DE LA RECHERCHE À L'INDUSTRIE

Explainable Hyperparameter Optimization using goal oriented sensitivity analysis

28/04/2021

Paul Novello[†,*], Gaël Poëtte[†], David Lugato[†], Pietro Congedo[*] |
[†]CEA-CESTA,[*]Inria Saclay-CMAP Ecole Polytechnique

Context : PDE-based numerical simulations of multi-scale and multi-physics phenomena using Deep Learning.

Common point of all these codes : trade-off between

- Cost efficiency for numerical studies (production, sensitivity analysis, uncertainty quantification...).
- Accuracy of the numerical prediction.

Approach : Acceleration of PDE-based simulation codes by approximating costly parts with a neural network.

$\Rightarrow$ Requires to work on the same trade-off during the construction of a neural network ...

... Let's construct a neural network !

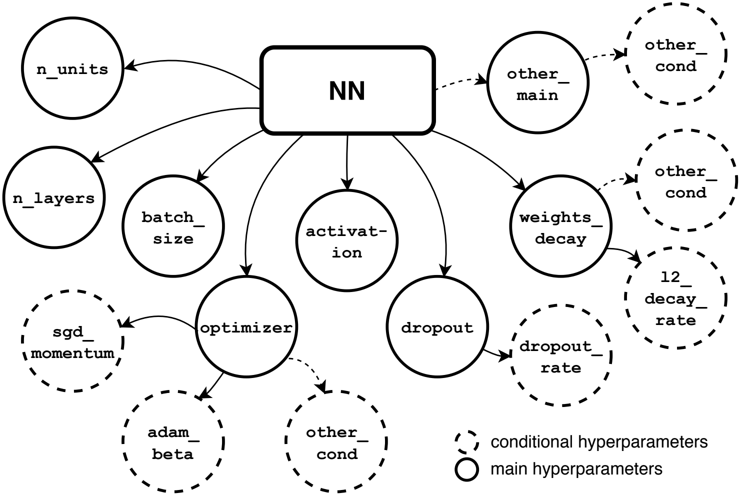
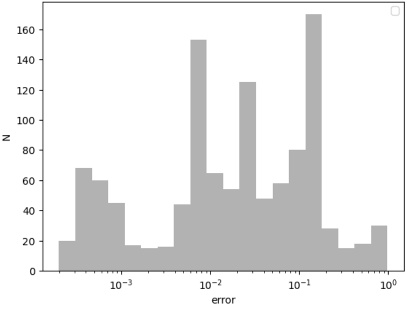
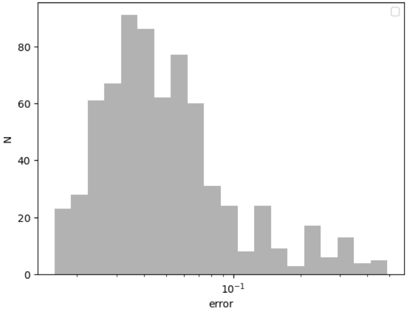


Figure 1 – Example of hyperparameters space. So many possibilities ...



(a) Bateman equations, L_2 error



(b) MNIST, $1 - \mathit{accuracy}$ (in %)

Histogram of the error of a NN for different instances of training corresponding to random hyperparameters.

Three attention points :

- There is a lot of possible hyperparameters combinations and we have to use Monte Carlo.
- Hyperparameters have a high influence on the error,
- but they also have an impact on execution or training time, especially width and depth.
  $\Rightarrow$ Cost-efficiency / accuracy trade-off !

Many HO techniques ... (random search, bayesian optimization, hyperband, etc) [1, 15, 16, 2, 17, 9, 11, 5, 19, 13, 20, 10] ... that all have at least one drawback : **lack of explainability** (black-box)

Motivations to work on HO :

- Correcting the lack of explainability of usual HO techniques
- Large potential impact on cost efficiency
- Large potential impact on accuracy

Sensitivity analysis (SA) evaluates the effect of input variables $(X_1, ..., X_{n_h}) \in \mathcal{X}_1 \times ... \times \mathcal{X}_n$ on an output $f(X_1, ..., X_{n_h}) \in \mathcal{Y}$ (exhaustive review in [14]).

Motivations to use SA :

- Analyzing the relative importance of input variables for explaining the output
  - $\Rightarrow$ Explainability
- Selecting practically convenient values for input variables with a limited negative impact on the output
  - $\Rightarrow$ Cost efficiency
- Identifying where to efficiently put research efforts in order to improve the output
  - $\Rightarrow$ Accuracy

These benefits will allow us to set up an explainable hyperparameter optimization algorithm.

Constraints :

- Since we are only interested in the most accurate neural networks, the analysis must be goal-oriented.
- One training of a neural network can be a long process so we would need scalable indices estimation.

Main possibilities :

- Goal-oriented variance analysis based indices [6, 3] : estimation error of $O(\frac{1}{\sqrt{n_s}})$ requires $(n_h + 2) \times n_s$ sample evaluations.
- Dependance measures [12, 4] : estimation error of $O(\frac{1}{\sqrt{n_s}})$ requires n_s sample evaluations. $\Rightarrow$ Best choice for us.

In this work, we focus on Hilbert Schmidt Independence Criterion (HSIC)

Let $X \in \mathcal{X}$ and $Y \in \mathcal{Y}$, and $\mathcal{G}$ a Restricted Kernel Hilbert Space (RKHS) of kernel $k : \mathcal{X}^2 \times \mathcal{Y}^2 \to \mathbb{R}$. HSIC [8] can be written

$$HSIC(X, Y) = \gamma_k^2(\mathbb{P}_{XY}, \mathbb{P}_Y \mathbb{P}_X) \tag{1}$$

Where $\gamma_k^2(\mathbb{P}_{XY}, \mathbb{P}_X \mathbb{P}_Y)$ is the Maximum Mean Discrepancy (MMD) between $\mathbb{P}_{XY}$ and $\mathbb{P}_X \mathbb{P}_Y$. HSIC is the distance between $\mathbb{P}_{XY}$ and $\mathbb{P}_Y \mathbb{P}_X$ embedded in $\mathcal{H}$.

$\Rightarrow$ Since $X \perp Y \Rightarrow \mathbb{P}_{XY} = \mathbb{P}_Y \mathbb{P}_X$, the closer these distributions are, in the sense of γ_k, the more independent they are.

The definition of HSIC involves a kernel choice for γ_k, which we will discuss shortly.

HSIC can be applied in the context of Hyperparameters Optimization (HO) by considering that :

- the set $\{X_1, ..., X_{n_h}\}$ is the set of hyperparameters.
- The output to consider is $Z = \mathbb{1}_{f(X_1,...,X_{n_h}) \in \mathrm{Y}}$, where Y is chosen to be the sub space of $\mathcal{Y}$ for which $f(X_1, ..., X_{n_h})$ is in the best percentile p of the error say $p = 10\%$ (as in [18]).
  $\Rightarrow$ $HSIC(X_i, Z)$ boils down to the distance between X_i and $X_i | f(X_1, ..., X_{n_h}) \in \mathrm{Y}$.

HSIC measures the importance of each hyperparameter in reaching the top 10% best NNs.

We denote HSIC(X,Y) Monte Carlo estimation by $S_{X,\mathrm{Y}}$.

Practical problems to circumvent :

P1 Hyperparameters do not live in the same measured space : Continuous (`weights_decay` $\in [10^{-6}, 10^{-1}]$), integers (`n_layers` $\in \{8, ..., 64\}$), categorical (`activation` $\in \{\texttt{relu}, ..., \texttt{sigmoid}\}$) ...

P2 They could interact with each others. For instance `batch_size` adds variance on the objective function optimized by `optimizer`.

P3 Some hyperparameters are not involved for every neural networks configurations : `dropout_rate` is not used when `dropout = False`.

Example Let $f : [0,2]^2 \rightarrow \{0,1\}$ such that

$$f(X_1, X_2) = \begin{cases} 1 & \text{if } X_1 \in [0,1], X_2 \in [0,1], \\ 0 & \text{otherwise.} \end{cases}$$

Let $X_1 \sim \mathcal{N}(1, 0.1, [0,2])$ (normal distribution of mean 1 and variance 0.1 truncated between 0 and 2) and $X_2 \sim \mathcal{U}[0,2]$

	X_1	X_2
$S_{X,\mathbf{Y}}$	3.7×10^{-3}	4.8×10^{-3}

Table 1 – $S_{X,\mathbf{Y}}$ values for X_1 and X_2

This conclusion is not desirable : in our case, the choice of the hyperparameter distribution is only related to practical concerns.

Let Φ_i be the CDF of X_i. We have that $\Phi_i(X_i) = U_i$, with $U_i \sim \mathcal{U}[0,1]$. We can first sample from U_i, record the sample and then train the network with hyperparameter $\Phi_i^{-1}(U_i)$.

- Variables U_i are iid and live in the same measured space, so can be compared pairwise with HSIC regardless of practical distribution choice for X_i.
- It strongly facilitates the kernel choice : we will use Gaussian Radial Basis Functions.

An important point :

- For continuous variables, $\Phi_i(X_i)$ is a bijection between $\mathcal{X}_i$ and $[0,1]$ so Φ_i^{-1} is well defined.
- For categorical, integer or boolean variables (discrete variables), it is not the case. We use a method, as in [7] to sample discrete variables from uniform variables.

	U_1	U_2
$S_{X,Y}$	4.8×10^{-3}	4.8×10^{-3}

Table 2 – $S_{X,Y}$ values for U_1 and U_2

⇒ Solution :

Transform hyperparameters values into samples of uniform random variables and compare U_i with $U_i|f(X_1,...,X_{n_h}) \in Y$ instead of X_i with $X_i|f(X_1,...,X_{n_h}) \in Y$.

Example For instance let $f : [0,2]^3 \to \{0,1\}$ such that

$$f(X_1, X_2, X_3) = \begin{cases} 1 & \text{if } X_1 \in [0,1], X_2 \in [1,2], X_3 \in [0,1], \\ 1 & \text{if } X_1 \in [0,1], X_2 \in [0,1], X_3 \in [1,2], \\ 0 & \text{otherwise.} \end{cases}$$

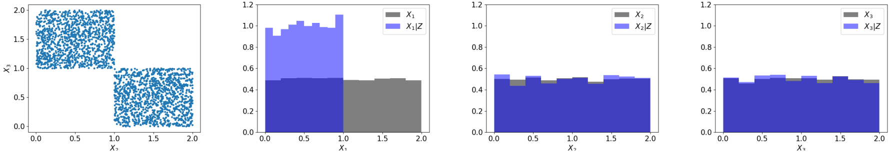


Figure 3 – **From left to right : 1** - Pairs of $(X_2|Z = 1, X_3|Z = 1)$. **2** - Histogram of X_1 and $X_1|Z = 1$. **3** - Histogram of X_2 and $X_2|Z = 1$. **4** - Histogram of X_3 and $X_3|Z = 1$.

$\Rightarrow S_{X_2,Y}$ and $S_{X_3,Y}$ will be very low (even 0 analytically)

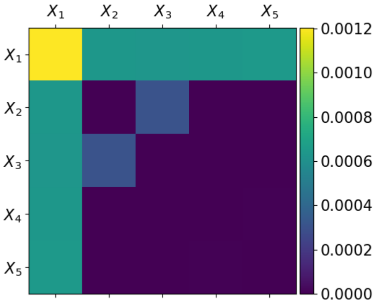


Figure 4 – $S_{(X_i,X_j),\mathbf{Y}}$ for each pair of variable (X_i, X_j).

	$S_{X,\mathbf{Y}}$
X_1	1.17×10^{-3}
X_2	8.11×10^{-7}
X_3	1.68×10^{-6}
(X_2, X_3)	3.04×10^{-4}
(X_4, X_5)	1.37×10^{-6}

Table 3 – $S_{X,\mathbf{Y}}$ values for variables of the experiment

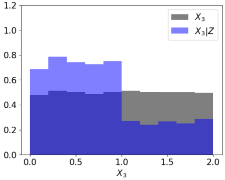
⇒ Solution :

Check for interactions using $S_{(X_i,X_j),\mathbf{Y}}$ before selecting hyperparameter values.

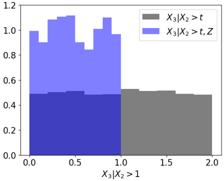
Example. Let $f : [0,2]^3 \to \{0,1\}$ such that :

$$f(X_1, X_2, X_3) = \begin{cases} B & \text{if } X_1 \in [0,1], X_2 \in [0,t] \\ 1 & \text{if } X_1 \in [0,1], X_2 \in [t,2], X_3 \in [0,1], \\ 0 & \text{otherwise,} \end{cases}$$

With B a Bernoulli variable of parameter 0.5 and $t \in [0,2]$ (so that $S_{X_2,Y}$ is low)



(a)



(b)

Let $\mathcal{J}_k \in \{1, ..., n_h\}$ be the set of indices of hyperparameters that can be involved in a training jointly with conditional hyperparameter X_k. We define $\mathcal{G}_{X_k} = \{X_i | X_k, i \in \mathcal{J}_k\}$, the set of hyperparameters involved jointly in hyperparameter configurations when X_k is also involved.

$\Rightarrow$ Each $S_{X_i,Y}$ must be only compared within $X_i \in \mathcal{G}_{X_k}$.

$\Rightarrow$ Solution :

Only compare hyperparameters' HSIC within a same conditional group.

Practical problems to circumvent :

P1 Hyperparameters do not live in the same measured space : Continuous ($\texttt{weights\_decay} \in [10^{-6}, 10^{-1}]$), integers ($\texttt{n\_layers} \in \{8, ..., 64\}$), categorical ($\texttt{activation} \in \{\texttt{relu}, ..., \texttt{sigmoid}\}$) ...

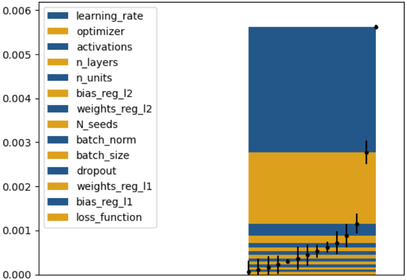
> Transform hyperparameters values into samples of uniform random variables and compare U_i with $U_i | f(X_1, ..., X_{n_h}) \in \mathrm{Y}$ instead of X_i with $X_i | f(X_1, ..., X_{n_h}) \in \mathrm{Y}$.

P2 They could interact with each others. For instance `batch_size` adds variance on the objective function optimized by `optimizer`.

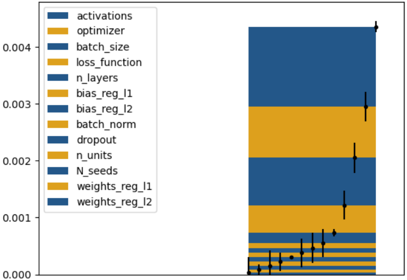
> Check for interactions using $S_{(X_i, X_j), \mathrm{Y}}$ before selecting hyperparameter values.

P3 Some hyperparameters are not involved for every neural networks configurations : `dropout_rate` is not used when `dropout = False`.

> Only compare hyperparameters' HSIC within a same conditional group.



(a) Bateman equations

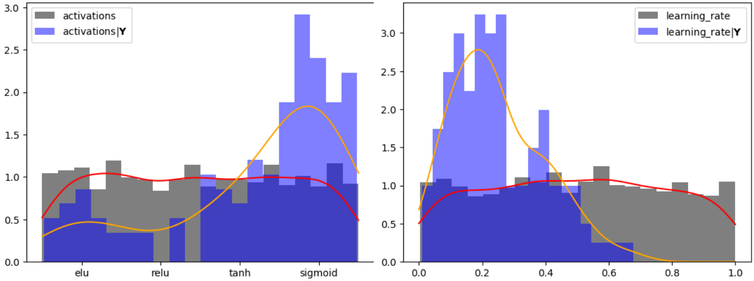


(b) MNIST

Comparison of $S_{X_i,Y}$, where Y is the error 10% percentile.

Cost-efficiency / Accuracy trade-off remark : For MNIST, `n_layers` has a very limited impact on accuracy, but the strongest impact on execution time !

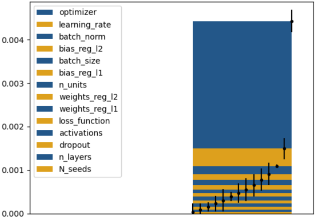
Application 1 : explainability



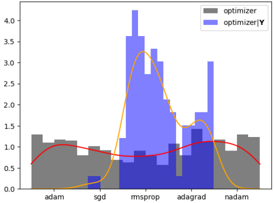
(a) MNIST

(b) Bateman

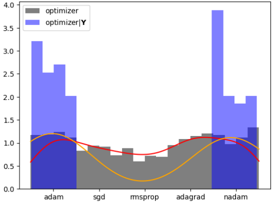
Figure 7 – Representation of $U_i|Z = 1$ (orange for KDE and blue for histogram) and U_i (red for KDE and grey for histogram), for hyperparameters X_i with high $S_{X_i,\mathbf{Y}}$



(a) $S_{X_i,\mathbf{Y}}$, Y = 10% worst errors

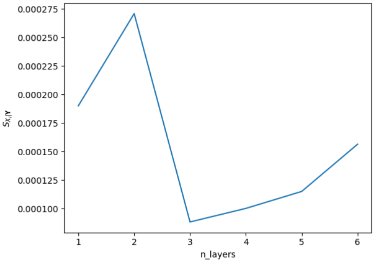


(b) $X_i|$Y, Y = 10% worst errors

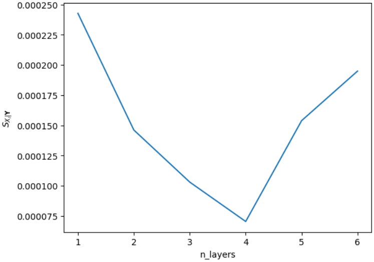


(c) $X_i|$Y, Y = 10% best errors

Figure 8 – Bateman : **(a)** Comparison of $S_{X_i,\mathbf{Y}}$ when Y is the set of the 10% worst errors. **(b)** Histogram of $X_i|$Y when Y is the set of 10% worst errors, with $X_i = \texttt{optimizer}$. **(c)** Histogram of $X_i|$Y when Y is the set of the 10% best errors, with $X_i = \texttt{optimizer}$.



(a) MNIST



(b) Bateman

Figure 9 – $S_{X_i|X_i \in [a+c,b],\mathbf{Y}}$ w.r.t. c for (a) `n_layers` in MNIST, (b) `n_layers` in Bateman.

Let $X_i \in [a, b]$ be a hyperparameter that have an impact on both accuracy and execution time. One can find a good trade off by computing $S_{X_i,\mathbf{Y}}$ for $X_i \in [a + c, b]$.

Results when pairing HSIC with Gaussian Processes Based Bayesian Optimization (GPBO) :

data set	baseline	test metric	params	MFLOPs	FLOPS factor
MNIST	RS	98.36	436,147	871	$\times 3$
-	GPBO	**98.42** $\pm$ 0.05	10,271,367	20,534	$\times 67$
-	HSIC-GPBO	**98.42** $\pm$ 0.02	**151,306**	**307**	$\times 1$ (ref)
Bateman	RS	**1.99** $\times 10^{-4}$	1,259,140	2,516	$\times 360$
-	GPBO	2.94 $\pm$ 0.42 $\times 10^{-4}$	1,588,215	3,173	$\times 453$
-	HSIC-GPBO	3.49 $\pm$ 0.31 $\times 10^{-4}$	**3,291**	**7**	$\times 1$ (ref)

Table 4 – Results of hyperparameter optimization for Random Search (RS), Gaussian Processes based Bayesian Optimization on full hyperparameters space (GPBO) and HSIC Gaussian Processes based Bayesian Optimization (HSIC-GPBO). HSIC-GPBO first optimizes most impactful hyperparameters defined by HSIC (as in [18]) with other hyperparameters values chosen in order to improve cost efficiency. It then fine-tunes the remaining hyperparameters that has no effect on execution time.

- The neural networks obtained have competitive test error w.r.t. GPBO (which is one of the most used HO algorithm).
- HSIC analysis allows dramatically reducing the number of FLOPs and params of the resulting neural networks.
- It is complementary to other HO techniques. Could even be used jointly with multi objective HO techniques to further reduce execution time.

Neural networks' hyperparameters optimization is a tedious and challenging task with high stakes because :

- Great impact on accuracy.
- Great impact on performances.
- Lack of explainability

⇒ HSIC based Sensitivity analysis approach to hyperparameter optimization.

In this work :

- We adapted HSIC based sensitivity analysis methodology to hyperparameter analysis
- We used it to explain hyperparameters' effect on the test error.
- We constructed a robust and explainable HO methodology that addresses the accuracy-performances trade-off.

Thank you for your attention

[1] James Bergstra et Yoshua Bengio. “Random Search for Hyper-Parameter Optimization”. In : *Journal of Machine Learning Research* 13.10 (2012), p. 281-305. url : http://jmlr.org/papers/v13/bergstra12a.html.

[2] James S. Bergstra et al. “Algorithms for Hyper-Parameter Optimization”. In : *Advances in Neural Information Processing Systems 24*. Sous la dir. de J. Shawe-Taylor et al. Curran Associates, Inc., 2011, p. 2546-2554. url : http://papers.nips.cc/paper/4443-algorithms-for-hyper-parameter-optimization.pdf.

[3] E. Borgonovo. “A new uncertainty importance measure”. In : *Reliability Engineering & System Safety* 92.6 (2007), p. 771 -784. issn : 0951-8320. doi : https://doi.org/10.1016/j.ress.2006.04.015. url : http://www.sciencedirect.com/science/article/pii/S0951832006000883.

[4] I. Csizar. “Information-type measures of difference of probability distributions and indirect observation”. In : *Studia Scientiarum Mathematicarum Hungarica* 2 (1967), p. 229-318. url : https://ci.nii.ac.jp/naid/10028997448/en/.

[5] Thomas Elsken, Jan Hendrik Metzen et Frank Hutter. “Neural Architecture Search : A Survey”. In : *Journal of Machine Learning Research* 20.55 (2019), p. 1-21. url : http://jmlr.org/papers/v20/18-598.html.

[6] Jean-Claude Fort, Thierry Klein et Nabil Rachdi. "New sensitivity analysis subordinated to a contrast". In : *Communications in Statistics - Theory and Methods* 45.15 (2016), p. 4349-4364. doi : 10.1080/03610926.2014.901369. eprint : https://doi.org/10.1080/03610926.2014.901369. url : https://doi.org/10.1080/03610926.2014.901369.

[7] Daniel T Gillespie. "A general method for numerically simulating the stochastic time evolution of coupled chemical reactions". In : *Journal of Computational Physics* 22.4 (1976), p. 403 -434. issn : 0021-9991. doi : https://doi.org/10.1016/0021-9991(76)90041-3. url : http://www.sciencedirect.com/science/article/pii/0021999176900413.

[8] Arthur Gretton et al. "Measuring Statistical Dependence with Hilbert-Schmidt Norms". In : *Proceedings of the 16th International Conference on Algorithmic Learning Theory*. ALT'05. Singapore : Springer-Verlag, 2005, 63–77. isbn : 354029242X. doi : 10.1007/11564089_7. url : https://doi.org/10.1007/11564089_7.

[9] Kevin Jamieson et Ameet Talwalkar. "Non-stochastic Best Arm Identification and Hyperparameter Optimization". In : *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*. Sous la dir. d'Arthur Gretton et Christian C. Robert. T. 51. Proceedings of Machine Learning Research. Cadiz, Spain : PMLR, 2016, p. 240-248. url : http://proceedings.mlr.press/v51/jamieson16.html.

[10] Kirthevasan Kandasamy et al. “Neural Architecture Search with Bayesian Optimisation and Optimal Transport”. In : *Proceedings of the 32nd International Conference on Neural Information Processing Systems*. NIPS'18. Montréal, Canada : Curran Associates Inc., 2018, 2020–2029.

[11] Lisha Li et al. “Hyperband : A Novel Bandit-Based Approach to Hyperparameter Optimization”. In : *Journal of Machine Learning Research* 18.185 (2018), p. 1-52. url : http://jmlr.org/papers/v18/16-558.html.

[12] Alfred Müller. “Integral Probability Metrics and Their Generating Classes of Functions”. In : *Advances in Applied Probability* 29.2 (1997), 429–443. doi : 10.2307/1428011.

[13] Hieu Pham et al. “Efficient Neural Architecture Search via Parameters Sharing”. In : sous la dir. de Jennifer Dy et Andreas Krause. T. 80. Proceedings of Machine Learning Research. Stockholmsmässan, Stockholm Sweden : PMLR, 2018, p. 4095-4104. url : http://proceedings.mlr.press/v80/pham18a.html.

[14] Saman Razavi et al. “The Future of Sensitivity Analysis : An essential discipline for systems modeling and policy support”. In : *Environmental Modelling & Software* 137 (2021), p. 104954. issn : 1364-8152. doi : https://doi.org/10.1016/j.envsoft.2020.104954. url : http://www.sciencedirect.com/science/article/pii/S1364815220310112.

[15] Bobak Shahriari et al. “Taking the Human Out of the Loop : A Review of Bayesian Optimization”. In : *Proceedings of the IEEE* 104 (2016), p. 148-175.

[16] Jasper Snoek, Hugo Larochelle et Ryan P. Adams. "Practical Bayesian Optimization of Machine Learning Algorithms". In : *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 2*. NIPS'12. Lake Tahoe, Nevada : Curran Associates Inc., 2012, 2951–2959.

[17] Jasper Snoek et al. "Scalable Bayesian Optimization Using Deep Neural Networks". In : *Proceedings of the 32nd International Conference on Machine Learning*. Sous la dir. de Francis Bach et David Blei. T. 37. Proceedings of Machine Learning Research. Lille, France : PMLR, 2015, p. 2171-2180. url : `http://proceedings.mlr.press/v37/snoek15.html`.

[18] Adrien Spagnol, Rodolphe Le Riche et Sébastien Da Veiga. "Global sensitivity analysis for optimization with variable selection". In : *SIAM/ASA J. Uncertain. Quantification* 7 (2018), p. 417-443.

[19] Kenneth O. Stanley et Risto Miikkulainen. "Evolving Neural Networks through Augmenting Topologies". In : *Evol. Comput.* 10.2 (juin 2002), 99–127. issn : 1063-6560. doi : `10.1162/106365602320169811`. url : `https://doi.org/10.1162/106365602320169811`.

[20] Mingxing Tan et al. "MnasNet : Platform-Aware Neural Architecture Search for Mobile". In : *CoRR* abs/1807.11626 (2018). arXiv : `1807.11626`. url : `http://arxiv.org/abs/1807.11626`.