ESANN 2024 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 9-11 October 2024, i6doc.com publ., ISBN 978-2-87587-090-2. Available from http://www.i6doc.com/en/.

Towards Explainable Evolution Strategies with Large Language Models

Jill Baumann and Oliver Kramer

Department of Computing Science
Carl von Ossietzky Universität Oldenburg
26111 Oldenburg, Germany
`jill.baumann@uni-oldenburg.de`

Abstract. This paper introduces an approach that integrates self-adaptive Evolution Strategies (ES) with Large Language Models (LLMs) to enhance the explainability of complex optimization processes. By employing a self-adaptive ES equipped with a restart mechanism, we effectively navigate the challenging landscapes of benchmark functions, capturing detailed logs of the optimization journey. The logs include fitness evolution, step-size adjustments and restart events due to stagnation. An LLM is then utilized to process these logs, generating concise, user-friendly summaries that highlight key aspects such as convergence behavior, optimal fitness achievements, and encounters with local optima. Our case study on the Rastrigin function demonstrates how our approach makes the complexities of ES optimization transparent. Our findings highlight the potential of using LLMs to bridge the gap between advanced optimization algorithms and their interpretability.

1 Introduction

Nowadays, Large Language Models (LLMs) excel at solving diverse tasks by leveraging the attention mechanism, which enables them to recognize long-range dependencies in texts. Explainable AI (XAI) states an AI model, its impact and potential biases helping to distinguish transparency in AI systems decisions. This approach aims to enhance AI systems' understandability by utilizing an LLM to generate user-friendly explanations of the Evolution Strategies (ES)[1] optimization process, which we refer to as Explainable ES (XES). XES can be used to draw conclusions from the LLM's explanations for potential hyperparameter tuning, such as adjusting the step size or other parameters in our evolutionary strategy use case. Through a detailed case study on the Rastrigin function, we demonstrate the efficacy of our approach in enhancing the understanding of ES, underscoring its potential implications for the field of XAI.

In Section 2, we explore the landscape of ES and XAI. In Section 3 we detail our framework XES. Following this, Section 4 presents an empirical analysis as a case study using the Rastrigin function. Finally, Section 5 summarizes our key discoveries and outlines issues for future research.

2 Related Work

XAI employs strategies to render AI systems' decisions transparent. Intrinsically interpretable models like decision trees offer direct clarity, while complex models may be simplified for better understanding. Post-hoc algorithms, such as SHAP [5] and LIME [6], provide insights by highlighting influential features or input parts but require white box access and are computationally intensive. Example-based explanations, including counterfactuals, clarify model decisions by presenting alternative scenarios. Interactive tools and comprehensive documentation further bridge the gap between AI complexity and user comprehension, enhancing transparency and trust.

Tools like FaAIr, HELM, and ANE demonstrate how LLMs comes to responsible decisions. Zhao et al. [7] categorize various explainability techniques for LLMs. Liu et al. [4] introduce a ChatGPT-aided explainable framework for medical image diagnosis combining CLIP and ChatGPT that increases the zero-shot image classification accuracy on five medical datasets. Kroeger et al. [3] utilize the in-context learning capability of an LLM to explain the predictions of other models. Experiments conducted on real datasets demonstrate that this approach is competitive with state-of-the-art methods. Chuang et al. [2] explore the potential of LLMs as explainers by introducing a framework designed to provide faithful explanations, accurately capturing the prediction behaviors of LLMs.

3 XES: ES with LLM Explanations

XES is based on LLM-generated explanations summarizing ES optimization runs. In our XES variant we employ a self-adaptive ES augmented with a restart mechanism to navigate complex optimization landscapes. The algorithm dynamically adjusts its mutation step sizes based on evolutionary history, enhancing exploration and exploitation capabilities. This self-adaptation mechanism fine-tunes the mutation parameters in response to the algorithm's performance, enabling a more efficient search process.

To counteract the potential for stagnation the algorithm contains a strategic restart mechanism. This mechanism activates when the algorithm detects a prolonged lack of fitness improvements, indicating a local optimum. Upon activation, the algorithm reinitializes its state, including the population and mutation parameters, thereby diversifying the search space and increasing the probability of escaping suboptimal regions.

Critical to our approach is the implementation of a detailed logging system that records the progression of the optimization process. The log file captures key metrics such as the development of fitness values across iterations, the evolution of mutation step sizes, stagnation events and subsequent restarts, offering a comprehensive view of the algorithm's dynamics.

An LLM is employed to transform the extensive data recorded in the log-file into concise, user-friendly narratives, describing the optimization process. The

LLM is prompted with prompts structured according to four different prompt strategies like Zero-Shot, Few-Shot, Chain-of-Thought (CoT), and Few-Shot CoT Prompting. To emphasize critical aspects of the ES's performance, prompts are manually structured according to their prompt strategy, as detailed in Table 1. Due to the way an LLM works, it is not guaranteed that the LLM will provide exactly the same response for the same prompt.

Prompt strategy	Prompt
Zero-Shot	*”Provide a summary of insights derived from the log file below, delimited by triple backticks, representing a minimization problem. Concentrate on best, worst, and mean fitness values, as well as convergence behavior and local optima. Ensure accurate identification of the best and worst fitness values and calculate the mean fitness over all iterations. Remember, in this context, the lowest value denotes the best fitness, while the highest value represents the worst fitness.”* ```{*log file*}```
Few-Shot	*Zero-Shot Prompt* {*Correct Answer*} *Zero-Shot Prompt*
CoT	*”Provide a summary of insights derived from the log file below, delimited by triple backticks, representing a minimization problem. Ensure accurate identification of the best and worst fitness values and calculate the mean fitness over all iterations. Remember, in this context, the lowest value denotes the best fitness, while the highest value represents the worst fitness.* *1. Begin by extracting the best fitness value achieved.* *2. Next, extract the worst fitness value achieved during the optimization process.* *3. Calculate the average fitness value across all iterations.* *4. Analyze the convergence behavior, including trends or patterns indicating convergence behavior.* *5. Lastly, determine if there are any instances of reaching local optima or encountering plateaus.”* ```{*log file*}```
Few-Shot CoT	*CoT Prompt* {*Correct Answer*} *CoT Prompt*

Table 1: Prompts according to their prompt strategy.

These instruction guides the LLM to focus on the pivotal elements defining the optimization journey: the development towards convergence, the attainment of optimal fitness values and interactions with local optima, including restarts due to stagnation. This process requires the LLM to view the detailed log data and summarize the most important information in a comprehensible summary.

4 Case Study

In this section we present a case study, which is a practical application of our self-adaptive ES with LLM-generated explanations, utilizing the highly multimodal Rastrigin function to demonstrate the efficacy of our approach.

4.1 Setting

In our study, we deployed a σ-self-adaptive $(\mu + \lambda)$-ES to optimize a 10-dimensional Rastrigin function, a challenging benchmark known for its complex landscape with numerous local minima that requires the detection of stagnation in local optima and the triggering of restart mechanisms. The algorithm was set to run until either a maximum of 10,000 iterations were reached or a fitness threshold of 10^{-5} was achieved. To monitor and analyze the optimization process, logs were systematically generated every 30 iterations of the Rastrigin function, detailing the fitness values and step sizes. Additionally, instances of stagnation and subsequent restarts, particularly at the initial point $(1, \ldots, 1)$, were recorded to understand the algorithm's behavior in trapping and escaping local optima. Overall, XES was tested with three log files of varying lengths. Log file 1 (”short”) documents 150 iterations, log file 2 (”middle”) spans 420 iterations and log file 3 (”long”) consists of 1260 iterations which encompass the highest number of restarts. Listing 1 presents log file 1. To translate these extensive logs into comprehensible narratives, we employed four different language models, setting the temperature to 0.0 for more deterministic text generation. The LLMs we use are Llama2:70B, Llama3:70B, Mistral 7B and Mixtral 8x7B. The prompts followed one of the four prompt strategies' structures. Each combination of LLM and prompt strategy was repeated ten times per log file. To evaluate the LLM's response, numerical information, such as the best and worst fitness, was automatically extracted and verified whether it was correct. Non-numerical information, such as convergence behavior and local optima, was assessed manually. The total score for a combination of LLM and prompt strategy was determined by assigning one point to each correct statement (best fitness, worst fitness, convergence behavior, local optima) and then normalizing the score to a value between 0 and 1.

```
Iteration 30: Fitness: 1.9899, Step size: 8.5475e-07
Iteration 60: Fitness: 1.9899, Step size: 3.8096e-10
Iteration 90: Fitness: 1.9899, Step size: 5.0931e-10
Iteration 120: Fitness: 1.9899, Step size: 7.7127e-11
Restarting at iteration 149 due to stagnation
Iteration 150: Fitness: 5.0554, Step size: 0.4503
```

Listing 1: Log file 1 (”short”)

4.2 Results

Table 2 shows the results of the experiments conducted w.r.t. the average score achieved across the ten repetitions.

		Zero-Shot	Few-Shot	CoT	Few-Shot CoT
Llama2:70B	*short*	0.76	**1.0**	0.33	**0.98**
	middle	0.45	0.55	0.55	0.35
	long	0.35	0.35	0.38	0.3
Llama3:70B	*short*	0.7	**1.0**	0.53	0.95
	middle	0.7	0.73	0.75	0.73
	long	0.85	**1.0**	0.8	0.88
Mistral 7B	*short*	0.78	0.88	0.7	0.93
	middle	0.5	0.85	0.7	0.38
	long	0.45	0.55	0.5	0.58
Mixtral 8x7B	*short*	0.93	0.98	0.83	0.88
	middle	0.8	**1.0**	0.78	0.65
	long	**0.95**	0.93	**0.9**	0.95

Table 2: Comparison of four different prompt strategies across three log files.

The outcomes of the conducted experiments are largely similar. Notably, Llama2:70B shows lower scores, whereas Mixtral 8x7B, especially with Few-Shot Prompting, outperforms the others, achieving a peak performance value of 1.0. Among the different prompt strategies, Few-Shot Prompting achieves the best average results across all models. For instance Mixtral 8x7B with Few-Shot-Prompting for log file 2 outputs:

"The algorithm initially converged to a local optimum near 3.98, then restarted and converged to a lower optimum near 2, followed by another restart and convergence to the best observed value of 0.0106. Multiple local optima were encountered during the optimization process. The lowest fitness value observed was 0.0106. The highest fitness value observed was 3.9852."

The LLM's summary of the ES run is clear and coherent, detailing key aspects such as best and worst fitness values, and critical events like convergence and stagnation. The inclusion of specific numerical details enhances the summary's precision, providing a transparent view of the ES's performance on the Rastrigin function. However, the analysis could benefit from further context on the significance of step size adjustments and a more explicit evaluation of the strategy's overall success.

XES tends to yield superior results with shorter log files, providing more detailed responses that mention specific iterations and values. In contrast, longer log file responses result in more general information, often referencing iterations and values from the log file's last iterations which is typical for the way an LLM works. A limitation is the parameter context length of an LLM, which may not be exceeded when formulating prompts based on the prompt strategy and the log file's content. Developing techniques to shorten long log files could enhance robustness. The best and worst fitness values can also be extracted deterministically. This approach demonstrates the LLM's capability to extract these values, making it an appropriate alternative for obtaining values and providing user-friendly explanations of a optimization process in an easy manner.

5 Conclusions

Our approach leverages the advanced natural language generation capabilities of LLMs to transform technical optimization logs into accessible explanations, thereby providing the intricacies of the ES optimization process for a broader audience. Looking ahead, we see substantial potential in further enriching this framework. Firstly, by integrating an interactive analysis layer that prompts user inquiries, so we can tailor explanations to individual needs, enhancing user comprehension. Secondly, the insights derived from LLM analyses could be used in agent-based systems capable of automating optimization actions.

References

[1] H. G. Beyer and H.-P. Schwefel. Evolution strategies - a comprehensive introduction. *Natural Computing*, 1:3–52, 2002.

[2] Y.-N. Chuang, G. Wang, C.-Y. Chang, R. Tang, F. Yang, M. Du, X. Cai, and X. Hu. Large language models as faithful explainers. *arXiv preprint*, arXiv:2402.04678, 2024.

[3] N. Kroeger, D. Ley, S. Krishna, C. Agarwal, and H. Lakkaraju. Are large language models post hoc explainers? In *XAI in Action: Past, Present, and Future Applications*, 2023.

[4] J. Liu, T. Hu, Y. Zhang, X. Gai, Y. FENG, and Z. Liu. A chatGPT aided explainable framework for zero-shot medical image diagnosis. In *ICML 3rd Workshop on Interpretable Machine Learning in Healthcare (IMLH)*, 2023.

[5] S. M. Lundberg and S.-I. Lee. A unified approach to interpreting model predictions. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 4765–4774. Curran Associates Inc., 2017.

[6] M. T. Ribeiro, S. Singh, and C. Guestrin. "why should i trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1135–1144. Association for Computing Machinery, 2016.

[7] H. Zhao, H. Chen, F. Yang, N. Liu, H. Deng, H. Cai, S. Wang, D. Yin, and M. Du. Explainability for large language models: A survey. *ACM Trans. Intell. Syst. Technol.*, 15(2):20:1–20:38, 2024.