Data-efficient Elitist Evolutionary Algorithm for Training Neural Networks

Yurui Yang
School of Artificial Intelligence
Sun Yat-sen University
Zhuhai, China
yangyr35@mail2.sysu.edu.cn

Zefeng Chen
School of Artificial Intelligence
Sun Yat-sen University
Zhuhai, China
chenzef5@mail2.sysu.edu.cn

***Abstract*—Currently, optimizers based on gradients dominate the training of neural networks. In contrast to gradient-based optimizers, evolutionary algorithms are population-based optimization algorithms that possess several notable advantages, such as strong parallel capability and the ability to escape from local optima. However, they have gradually been marginalized due to their low efficiency. In this paper, we propose an approach called the data-efficient elitist evolutionary algorithm (DeiEA) to address the low-efficiency issue of evolutionary-based optimizers. DeiEA utilizes a mini-batch of data, replacing the entire training dataset, to evaluate individuals in the population. Additionally, by carefully designing the procedure for producing offspring using three effective strategies, the proposed DeiEA can achieve faster convergence while maintaining population diversity.**

Keywords—Neural network; machine learning; evolutionary algorithms; optimizer

I. Introduction

With the growth of computational power, machine learning (ML) has been rapidly evolving in the last decade [1]. Usually, the gradient-based optimizers are adopted to train the neural networks in the field of ML. However, they have significant limitations. On one hand, gradient descent methods may get trapped in local optima, making it difficult to achieve the global optimum. On the other hand, they are vulnerable to saddle points or areas of long, gradual error plateaus [2]. In contrast to gradient-based optimizers, nature-inspired evolutionary algorithms (EAs) have lower solving efficiency and require larger computational resources. However, EAs do not rely on gradients, which allows them to partially overcome the aforementioned problems [3]. Furthermore, EAs possess excellent intrinsic parallelism compared to gradient-based optimizers [4]. Therefore, it is crucial to not overlook the research on EAs and to strive for improved solving efficiency in EAs.

In this paper, we introduce the data-efficient elitist evolutionary algorithm (referred to as DeiEA). Inspired by the key idea of LEEA [5], DeiEA utilizes a mini-batch of data instead of the entire training dataset to evaluate individuals in the population. This approach significantly reduces the evaluation overhead in traditional EAs. Additionally, we carefully design a procedure for producing offspring using three effective strategies, enabling DeiEA to converge faster while maintaining population diversity. Preliminary experimental results demonstrate the nice performance of DeiEA, which competes favorably with gradient-based optimizers. Given the notable parallelism characteristics of population-based EAs, the performance of DeiEA is expected to be further improved with sufficient hardware resources.

II. Proposed Approach

The overall process of the proposed DeiEA framework is illustrated in Figure 1. The framework is adaptable to training various types of neural networks. DeiEA primarily follows the fundamental process of EAs. Notably, in the initial stage and at each generation, a mini-batch is constructed based on the complete training dataset, and this mini-batch is employed to evaluate the individuals in the population.

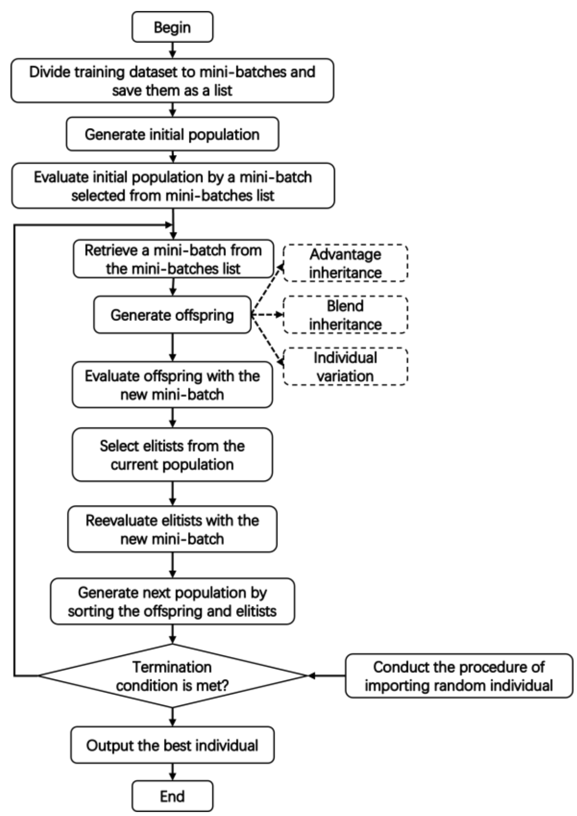


Figure 1：Process of proposed DeiEA

When producing offspring, three strategies are employed: 'advantage inheritance' (where the child inherits more from

Corresponding author: Zefeng Chen.
This research is supported by the National Natural Science Foundation of China under Grant 62206313 and 62232008.

the superior parent), 'blend inheritance' (where the genes of parents are fused during inheritance), and 'individual variation'. To mitigate the randomness introduced by using a mini-batch to evaluate the population, we have designed strategies that enable offspring to inherit fitness from their parents. For instance, in 'advantage inheritance', the fitness of offspring is defined as follows.

$$f' = \frac{\alpha_1 * f_1 + \alpha_2 * f_2}{2} * (1 - \beta) + f \quad (1)$$

Where f' is the final fitness of child, f_1 and f_2 is the fitness of parents, f is the initial fitness of the child evaluated on the base of the current mini-batch. α_i refers to the percentage of inheritance from each parent and β is the decay factor. Certain elitists from the current population are preserved to compete with the offspring. It is important to note that directly comparing parents and offspring would be unfair due to the fitness inheritance mechanism. Therefore, the fitness of parents needs to be adjusted through re-evaluation.

At regular intervals, an additional individual, that is randomly generated, is introduced to diversify the gene pool. Subsequently, a new population is formed to proceed with the next generation. Once the maximum number of generations is reached, the individual with the highest fitness is selected as the final result.

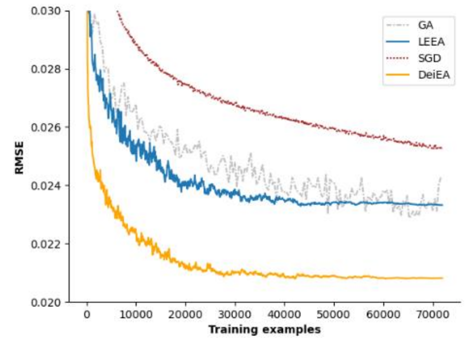
III. Experiments

We conducted experiments to evaluate the effectiveness of the proposed DeiEA as an alternative optimizer for training neural networks. The experiments were conducted on two types of tasks: a regression task based on the Boston House dataset, as well as a classification task based on the Default of Credit Card Clients dataset. These datasets were obtained from Kaggle. In the experiments, DeiEA was compared with other optimization algorithms, including Genetic Algorithm (GA), Limited Evaluation Evolutionary Algorithm (LEEA), and Stochastic Gradient Descent (SGD). Neural networks optimized by different optimizers are constructed with the same structure and size. For the regression and classification tasks, we respectively adopt the root mean squared error (RMSE) and accuracy to serve as the performance metric.

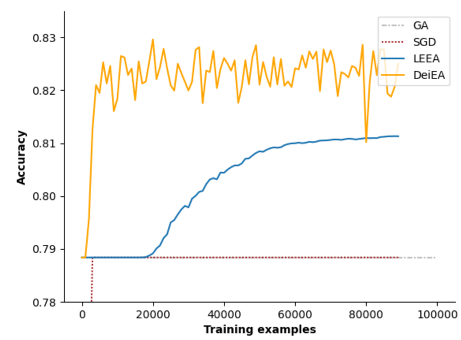
The results of the experiments are presented in Figure 2. As can be seen, DeiEA exhibits a clear advantage over the traditional GA by significantly reducing the computational requirements per generation. In comparison to LEEA, DeiEA achieves better metric values, and their performance difference is becoming larger along with the evolution. However, DeiEA exhibits more fluctuations and lacks the smoothness observed in LEEA, particularly in the later stages. Overall, DeiEA outperforms LEEA. When compared to the gradient-based SGD optimizer, DeiEA achieves faster convergence speed and attains better metric values.

IV. Conclusions

This paper introduces the DeiEA algorithm and demonstrates its performance improvement over traditional EAs. We also highlight its superior ability to search for optimal solutions compared to LEEA.



(a) Boston House



(b) Default of Credit Card Clients

Figure 2: Performance curves of all algorithms on two datasets.

The proposed DeiEA enhances the population's search capability by carefully designing the offspring generation mechanism. This design enables the population to efficiently explore small regions where viable solutions may exist without prematurely converging. Furthermore, the regular introduction of random individuals empowers DeiEA with a stronger ability to escape local optima.

DeiEA represents an advancement in evolutionary algorithms. Looking ahead, it would be worthwhile to explore how to further enhance the efficiency of EAs by incorporating more advanced mechanisms such as novelty search, especially when applying EAs to train large-scale neural networks. This area presents an avenue for future research and investigation.

V. Reference

[1] Geoffrey E. Hinton, Simon Osindero, and Yee-Whye, “A Fast Learning Algorithm for Deep Belief Nets,” Neural Computation, vol. 18, no. 7, pp. 1527–1554, 2006

[2] Sebastian, Ruder, An overview of gradient descent optimization algorithms. [Online]. https://doi.org/10.48550/arXiv.1609.04747

[3] Agoston E. Eiben, Jim Smith, “From evolutionary computation to the evolution of things,” Nature, vol. 521, no. 7553, pp. 476–482, 2015

[4] Forrest, Stephanie. "Genetic algorithms: principles of natural selection applied to computation," Science, vol. 261, no. 5123, pp. 872-878, 1993

[5] Gregory Morse and Kenneth O. Stanley. “Simple Evolutionary Optimization Can Rival Stochastic Gradient Descent in Neural Networks,” in Proceedings of the Genetic and Evolutionary Computation Conference, 2016, pp. 477-48