Bayesian optimization

Université Claude Bernard Lyon 1

7 December 2022

Machine Learning and new challenges

Better models

- Model complexity with high number of parameters
- High performances not easily accessible
  Expert knowledge, cost of computing power
- Less reproducible
  Details missing, different computers

Hyper-parameter tuning

Deep Learning

- High interest in DL
- High investment in DL
- When well tuned, very successful for visual object identification, speech recognition and many other tasks
- Many hyper-parameters: number of layers, activation function, layer size, etc.

Hyper-parameter tuning

- Hyper-parameters that maximize or minimize an objective
- Define an objective

  e.g. Minimize 5-Fold average MSE for a regression
- You have to fit your model each time you want to know the resulting "score"

  You need pairs (hyper-parameters, score)

$\Rightarrow$ This is an optimization problem

Optimization problem

Definition : black box optimization

Find the global optimum of an unknown function $f(\mathbf{x})$

There is no analytical form of the function you want to optimize !

Optimization problem

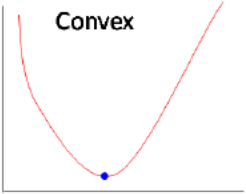
Definition : black box optimization

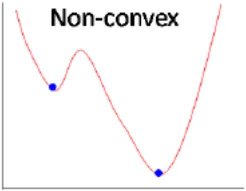
Find the global optimum of an unknown function f(**x**)

There is no analytical form of the function you want to optimize !

Summarized problem

- Unknown function
- Global optimization
- Multi-dimensional inputs
  i.e. number of hyper-parameters can vary
- Non-convex (i.e. local optima)
  i.e. be aware of traps





Hyperparameter tuning

Input Hyperparameters
e.g. learning rate, number of layers, etc.

Output Metrics
e.g. mse, rse, mae, etc.

Some properties ?

Expensiveness

Time to fit a model ?

Hours, days, months executed on costly computers. Especially for deep learning...
e.g. "Grandmaster level in StarCraft II using multi-agent reinforcement learning" - Nature 2019 - Vynials *et al.* [1]

Expensiveness

Time to fit a model ?

Hours, days, months executed on costly computers. Especially for deep learning...
e.g. "Grandmaster level in StarCraft II using multi-agent reinforcement learning" - Nature 2019 - Vynials *et al.* [1]

Optimization property

We want to minimize drastically the number of evaluations
evaluations $\Leftrightarrow$ experiment$\Leftrightarrow$ input-output pair

Problem properties: noise

Repeatability

- Randomization is involded in the evaluation process.
- Most of the models are sensitive to measurement noise.

Problem properties: search space

- $\mathbf{x} = \{x_0, x_1, ..., x_n\}$ where n in the number of input dimensions.

Input parameters are most of the time bounded, sometimes constrained.

- Each x_i is bounded in $U_i = \{lb, ub\}$
- most of the time, bounds are rescaled to { 0, 1 }

$\Rightarrow$ The global optimum (x_{best}, y_{best}) is located in the domain $(0, 1)^n$

Problem definition

- Minimize an unknown noisy function
- Lowest number of evaluations
- Bounded and more generally constrained variables

Bibliography: black box optimization

This list is not exhaustive !

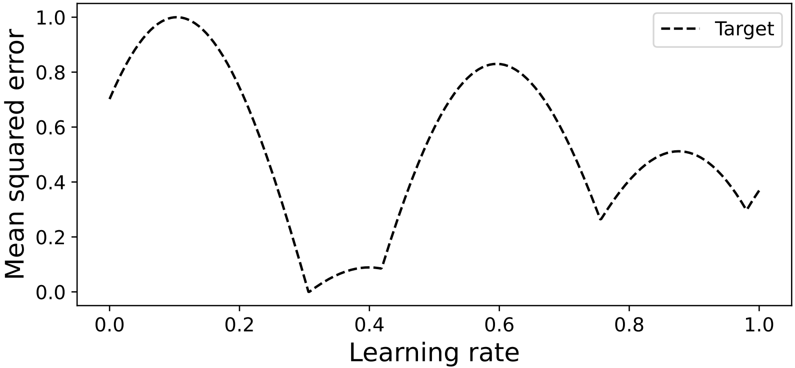
- Grid search
- Design of Experiments
- Simplex based method - 1965 [2]
- Gradient descent - [3]
- Branch and fit - 2008 [4]
- **Bayesian optimization** - 2012 [5]

Bayesian optimization algorithm

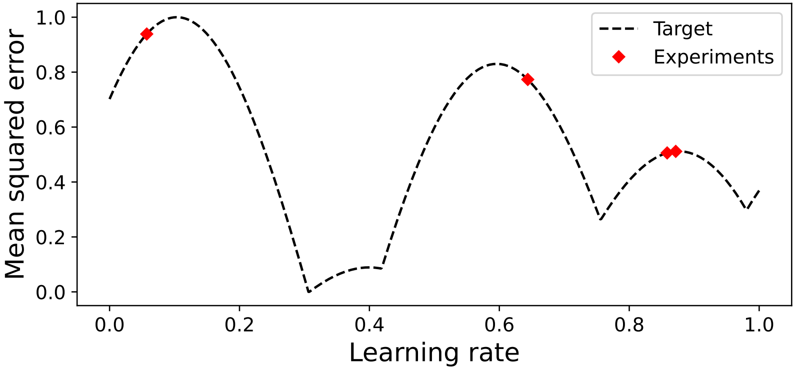
Algorithm Standard bayesian algorithm

X, Y = initialize with n number of random evaluations
for budget **do**
    model = fit(X, Y)
    $\mathrm{x}_{sugg} = argmax(f_{acq}(model))$
    $y_{sugg} = f(x_{sugg})$
    X, Y = [X, x_{sugg}], [Y, y_{sugg}]
end for

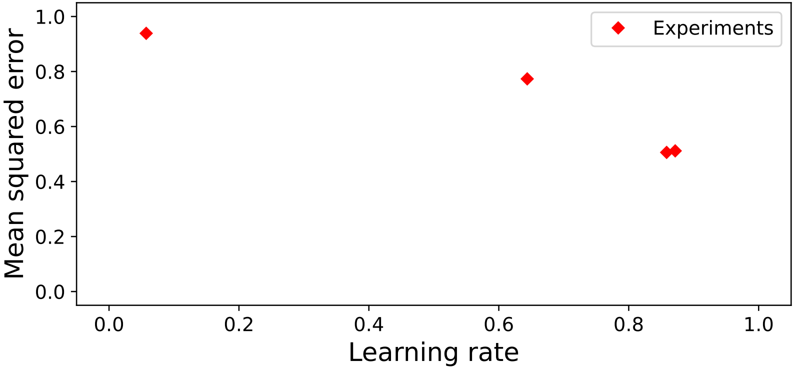
Example: bayesian optimization



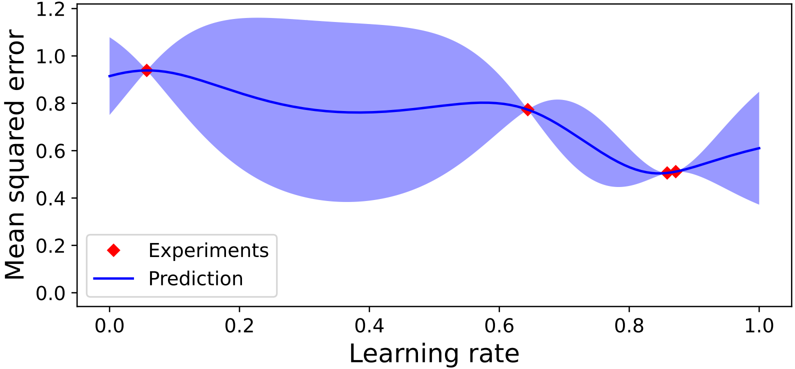
Example: bayesian optimization



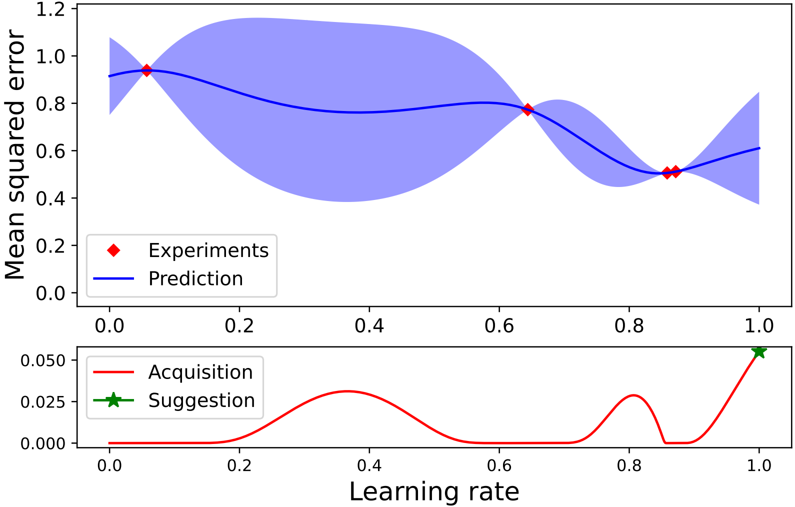
Example: bayesian optimization



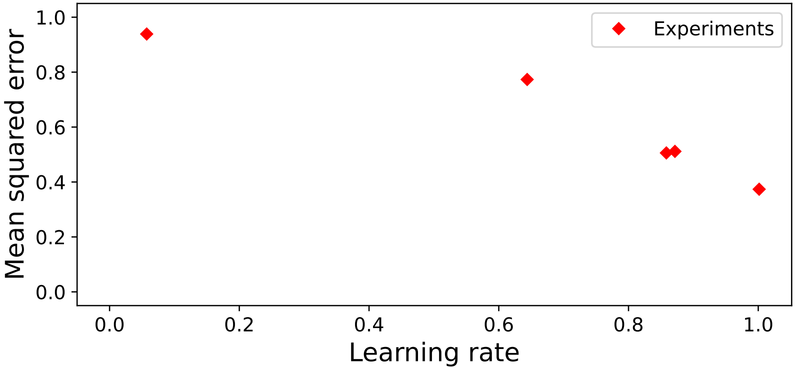
Example: bayesian optimization



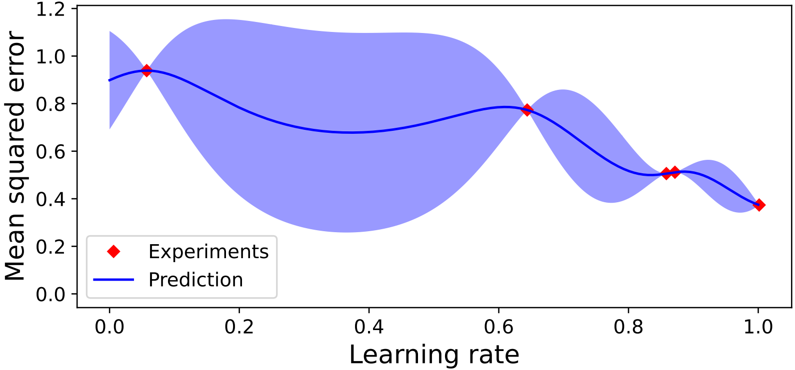
Example: bayesian optimization



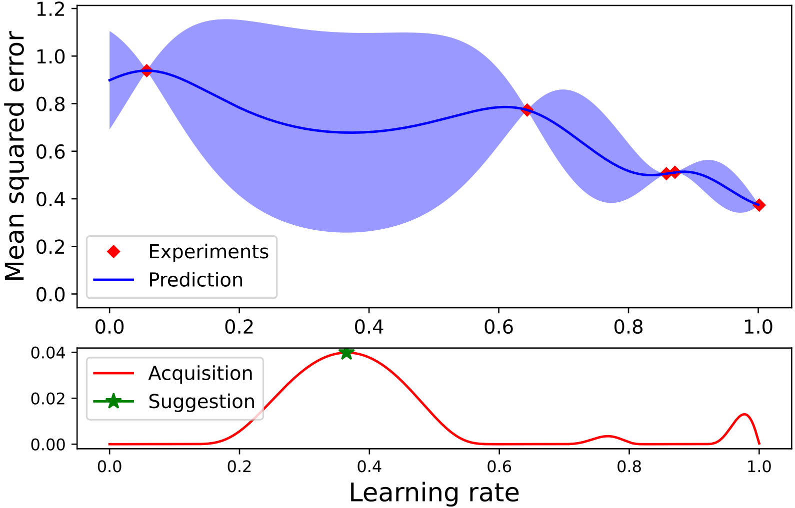
Example: bayesian optimization



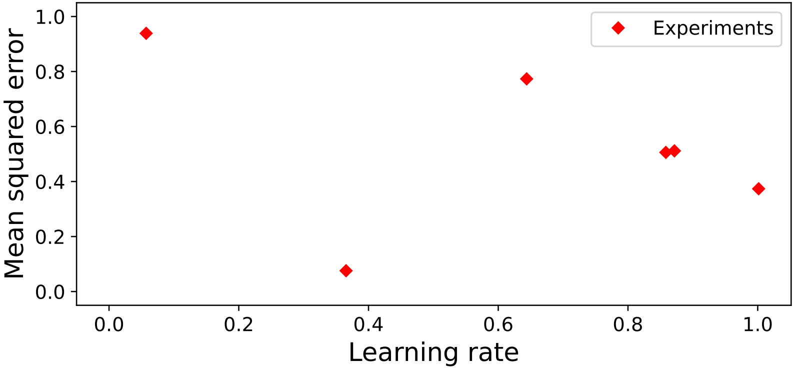
Example: bayesian optimization



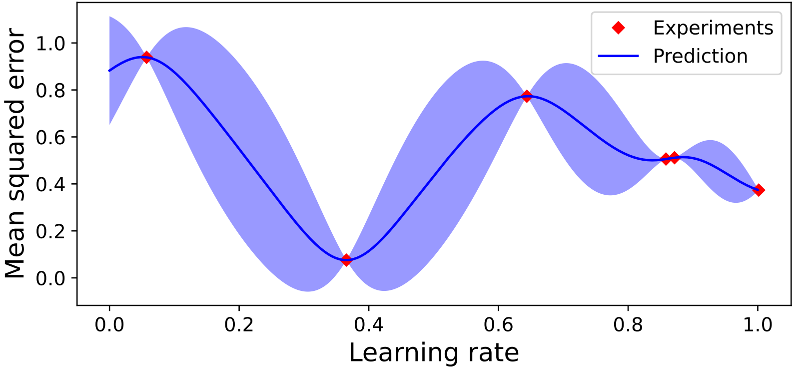
Example: bayesian optimization



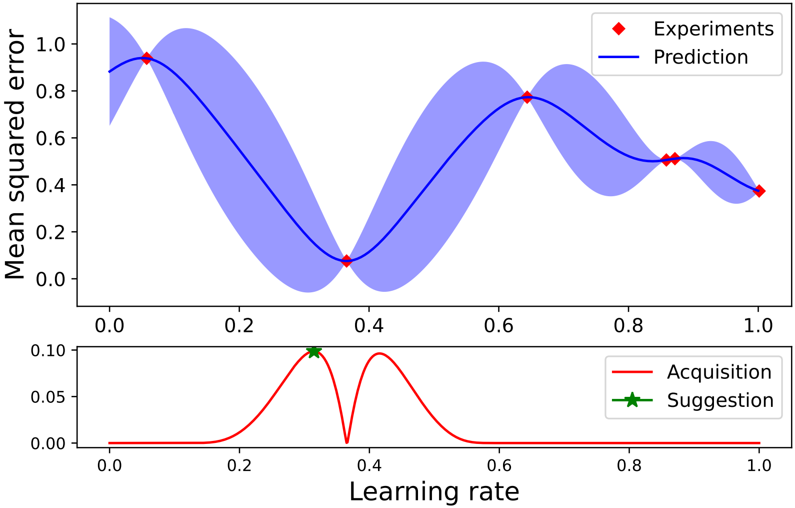
Example: bayesian optimization



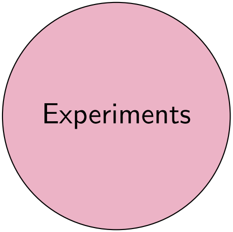
Example: bayesian optimization



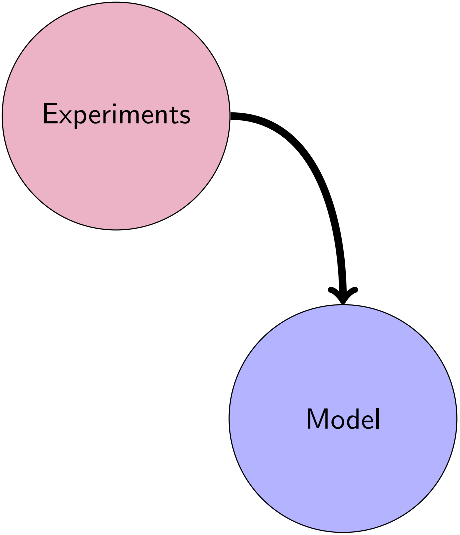
Example: bayesian optimization



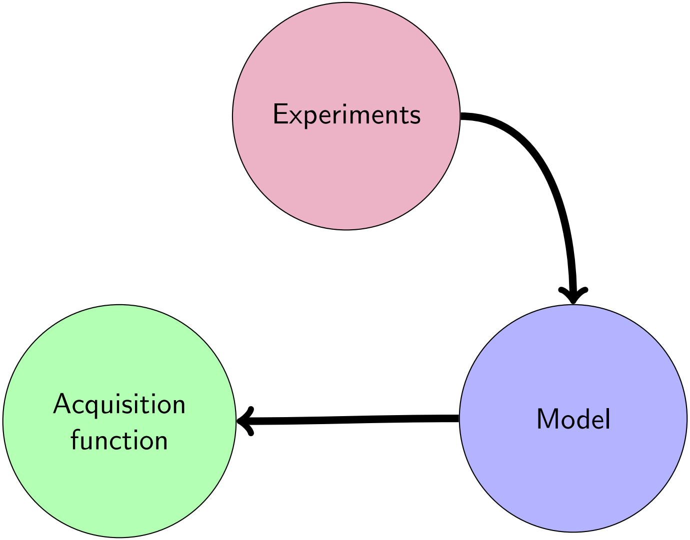
Schema bayesian optimization



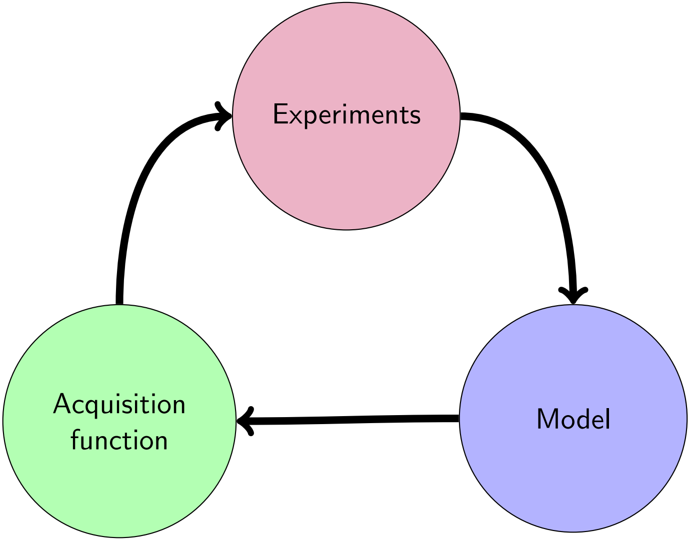
Schema bayesian optimization



Schema bayesian optimization



Schema bayesian optimization



Bayesian optimization: model

- Gaussian processes [5]
- Random forest [6]
- Gradient boosting [7]
- Kernel regression [8]
- ...

Bayesian optimization: model

- Gaussian processes
- Random forest
- GBM
- kernel regression
- ...

Gaussian processes

Kernel based model

- GPs are mainly defined by their covariance function (i.e. kernel).

Gaussian processes

Kernel based model

- GPs are mainly defined by their covariance function (i.e. kernel).
- Standard covariance function for smooth regression:

$$Matérn_{5/2}(X, X') = \sigma^2(1 + \frac{\sqrt{5}d}{\rho} + \frac{5d^2}{3\rho^2})exp(-\frac{\sqrt{5}d}{\rho})$$

Gaussian processes

Kernel based model

- GPs are mainly defined by their covariance function (i.e. kernel).
- Standard covariance function for smooth regression:

$$Matérn_{5/2}(X, X') = \sigma^2(1 + \frac{\sqrt{5}d}{\rho} + \frac{5d^2}{3\rho^2})exp(-\frac{\sqrt{5}d}{\rho})$$

- Multivariate normal distributions

Gaussian processes

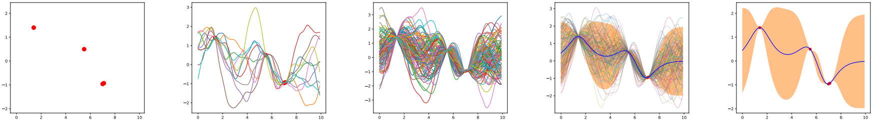
Kernel based model

- GPs are mainly defined by their covariance function (i.e. kernel).
- Standard covariance function for smooth regression:

$$Mat\acute{e}rn_{5/2}(X, X') = \sigma^2(1 + \frac{\sqrt{5}d}{\rho} + \frac{5d^2}{3\rho^2})exp(-\frac{\sqrt{5}d}{\rho})$$

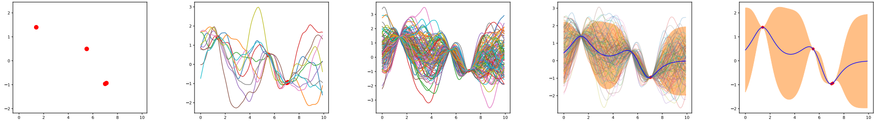
- Multivariate normal distributions

Function distribution



Mean and variance as predictions

Function distribution



Predictions

For any points you have :

- The mean of the distribution
- The variance of the distribution which can be interpreted as the uncertainty of the model about the prediction (mean)

Gaussian processes: fitting

Optimization as fitting

Find optimal covariance hyper-parameters

- Maximize the (log) marginal likelihood
- e.g. :

$$Matérn_{5/2}(X, X') = \sigma^2(1 + \frac{\sqrt{5}d}{\rho} + \frac{5d^2}{3\rho^2})exp(-\frac{\sqrt{5}d}{\rho})$$

Going further ?

Gaussian processes for Machine Learning (GPML) - TODO

Acquisition function ?

From this point you have :

- Some experiments (i.e. pairs of inputs X, outputs Y)
- A fitted model (GP fitted on X and Y)
- What to do next ?

Acquisition function: definition

Definition

The acquisition function takes the predictions as inputs and results in a "score" representing the **exploration** and **exploitation** potential of the inputs.

Example: Upper Confidence Bound

$$UCB(\mathbf{X}) = \mu(\mathbf{X}) + \alpha \times \sigma(\mathbf{X})$$

w.r.t :

- $\mathbf{X}$ is a set of parameters that haven't been tested yet.
- $\mu(\mathbf{X})$ is the predicted output of $\mathbf{X}$.
- α is a parameter controlling the exploration during the optimization.
- $\sigma(\mathbf{X})$ is the uncertainty of the model on the prediction.

Acquisition function optimization

You want to suggest the input parameters with the highest potential possible.
$\Rightarrow$ Maximize the acquisition function.

Acquisition function optimization

Sub-problem definition

$$\text{Optimize } acq(\mathbf{X})$$

w.r.t.:

- No need to spare on the number of evaluations.
- Gradient usually exists.
- Same search space properties as the global optimization properties.
- Highly dependant on the model.

Acquisition function optimization

How to optimize ?

- Gradient descent is a good first option
- scipy provides a descent interface for optimization (scipy.optimize)
- You can go further by developping your own optimizer (e.g. evolutionary algorithm)

Problem variants

- **Type of variables** : "learning rate" is easy, it's a continuous variable. What to do with "number of layers" or "type of activation function" (discrete, non-numerical, dimensional, ...)
- **Parallelization** : if the model take 2 days to train, I want to be able to do multiple experiments at the time
- **Hiddden constraints** : If you can't use some combinations but you don't know it *a priori*
- etc.

What to do now ?

Practical work: first steps

- Optimize a toy function using your bayesian optimization algorithm
- Evaluate your results

Practical work: real problem

- Create a model
- Create an objective (e.g. metrics + cross validation)
- Optimize this model using your bayesian optimization algorithm
- Observe the results and see what can be improved !

References I

- O. Vinyals, I. Babuschkin, W. M. Czarnecki, M. Mathieu, A. Dudzik, J. Chung, D. H. Choi, R. Powell, T. Ewalds, P. Georgiev, *et al.*, "Grandmaster level in starcraft ii using multi-agent reinforcement learning," *Nature*, vol. 575, no. 7782, pp. 350–354, 2019.
- J. A. Nelder and R. Mead, "A simplex method for function minimization," *The computer journal*, vol. 7, no. 4, pp. 308–313, 1965.
- S. Ruder, "An overview of gradient descent optimization algorithms," *arXiv preprint arXiv:1609.04747*, 2016.
- W. Huyer and A. Neumaier, "Snobfit–stable noisy optimization by branch and fit," *ACM Transactions on Mathematical Software (TOMS)*, vol. 35, no. 2, pp. 1–25, 2008.

References II

J. Snoek, H. Larochelle, and R. P. Adams, "Practical bayesian optimization of machine learning algorithms," *Advances in neural information processing systems*, vol. 25, 2012.

F. Hutter, H. H. Hoos, and K. Leyton-Brown, "Sequential model-based optimization for general algorithm configuration," in *International conference on learning and intelligent optimization*, pp. 507–523, Springer, 2011.

J. van Hoof and J. Vanschoren, "Hyperboost: Hyperparameter optimization by gradient boosting surrogate models," *arXiv preprint arXiv:2101.02289*, 2021.

References III

F. Häse, M. Aldeghi, R. J. Hickman, L. M. Roch, and A. Aspuru-Guzik, "Gryffin: An algorithm for bayesian optimization of categorical variables informed by expert knowledge," *Applied Physics Reviews*, vol. 8, no. 3, p. 031406, 2021.