Optimization des Hyperparamètres appliquée au Fine Tuning de LLM

Basé sur l'article : *Bayesian and Partition-Based Optimization for Hyperparameter Optimization of LLM Fine-Tuning*

Nathan Davouse

Sommaire

1. Introduction

2. Design et Implémentation

3. Résultats et Analyses

4. Conclusion

Large Language Models

Point clés

- Etat de l'art pour le traitement de language naturel.
- Réseaux de Neurones avec une architecture basé sur le transformer[1] (annexe 3)
- Taille : entre 1 et 405 Milliards de neurones

[1] Vaswani et al, Attention is all you need,2017

Auto-attention

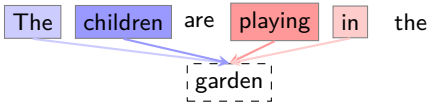


Figure: Illustration du mécanisme d'auto-attention

L'auto-attention est la clé du LLM, en permettant de comprendre le contexte

Fine Tuning

Aspect	Pre-entrainement	Fine Tuning
Objectif	Apprentissage general	Adaptation à un domaine
Données	Larges et diverses	Restreintes et Spécifiques
Ressources	Centaines de GPU	au moins 1 GPU
Durée	Semaine/Mois	Heures/Jours

Table: Comparaison entre le Pre-entrainement et le Fine Tuning de LLM

Parameter-Efficient Fine-Tuning (PEFT)

- Ensemble de méthodes pour réduire le nombre de paramètres à entrainer
- Utilisation de la méthode LoRA (annexe 5)
- Amène des nouveaux hyperparamètres

Optimisation des Hyperparamètres (OHP)

Hyperparamètres

Paramètres qui ne sont pas entrainés par le modèle (learning rate, dropout ...)

Objectifs

- Meilleur performance qu'en manuel
- Retirer le besoin d'expertise

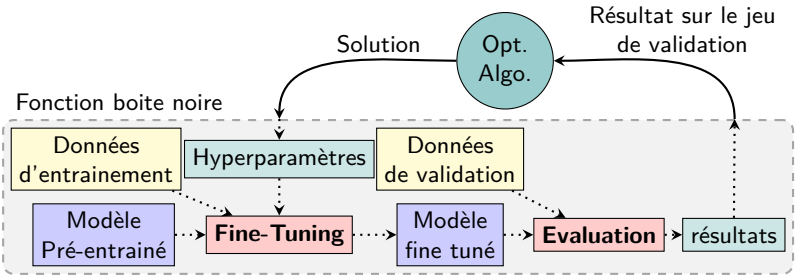


Figure: Fonctionnement général de l'optimisation des hyperparamètres

Formulation du problème

Equation

$$\eta^* \in \arg\max_{\eta \in \mathcal{A}} f(\eta), \quad f : \mathbb{R}^d \to \mathbb{R} \tag{1}$$

Avec η une solution de dimension d et f la fonction représentant l'entrainement et l'évaluation d'un modèle.

Charactéristiques de la fonction f

- Boite-noire : non dérivable
- Couteux : une évaluation se compte en dizaine de minutes
- Bruité : évaluer 2 fois la même solution peut donner un résultat différent
- Variables mixes : les variables sont de plusieurs type (entier, continu...)

Sommaire

1. Introduction

2. Design et Implémentation

3. Résultats et Analyses

4. Conclusion

Espace de Recherche

Hyperparamètres	Plage d'Optimisation		Type	Conversion
	Borne Inf.	Borne Sup.		
Learning Rate	-10	-1	log.	$f(x) = 10^x$
LoRA rank	1	64	ent.	$f(x) = \text{round}(x)$
LoRA scale	1	64	ent.	$f(x) = \text{round}(x)$
Dropout	0	0.5	cont.	$f(x) = x$
Weight Decay	-3	-1	log.	$f(x) = 10^x$

Table: Résumé de l'espace de recherche

- Variables mixes : étape de conversion nécessaire
- Aucun *a priori* sur l'importance de chaque variable

Strategie de Recherche : Optimisation Bayésienne par Process Gaussien

Principe

Utiliser un substitut moins cher à optimiser pour explorer l'espace de recherche

Algorithme

- Echantillon de n Points (LHS)

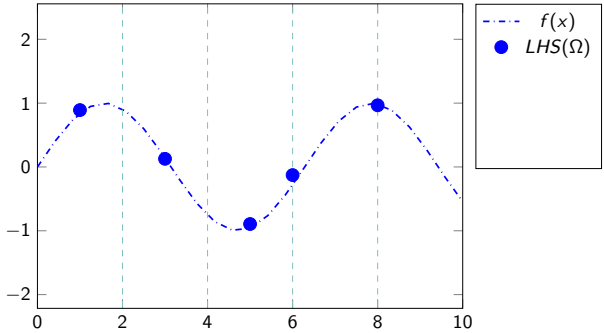


Figure: Example d'un surrogate sur une fonction en 1D

Strategie de Recherche : Optimisation Bayésienne par Process Gaussien

Principe

Utiliser un substitut moins cher à optimiser pour explorer l'espace de recherche

Algorithme

- Echantillon de n Points (LHS)
- Entrainer le Process Gaussien (GP)

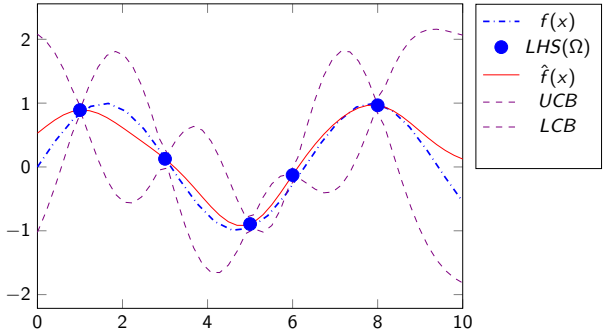


Figure: Example d'un surrogate sur une fonction en 1D

Strategie de Recherche : Optimisation Bayésienne par Process Gaussien

Principe

Utiliser un substitut moins cher à optimiser pour explorer l'espace de recherche

Algorithme

- Echantillon de n Points (LHS)
- Entrainer le Process Gaussien (GP)
- Optimiser la fonction d'acquisition
- Evaluer ce nouveaux point

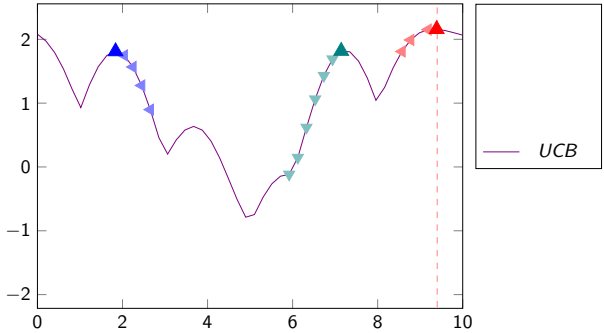


Figure: Example d'un surrogate sur une fonction en 1D

Stratégie de Recherche : Simultaneous Optimistic Optimization

K-section successive de l'espace, en évaluant le centre de chaque sous-espace. Maximum une expansion /itération/profondeur.

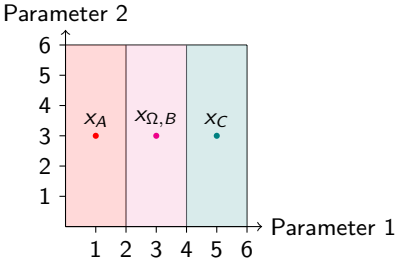


Figure: Partition de l'espace de recherche par SOO

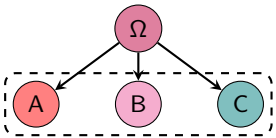


Figure: Arbre correspondant à SOO

Stratégie de Recherche : Simultaneous Optimistic Optimization

K-section successive de l'espace, en évaluant le centre de chaque sous-espace. Maximum une expansion /itération/profondeur.

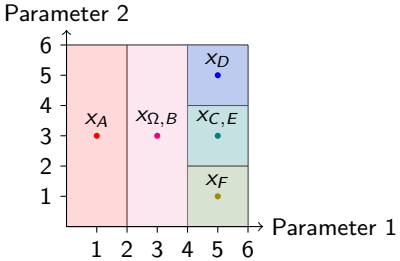


Figure: Partition de l'espace de recherche par SOO

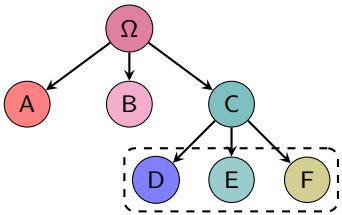


Figure: Arbre correspondant à SOO

Stratégie de Recherche : Simultaneous Optimistic Optimization

K-section successive de l'espace, en évaluant le centre de chaque sous-espace. Maximum une expansion /itération/profondeur.

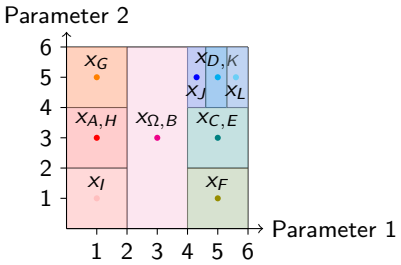


Figure: Partition de l'espace de recherche par SOO

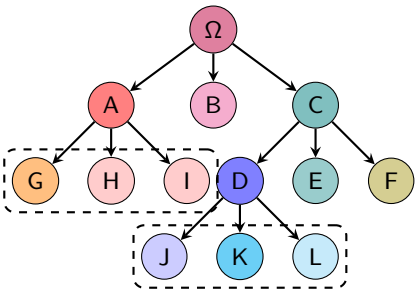


Figure: Arbre correspondant à SOO

Stratégie de Recherche : Bayesian Multi-Scale Optimistic Optimization (BaMSOO)

Décomposition suivant SOO, mais utilisant des Process Gaussien pour éviter les évaluations non prometteuses.

Evaluation par BaMSOO

- If $UCB(x) > f^+$:
  - $g(x) = f(x)$ *r*eal evaluation
- Else :
  - $g(x) = LCB(x)$ *u*se LCB to replace $f(x)$

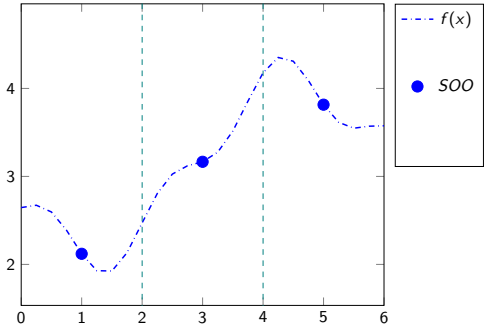


Figure: Illustration de l'Algorithme BaMSOO

Stratégie de Recherche : Bayesian Multi-Scale Optimistic Optimization (BaMSOO)

Décomposition suivant SOO, mais utilisant des Process Gaussien pour éviter les évaluations non prometteuses.

Evaluation par BaMSOO

- If $UCB(x) > f^+$:
  - $g(x) = f(x)$ *r*eal evaluation
- Else :
  - $g(x) = LCB(x)$ *u*se LCB to replace $f(x)$

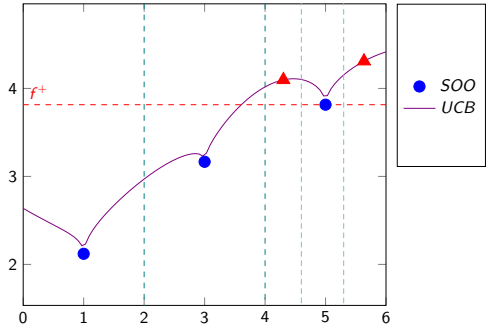


Figure: Illustration de l'Algorithme BaMSOO

Stratégie de Recherche : Bayesian Multi-Scale Optimistic Optimization (BaMSOO)

Décomposition suivant SOO, mais utilisant des Process Gaussien pour éviter les évaluations non prometteuses.

Evaluation par BaMSOO

- If $UCB(x) > f^+$:
  - $g(x) = f(x)$ *r*eal evaluation
- Else :
  - $g(x) = LCB(x)$ *u*se LCB to replace $f(x)$

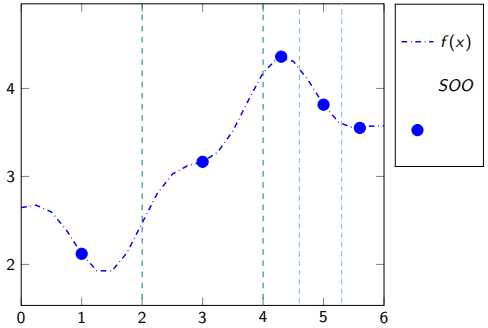


Figure: Illustration de l'Algorithme BaMSOO

Stratégie de Recherche : Bayesian Multi-Scale Optimistic Optimization (BaMSOO)

Décomposition suivant SOO, mais utilisant des Process Gaussien pour éviter les évaluations non prometteuses.

Evaluation par BaMSOO

- If $UCB(x) > f^+$:
  - $g(x) = f(x)$ *r*eal evaluation
- Else :
  - $g(x) = LCB(x)$ *u*se LCB to replace $f(x)$

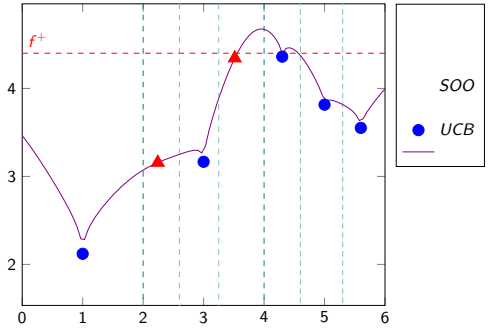


Figure: Illustration de l'Algorithme BaMSOO

Stratégie de Recherche : Bayesian Multi-Scale Optimistic Optimization (BaMSOO)

Décomposition suivant SOO, mais utilisant des Process Gaussien pour éviter les évaluations non prometteuses.

Evaluation par BaMSOO

- If $UCB(x) > f^+$:
  - $g(x) = f(x)$ *r*eal evaluation
- Else :
  - $g(x) = LCB(x)$ *u*se LCB to replace $f(x)$

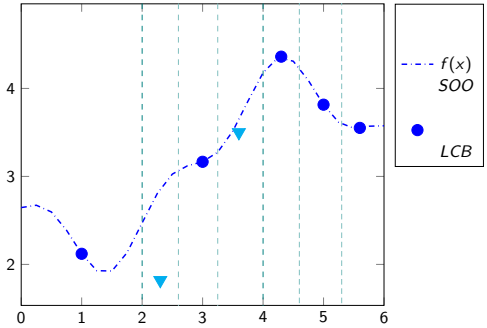


Figure: Illustration de l'Algorithme BaMSOO

Implémentation

- Programmation Orienté Object en Python
- Travail de documentation : *readme*, indication de type...
- Objectif : permettre le réusage
- Utilisable en ligne de commande pour Grid5000
- Intégralement open-source[2]

[2] `https://github.com/Kiwy3/BO_PBO_HPO_LLM`

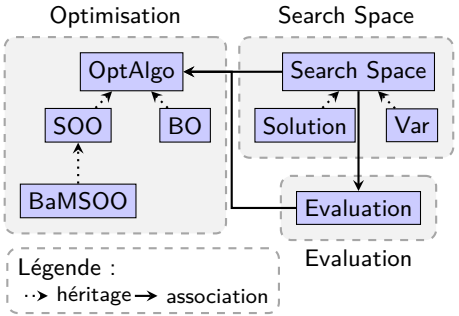


Figure: Diagramme de l'implémentation

Sommaire

1. Introduction

2. Design et Implémentation

3. Résultats et Analyses

4. Conclusion

Echantillonnage par Latin Hypercube Sampling (LHS)

Objectif : Explorer l'espace et proposer une borne inférieure

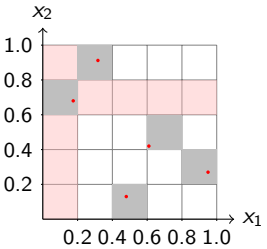


Figure: Illustration du Latin Hypercube Sampling avec $g = 5$

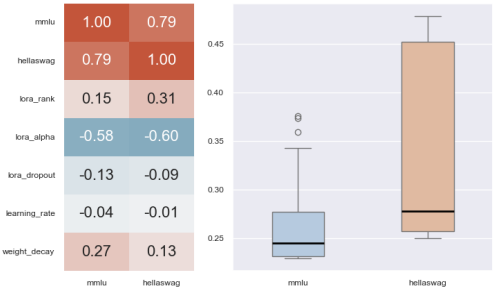


Figure: Résumé des résultats par sampling

Résultats des 3 algorithms

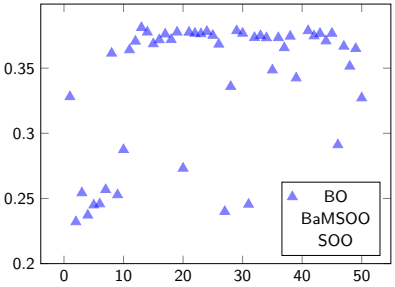


Figure: Résultats sur MMLU (test)

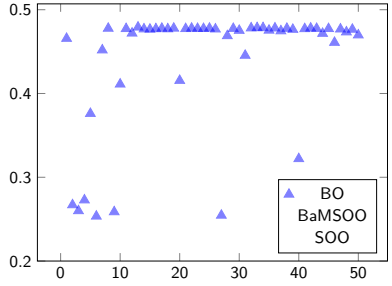


Figure: Résultats sur Hellaswag (Validation)

Résultats des 3 algorithms

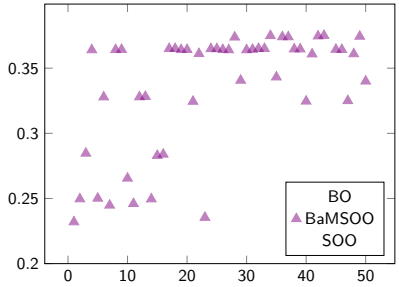


Figure: Résultats sur MMLU (test)

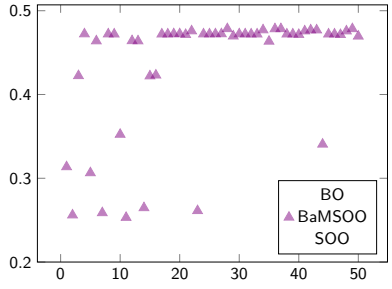


Figure: Résultats sur Hellaswag (Validation)

Résultats des 3 algorithms

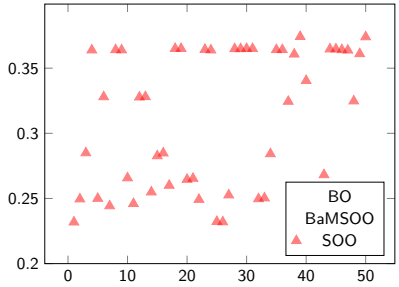


Figure: Résultats sur MMLU (test)

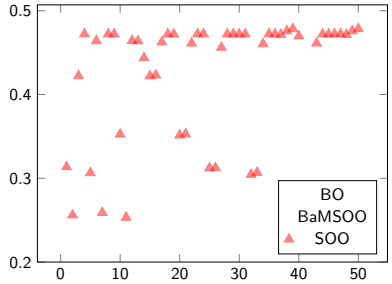


Figure: Résultats sur Hellaswag (Validation)

Résultats des 3 algorithms

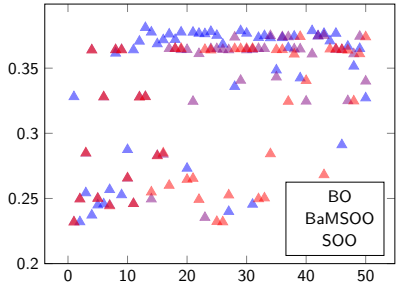


Figure: Résultats sur MMLU (test)

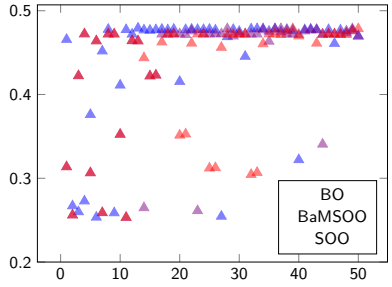


Figure: Résultats sur Hellaswag (Validation)

Analyse

Jeu de données	Borne Inf.[1]	Borne Sup.[2]	BO-GP	SOO	BaMSOO
Hellaswag (validation)	47.90	*41.5*	**47.91**	47.84	47.84
MMLU (testing)	37.61	49.3	**38.11**	37.42	37.50

Table: Bornes et meilleurs résultats sur les 2 jeu de données

1 : expérience avec LHS; 2 : Fine tuning par Meta

Points clés

- Borne Sup. sur Hellaswag non pertinente
- Seul BO arrive au dessus de LHS
- BaMSOO n'améliore que peu SOO
- principe de BaMSOO fonctionnel (visible annexe 8)
- Espace de solution n'évolue que peu, le retravailler pour mesurer pleinement la performance des algorithmes

Perspectives

Poursuite du travail

- Retour sur l'article et présentation en conférence (si validation)
- Elargissement de l'espace de recherche
- Diversification sur les modèles/données

Optimisation fractale parallèle enrichie par approche bayésienne[3]

- Généralisation de l'hybridation décomposition/bayésien dans un cadre parallèle
- 5 moyens d'exploiter le *surrogate* pour améliorer l'optimisation fractale

[3] *P*arallel Bayesian-enhanced Fractals Optimization

Sommaire

1. Introduction

2. Design et Implémentation

3. Résultats et Analyses

4. Conclusion

Résultats du stage

Implémentation d'OHP pour du Fine Tuning de LLM

Preuve du concept d'utilisation des algorithmes utilisés, ainsi qu'une base pour de futur travaux de l'équipe

Comparaison entre approche bayésien/décomposition pour fonction couteuse

Incluant l'exploration d'une première partie de l'hybridation des deux
Tend vers une généralisation d'une approche par décomposition améliorée par l'utilisation d'un *surrogate*

Apprentissage

1ere expérience de recherche

- Apprentissage de la rigueur
- gestion de la littérature
- première écriture ...

Programmation pour une démarche de recherche

- Habitude de programmation pour l'optimisation globale
- Prototypage et versionnage
- Transmission pour l'équipe
- Approche du paralellisme

Poursuite du projet profressionel

Poursuite en recherche

Confirmation de l'attrait pour le domaine

Début d'une thèse en mars

Sujet : Ecological and economic logistics service network design : Models and Decision Support Algorithms
Equipe INOCS, au sein de l'INRIA Lille, dirigé par Frederic Semet

Merci.

Annexe 1 : Travaux connexes

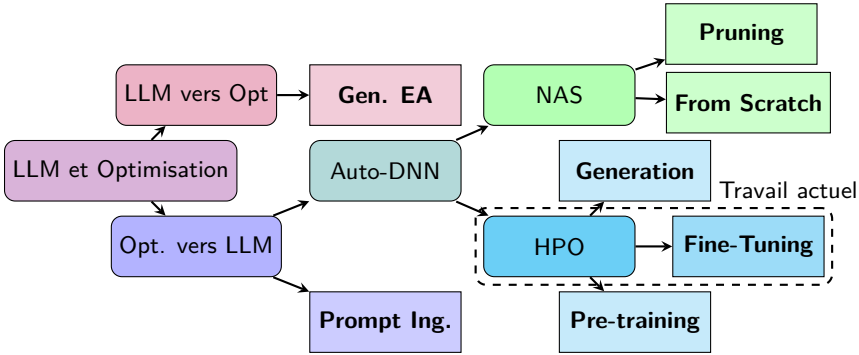


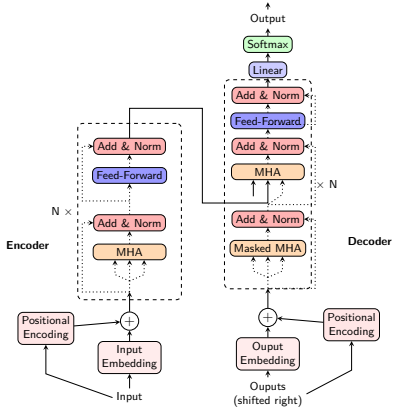
Figure: Classification des travaux similaires

Annexe 2 : Stratégie d'Evaluation de Solutions

Implémentation

- Fine Tuning
  - Modèle : LlaMa-3.2-1B
  - Jeu de données d'entrainement : Alpaca
  - LitGPT framework : basé sur Pytorch, facilite le Fine Tuning de LLM
- Evaluation
  - librairie lm_eval : standard pour l'évaluation de LLM
  - Evaluation par la précision sur des jeu de données Benchmark : Hellaswag et MMLU

Annexe 3 : Transformer architecture



Annexe 4 : Multi-Head Attention

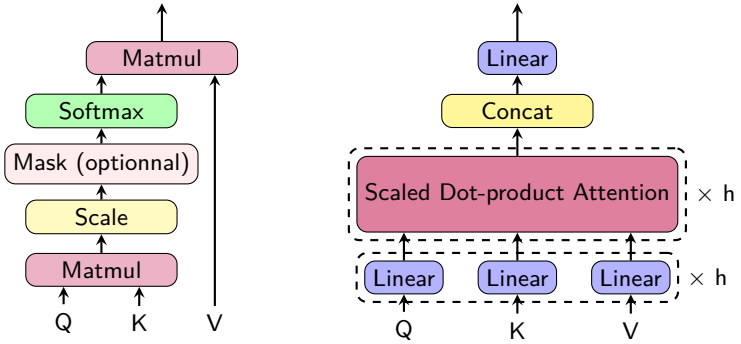


Figure: Illustration du mécanisme d'auto-attention : A droite le mécanisme complet, a gauche le *Scaled Dot-product Attention*

Annexe 5 : Low Rank Adaptation (LoRA)

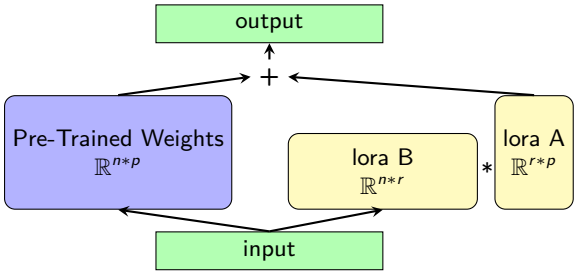


Figure: Illustration de l'application du Low Rank Adaptation (LoRA)

Annexe 6 : Résultats pour BO

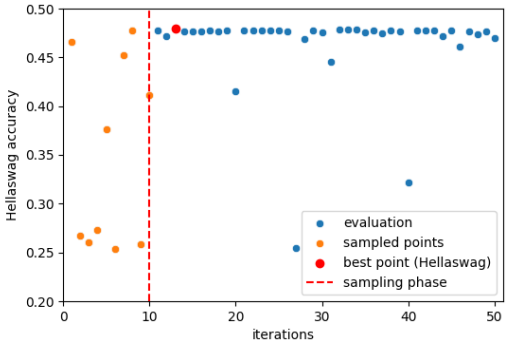


Figure: Evolution des score lors de l'expérience BO

Annexe 7 : Résultats pour SOO

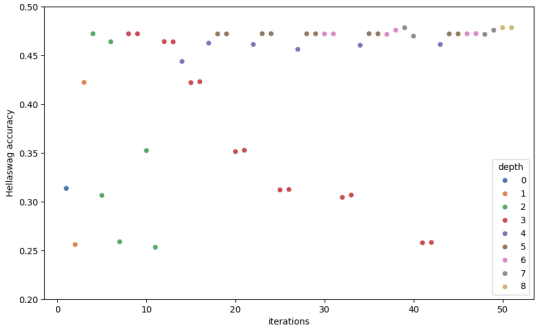


Figure: Evolution des score lors de l'expérience SOO

Annexe 8 : Résultats pour BaMSOO

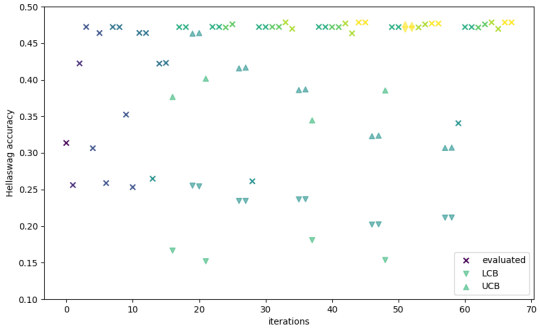


Figure: Evolution des score lors de l'expérience BaMSOO

Annexe 9 : Aperçu du Bayésien Fractal

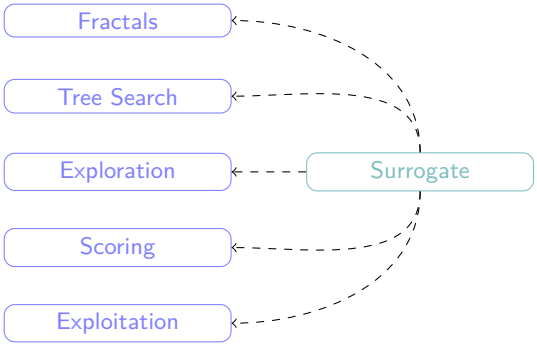


Figure: Interaction du composant Surrogate avec les éléments de base de la décomposition fractale

Annexe 10 : Modèle Beamer UTT

Figure: Presentation du template beamer sur github : lien