A meta-learning approach to tune hyperparameters in Neural Networks
Micaela Verucchi\textsuperscript{1,3}, Giorgia Franchini\textsuperscript{3}, Matteo Spallanzani\textsuperscript{1}, Marko Bertogna\textsuperscript{3} and Luca Zanni\textsuperscript{3}

(*) Department of Physics, Informatics and Mathematics, Modena (Italy)

**PROBLEM STATEMENT**

- Hyperparameters are a fundamental component of most machine learning algorithms; these parameters strongly influence both the space of approximating functions and the way this space is explored during the training phase.
- Hyperparameters tuning is particularly critical for Deep Learning, as it affects both the convergence time of the training procedure and the network’s accuracy.
- Hyperparameters have usually been tuned through careful handmade procedures. Nowadays, optimized search algorithms which aim the goal of learning to learn are emerging: this research field deals with the so-called meta-learning approach.

**CNN EXPERIMENTAL TEST ARCHITECTURE**

![CNN Experimental Test Architecture Diagram]

**ALGORITHM 1**

```
Algorithm 1 AdjustLR

Input: \( \lambda_0, p_1, p_2, p_3 \)
Output: \( \lambda_{\text{out}} \in \mathbb{C} \)
Sample \( u \sim U[0,1] \)
if \( u < p_1 \) then
\( \lambda_{\text{out}} = \lambda_0 + \epsilon, \epsilon = 1 \)
else if \( u < p_1 + p_2 \) then
\( \lambda_{\text{out}} = \lambda_0 + \epsilon, \epsilon = 2 \)
else
\( \lambda_{\text{out}} = \lambda_0 - \epsilon, \epsilon = 3 \)
end if
```

**ALGORITHM 2**

```
Algorithm 2 TuneLearningRate

1. Set \( i = 1, \) \( \text{noprog} = 0, \text{noprog}_{\text{max}} \in \mathbb{N} \)
2. Set \( \lambda_0, \epsilon, p_1, p_2, p_3 \in \mathbb{R}^+ \)
3. \( \text{acc}_{\text{old}} = \text{Train}_\text{NN}(\lambda_0) \)
4. Set \( p_1 = p_2 = p_3 = 1/3 \)
5. \( \lambda_0, \epsilon, \text{adj} = \text{AdjustLR}(\lambda_0, p_1, p_2, p_3) \)
6. while \( \epsilon < \text{max} \) and \( \text{noprog} < \text{noprog}_{\text{max}} \) do
7. \( \text{acc}_{\text{new}} = \text{Train}_\text{NN}(\lambda_0) \)
8. if \( \text{acc}_{\text{new}} > \text{acc}_{\text{old}} \) then
9. \( \lambda_0 = \lambda_0 + \epsilon \)
10. \( \text{noprog} = \text{noprog} + 1 \)
11. else
12. \( \lambda_0 = \lambda_0 - \epsilon \)
13. \( \text{noprog} = \text{noprog} + 1 \)
14. end if
15. end while
```

**EXPERIMENTAL RESULTS**

![ExperimentalResultsDiagram]

**PERSPECTIVE WORK**

- Apply this methodology to other hyperparameters: mini-batch size, optimizer, regularization technique. This exploration will benefit of 25000 hours of HPC resources that we have been granted by the CINECA consortium [6].
- Apply our algorithm to train a CNN that should solve behavioural cloning (steering angle prediction) and be deployed on an autonomous F1/10th toy car [7].
- Apply our algorithm to train a CNN that should solve image segmentation on both a classification task (semantic segmentation) and a regression task (depth estimation), and should be deployed on crossroads cameras in the Modena Automotive Smart Area (MASA) [8].

**REFERENCES**

[7] f1tenth.org
[8] https://class-project.eu/