LLMs Optimization Landscapes

Jacopo Boscariol, Leonardo De Novellis, Salya Amanda Diallo
CS-439: Optimization for machine learning, EPF Lausanne, Switzerland

***Abstract*—The geometry of a model loss landscape has often been linked to its generalisation performance. The loss landscape of `DistilGPT2` fine-tuned on the `WikiText-2` corpus was studied by varying batch size, weight decay, optimizer (*SGD* and *AdamW*), and data schedule (random order and a curriculum-learning approach). Layer-wise normalisation was adopted to obtain two-dimensional projections of the local loss surface that are robust to scale invariance of neural networks. The results are that larger batches converge to flatter minima, and a short-to-long curriculum further increases the flatness of the landscape. However, the relationship between flatness and generalization is not strictly positive: flatter minima do not consistently lead to better generalization performance.**

I. Introduction

Understanding the relationship between the optimization landscapes of deep neural networks and their performance is critical for improving their generalization, stability, and convergence behaviour. Past works have linked the geometric properties of trained models, in particular through various notions of sharpness, to their generalization performance [1], [2]. This study focuses on the loss landscapes of large language models (LLMs), specifically `DistilGPT2` fine-tuned on the `Wikitext-2` dataset. The goal is to characterize how training hyper-parameters and data ordering influence the model optimization properties. The work builds primarily on the approach proposed by Li et al. [3] and extends it to LLMs.

The analysis considers multiple optimization configurations, including variations in batch size, learning rate, weight decay, and optimizer choice, as well as the effect of training on randomly shuffled versus length-sorted data. Two-dimensional (2D) projections of the loss surface around converged model parameters are visualized using established techniques [3]. Sharpness is quantified using two complementary metrics: ϵ-sharpness [1], [4], which measures the maximum increase in loss within an ϵ-ball around the solution, and Hessian sharpness, estimated via the top eigenvalue of the Hessian matrix, $\lambda_{\max}$.

The results provide empirical insight into the influence of training regimes on the geometry of the loss landscape, contributing to a deeper understanding of the relationship between optimization process and generalization in LLMs. All codes are publicly available in GitHub: https://github.com/JackBosca/cs439_project

II. Models and Methods

The experiments utilized a subset of the `WikiText-2` dataset, a medium-sized corpus of verified Wikipedia articles, selected for its moderate size, which aligns with the available computational resources. All text inputs were tokenized using the the `DistilGPT2` tokenizer. Sequences were truncated or padded to a uniform maximum length of 128 tokens to ensure consistent input dimensions during training. The model employed across all experiments was `DistilGPT2`, a distilled variant of `GPT2` that offers a trade-off between computational efficiency and performance. This choice was motivated by computational constraints, as it allows for efficient experimentation. The model was fine-tuned using the preprocessed `WikiText-2` data.

All experiments were conducted using a unified pipeline comprising training, evaluation, visualization, and sharpness estimation components. Models were initialized using pre-trained weights and fine-tuned for 60 epochs with a specified optimizer (Stochastic Gradient Descent, *SGD*, or *AdamW* [5]), learning rate, weight decay, and batch size. The data loader supported randomized or sequential ordering to accommodate curriculum-related settings [6]. Training data difficulty for curriculum learning was estimated using the heuristic that shorter sequences are easier to learn due to lower contextual complexity [7], [8]. After training, validation loss and perplexity were computed.

Visual analysis included one-dimensional linear interpolations and two-dimensional loss surface projections along normalized orthogonal directions in parameter space. When visualizing the loss landscape surrounding a minima, it is important to account for the scale invariance of certain neural networks. Dinh et al. [4] demonstrated that networks with ReLU activations or batch normalization can be rescaled without affecting their outputs, allowing the construction of equivalent models with arbitrarily sharp minima. To mitigate this, the layer-wise normalization method from [3] was employed, which makes the scale depend on the norm of the network weights.

1D linear interpolation consists of evaluating the loss function along the straight line connecting two sets of model parameters. Given models $\boldsymbol{\theta}_0$ and $\boldsymbol{\theta}_f$, the interpolated parameters are defined as $\boldsymbol{\theta}(\alpha) = (1-\alpha)\boldsymbol{\theta}_0 + \alpha\boldsymbol{\theta}_f$. The interpolation is performed over $\alpha \in [-0.5, 1.5]$ to capture also the surrounding landscape. For 2D projections, two layer-normalized orthogonal directions were generated in parameter space. A grid of perturbed models was constructed by applying combinations of these directions to the base model, and the loss was computed at each grid point to visualize the curvature and structure of the local loss landscape.

Sharpness was quantified using Hessian-based power iteration and ϵ-perturbation metrics [4], with verification via second-order Taylor approximation. Both metrics were computed using a subset of 3 batches from the training set

for computational efficiency. While this introduces some approximation, it retains the essential trends and relative differences between models. The dominant Hessian eigenvalue $\lambda_{\max}$ was estimated using power iteration with batch-averaged Hessian-vector products. The ϵ-sharpness was computed as the maximum relative increase in loss under norm-constrained perturbations of the model parameters. Specifically, the parameters were perturbed along the normalized leading eigenvector of the Hessian and the loss was evaluated across multiple perturbation samples. The sharpness is eventually defined as:

$$\epsilon\text{-sharpness} = \max_{i=1,\dots,n} \left(\frac{\mathcal{L}(\boldsymbol{\theta} + \epsilon \mathbf{v}_i) - \mathcal{L}(\boldsymbol{\theta})}{\mathcal{L}(\boldsymbol{\theta})} \right) \times 100\,, \quad (1)$$

where $\boldsymbol{\theta}$ denotes the model parameters, n is the number of perturbation samples, $\mathbf{v}_i$ is the dominant Hessian eigenvector or a random direction, with $||\mathbf{v}_i|| = 1$, ϵ controls the perturbation magnitude and $\mathcal{L}$ denotes the loss function. Approximation fidelity was validated by comparing the empirical loss change to the predicted increase given by a second order Taylor expansion:

$$\begin{aligned} \mathcal{L}(\boldsymbol{\theta} + \boldsymbol{\delta}) &\approx \mathcal{L}(\boldsymbol{\theta}) + \nabla \mathcal{L}(\boldsymbol{\theta})^\top \boldsymbol{\delta} + \tfrac{1}{2} \boldsymbol{\delta}^\top \nabla^2 \mathcal{L}(\boldsymbol{\theta}) \boldsymbol{\delta} \\ &\approx \mathcal{L}(\boldsymbol{\theta}) + \epsilon \nabla \mathcal{L}(\boldsymbol{\theta})^\top \mathbf{v} + \tfrac{1}{2} \epsilon^2 \lambda_{\max}\,, \end{aligned} \quad (2)$$

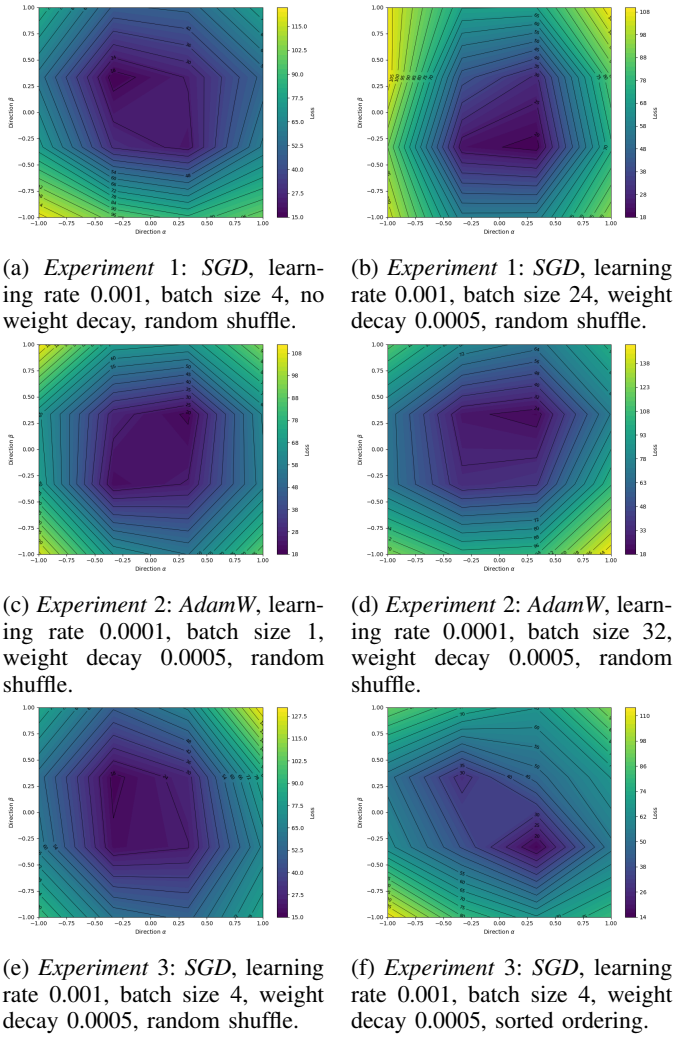
where the last approximation holds since $\delta = \epsilon \mathbf{v}$ has been chosen, with $\mathbf{v}$ the dominant Hessian eigenvector. In Equation 2, one batch has been used for computations, since this appeared sufficient to obtain a meaningful validation.

III. Results

Three experiments were conducted. The first (*Experiment* 1) compared models trained with *SGD* using identical learning rates but varying batch sizes (4 vs. 24) and weight decay values (0 vs. 0.0005). The second (*Experiment* 2) investigated optimizer performance by holding learning rate, weight decay, and data order constant while utilizing the *AdamW* optimizer across two batch sizes (1 and 32). The third (*Experiment* 3) examined the impact of data order by training models using identical configurations while varying the data presentation: randomly shuffled versus length-sorted, corresponding to baseline and curriculum-learning respectively.

Figure 1 shows the 2D loss surfaces for each experiment and model, providing a visualization of the geometry of the loss function and its impact on optimization. Figure 2 presents 1D loss interpolation plots (loss vs. interpolation factor) for each experiment, illustrating how loss varies between two sets of parameters to assess landscape smoothness and connectivity.

Table I reports the ϵ-sharpness and the largest eigenvalue of the Hessian, $\lambda_{\max}$, for each experiment and model. The largest eigenvalue reflects the maximum curvature of the loss landscape, indicating the sharpness of the local minimum and the robustness of the model to parameter perturbations.



(a) *Experiment* 1: *SGD*, learning rate 0.001, batch size 4, no weight decay, random shuffle.

(b) *Experiment* 1: *SGD*, learning rate 0.001, batch size 24, weight decay 0.0005, random shuffle.

(c) *Experiment* 2: *AdamW*, learning rate 0.0001, batch size 1, weight decay 0.0005, random shuffle.

(d) *Experiment* 2: *AdamW*, learning rate 0.0001, batch size 32, weight decay 0.0005, random shuffle.

(e) *Experiment* 3: *SGD*, learning rate 0.001, batch size 4, weight decay 0.0005, random shuffle.

(f) *Experiment* 3: *SGD*, learning rate 0.001, batch size 4, weight decay 0.0005, sorted ordering.

Fig. 1: 2D training loss surfaces along normalized orthogonal directions in parameter space. Each row compares models from a single experiment.

Model	ϵ-sharpness	Check Equation 2	$\lambda_{\max}$
1	21.88%	0.01%	5042.6
2	15.74%	0.01%	8943.2

(a) *Experiment* 1: *SGD*, varying batch size and weight decay.

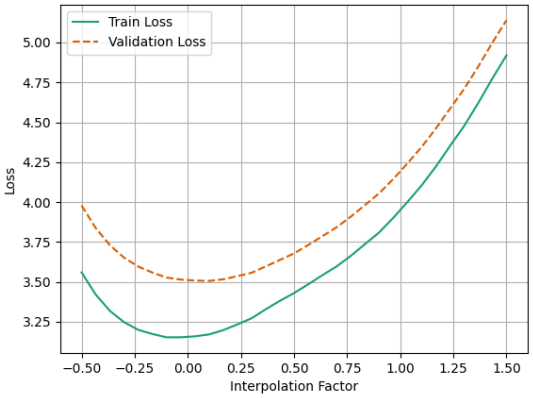
Model	ϵ-sharpness	Check Equation 2	$\lambda_{\max}$
1	−15.79%	0.02%	17.3
2	−1.62%	0.04%	2353.7

(b) *Experiment* 2: *AdamW*, varying batch size.

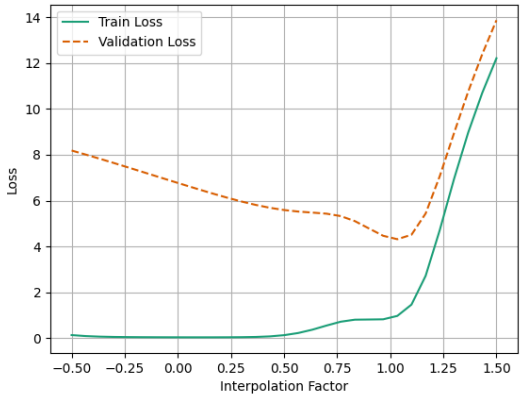
Model	ϵ-sharpness	Check Equation 2	$\lambda_{\max}$
1	19.65%	0.01%	4793.3
2	15.38%	0.01%	4029.0

(c) *Experiment* 3: *SGD*, varying data order.

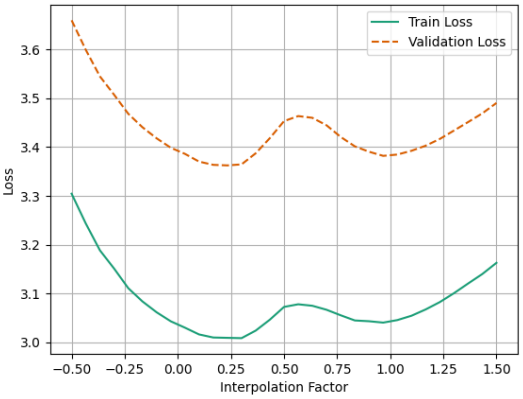
TABLE I: ϵ-sharpness and $\lambda_{\max}$ of the Hessian for each model corresponding to the experimental setups shown in Figure 1. $\epsilon = 10^{-3}$ perturbs along the largest eigenvalue direction. The check value from Equation 2 reports the absolute difference between the measured and predicted relative loss increases, as described in Section II.



(a) *Experiment* 1: *SGD*, varying batch size and weight decay.



(b) *Experiment* 2: *AdamW*, varying batch size.



(c) *Experiment* 3: *SGD*, varying data order.

Fig. 2: Loss curve showing training and validation losses evaluated along a linear interpolation between first ($\boldsymbol{\theta}_0$) and second ($\boldsymbol{\theta}_f$) model parameters. The presence of a local maximum in the training loss (blue) line segment indicates that the two models converged to different minima.

IV. Discussion

Experiment 1 - *SGD*, varying batch size and weight decay: Model 1 (batch size $= 4$, no weight decay) exhibits higher ϵ-sharpness (21.88%) compared to model 2 (batch size $= 24$, weight decay $= 0.0005$), which achieves 15.74%. Despite its lower sharpness, model 2 displays a larger maximum Hessian eigenvalue (8943.2 vs. 5042.6), indicating sharper curvature along specific directions. These findings align with prior work showing that larger batches and regularisation tend to lead to flatter regions, however in this case this does not necessarily lead to stronger generalisation performance (see Figure 2a).

Experiment 2 - *AdamW*, varying batch size:
Model 1 (batch size $= 1$) yielded a negative ϵ-sharpness (-15.8%), indicating a reduction in loss under perturbation. This decrease may result from the combination of limited loss estimation using only three batches and noisy gradient updates caused by the minimal batch size. The low maximum Hessian eigenvalue (17.25) further supports the possible convergence to a non-informative minimum. In contrast, model 2 (batch size $= 32$) reached a substantially higher Hessian norm (2353.7) and exhibited only slightly negative sharpness (-1.6%). Also in this case, as observed in Figure 2b, the sharpest minima provides the best generalisation performance, as the discrepancy between validation and training line segments is lower in the model 2 (interpolation factor $\alpha = 1$) case. These observations highlight the instability introduced by extremely small batch sizes, even when using adaptive optimizers such as *AdamW*.

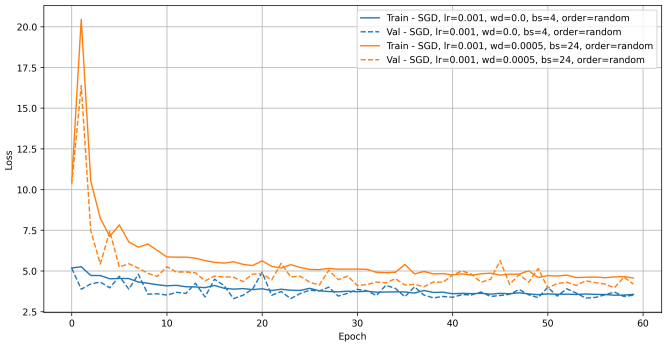
Experiment 3 - *SGD*, varying data ordering:
Model 2, which was trained using data sorted by length, achieved slightly lower sharpness (15.38%) than model 1, which was trained using randomly shuffled data (19.65%). The 2D loss surface in Figure 1f appears smoother than that in Figure 1e, supporting the idea that curriculum learning helps guiding the optimizer towards flatter minima. The loss segment in Figure 2c also shows a smoother path, which reinforces the hypothesis that data ordering can act as a mild regularizer. The two configurations converge to similarly low validation losses, see Appendix I.

V. Summary

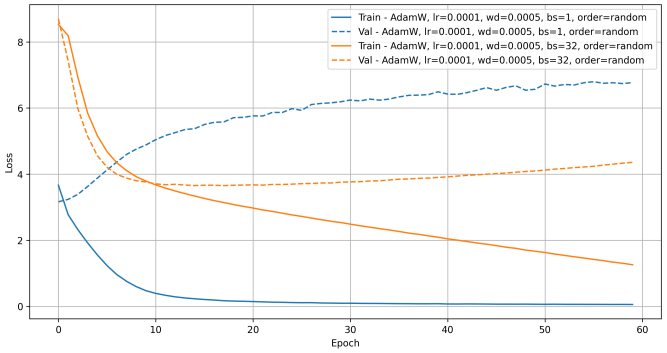
This study explored the optimization landscapes of `DistilGPT2` fine-tuned on the `WikiText-2` dataset. The results demonstrate that larger batch sizes (e.g., 24 vs. 4) combined with weight decay lead to flatter minima. When comparing optimizers, *AdamW* exhibited instability with extremely small batches (e.g., 1), while larger batches (e.g., 32) produced more structured minima. This underscores the critical role of batch size in adaptive optimization. Additionally, curriculum learning—training on length-sorted data—yielded flatter minima (15.38% vs. 19.65% ϵ-sharpness, 4029.0 vs. 4793.3 $\lambda_{\max}$) compared to random data ordering, supporting the idea that progressive difficulty acts as an implicit regularizer. However, these findings do not strictly align with the hypothesis that flat minima enhance generalization by being more robust to parameter perturbations, as experiments did not consistently link lower sharpness metrics to better performance.

These observations, supported by empirical evidence, help bridge theoretical principles of non-convex optimization with the behaviour of large language models during training. Future work could refine sharpness metrics and extend this analysis to larger-scale models, offering deeper insight into how optimization dynamics influence performance in modern neural architectures.

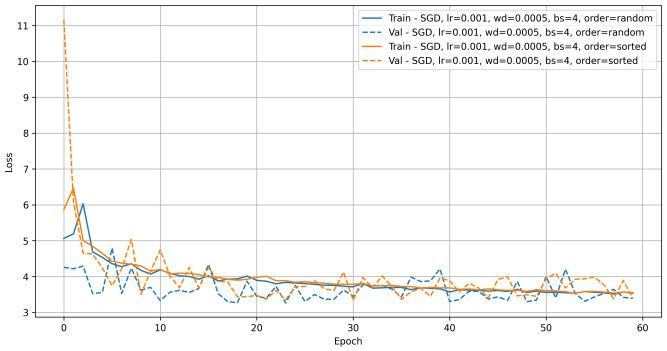
APPENDIX I



(a) *Experiment* 1: *SGD*, varying batch size and weight decay.



(b) *Experiment* 2: *AdamW*, varying batch size.



(c) *Experiment* 3: *SGD*, varying data order.

Fig. 3: Learning curves for training and validation sets across the three experiments. The second plot highlights a sub-optimal convergence of the *AdamW* optimizer, which clearly over-fits on training data.

Experiment 1	Model 1	33.39 ± 0.021
	Model 2	66.94 ± 0.67
Experiment 2	Model 1	889.5 ± 12.6
	Model 2	78.34 ± 0.19
Experiment 3	Model 1	29.34 ± 0.24
	Model 2	29.30 ± 0.14

TABLE II: Perplexity mean values and standard errors for each one of the trained models, after 60 epochs.

REFERENCES

[1] N. S. Keskar, D. Mudigere, J. Nocedal, M. Smelyanskiy, and P. T. P. Tang, "On large-batch training for deep learning: Generalization gap and sharp minima," 2017. [Online]. Available: https://arxiv.org/abs/1609.04836

[2] S. Hochreiter and J. Schmidhuber, "Flat minima," *Neural Computation*, vol. 9, pp. 1–42, 01 1997.

[3] H. Li, Z. Xu, G. Taylor, C. Studer, and T. Goldstein, "Visualizing the loss landscape of neural nets," *arXiv preprint arXiv:1712.09913v3*, 2017.

[4] L. Dinh, R. Pascanu, S. Bengio, and Y. Bengio, "Sharp minima can generalize for deep nets," *arXiv preprint arXiv:1703.04933*, 2017.

[5] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *International Conference on Learning Representations (ICLR)*, 2019.

[6] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," in *Proceedings of the 26th Annual International Conference on Machine Learning (ICML)*. Montreal, Canada: ACM, 2009, pp. 41–48.

[7] H. Jin, X. Han, J. Yang, Z. Jiang, C.-Y. Chang, and X. Hu, "Growlength: Accelerating llms pretraining by progressively growing training length," 2023. [Online]. Available: https://arxiv.org/abs/2310.00576

[8] H. Pouransari, C.-L. Li, J.-H. R. Chang, P. K. A. Vasu, C. Koc, V. Shankar, and O. Tuzel, "Dataset decomposition: Faster llm training with variable sequence length curriculum," 2025. [Online]. Available: https://arxiv.org/abs/2405.13226