Project: Using evolutionary algorithms for the selection of optimal neural networks architectures.

General Description:

Generally, Neural Networks architecture/model selection (number of layers, number of neurons per layer, regularization terms, etc.) is a trial and error task which entails very time consuming computations. This project proposes to automate the model selection using evolutionary algorithms and a distributed computing framework (Ray [1]).

Evolutionary algorithms are a class of algorithms widely used to optimize a **fitness function.** In doing so, evolutionary algorithms emulate the process of evolution to make a set of solutions converge to a minimum of the fitness function. Usually, the **fitness** of a neural network given a specific problem (image classification, regression, etc.) is measured in terms of one or more **functions** such as *accuracy, R2 norm, precision, recall, F2 metric, etc.*

In this project, we propose the use of an evolutionary algorithm for the selection of an optimal neural network (only fully connected networks) architecture for a specific problem (hand written digit classification [2] and estimation of remaining useful life [3] are the two proposed demonstrators here). Nevertheless, evolutionary algorithms are known to be computationally expensive, requiring sometimes thousands of function evaluations to converge to a solution. This scenario is restrictive for a single computer setting, nevertheless by using a distributed computing framework such as Ray the workload can be balanced throughout several machines.

Milestones (11 weeks in total):

(1) (1 week) Install and run Ray, do some basic computations using the framework and the available resources (CPU, GPU, multiple machines, etc.)
(2) (1 week) Define suitable metrics for each of the problems at hand e.g., a composed metric (precision and recall) for classification.
(3) (1 week) Select an appropriate evolutionary algorithm (genetic algorithm, differential evolution) as a solver for the optimization problem of choosing the most fit neural network.
(4) (1 week) Define an appropriate encoding for the neural network model to be used by the evolutionary algorithm (array of bits, array of integers, array of real and integer values, etc.)
(5) (4 weeks) Implement the proposed framework in python and test it in a single computer
(6) (2 weeks) Implement the distributed version of the framework (using ray) and test it.
(7) (1 week) Evaluate the performance of the framework and synthesize the results.

References:
[1] P. Moritz et al. Ray: A distributed framework for emerging AI applications. Arxiv, 2018
[2] Q. Yu. The MNIST database of handwritten digits. http://yann.lecun.com/exdb/mnist/
[3] P. Lim. A time window neural networks based framework for remaining useful life estimation. International Joint Conference on Neural Networks, 2016