

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

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 1)
- ▶ Taille : entre 1 et 405 Milliards de neurones

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

Auto-attention

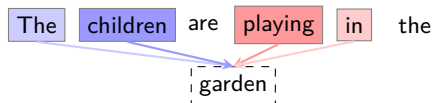


Figure: Illustration du mécanisme d'auto-attention

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

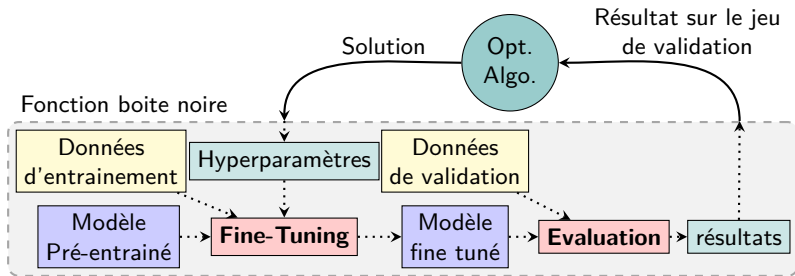


Figure: Fonctionnement général de l'optimisation des hyperparamètres

Travaux connexes

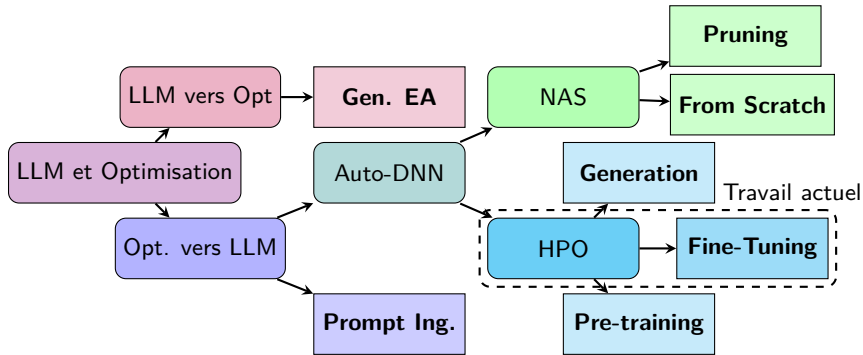


Figure: Classification des travaux similaires

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 mixtes : 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)
 - Entraîner le Process Gaussien (GP)

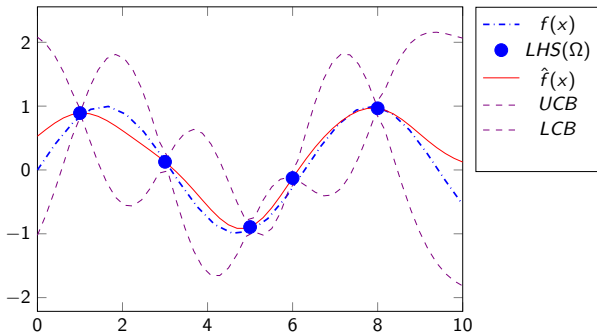


Figure: Exemple d'un surrogate sur une fonction en 1D

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)
 - Entraîner le Process Gaussien (GP)
 - Optimiser la fonction d'acquisition
 - Evaluer ce nouveaux point

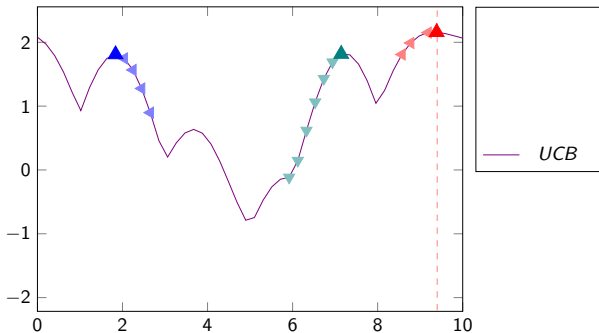


Figure: Exemple d'un surrogate sur une fonction en 1D

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 $f(x)$

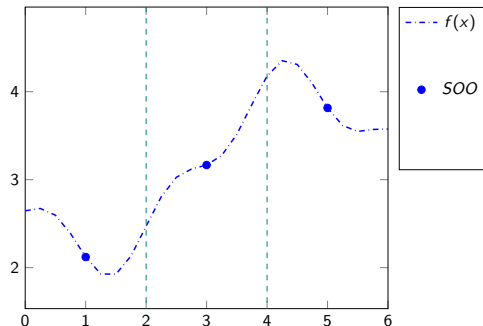


Figure: Illustration de l'Algorithme BaMSOO

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 $f(x)$

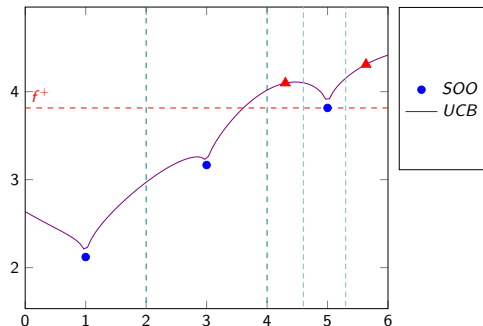


Figure: Illustration de l'Algorithme BaMSOO

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 $f(x)$

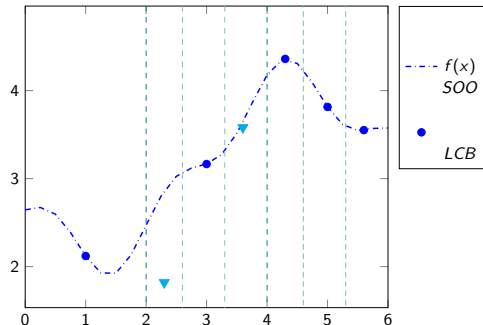


Figure: Illustration de l'Algorithme BaMSOO

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

Sommaire

1. Introduction

2. Design et Implémentation

3. Résultats et Analyses

4. Conclusion

Résultats des 3 algorithmes

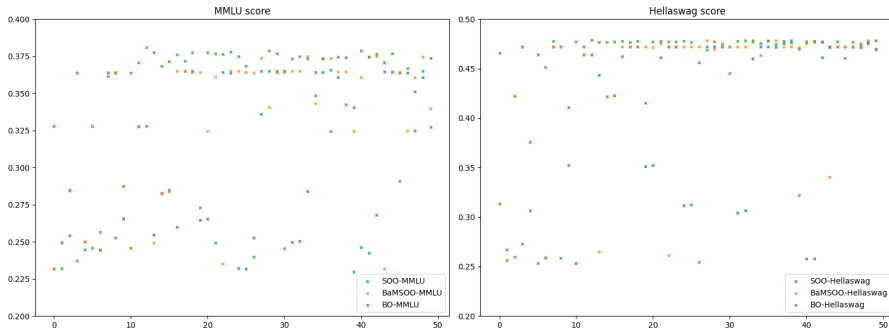


Figure: Résumé des résultats par sampling. Détails par algorithme dans les annexes 4, 5 et 6

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 6)
- ▶ 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

Généralisation hors LLM

- ▶ Parallélisation d'algorithme d'optimisation de fonction coûteuse pour Exascale (project Exa-MA)
- ▶ Intégration de substitut dans les algorithme parallèle à Partition ³

³Partition-based Parallel Bayesian Optimisation

Sommaire

4. Conclusion

Résultats du stage

Une conclusion

Apprentissage

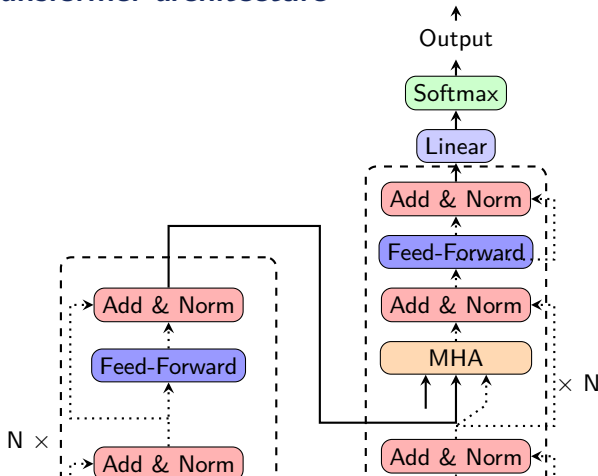
Une conclusion

Poursuite du projet professionnel

Une conclusion

Merci.

Annexe 1 : Transformer architecture



Annexe 2 : Multi-Head Attention

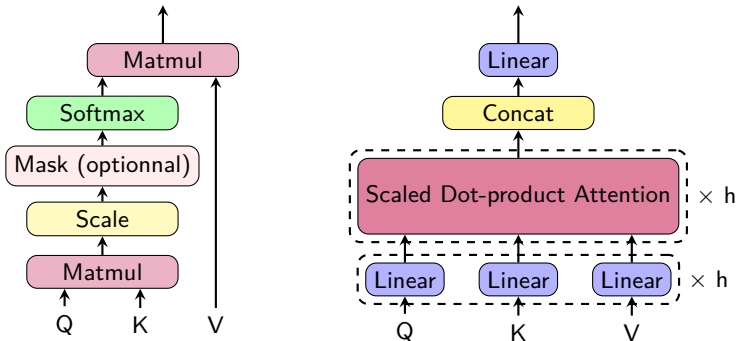


Figure: Illustration du mécanisme d'auto-attention : A droite le mécanisme complet, à gauche le *Scaled Dot-product Attention*

Annexe 5 : Résultats pour S00

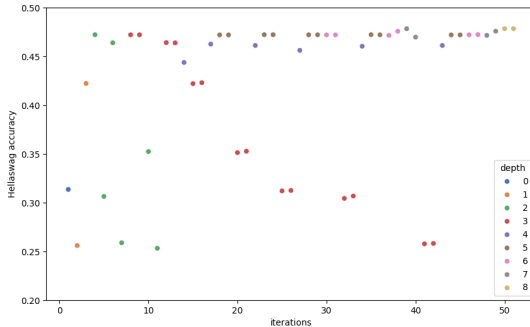


Figure: Evolution des score lors de l'expérience SOO

