# **Neural network with torch lightning and optuna**

“*Optuna* is an automatic hyperparameter optimization software framework, particularly designed for machine learning. It features an imperative, *define-by-run* style user API… the user of Optuna can dynamically construct the search spaces for the hyperparameters.” [[1]](#footnote-0)

In this project, we are using Torch Lightning and Optuna to optimize a neural network. Torch Lightning handles the “model” which is the neural network, while Optuna handles tuning the hyperparameters. Optuna can be used to search for the optimal hyperparameters of a network and the results are used to fine-tune the network’s performance.

Optuna uses various different sampling algorithms to narrow the search space. Some of the algorithms are for example Grid search, random search, tree-structured Parzen Estimator, CMA-ES based algorithm, and quasi-Monte Carlo sampling algorithm.

The default sampler is the Tree-structured Parzen Estimator, which is also being used in this project.

Tree-structured Parzen estimator is a type of bayesian estimator, where we fit a surrogate model to prior observations to estimate the hyperparameter space. The Key idea is instead of P(y|x), we model P(x|y), where x is a single hyperparameter and y is the loss. In TPE we maintain two surrogate models, one for the best cases (minimizing loss) and one for the worst cases (maximizing the loss). With each new observation, the model is fitted to the best and worst-case scenarios, giving us the most probable hyperparameters that result in the best and worst-case scenarios. [[2]](#footnote-1)

Optuna gives the user control of the surrogate model (or surrogate) by setting a hierarchy of parameters and the tree structure. This allows the user to control the model's fidelity, which is important when searching for optimal hyperparameters.

Besides Sampling algorithms, Optuna also provides pruning algorithms for the early stopping of inefficient search trials. Pruning algorithms are useful when the search space is large, as they help to focus on only the efficient search trials while discarding the inefficient ones early on.

Since the dataset chosen for this project is rather on a smaller scale of things, we defined a rather small search space so no pruning was needed.

The steps in this project are as follows:

* Train the model with default hyperparameters
* Set up hyperparameter space
* Initiate the Optuna

Once Optuna has finished the trial runs and made a suggestion of the optimal hyperparameters, a final model with the best parameters is trained, validated, and finally tested.

Optuna and Torch Lightning can be combined to produce an efficient neural network model which can provide accurate and reliable results for machine learning tasks. It is best used when the complexity of the problem and the size of the dataset are large.

1. https://optuna.readthedocs.io/en [↑](#footnote-ref-0)
2. https://www.jair.org/index.php/jair/article/view/13188 [↑](#footnote-ref-1)