Efficacy of SWATS Optimizer on Various Neural Network Architectures

Sourav Panda
College of IST, Penn State
sbp5911@psu.edu

Srivatsa Bhamidipati
College of IST, Penn State
sqb6374@psu.edu

Shubhang Kaushik
College of IST, Penn State
smp7224@psu.edu

Abstract

This study explores the efficacy of different optimization strategies for deep neural networks on CIFAR-10 and CIFAR-100 datasets. It compares traditional methods like SGD and Adam with the hybrid SWATS (Switching from Adam to SGD), across various architectures. The focus is on SWATS' potential to merge Adam's early efficiency with SGD's generalization. Performance is evaluated using metrics like accuracy, recall, and F1 score. The goal is to assess if hybrid training can balance rapid convergence with strong generalization, offering insights for enhancing neural network models, especially those with attention mechanisms.

1 SWATS

SWATS (Switching from Adam to SGD)[1] represents a novel strategy in deep learning optimization, merging the initial rapid convergence of Adam[2] with the proven generalization benefits of SGD[3]. Unlike conventional methods that use either Adam or SGD exclusively, SWATS begins with Adam to quickly navigate the parameter space and then switches to SGD for fine-tuning. This transition is automated and doesn't involve additional hyperparameters, facilitating a more streamlined optimization process.

Switchover Point: The switchover point is a pivotal aspect of SWATS, marking the shift from Adam to SGD. This transition is dictated by the convergence behavior of Adam. The criterion for the switchover is based on the stabilization of the learning rate, monitored via the projection of the Adam step on the gradient. The condition for the switchover is given by the Switchover Criterion equation:

$$\|\hat{\eta}_k - \eta_k\| < \epsilon \tag{1}$$

In this equation, η_k is the estimated learning rate at step k, and $\hat{\eta}_k$ is its exponentially averaged value, with ϵ being a predefined threshold. This criterion ensures that the switch occurs when the advantage of Adam's rapid progress diminishes, hence leveraging the benefits of both optimizers efficiently.

Learning Rate After Switch: Determining the learning rate for SGD after the switch is crucial for maintaining the convergence momentum gained from Adam. SWATS calculates this rate based on the alignment of Adam's updates with the gradient direction. The learning rate, η_k, is derived as follows:

$$\eta_k = \frac{\mathbf{p}_k^T \mathbf{p}_k}{-\mathbf{p}_k^T \mathbf{g}_k} \tag{2}$$

().

Here, $\mathbf{p}_k$ is the step computed by Adam at iteration k, and $\mathbf{g}_k$ is the corresponding stochastic gradient. This formula ensures that the step size for SGD is optimally scaled relative to the gradient's magnitude and direction. To stabilize this estimate, an exponential moving average is used:

$$\hat{\eta}_k = \beta_2 \hat{\eta}_{k-1} + (1 - \beta_2)\eta_k \tag{3}$$

This moving average $\hat{\eta}_k$ is initialized at zero and updated each step, using β_2, the second-moment decay rate from Adam. This approach ensures a smooth transition and a dynamically adapted learning rate for SGD, which is crucial for the continued optimization post-switch.

The goal of this work is less to propose a new training algorithm but rather to empirically investigate and compare the performance of the hybrid optimizer SWATS against traditional optimizers like Adam and SGD across various neural network architectures on CIFAR-10[4] and CIFAR-100[5] datasets for improving generalization. The significance of this research lies in its potential to simplify the hyperparameter tuning process, improve generalization on complex datasets, and provide empirical insights that could benefit a wide range of applications in deep learning.

2 Methodology

Datasets Description: The CIFAR-10 dataset includes 60,000 32x32 color images across 10 classes, with a training set of 50,000 and a test set of 10,000 images. It features various objects and animals. CIFAR-100, similar in size to CIFAR-10, presents a more detailed challenge with 100 classes and 600 images per class, divided into 500 training and 100 test images per class, covering a wide range of subjects. Both datasets are key in machine learning for testing image recognition models due to their balanced complexity.

Model Architectures:

ResNet18 (pre-trained=False): ResNet18[6], a ResNet variant, is initialized without pre-trained weights, requiring learning from scratch. It features 18 layers, including convolutional, batch normalization, ReLU activation, and fully connected layers. Unique "skip connections" in its architecture help address the vanishing gradient problem, providing gradient pathways during backpropagation. This design efficiently learns hierarchical features, making it effective in image classification, particularly for datasets where pre-trained models are less beneficial.

ResNet18 (pretrained=False) with Attention Layer: The enhanced ResNet-18 architecture, integrated with a Squeeze-and-Excitation (SE) block[7], is tailored for advanced image classification. The SE block, an attention mechanism, recalibrates channel-wise features, applying global average pooling and a series of activations. It generates channel-specific weights to highlight key features, enhancing the model's focus and information processing. Adapted for CIFAR-10 and CIFAR-100's class structure, this version blends ResNet-18's depth with SE's precision, making it highly effective for complex image recognition tasks.

3 Training Procedure

Data Preparation: Our experiments were conducted on two benchmark datasets: CIFAR-10 and CIFAR-100. Each dataset was divided into three subsets: training, validation, and testing. This division was crucial for training our models effectively and evaluating their performance under different conditions.

Training Process: Each model was trained on both CIFAR-10 and CIFAR-100 datasets. To ensure a comprehensive analysis, we employed three distinct optimizers: Adam, Stochastic Gradient Descent (SGD), and SWATS (Switching from Adam to SGD). This selection of optimizers allowed us to investigate their respective impacts on the learning process.

We conducted the training over 200 epochs for each model and optimizer combination. To address the variability in model initialization, each training session was repeated with five different seed values. This approach enabled us to assess the robustness of our models against different initialization conditions.

Monitoring Overfitting and Underfitting: To monitor and analyze overfitting and underfitting, we plotted the training and validation losses for each run. These plots provided insights into the model's learning dynamics and helped in identifying any discrepancies between the training and validation performance.

Assessing Model Performance Over Time: Another crucial aspect of our analysis was tracking the test error over epochs. By plotting these errors, we were able to evaluate how the models' performance evolved throughout the training process. This longitudinal analysis was instrumental in understanding the learning trajectory and stability of each model.

Comparative Analysis: A key component of our study was the comparative analysis of the different optimizers. We achieved this by collating and plotting the training losses and test errors for each model, run, and optimizer combination. This comprehensive comparison allowed us to draw nuanced conclusions about the efficacy of each optimizer in our experimental setup.

4 Evaluation Metrics:

In this project, we employed several key metrics to evaluate the performance of our models comprehensively. These metrics included accuracy, recall, and the F1 score, each offering unique insights into different aspects of model performance.

Accuracy: This measures the proportion of correct predictions. We evaluated accuracy on both validation and test sets for the CIFAR-10 and CIFAR-100 datasets, providing a clear view of our model's overall classification effectiveness.

Recall: Recall indicates the proportion of actual positives correctly identified. It's crucial where missing a positive instance is significant. We calculated recall for both validation and test datasets to gauge our model's ability to identify all relevant instances.

F1 Score: The F1 score, a harmonic mean of precision and recall, is valuable when dealing with uneven class distribution. It considers both false positives and negatives. Calculating this for validation and test datasets offered a nuanced understanding of model performance.

Aggregated Metrics: To capture a more robust measure of model performance, we didn't rely solely on metrics from individual runs. Instead, after completing all five runs with different seeds for each model and optimizer combination, we calculated mean accuracy, recall, and F1 score, after all, runs for each model and optimizer combination. This aggregation helped mitigate outlier effects, providing a more reliable assessment of true performance.

5 Results and Discussion

5.1 Results on CIFAR-10:

ResNet-18 : After analyzing the training loss and test errors for the ResNet-18 model trained on the CIFAR-10 dataset with Adam, SGD, and SWATS optimizers, the following observations were made: Adam exhibited a steady decrease in training loss over 200 epochs, indicating effective learning. SGD demonstrated a gradual and consistent decrease in training loss, which may imply better generalization compared to Adam. Test errors also decreased steadily, suggesting reliable performance improvements over epochs. SWATS showed an initial rapid decrease in training loss, similar to Adam, followed by a more stable pattern akin to SGD. This indicates the successful transition from Adam to SGD, combining the strengths of both optimizers. The test errors displayed a trend indicative of good generalization abilities.

These results suggest that while Adam is efficient for rapid convergence in the initial phase of training, it might be prone to overfitting. SGD, although slower, seems to offer better generalization. SWATS appears to effectively bridge the gap between the rapid learning of Adam and the stable generalization of SGD, showcasing its potential as a hybrid optimization strategy.

ResNet-18 with SE block Attention layer : After analyzing the training loss and test errors for the ResNet-18 model with the SE attention layer trained on the CIFAR-10 dataset with Adam, SGD, and SWATS optimizers, the following observations were made: For Adam the training loss decreased consistently over 200 epochs, indicating effective learning. The loss started at around 2.3 and re-

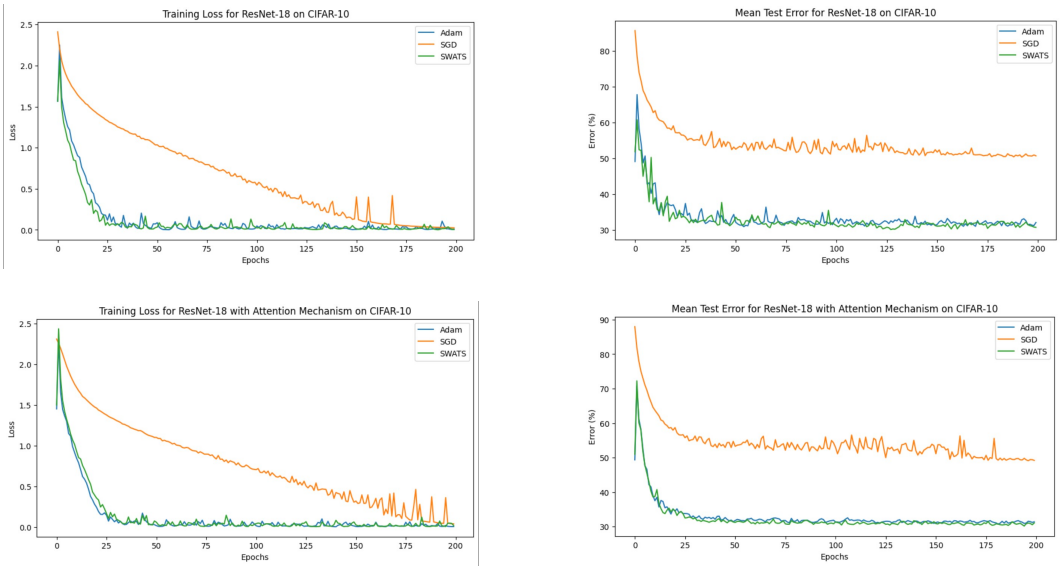

Figure 1: Training Loss and Test Error Plots for different model architecture and optimizer combinations on the CIFAR-10 dataset

duced to below 0.01 by the end of training. There was a consistent decrease in test errors, aligning with the reduction in training and validation losses. This suggests the model performed well on unseen data. For SGD the training loss for SGD also decreased over time, starting from around 2.3 and reaching values below 0.1. This indicates effective learning, but not as efficient as with Adam. Test errors decreased over epochs, but not as significantly as with Adam, suggesting slightly lower performance on unseen data. For SWATS the training loss with SWATS decreased impressively over 200 epochs, starting from around 1.5 and reaching values as low as 0.004. This suggests a very effective learning process. There was a decrease in test errors, although the data showed some inconsistencies in later epochs, possibly due to overfitting or other issues in the model optimization process.

Adam showed the most consistent and effective performance across training loss, validation loss, and test errors. SGD was effective but slightly less efficient than Adam, showing more variability in validation loss and test errors. SWATS had an impressive start but showed signs of overfitting or instability in later epochs, as indicated by fluctuations in validation loss and test errors. These results suggest that for this specific task and dataset, Adam might be the most reliable optimizer, although SWATS shows potential if issues with overfitting are addressed.

5.2 Results on CIFAR-100:

ResNet-18 : After analyzing the training loss and test errors for the ResNet-18 model trained on the CIFAR-100 dataset with Adam, SGD, and SWATS optimizers, the following observations were made: For the Adam optimizer, the initial training loss is approximately 4.65 and it gradually decreased to about 0.22 by the 200th epoch. Adam begins at a training loss of around 4.65, reducing to about 0.22 by the 200th epoch. For the SGD optimizer, the training loss starts at around 4.84 and decreases steadily to approximately 0.29 by the 200th epoch. SGD initiates at a training loss near 4.84, dropping to around 0.29 by the final epoch. For the SWATS optimizer, the initial training loss is around 3.71 and it significantly reduces to approximately 0.0018 by the 200th epoch. The test errors start at 82.61% and gradually decrease to 63.01% by the end of the training period.

Each optimizer shows a significant reduction in training loss over 200 epochs. SWATS stands out with the lowest final training loss and a substantial decrease in test error, suggesting better overall performance on the CIFAR-100 dataset with the ResNet-18 model. Adam and SGD also show effective loss reduction, but SWATS appears to be the most efficient in this scenario.

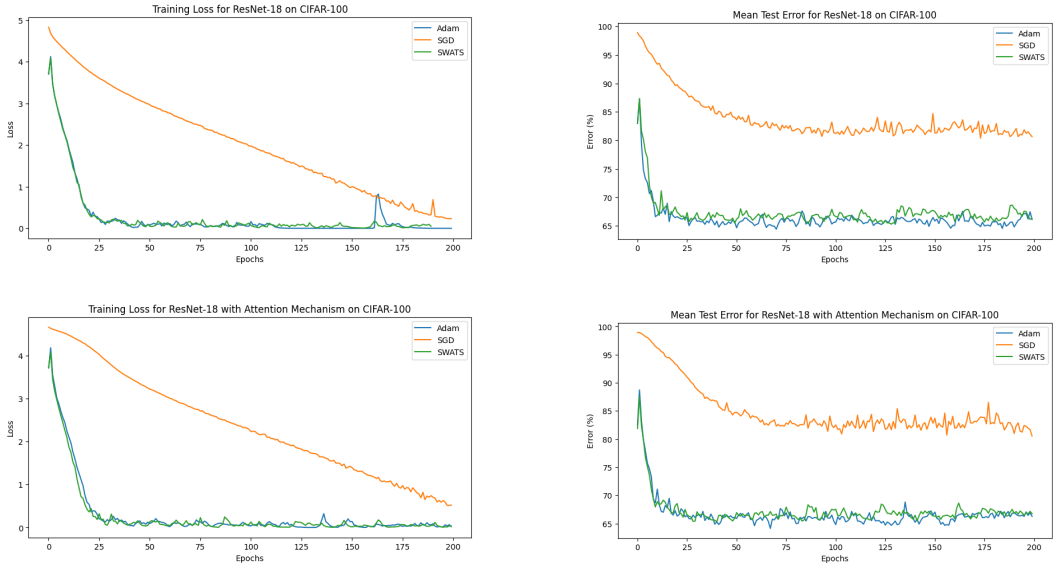

Figure 2: Training Loss and Test Error Plots for different model architecture and optimizer combinations on the CIFAR-100 dataset

ResNet-18 with SE block Attention layer After analyzing the training loss and test errors for the ResNet-18 model with SE attention block trained on the CIFAR-100 dataset with Adam, SGD, and SWATS optimizers, the following observations were made: For ADAM The training loss started at around 4.4 and steadily decreased, reaching a low of approximately 3.4. Fluctuations are observed, but the overall trend is downward, indicating effective learning. The test error showed a gradual decrease, starting from the 70s and reducing to the mid-60s, which implies a steady improvement in model performance on the test dataset. For SGD The training loss began at a higher value compared to ADAM, around 5.0, and decreased to a value close to 4.0. The decrease is more gradual compared to ADAM. The test error rates are relatively higher than ADAM, starting around the mid-70s and reducing to the high 60s. This suggests that while the model is learning, it's not as effective as with the ADAM optimizer. For SWATS the training loss shows an interesting pattern, starting around 4.5, decreasing slightly, and then increasing again. The final values are around 6.5, which is higher than the starting value, suggesting potential issues with the training process. The test error rates start in the low 70s and exhibit minor improvements, reaching the mid-60s. The pattern is less consistent compared to the other optimizers.

ADAM seems to be the most effective optimizer in terms of both reducing the training loss and test error for this specific model and dataset. SGD shows a consistent but slower improvement. It might benefit from more epochs or parameter tuning. SWATS shows an unusual pattern, especially in training loss, which could indicate problems with optimizer settings or compatibility with this specific model.

5.3 Results based on evaluation metrics

In the evaluation of three different optimizers—ADAM, SGD, and SWATS—on the CIFAR-10 dataset with a ResNet18 model featuring an attention layer, clear patterns emerge. ADAM consistently demonstrates the highest mean validation accuracy and F1 score, coupled with relatively low standard deviations, suggesting its stable and superior performance across runs. In comparison, SGD lags, ranking the lowest among the optimizers in terms of validation accuracy and F1 score, with notably higher standard deviations indicating less reliability. SWATS, although performing better than SGD, displays the highest standard deviations, indicating erratic performance. Despite its relatively higher accuracy and F1 score compared to SGD, the inconsistency in SWATS' results suggests caution in considering it as the optimal choice. In conclusion, ADAM emerges as the most stable and effective optimizer for this task, outperforming both SGD and SWATS in accuracy and F1 score, making it the preferable choice for training the ResNet18 model on the CIFAR-10 dataset.

In the evaluation of three optimizers—ADAM, SGD, and SWATS—on the CIFAR-100 dataset with a ResNet18 model incorporating an attention layer, several trends become evident. ADAM, while not achieving exceptionally high accuracy, exhibits a stable and consistent performance, with moderate mean accuracy, F1 score, and recall values and relatively low standard deviations. In contrast, SGD lags behind significantly, displaying the lowest mean accuracy and performance scores, although it maintains stable results. SWATS falls in between ADAM and SGD in terms of performance metrics but presents the highest variability across runs, making it less preferable due to its instability. Overall, for the CIFAR-100 dataset, ADAM appears as the most dependable optimizer, offering reliable and moderate performance, while SGD delivers consistent but suboptimal results, and SWATS, despite intermediate performance, suffers from high variability. Further optimization may be necessary to enhance performance on this challenging dataset.

6 Conclusion

The study provided valuable insights into the performance of various optimizers for training the ResNet-18 model on CIFAR-10 and CIFAR-100 datasets. For CIFAR-10, Adam optimizer showed consistent and effective performance across training loss, validation loss, and test errors, suggesting it might be the most reliable optimizer for this task. However, for CIFAR-100, SWATS stood out with the lowest final training loss and a substantial decrease in test error, indicating its potential efficacy on more complex datasets.

The incorporation of the SE attention block further influenced the optimizer performance. While Adam consistently showed effective learning, SWATS exhibited an impressive start but revealed signs of instability or overfitting in later epochs. This highlights the importance of choosing the right optimizer based on the specific model architecture and dataset complexity.

7 Limitations

Several limitations were observed in this study:

Model Complexity: The study was limited to the ResNet-18 model. Different architectures might yield different results with these optimizers.

Dataset Specificity: The performance of optimizers was only evaluated on CIFAR-10 and CIFAR-100 datasets. Results might vary with other datasets.

Optimizer Parameters: The default settings were used for each optimizer. Custom tuning of parameters might lead to different outcomes.

Training Duration: All models were trained for 200 epochs, which may not have been optimal for all optimizer and dataset combinations.

8 Future Work

Future research can focus on several areas to expand upon the findings of this study:

Exploring Different Architectures: Testing these optimizers on a variety of neural network architectures to generalize the findings.

Parameter Tuning: Experimenting with different hyperparameters for each optimizer to optimize model performance.

Longer Training Durations: Extending the number of training epochs to observe long-term trends in optimizer performance.

Regularization Techniques: Implementing and testing various regularization methods to mitigate overfitting, especially for optimizers like SWATS.

Cross-dataset Validation: Applying the models to different datasets to evaluate the generalizability of the optimizers across various data types and complexities.

Real-world Application: Testing these models on real-world datasets and tasks to assess their practical applicability and performance.

References

[1] Keskar, Nitish Shirish, and Richard Socher. "Improving generalization performance by switching from adam to sgd." arXiv preprint arXiv:1712.07628, 2017.

[2] Kingma, Diederik P., and Jimmy Ba. "Adam: A method for stochastic optimization." arXiv preprint arXiv:1412.6980, 2014.

[3] Hardt, Moritz, Ben Recht, and Yoram Singer. "Train faster, generalize better: Stability of stochastic gradient descent." International Conference on Machine Learning. PMLR, 2016.

[4] Çalik, Rasim Caner, and M. Fatih Demirci. "Cifar-10 image classification with convolutional neural networks for embedded systems." 2018 IEEE/ACS 15th International Conference on Computer Systems and Applications (AICCSA). IEEE, 2018.

[5] A. Krizhevsky, V. Nair, and G. Hinton, "CIFAR-100 (Canadian Institute for Advanced Research)," [Online]. Available: https://www.cs.toronto.edu/ kriz/cifar.html.

[6] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770-778, doi: 10.1109/CVPR.2016.90.

[7] Hu, Jie, Li Shen, and Gang Sun. "Squeeze-and-excitation networks." Proceedings of the IEEE conference on computer vision and pattern recognition. 2018.

[8] Sekhari, Ayush, Karthik Sridharan, and Satyen Kale. "Sgd: The role of implicit regularization, batch-size, and multiple-epochs." Advances In Neural Information Processing Systems 34, 2021: 27422-27433.

[9] Balles, Lukas, and Philipp Hennig. "Dissecting adam: The sign, magnitude and variance of stochastic gradients." International Conference on Machine Learning. PMLR, 2018.

[10] Zou, Difan, et al. "Understanding the generalization of adam in learning neural networks with proper regularization." arXiv preprint arXiv:2108.11371, 2021.

[11] Chen, Jinghui, et al. "Closing the generalization gap of adaptive gradient methods in training deep neural networks." arXiv preprint arXiv:1806.06763, 2018.

[12] Wilson, Ashia C., et al. "The marginal value of adaptive gradient methods in machine learning." Advances in Neural Information Processing Systems 30, 2017.

[13] Luo, Liangchen, et al. "Adaptive gradient methods with dynamic bound of learning rate." arXiv preprint arXiv:1902.09843, 2019.

[14] Zhang, Zijun. "Improved adam optimizer for deep neural networks." 2018 IEEE/ACM 26th international symposium on quality of service (IWQoS), 2018.

[15] Liu, Liyuan, et al. "On the variance of the adaptive learning rate and beyond." arXiv preprint arXiv:1908.03265, 2019.

[16] Reddi, Sashank J., Satyen Kale, and Sanjiv Kumar. "On the convergence of Adam and beyond." arXiv preprint arXiv:1904.09237, 2019.

[17] Zhang, Jingzhao, et al. "Why adam beats sgd for attention models." arXiv preprint arXiv:1911.11135, 2019.

[18] Choi, Dami, et al. "On empirical comparisons of optimizers for deep learning." arXiv preprint arXiv:1910.05446, 2019.

[19] Landro, Nicola, Ignazio Gallo, and Riccardo La Grassa. "Mixing Adam and SGD: a combined optimization method." arXiv preprint arXiv:2011.08042, 2020.

[20] MadhanMohan, S., and E. Karthikeyan. "Classification of Image using Deep Neural Networks and SoftMax Classifier with CIFAR datasets." 2022 6th International Conference on Intelligent Computing and Control Systems (ICICCS). IEEE, 2022.

[21] Lei, Yunwen, and Yiming Ying. "Fine-grained analysis of stability and generalization for stochastic gradient descent." International Conference on Machine Learning. PMLR, 2020.