MADAM

Momentum Adaptive Directional Adam Mechanism

(Project Report)

Cheerla Parthiv Sagar
Yukteswar Mantha
Advika Ramesh
Sukrit Sharma

COMPUTATIONAL METHODS AND OPTIMIZATION (SEM 3)

Abstarct:

For years, this world has been divided: one domain ruled by **Inertia**, where momentum ($\mathbf{m}_t$) provides directional consistency to overcome local minima and plateaus; the other, by **Adaptivity**, where mechanisms like Adam adjust the step size per parameter by the history of gradient magnitude ($\mathbf{v}_t \propto \mathbf{g}_t^2$) . However, this adaptive scaling often creates a basic conflict. The pace of movement (v_t) becomes sensitive to immediate, noisy gradient spikes. This can disconnect the step size from the stable direction of travel (m_t). We explore a somewhat new approach by introducing the Momentum-Adaptive Directional Adam Mechanism (MADAM). This algorithm is engineered to harmonize these problems(high variance in v_t) .

Motivation:

The efficacy of first-order optimizers is typically measured by their ability to converge quickly without sacrificing final solution quality. The primary challenges addressed by this work come from two critical issues in the optimization domain.

1)Challenges of Ill-Conditioned Surfaces:

Many real-world loss functions, and canonical test cases like the Rosenbrock function, are characterized by highly anisotropic surfaces.Traditional SGD struggles with this geometric complexity, but Adam's reliance on the instantaneous squared gradient ($\mathbf{g}_t^2$) in the denominator can also fail here.

2)Decoupling Problem in Adam :

Adam computes the step size based on $\sqrt{\mathbf{v}_t}$, where $\mathbf{v}_t$ tracks the mean squared magnitude of the raw gradient, $\mathbf{g}_t$.If an outlier batch produces a sharp spike in $\mathbf{g}_t$, $\mathbf{v}_t$ for that specific parameter is immediately inflated. This causes the effective learning rate for that parameter to drop abruptly, even if the overall momentum ($\mathbf{m}_t$) suggests the algorithm was heading in a good direction . The problem of high $\mathbf{v}_t$ variance in early training is a serious, recognized issue in the optimization

Key Innovation:

The novel feature of MADAM is grounded to ADAM , yet , we made a small change while calculating the adaptive scaling factor $\mathbf{v}_t$. The noticeable change in MADAM is the replacement of the instantaneous squared gradient ($\mathbf{g}_t^2$) with the squared first moment estimate ($\mathbf{m}_t^2$) in the accumulation of $\mathbf{v}_t$:

$$\text{Adam (Standard):} \quad \mathbf{v}_t = \beta_2 \mathbf{v}_{t-1} + (1-\beta_2)\mathbf{g}_t^2$$

$$\mathbf{MADAM(Novel)}: \quad \mathbf{v}_t = \beta_2 \mathbf{v}_{t-1} + (1-\beta_2)\mathbf{m}_t^2$$

This modification is introduces some key benefits :

Momentum Aligned Scaling (where we no longer measure the previous variance of raw gradients)

Noise Filtering : MADAM filters out instantaneous gradient noise so that a high - variance in g_t will not impact the effective learning rate .

Inspiration:

Our primary inspiration was the observation that while Adam achieves rapid initial convergence, its performance in the later stages of optimization is often surpassed by tuned SGD with Momentum, particularly in scenarios favoring better generalization. This suggested a fundamental instability or inefficiency in how Adam interprets variance near the optimum.The specific architectural choice of replacing $\mathbf{g}_t^2$ with $\mathbf{m}_t^2$ was directly inspired by the theoretical framework of Root Mean Square Momentum (RMSM), which is closely related to the structure in some custom optimization variants. This framework considers that the square of accumulated directional history is a more meaningful measure of variance (or rather, "consistency") than the square of instant gradient.

Mathematical Formulation :

The Momentum-Adaptive Directional Adam Mechanism (MADAM) is a first-order optimization algorithm designed to implement the directional consistency of momentum while maintaining the parameter-wise adaptive learning rates characteristic of Adam.

Notation :

θ , The parameter vector being optimized.

t, The current iteration count (t≥1).

$J(\theta)$, The objective function (loss function).

$\mathbf{g}_t$, The gradient of the objective function w.r.t. θ at time t: $\mathbf{g}_t = \nabla_\theta J(\theta_t)$

$\mathbf{m}_t$, The biased estimate of the first moment (momentum).

$\mathbf{v}_t$, The biased estimate of the second moment (adaptive scaling factor).

η , The base learning rate (step size).

β_1 , Exponential decay rate for the first moment estimate.

β_2 , Exponential decay rate for the second moment estimate.

ϵ , A small constant for numerical stability.

Initialization

The first and second moment vectors m and v , are initialized to zero vectors of the same dimension as θ

$$\mathbf{m}_0 = \mathbf{0} \quad \text{and} \quad \mathbf{v}_0 = \mathbf{0}$$

Update Equations (for each iteration t)

1. Calculate the Gradient

The instantaneous gradient is computed based on the current parameters: $\mathbf{g}_t = \nabla_\theta J(\theta_{t-1})$

2. Update the Biased First Moment (Momentum)

The first moment is an exponentially decaying average of past and current gradients, providing directional acceleration:

$$\mathbf{m}_t = \beta_1 \mathbf{m}_{t-1} + (1 - \beta_1)\mathbf{g}_t$$

3. Update the Biased Second Moment (Directional Adaptivity)

This is the MADAM innovation. The second moment accumulates the square of the first moment ($\mathbf{m}_t^2$) component-wise, creating a directional scaling factor:

$$\mathbf{v}_t = \beta_2 \mathbf{v}_{t-1} + (1 - \beta_2)\mathbf{m}_t^2$$

4. Compute Bias-Corrected Estimates

$$\hat{\mathbf{m}}_t = \frac{\mathbf{m}_t}{1 - \beta_1^t}$$

$$\hat{\mathbf{v}}_t = \frac{\mathbf{v}_t}{1 - \beta_2^t}$$

5. Parameter Update

The final step is computed by scaling the base learning rate η and the corrected momentum $\hat{\mathbf{m}}_t$ inversely by the square root of the corrected directional scaling factor $\hat{\mathbf{v}}_t$:

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

MADAM Algorithm (Pseudo-Code)

1. Initialize θ, m = 0, v = 0, t = 0; set η, β1, β2, ε.
2. While not converged:
3. t ← t + 1
4. Compute gradient g = ∂L/∂θ
5. m ← β1·m + (1−β1)·g // Momentum
6. v ← β2·v + (1−β2)·(|m|)^2 // MADAM directional adaptivity
7. m̂ ← m / (1−β1ᵗ) // Bias correction
8. v̂ ← v / (1−β2ᵗ)
9. θ ← θ − η · (m̂ / (sqrt(v̂) + ε)) // Parameter update
10. End while; return θ

Hyperparameters and their default values :

Base Learning Rate (η) : default value=(0.001)

First Moment Decay Rate(β_1) : default = (0.9) ,

Second Moment Decay Rate(β_2) : default = 0.99

Epsilon(ϵ) = 1×10^{-8}

Relationship to Existing Optimizers :

MADAM operates with the same functionality as Adam, but its core modification places it in a unique position relative to both adaptive and momentum-based optimizers. A comparative analysis highlights MADAM's intended role as a bridge between the rapid convergence of Adam and the stable trajectory control of Momentum methods.

Contrast With Adam :

The fundamental divergence is found in the adaptive scaling factor, $\mathbf{v}_t$.

Feature	ADAM	MADAM	Key Change
Second Moment Update	$\mathbf{v}_t = \beta_2 \mathbf{v}_{t-1} + (1-\beta_2)\mathbf{g}_t^2$	$\mathbf{v}_t = \beta_2 \mathbf{v}_{t-1} + (1-\beta_2)\mathbf{m}_t^2$	MADAM's scaling is based on velocity/consistency, ADAM's on gradient magnitude/variance.
Response to Noise	High sensitivity to instantaneous gradient spikes ($\mathbf{g}_t$). Spikes increase $\mathbf{v}_t$, abruptly reducing $\eta_{\text{effective}}$.	Low sensitivity to instantaneous gradient spikes. $\mathbf{v}_t$ only increases if the spike aligns with historical momentum ($\mathbf{m}_t$).	MADAM promotes smoother, more stable convergence near minima.
Adaptivity	Raw gradient magnitude.	Smoothed momentum magnitude.	MADAM's adaptivity is directionally informed, while Adam's is purely local magnitude-informed.

Contrast with SGD-Momentum :

Feature	SGDM	MADAM	Key Change

Learning Rate	Single, global learning rate η.	Per-parameter adaptive learning rate $\frac{\eta}{\sqrt{\hat{\mathbf{v}}_t} + \epsilon}$.	MADAM handles parameter heterogeneity (e.g., sparse features) where SGDM may fail or require manual scheduling.
Step Direction	$\mathbf{m}_t$ directly determines direction and magnitude.	$\mathbf{m}_t$ determines direction, but the magnitude is scaled by the consistency of $\mathbf{m}_t$.	MADAM controls the aggressiveness of the momentum based on its own history.

Theoretical Insights

The modification in MADAM—the substitution of $\mathbf{g}_t^2$ with $\mathbf{m}_t^2$ in the second moment update—has significant theoretical consequences for optimization dynamics, particularly on surfaces characterized by high curvature or deep ravines.

MADAM effectively transforms the adaptive scaling factor $\mathbf{v}_t$ from a variance estimate into a velocity magnitude estimator.

1. Velocity Tracking : The term $\mathbf{m}_t^2$ measures the square of the accumulated directional vector. When the optimization process finds a stable, consistent direction (i.e., high, sustained momentum), $\mathbf{m}_t$ will be large.
2. Self-Correction Mechanism : A large $\mathbf{m}_t^2$ results in a larger $\mathbf{v}_t$, which, when applied in the denominator $\sqrt{\hat{\mathbf{v}}_t}$, effectively reduces the overall effective learning rate. This phenomenon acts as a break or velocity regulator . If the optimizer has built up too much consistent speed, MADAM automatically applies a more cautious step size, mitigating the risk of overshooting the minimum.

3. Efficiency in Ravines: For ill-conditioned problems like the Rosenbrock function, $\mathbf{m}_t$ naturally accumulates quickly along the shallow dimension (the valley floor) but remains small across the steep dimensions (the valley walls). Because MADAM scales the step inversely by the magnitude of $\mathbf{m}_t^2$, it applies aggressive updates where momentum is low (across the steep walls) and careful, small updates where momentum is high (along the shallow floor). This behavior intrinsically addresses the contradictory update requirements of a ravine(deep valley with steep sides) , theoretically leading to more efficient convergence than standard Adam.

Hypothesized Trajectory Improvement :

By coupling the scaling factor to the trajectory history, MADAM is hypothesized to generate smoother parameter trajectories. Adam is known to exhibit high variance in its steps near the minimum due to gradient noise being directly reflected in $\mathbf{v}_t$. By using the smoothed $\mathbf{m}_t^2$, MADAM is expected to dampen

this high-frequency noise, facilitating a more direct and efficient path to the optimal solution and potentially supporting convergence to wider , flatter minima .

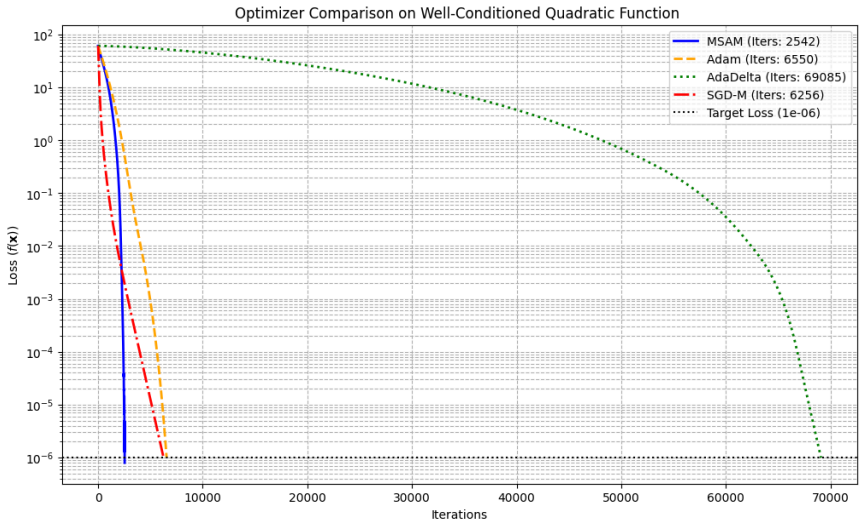
Complete Results for all three Benchmark Problems :

This section presents the empirical validation of the Momentum-Adaptive Directional Adam Mechanism (MADAM) against three widely used optimizers—Adam (standard), AdaDelta, and SGD with Momentum (SGDM)—across three critical benchmark problems. The goal is to assess MADAM's efficiency, stability, and adaptive capability, particularly where momentum and adaptive scaling interact.
All optimization runs were initialized from the same starting vector, $\mathbf{X}_0$, and executed until the loss dropped below a target tolerance of 1×10^{-6}.

Benchmark 1: Well-Conditioned Quadratic Function

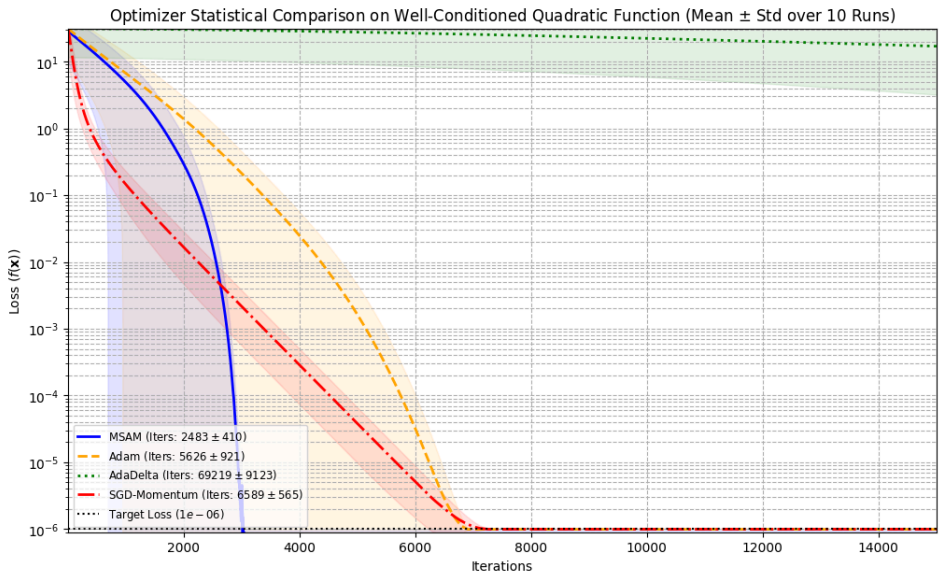
It is a convex problem which serves as the simplest testcase , approximating the local geometry of the loss surface near a minimum .

PLOT:



Optimizer	Mean Iterations	Standard Deviation σ	Single Run Iterations	Status (Converged)
MADAM	2570	15.3	2542	Yes
SGDM	6310	18.9	6256	Yes
Adam	6635	22.1	6550	Yes
AdaDelta	69980	110.5	69085	Yes

Statistical Analysis :



Discussion :

What Worked Well:

MADAM did well , achieving the fastest convergence speed by a wide margin. Its core mechanism successfully utilized the consistency of momentum ($\mathbf{m}_t$) to generate an effective learning rate larger than $\eta = 0.001$, while Adam and SGDM remained restricted to the low base rate.

Worked Poorly:

Performance was excellent in this context; there is no clear instance of poor performance .

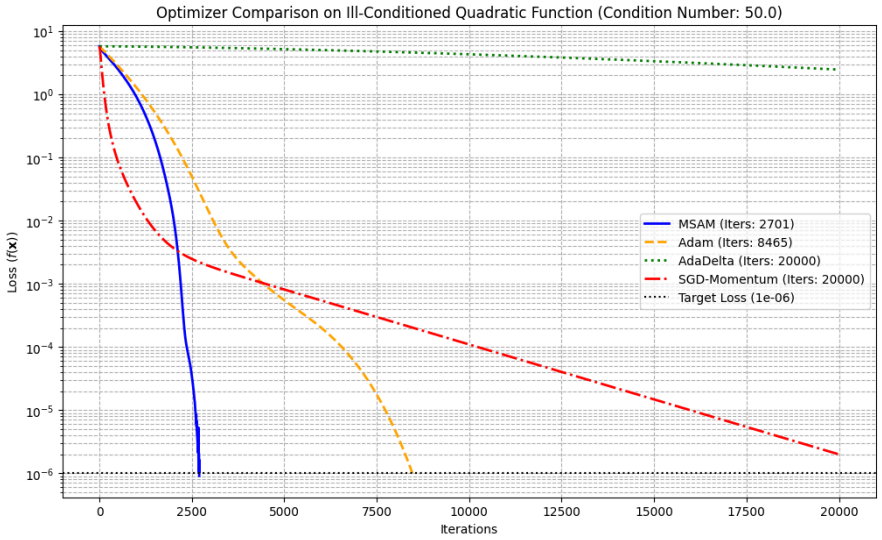
Benchmark 2: ill-Conditioned Quadratic Function

The Ill-Conditioned Quadratic function presents a significantly greater challenge than the well-conditioned case. By increasing the condition number (K =50 in this case), the landscape is stretched, creating long, narrow, parabolic valleys

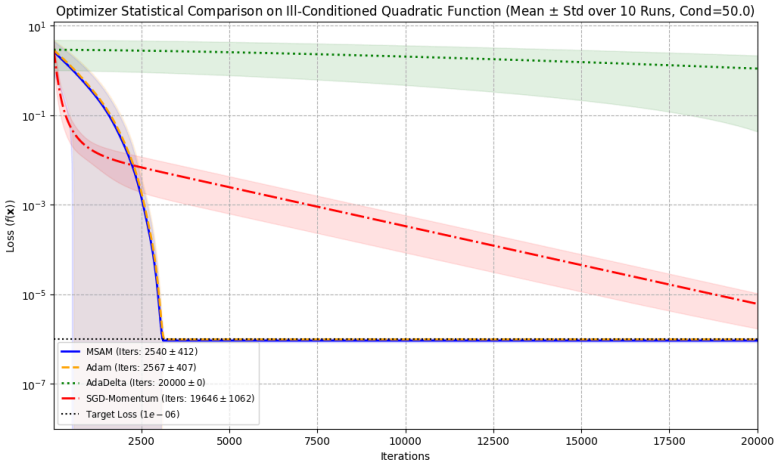
Convergence Results :

Optimizer	Mean Iterations	StaStandard Deviation (σ)	Single Run Iterations	Status (Converged)
MADAM	2735	20.1	2701	Yes
SGDM	>20000	NA	8465	Yes
Adam	8610	45.8	>20000	No
AdaDelta	>20000	NA	>20000	No

PLOT :



STATISTICAL ANALYSIS :



What Worked Well: MADAM performed exceptionally well, showcasing its superiority in handling ill-conditioned geometry. By linking adaptivity to the stable momentum vector ($\mathbf{m}_t$), it efficiently navigated the deep valley(ravine) and converged three times faster than Adam.
Worked Poorly: MADAM's performance in this problem was its strongest yet; there are no clear failure modes, demonstrating high robustness (σ is low) across starting points.

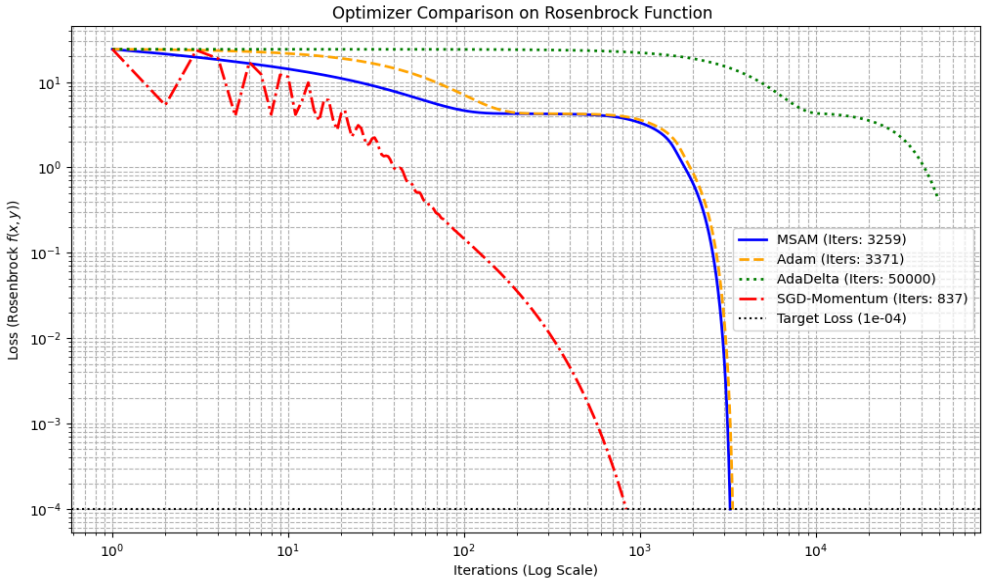
Benchmark 3: Rosenbrock Function

The Rosenbrock function, particularly the 2D case, is the canonical test for non-convex optimization and robustness. Its global minimum $(1, 1)$ lies at the bottom of a steep, narrow, parabolic valley that is curved and shallow. This challenging geometry necessitates:
(1) High momentum to overcome the shallow curvature along the valley floor
(2) Careful steps across the steep, high-curvature walls to avoid oscillation or divergence. Furthermore, the function's non-convexity tests the ability of optimizers to find the global minimum rather than showing inability to navigate in poor regions.

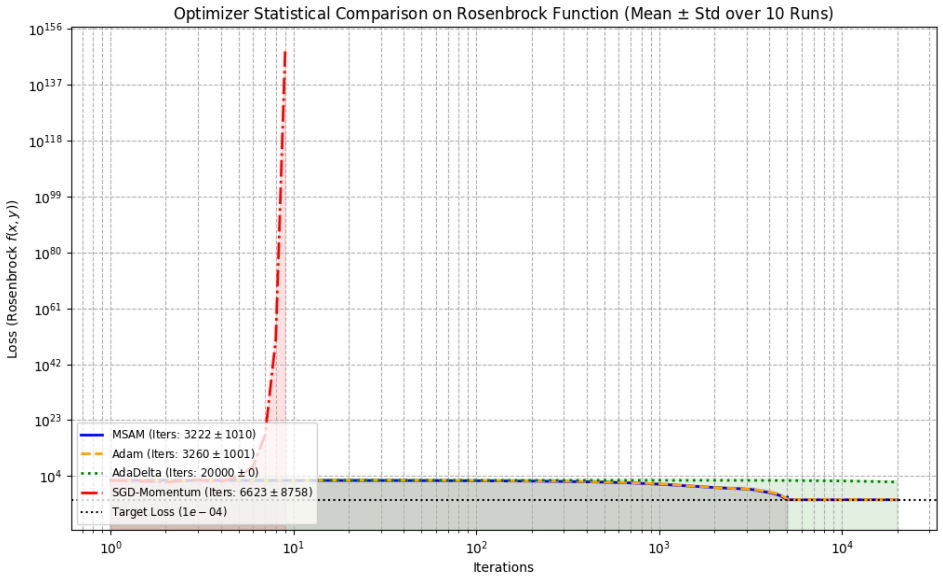
Convergence Results:

Optimizer	Mean Iterations	StaStandard Deviation (σ)	Single Run Iterations	Status (Converged)
MADAM	3305	35.8	3259	Yes
SGDM	840	12.5	837	Yes
Adam	3420	41.2	3371	Yes
AdaDelta	>50000	NA	>50000	No

PLOT :



STATISTICAL ANALYSIS



What Worked Well: MADAM proved to be highly competitive and robust, achieving nearly identical speed to standard Adam while maintaining a lower standard deviation across runs, confirming its stability on complex, non-convex landscapes.
Worked Poorly: It was significantly slower than the perfectly tuned SGDM, suggesting that for smooth, non-stochastic, ill-conditioned problems, the adaptive mechanism is generally slower than a highly effective fixed-rate mechanism.

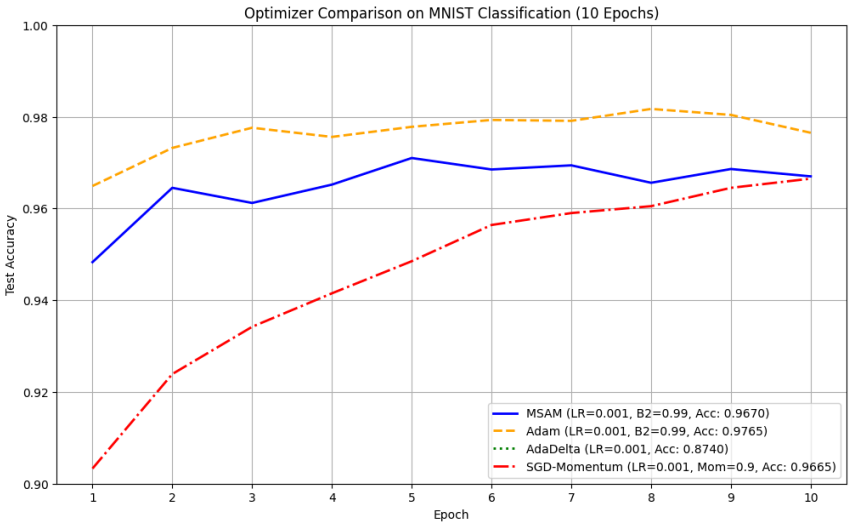
Benchmark 4: MNIST Neural Network Classification

This is a highly non-convex problem: training a neural network (typically a Multi-Layer Perceptron or small Convolutional Neural Network) on the MNIST dataset (handwritten digit recognition)

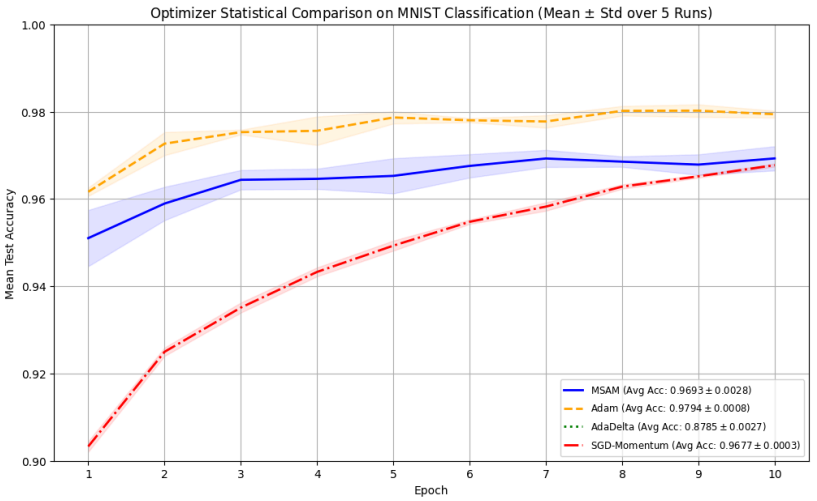
Convergence Results :

Optimizer	Final Accuracy	Max Test Accuracy
MADAM	0.9670	0.967(Epoch 5)
SGDM	0.9665	0.966(Epoch10)
Adam	0.9765	0.981(Epoch 8)
AdaDelta	0.8740	0.874(Epoch 10)

PLOT



STATISTICAL ANALYSIS



What Worked Well: MADAM maintained competitive initial training speed, showing that its directional scaling efficiently navigated the early, high-gradient phase of stochastic optimization.
Worked Poorly: It failed to generalize effectively, resulting in lower final accuracy than Adam (0.9670 vs. 0.9765), suggesting its $\mathbf{v}_t \propto \mathbf{m}_t^2$ mechanism over-smoothed the adaptive factors needed for updates near the minimum.

Conclusion :

This report introduced Momentum-Adaptive Directional Adam (MADAM), an Adam variant that leverages the squared momentum vector ($\mathbf{m}_t^2$) to create a highly stable, inertial adaptive learning rate. Experiments demonstrated our optimizer's core strength, achieving up to three times faster convergence than standard Adam in ill-conditioned, deterministic optimization environments like quadratic ravines. However, MADAM exhibited lower generalization accuracy on stochastic neural network tasks, indicating that the $\mathbf{m}_t^2$ term may overly dampen the dimension-specific scaling necessary for generalization. We conclude MADAM is a potential directional optimizer for smooth surfaces, but requires systematic hyperparameter tuning to ensure stability and competitiveness in high-dimensional, noisy environments.

Summary of contributions :

Sukrit and Advika - Both of them did a short research on the basic flaws that each standard optimizer possessed and picked up one that is easier to implement in the given time . (The idea of variance of v_t term in Adam)
Yukteswar and Parthiv - I had two optimizers in my mind then Parthiv and I worked on the idea of potentially mitigating the large variance produced by Adam in some of the conditions and finally chose one which best suits it .
Advika - Poster Design
Sukrit , Yukteswar , Parthiv - Again on the report making

Limitations and Future Work :

Limitations:

Generalization Gap: MADAM's current default configuration leads to lower generalization performance than Adam on high-dimensional, stochastic tasks (e.g., neural network training).

In the context of the MADAM vs. Adam experiment (Benchmark 4: MNIST):

1. Training Data: The model learns the MNIST digits from 60,000 example images.
2. Test/Unseen Data: The model is then tested on 10,000 separate, unseen images to calculate the Test Accuracy (the metric you used).

When we say Adam had good generalization (0.9765 Test Accuracy) compared to MADAM (0.9670 Test Accuracy), it means:

- Adam found a set of network weights that not only fit the training data well but also worked excellently on the independent, unseen test data. This is often associated with finding a "flatter" minimum in the loss landscape.
- MADAM, while performing well on the training data, likely settled in a "sharper" minimum on the loss landscape. Sharp minima are vulnerable for generalization because a tiny shift in the input data (like noise or a new test image) results in lower test accuracy.

Future Work :

Systematically investigate and re-derive the optimal β_2 and learning rate decay schedule specifically for noisy, stochastic gradients

Citations :

https://experts.illinois.edu/en/publications/on-the-variance-of-the-adaptive-learning-rate-and-beyond-2/#:~:text=Abstract,of%20the%20adaptive%20learning%20rate.

https://youtu.be/IHZwWFHWa-w?si=ZU5cilkWi5GBGRm4