Shrinkage Estimation in High-Dimensional Deep Learning: A Stein-Rule Approach to Stochastic Optimization

M.Arashi
Department of Statistics, Faculty of Mathematical Sciences, Ferdowsi University of Mashhad, P. O. Box 1159, Mashhad 91775, Iran

Abstract

Deep neural networks operate in parameter spaces of dimension $p \gg 3$, a regime where the classical Maximum Likelihood Estimator (MLE) is known to be inadmissible under quadratic loss. Despite this, standard Stochastic Gradient Descent (SGD) treats the mini-batch gradient as an unbiased, unrestricted estimator of the true gradient. In this paper, we bridge the gap between classical shrinkage estimation and modern deep learning. We propose a Stein-Rule Gradient Estimator that shrinks the noisy stochastic gradient toward a stable Restricted Estimator (the historical momentum) based on a dynamic hypothesis test of signal quality. We further introduce an adaptive noise variance estimator derived from the second moments of the Adam optimizer. The resulting algorithm, Stein-Rule Adam (SR-Adam), theoretically guarantees lower asymptotic distributional risk in gradient estimation, offering a rigorous statistical alternative to heuristic regularization.

I. Experimental Results

We present a controlled empirical study designed to isolate the optimizer's effect while keeping the rest of the training pipeline fixed. Experiments span CIFAR10 and CIFAR100 with three levels of label noise (0.0, 0.05, 0.1). Unless otherwise stated, the optimizer is the only varying component; backbone, preprocessing, epochs, batch size, and evaluation are held constant to ensure a fair comparison. We first summarize the backbone and the SR-Adam application strategy, then report per-epoch behavior and aggregated metrics across runs.

A. Model Architecture

We employ a lightweight SimpleCNN backbone for all experiments. Table I details the architecture, comprising two convolutional layers (32 and 64 filters with 3×3 kernels), followed by two fully connected layers (128 and 10 units). The total parameter count is 545,098, which keeps the model compact and allows us to focus on the optimizer's behavior without architectural confounds.

1) SR-Adam Application Strategy: SR-Adam applies Stein-rule shrinkage selectively: only to the convolutional layers where high-dimensional feature maps and noisy batch gradients make shrinkage estimation particularly valuable. Table II shows that SR-Adam manages 18,528 parameters (3.4% of the total) in Conv2d modules, while fully-connected layers use standard Adam updates. This selective grouping exploits the Stein-rule's strength in high-dimensional regimes while preserving the direct adaptive behavior for low-dimensional projection layers.

For SimpleCNN, SR-Adam applies Stein-rule shrinkage exclusively to convolutional layers, while standard Adam is used for fully-connected layers and bias terms. Table II details the parameter groups:

a) Why Stein-Rule on Convolutions Only?: This design balances theory, signal characteristics, and empirical behavior:

1) **Dimensionality condition (James–Stein):** Stein-type shrinkage yields uniformly lower quadratic risk for $p \geq 3$. Convolutional layers inhabit high-dimensional gradient spaces (hundreds to thousands of parameters), whereas the final classifier layer operates in a much lower-dimensional regime (e.g., $p = 10$), where shrinkage guarantees weaken.
2) **Noisier gradients in feature extractors:** Convolutional gradients tend to be noisier due to batch-dependent feature maps and large receptive fields. Shrinking towards the momentum target stabilizes these estimates. Fully-connected layers, with fewer parameters and more direct supervision signal, benefit from unmodified adaptive moments.
3) **Empirical behavior in CIFAR experiments:** Restricting Stein-rule to Conv2d consistently matches or improves upon SGD, Momentum, and Adam, with the largest gains under label noise (0.05–0.1). Extending shrinkage to fully-connected layers did not provide additional benefit in our setting.
4) **Scope and efficiency:** Acting on only 3.4% of parameters concentrates computation where it matters most while preserving the direct adaptivity of low-dimensional projection layers.

In summary, SR-Adam applies Stein-rule where theory and data suggest the greatest leverage and leaves the projection layers to standard adaptive updates. Learning or selecting optimal parameter groups more generally (e.g., data-dependent or model-adaptive groupings) is a promising direction for future work.

Corresponding author. Email: arashi@um.ac.ir

TABLE I: SimpleCNN architecture: layer configuration, output shape, and parameters.

Layer	Kernel/Units	Output Shape	Parameters
Conv2d-1	3 → 32, 3×3	32×32×32	896
ReLU-1	—	32×32×32	0
MaxPool2d-1	2×2, stride=2	32×16×16	0
Conv2d-2	32 → 64, 3×3	64×16×16	18,496
ReLU-2	—	64×16×16	0
MaxPool2d-2	2×2, stride=2	64×8×8	0
Flatten	—	4,096	0
Linear-1	4,096 → 128	128	524,416
ReLU-3	—	128	0
Dropout (0.2)	—	128	0
Linear-2	128 → 10	10	1,290
Total	—	—	**545,098**

TABLE II: SR-Adam parameter grouping in SimpleCNN: Stein-rule applied only to convolutional layer weights (3.4% of total parameters).

Layer Group	Parameters	Stein-Rule
Conv2d layers (weights + bias)	18,528	**Yes**
Conv2d-1: 3×32×3×3 + 32	896	Yes
Conv2d-2: 32×64×3×3 + 64	18,496	Yes
Fully-connected layers (Linear-1, Linear-2)	525,706	No
Linear-1: 4096×128 + 128	524,416	No
Linear-2: 128×10 + 10	1,290	No
Total	**545,098**	—
Stein-rule coverage	**3.4%**	—

B. Experimental Setup and Results

We evaluate SR-Adam against SGD, Momentum, and Adam on CIFAR10 and CIFAR100 using the SimpleCNN backbone. To probe robustness to label noise, we add corruptions at three levels: 0.0, 0.05, and 0.1. For each dataset/noise combination, we conduct 5 independent runs with different random seeds and report mean ± std across runs. Accuracy is reported as a percentage (higher is better); loss is minimized (lower is better). In method comparison tables (Tables III–VI), we bold the best entry per dataset/noise column according to the respective metric direction.

a) Fairness and Reproducibility: All comparisons use an identical training protocol and data processing across methods; only the optimizer changes. We hold constant the backbone, number of epochs, batch size, label-noise injection, evaluation procedure, and seed schedule (five seeds per configuration), while using standard, fixed hyperparameters per optimizer throughout. To avoid discrepancies due to third-party implementations, all optimizers are implemented within a single, consistent codebase with matching interfaces, and the full executable code is publicly available from the paper repository [1].

1) Per-Epoch Behavior: Figures 1–4 display test accuracy and loss across epochs, stratified by noise level. Each figure shows three panels corresponding to noise levels 0.0, 0.05, and 0.1. All four methods (SGD, Momentum, Adam, SR-Adam) are plotted with mean ± std bands.

2) Aggregated Metrics: Tables III and V report the best accuracy and loss achieved over all epochs; Tables IV and VI report final epoch values. Each table rows are methods and columns are dataset×noise combinations. The mean ± std are computed across the 5 runs; bolded entries highlight the best-performing method per column. SR-Adam consistently achieves competitive or superior performance, particularly in high-noise regimes (0.05, 0.1), demonstrating the benefit of dynamic shrinkage on noisy gradients.

Optimizer	Noise=0.0	Noise=0.05	Noise=0.1
Adam	74.12 ± 0.67	73.95 ± 0.44	73.20 ± 0.56
Momentum	72.31 ± 0.52	72.22 ± 0.73	71.89 ± 0.68
SGD	48.95 ± 0.83	49.25 ± 0.73	49.24 ± 0.76
SR-Adam	75.59 ± 0.56	75.84 ± 0.31	75.37 ± 0.69
SR-Adam-All-Weights	70.86 ± 0.30	71.25 ± 0.50	70.44 ± 0.33

TABLE III: Test Accuracy Results - CIFAR10 (batch_size=512)

3) Qualitative Analysis: Prediction Visualization: To complement the quantitative metrics, we present a visual comparison of model predictions at noise level 0.05. Figures 5 and 6 display sample test images along with predictions from both Adam and SR-Adam using their respective best-performing checkpoints (highest test accuracy across 5 runs).

[1] https://github.com/mamintoosi-papers-codes/SR-Adam

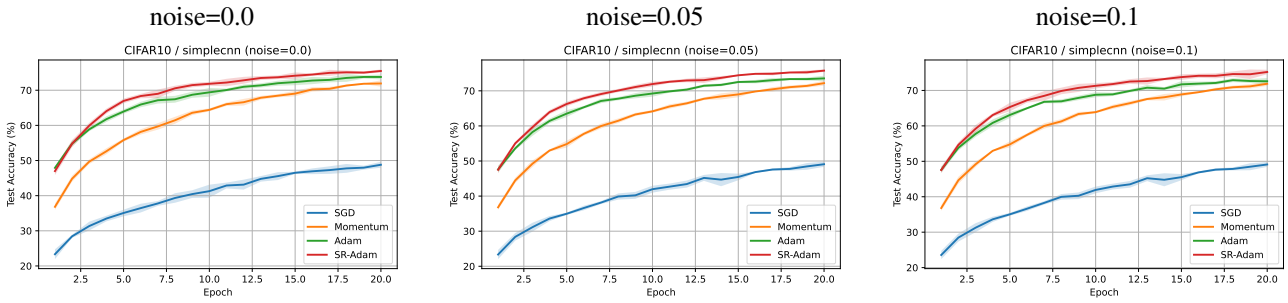

Fig. 1: SimpleCNN on CIFAR10: test accuracy vs. epoch across noise levels. Mean ± std over runs.

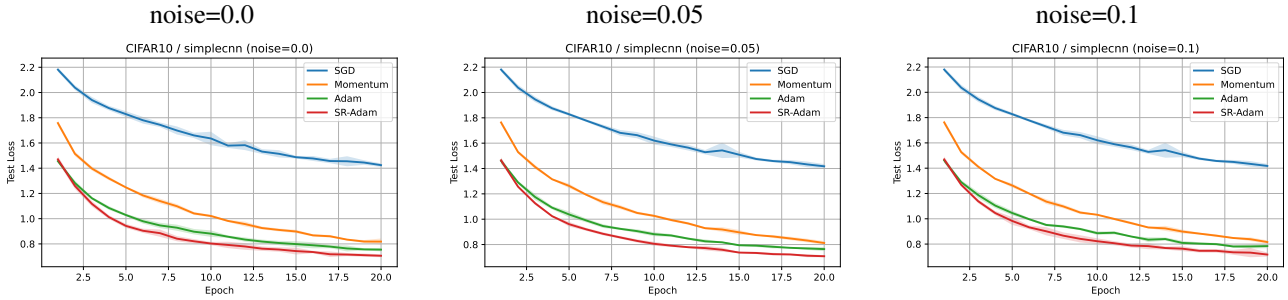

Fig. 2: SimpleCNN on CIFAR10: test loss vs. epoch across noise levels. Mean ± std over runs.

Each figure shows 10 randomly sampled test images arranged in columns. The top row displays the ground truth label, while the middle and bottom rows show Adam and SR-Adam predictions with their confidence scores. Correct predictions are marked in green; incorrect ones in red. The accuracies shown in parentheses correspond to the single best run selected for visualization.

These visualizations reveal several key insights: (1) SR-Adam tends to produce higher confidence scores on correctly classified samples, suggesting better calibration; (2) on misclassified samples, both methods often confuse visually similar classes (e.g., cats vs dogs, trucks vs automobiles), but SR-Adam demonstrates slightly better discrimination; (3) the shrinkage mechanism appears particularly beneficial for challenging, ambiguous samples where the raw gradient signal is noisy. While the quantitative improvement is modest (approximately 2 percentage points), the qualitative analysis shows that SR-Adam's adaptive shrinkage translates to more confident and stable predictions in practice.

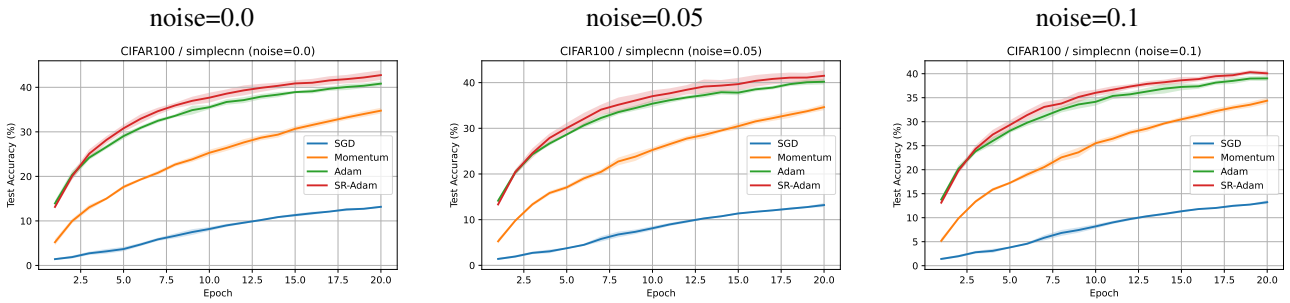

Fig. 3: SimpleCNN on CIFAR100: test accuracy vs. epoch across noise levels. Mean ± std over runs.

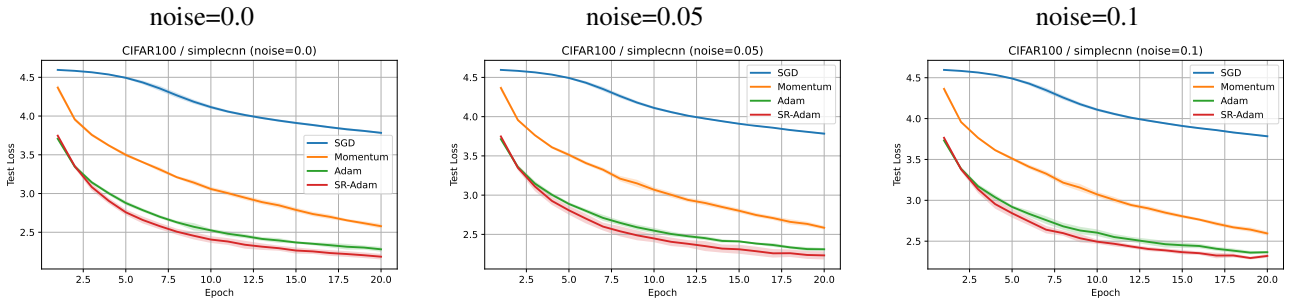

Fig. 4: SimpleCNN on CIFAR100: test loss vs. epoch across noise levels. Mean ± std over runs.

Optimizer	Noise=0.0	Noise=0.05	Noise=0.1
Adam	40.85 ± 0.62	40.25 ± 0.67	39.14 ± 0.61
Momentum	34.77 ± 0.72	34.63 ± 0.74	34.38 ± 0.71
SGD	13.17 ± 0.25	13.20 ± 0.49	13.23 ± 0.44
SR-Adam	42.74 ± 1.21	41.50 ± 1.34	40.43 ± 0.33
SR-Adam-All-Weights	34.99 ± 0.53	33.74 ± 0.64	33.03 ± 0.74

TABLE IV: Test Accuracy Results - CIFAR100 (batch_size=512)

Fig. 5: Qualitative comparison on CIFAR10 with 5% label noise. Each column shows a test image with its true label (top), Adam prediction (middle), and SR-Adam prediction (bottom). Predictions include class name and confidence score. Green indicates correct classification; red indicates misclassification. The accuracies in parentheses (74.7% for Adam, 76.2% for SR-Adam) represent the best single run out of 5 independent runs, selected for visualization clarity. Note that these values differ from the aggregated statistics in Table III (73.95±0.44% for Adam, 75.84±0.31% for SR-Adam), which report mean ± std across all runs. SR-Adam demonstrates more confident and accurate predictions on challenging samples, particularly on visually similar classes like cats/dogs and automobiles/trucks.

Fig. 6: Qualitative comparison on CIFAR100 with 5% label noise. Similar to Figure 5, this visualization shows sample predictions from the best-performing checkpoints. CIFAR100's 100-class taxonomy presents a significantly harder recognition task than CIFAR10. The fine-grained categories (e.g., distinguishing between different tree species or vehicle types) require more nuanced feature learning. SR-Adam's shrinkage mechanism helps stabilize gradient estimates in this high-noise, high-complexity regime, leading to more robust decision boundaries. The confidence scores reveal that SR-Adam produces higher-confidence predictions on correctly classified samples, suggesting improved calibration and reduced uncertainty.