

PROGRAMME DE FORMATION

ADAPTER UN MODÈLE IA À SON DOMAINE

FICHE SYNTHÉTIQUE DE LA FORMATION

Intitulé	Adapter un modèle IA à son domaine — Transformers, fine-tuning et LoRA
Référence	IA-FINETUNE-2J
Durée	2 jours (14 heures)
Modalités	Présentiel ou distanciel synchrone (classe virtuelle)
Tarif	1 400 € HT / participant (inter) — intra sur devis
Délais d'accès	Inscription possible jusqu'à 14 jours avant le début de la session
Lieu	Paris 17e ou distanciel via Zoom
Effectif	4 participants maximum par session
Contact	contact@pythonia.fr – 06 66 06 38 59
Accessibilité PSH	Formation accessible aux PSH. Contactez notre référent handicap pour toute adaptation.
Dernière mise à jour	Octobre 2026

1. Public visé

Cette formation s'adresse aux data scientists, ingénieurs machine learning, développeurs et équipes R&D; qui souhaitent spécialiser un modèle de langage open source sur leur vocabulaire, leurs documents et leurs tâches métier (classification, extraction, rédaction, assistance), en gardant la propriété du modèle obtenu.

2. Prérequis

- Bonne maîtrise de Python et des bibliothèques de données (NumPy, Pandas)
- Notions de machine learning : jeu d'entraînement et de test, sur-apprentissage, métriques
- Avoir déjà utilisé un assistant IA générative
- Accès à un GPU NVIDIA (8 Go de VRAM minimum) ou à un environnement cloud GPU fourni par Pythonia pendant la formation

Un test de positionnement est proposé avant l'entrée en formation afin de vérifier l'adéquation du parcours avec le profil du participant et la compatibilité de son poste.

3. Objectifs pédagogiques

À l'issue de cette formation, le stagiaire sera capable de :

- Expliquer l'architecture Transformer, le mécanisme d'attention et la différence entre pré-entraînement et adaptation
- Choisir la bonne stratégie : prompt, RAG, fine-tuning complet ou efficace (LoRA, QLoRA)
- Préparer un dataset d'entraînement de qualité à partir de données métier
- Fine-tuner un modèle open source avec LoRA / QLoRA grâce à l'écosystème Hugging Face
- Évaluer le modèle spécialisé, le comparer au modèle de base et le déployer en local

4. Programme détaillé

Fil rouge : les participants spécialisent un modèle open source sur un cas métier réel, depuis la préparation des données jusqu'à son exécution en local avec Ollama.

JOUR 1 — Comprendre les Transformers et préparer ses données (7h)

Matin (3h30) — Les Transformers de l'intérieur

Architecture

- Tokenisation : BPE, vocabulaire, impact sur le français et les termes métier
- Embeddings, attention, multi-head attention, couches feed-forward
- Encodeurs (BERT), décodeurs (GPT, Llama, Mistral), encodeurs-décodeurs (T5)

Cycle de vie d'un modèle

- Pré-entraînement, instruction tuning, alignement (RLHF, DPO)
- Modèles open-weight et licences : Mistral, Llama, Qwen, Gemma, CamemBERT
- L'écosystème Hugging Face : Hub, transformers, datasets, PEFT, TRL

Choisir sa stratégie

- Prompt engineering, RAG ou fine-tuning : arbre de décision
- Fine-tuning complet vs efficace en paramètres (PEFT)
- Estimation des besoins : données, GPU, temps, coûts

Travaux pratiques :

→ TP1 : explorer un Transformer avec Hugging Face (tokenisation, attention, génération)

Après-midi (3h30) — Préparer le dataset

Construire ses données

- Formats : classification, instruction / réponse, conversation (chat template)
- Sources internes : documents, tickets, e-mails, bases de connaissances
- Génération de données synthétiques avec un LLM local et contrôle qualité

Qualité et conformité

- Nettoyage, déduplication, équilibrage des classes
- Anonymisation des données personnelles (RGPD)
- Découpage entraînement / validation / test et jeu d'évaluation métier

Travaux pratiques :

→ TP2 : construire et publier en interne un dataset d'instructions à partir de documents métier

✓ Évaluation formative : quiz de 10 questions et revue collective des TP en fin de journée

JOUR 2 — Fine-tuning, évaluation et spécialisation (7h)

Matin (3h30) — Fine-tuning efficace

Fine-tuning d'un modèle encodeur

- Classification de textes métier avec CamemBERT ou ModernBERT
- Trainer Hugging Face : hyperparamètres, suivi des métriques

LoRA et QLoRA

- Principe des adaptateurs de rang faible, choix du rang et des couches
- Quantification 4 bits (bitsandbytes) pour entraîner sur un seul GPU
- Supervised fine-tuning d'un LLM avec TRL (SFTTrainer) ou Unsloth
- Suivi de l'entraînement : courbes de perte, sur-apprentissage, early stopping

Travaux pratiques :

- TP3 : fine-tuner un classifieur de documents métier
- TP4 : adapter un LLM open source avec QLoRA sur le dataset du jour 1

Après-midi (3h30) — Évaluer, spécialiser et déployer

Évaluer le modèle

- Métriques automatiques : exactitude, F1, perplexité, ROUGE
- Évaluation métier : jeu de questions de référence et comparaison avec le modèle de base
- LLM-as-a-judge local et revue humaine
- Risques : oubli catastrophique, biais, hallucinations

Spécialiser et déployer

- Fusion des adaptateurs LoRA, conversion au format GGUF et quantification
- Exécution du modèle spécialisé en local avec Ollama
- Combiner fine-tuning et RAG ; versionner et maintenir ses modèles

Travaux pratiques :

- TP5 : évaluer le modèle adapté face au modèle de base
- TP6 : exporter le modèle en GGUF et l'utiliser avec Ollama

✓ **Évaluation sommative : QCM final de 20 questions et présentation du modèle adapté — critère de réussite : 60 %**

5. Méthodes et moyens pédagogiques

Méthodes mobilisées

- Apports théoriques : exposés interactifs avec diaporama et démonstrations en direct
- Mise en pratique : TP progressifs sur un cas métier réel, du dataset au modèle déployé
- Pédagogie active : cas d'usage apportés par les participants
- Alternance théorie/pratique : ratio 30 % théorie / 70 % pratique

Moyens techniques

- Salle de formation équipée : vidéoprojecteur, paperboard, connexion Wi-Fi haut débit
- Poste informatique par stagiaire (ou poste personnel répondant aux prérequis)
- En distanciel : Zoom avec partage d'écran, tableau blanc collaboratif
- Environnement GPU dans le cloud fourni pendant la formation si nécessaire
- Supports de cours, notebooks et code source complet remis aux stagiaires

6. Modalités d'évaluation

Positionnement à l'entrée

Avant l'entrée en formation, chaque participant complète un test de positionnement évaluant son niveau en Python, ses connaissances en machine learning et ses cas d'usage.

Évaluations en cours de formation

- Évaluation formative : quiz de 10 questions en fin de jour 1

- Évaluation sommative : QCM final de 20 questions et présentation du modèle adapté en fin de jour 2
- Critère de réussite : obtenir 60 % au QCM final
- Attestation de fin de formation : remise à l'issue, mentionnant objectifs, durée et résultats

Satisfaction

- Questionnaire de satisfaction en fin de formation (formateur, contenu, logistique, atteinte des objectifs)

7. Accessibilité aux personnes en situation de handicap

Pythonia s'engage à rendre ses formations accessibles aux personnes en situation de handicap. Notre référent handicap est disponible pour étudier toute demande d'adaptation :

- Adaptation des supports pédagogiques (taille de police, contraste, format audio)
- Aménagement des conditions de passage (temps majoré, pauses adaptées)
- Orientation vers nos partenaires spécialisés : Agefiph, Cap Emploi, MDPH

Contact référent handicap : contact@pythonia.fr – 06 66 06 38 59

8. Engagement et suivi

- Suivi de l'assiduité : feuille d'émargement signée par demi-journée
- Prévention des abandons : suivi individualisé, adaptation du rythme, relance en cas d'absence
- Réclamations : toute réclamation peut être adressée par email à contact@pythonia.fr. Elle sera traitée sous 48h ouvrées.

PythonIA — Centre de Formation Python & Intelligence Artificielle

255 boulevard Pereire, 75017 Paris | contact@pythonia.fr | 06 66 06 38 59