Assignment 5
Advanced Machine Learning
University of Milano-Bicocca
M.Sc. Data Science

Alessandro Riboni
December 9, 2019
ID number: 847160
a.riboni2@campus.unimib.it

Hyperparameter Optimization of a neural network

The task of this assignment is Hyperparameter Optimization (HPO) of a neural network, with the aim to maximize its Accuracy on 10 fold cross-validation.
In this report are presented two different steps:

- In the first one, it was carried out HPO for two hyperparameters using 5 initial random configurations as initial design. After choosing a surrogate model, it was performed 20 iterations of Sequential Model Based Optimization (SMBO) using two different acquisition functions. The results are compared against 25 configurations in Grid Search and 25 configurations sampled via Pure Random Search.

- In the second step, instead, four hyperparameters were optimized, starting from 10 initial random configurations as initial design. Also, in this situation, a surrogate model has been chosen, but it was performed 100 iterations of Sequential Model Based Optimization (SMBO) using two different acquisition functions.

In both cases, it was used the *SMAC3* library for HPO and *scikit-learn* to implement the neural network in Python. It is necessary to consider that the results obtained and presented in this report are not always replicable due to the probabilistic nature of Bayesian optimization.

Dataset

The dataset used is named *Fertility* and it is available on the *OpenML* website: *Fertility-dataset*. It refers to a binary classification diagnosis and, consists of around 100 instances and 9 features (excluding the class). The class attribute can take on the values 1 (*normal*) or 2 (*altered*).
The data has already been pre-processed and all features are numeric. As can be seen from the following figure, there is a class imbalance problem. Nevertheless, the accuracy has been used as a metric, since the purpose is to evaluate the optimization of hyperparameters and not particularly interested in the performance of the model on the rare class.

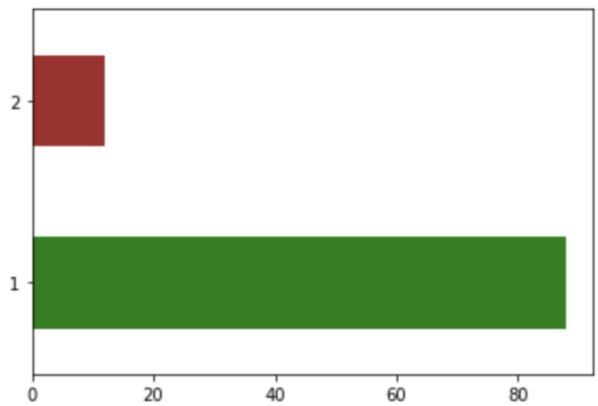

Figure 1: Distribution of Class attribute

Step 1: HPO for two neural network's hyperparameters

In this first step, it was implemented a neural network with two hidden layers with four units in the first hidden layer and two neurons in the second one. It was performed HPO on two parameters:

- **learning_rate_init**: the function of SMAC3 *UniformFloatHyperparameters* has been used, which can take on values in range 0.01 - 0.1;

- **momentum**: the function of SMAC3 *UniformFloatHyperparameters* has been used, which can take on values in range 0.1 - 0.9.

For the other hyperparameters of the *MLPclassifier*, the default values were used (i.e. activation = "relu", solver = "adam", etc.).
To apply HPO, it was built a Configuration Space, which represents the domain of hyperparameters, and it was saved in a Scenario object. The parameter *runcount-limit* of this object has been set to 25 to indicate that the process must be iterated 20 times after the 5 initial random configurations.
The Gaussian Process (GP) has been chosen as a probabilistic surrogate model. It is a collection of random variables, any finite number of which have a joint Gaussian distribution. In addition, it is a generalization of the Gaussian probability distribution: it is a distribution over functions rather than a distribution over vectors. This model was preferred to the Random Forest because in the case studies that involve two continuous variables and are not present condition variables, GP-based SMBO is more suitable.
The initial design consists of five initial random configurations of the hyperparamenters, which are the same between the two SMBO experiments related to the different acquisition functions.
In the first experiment have been carried out 20 iterations of SMBO using Lower Confidence Bound (**LCB**) as acquisition function. In general, this is the most optimistic estimation on predictions.
In the second one have been performed 20 iterations using Expected Improvement (**EI**) as acquisition function. In general, this function deals better with the trade-off between exploitation and exploration and offers significant convergence results.
In the following figure is presented the cumulative accuracy considering the maximum values obtained at each iteration of SMBO with the two different acquisition function.

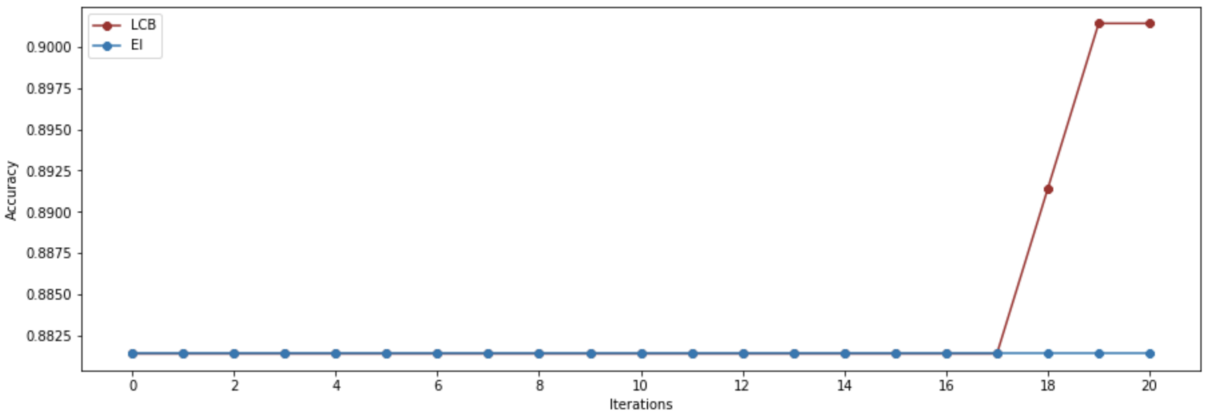

Figure 2: 20 iterations of SMBO using LCB and EI as acquisition functions

In the first case, using LCB as an acquisition function, it can be seen that after 17 iterations the maximum accuracy increases reaching 0.9014. Instead, using EI, the maximum accuracy

remains constant for all iterations, i.e. no improvement has been achieved with respect to the result obtained with the initial configurations.
The results obtained were compared with 25 configuration in Grid Search and 25 configurations sampled via Pure Random Search and the best configuration for each method is reported in the following table:

Best confiiguration	Learining_rate_init	Momentum	Accuracy
SMBO with GP and LCB	0.033002	0.751144	0.9014
SMBO with GP and EI	0.037797	0.730016	0.8814
Grid Search	0.010000	0.100000	0.8814
Pure Random Search	0.043467	0.115229	0.8814

Figure 3: Comparison of the best configurations

In this situation, the maximum accuracy is achieved by running an SMBO and using LCB as the acquisition function.

Step 2 - HPO for four neural network's hyperparameters

In this second step, it was implemented a neural network with two hidden layers and it was performed HPO on four parameters:

- **learning_rate_init**: the function of SMAC3 *UniformFloatHyperparameters* has been used, which can take on values in range 0.01 - 0.1;
- **momentum**: the function of SMAC3 *UniformFloatHyperparameters* has been used, which can take on values in range 0.1 - 0.9;
- **# of units in the first hidden layer**: the function of SMAC3 *UniformIntegerHyperparameters* has been used, which can take on integer values in range 1 - 5;
- **# of units in the second hidden layer**: the function of SMAC3 *UniformIntegerHyperparameters* has been used, which can take on integer values in range 1 - 5.

For the other hyperparameters of the *MLPclassifier*, the default values were used.
Also in this case, it was built a Configuration Space which represents the domain of hyperparameters and it was saved in a Scenario object. The parameter *runcount-limit* of this object has been set to 110 to indicate that the process must be iterated 100 times after the 10 initial random configurations.
Contrary to the previous case, a Random Forest has been chosen as probabilistic surrogate model. The RF algorithm combines Bagging and random selection of features to construct a collection of decision trees with controlled variance.
The initial design consists of ten initial random configurations of the hyperparamenters, which are the same between the two SMBO experiments related to the different acquisition functions.
In the first experiment have been carried out 100 iterations of SMBO using Lower Confidence Bound (**LCB**) as acquisition function. Commonly, this is the most optimistic estimation on predictions.
In the second one have been performed 100 iterations using Expected Improvement (**EI**) as acquisition function. Commonly, this function deals better with the trade off between exploitation and exploration and offers significant convergence results.
In the following figure is presented the cumulative accuracy considering the maximum values obtained at each iteration of SMBO with the two different acquisition function.

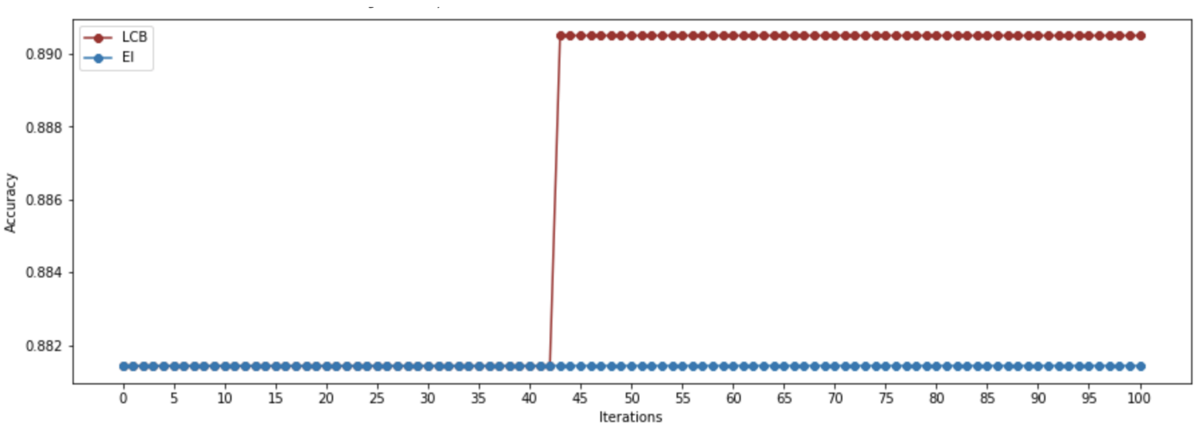

Figure 4: 100 iterations of SMBO using LCB and EI as acquisition functions

In the first case, using LCB as an acquisition function, it can be seen that after more than 40 iterations the maximum accuracy increases from 0.8814 to 0.8905 and tends to remain constant until the end of process. Instead, using EI, there is no improvement in the maximum accuracy, which remains constant at 0.8814 for all iterations.
The following table shows the configurations obtained at the end of the two algorithms.

Best confiiguration	Learining_rate_init	Momentum	# units HL_1	# units HL_2	Accuracy
SMBO with RF and LCB	0.030107	0.208901	4	2	0.8905
SMBO with RF and EI	0.026349	0.788666	3	5	0.8814

Figure 5: Comparison of the best configurations

Again, the LCB acquisition function allows finding a better configuration in order to maximize the accuracy of the binary classification of the neural network. However, the result achieved is worse than that obtained in the first step, in which the Gaussian Process was used as a surrogate model.