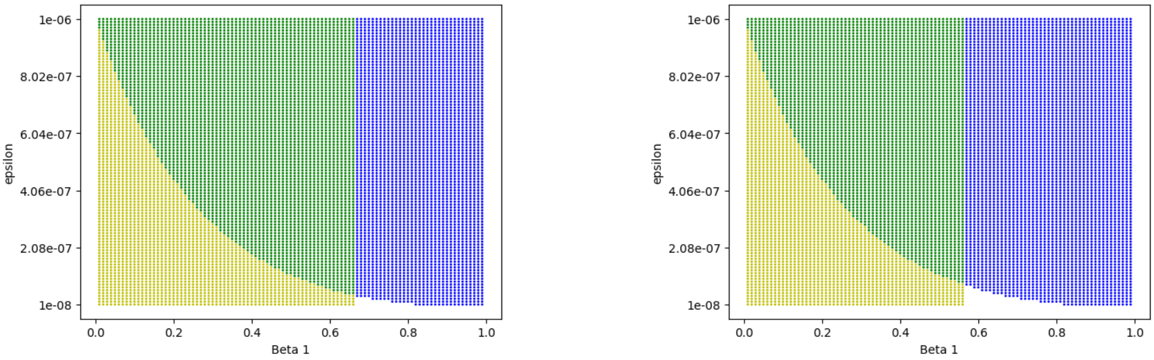
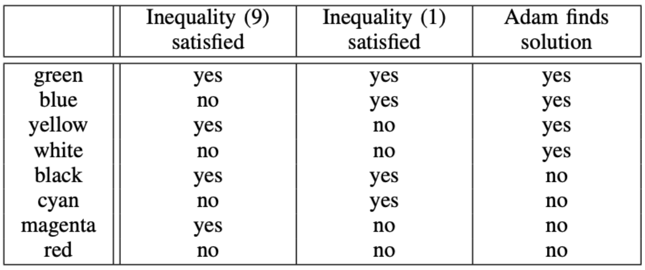


Figure 5: Running the Adam optimizer for different values of β_1 and ϵ for the rosenbrok function and the function from the paper.

The points in the graphs shown above are coloured based on the following coloring scheme



	Inequality (9) satisfied	Inequality (1) satisfied	Adam finds solution
green	yes	yes	yes
blue	no	yes	yes
yellow	yes	no	yes
white	no	no	yes
black	yes	yes	no
cyan	no	yes	no
magenta	yes	no	no
red	no	no	no

Figure 6: Coloring Scheme

We observe that the adam optimizer converges to the optima of the function in reasonably fast time without many oscillations. We used the learning rate to be 0.01 for the above simulations and the β_2 values used are 0.3,0.2 and 0.1 respectively and they were run for 10000 iterations. We can see that the Adam optimizer converges to the optima in almost every case.

References

[1] D. P. Kingma and J. L. Ba. *Adam: A Method for Stochastic Optimization.* CoRR, vol. abs/1412.6980, 2014.

[2] S. Bock. *Rotationsermittlung von Bauteilen basierend auf neuronalen Netzen (unpublished).* Ostbayerische Technische Hochschule Regensburg, M.Sc. thesis, 2017.

[3] D. M. Rubio. *Convergence Analysis of an Adaptive Method of Gradient Descent.* University of Oxford, M.Sc. thesis, 2017.

[4] S. Kale J. S. Reddi and S. Kumar. *On the Convergence of Adam and Beyond.* in International Conference on Learning Representation, 2018.

[5] Xiangyi Chen et al. "On the convergence of a class of Adam-type algorithms for non-convex optimization". English (US). In: 7th International Conference on Learning Representations, ICLR 2019 ; Conference date: 06-05-2019 Through 09-05-2019. Jan. 2019.