IRISA

Fine-tuning des LLM

Formation continue – 1 jour

Fine-tuning des LLM

Fine-tuning : ajustement / réglage fin

Idée : Continuer l'entraînement d'un grand modèle de langage, par exemple pour effectuer une tâche spécifique :

- classification de texte,
- spécification de vocabulaire dans un domaine,
- synthèse de texte

Fine-tuning des LLM

Modèle de fondation

Modèles ajustés (« *fine-tunés* »)

Transforme du texte en requêtes SQL (base de données)

agent-conversationnel

Agent conversationnel spécialisé dans un domaine (connaît un vocabulaire spécifique)

Agent conversationnel très bon en synthèse de texte

Modèle de classification de phrases

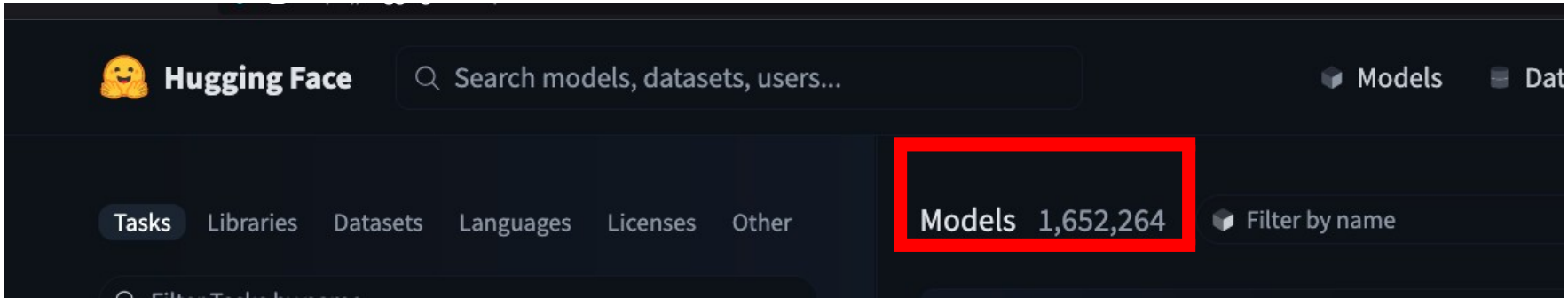
Fine-tuning des LLM

En pratique, plusieurs familles de modèles de fondation (avec plusieurs types de modèles : textes, audio, vision, multimodaux, …) :

- Llama,
- BERT,
- GPT,
- Qwen,
- DeepSeek,
- BLOOM

….

Et beaucoup de *fine-tuning* de ces modèles (https://huggingface.co/models):



Fine-tuning des LLM

Quelques ingrédients nécessaires :

- des données (**structurées**!),

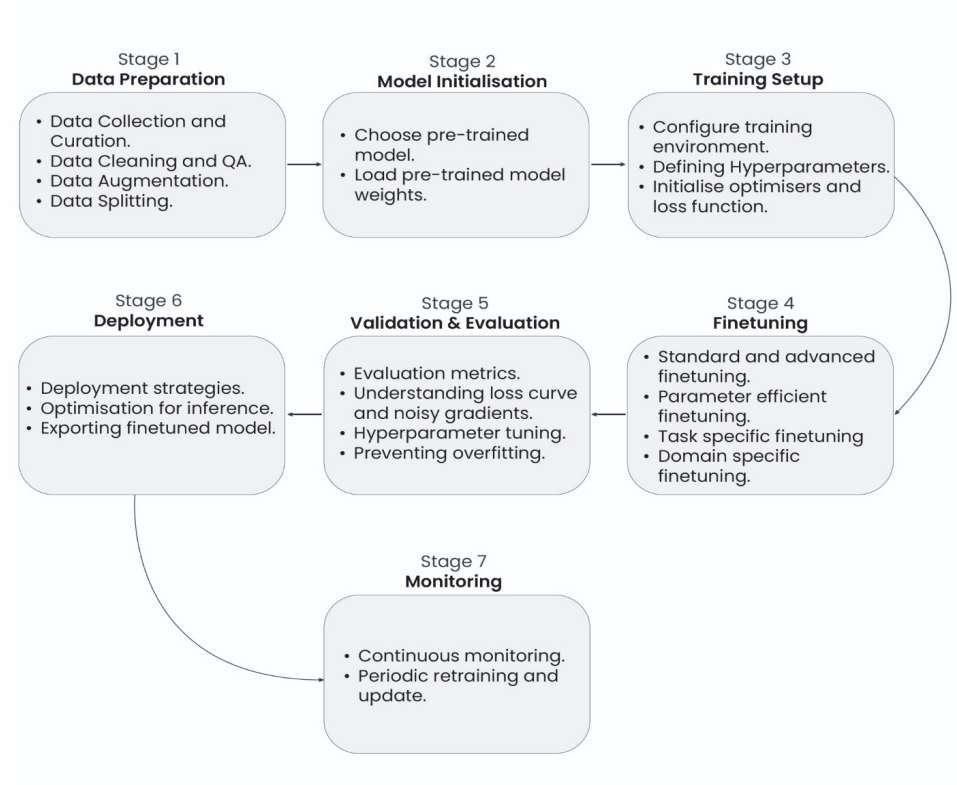
- des moyens de calculs (local ou distant),

- un modèle pré-entraîné,

- les outils permettant d'entraîner le modèle sur les données (bibliothèque Python, accès haut niveau, gestion des sessions d'entraînements, ...),

- une méthode d'entraînement (mise à jour de tous les paramètres, seulement certains, LoRA, ...)

Fine-tuning des LLM



Source : https://arxiv.org/pdf/2408.13296v1

Fine-tuning des LLM

les données :

- dépend de l'objectif de l'entraînement :
  - → résumé de textes : des listes de textes et leurs résumés,
  - → classification de sentiments : des phrases et des scores,
  - → agent conversationnel : des exemples de conversations

- jeux de données publics (*potentiellement sous licence*) :
  - HuggingFace Datasets

- données propres à votre domaine :
  - bases de données **bien indexées**

Dans tous les cas, les données peuvent être totalement ou partiellement générées par des LLMs

Fine-tuning des LLM

choisir l'infrastructure:

- **Local** : possible selon la / les machines à disposition (carte graphique / puces Apple Silicon préférables). Le fine-tuning peut ne pas demander beaucoup de ressources (~quelques heures d'entraînement sur un PC perso)

- **Distant** : (source https://towardsai.net)
    - sur des plateformes proposant des accès distants à des machines :
        - classiques : Google Cloud Platform, Microsoft Azure, ..
        - **plus locales** : IDRIS, Onyxia, ...
    - *via* une API de *fine-tuning* (ex : OpenAI)

Fine-tuning des LLM

choisir les outils :

- Avec une / plusieurs bibliothèques Python :
    - *HuggingFace Trainer*, et tous ses dérivés (beaucoup de modèles différents supportés)

    - Autres bibliothèques plus spécifiques, mais plus simples d'accès :
        - Llama CookBook,
        - Unsloth,
        - …
- *Via* un accès haut niveau (ex : API de fine-tuning OpenAI) :
    - Pas/ peu la main sur l'entraînement
    - Diffusion des modèles bloquée par le propriétaire de la plateforme

Fine-tuning des LLM

choisir les outils :

- plateforme d’entraînement (gestion des « runs », gestionnaire de workflow, analyse des performances du modèle entraîné)

- quelques exemples (local ou distant) :
    - AirFlow, Prefect, MLFlow, MetaFlow, …

Fine-tuning des LLM

choisir un modèle :

- le *fine-tuning* passe par la **modification des paramètres** du modèle. Impossible pour les modèles à tendance fermés, accessibles uniquement *via* **API –** à moins d'avoir accès à une API pour le *fine-tuning*
→ privilégier les modèles à **tendance ouverte**

- le choix du modèle peut être conditionné par les outils utilisés pour le *fine-tuning*.
→ ex : Llama CookBook n'accepte que les modèles Llama

- taille du modèle : incidence sur le temps d'entraînement (+de paramètres => +long).

Fine-tuning des LLM

choisir des méthodes de fine-tuning (complémentaires!):

- Supervised Fine-Tuning,
- LoRA,
- PEFT,
- Reinforcement Learning with Human Feedback (RLHF),
- DPO,
- QLoRA,
- ...

→ diffèrent selon les cas d'usages, et sont plus ou moins complexes à mettre en place

Fine-tuning des LLM

Exemple / expérimentation : Intégrer des acronymes et leurs définitions dans un agent conversationnel

0 – Avoir une liste **structurée** d'acronymes et leurs définitions

1 – Générer des conversations (avec un LLM par exemple) sur ces acronymes

2 – Continuer l'entraînement d'un agent conversationnel sur ces données

- choisir un modèle
- choix de l'infrastructure d'entraînement (local, distant)
- méthode et outils pour le *fine-tuning*

3 – Tester le modèle

4 – Partager, utiliser le modèle *fine-tuné*

Exemple / expérimentation : Mémorisation d'acronymes

0 – Avoir une liste **structurée** d'acronymes et leurs définitions

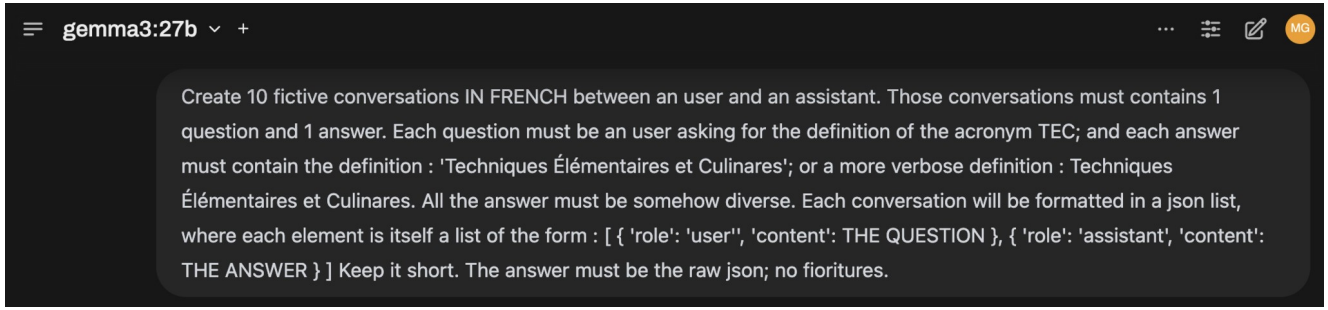
```
    },
    {
        "acronym": "PER",
        "definition": "Purée et Épice de la Réussite"
    },
    {
        "acronym": "FUN",
        "definition": "Fusion Universelle de la Nourriture"
    },
    {
        "acronym": "TEC",
        "definition": "Techniques Élémentaires et Culinares"
    },
    {
        "acronym": "SAL",
        "definition": "Savoir-Faire et Aromes de la Légèreté"
    },
    {
        "acronym": "CAMP",
        "definition": "Création Artisanale et Maîtrisée de la Pâtisserie"
    },
```

```
    <data skos:Concept="http://www.my.com/#institute" uri="http://www.my.com/#kisti">
        <skos:prefLabel>kisti</skos:prefLabel>
        <skos:altLabel>Korean Institute for Science and Technology Information</skos:altLabel>
        <data uri="http://www.my.com/#korea"/>
        <skos:related rdf:resource="http://www.my.com/#research"/>
        <skos:definition>KISTI is a government-funded research institute designed to maximize the efficiency of science and t
        <link skos:Concept="http://www.my.com/#url" value="http://en.kisti.re.kr"/>
    </data>

<data skos:Concept="http://www.my.com/#laboratory" uri="http://www.my.com/#BerkeleyLab">
        <skos:prefLabel>Lawrence Berkeley National Laboratory</skos:prefLabel>
        <data uri="http://www.my.com/#USA"/>
        <data uri="http://www.my.com/#AnshuDubey"/>
        <skos:related rdf:resource="http://www.my.com/#research"/>
        <skos:definition>In the world of science, Lawrence Berkeley National Laboratory (Berkeley Lab) is synonymous with "ex
        <link skos:Concept="http://www.my.com/#url" value="http://www.lbl.gov"/>@
    </data>
    <data skos:Concept="http://www.my.com/#supercomputer" uri="http://www.my.com/#Yellowstone">
        <data uri="http://www.my.com/#USA"/>
        <data uri="http://www.my.com/#NCAR"/>
        <skos:definition>Yellowstone is NCAR's 1.5-petaflops high-performance IBM iDataPlex cluster, which features 72,576 Ir
        <link skos:Concept="http://www.my.com/#url" value="https://www2.cisl.ucar.edu/resources/yellowstone"/>
    </data>
```

Exemple / expérimentation : Mémorisation d'acronymes

1 – Générer des conversations (avec un LLM par exemple) sur ces acronymes




```
{
    "role": "user",
    "content": "Bonjour, pouvez-vous me dire ce que signifie l'acronyme
    TEC ?"
},
{
    "role": "assistant",
    "content": "TEC signifie Techniques \u00c9l\u00e9mentaires et
    Culinaire."
}
```

Exemple / expérimentation : Mémorisation d’acronymes

2 - Entraînement

Mini-cours sur l’entraînement et le fonctionnement des réseaux de neurones

https://www.figma.com/design/bdHDd6JYnM5Dfsoi8lhBwP/Untitled?m=auto&t=LQAaI7R3

Mini-cours sur l’entraînement et le fonctionnement des réseaux de neurones

Résumé du vocabulaire :

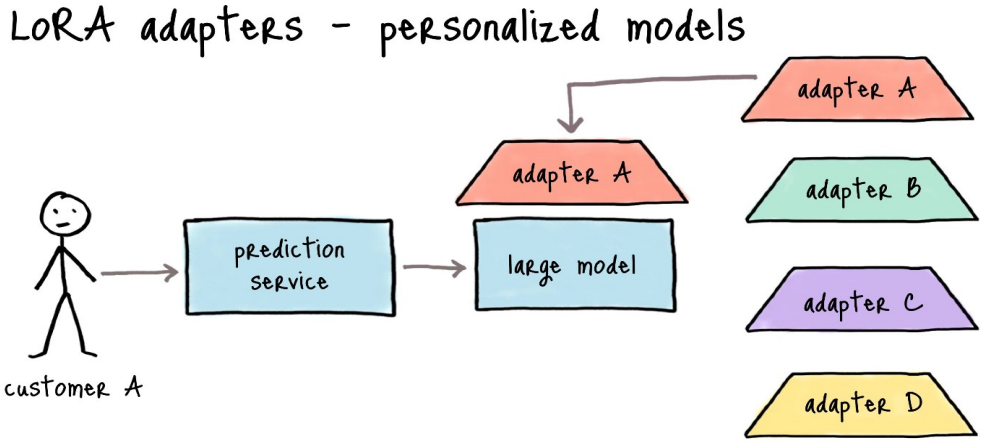
- learning-rate : « pas » du gradient, mesure la vitesse d’apprentissage

- n_epochs : nombre de fois où le modèle à « vu » toute les données d’entraînement

- mini-batch-size (per_device_train_batch_size dans « transformers »): nombre d’éléments du jeu de données utilisés dans 1 mise à jour des poids du modèle

LoRA

Zoom sur LoRA (Low-Rank Adaptation) **:**

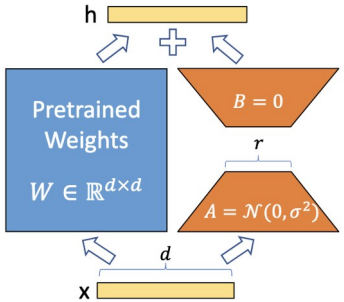


Source : https://huggingface.co/learn/llm-course/en/chapter11/4

Idée : On **rajoute** des **paramètres,** que l'on entraîne avec les données. Le nombre de paramètres rajoutés est bien **plus faible** (par ex. 5%) que le nombre de paramètres du modèle.

- entraînement plus rapide ; et donne des résultats *presque* aussi bons,
- sépare le modèle de base (**lourd)** de la partie *fine-tuning (**légère**),*
- permet d'utiliser plusieurs *fine-tuning* différents sur un même modèle.

LoRA



Source: https://arxiv.org/pdf/2106.0968

Couche d'un modèle pré-entraîné :
h = W.x (W : matrice avec d lignes et d colonnes)

Fine-Tuning « classique » :

h = W'.x (on modifie W en W' pendant le fine-tuning)

h = W.x + ΔW.x (ΔW = W' - W)
→ ΔW et W ont la même taille (d lignes, d colonnes ! → d^2 paramètres mis à jours. SI d = 3072, cela fait : 9 437 184)

Fine-tuning LoRA :

h = W.x + B.A.x

→ A et B ont des dimensions plus petites que W
→ On choisit A de taille (d, r) et B de taille (r, d).
(A a d lignes et r colonnes, B a r lignes et d colonnes)
→ les coefficients de A et B seront les seuls modifiés pendant l'entraînement
→ Typiquement, on fixe r = 16.
→ 2.d.r paramètres : (si d = 3072, cela fait : 98 304)

Exemple / expérimentation : Mémorisation d'acronymes

2 – Continuer l'entraînement d'un agent conversationnel sur ces données

choisir un modèle

Llama 3.2-1B Instruct

- petit modèle, donc rapide à entraîner

- à tendance ouverte, possible de le *fine-tuner* localement

- ! licence de diffusion **restrictive**

choisir une infrastructure

- local (Mac M3) : pour expérimenter
- distant (GCP / Onyxia) : entraînement plus important **(mais pas nécessaire ici – facteur x3 temps)**

outils d'entraînement

Hugging Face Trainer :
- gestion fine des hyper-paramètres,
- bien documenté
- prise en main très simple
- gratuit + open-source

Metaflow, MLFlow (gestionnaires de workflow)

méthode de *fine-tuning*

- LoRA : **léger** et **modulable**

Exemple / expérimentation : Mémorisation d’acronymes

3 – Tester le modèle

On veut tester le modèle sur sa capacité à **mémoriser des définitions.**

On pose au modèle les questions du jeu de test, et on évalue ensuite la qualité de ses réponses.

2 exemples pour évaluer la qualité des réponses :

- modèles d’« embeddings » : petits modèles, qui sont entraîné à prédire le taux de similarité entre deux séquences de mots

- « LLM as a judge » : on demande à un LLM tiers d’évaluer la qualité des réponses du LLM « fine-tuné ». Par exemple : « Réponds 1 si les deux réponses sont proches, et 0 sinon ».

Exemple / expérimentation : Mémorisation d’acronymes

3 – Tester le modèle

Selon les résultats du modèles, recommencer l’entraînement avec des paramètres différents .. !

Exemple / expérimentation : Mémorisation d'acronymes

4 – Partager et utiliser le modèle

Quelques exemples :

- publier sur Hugging Face Hub
  - → type d'accès à décider : accès ouvert, accès restreint
  - → licence à décider

- partager le modèle brut :
  - → poids : .safetensors (~pkl) ou autres formats quantisés
  - → directement les checkpoints de l'entraînement pour l'experimentation

- dans le cas d'un modèle fine-tuné avec la méthode LoRA on peut se contenter de partager la surcouche (+ légère que le modèle entier)

Exemple / expérimentation : Mémorisation d’acronymes

Pour expérimenter :

→ code disponible sur github :

https://github.com/mariusgarenaux/fine_tuning_acronym

FOLLOW US!

www.irisa.fr

irisa-lab

@irisa_lab

IRISA

1975 2025