# Stochastic bi-level optimization for hyper-parameter selection

**Supervision:**

- Thomas Moreau, Parietal, Inria, Palaiseau (thomas.moreau@inria.fr)
- Pierre Ablin, DMA, ENS ULM, Paris (pierre.ablin@ens.fr)

**Context** Most machine learning methods have hyper-parameters that are critical for their the performance. In most practical cases, the dataset is split in two sets, the training and validation set. The parameters of the method are computed by minimizing a loss function on training set, and the hyper-parameters are then set by minimizing the loss function on the validation set. This is a bi-level optimization problem. The most popular methods such as Grid-Search or Random Search [3, 6] sample the hyper-parameter space, and select the hyper-parameters that yield the best loss. These methods are widely used because they can be easily adapted to any model, and are simple to implement. However, as the number of hyper-parameters grow, they quickly become intractable. In order to efficiently search the hyper-parameter space, algorithms based on bayesian optimization have also been proposed [1]. Other approaches propose to select hyper-parameters are based on statistical control of the generalization error. Using the Stein Unbiased Risk Estimate (SURE, [8]), Deledalle et al. [4] proposed a gradient based method which allows to use all the data to train and tune the hyper-parameters. Recently, Lounici et al. [5] proposed another method to also directly control the generalization error when selecting the hyperparameters.

Finally, a promising approach consists in using gradient descent on the validation loss [2, 7]. This method scales better with the number of hyper-parameters than grid-search [**?** ], and can be implemented either by automatic differentiation or by leveraging the implicit function theorem.

These methods all require to use the whole training set before doing an update on the hyper-parameters: in this sense, they are *full-batch* methods. We propose to study *stochastic* methods for this task, where one would make progress on the hyper-parameters by using only a few samples from the training data. Stochastic algorithms are notoriously faster than full-batch methods for large datasets, but are also generally harder to analyse. In addition to being fast, the proposed algorithm should come with some statistical guarantees.

**Methods** We denote the validation and training distributions with $val$ and $train$. The hyper-parameters are $\lambda$, and the parameters are $\theta$. We have a risk function $f(x, \theta, \lambda)$, which indicates how much the data sample $x$ is explained by the hyper-parameters $\lambda$ and the parameters $\theta$. When the hyper-parameters $\lambda$ are fixed, the parameters $\theta$ are obtained by empirical risk minimization (ERM):

$$\theta^*(\lambda) \in \arg\min_\theta \mathbb{E}_{x\sim train}[f(x, \theta, \lambda)] \tag{1}$$

The hyperparameters are such that $\mathbb{E}_{x\sim val}[f(x, \theta, \theta^*(\lambda))]$ are minimized, which gives the bi-level optimization problem

$$\min_\lambda \mathbb{E}_{x\sim val}[f(x, \theta, \theta^*(\lambda))] \quad s.t. \quad \theta^*(\lambda) \in \arg\min_\theta \mathbb{E}_{x\sim train}[f(x, \theta, \lambda)] \tag{2}$$

We propose to introduce a stochastic algorithm to minimize this problem. Then we will study the convergence properties of these algorithm, and in particular its impact on the generalization of the learned estimator. We propose to first study this for simple models such as the logistic and ridge regression. The proposed approach will be benchmarked against classical approaches based on validation score and SURE methods.

**Environment** The internship will take place in Inria Saclay, in the Parietal team. This is a large and inclusive team focused on mathematical methods for statistical modeling of brain function using neuroimaging data (fMRI, MEG, EEG). Particular topics of interest of the team include machine learning techniques, numerical and parallel optimization, applications to human cognitive neuroscience, and scientific software development.

**Requirements**

- Strong mathematical background. Knowledge in numerical optimization is appreciated.
- Some knowledge of Python, and willingness to learn how to develop good scientific code. Knowledge of a deep-learning library like Pytorch or Tensorflow is a plus.

**References**


[1] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A Next-generation Hyperparameter Optimization Framework. In *ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2623–2631, Anchorage, AK, USA, July 2019.

[2] Yoshua Bengio. Gradient-Based Optimization of Hyperparameters. *Neural Computation*, 12(8):1889–1900, August 2000.

[3] James Bergstra and Yoshua Bengio. Random Search for Hyper-Parameter Optimization. *Journal of Machine Learning Research (JMLR)*, 13(1):281–305, 2012.

[4] Charles Alban Deledalle, Samuel Vaiter, Jalal Fadili, and Gabriel Peyré. Stein Unbiased GrAdient estimator of the Risk (SUGAR) for multiple parameter selection. *SIAM Journal on Imaging Sciences*, 7(4):2448–2487, 2014.

[5] Karim Lounici, Katia Meziani, and Benjamin Riu. Optimizing generalization on the train set: A novel gradient-based framework to train parameters and hyperparameters simultaneously. *preprint ArXiv*, 2006.06705, June 2020.

[6] F Pedregosa, G Varoquaux, A Gramfort, V Michel, B Thirion, O Grisel, Mathieu Blondel, P Prettenhofer, R Weiss, V Dubourg, J Vanderplas, A Passos, D Cournapeau, M Brucher, M Perrot, and E Duchesnay. Scikit-learn: Machine Learning in {P}ython. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[7] Fabian Pedregosa. Hyperparameter optimization with approximate gradient. In *International Conference on Machine Learning (ICML)*, pages 737–746, New-York, NY, USA, August 2016.

[8] Charles M. Stein. Estimation of the Mean of a Multivariate Normal Distribution. *The Annals of Statistics*, 9(6): 1135–1151, November 1981.