

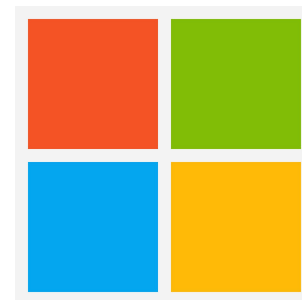
Découvrir LoRA avec 2 articles

- (Original Paper) LoRA: Low Rank Adaptation
- Limitations of LoRA

$$W_0 + \Delta W = W_0 + BA$$

LoRA: Low Rank Adaptation (Edward Hu et al.)

- **ICLR 2022**
- **Microsoft**
- **SOTA benchmarks; Modèles Open source; 10k citations**



Computational Limits of Low-Rank Adaptation for Transformer-Based Models (Yao-Chieh Hu et al.)

- **Récent !** (soumis pour **ICLR 2025**)
- **Universitaires américains**
- Surtout sur la **Complexité temporelle de LoRA**



L'adaptation avant LoRA

Adaptation: Modifier un Language Model pour **une tâche spécialisée:**

- écrire des résumés,
- commenter un article de recherche,
- écrire un TP de C++,
- etc ...

L'adaptation avant LoRA

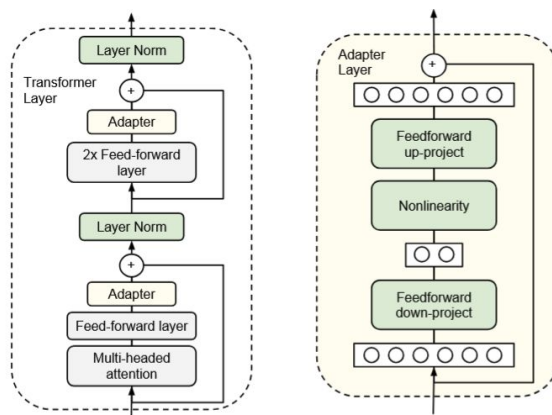
Full fine-tuning

Idée simple: Changer tous les poids du modèle en reprenant le training avec un nouveau set



- + Bonne Performance
- Cher à l'entraînement
- Résultat lourd à stocker

Adapters Layers



Parameter-Efficient Transfer Learning for NLP
(Google, 2019)

- Latence à l'inférence

Prompt-based adaptation

Few-Shot Learning

Démonstrations dans le prompt
"Language Models are Few-Shot Learners" (OpenAI, NeurIPS, 2020)

Prefix Tuning

Apprendre l'embedding d'un préfixe d'entrée

"Prefix-Tuning: Optimizing Continuous Prompts for Generation" (Li&Liang2021)

- + Modèle reste identique
- Plus faible performance

LoRA: Fine-tuning *plus efficace !*

Full fine-tuning

- + Bonne performance
- Cher à l'entraînement
- Résultat lourd à stocker

Low-Rank Adaptation

- + Performance conservée
- + Plus économe à l'entraînement
- + Résultat plus léger à stocker
- + Vitesse d'inférence conservée

L'adaptation est plus facile à **implémenter** et à **scale**

LoRA: L'intuition

- **Observation** : Les matrices $n \times n$ de poids dans les modèles pré-entraînés ont un “rang intrinsèque” $\ll n$.
- **Hypothèse** : les modifications ΔW de ces matrices lors du fine-tuning sont aussi de bas “rang intrinsèque”.

$$W = W_0 + \Delta W$$

=> Avoir des **milliards de degrés de libertés** dans ΔW c'est **overkill**

LoRA: L'intuition

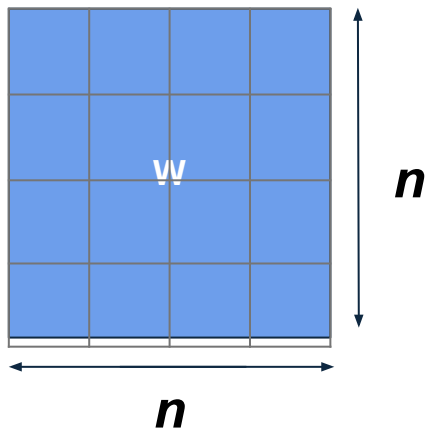
=> Avoir des **milliards de degrés de libertés** dans ΔW c'est **overkill**

- LoRA apprend un **sous-espace de faible dimension** qui capture les **modifications les plus importantes**
- Thème **récurrent** de “Low-Rank Structures” en Machine Learning

LoRA = **Low Rank** **A**daptation

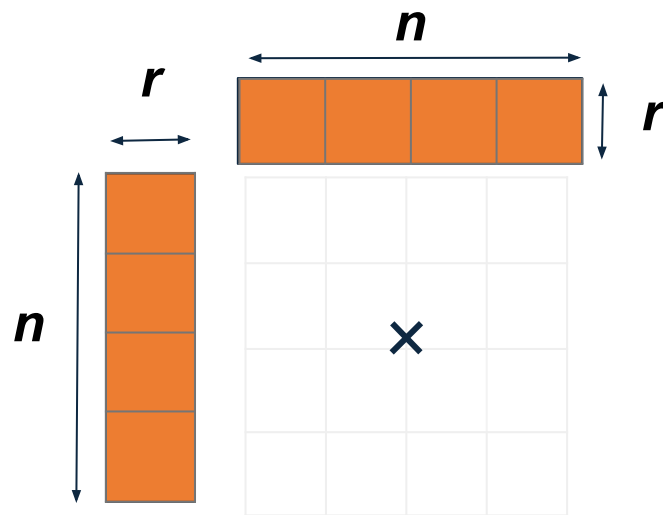
2 manières d'obtenir une matrice $n \times n$:

Rang = n



Définie par n^2 coefficients

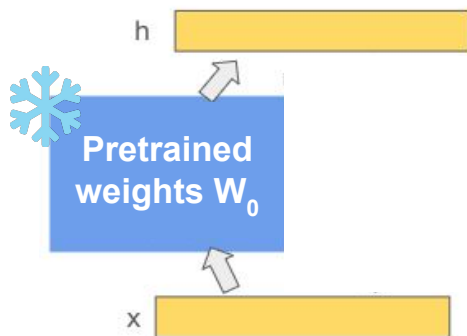
Rang = r



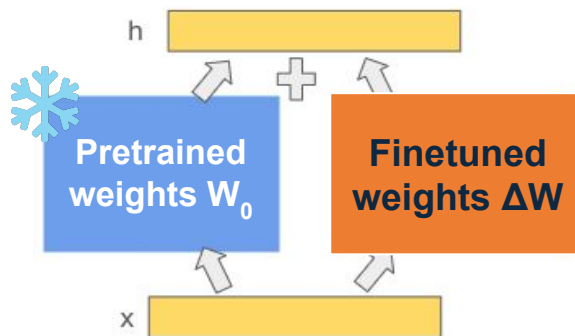
Définie par $2 \times n \times r$ coefficients

LoRA: Comment ça marche ?

No tuning

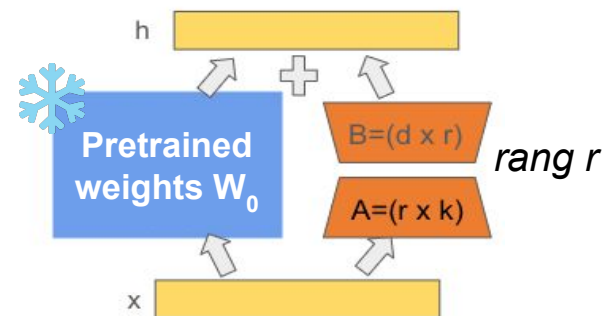


Full fine-tuning



newer dataset

Low-Rank Adaptation



$$W_0 + \Delta W = W_0 + BA$$



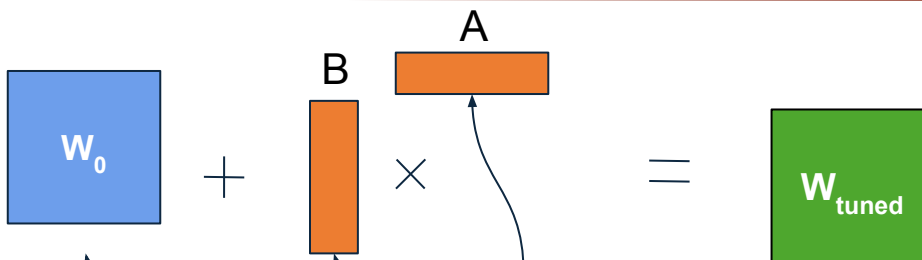
newer dataset

LoRA: Comment ça marche ?

Full fine-tuning



LoRA



Poids fixe pendant
le fine tuning

Poids variable
pendant le fine tuning

$$W_0 + \Delta W = W_0 + BA$$

matrices:
K, Q, V
(non carrés)

Que disent les benchmarks ?

WikiSQL benchmark (fine-grained NLG):

80k exemples

Table:

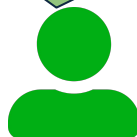
Player	Country	Points	Winnings (\$)
Steve Stricker	United States	9000	1260000
K.J. Choi	South Korea	5400	756000
Rory Sabbatini	South Africa	3400	4760000
Mark Calcavecchia	United States	2067	289333
Ernie Els	South Africa	2067	289333

Question: What is the points of South Korea player?

SQL: SELECT Points WHERE Country = South Korea

Answer: 5400

Génère-moi une **requête SQL** pour accéder aux **points du joueur coréen**



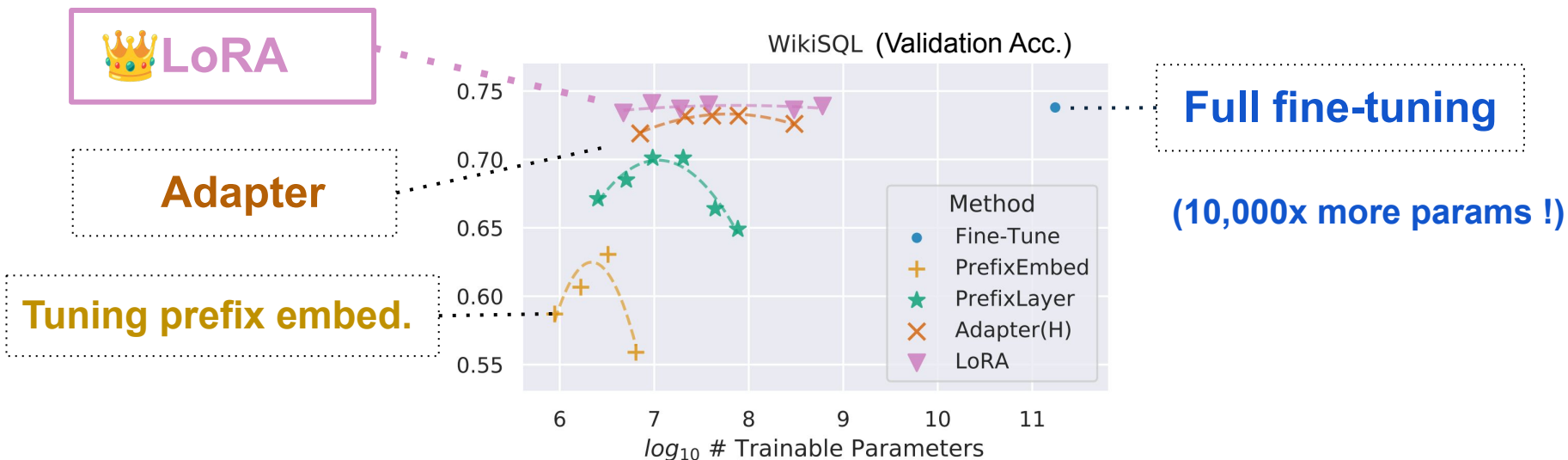
???



source: “Seq2SQL: Generating Structured Queries from Natural Language using Reinforcement Learning”(Zhong et al. Salesforce, 2017)

Que disent les benchmarks ?

WikiSQL benchmark (fine-grained NLG):




source: LoRA: Low Rank Adaptation (Edward Hu et al., NeurIPS 2022)

(all w/ GPT-3 175B)

Que disent les benchmarks ?

GLUE: General Language Understanding Evaluation (fine-grained NLU):
Quora Question Pairs, MNLI, Paraphrase Corpus, ...

Model & Method		# Trainable Parameters	MNLI	SST-2	MRPC	CoLA	QNLI	QQP	RTE	STS-B	Avg.
Full fine-tuning	RoB _{base} (FT)*	125.0M	87.6	94.8	90.2	63.6	92.8	91.9	78.7	91.2	86.4
	RoB _{base} (BitFit)*	0.1M	84.7	93.7	92.7	62.0	91.8	84.0	81.5	90.8	85.2
Adapter LoRA 	RoB _{base} (Adpt ^D)*	0.3M	87.1 \pm 0	94.2 \pm 1	88.5 \pm 1.1	60.8 \pm 4	93.1 \pm 1	90.2 \pm 0	71.5 \pm 2.7	89.7 \pm 3	84.4
	RoB _{base} (Adpt ^D)*	0.9M	87.3 \pm 1	94.7 \pm 3	88.4 \pm 1	62.6 \pm 9	93.0 \pm 2	90.6 \pm 0	75.9 \pm 2.2	90.3 \pm 1	85.4
	RoB _{base} (LoRA)	0.3M	87.5 \pm 3	95.1\pm2	89.7 \pm 7	63.4 \pm 1.2	93.3\pm3	90.8 \pm 1	86.6\pm7	91.5\pm2	87.2

source: LoRA: Low Rank Adaptation (Edward Hu et al., NeurIPS 2022)

(all w/ BaseRoBERTa-base)



Les forces de LoRA

- + Capacité d'adaptation
- + 3 x moins de mémoire GPU
- + Résultat plus léger à stocker (30 MB pour un custom GPT)
- + Valable pour tous les Transformers et Diffuseurs

Facile à implémenter pour:

- des POC
- des stages
- les profils personnalisés (Claude AI, ChatGPT, etc)

huggingface.co/docs/peft/ 🤗

? Les limitations de LoRA

1: Manque de fondements théoriques:

- Solution très populaire, **mal comprise théoriquement** (choix de **r**, choix des **blocs**)

2 : L'update des attention blocks est sous-optimale:

- $O(L^2)$ Computational Complexity (**L = taille du contexte**)
- Peut se réduire à $O(L)$ grâce à une normalisation ! (sous certaines conditions)

Theorem 1.2 (Informal Version of **Theorems 3.1** and **A.1**). Given appropriately normalized inputs X , pretrained attention weights W_K^* , W_Q^* , W_V^* , and LoRA matrices $\{\alpha A_\mu B_\mu / r\}_{\mu=K,Q,V}$, there exists an algorithm that solves ALoRAGC in almost linear time $O(L^{1+o(1)})$.

source: Computational Limits of Low-Rank Adaptation for Transformer-Based Models (Yao-Chieh Hu et al.)