# Designing Convolutional Neural Network Architectures using a Genetic Algorithm

Muhammad Junaid Ali[1], Laurent Moalic[1], Mokhtar Essaid[1],
Muhammad Sulaman[1] and Lhassane Idoumghar[1]

Université de Haute-Alsace, IRIMAS UR 7499, F-68093 Mulhouse, France
{firstname.lastname@uha.fr}

**Keywords** : *Neural architecture Search, genetic algorithm, automatic machine learning, deep learning*

## 1 Introduction

Deep Neural Networks (DNNs) are powerful models that can solve different tasks as speech, images, and natural language understanding. In recent years, they have outperformed classical Machine Learning (ML) algorithms on numerous tasks. The problem with deep learning algorithms is designing them manually and tuning their parameters, which consumes time and requires human effort. To solve this issue, researchers have proposed Neural Architecture Search (NAS), which automatically designs a DNN for a specific task. NAS automates network architecture engineering. It aims to find a network topology to achieve the best performance on a specific task.

Typically, NAS consists of two stages: the searching stage, which aims to find a good-performing architecture, and the evaluation stage, in which the best-performing architecture is first trained using the training dataset and validated on the test set of the dataset. Corresponding to these two stages, NAS consists of three main components: search space $\mathcal{A}$, search strategy, and performance estimation strategy. Let A be a single network in the $\mathcal{A}$ search space, i.e., A $\in \mathcal{A}$. Let $\mathcal{D}_{train}$ and $\mathcal{D}_{val}$ be two subsets of dataset belonging to train and validation, respectively. The performance of architecture A trained on $\mathcal{D}_{train}$ and validated on $\mathcal{D}_{val}$ sets is measured by a function $\mathcal{F}$. NAS is mathematically formulated as follows:

$$\underset{A}{argmax} = \mathcal{F}(S^{val}(S^{train}(A))) \tag{1}$$

This study proposes a NAS approach based on evolutionary algorithm for searching Convolutional Neural Networks (CNNs) architectures. For solution representation, a continuous encoding scheme is also proposed.

## 2 Methodology

In this study, we proposed an approach for designing CNN architectures using Genetic Algorithm (GA). A number of studies have used GA to design CNN based architectures [2]. Combined with GA, a continuous encoding scheme is adopted to represent individuals, which maps each individual to corresponding layers with specific filter sizes. A total of 10 operations are used including convolution blocks, residual blocks, and pooling layers with different parameter settings. Numerous encoding schemes have been proposed in the literature to represent the CNN architecture.

These encoding schemes have a direct impact on the search space's complexity and architecture performance [1]. To evaluate the effectiveness of the proposed approach, we performed experiments on the CIFAR-10 dataset [3]. The implemented GA consists of four steps: (i)First step, the individuals are generated randomly using uniform distribution between the range of 0 and 1. (ii)Second step, the individuals are selected using a binary tournament approach. (iii)Third step, for reproduction one-point

crossover and bit-flipping mutation operators are used. (iv)Fourth step, for replacement strategy, the offspring replaces the worst individual in the population to update the population. The visualization of the encoding scheme adopted in this study is shown in FIG. 1.

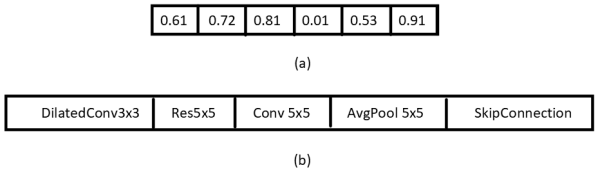


FIG. 1: Examples of genotype and phenotype representations of the proposed approach. The genotype representation is shown in (a) where each gene consists of probability value between 0 and 1 mapping towards some layers. In (b) the phenotype of each individual is shown in which each gene represents some convolution layers with kernel size, skip connection or some pooling layers.

# 3 Experimental Results

To evaluate the effectiveness of proposed approach, we perform experiments on CIFAR-10 dataset [3] with different parameter settings.

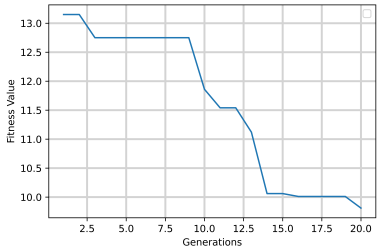


FIG. 2: Convergence of the best fitness values on 20 generations

| | Population Size 20<br>Training Epochs 20 | Population Size 30<br>Training Epochs 200 |
|---|---|---|
| Number of Generations = 10 | 79% | 86% |
| Number of Generations = 20 | 81% | 92% |

TAB. 1: Accuracy scores on different population sizes with the number of generations and training epochs.

The results on different parameters settings such as population size, number of training epochs, and number of generations are shown in TAB. 1.It is observed from the results that increasing the population size and the number of generations leads to the exploration of more diverse solutions, which increases the chances of finding an optimal solution. A good solution is found with a better accuracy score. We consider the loss obtained from model on testing data as the fitness. Furthermore, the convergence graph of fitness on 20 generations is shown in FIG. 2

# References

[1] Vargas-Hakim, Gustavo-Adolfo, Efren Mezura-Montes, and Hector-Gabriel Acosta-Mesa. "A review on convolutional neural network encodings for neuroevolution." IEEE Transactions on Evolutionary Computation 26.1 (2021): 12-27.

[2] Mirjalili, Seyedali. "Genetic algorithm." Evolutionary algorithms and neural networks. Springer, Cham, 2019. 43-55.

[3] Krizhevsky, Alex, and Geoffrey Hinton. "Learning multiple layers of features from tiny images." (2009): 7.