Beyond First-Order: Training LLMs with Stochastic Conjugate Subgradient and AdamW

Di Zhang, Yihang Zhang and Suvrajeet Sen
Department of Industrial and System Engineering
University of Southern California
Los Angeles, CA 90089

Abstract

Algorithms based on Stochastic Gradient-based Descent (SGD), have long been central to training large language models (LLMs). However, their effectiveness can be questionable, particularly in large-scale applications where empirical evidence suggests potential performance limitations. In response, this paper proposes a stochastic conjugate subgradient method together with adaptive sampling tailored specifically for training LLMs. The method not only achieves faster convergence per iteration but also demonstrates improved scalability compared to traditional SGD techniques. It leverages several fundamental concepts including adaptive sample complexity analysis, an adaptive method to choose learning-rate, as well as a stochastic conjugate subgradient approach to determine search directions and utilizing an AdamW-like algorithm to adaptively adjust learning-rate. This approach preserves the key advantages of first-order methods while effectively addressing non-smoothness inherent in training LLMs. Experimental results show that the proposed method not only maintains, but in many cases surpasses, the scalability of traditional SGD techniques, significantly enhancing both the speed and accuracy of the optimization process.

1 Introduction

1.1 Problem Setup

Large Language Models (LLMs) are widely used to predict the next token in a sequence, given the preceding tokens [3, 8, 17]. The training objective is defined as minimizing the Negative Log-Likelihood (NLL) of the training data. Given a dataset $\mathcal{D} = \{(x^{(i)}, y^{(i)})\}_{i=1}^{N}$, where $x^{(i)}$ is the input text sequence and $y^{(i)}$ is the target token sequence, the objective function can be written as:

$$L(\theta) = -\sum_{i=1}^{N}\sum_{t=1}^{T_i} \log P_\theta(y_t^{(i)} \mid y_{1:t-1}^{(i)}) \tag{1}$$

where T_i is the length of the i-th sequence, $y_{1:t-1}^{(i)}$ represents the tokens preceding token $y_t^{(i)}$ and P_θ is the model's predicted probability distribution over the vocabulary, parameterized by θ.

To train the LLMs, many stochastic gradient-based descent (SGD) algorithms have been proposed such as Adam and AdamW. SGD methods dominate the field for convincing reasons: low computational cost, simplicity of implementation, and strong empirical results. However, despite the advantages, there are also many limitations: a) SGD methods are not known for their numerical accuracy

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: MLxOR: Mathematical Foundations and Operational Integration of Machine Learning for Uncertainty-Aware Decision-Making.

especially when the functions are non-smooth and b) establishing rigorous convergence guarantees is challenging because of the non-smooth and non-convex nature of the objective function.

1.2 Contributions

Instead of relying solely on the SGD approaches, this paper considers a stochastic conjugate subgradient (SCS) [21, 20, 19] approach, together with an adaptive sampling strategy [7, 4, 23]. Originating from [16], this method not only accommodates the curvature of objective functions, but also inherits the spirit of momentum methods by utilizing sample complexity analysis to reduce computational burden. This combination enhances the power of SGD approaches without the additional burden of second-order approximations. In other words, our method shares the spirit of Hessian-Free (HF) optimization, which has gained attention in the machine learning community [11, 1, 24]. As our theoretical analysis and computational results reveal, the new method provides a "sweet spot" at the intersection of speed, accuracy, and scalability of algorithms. Our main contributions include:

- Addressing challenges posed by non-smoothness of LLMs, while extending the non-smooth SCS method [16, 19] to training LLMs. The SCS method exhibits behavior that is similar to higher-order methods but avoids the significant computational overhead associated with maintaining a Hessian matrix at each iteration. To the best of our knowledge, this is the first attempt to apply an almost-higher-order optimization technique to training LLMs. (Note that a true higher-order-method would entail Hessian-approximation updates which we avoid.)
- While a straightforward application of [16] to LLMs using sample average approximation (SAA) [13] might be considered, our focus is on solving LLMs with extremely large datasets. Consequently, we employ an adaptive sampling strategy over a deterministic finite sum approximation. By leveraging the sample complexity analysis, we propose an adaptive sampling strategy which dramatically reduces the computational burden associated with training LLMs.
- The computational results demonstrate that, from an optimization standpoint, the solutions produced by SCS yield lower objective function values compared with SGD methods. More crucially, the efficiency and accuracy of our algorithm surpasses those of algorithms such as Adam and AdamW, which are specialized SGD algorithms for training LLMs.
- The proposed algorithm integrates several disparate notions such as adaptive sampling, decomposition, conjugate subgradients, and dynamic learning-rate as in Adam and AdamW. We demonstrate that this amalgamation effectively approximates solutions for LLMs with extremely large datasets.

2 The Method

The stochastic conjugate subgradient + AdamW (SCSAdamW) algorithm is a hybrid optimization method which integrates the AdamW optimizer with a conjugate subgradient approach to improve convergence. The method combines adaptive learning-rates (AdamW) with conjugate subgradient updates, allowing for better handling of curvature information while maintaining stability and convergence properties. A summary of the major steps of the algorithm is provided below:

- **Sequential adaptive sampling** At any iteration k, we depart from classic LLMs (which use an "all-in-one" optimization problem), by randomly sampling $|N_k|$ subproblems. This is similar to using sample average approximation (SAA) to approximate function L using L_k, where
$$L_k(\theta) = -\sum_{n=1}^{N_k} \sum_{t=1}^{T_i} \log P_\theta(y_t^{(i)} \mid y_{1:t-1}^{(i)}), \tag{2}$$
and $|N_k|$ will be determined by using a recommendation provided by using concentration inequalities.
- **Direction finding** The idea here is inspired by non-smooth conjugate subgradient method [15] which uses the smallest norm of the convex combination of the previous search direction (d_{k-1}) with the current subgradient (g_k). More specifically, we first solve the following

one-dimensional QP

$$\begin{aligned}\lambda_k^* &= \arg\min_{\lambda_k \in [0,1]} \frac{1}{2}||\lambda_k \cdot d_{k-1} + (1-\lambda_k)\cdot g_k||^2 \\ &= \Pi_{[0,1]}\big(\frac{-\langle d_{k-1}, g_k\rangle + ||g_k||^2}{||d_{k-1}||^2 - 2\langle d_{k-1}, g_k\rangle + ||g_k||^2}\big),\end{aligned} \tag{3}$$

where $\Pi_{[0,1]}$ denotes the projection operator on $[0, 1]$. We set the new search direction as

$$d_k = \lambda_k^* \cdot d_{k-1} + (1-\lambda_k^*)\cdot g_k, \tag{4}$$

Clearly if one fixes $\lambda_k = 0$, then the search direction reduces to that of the subgradient method.

- **Dynamic learning-rate** SCS-AdamW leverages an adaptive learning-rate, meaning each parameter has its own learning-rate. This is achieved by normalizing the conjugate subgradient updates using second moment estimates. The term $\frac{\eta}{\sqrt{\hat{v}_k}+\varepsilon}$ ensures that the update magnitude adapts to the scale of the gradients, preventing large updates in high-gradient regions (i.e., $\|g_k\|$ is relatively large) and avoids vanishing updates in low-gradient regions.
- **Decoupled weight decay** In standard Adam (without decoupled weight decay), L2 regularization is applied inside the subgradient update, meaning the weight decay term is included in the computation of g_k. However, this can interfere with the adaptive learning rate mechanism of Adam, making weight decay behave inconsistently across different parameters.AdamW enforces this by subtracting $\eta\lambda\theta_{k-1}$ after the Adam update step, ensuring that: (a) Weight decay acts as a separate force reducing parameter values without altering gradient estimates. (b) Improved generalization: Since weight decay is applied consistently across parameters, thus avoiding interference with Adam's adaptive updates.
- **Termination criteria** The algorithm concludes its process when $||d_k|| \leq \varepsilon$. A diminutive value of $||d_k||$ indicates a small norm of the subgradient g_k, fulfilling the stationary point condition for an unconstrained non-convex optimization problem.

Algorithm 1 Stochastic Conjugate Subgradient + AdamW (SCSAdamW) Algorithm

Require: Parameters θ, learning-rate η, decay rates β_2, weight decay λ, small constant ζ.

Initialize $m = 0$, $v = 0$, $d = 0$, $k = 0$.
while $||d_k|| > \varepsilon$ **do**
  $k \leftarrow k + 1$
  Compute gradient $g_k = \nabla L_k(\theta_k)$.
  if $k > 1$ **then**
    Compute conjugate subgradient direction d_k by (4).
  else

$$d_k = g_k$$

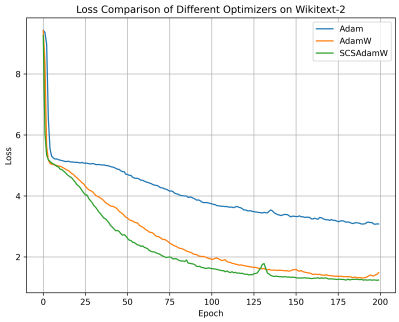
  .
  end if
  Update estimated second moment: $v_k = \beta_2 v_{k-1} + (1-\beta_2)\|g_k\|^2$.
  Bias correction: $\hat{d}_k = d_k/[1-(\lambda_k^*)^k]$, $\hat{v}_k = v_k/(1-\beta_2^k)$.
  Update parameters: $\theta_k = \theta_{k-1} - \frac{\eta}{\sqrt{\hat{v}_k}+\zeta}\hat{d}_k$.
  Apply decoupled weight decay (AdamW): $\theta_k \leftarrow \theta_k - \eta\lambda\theta_{k-1}$.
end while

3 Experiments

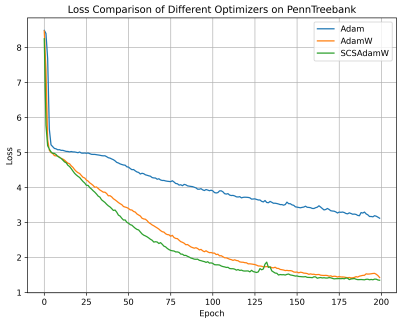
The experiments aim to compare the performance of different optimization algorithms on training LLMs based on LSTM (widely used in many different areas ([2, 5, 6, 14, 18])). The optimizers are Adam, AdamW and SCSAdamW. The comparison is conducted by training the language model Wikitext-2 [12],PennTreebank [10], ag-news [22] and imbd [9], measuring the loss across multiple epochs, and analyzing the convergence behavior of each optimizer. The training process is conducted

over 200 epochs for each optimizer. To ensure a fair comparison, each optimizer is configured with a learning-rate of 0.001 and a weight decay of 0.001. The hidden states of the LSTM are reset and detached at each epoch to prevent gradient explosion. Throughout training, the loss values are recorded for each epoch, and performance is evaluated by comparing the loss curves across optimizers. The algorithms of this paper were implemented on NVIDIA H100 GPU. The code used in this research was written in Python and is available at the following GitHub repository: SCSAdamW (accessed on April 24, 2025), and the computational results are shown in Figure 1.

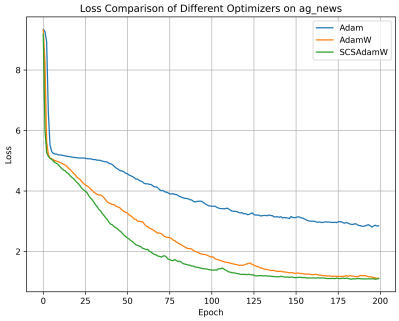
Figure 1: Objective function values for different algorithms.



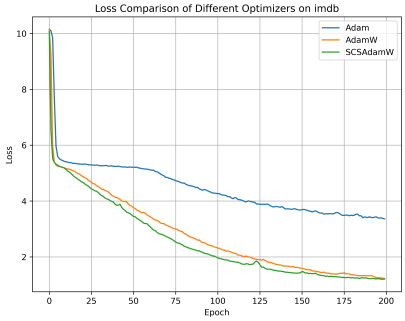
(a) Wikitext-2



(b) PennTreebank



(c) Ag-news



(d) Imdb

Remark: Note that obtaining true optimality when training LLMs is usually too demanding. For this paper, we limit computations to only 200 epochs. With this limited computational budget, the SCSAdamW method usually converges faster than any of the other methods (i.e., attains lowest objective values).

4 Conclusion

This paper introduced SCSAdamW, an optimization algorithm which can be used for training LLMs. Our approach presents several key innovations compared to conventional SGD-based benchmark algorithms: (a) It employs a stochastic conjugate subgradient direction, as opposed to standard subgradient directions; (b) It maintains a computational resource requirement comparable to that of standard SGD methods; (c) It dynamically adjusts the sample size during training. This adaptive approach can offer improved robustness and efficiency compared to traditional training paradigms that rely on fixed sample sizes throughout optimization. Our preliminary computational results highlight the algorithm's strong empirical performance across several test instances. On these benchmarks, SCSAdamW demonstrated faster convergence and achieved lower objective function values compared to widely used optimizers such as Adam and AdamW. Given the broad applicability of LLMs, we believe that our algorithm will prove valuable in a wide range of practical scenarios.

References

[1] F. E. Curtis and K. Scheinberg. Optimization methods for supervised machine learning: From linear models to deep learning. In *Leading Developments from INFORMS Communities*, pages 89–114. INFORMS, 2017.

[2] C. Feng, B. Bačić, and W. Li. Sca-lstm: A deep learning approach to golf swing analysis and performance enhancement. In *International Conference on Neural Information Processing*, pages 72–86. Springer, 2025.

[3] X. Hu, Y. Cheng, D. Yang, Z. Xu, Z. Yuan, J. Yu, C. Xu, Z. Jiang, and S. Zhou. Ostquant: Refining large language model quantization with orthogonal and scaling transformations for better distribution fitting. *The Thirteenth International Conference on Learning Representations*, 2025.

[4] Y. Li, C. Yang, J. Dong, Z. Yao, H. Xu, Z. Dong, H. Zeng, Z. An, and Y. Tian. Ammkd: Adaptive multimodal multi-teacher distillation for lightweight vision-language models. *arXiv preprint arXiv:2509.00039*, 2025.

[5] Z. Li and Z. Ke. Mitigating demographic bias in vision transformers via attention-guided fair representation learning. In *Workshop on Demographic Diversity in Computer Vision@ CVPR 2025*.

[6] Z. Li, B. Wang, and Z. Ke. From tabular to time series: Can tabpfn handle mixed data? a study on physionet. In *1st ICML Workshop on Foundation Models for Structured Data*.

[7] W. Liu, J. Xu, F. Yu, Y. Lin, K. Ji, W. Chen, Y. Xu, Y. Wang, L. Shang, and B. Wang. Qfft, question-free fine-tuning for adaptive reasoning. *arXiv preprint arXiv:2506.12860*, 2025.

[8] Y. Lu, H. Cheng, Y. Fang, Z. Wang, J. Wei, D. Xu, Q. Xuan, X. Yang, and Z. Zhu. Reassessing layer pruning in llms: New insights and methods. *arXiv preprint arXiv:2411.15558*, 2024.

[9] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts. Learning word vectors for sentiment analysis. *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, 1:142–150, 2011.

[10] M. Marcus, B. Santorini, and M. A. Marcinkiewicz. Building a large annotated corpus of english: The penn treebank. *Computational linguistics*, 19(2):313–330, 1993.

[11] J. Martens et al. Deep learning via hessian-free optimization. In *ICML*, volume 27, pages 735–742, 2010.

[12] S. Merity, C. Xiong, J. Bradbury, and R. Socher. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*, 2016.

[13] A. Shapiro. Monte carlo sampling methods. *Handbooks in operations research and management science*, 10:353–425, 2003.

[14] Y. Sun, Y. Li, R. Sun, C. Liu, F. Zhou, Z. Jin, L. Wang, X. Shen, Z. Hao, and H. Xiong. Audio-enhanced vision-language modeling with latent space broadening for high quality data expansion, 2025.

[15] P. Wolfe. Note on a method of conjugate subgradients for minimizing nondifferentiable functions. *Mathematical Programming*, 7:380–383, 1974.

[16] P. Wolfe. A method of conjugate subgradients for minimizing nondifferentiable functions. In *Nondifferentiable optimization*, pages 145–173. Springer, 1975.

[17] D. Xiang, W. Xu, K. Chu, Z. Shen, T. Ding, and W. Zhang. Promptsculptor: Multi-agent based text-to-image prompt optimization. *arXiv preprint arXiv:2509.12446*, 2025.

[18] S. Zeng, X. Chang, M. Xie, X. Liu, Y. Bai, Z. Pan, M. Xu, and X. Wei. Futuresightdrive: Thinking visually with spatio-temporal cot for autonomous driving. *arXiv preprint arXiv:2505.17685*, 2025.

[19] D. Zhang. *A Stochastic Conjugate Subgradient Framework for Large-Scale Stochastic Optimization Problems*. PhD thesis, University of Southern California, 2024.

[20] D. Zhang and S. Sen. A sampling-based progressive hedging algorithm for stochastic programming. *arXiv preprint arXiv:2407.20944*, 2024.

[21] D. Zhang and S. Sen. The stochastic conjugate subgradient algorithm for kernel support vector machines. *SIAM Journal on Optimization*, 35(2):1194–1215, 2025.

[22] X. Zhang, J. Zhao, and Y. LeCun. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems*, 2015.

[23] Y. Zheng, B. Zhong, Q. Liang, S. Zhang, G. Li, X. Li, and R. Ji. Towards universal modal tracking with online dense temporal token learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

[24] C. Zhou, H. Xu, N. Gu, Z. Wang, B. Cheng, P. Zhang, Y. Dong, M. Hayashibe, Y. Zhou, and B. He. Language-guided long horizon manipulation with llm-based planning and visual perception, 2025.