

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

Large Language Models

Point clés

- ▶ Etat de l'art pour le traitement de langage naturel.
- ▶ Réseaux de Neurones avec une architecture basé sur le transformer¹ (annexe 3)
- ▶ Taille : entre 1 et 405 Milliards de neurones

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

Auto-attention

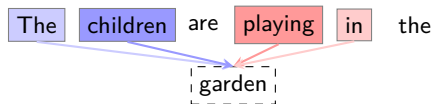


Figure: Illustration du mécanisme d'auto-attention

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

Optimisation des Hyperparamètres (OHP)

Hyperparamètres

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

Objectifs

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

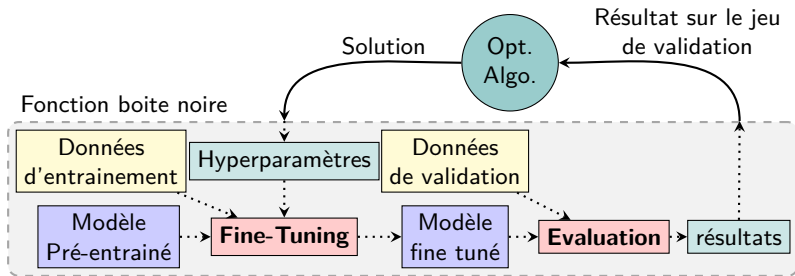


Figure: Fonctionnement général de l'optimisation des hyperparamètres

Sommaire

1. Introduction

2. Design et Implémentation

3. Résultats et Analyses

4. Conclusion

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.

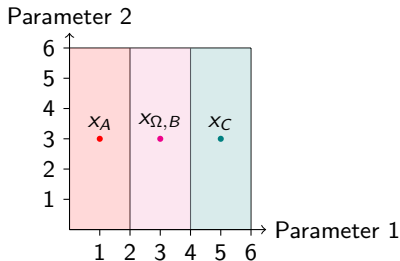


Figure: Partition de l'espace de recherche par SOO

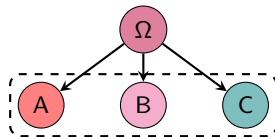


Figure: Arbre correspondant à S00

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 $UCB(x) > f^+$:
 - $g(x) = f(x)$ real evaluation
- ▶ Else :
 - $g(x) = LCB(x)$ use LCB to replace $f(x)$

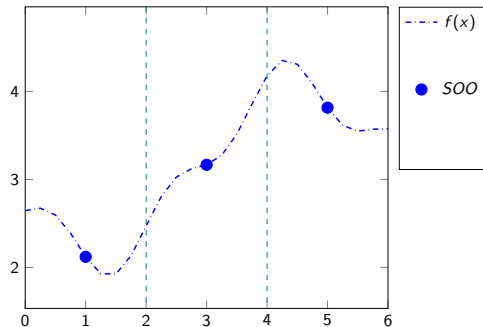


Figure: Illustration de l'Algorithme BaMSOO

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 $UCB(x) > f^+$:
 - $g(x) = f(x)$ real evaluation
- ▶ Else :
 - $g(x) = LCB(x)$ use LCB to replace $f(x)$

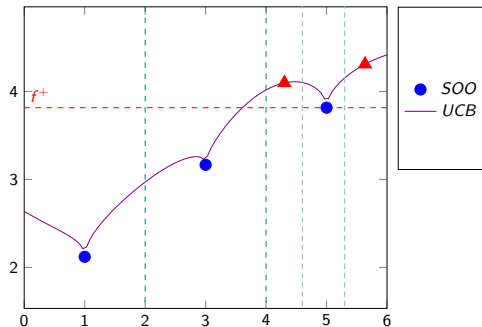


Figure: Illustration de l'Algorithme BaMSOO

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 $UCB(x) > f^+$:
 - $g(x) = f(x)$ real evaluation
- ▶ Else :
 - $g(x) = LCB(x)$ use LCB to replace $f(x)$

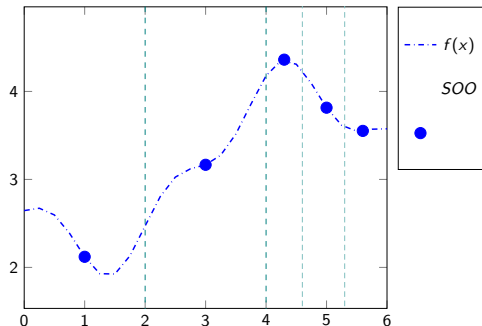


Figure: Illustration de l'Algorithme BaMSOO

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-source²

²https://github.com/Kiwy3/BO_PBO_HPO_LLM

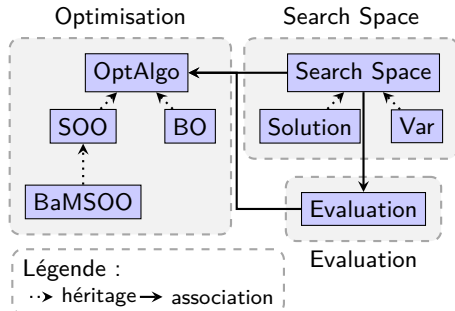


Figure: Diagramme de l'implémentation

Sommaire

1. Introduction

2. Design et Implémentation

3. Résultats et Analyses

4. Conclusion

Résultats des 3 algorithmes

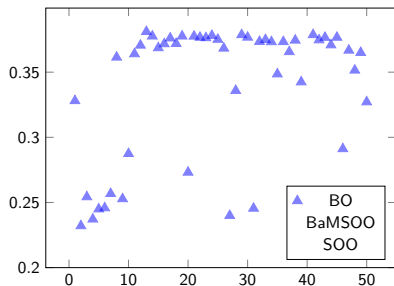


Figure: Résultats sur MMLU (test)

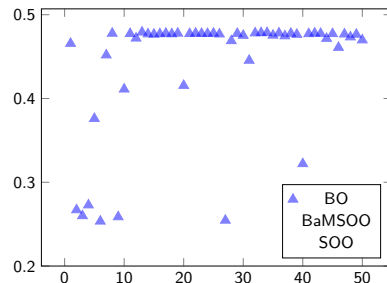


Figure: Résultats sur Hellaswag (Validation)

Résultats des 3 algorithms

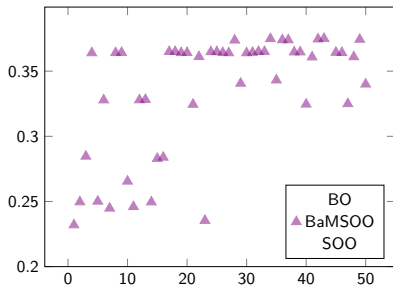


Figure: Résultats sur MMLU (test)

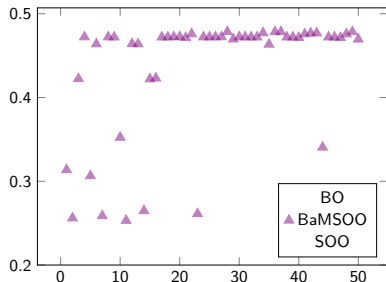


Figure: Résultats sur Hellaswag (Validation)

Résultats des 3 algorithms

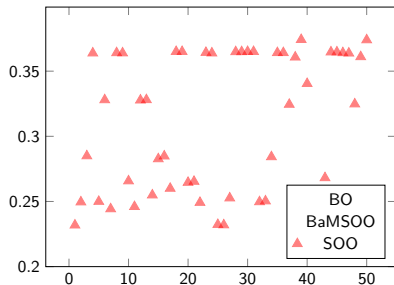


Figure: Résultats sur MMLU (test)

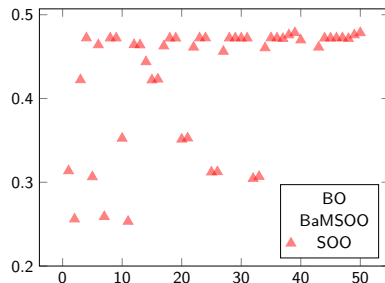


Figure: Résultats sur Hellaswag (Validation)

Résultats des 3 algorithms

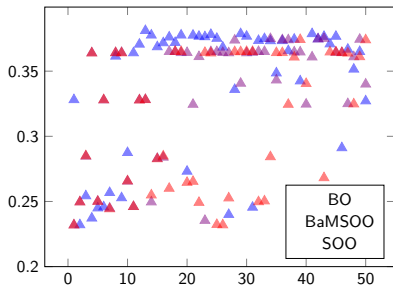


Figure: Résultats sur MMLU (test)

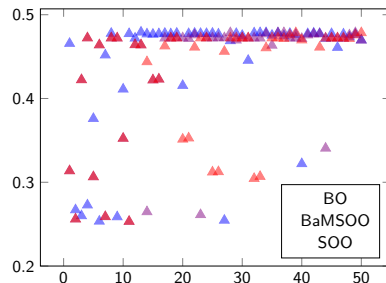


Figure: Résultats sur Hellaswag (Validation)

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 bayésienne³

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

³Parallel Bayesian-enhanced Fractals Optimization

Sommaire

4. Conclusion

Résultats du stage

Une conclusion

Apprentissage

Une conclusion

Poursuite du projet professionnel

Une conclusion

Merci.

Annexe 1 : Travaux connexes

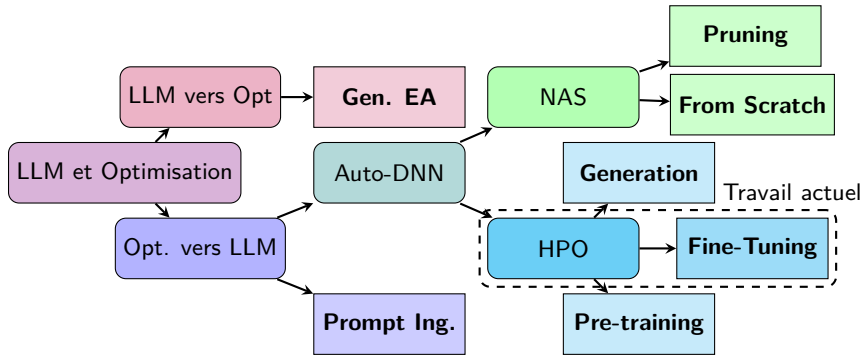


Figure: Classification des travaux similaires

Annexe 2 : Stratégie d'Evaluation de Solutions

Implémentation

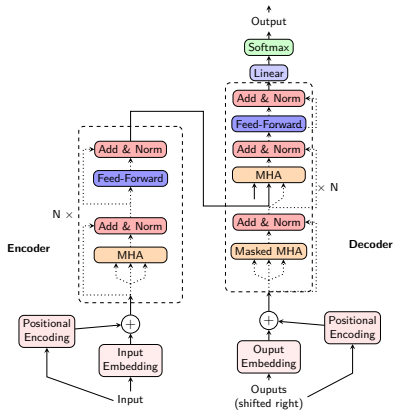
► Fine Tuning

- Modèle : LLaMa-3.2-1B
- Jeu de données d'entraînement : 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

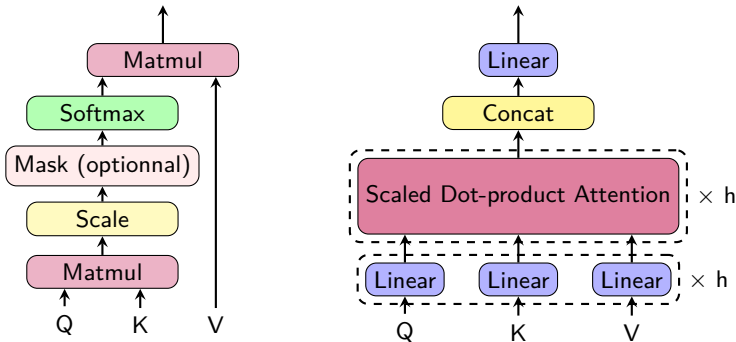


Figure: Illustration du mécanisme d'auto-attention : A droite le mécanisme complet, à gauche le *Scaled Dot-product Attention*

Annexe 5 : Low Rank Adaptation (LoRA)

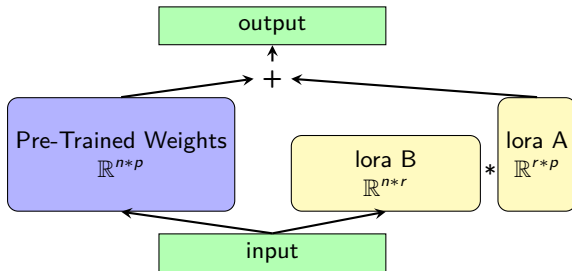


Figure: Illustration de l'application du Low Rank Adaptation (LoRA)

Annexe 7 : Résultats pour S00

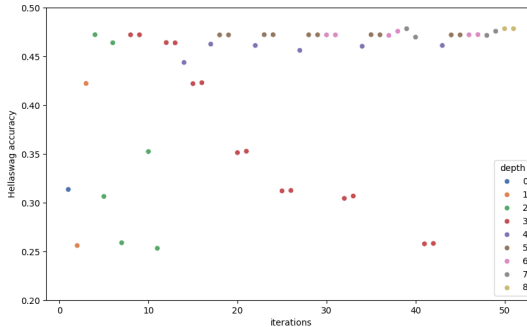


Figure: Evolution des score lors de l'expérience SOO

Annexe 9 : Aperçu du Bayésien Fractal

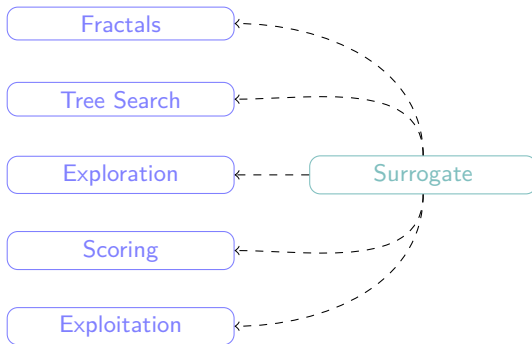


Figure: Interaction du composant Surrogate avec les éléments de base de la décomposition fractale

Annexe 10 : Modèle Beamer UTT

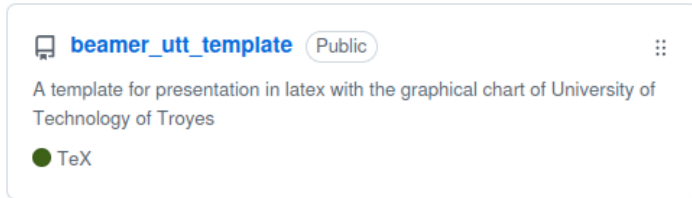


Figure: Presentation du template beamer sur github : lien