Hyperparameter Optimization for Large Language Model Instruction-Tuning

Christophe Tribes[1] Sacha Benarroch-Lelong[1]
Peng Lu[2,3] Ivan Kobyzev[2]

[1]GERAD and Polytechnique Montréal, [2]Huawei Noah's Ark Lab, [3]RALI, Université de Montréal

Abstract

The fine-tuning of Large Language Models (LLMs) has enabled them to recently achieve milestones in natural language processing applications. The emergence of ever larger LLMs has paved the way for more efficient fine-tuning methods. Among these, the Low-Rank Adaptation (LoRA) method keeps most of the weights of the pre-trained LLM frozen while introducing a low-rank decomposition of the weight matrix, enabling the tuning of only a very small proportion of the network. The performance on downstream tasks of models fine-tuned with LoRA heavily relies on a set of hyperparameters including the rank of the decomposition. In this work, we examine the whole pipeline of performing fine-tuning and validation on a pre-trained LLM as a blackbox. Two blackbox optimization (BBO) techniques (NOMAD and NNI-TPE) are compared to explore the space of hyperparameters, both achieving a boost in performance and human alignment of the tuned model.

Motivation

Parameter Efficient Fine Tuning (PEFT) methods such as LoRA are quite sensitive to the choice of hyperparameters. In this work we investigate how performing hyperparameter optimization (HPO) through blackbox optimization (BBO) techniques can better the instruction-tuned results of LLMs.

Contributions

- Apply two blackbox optimization (BBO) techniques to optimize LoRA fine-tuning hyperparameters :
  - MADS (Mesh Adaptive Direct Search) implemented in NOMAD;
  - TPE (Tree-structured Parzen Estimator) implemented in NNI (Neural Network Intelligence).
- For the best sets of hyperparameters we study the correlation between validation losses and downstream instruction-following tasks scores.
- Full paper:

Method

- Fine-tuning pipeline (inner loop):
  - Backbone model: LLaMA 2 (7 billions parameters).
  - PEFT technique: LoRA with AdamW.
  - Fine-tuning dataset: 54k sized instruction-following dataset: mix of entries from Stanford Alpaca Project dataset and Databricks' Dolly dataset.
  - Validation dataset: 13k-sized entries from Alpaca and Dolly.
  - HuggingFace Tranformers API: handling model, training and validation on datasets.
  - Hardware: Training and validation conducted on four NVIDIA-A100 GPUs with 80 GBs memory.
- HPO outer loop:
  - Objective: minimize the validation loss by adapting LoRA fine-tuning hyperparameters.
  - Blackbox optimization: NOMAD and NNI-TPE.
  - Iterations: 100 evaluations per optimization.

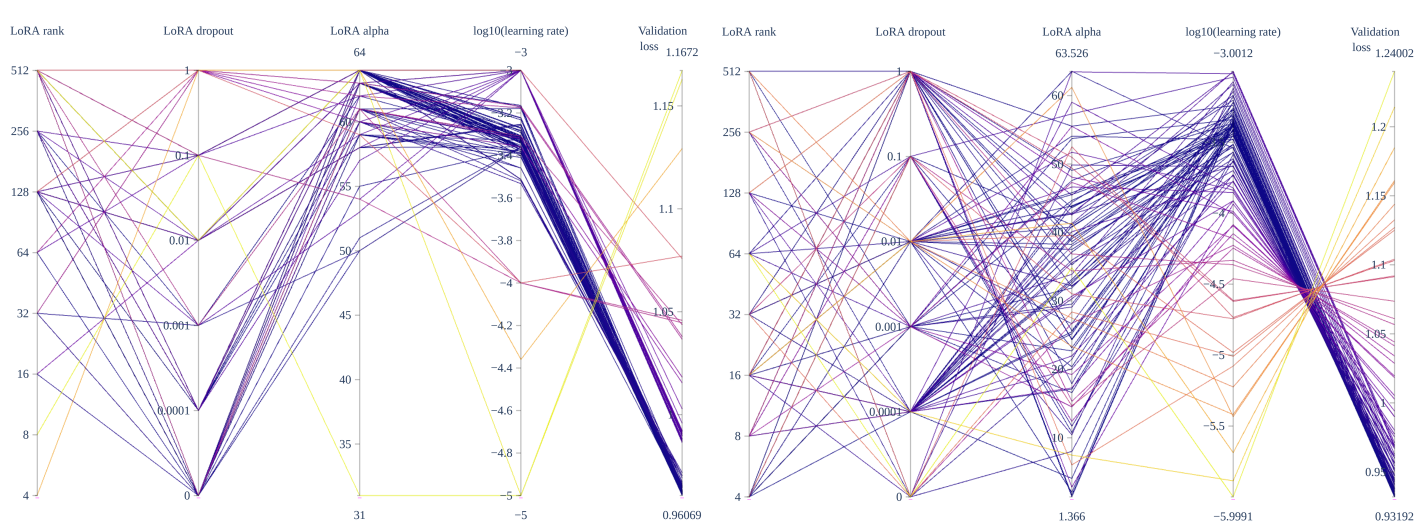
Parameter	Type	Possible values	Default value
LoRA rank	int	$\{4, 8, 16, 32, 64, 128, 256, 512\}$	8
LoRA α	int	$[\![1, 64]\!]$	32
AdamW dropout	float	$\{0, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1\}$	0.1
AdamW lr	float	$[10^{-6}, 10^{-3}]$	10^{-5}

Table: Treatment of hyperparameters in NOMAD, possible and default values

- Perform post optimization evaluation of the best candidate models on a series of downstream instruction-following tasks of quite different natures.
  - Instruct-Eval benchmarks: MMLU, BBH, DROP and HumanEval.
- Human evaluation to check whether the generated results are aligned with human preferences:
  - Dataset: 30 questions from Vicuna Human Preference.
  - Compared models: NOMAD best vs default LoRA hyperparameters.
  - Methodology: Ask preference of human evaluators to answers provided by models.

Results

- Hyperparameters tested by NOMAD (left) and NNI-TPE (right)



- Instruct-Eval performance measures:
  - HPO results in better models.
  - But lower validation losses do not necessarily translate into higher benchmark scores.

Method		min	max	avg.	st. d.
NOMAD	MMLU	45.88	46.7	46.24	0.29
	HumanEval	14.63	18.9	16.94	1.52
NNI-TPE	MMLU	45.49	46.56	46.08	0.31
	HumanEval	14.02	16.46	15.24	0.91
Default HPs	MMLU	43.56			
	HumanEval	15.24			

Table: Statistics of the 10 best models on downstream instruction-following tasks

- HP-tuned model has a clear human preference compared to the default one by an overall preference score of 5%.

Conclusion

- Hyperparameters optimization using blackbox optimization algorithms improves the performance of fine-tuned LLMs on downstream tasks and human evaluation.
- The validation losses are not perfectly aligned with downstream tasks scores.
- Future work: develop an efficient and robust methodology to pickup a single best model. We believe this can be achieved by guiding the blackbox optimization to consider more criteria into the HPO problem.