

Nouveau

Formation Mistral : spécialiser et auto-héberger un LLM open source

Réf. SF34734

Adaptez un modèle open source Mistral à vos données métier et faites-le tourner sur votre propre infrastructure : un projet de bout en bout, mesuré, sécurisé et chiffré.

3 j
Durée

2 400 € HT
Inter / participant

0
Sessions à venir

Objectifs pédagogiques

- ✓ Choisir un modèle open-weight Mistral adapté au cas d'usage, à sa licence et aux contraintes de souveraineté de l'entreprise
- ✓ Évaluer si la spécialisation est le bon levier face à l'amélioration du prompt et au RAG, à partir d'un coût complet chiffré
- ✓ Produire des jeux d'entraînement et d'évaluation au format conversationnel attendu, disjoints et sans données personnelles
- ✓ Implémenter la spécialisation d'un modèle Mistral par LoRA ou QLoRA et dimensionner les ressources GPU nécessaires
- ✓ Comparer le modèle spécialisé à son modèle de base et détecter les régressions sur les tâches générales
- ✓ Déployer le modèle en auto-hébergement avec vLLM derrière une API compatible OpenAI, quantification et dimensionnement de l'infrastructure inclus
- ✓ Concevoir l'exploitation sécurisée du service : contrôle d'accès, supervision, cycle de vie et seuil de rentabilité face à une API managée

Public visé & prérequis

Public visé Architectes techniques, Data scientists, Développeurs

Prérequis- Pratique autonome de Python pour écrire et déboguer un traitement de données - Notions de machine learning : jeux d'entraînement et de test, sur-apprentissage, métriques d'évaluation - Pratique de la ligne de commande Linux et des conteneurs Docker - Bases des API REST ; une première utilisation d'un LLM par API est un plus (aucune expérience de l'entraînement de LLM n'est requise)

Programme détaillé

Modèles open-weight Mistral, licences et décision de spécialiser

- Familles de modèles open-weight Mistral : généralistes, code, raisonnement et petits modèles pour l'embarqué, face aux modèles commerciaux non publiés en poids ouverts
- Licences d'usage : licence permissive Apache 2.0, licences restrictives (recherche, non-production) et obligations associées pour un usage en entreprise
- Pourquoi auto-héberger : souveraineté et confidentialité des données, contraintes réglementaires (RGPD, AI Act), économies d'échelle
- Trois leviers d'amélioration : prompt, RAG ou spécialisation — ce que la spécialisation sait faire (format, style, tâche étroite) et ce qu'elle ne sait pas faire (ajouter des connaissances)
- Coût complet d'un projet de spécialisation : entraînement, hébergement, maintenance et réentraînement
- Choisir le modèle de départ : taille, langues, tâche visée, licence et empreinte matérielle

Activités pratiques :

- Comparaison des trois leviers sur le cas d'usage fil rouge et rédaction d'une note d'arbitrage d'une page
- Sélection argumentée d'un modèle de départ à l'aide d'une grille licence / taille / matériel

Jeux de données d'entraînement et d'évaluation

- Volume, diversité et équilibre : dimensionner un jeu de données pour une tâche étroite
- Format conversationnel attendu : messages system / user / assistant en JSONL, gabarit de chat et tokenizer du modèle Mistral
- Conversion depuis les données métier : passer de tickets, documents ou historiques à des exemples d'entraînement exploitables

- Nettoyage et déduplication : la qualité des exemples avant leur quantité
- Jeu d'évaluation constitué avant l'entraînement : séparation stricte et détection des fuites entre les deux jeux
- Données personnelles : ce qui est proscrit dans un jeu d'entraînement, anonymisation et traçabilité des sources

Activités pratiques :

- Conversion d'un extrait de données métier au format conversationnel et vérification du gabarit de chat
- Production d'un couple de jeux entraînement / évaluation documenté, sans fuite ni données personnelles

Spécialiser un modèle Mistral avec LoRA et QLoRA

- Adaptateurs de faible rang (LoRA) : principe, rang, alpha et modules ciblés
- QLoRA : entraîner sur un modèle quantifié pour tenir sur un seul GPU
- Outillage d'entraînement : outillage de référence de l'éditeur pour ses modèles open-weight, écosystème Hugging Face (Transformers, PEFT, TRL) et alternatives optimisées
- Dimensionner l'entraînement : mémoire GPU, durée et coût d'une session
- Hyperparamètres : ceux qui comptent (taux d'apprentissage, nombre d'époques, rang) et ceux à laisser par défaut
- Lire les courbes d'entraînement : repérer sous-apprentissage, sur-apprentissage et plateau
- Versionner ensemble modèle de base, adaptateur, données et expériences
- Voie managée de l'éditeur ou voie autonome : coût, maîtrise, réversibilité et souveraineté

Activités pratiques :

- Entraînement d'un adaptateur QLoRA sur le cas fil rouge, avec mesure de la mémoire GPU et de la durée
- Variation du rang de l'adaptateur et comparaison des résultats obtenus
- Sur-apprentissage provoqué volontairement pour apprendre à le reconnaître sur les courbes

Évaluer le modèle spécialisé et décider

- Comparaison au modèle de base : la seule comparaison valable, menée tâche par tâche
- Métriques de la tâche : exactitude, respect du format attendu, métriques automatiques adaptées au cas d'usage
- Évaluation par LLM juge et revue humaine : complémentarité, biais et coût de chaque approche
- Oubli catastrophique : mesurer la régression du modèle spécialisé sur les tâches générales
- Décision de mise en production : savoir écarter un modèle qui n'apporte rien
- Jeu de non-régression : capitaliser l'évaluation pour les versions suivantes du modèle

Activités pratiques :

- Évaluation comparée du modèle de base et du modèle spécialisé sur le jeu d'évaluation
- Mesure de la régression sur un jeu de tâches générales
- Rédaction d'une décision argumentée de retenir ou non le modèle spécialisé

Auto-héberger le modèle : quantifier, servir, dimensionner

- Choisir un moteur d'inférence : vLLM comme référence pour le service multi-utilisateurs, critères de choix des alternatives, llama.cpp et format GGUF pour les petits modèles et l'embarqué
- Fusionner l'adaptateur ou servir plusieurs adaptateurs sur un même modèle de base : compromis de performance et de gestion
- Quantification (AWQ, GPTQ, FP8, GGUF) : gain d'empreinte mémoire, perte de qualité et méthode de mesure
- API compatible OpenAI : brancher le modèle auto-hébergé sur les applications et frameworks existants sans réécriture
- Performance du service : continuous batching, PagedAttention et gestion du cache KV
- Dimensionner l'infrastructure : VRAM (poids et cache KV), débit, concurrence, latence du premier jeton, et seuil de passage au multi-GPU
- Conteneuriser le service pour le déployer sur ses propres serveurs ou sur un cloud de confiance

Activités pratiques :

- Mise en service du modèle spécialisé avec vLLM et appel depuis une application via l'API compatible OpenAI
- Quantification du modèle et rejeu du jeu d'évaluation pour mesurer la perte de qualité
- Montée en charge jusqu'à la rupture et rédaction d'une fiche de dimensionnement matériel

Sécuriser et exploiter dans la durée

- Contrôle d'accès : authentification, passerelle d'API et gestion des droits par application ou par équipe
- Isolation réseau et environnements déconnectés : exploiter un modèle sans accès sortant
- Protection des entrées et des sorties : injection de prompt, jailbreak, filtrage par guardrails et données personnelles
- Supervision : métriques du moteur, qualité des réponses, dérive et gestion des incidents
- Coût au jeton : auto-hébergement contre API managée et calcul du seuil de bascule

- Cycle de vie : réentraînement, adoption d'une nouvelle version du modèle de base et tests de non-régression
- Réversibilité : pouvoir revenir à une API managée ou changer de modèle sans tout reconstruire

Activités pratiques :

- Mise en place d'une authentification et d'un tableau de bord de supervision devant le service d'inférence
- Établissement du modèle de coût complet et du seuil de rentabilité face à une API managée
- Détection d'une régression après réentraînement grâce au jeu de non-régression

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 — 65 avis

Rachelle Y. — Merci c'était super :)

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...

Ils nous font confiance

Cette année notre filière Intelligence artificielle & Data Science a accompagné plus de 250 entreprises dont :

SOPRA STERIA CEGEDIM RTE AKANEA AGRO SOFTWARE AKANEA DEVELOPPEMENT AMADEUS SAS GROUPE LDLC LBC FRANCE PROXIAD AXE SEINE SNCF - E.SNCF SOLUTIONS +240 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