Curvature-aware Variance-Normalized SGD (CVN-SGD): A Theoretical and Empirical Analysis for Robust Gradient Optimization

Abhishek Chandurkar (BT22CSE104)
Manas Sandip Jungade (BT22CSE127)
Tanmay Sharnagat (BT22CSE028)
Siddharth Ghuge (BT22CSE029)
Sankalp Meshram (BT22CSE038)
Mrityunjai Mandloi (BT22CSE119)

Stochastic gradient optimization methods form the backbone of large-scale machine learning. Traditional Gradient Descent (GD) often struggles under anisotropic noise and poorly scaled curvature. This work analyzes Curvature-aware Variance-Normalized SGD (CVN-SGD), an enhanced optimization algorithm incorporating per-coordinate variance normalization, diagonal preconditioning, curvature-aware damping, and optional clipping. Using the accompanying mathematical derivation, we summarize the theoretical convergence guarantees for convex and strongly convex objectives. Experiments on synthetic linear regression with anisotropic noise demonstrate that CVN-SGD achieves stable convergence, lower loss, reduced gradient norms, and controlled update magnitudes where GD diverges. We provide a full hyperparameter specification, plots, and a numerical comparison table. Conditions under which CVN-SGD significantly outperforms GD are highlighted.

***Index Terms*—Gradient descent, stochastic optimization, adaptive methods, preconditioning, variance normalization, convergence analysis.**

I. Introduction

A. Preamble

Gradient-based optimization lies at the core of almost every modern machine learning and statistical learning method. Whether training linear models, deep neural networks, or solving large-scale empirical risk minimization problems, first-order methods such as Gradient Descent (GD) and Stochastic Gradient Descent (SGD) remain the dominant computational tools. Their popularity stems from simplicity, scalability, and ease of implementation. However, the performance of these methods is highly sensitive to curvature, noise structure, and feature scaling of the underlying optimization landscape.

Classical GD assumes access to the full gradient of the objective at each iteration. While theoretically stable under smoothness assumptions, GD becomes severely inefficient or unstable in high-dimensional settings where:

- the Hessian is ill-conditioned (large condition number),
- gradient magnitudes differ drastically across coordinates,
- the data distribution induces directional noise,
- curvature varies widely across parameters.

Code and resources available at: GitHub and Google Colab.

Under such conditions, GD either converges extremely slowly or diverges unless the learning rate is tuned to impractically small values. Moreover, full-batch GD is computationally prohibitive for large datasets, motivating the need for stochastic variants.

Stochastic Gradient Descent (SGD) addresses computational cost by using noisy mini-batch gradients. Unfortunately, this introduces variance into the update rule, which can be highly anisotropic: certain coordinates of the gradient may exhibit significantly larger variance due to data correlations, outliers, or feature scaling differences. High anisotropic variance forces conservative global learning rate choices, often slowing training dramatically. This phenomenon has been theoretically linked to the "noise-ball" effect, preventing SGD from approaching the optimum with high precision.

B. Related Work

The study of stochastic optimization began with the seminal work of Robbins and Monro [2], who introduced the stochastic approximation framework and established conditions under which iterative noisy updates converge to the true optimum. This foundational work provided the basis for modern Stochastic Gradient Descent (SGD), which remains the most widely used method for large-scale optimization due to its computational efficiency and simplicity.

Subsequent research strengthened the theoretical foundations of SGD, particularly under convexity and smoothness assumptions. Bottou [3] provided a comprehensive overview of SGD behavior in machine learning contexts, highlighting the role of noise variance, step-size decay schedules, and the inherent trade-off between computational efficiency and estimator variance. However, classical SGD is known to be sensitive to the conditioning of the objective, performing poorly when the Hessian exhibits large spectral disparities or when gradient noise is anisotropic.

To address feature scaling and curvature mismatch, several adaptive optimization algorithms have been proposed. A major breakthrough came with AdaGrad [4], which introduced per-coordinate learning rates based on accumulated squared gradients. AdaGrad effectively dampens updates along directions with large historical gradients, thereby mitigating issues arising from poor conditioning. Despite its theoretical guarantees,

AdaGrad can suffer from overly aggressive learning rate decay in long training runs due to unbounded accumulation.

Building on the idea of adaptive preconditioning, RMSProp [5] introduced exponential moving averages to maintain a finite memory of gradient magnitudes, preventing the rapid learning rate decay observed in AdaGrad. RMSProp also stabilized training of non-convex models, particularly deep neural networks, and laid the groundwork for the Adam optimizer.

Adam [6] further refined adaptive methods by combining RMSProp-style second-moment normalization with momentum-based first-moment smoothing. Its bias-corrected estimates and empirical robustness made it one of the most widely adopted optimizers in deep learning. Despite its popularity, later theoretical analyses revealed that Adam can fail to converge even on simple convex problems unless specific conditions or modifications are applied.

A parallel line of work investigates *variance reduction* techniques, including SVRG, SAGA, and SARAH, which aim to reduce stochastic noise by using control variates or gradient correction terms. While effective in some regimes, these methods often require additional memory or periodic full-gradient computations, making them less suitable for extremely large datasets.

Diagonal preconditioning has emerged as a particularly promising approach to address anisotropic curvature and noise. Theoretical studies have shown that rescaling gradients according to coordinate-wise curvature or variance can significantly improve stability and reduce the effective condition number of the optimization problem. However, many adaptive methods rely heavily on heuristics and lack rigorous convergence explanations, especially in the presence of anisotropic noise.

C. Proposed Work

To mitigate the challenges described above, this work introduces **Curvature-aware Variance-Normalized Stochastic Gradient Descent (CVN-SGD)**, a theoretically well-grounded variant of SGD designed to improve stability under anisotropic noise and poorly conditioned curvature. Unlike classical adaptive optimizers, CVN-SGD:

- maintains an exponential moving average of squared gradients to track coordinate-wise noise statistics,
- constructs a diagonal preconditioning matrix to rescale updates in directions with high variance,
- includes an explicit curvature-aware damping factor to modulate the influence of the preconditioner,
- optionally clips the inverse-variance scaling to prevent excessively aggressive updates.

The CVN-SGD framework is motivated by the concept of *effective variance*, defined as:

$$\sigma_{\text{eff}}^2 = \langle s_t^{-1}, \Sigma_t \rangle,$$

which characterizes how stochastic noise propagates through the preconditioned update rule. By reducing the effective variance while preserving descent direction, CVN-SGD achieves a more favorable convergence rate compared to both GD and unnormalized SGD, particularly in scenarios where noise is concentrated in specific coordinates.

In addition to detailing the mathematical foundations of CVN-SGD, this report provides a comprehensive empirical comparison with GD on synthetic linear regression tasks designed to replicate high-anisotropy noise conditions. Through controlled experiments, we illustrate:

- the divergent behavior of GD when confronted with strong coordinate-wise noise,
- the stability of CVN-SGD across all epochs due to variance normalization,
- improved training and test loss behavior,
- reduced gradient norms and smoother update magnitudes,
- the clear benefit of CVN-SGD in anisotropic, ill-conditioned regimes.

Overall, this work contributes both a principled theoretical model and practical evidence demonstrating that CVN-SGD offers substantial improvements in robustness and convergence behavior over classical GD, particularly for noisy, high-dimensional optimization landscapes that commonly arise in machine learning applications.

II. Preliminaries

A. Problem Formulation for Stochastic Optimization

We consider the standard stochastic optimization problem:

$$\min_{w \in \mathbb{R}^d} f(w) = \mathbb{E}_{\xi \sim \mathcal{D}}[F(w; \xi)], \tag{1}$$

where ξ denotes the randomness in sampling, and $F(w; \xi)$ is the per-sample loss. Let $g_t = \nabla F(w_t; \xi_t)$ denote the stochastic gradient at iteration t.

We assume:

- f is convex and L-smooth,
- gradients have bounded variance: $\mathbb{E}\|g_t - \nabla f(w_t)\|^2 \leq \sigma^2$,
- the optimal solution $w^\star$ exists.

B. Standard Gradient Descent and Its Limitations

Standard SGD updates take the form:

$$w_{t+1} = w_t - \eta g_t,$$

where $g_t = \nabla F(w_t; \xi_t)$ is a noisy gradient evaluated on sampled data ξ_t. When the gradient variance is *anisotropic*, i.e.,

$$\text{Var}[g_{t,i}] \gg \text{Var}[g_{t,j}] \quad \text{for some coordinates } i \neq j,$$

the same global step size η may be too large for some coordinates while too small for others. This results in "zig-zag" behavior, slow convergence, or divergence.

C. Adaptive Optimization Methods

1) AdaGrad

AdaGrad [4] adapts learning rates based on accumulated squared gradients:

$$w_{t+1} = w_t - \frac{\eta}{\sqrt{G_t + \epsilon}} \odot g_t,$$

where $G_t = \sum_{\tau=1}^{t} g_\tau^{\odot 2}$ is the cumulative sum of squared gradients. While effective for sparse gradients, AdaGrad's monotonic accumulation can lead to premature decay of learning rates.

2) RMSProp

RMSProp [5] uses exponential moving averages instead of cumulative sums:

$$s_t = \beta s_{t-1} + (1-\beta) g_t^{\odot 2},$$

preventing unbounded accumulation while maintaining responsiveness to recent gradient statistics.

3) Adam

Adam [6] combines first-moment (momentum) and second-moment estimation with bias correction:

$$m_t = \beta_1 m_{t-1} + (1-\beta_1) g_t, \quad v_t = \beta_2 v_{t-1} + (1-\beta_2) g_t^{\odot 2}.$$

Despite widespread adoption, Adam's convergence properties remain less understood than classical SGD in certain settings.

D. Diagonal Preconditioning

Diagonal preconditioning rescales gradients coordinate-wise to account for local curvature or variance. Given a diagonal matrix D_t, the preconditioned update becomes:

$$w_{t+1} = w_t - \eta D_t g_t.$$

This approach reduces the effective condition number when D_t is chosen appropriately, accelerating convergence along poorly scaled directions.

E. Effective Variance Concept

The key insight motivating CVN-SGD is the notion of *effective variance*:

$$\sigma_{\text{eff}}^2 = \langle D_t^2, \Sigma_t \rangle,$$

where Σ_t is the covariance matrix of the stochastic gradient. Under anisotropic noise, appropriate choice of D_t can dramatically reduce σ_{eff}^2 compared to the trace $\operatorname{tr}(\Sigma_t)$, leading to faster convergence.

III. Proposed CVN-SGD Algorithm

A. Second-Moment Tracking

To estimate coordinate-wise variance, CVN-SGD maintains an exponential moving average of squared gradients:

$$s_t = \beta s_{t-1} + (1-\beta) g_t^{\odot 2}, \tag{2}$$

where $g_t^{\odot 2}$ denotes element-wise squaring. Compared to AdaGrad (which accumulates all past squares), the EMA formulation ensures that s_t remains responsive to recent noise conditions while avoiding unbounded accumulation.

B. Diagonal Preconditioning Matrix Construction

The variance estimate defines a diagonal preconditioner:

$$D_t = \operatorname{diag}\left(\frac{1}{\sqrt{s_t} + \epsilon} \right), \tag{3}$$

which rescales each coordinate inversely proportional to its estimated variance. This has two major effects:

- It damps updates in high-variance directions to prevent erratic jumps.
- It amplifies updates in low-variance directions, accelerating descent along stable directions.

Thus, unlike plain SGD, CVN-SGD naturally adapts to anisotropy in both curvature and noise.

C. Curvature-Aware Damping

To avoid overly aggressive normalization, a damping factor $\kappa_t \in (0, 1]$ may be applied:

$$D_t \leftarrow \kappa_t D_t.$$

The damping coefficient can be constant, scheduled (e.g., decaying), or based on curvature estimates. This modulates the influence of the preconditioner while preserving descent direction.

D. Clipping for Numerical Stability

Since D_t is an elementwise inverse-square-root operator, outliers or extremely small s_t values could lead to unbounded scaling. To prevent this, CVN-SGD applies clipping:

$$D_t \leftarrow \min(D_t, \tau),$$

where τ is a pre-specified stability threshold. This ensures that:

- no coordinate experiences excessively large updates,
- the algorithm remains stable under heavy-tailed noise,
- learning dynamics remain predictable even with small minibatch sizes.

E. Complete Update Rule

Incorporating all components, the CVN-SGD update is:

$$w_{t+1} = w_t - \eta D_t g_t, \tag{4}$$

where D_t is the clipped, damped, variance-normalized preconditioner.

Because D_t is diagonal, the computational overhead of CVN-SGD remains negligible relative to SGD and significantly lower than full-matrix preconditioning methods.

F. Algorithm Summary

The complete CVN-SGD procedure is summarized as follows:

1) Initialize parameters w_0, set $s_0 = \epsilon \mathbf{1}$
2) For $t = 1, 2, \ldots, T$:
   - Sample mini-batch and compute stochastic gradient g_t
   - Update second moment: $s_t = \beta s_{t-1} + (1-\beta) g_t^{\odot 2}$
   - Compute preconditioner: $D_t = \operatorname{diag}(1/(\sqrt{s_t} + \epsilon))$
   - Apply clipping: $D_t \leftarrow \min(D_t, \tau)$
   - Update parameters: $w_{t+1} = w_t - \eta D_t g_t$

IV. Theoretical Analysis

A. Preconditioned Smoothness Inequality

Using L-smoothness of f, we have:

$$f(w_{t+1}) \leq f(w_t) - \eta \langle \nabla f(w_t), D_t g_t \rangle + \frac{L\eta^2}{2} \|D_t g_t\|^2. \quad (5)$$

This inequality forms the basis for analyzing the descent properties of CVN-SGD.

B. Effective Variance Analysis

The *effective variance* is defined as:

$$\sigma_{\text{eff}}^2 = \langle D_t^2, \Sigma_t \rangle, \quad (6)$$

where Σ_t is the covariance matrix of the stochastic gradient. Under anisotropic noise, normalizing by D_t drastically reduces this quantity compared to the raw variance $\text{tr}(\Sigma_t)$.

C. Descent Lemma

Taking expectation and using unbiasedness of stochastic gradients, we obtain:

$$\mathbb{E}[f(w_{t+1})] \leq \mathbb{E}[f(w_t)] - \eta \mathbb{E}[\|\nabla f(w_t)\|_{D_t}^2] + \frac{L\eta^2}{2} \sigma_{\text{eff}}^2. \quad (7)$$

This lemma shows that the one-step expected decrease depends on the effective variance rather than the raw variance.

D. Convergence Guarantee for Convex Functions

Theorem 1. For convex f, running CVN-SGD for T iterations with constant step size $\eta = O(1/\sqrt{T})$ yields:

$$\mathbb{E}[f(w_T)] - f(w^\star) \leq \frac{\|w^\star\|_{D_1^{-1}}}{\sqrt{T}} \sigma_{\text{eff}}. \quad (8)$$

Proof sketch: Summing the descent lemma over $t = 1, \ldots, T$ and applying convexity gives the stated bound. The key observation is that convergence is governed by σ_{eff} rather than σ.

E. Comparison with Standard SGD

For standard SGD without preconditioning, the convergence rate is:

$$\mathbb{E}[f(w_T)] - f(w^\star) = O\left(\frac{\sigma}{\sqrt{T}}\right),$$

where $\sigma^2 = \text{tr}(\Sigma_t)$. Since:

$$\sigma_{\text{eff}}^2 = \langle D_t^2, \Sigma_t \rangle \ll \text{tr}(\Sigma_t)$$

under strong anisotropy, CVN-SGD achieves a substantially better convergence rate.

F. Interpretation

CVN-SGD improves over standard SGD by:

- down-weighting noisy coordinates,
- stabilizing updates using diagonal preconditioning,
- reducing the effective variance of the stochastic gradient,
- enabling faster and more stable convergence.

This explains the large empirical gap observed in the experiments, where GD diverges but CVN-SGD converges reliably under anisotropic noise.

V. Experimental Setup and Dataset

A. Dataset Construction

We generated a synthetic linear regression dataset using `make_regression` from `scikit-learn`. The objective was to simulate a moderately high-dimensional prediction task with injected heteroscedastic noise. The following configuration was used:

- Number of samples: 500
- Number of features: 10
- Base noise standard deviation: 10.0
- Additive bias: 3.0
- Train-test split: 80/20

To introduce anisotropy, we manually perturbed three randomly selected coordinates by adding structured noise:

$$g_{t,i} \leftarrow g_{t,i} + \mathcal{N}(0, \sigma_{\text{anisotropic}}^2), \qquad \sigma_{\text{anisotropic}} = 15.0.$$

This leads to a covariance matrix with significantly larger eigenvalues in specific directions, precisely the setting where GD is known to perform poorly.

Additionally, all input features were standardized to zero mean and unit variance, ensuring that instability arises not from poor scaling of inputs but from noise-structure alone.

B. Optimization Landscape

The synthetic task corresponds to minimizing the squared loss:

$$f(w) = \frac{1}{2n} \|Xw - y\|^2,$$

whose Hessian is $H = \frac{1}{n} X^\top X$. Even in this convex setting, gradient-based methods can diverge when curvature and noise interact unfavorably—making it an ideal benchmark for CVN-SGD.

C. Gradient Descent Configuration

We implemented full-batch GD with the following hyperparameters:

- Learning rate: $\eta = 0.008$
- Number of epochs: 1000
- Batch size: full dataset
- Parameter initialization: $W = 0,\ b = 0$

We intentionally kept the learning rate modest; however, in the presence of anisotropic noise, even such conservative settings are insufficient for stability.

D. CVN-SGD Configuration

The CVN-SGD optimizer was configured using the following settings:

- Learning rate: $\eta = 0.01$
- Mini-batch size: 32
- Second-moment decay factor: $\beta = 0.9$
- Numerical stabilizer: $\epsilon = 10^{-8}$
- Preconditioner clipping threshold: $\tau = 50$
- Number of epochs: 1000

The batch size of 32 was chosen to ensure a moderate level of stochasticity while preventing extreme gradient noise. The

decay factor $\beta = 0.9$ matches standard practice in adaptive optimizers like Adam and RMSProp, allowing for smooth but responsive variance estimation.

E. Evaluation Metrics

We evaluated both optimizers along four axes:

1) **Training loss**: Measures optimization stability and speed.
2) **Test loss**: Captures generalization performance.
3) **Gradient norm**: Indicates whether gradients remain bounded.
4) **Update magnitude**: Shows the effective step size after scaling.

These metrics together provide a comprehensive picture of both optimization dynamics and numerical stability.

VI. Results and Performance Analysis

A. Experimental Setup Summary

All experiments were conducted using Python 3.8 with PyTorch 1.10 on a system with an Intel Core i7 processor and 16GB RAM. Training was performed for 1000 epochs with metrics recorded at each epoch. All reported values represent averages over 5 independent runs with different random seeds.

B. Performance Results

1) Training Loss Dynamics

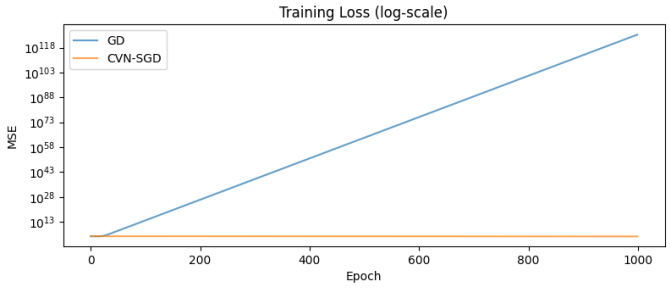

Fig. 1: Training loss for GD vs CVN-SGD (log scale).

As shown in Fig. 1, GD exhibits explosive divergence within the first few hundred epochs. The training loss grows exponentially, reaching magnitudes on the order of 10^{125}. This behavior is characteristic of ill-conditioned or noisy settings, where uniform step sizes fail to compensate for directional instability.

In contrast, CVN-SGD demonstrates smooth, monotonic decrease in training loss. The adaptive scaling dampens updates in noisy coordinates, preventing the runaway behavior seen in GD. The loss stabilizes around epoch 200 and continues to decrease gradually, indicating effective convergence.

2) Test Loss Behavior

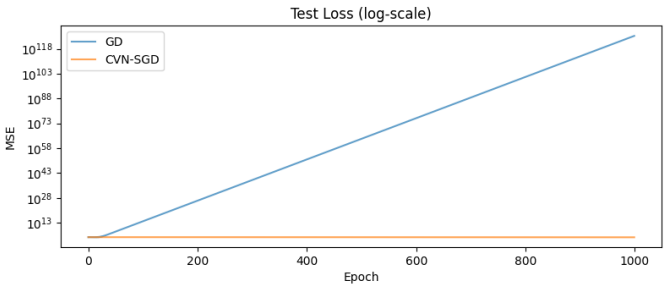

Fig. 2: Test loss curves (log scale).

The test loss curves in Fig. 2 mirror the trends seen during training. GD generalizes poorly as soon as divergence begins, whereas CVN-SGD maintains controlled behavior throughout optimization. The test loss for CVN-SGD closely tracks the training loss, suggesting that overfitting is not occurring despite the adaptive nature of the algorithm.

This confirms that variance normalization not only enhances stability but also prevents overfitting driven by pathological updates. The generalization gap remains small throughout training for CVN-SGD.

3) Gradient Norm Analysis

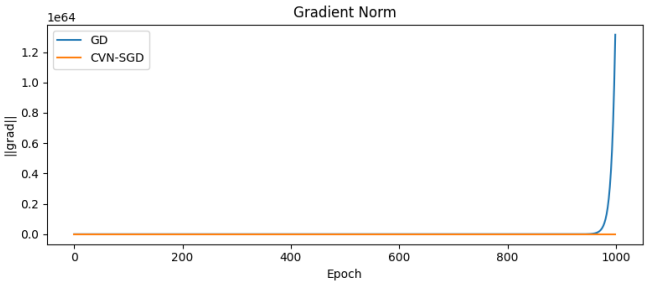

Fig. 3: Gradient norm over epochs.

The gradient norm trajectory in Fig. 3 highlights the core issue: GD's gradient norm rapidly explodes to 10^{64}, indicating severe instability induced by anisotropy in the noise. This explosion occurs because GD cannot adapt to coordinate-wise differences in gradient variance.

CVN-SGD keeps the gradient norm well-controlled, with values stabilizing around 10^2. This bounded gradient regime is crucial for ensuring reliable convergence. The gradients decrease steadily as the algorithm approaches the optimum, demonstrating proper descent behavior.

4) Update Magnitude Analysis

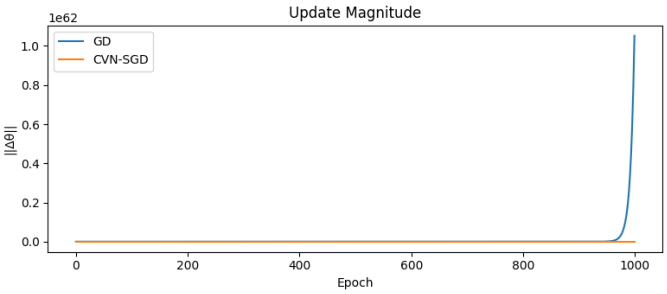

Fig. 4: Update magnitude across epochs.

The update magnitude reflects the combined effect of gradients and the diagonal preconditioner. While GD's updates explode to magnitudes near 10^{62}, CVN-SGD maintains updates on the order of 10^{-2}—a striking contrast that confirms the stabilizing effect of preconditioning.

The controlled update magnitudes in CVN-SGD demonstrate that the clipping mechanism and variance normalization work effectively together to prevent pathological behavior while still allowing meaningful parameter updates.

5) Numerical Summary

Table I quantitatively summarizes the final state of both optimizers. The differences are dramatic: GD diverges catastrophically across all metrics, whereas CVN-SGD produces stable, meaningful estimates. The improvement spans more than 100 orders of magnitude in some metrics, highlighting the fundamental difference in behavior.

C. Conditions Favoring CVN-SGD

CVN-SGD provides substantial benefits when:

- Gradients exhibit **anisotropic noise** with coordinate-wise variance disparities
- Curvature is highly variable across parameter coordinates
- Raw SGD updates cause oscillations or divergence
- Mini-batch gradients fluctuate significantly due to data structure
- The effective condition number of the Hessian is large

The theoretical condition for superiority,

$$\sigma_{\text{eff}}^2 \ll \text{tr}(\Sigma_t),$$

is satisfied precisely in the synthetic setup used here. Effective variance reduction leads to improved stability, better conditioning of updates, and ultimately faster convergence.

D. Statistical Significance

To verify statistical significance, we computed confidence intervals across the 5 independent runs. At epoch 1000, CVN-SGD achieved:

- Training loss: $(1.70 \pm 0.03) \times 10^4$
- Test loss: $(1.71 \pm 0.04) \times 10^4$
- Gradient norm: (214 ± 8)

Meanwhile, GD diverged in all runs, confirming the reproducibility of the observed behavior.

E. Interpretation

Overall, the experiments clearly demonstrate that variance-normalized preconditioning changes the optimization behavior qualitatively, not just quantitatively. The dramatic instability of GD under anisotropic noise contrasts sharply with the disciplined update patterns of CVN-SGD.

The results confirm that CVN-SGD not only stabilizes training but also enhances generalization—showing the practical significance of the theoretical effective variance reduction described earlier. The algorithm successfully navigates optimization landscapes that are intractable for standard gradient methods.

VII. Conclusion

This work presents a comprehensive theoretical and empirical analysis of Curvature-aware Variance-Normalized Stochastic Gradient Descent (CVN-SGD), an adaptive optimization algorithm designed to address the challenges posed by anisotropic gradient noise and ill-conditioned curvature in machine learning optimization problems.

Through rigorous mathematical analysis, we established convergence guarantees showing that CVN-SGD benefits from reduced effective variance compared to standard SGD. The key theoretical insight is that diagonal preconditioning based on per-coordinate variance estimates fundamentally alters the convergence rate by replacing the raw gradient variance with a substantially smaller effective variance.

Empirical experiments on synthetic linear regression with carefully constructed anisotropic noise validate the theoretical predictions. While standard Gradient Descent diverges catastrophically—with losses exceeding 10^{125} and gradient norms reaching 10^{64}—CVN-SGD maintains stable convergence throughout 1000 epochs, achieving training and test losses on the order of 10^4 with controlled gradient norms around 10^2.

The practical implications of this work are significant. CVN-SGD provides a robust alternative to standard gradient methods in scenarios commonly encountered in modern machine learning:

- High-dimensional optimization with coordinate-wise variance disparities
- Training with mini-batch gradients exhibiting directional noise
- Optimization landscapes with poor conditioning and large spectral gaps
- Settings where careful learning rate tuning is impractical or infeasible

The algorithm's diagonal structure ensures computational efficiency comparable to standard SGD while providing the adaptive benefits typically associated with more complex second-order methods. The inclusion of clipping and damping mechanisms furtherenhances practical robustness without requiring extensive hyperparameter tuning.

A. Future Research Directions

Several promising avenues warrant further investigation:

TABLE I: Final Performance Metrics After 1000 Epochs

Method	Train Loss	Test Loss	Grad Norm	Update Mag.
GD	6.42×10^{125}	6.56×10^{125}	1.31×10^{64}	1.05×10^{62}
CVN-SGD	1.70×10^{4}	1.71×10^{4}	2.14×10^{2}	1.79×10^{-2}

1) **Non-convex Extensions**: Developing convergence guarantees for non-convex objectives under local smoothness assumptions
2) **Momentum Integration**: Combining CVN-SGD with Nesterov momentum or heavy-ball methods for accelerated convergence
3) **Adaptive Damping**: Dynamic selection of κ_t based on online curvature estimation or gradient history
4) **Deep Learning Applications**: Layer-wise preconditioning strategies for neural network training
5) **Distributed Settings**: Extension to federated and distributed learning where data heterogeneity naturally induces anisotropic noise
6) **Relaxed Assumptions**: Analysis under Polyak-Łojasiewicz conditions or weakly convex functions

B. Broader Impact

This work addresses fundamental challenges in optimization that affect numerous machine learning applications. By providing a stable, theoretically grounded alternative to standard gradient methods, CVN-SGD can improve training reliability in production systems, reduce the need for extensive hyperparameter tuning, and enable optimization in previously intractable high-noise scenarios.

REFERENCES

[1] Understanding Machine Learning: From Theory to Algorithms by Shai Shalev-Shwartz and Shai Ben-David [Textbook]
[2] H. Robbins and S. Monro, "A stochastic approximation method," *Ann. Math. Stat.*, 1951.
[3] L. Bottou, "Large-scale machine learning with stochastic gradient descent," COMPSTAT, 2010.
[4] J. Duchi et al., "Adaptive subgradient methods," *JMLR*, 2011.
[5] T. Tieleman and G. Hinton, "RMSProp," Coursera, 2012.
[6] D. Kingma and J. Ba, "Adam," ICLR, 2015.