VISUAL COMPUTING & MACHINE INTELLIGENCE RESEARCH GROUP — INESCTEC — RECPAD 2023

Optimisation of Deep Neural Networks using a Genetic Algorithm: A Comparative Study

Tiago Gonçalves, Leonardo Capozzi, Jaime S. Cardoso, Ana Rebelo

Introduction

- Currently, most of the methods to design the architectures of deep learning models and to select the training hyper-parameters are still based on trial-and-error strategies.
- Practitioners recognise that there is a need for tools and frameworks that can achieve high-performing models almost automatically.
- To address this challenge, we implemented a genetic algorithm with a variable length Chromosome that allows us to generate an optimal network architecture.

Data

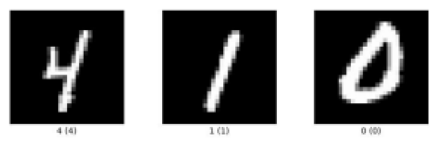


Figure 1: Images from the MNIST data set.

Figure 2: Images from the Fashion-MNIST data set.

Figure 3: Images from the CIFAR-10 data set.

Algorithm

Table 1: Names and possible values for the convolutional layers.

Name	Possible Values
Number of Convolutional Filters	$v \in \{8, 16, 32, 64, 128, 256, 512\}$
Size of the Convolutional Kernel	$v \in \{1, 3, 5, 7, 9\}$
Type of Activation Function	Identity, ReLU, Tanh
Range of Dropout Probability	$v \in [0, 1]$
Type of Pooling Function	Identity, Max, Avg

Table 2: Names and possible values for the fully-connected layers.

Name	Possible Values
Number of Output Neurons	$v \in [1, 100]$
Type of Activation Function	Identity, ReLU, Tanh
Range of Dropout Probability	$v \in [0, 1]$

Algorithm 1: Evolutionary Genetic Algorithm with Variable Chromosome Length.

```
Result: The fittest solution.
Initialise input shape;
Initialise number of labels;
Initialise $f_{max} = 5$;
Initialise $g_{max} = 100$;
Initialise $p_{max} = 40$;
Initialise $i = 2$;
Initialise $e = 5$;
Initialise $s = 4$;
Initialise $m = 0.2$;
Initialise $f = 0$;
while $f < f_{max}$ do
    $g = 0$;
    while $g < g_{max}$ do
        if $g = 0$ then Generate $p = p_{max}$ random solutions;
        else Select the $s$ best solutions automatically;
        Select the remaining $p = p_{max} - s$ solutions, according to a given probability;
        Perform the crossover operation;
        Perform the mutation operation at the rate $m$;
        for $p$ in range of $p_{max}$ do
        if $f > 0$ then Perform the transfer learning operation;
        Train the solution for $e$ epochs;
        Compute fitness of the all solutions (i.e., accuracy);
        $g += 1$
    end
    $f += 1$
end
```

Solution = [Convolutional Layers Block   Fully-Connected Layers Block   Learning Rate]

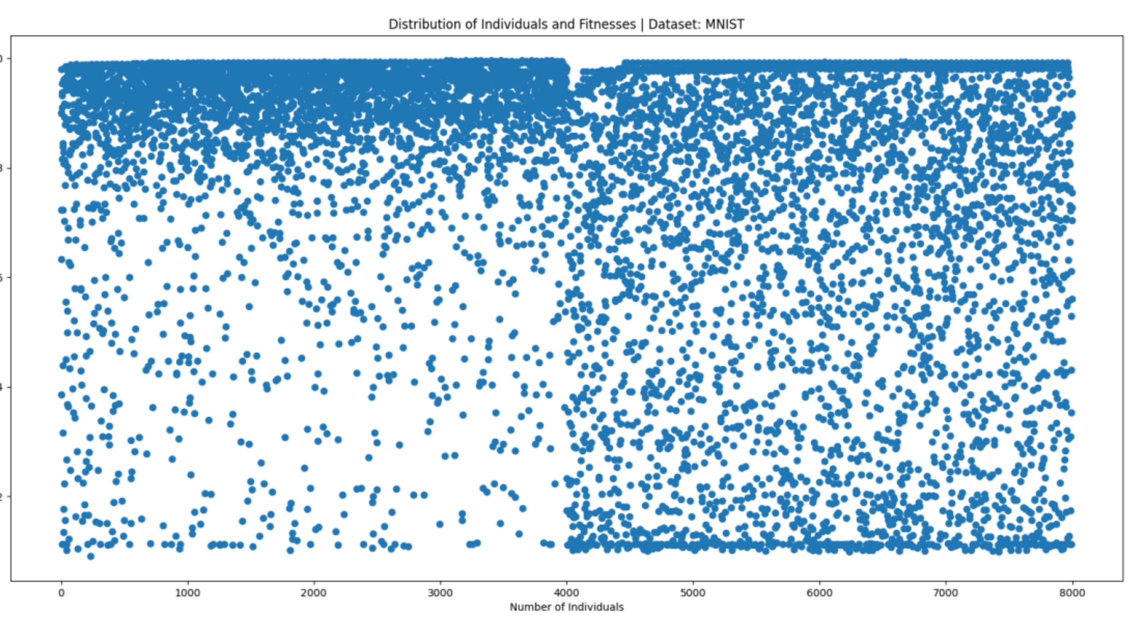
$$\mathcal{F}(s) = (1/loss(s)) * accuracy(s) \quad (1)$$

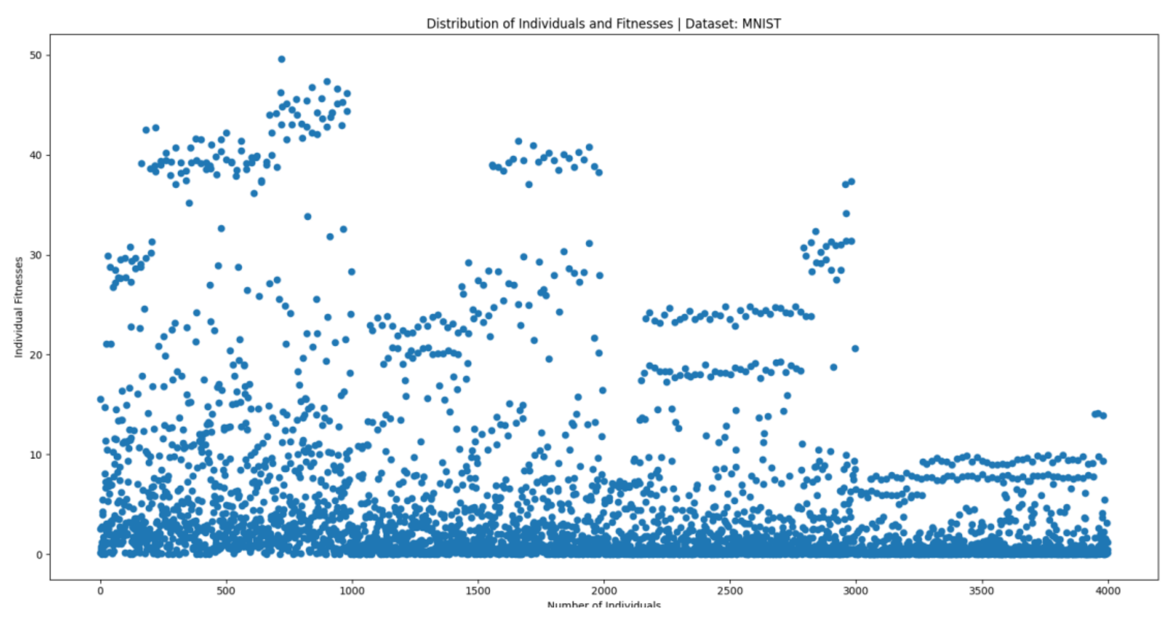
Maximising this function leads to solutions that have a large accuracy and a low loss value.
One of the downsides of this function is that small variations in the loss can lead to very large changes in the overall value of the fitness, which often resulted in the selection of individuals that did not have the best accuracy but had a low loss.

$$\mathcal{F}(s) = accuracy(s) \quad (2)$$

To overcome this issue we used this fitness function, which does not include the loss value and focuses only on the accuracy of the model.

Results





The results show that, in all data sets, that from one phase to the next one (i.e., when we increase by the size of the chromosome), the solutions seem to lose quality.

While in phase 0 most of the solutions seem to have fitnesses that are closer to the best, in phase 1, the solutions are almost uniformly distributed by all fitnesses.

Intuitively, increasing the size of the chromosome from one phase to another (i.e., increasing the complexity of the model) would mean that the model would over-fit easier.

However, we believe that this did not happen due to two possible causes:

- Our transfer learning procedure is not benefiting the model in terms of the initialisation of its weights.
- The number of training epochs (e) is not enough.

Data set	Accuracy (%)	Loss
MNIST	98.73	0.01878
Fashion-MNIST	90.81	0.11558
CIFAR-10	54.71	1.46837

Data set	Accuracy (%)	Loss
MNIST	98.87	0.01813
Fashion-MNIST	88.76	0.38088
CIFAR-10	58.81	0.52339

Acknowledgements

This work is co-financed by Component 5 - Capitalization and Business Innovation, integrated in the Resilience Dimension of the Recovery and Resilience Plan within the scope of the Recovery and Resilience Mechanism (MRR) of the European Union (EU), framed in the Next Generation EU, for the period 2021 - 2026, within project HfPT, with reference 11, and by FCT -- Fundação para a Ciência e a Tecnologia within the PhD grants "2020.06434.BD" and "2021.06945.BD".