

Nouveau Sessions garanties disponibles

Formation Fine-tuning des LLM : techniques, outils et bonnes pratiques

Réf. SF34428

Adaptez un modèle de langage open source à votre métier avec LoRA et QLoRA, et prouvez le gain obtenu par une évaluation rigoureuse.

2 j Durée	1 600 € HT Inter / participant	4 200 € HT Intra (≤12 pers.)	12 Sessions à venir
--------------	-----------------------------------	---------------------------------	------------------------

Objectifs pédagogiques

- ✓ Choisir entre prompt engineering, RAG et fine-tuning selon le cas d'usage, le coût et les contraintes de confidentialité
- ✓ Produire un jeu de données d'entraînement métier de qualité, du choix du format à la prévention des fuites de données
- ✓ Implémenter un fine-tuning supervisé avec LoRA et QLoRA dans l'écosystème Hugging Face
- ✓ Optimiser l'empreinte mémoire et le coût d'un entraînement par la quantification et le réglage des hyperparamètres
- ✓ Évaluer un modèle fine-tuné face au modèle de base avec un protocole outillé (jeu de tests métier, LLM juge, tests de non-régression)
- ✓ Implémenter un alignement par préférences (DPO) pour ajuster le ton, le format ou la conformité des réponses

Public visé & prérequis

Public visé Data scientists, Ingénieurs Big Data

Prérequis- Pratique régulière de Python - Bases du machine learning et du deep learning (entraînement, fonction de perte, surapprentissage) - Expérience des LLM, via API ou en local - Notions de PyTorch appréciées

Programme détaillé

Cadrer un projet de fine-tuning

- Prompt engineering, RAG ou fine-tuning : critères de choix selon la tâche, le coût, la latence, la qualité attendue et la confidentialité des données
- Apports et limites du fine-tuning : adapter le format, le ton, le comportement ou une tâche précise, sans en attendre l'ajout de connaissances factuelles à jour
- Panorama des approches : full fine-tuning, méthodes PEFT, fine-tuning managé proposé par les fournisseurs d'API
- Choix du modèle de base : familles de modèles ouverts, taille, licence et contraintes d'hébergement visées
- Estimation des ressources : mémoire GPU, durée d'entraînement, arbitrage entre cloud et infrastructure interne
- Cadre juridique et éthique : licences des modèles ouverts, propriété du modèle affiné, RGPD appliqué aux données d'entraînement

Activités pratiques :

- Analyse de plusieurs cas d'usage métier et choix argumenté entre prompt, RAG et fine-tuning
- Mesure d'un point de référence : performances d'un modèle de base en zero-shot et few-shot sur la tâche du fil rouge

Préparer les données d'entraînement

- Formats de données : instruction / réponse, conversations multi-tours, paires de préférences, et rôle des chat templates
- Sources exploitables : FAQ, tickets, documents internes et bases métier transformés en exemples d'entraînement
- Curation : nettoyage, déduplication, filtrage qualité, équilibrage des cas et traitement des exemples contradictoires
- Données synthétiques : génération d'exemples par un LLM, diversification et contrôle des biais introduits
- Découpage entraînement / validation / test : prévention des fuites de données entre jeux
- Dimensionnement : volume d'exemples nécessaire selon la tâche et rendements décroissants
- Outillage : bibliothèque Datasets de Hugging Face, publication et versioning des jeux de données sur le Hub

Activités pratiques :

- Construction d'un jeu de données instruction / réponse à partir de données métier brutes
- Nettoyage, déduplication et découpage du jeu, puis publication sur le Hub

Adaptation efficace : PEFT, LoRA, QLoRA et quantification

- Full fine-tuning ou PEFT : impact sur la mémoire, la vitesse d'entraînement et la qualité obtenue
- LoRA : principe des matrices de rang faible, choix du rang et du facteur alpha, sélection des couches ciblées
- Quantification des poids : représentations 16, 8 et 4 bits, format NF4, double quantification et compromis précision / mémoire
- QLoRA : entraînement d'adaptateurs LoRA sur un modèle quantifié en 4 bits, configuration avec bitsandbytes et optimiseurs paginés
- Leviers mémoire complémentaires : précision mixte, gradient checkpointing, accumulation de gradient
- Bibliothèques d'accélération (Unsloth...) : gains de vitesse et de mémoire pour entraîner sur un GPU unique
- Choix d'une stratégie : arbitrage entre qualité, temps d'entraînement et budget matériel

Activités pratiques :

- Chargement d'un même modèle en 16, 8 puis 4 bits et mesure de l'empreinte mémoire GPU
- Comparaison d'une configuration LoRA légère et d'une configuration plus ambitieuse (rang, couches ciblées)
- Configuration complète d'un entraînement QLoRA sur un modèle de quelques milliards de paramètres

Entraîner et piloter un fine-tuning supervisé

- Fine-tuning supervisé (SFT) avec Transformers, PEFT et TRL : chaîne d'entraînement de référence de l'écosystème Hugging Face
- Outils de configuration déclarative (Axolotl, LLaMA-Factory) : alternatives pour industrialiser les entraînements
- Hyperparamètres clés : taux d'apprentissage, nombre d'époques, taille de batch, warmup, et stratégies de réglage
- Lecture des courbes de perte : diagnostic du surapprentissage et du sous-apprentissage, itérations courtes
- Checkpoints et reprise d'entraînement : sauvegarde des adaptateurs et sélection du meilleur point
- Suivi des expériences : traçabilité des runs avec Weights & Biases ou MLflow
- Passage à l'échelle : principes de l'entraînement multi-GPU (FSDP, DeepSpeed) et recours au cloud

Activités pratiques :

- Fine-tuning QLoRA d'un modèle ouvert sur le jeu de données du fil rouge, avec suivi des courbes en temps réel
- Ajustement des hyperparamètres après lecture des courbes et comparaison de deux runs

Atelier : évaluer un modèle fine-tuné

- Métriques automatiques : perplexité, exact match, F1, ROUGE, et limites de ces métriques pour la génération
- Jeu de tests métier : construction d'un échantillon représentatif des usages réels et scoring automatique
- LLM juge : grille d'évaluation, biais connus et calibration par une annotation humaine sur un échantillon
- Comparaison systématique : modèle de base, modèle fine-tuné et approche par prompt seul sur les mêmes cas de test
- Régressions : vérification des capacités générales du modèle et détection de l'oubli catastrophique
- Outils d'évaluation open source : harnais de benchmarks et frameworks d'évaluation pour rendre le protocole reproductible
- Analyse d'erreurs : identification des cas d'échec et décision d'itérer sur les données ou sur l'entraînement

Activités pratiques :

- Rechargement du modèle de base avec l'adaptateur entraîné et premier comparatif chiffré avant / après
- Mise en place d'une évaluation par LLM juge et confrontation avec les métriques automatiques
- Tests de non-régression sur des tâches générales et analyse des erreurs du modèle adapté

Aligner un modèle sur des préférences métier

- RLHF : principe, rôle du modèle de récompense et raisons de son coût de mise en œuvre
- Optimisation directe des préférences (DPO) : alternative simplifiée au RLHF et mise en œuvre avec TRL
- Variantes récentes de l'optimisation par préférences : panorama et cas d'usage
- Jeu de préférences : construction de paires réponse préférée / rejetée, annotation et contrôle qualité
- Critères d'alignement métier : ton, format, conformité, refus appropriés

Activités pratiques :

- Constitution d'un petit jeu de préférences sur un critère métier (ton ou conformité)
- Alignement DPO du modèle fine-tuné et comparaison avant / après avec le protocole d'évaluation de l'atelier

Livrer et faire vivre le modèle adapté

- Fusion d'un adaptateur LoRA dans son modèle de base : quand fusionner et quand conserver l'adaptateur séparé
- Formats d'export : safetensors, GGUF, et principe de la quantification post-entraînement selon la cible
- Mise à disposition du modèle : panorama des options de serving et intégration à une application existante
- Versioning et traçabilité : lien entre jeu de données, configuration d'entraînement et version du modèle
- Cycle de vie : dérive des données, réentraînement planifié et tests de non-régression
- Bonnes pratiques et dépannage : erreurs fréquentes (mémoire, chat template, données) et pistes de résolution

Activités pratiques :

- Fusion et export du modèle adapté, puis test de chargement et d'inférence
- Rédaction d'une fiche de traçabilité du modèle (données, configuration, résultats d'évaluation)

Prochaines sessions

DATE	LIEU	STATUT
03/12/2026	à distance	Session garantie
25/02/2027	Paris	Sur confirmation
25/02/2027	Toulouse	Sur confirmation
25/02/2027	à distance	Sur confirmation
02/06/2027	Aix-en-Provence	Sur confirmation
02/06/2027	Paris	Sur confirmation

Pourquoi choisir Sparks Formation ?

20 ans d'expertise IT, des formateurs experts et des sessions garanties contre l'annulation.

20 ans
d'expertise

1 068+
formations

11 178
apprenants formés/an

+67
Net Promoter Score

Avis des participants

4,9/5 — 21 avis

Anthony G. — Formation très bien organisée, très claire Alison explique très clairement et est très à l'écoute, avec toujours une approche pratique essentielle. je n'avais jamais fait de Python et...

François H. — François-Marie est un excellent formateur ! Un grand merci à lui d'avoir été aussi pédagogue et clair dans ses explications du début à la fin de la formation,...

Votre formateur

Fridy H. — Formateur Data Science & Machine Learning

Fridy intervient sur des formations portant sur la Data Science et le Machine Learning. Parmi les formations qui lui sont confiées figurent notamment « Machine Learning » et « Fine-tuning ».

Ils nous font confiance

Cette année notre filière Data Science & Machine Learning a accompagné plus de 70 entreprises dont :

SOPRA STERIA RTE AMADEUS SAS CEGEDIM LBC FRANCE SNCF - E.SNCF SOLUTIONS AMADEUS DATA PROCESSING GMBH CSTB GERS SA
ADENTIS +60 autres

Financement & certifications

Cette formation est éligible aux financements OPCO et FNE-Formation. Nos équipes vous accompagnent dans le montage de votre dossier.



Réservez votre place dès maintenant

20 ans d'expertise · 1 000+ formations · NPS +67 · réponse sous 24h ouvrées

0 805 950 800

demande@sparks-formation.com