ESILV

Course: AI Algorithms

Automated Model Compression

with Genetic Algorithms on GPT-2

Authors:
GABRIEL MATAR
OCTAVE PEDERGNANA
PAUL PLANTIER
GABRIEL MENDES

December 19, 2025

Contents

List of Figures

1 Introduction

1.1 Why is your chosen facet of LLM optimization a critical problem?

Artificial Intelligence is currently facing a scalability crisis. While Large Language Models (LLMs) like GPT-2 demonstrate exceptional reasoning capabilities, their deployment hits a significant hardware wall: computational cost. This problem is critical for three major reasons:

The Deployment Gap: Model size is growing exponentially, while available memory on standard devices remains limited. Without drastic compression, modern AI remains confined to cloud servers, inaccessible to Edge devices.

Inference Latency: An uncompressed model involves millions of superfluous matrix calculations, inevitably slowing down real-time text generation.

Energy Efficiency: Running dense models is extremely energy-intensive. Optimization is therefore not just a question of speed, but an ecological and economic necessity.

1.2 What are the key challenges specific to this area?

The central challenge of our project lies in the mathematical nature of structural optimization, which differs radically from classical machine learning training:

The Discrete Obstacle (vs. Continuous): Weight training (W) is performed via gradient descent because the parameter space is continuous and differentiable. Conversely, Pruning is a combinatorial and discrete problem: deciding whether a neuron should be kept (1) or removed (0) makes standard gradient calculation impossible.

Non-Convexity of the Search Space: As highlighted by Wang et al., the loss landscape associated with an LLM's architecture is highly non-convex. Classical optimization methods, known as "greedy" approaches, tend to converge prematurely towards suboptimal local minima.

Structural Heterogeneity: Not all layers of a Transformer share the same sensitivity. A naive approach (uniform compression) often destroys the syntactic capabilities of the initial layers. The challenge, therefore, is to automatically discover the optimal compression strategy without human intervention.

2 Application of AI Techniques in your Chosen Area

Our approach is grounded in a rigorous review of key research papers. These studies allowed us to construct a theoretical framework justifying the shift from classical optimization methods to evolutionary algorithms for model compression.

2.1 The Theoretical Foundation: LLMs & Evolutionary Algorithms

The theoretical justification for our project is primarily anchored in the comprehensive study by Wang et al. (2025), titled *"When Large Language Models Meet Evolutionary Algorithms: Potential Enhancements and Chal-lenges"*. This work is fundamental as it establishes a precise taxonomy of the optimization challenges unique to LLMs. The authors demonstrate that the architectural search space of a neural network is not only discrete but also riddled with local optima that defeat gradient-based methods.

Our in-depth analysis of this paper allowed us to understand that Evolutionary Algorithms (EAs) are not merely an alternative, but often the only viable method for problems where the objective function (in this case, the network structure) is non-differentiable. Wang et al. further

highlight the capacity of EAs to perform a Global Search, enabling the discovery of counter-intuitive architectures that human intuition or greedy search methods would never have explored. This finding validated our initial technical choice.

2.2 The Methodology: Layer-Wise Sensitivity and Genetic Encoding

To operationalize this theory, we relied on the joint works of Liu et al. (OptiShear) and the team behind EvoP (2025). The OptiShear paper provides a major analytical contribution. Through sensitivity analyses, the authors empirically prove that the layers of a Transformer exhibit strong functional heterogeneity. Concretely, they demonstrate that the initial layers (dedicated to syntactic feature extraction) and the final layers (responsible for decision making) are extremely sensitive to parameter removal. Conversely, intermediate layers exhibit high redundancy.

This critical analysis renders the standard "Uniform Pruning" approach obsolete. To exploit this discovery, the EvoP paper provided us with the necessary algorithmic framework. Rather than pruning manually, EvoP proposes encoding the network configuration as a digital genome. The major outcome of this work is the demonstration that a genetic algorithm can autonomously learn to allocate different sparsity budgets to each layer. By merging the sensitivity analysis of Liu et al. with the genetic encoding of EvoP, we were able to design an agent capable of sculpting the model while respecting its internal hierarchy.

2.3 The Contrast: Structural Sparsity vs. Quantization

It is also imperative to delineate our research scope regarding other dominant compression techniques. The study by Lang et al., *"A Comprehensive Study on Quantization Techniques for Large Language Models"*, served as a contrast study in our literature review. This paper details methods for reducing the numerical precision of weights (e.g., shifting from 16-bit to 4-bit).

Our analysis of this text allowed us to clearly distinguish two orthogonal axes: precision reduction (Quantization) and dimension reduction (Pruning/Sparsity). Although quantization is effective for reducing memory footprint, Lang et al. show that it does not alter the structural complexity of the computational graph. By choosing Pruning, we address a different problem: modifying the very topology of the network.

2.4 The Future and Critical Perspective: Self-Guided Optimization

Finally, we analyzed the paper *"Towards Efficient Automatic Self-Pruning of Large Language Models"* (Self-Pruner, 2025). While our method utilizes an external agent (the genetic algorithm), the Self-Pruner employs the model's internal signals.

The analysis of this paper highlights the inherent limitations of our own approach, particularly the high computational cost associated with evaluating an entire population. However, it also underscores the robustness of our choice: where the Self-Pruner risks being biased by local gradient approximations, our Genetic Algorithm retains a global perspective, making it more capable of avoiding local minima, albeit at the cost of higher computation time.

3 Implementation of a Selected Technique

3.1 Specific Sub-problem Addressed

We chose to focus on the problem of Heterogeneous Layer-Wise Sparsity Allocation.

In the context of model compression, the standard approach often involves imposing a unique compression rate (e.g., 50%) across the entire network. However, as demonstrated in our literature review, this approach ignores the variable sensitivity of different layers. The sub-problem we address is, therefore, a combinatorial optimization problem: determining the optimal pruning rate vector $\alpha = [\alpha_1, \alpha_2, ..., \alpha_{12}]$ such that model degradation is minimized for a given target size.

3.2 Technique Implemented: Genetic Algorithm with Aggressive Elitism

To solve this problem, we implemented a Genetic Algorithm (GA) specifically calibrated for rapid convergence. We selected this technique because the search space is discrete, rendering gradient-based methods ineffective. Furthermore, since the evaluation of each individual (calculating perplexity on a complete model) is computationally expensive, we had to adapt the genetic operators to maximize search efficiency with a reduced number of generations.

3.2.1 Encoding and Evaluation

Each individual in our population is represented by a vector chromosome of dimension 12 (corresponding to the 12 layers of GPT-2). Each gene is a value representing the percentage of weights to remove in the corresponding layer.

The evaluation of solution quality relies exclusively on Perplexity. We aim to minimize this metric, which measures the model's uncertainty during text prediction. Lower perplexity indicates better conservation of the model's cognitive capabilities after pruning.

3.2.2 Evolutionary Strategy: Strong Elitism

Faced with limited computational resources and the need to obtain high-quality results in few iterations, we opted for a selection strategy based on strong elitism.

Unlike classical approaches that renew a large portion of the population, we systematically retain the top 50% best individuals from the previous generation. These "elites" pass directly to the next generation without modification. This approach guarantees monotonic convergence.

3.2.3 Reproduction and Bounded Mutation

The remaining 50% of the new generation is generated to introduce diversity and explore the search space:

Crossover: Offspring are created from randomly selected parents.

Mutation: To refine solutions, we apply mutation to the offspring. This mutation is intentionally bounded: it applies a maximum variation of 10% to a gene's pruning rate. This prevents breaking the promising structures discovered by parents while allowing for precise layer-by-layer compression adjustments.

4 Empirical Evaluation

4.1 Experimental Protocol

Our experimental protocol was conducted in two phases:

Baseline: Application of uniform pruning (identical rate for all layers) ranging from 0% to 55% to establish a standard degradation curve.

Evolutionary Run: Execution of the genetic algorithm with a global sparsity target constraint of 45%.

4.2 Results & Analysis

4.2.1 Comparative Performance: Avoiding the "Cliff of Death"

The analysis of the performance curve (see Figure 1) reveals a critical limitation of the uniform approach.

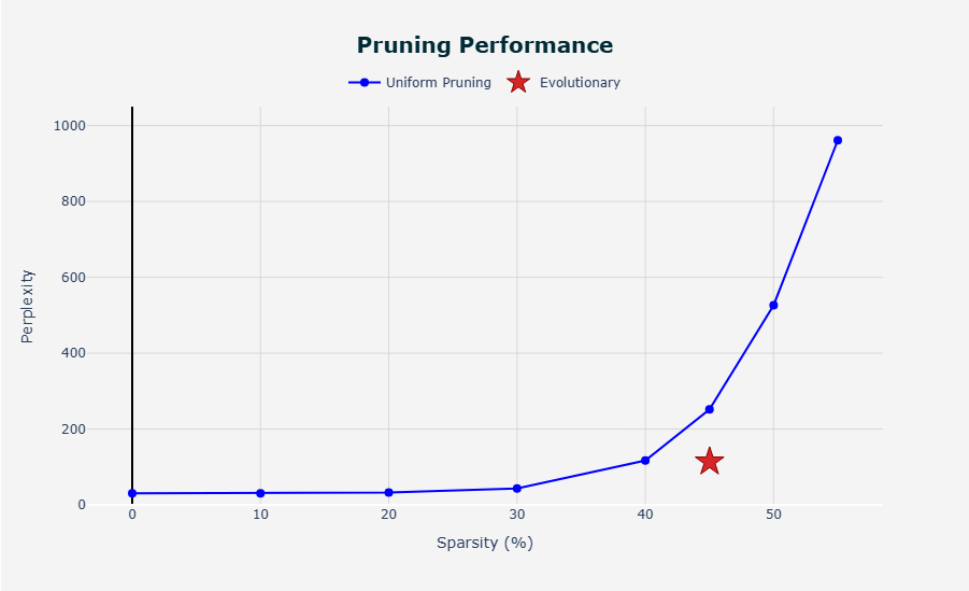

Figure 1: Pruning Performance: Uniform Baseline (Blue) vs. Evolutionary Strategy (Red Star).

- **The Uniform Rupture:** As illustrated by the blue curve, uniform pruning is tolerable up to approximately 30-35%. However, beyond this threshold, we observe an exponential degradation in perplexity. At 45% sparsity, the uniform method reaches a perplexity exceeding 250, rendering the model incoherent and unusable. This phenomenon corresponds to what Paper 1 (*OptiShear*) terms the "Cliff of Death".
- **Evolutionary Stability:** Our method (represented by the red star) successfully maintains a perplexity of approximately 120 at this same 45% rate.

Conclusion: For an identical memory budget, the heterogeneous allocation discovered by the AI preserves information significantly better than homogeneous reduction. This confirms that redundancy in GPT-2 is not uniformly distributed.

4.2.2 Structural Discovery: The Layer-Wise Profile

Examining the final pruning strategy (see Figure 2) allows us to understand where the algorithm chose to cut.

The resulting profile follows a global trend resembling an "inverted U":

Protected Layers (L0, L2, L4): The algorithm applied a low pruning rate ($< 35\%$) to the initial layers. This aligns with *OptiShear*'s theory: these layers process input embeddings and low-level syntax. Altering them destroys the model's ability to "read" the input.

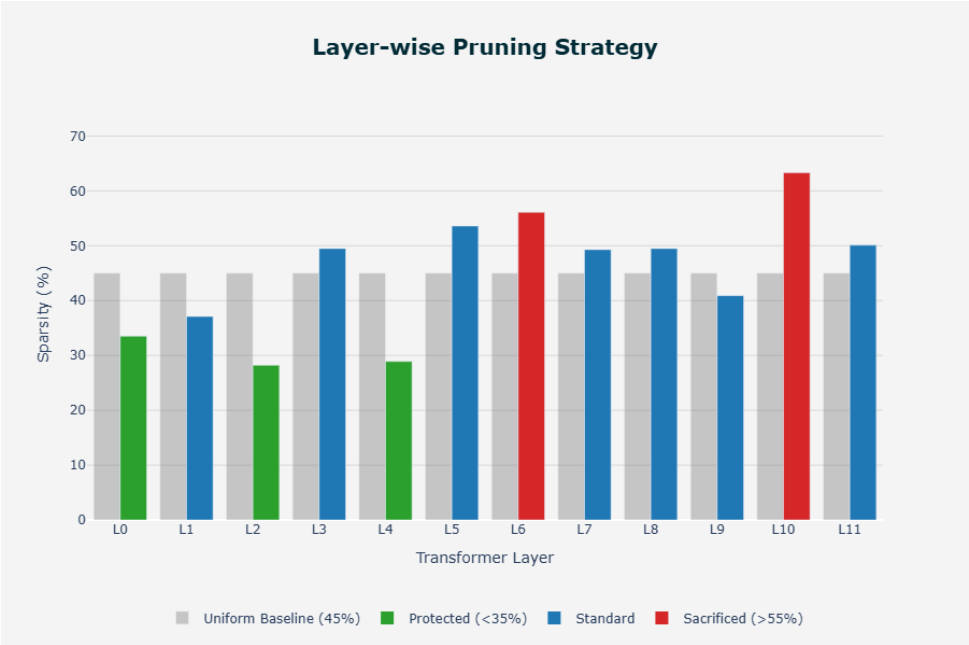

Figure 2: Layer-wise Pruning Strategy discovered by the Genetic Algorithm.

Sacrificed Layers (L5, L6, L10): Intermediate layers undergo stronger compression. The algorithm identified that these Feed-Forward blocks contain high semantic redundancy or "dead" neurons that can be removed without major impact.

4.2.3 Critical Discussion: Noise and Convergence Limits

It is crucial to note that the resulting profile is not perfectly smooth. We observe structural irregularities: for instance, layer L3 is pruned at nearly 50% while being surrounded by protected layers (L2 and L4). Similarly, layer L10 undergoes a very strong suppression peak compared to L9 and L11.

These irregularities are not intrinsic characteristics of GPT-2, but artifacts linked to our implementation constraints:

- **Stochastic Convergence:** Our genetic algorithm involves randomness (mutation/crossover). With a limited number of generations (due to restricted computational resources), the algorithm did not have time to smooth out these local disparities.
- **Selection Pressure (Elitism):** Our choice of strong elitism (keeping 50% of parents) favored rapid convergence towards a "globally good" but "locally imperfect" solution. The algorithm found a satisfactory local minimum but could not finely tune the relationship between L2, L3, and L4, for example.

Trade-offs: Despite this structural noise, the final result is valid: the genetic algorithm's model largely outperforms the perfect uniform method. This demonstrates that an imperfectly optimized structure is better than an arbitrarily uniform structure.

5 Conclusion

5.1 Main Trends in AI for LLM Optimization

LLM optimization is rapidly evolving from static approaches toward AutoML (Automated Machine Learning). The prevailing trend is no longer to apply fixed heuristic rules—such as uniform pruning—but to consider the network architecture as a dynamic system capable of adapting to specific constraints.

5.2 Key Lessons Learned

This project has taught us two fundamental lessons regarding the nature of neural networks:

The Combinatorial Nature of Optimization: We understood that architectural optimization is structurally distinct from weight training. While gradient descent operates effectively in continuous spaces, structural search requires global exploration methods, such as genetic algorithms, to navigate a discrete and combinatorial space.

Layer Heterogeneity: The failure of our uniform baseline proved that treating all Transformer layers equally is a fundamental error. The efficiency of a compressed model relies entirely on its ability to distinguish between layers that must be protected (syntactic/input) and layers that can be aggressively compressed (redundant/semantic).

5.3 Future Research Recommendations

To address the high computational cost inherent in our evolutionary approach, we recommend pursuing a hybrid strategy.

Ideally, future work should combine the robustness of the Genetic Algorithm (for global architectural exploration) with techniques from the Self-Pruner (Paper 5). Using the model's internal signals (such as gradients or activation magnitudes) to locally guide mutations would likely accelerate convergence and smooth out the structural irregularities observed in our empirical results.

References

[1] Shuqi Liu, Bowei He, Han Wu, Linqi Song. *Beyond One-Size-Fits-All Pruning via Evolutionary Metric Search for Large Language Models*. ArXiv preprint arXiv:2502.10735, 2025.

[2] Chao Wang, Jiaxuan Zhao, Licheng Jiao, Lingling Li, Fang Liu, Shuyuan Yang. *When Large Language Models Meet Evolutionary Algorithms: Potential Enhancements and Challenges*. ArXiv preprint arXiv:2401.10510, 2024 (Updated 2025).

[3] Jiedong Lang, Zhehao Guo, Shuyu Huang. *A Comprehensive Study on Quantization Techniques for Large Language Models*. ArXiv preprint arXiv:2411.02530, 2024.

[4] Shangyu Wu, Hongchao Du, Ying Xiong, Shuai Chen, Tei-Wei Kuo, Nan Guan, Chun Jason Xue. *EvoP: Robust LLM Inference via Evolutionary Pruning*. ArXiv preprint arXiv:2502.14910, 2025.

[5] Weizhong Huang, Yuxin Zhang, Xiawu Zheng, Fei Chao, Rongrong Ji. *Towards Efficient Automatic Self-Pruning of Large Language Models*. ArXiv preprint arXiv:2502.14413, 2025.