OACE: Multi-criteria Assessment of Assertiveness and Cost for Deep Learning Models

Lyanh Vinicios Lopes Pinto
Federal University of Pará, Institute of Technology
Belém, Pará
lyanh.pinto@itec.ufpa.br

Marcos Cesar da Rocha Seruffo
Federal University of Pará, Institute of Technology
Belém, Pará
seruffo@ufpa.br

ABSTRACT

The rapid expansion of Deep Learning (DL) has intensified the need for more rigorous evaluation methodologies for implementing solutions in real-world scenarios. In view of this, this work presents Optimized Assertiveness-Cost Evaluation (OACE), a method based on Multi-Criteria Decision Making that integrates assertiveness and computational cost criteria into a single parameterizable function, aiming to systematize the evaluation of DL models. To demonstrate its effectiveness, an experiment was conducted with Random Walk and the CIFAR-10 database, evaluating five DL architectures in a balanced scenario. The findings identified MobileNetV2 as the best model for the defined scenario, with a score of $S_\phi(m) = 0.9541$, surpassing MobileNetB0 by 38%, due to its superior efficiency.

KEYWORDS

Deep Learning, Assertiveness, Computational Cost, Multi-Criteria Methods, Performance Evaluation

1 INTRODUCTION

The advancement of Deep Learning (DL) models has caused a transformation in several areas [11]. The success of these technologies applicability was motivated by their ability to optimize processes, offering effective solutions to complex and routine problems. As a result, there has been an exponential increase in the availability of models for implementation in a wide variety of scenarios [12]. Given this variety of options, the successful implementation of any DL solution has become dependent on a careful decision-making process, capable of guiding the choice of the most appropriate model for the demands and constraints of each context.

The increase in the diversity of DL models has intensified a recurring dilemma for researchers: "How to evaluate in order to carefully choose the most appropriate model for a given application context?". The answer requires the analysis of several criteria, two of which stand out: assertiveness, which reflects the model's ability to make correct predictions, and computational cost, which quantifies the consumption of hardware resources [7].

The balance between these two factors is a critical point in the DL ecosystem [6]. In applications that prioritize error minimization, such as medical diagnostics, complex and costly models are acceptable. On the other hand, in systems with limited hardware, such as mobile devices, efficiency is key, requiring the use of lightweight architectures. Managing this trade-off between predictive performance and efficiency is therefore central to developing effective solutions that are appropriate for their deployment environment.

Despite the importance of this trade-off, the evaluation of DL models is still conducted manually, exhaustively, and subjectively [8]. This approach is usually performed by analyzing each metric in isolation. This practice makes the evaluation process increasingly costly, as its complexity grows proportionally to the number of models and criteria evaluated, a challenge that makes more robust analyses unfeasible. The absence of systematic approaches results in inconsistent choices, revealing a gap in the literature for methods that integrate conflicting criteria in a structured manner [5].

This paper presents the Optimized Assertiveness-Cost Evaluation (OACE) method, a holistic approach to evaluating DL models for different contexts. The method is based on Multi-Criteria Decision Making (MCDM), integrating assertiveness and computational cost metrics into a single parameterizable objective function [16]. This study innovates through the MCDM approach and evaluation with an experiment on the CIFAR-10. The experiment focuses on a scenario of balance between assertiveness and cost criteria in order to determine the most balanced architecture.

This article contributes (a) with the proposal of a new MCDM method for the holistic evaluation of DL models and (b) with an experiment to analyze the performance of different architectures in a scenario of balance between assertiveness and cost.

2 RELATED WORK

The literature on DL model evaluation highlights assertiveness as a fundamental criterion for system reliability. In [3], it is indicated that, in critical applications such as surveillance, evaluation should go beyond accuracy, emphasizing the need for metrics that better capture the predictive robustness of models.

Concurrently, computational cost is recognized as an equally important factor, especially in contexts with limited resources. Studies such as [2] and [14] demonstrate that predicting and optimizing execution time and memory usage are essential for application efficiency and scalability. Tools such as DNNAbacus, which estimate the costs of deep networks, exemplify the community's effort to quantify and manage the impact of models on hardware.

Despite the recognition of these criteria, there is a gap in how they are integrated. Most studies analyze them separately, treating the metrics in isolation [8]. Although MCDM methods, such as Analytic Hierarchy Process (AHP) and Weighted Sum Model (WSM), have proven effective in selecting software based on multiple indicators, their application for an integrated evaluation of DL models, which weighs assertiveness and cost, is under-explored [9].

In: V Concurso de Trabalhos de Iniciação Científica (CTIC 2025). Anais Estendidos do XXXI Simpósio Brasileiro de Sistemas Multimídia e Web (CTIC'2025). Rio de Janeiro/RJ, Brasil. Porto Alegre: Brazilian Computer Society, 2025.

ISSN 2596-1683

This paper proposes the OACE method to fill the identified gap. The proposal's unique feature lies in its MCDM modeling, which systematizes the weighting of conflicting criteria. By integrating multiple assertiveness and cost metrics into an adjustable objective function, OACE offers a systematic and adaptable evaluation of DL models, overcoming the limitations of traditional approaches.

3 PROPOSED METHOD

The methodology is based on three steps, starting with Subsection 3.1 with the formulation of the problem and the definition of the method. Next, Subsection 3.2 describes the integration of RW and, finally, Subsection 3.3 details the algorithm solution of the study.

3.1 Formulation and Definition

Developing DL models that operate effectively on limited resources and are highly assertive is a significant challenge. The problem arises when trying to balance metrics while maximizing reliability and minimizing the cost of the models. Thus, the formulation presented in Equation 1 is adopted.

The mathematical modeling of the method employs a central optimization function, $S_\phi(m)$, which is based on two essential components: the assertiveness function ($A(m)$) and the cost function ($C(m)$). Both components, with their inherent constraints, are designed based on the principles of WSM, a fundamental MCDM technique for aggregating multiple criteria into a single score.

$$\begin{aligned} max_m \quad & : S_\phi(m) = \lambda \cdot A(m) + (1-\lambda) \cdot C(m) \\ subject\ to \quad & : A(m) = \sum_i w_a^i \cdot a_i(m) \\ & \quad C(m) = \sum_j w_c^j \cdot c_j(m) \\ & \quad m \in M,\ (models\ set) \\ & \quad 0 \le \lambda \le 1 \\ & \quad 0 \le w_a^i, w_c^j \le 1,\ \forall i, j \in \mathbb{N} \end{aligned} \tag{1}$$

The function $A(m)$ evaluates the predictive capacity of the model, calculated by the weighted sum of its normalized metrics ($a_i(m)$), each with its weight of importance w_a^i. The cost function $C(m)$ quantifies the consumption of computational resources, aggregating its normalized cost metrics $c_j(m)$ with their respective weights w_c^i. The final score $S_\phi(m)$ reflects the balance between these components, adjusted by the weighting factor λ, a parameter in the range [0,1]. A value of λ close to 1 prioritizes assertiveness $A(m)$, while a value close to 0 emphasizes low cost $C(m)$.

This formulation aligns perfectly with MCDM modeling. According to Yang (2020), an MCDM problem, in the context of OACE, is formally defined by four essential components. First, the alternatives correspond to the set of candidate architectures $M = \{m_1, ..., m_n\}$ that need to be evaluated. The criteria $N = \{a_1, ..., a_p, c_1, ..., c_q\}$ are the performance metrics used to judge these alternatives, divided into assertiveness criteria (a_k) and cost criteria (c_l).

The relative importance of the criteria is established by a set of weights $w = \{w_1^a, .., w_p^a, w_1^c, .., w_q^c\}$, obtained objectively via AHP, an MCDM that hierarchizes and derives weights from paired comparisons. The objective function $S_\phi(m)$ operates as a decision function, consolidating the performance of each alternative by the weighted criteria, generating a score for each architecture. The parameter λ is the weight that allows adjusting the priority between maximizing assertiveness or minimizing cost, making OACE flexible to optimize model selection in different scenarios.

3.2 Random Walk

To understand the methodology, it is crucial to consider the Random Walk (RW) algorithm. Developed by [15], RW describes a process where an agent moves randomly in a multidimensional space, with no memory of previous steps. Its relevance lies in the exploration of high-dimensional spaces, common in DL problems.

RW is used to explore the search space when training architectures with varying configurations, adjusting parameters such as learning rate and the selected architecture to create different training conditions. It is important to note that RW does not aim to optimize parameter configuration, but rather to generate distinct scenarios to evaluate the applicability of the OACE method.

3.3 Solution Algorithm

The process, detailed in Algorithm 1, is executed for T iterations. Initially, the dataset D, the set of models M, and a base seed (base_seed) are defined to ensure reproducibility and fair comparison. The assertiveness (a_i) and cost (c_j) metrics are established as decision criteria. The relative importance of these criteria is then quantified using AHP (Subsection 3.1), allowing the derivation of weights (w_i^a and w_j^c) via pairwise comparisons between the metrics.

Algorithm 1 MCDM OACE: Problem Solution

```
Define dataset D, λ, number of iterations T and base_seed
Define alternatives: set of models M, ∀ m ∈ M
Define criteria: assertiveness a_i and cost c_j, where i, j index
Define weights of criteria: w_i^a and w_j^c
Define normalization function: N(x, x^min, x^max) = (x − x^min) / (x^max − x^min)
Initialize the warm-up for cost metrics to fix c_j^min
Initialize a_i^max ← −∞, a_i^min ← ∞, c_j^max ← −∞
for each iteration t from 1 to T do
    Set current_seed = base_seed + (t − 1)
    Set m and lr as parameters and train
    Evaluate a_i, update a_i^max, a_i^min and a_i ← N(a_i)
    Compute A(m) = Σ_i N(a_i) · w_i^a
    Evaluate c_j, update c_j^max and c_j ← N(c_j)
    Compute C(m) = Σ_j N(c_j) · w_j^c
    Compute S_φ(m) = λ · A(m) + (1 − λ) · C(m)
    if S_φ(m) is higher than the current best S_φ then
        Update best model for m and best S_φ value
    end if
    Perform a random walk step to adjust parameters
end for
Store the metrics and return the S_φ and best model
```

Subsequently, the metrics are scaled to the interval [0,1] using iterative normalization $N(x, x^{\min}, x^{\max})$. A warm-up period precedes this normalization, establishing a minimum cost ($c_j^{\min}$) for the basic models, while the maximum and minimum values of the assertiveness metrics ($a_i^{\min}$ and $a_i^{\max}$) and the maximum cost ($c_j^{\max}$) are dynamically updated at each iteration, reflecting RW variations.

The iterative process adjusts the learning rate (lr) and the model (m), where RW advances at each iteration t with the base seed incremented, expanding exploration. The metrics are evaluated and normalized (lines 10 and 12), and the assertiveness ($A(m)$) and cost ($C(m)$) functions are calculated. The score $S_\phi(m)$ is then computed, and the model with the highest score is retained. After T iterations, the algorithm returns to the best model and its respective score.

4 EXPERIMENT

To evaluate OACE, an experiment was conducted using procedures from Algorithm 1. The experiment was conducted in a scenario balancing, with λ = 0.5. Algorithm 1 operated in 50 iterations in three rounds, totaling 150 iterations with fixed seeds to mitigate dependence on randomness and ensure the robustness of the results. The source code for the experiment is publicly available[1].

4.1 Database

The dataset used was CIFAR-10, developed by Krizhevsky (2009)[2], widely used in image classification research. This dataset strikes a balance between size and complexity, allowing different architectures to be tested for different application scenarios.

CIFAR-10 consists of 60,000 color images, evenly distributed across 10 classes: airplane, automobile, bird, cat, deer, dog, frog, horse, ship, and truck. Each class in the dataset contains 6,000 images, 5,000 for training and 1,000 for testing. The preprocessing pipeline standardizes images to 224x224 pixels and applies data augmentation, with horizontal flips and 10° rotations. Finally, images are normalized to ImageNet's standard mean and standard deviation statistics, adjusting the data input to the networks.

4.2 Architecture Selection

The less complex architectures include EfficientNet, known for its optimization through compound scaling, and MobileNetV2 , which uses separable convolutions and residual blocks to significantly reduce parameters and costs. Both demonstrated performance superior to 89% in CIFAR-10, which justifies their inclusion to evaluate OACE in scenarios that demand low cost [17].

In the moderate complexity segment, ResNet-50 innovates with residual connections to handle gradients in deep networks, and InceptionV3 employs Inception modules to optimize resource usage. With accuracy above 77% in CIFAR-10 and moderate cost [4], these architectures allow OACE to be evaluated in intermediate contexts.

Finally, VGG-16 represents the high-complexity architecture, with its deep and uniform 16-layer structure. Although it has a high cost and number of parameters, its performance on CIFAR-10, with accuracy above 70%, makes it relevant for scenarios where assertiveness is a priority and cost constraints are secondary [1].

4.3 Warm-up and Parametrization

After selecting the architectures, Warm-up and Parameterization established the operating conditions for training. The warm-up of the models sought to capture the minimum cost metrics ($c_j^{\min}$) and prepare them for normalization according to Algorithm 1. Each model was run in its basic form on a sample of the CIFAR-10 dataset to determine these values, which were then fixed for normalization $N(c_j, c_j^{\min}, c_j^{\max})$. This ensured that the cost metrics were scaled fairly across the iterations of the process.

The parameterization involved the seed base_seed = [100, 600, 1100], fixed for each of the three rounds. lr was dynamically adjusted at each iteration in the range $[1 \times 10^{-4}, 1 \times 10^{-2}]$. The λ equal to 0.5 reflected an equilibrium scenario. The implementation used Python 3.12 and PyTorch on a server with an NVIDIA A100-SXM4 (40GB VRAM) and 196GB DDR4 RAM. The Adam optimizer, 5-fold cross-validation, and early stopping at 10 epochs were used. Collecting the results took one day of processing.

4.4 Performance Evaluation

The performance evaluation stage applies the OACE method to calculate the optimal solution. For assertiveness, the criteria of precision, accuracy, and recall were adopted, selected for their fundamental relevance in the literature [13]. Using the AHP method for weighting, the weights were defined as: precision (0.731), accuracy (0.188), and recall (0.081), prioritizing the metric that best reflects the proportion of correct predictions and avoids false positives.

In parallel, the cost was evaluated by three metrics: Model Total Parameters (MTP), Time Per Inference (TPI), and Memory Used (MS). These metrics were chosen because they represent the complexity of the model, its hardware requirements, and its response latency. The weights, also obtained via AHP, were: MTP (0.731), TPI (0.188), and MS (0.081), with MTP being prioritized due to its direct impact on execution speed and memory consumption. Details on capturing these weights with AHP can be found in the repository[1].

After capture, the metrics are scaled to the range [0, 1] through iterative normalization, with dynamically updated limits to ensure a fair comparison. With the normalized values, the functions $A(m)$ and $C(m)$ are calculated, resulting in the final score $S_\phi(m)$. At the end of the iterations, the models are ranked by this score to select the best alternative according to the scenario criteria.

5 RESULTS

The results of this study detail the performance of the MCDM OACE and the algorithm for a balanced scenario (λ = 0.5). In order to ensure statistical reliability, the results were consolidated by averaging three rounds of 50 iterations, which mitigates random variations and reinforces the validity of the analysis.

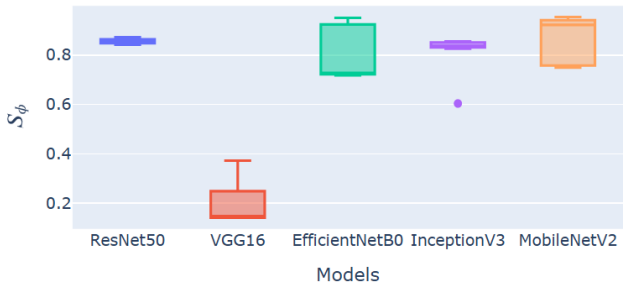

Figure 1: Variation of score $S_\phi(m)$ by trained model

[1]https://github.com/LyanhVini/OACE-randomWalk-monofocal-method
[2]https://www.cs.toronto.edu/ kriz/cifar.html

The distribution of scores (Figure 1), reveals that the lightweight architectures, MobileNetV2 and EfficientNetB0, were the main competitors, although they showed high variability in their results, reflecting sensitivity to the lr. ResNet-50 demonstrated high consistency, with a median of 0.85, while InceptionV3 was penalized for its higher cost. VGG-16 obtained the lowest performance, indicating its unsuitability for this equilibrium scenario.

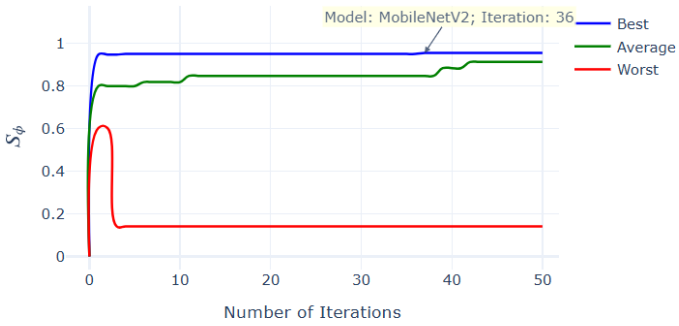

Figure 2: Convergence of Algorithm 1 to the best solution

The convergence of the algorithm (Figure 2) illustrates how the optimal solution was achieved. The best performance curve ("Best") quickly stabilizes at a score $S_\phi(m)$ close to 0.95, indicating that the method found high-performance configurations early in the process. The best global iteration occurred in run 36, in which MobileNetV2 reached a maximum of $S_\phi(m) = 0.9542$. This result was obtained with a maximum of $A(m) = 0.8167$ and a minimum cost, reflected by $C(m) = 1.0$, using a learning rate of $lr = 0.0074$.

Table 1: Best Solutions for the scenario

Model	S_ϕ	$A(m)$	$C(m)$	lr	**i**
MobileNetV2	0.9542	0.8167	1.0000	0.007404	36
EfficientNetB0	0.9505	0.9080	0.9647	0.001522	41
EfficientNetB0	0.9496	0.9043	0.9647	0.001679	44
MobileNetV2	0.9496	0.7982	1.0000	0.010000	03
MobileNetV2	0.9485	0.7940	1.0000	0.006239	22

The analysis of the five best solutions (Table 1) points the superiority of MobileNetV2, which not only obtained the highest score but also appeared most frequently in Table 1. Although EfficientNetB0 proved to be a good alternative, with a score only 0.38% lower, its greater assertiveness ($A(m) = 0.9080$) was not enough to offset its cost, which was 3.53% higher. The advantage of MobileNetV2 is rooted in its consistently minimal cost ($C(m) = 1.0$), reinforcing it as the most effective choice for balance.

6 CONCLUSION

This work introduced OACE, a new MCDM-based method that systematizes the selection of DL models based on the trade-off between assertiveness and computational cost. Its effectiveness was demonstrated in a balanced scenario experiment using the CIFAR-10. The results revealed the superiority of MobileNetV2, with a score $S(m) = 0.9541$, surpassing EfficientNetB0 by 0.38%.

Although the experiment focused on the equilibrium scenario, the methodology allows to be adjusted to meet different priorities, whether to prioritize assertiveness in critical applications or efficiency in limited environments (experiments in other scenarios can be viewed in the repository[1]). The experiment focused on classification, but for application in other tasks, adaptations to the set of metrics to be used are necessary. The contributions present OACE as a replicable and robust tool for optimizing the evaluation of DL models, in addition to discussing insights into the performance of the different state-of-the-art architectures selected.

For future work, we propose (i) expanding the method with more evaluation metrics, (ii) conduct a comparative study with other MCDM approaches in order to ground the method in the state of the art, and (iii) incorporating multiple hyperparameter optimization to improve the solution robustness, and address the practical challenge discussed in the Introduction.

REFERENCES

[1] Sidra Aslam and Ali B Nassif. 2023. Deep learning based CIFAR-10 classification. In *Advances in Science and Engineering Technology International Conferences*. IEEE.
[2] Lu Bai, Weixing Ji, Qinyuan Li, Xilai Yao, Wei Xin, and Wanyi Zhu. 2022. Dnnabacus: Toward accurate computational cost prediction for deep neural networks. *arXiv preprint arXiv:2205.12095* (2022).
[3] Siddhant Bhadauria, Dharmendra Bisht, T Poongodi, and Suman Yadav. 2022. Assertive vision using deep learning and LSTM. In *2022 2nd International Conference on Innovative Practices in Technology and Management (ICIPTM)*. IEEE.
[4] Alfredo Canziani, Adam Paszke, and Eugenio Culurciello. 2016. An analysis of deep neural network models for practical applications. *arXiv preprint* (2016).
[5] Maarten V de Hoop, Daniel Z Huang, Elizabeth Qian, and Andrew M Stuart. 2022. The cost-accuracy trade-off in operator learning with neural networks. *arXiv preprint arXiv:2203.13181* (2022).
[6] Salmani et al. 2023. Reconciling High Accuracy, Cost-Efficiency, and Low Latency of Inference Serving Systems. In *Proceedings of the 3rd Workshop on Machine Learning and Systems* (Rome, Italy). 78–86.
[7] Christian Gianoglio, Edoardo Ragusa, Paolo Gastaldo, and Maurizio Valle. 2021. A novel learning strategy for the trade-off between accuracy and computational cost: a touch modalities classification case study. *IEEE Sensors Journal* 22 (2021).
[8] Vishu Gupta, Youjia Li, Alec Peltekian, Muhammed Nur Talha Kilic, Wei-keng Liao, Alok Choudhary, and Ankit Agrawal. 2024. Simultaneously improving accuracy and computational cost under parametric constraints in materials property prediction tasks. *Journal of Cheminformatics* 16, 1 (2024), 17.
[9] Anil Jadhav and Rajendra Sonar. 2009. Analytic Hierarchy Process (AHP), Weighted Scoring Method (WSM), and Hybrid Knowledge Based System (HKBS) for Software Selection: A Comparative Study. In *Second International Conference on Emerging Trends in Engineering and Technology, ICETET-09*. IEEE, 991–997.
[10] Alex Krizhevsky. 2009. *Learning Multiple Layers of Features from Tiny Images*. Technical Report. University of Toronto. https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf Technical Report.
[11] Hasmat Malik, Gopal Chaudhary, and Smriti Srivastava. 2022. Digital transformation through advances in artificial intelligence and machine learning.
[12] Sonia et al. Mijwil, Aggarwal. 2022. Has the Future Started? The Current Growth of ArtificialIntelligence, Machine Learning, and Deep Learning. *Iraqi Journal for Computer Science and Mathematics* 3, 1 (2022), 13.
[13] Gireen Naidu, Tranos Zuva, and Elias Mmbongeni Sibanda. 2023. A review of evaluation metrics in machine learning algorithms. In *Computer science on-line conference*. Springer, 15–25.
[14] Ebenezer Oluwasakin, Thomas Torku, S Tingting, Ahmeed Yinusa, S Hamdan, Samir Poudel, N Hasan, J Vargas, and K Poudel. 2023. Minimization of high computational cost in data preprocessing and modeling using MPI4Py. *Machine Learning with Applications* 13 (2023), 100483.
[15] KARL PEARSON. 1905. The Problem of the Random Walk. *Nature* 72, 1865 (07 1905), 294. https://doi.org/10.1038/072294b0
[16] Lyanh Pinto, André Alves, Adriano Dos Santos, Flávio Moura, Walter Oliveira, Jefferson Morais, Roberto de Oliveira, Diego Cardoso, and Marcos Seruffo. 2024. Optimized Assertiveness-Cost Evaluation: An Innovative Performance Measuring Method for Machine Learning Models. In *2024 IEEE LA-CCI*. IEEE, 1–6.
[17] Joao Schuler, Santiago Romani, Mohamed Abdel-Nasser, Hatem Rashwan, and Domenec Puig. 2022. Grouped pointwise convolutions reduce parameters in convolutional neural networks. In *Mendel*, Vol. 28. 23–31.
[18] Xin-She Yang. 2020. *Nature-inspired optimization algorithms*. Academic Press.