

Mis à jour le 30/10/2026

S'inscrire

Formation DeepSeek DeepInfer

3 jours (21 heures)

Présentation

DeepInfer est le framework de DeepSeek dédié à l'inférence haute performance de ses modèles. Avec ses kernels optimisés et ses mécanismes de parallélisme, de génération structurée et de décodage spéculatif, il apporte les briques essentielles pour étudier et optimiser l'exécution de modèles à grande échelle.

Notre formation DeepSeek DeepInfer vous permettra de comprendre le fonctionnement du moteur et de prendre en main son environnement d'inférence hors ligne, en distinguant ses fonctionnalités des composants supplémentaires nécessaires à un service complet.

Vous apprendrez à préparer un environnement compatible, à exécuter un modèle pris en charge, à configurer les paramètres de génération et à produire des sorties structurées. Vous explorerez ensuite les stratégies de parallélisme et les principaux leviers d'optimisation.

À l'issue de la formation, vous serez en mesure de construire un protocole de benchmark reproductible, de comparer plusieurs configurations et d'interpréter les compromis entre débit, latence et consommation mémoire pour orienter vos choix techniques.

Cette formation aborde également les tests de non-régression et les exigences d'exploitation. Les ateliers vous aideront à identifier les limites du moteur, à documenter vos résultats et à préparer une architecture de service adaptée à vos usages.

Comme toutes nos formations, celle-ci vous présentera **la dernière version stable** de la technologie et ses nouveautés.

Objectifs

- Identifier le périmètre de DeepInfer et les composants nécessaires à un service d'inférence complet
- Configurer un environnement compatible et exécuter une inférence hors ligne sur une infrastructure adaptée
- Paramétrer la génération et valider des sorties structurées selon un schéma attendu
- Comparer les stratégies de parallélisme et les mécanismes d'accélération du moteur
- Mesurer les performances et vérifier la non-régression avec un protocole reproductible
- Définir les exigences techniques et opérationnelles d'une architecture de service autour du moteur

Public visé

- Ingénieur Machine Learning
- Ingénieur MLOps
- Développeur IA
- Architecte de solutions IA

Pré-requis

- Pratique de Python et de la gestion des environnements virtuels
- Connaissance des tenseurs et de l'exécution de modèles avec PyTorch en inférence
- Compréhension des modèles de langage, de la tokenisation et des principes de génération
- Aisance avec le terminal Linux et les opérations courantes de Git en ligne de commande

Pré-requis techniques

- Poste de travail recommandé avec 16 Go de RAM, processeur quatre cœurs et client SSH pour accéder à l'environnement distant
- Accès réservé avant la session à une infrastructure distante sous Linux équipée de GPU NVIDIA SM100, avec mémoire et stockage dimensionnés pour le modèle et les exercices
- Environnement distant avec Python 3.12 et GCC 13 ou supérieur disponibles pour la compilation
- Pile NVIDIA comprenant CUDA 13 et des pilotes compatibles, avec dépendances du dépôt installées dans un environnement isolé
- Connexion Internet stable autorisant les accès distants et les téléchargements nécessaires, avec Git disponible et les poids du modèle préparés avant les ateliers

Programme de notre formation DeepSeek DeepInfer

[Jour 1 - Matin]

Architecture et environnement d'exécution

- Périmètre de DeepInfer, prise en charge de DeepSeek-V4.1-Flash et distinction entre moteur hors ligne et service complet ([raw.githubusercontent.com](https://raw.githubusercontent.com/usercontent))

- Cycle d'inférence des modèles de langage : tokenisation, prefill, decode et mémoire des clés et valeurs
- Lecture de l'architecture Mixture of Experts et identification des contraintes de calcul, de mémoire et de communication
- Préparation de l'environnement Linux, contrôle des dépendances et repérage des scripts de compilation et des exemples
- Atelier pratique : Vérifier l'environnement distant, préparer un environnement isolé et contrôler la disponibilité du modèle et des accélérateurs.

[Jour 1 - Après-midi]

Premières inférences et sorties structurées

- Chargement du checkpoint depuis un répertoire local et exécution de l'exemple d'inférence hors ligne
- Lecture des objets de requête et de réponse, gestion des tokens et interprétation des journaux d'exécution
- Réglage des paramètres d'échantillonnage, des longueurs de génération et des conditions d'arrêt
- Génération contrainte par schéma JSON, expression régulière ou grammaire EBNF et validation des sorties (github.com)
- Atelier pratique : Adapter un exemple pour extraire des informations dans un format JSON imposé et vérifier les résultats sur un jeu de requêtes.

[Jour 2 - Matin]

Parallélisme et équilibrage des experts

- Répartition de l'attention avec Data Parallelism et Context Parallelism, puis des experts avec Expert Parallelism (github.com)
- Lecture des paramètres EP, DP et CP dans les exemples et vérification de leur cohérence
- Rôle de FlashMLA, DeepGEMM et DeepEP dans l'attention, le calcul matriciel et les communications
- Équilibrage des experts avec EPLB, experts redondants et distinction avec l'équilibrage des requêtes entre réplicas
- Atelier pratique : Comparer deux configurations de parallélisme compatibles avec le cluster pédagogique et analyser leurs journaux de charge et d'utilisation mémoire.

[Jour 2 - Après-midi]

Accélération et reproductibilité

- Principes de DSpark, proposition de tokens et vérification planifiée selon la confiance (raw.githubusercontent.com)
- Réglage de la longueur des propositions et interprétation du taux d'acceptation selon les requêtes
- Fonctionnement du SWA bounded replay et contraintes de conservation du suffixe de contexte

- Rôle de l'ordonnancement asynchrone dans le recouvrement du travail CPU et de l'exécution sur accélérateur
- Conditions de l'invariance au batch et de l'échantillonnage déterministe sur NVIDIA, avec graine fixée et configuration adaptée
- Atelier pratique : Comparer une exécution avec et sans décodage spéculatif, puis vérifier la reproductibilité d'une génération dans une configuration déterministe.

[Jour 3 - Matin]

Benchmarks et validation des résultats

- Construction d'un jeu de mesure représentatif : tailles de contexte, volumes de requêtes et longueurs des sorties
- Mesure séparée du prefill et du decode, échauffement et collecte hors surcharge du profileur (raw.githubusercontent.com)
- Lecture du débit par GPU et des pas par seconde, avec distinction entre comptage à acceptation complète et tokens réellement acceptés
- Analyse des compromis entre débit, latence, taille de batch et consommation mémoire sans extrapolation abusive au service complet
- Validation des sorties structurées, comparaison des références de perplexité et préparation des tests de non-régression
- Atelier pratique : Exécuter une campagne de benchmark, consigner les configurations et produire un tableau comparatif accompagné de contrôles de non-régression.

[Jour 3 - Après-midi]

Architecture de service et exigences d'exploitation

- Identification des composants à ajouter : système de cache KV, orchestration de la désagrégation prefill et decode, équilibrage entre réplicas
- Positionnement de deepseek-recipe pour la conversion des protocoles et des séquences de tokens, sans le confondre avec un serveur API (raw.githubusercontent.com)
- Conception de la couche de service : authentification, quotas, validation des entrées et protection des données
- Définition des besoins d'observabilité, des seuils de mémoire et de latence, des délais d'attente et des procédures de reprise
- Préparation d'un dossier de déploiement : versions figées, capacité, critères d'acceptation et limites connues
- Atelier pratique : Concevoir une architecture cible autour de DeepInfer et présenter un dossier de validation fondé sur les mesures et les tests réalisés.

Sociétés concernées

Cette formation s'adresse à la fois aux particuliers ainsi qu'aux entreprises, petites ou grandes, souhaitant former ses équipes à une nouvelle technologie informatique avancée ou bien à acquérir des connaissances métiers spécifiques ou des méthodes modernes.

Positionnement à l'entrée en formation

Le positionnement à l'entrée en formation respecte les critères qualité Qualiopi. Dès son inscription définitive, l'apprenant reçoit un questionnaire d'auto-évaluation nous permettant d'apprécier son niveau estimé sur différents types de technologies, ses attentes et objectifs personnels quant à la formation à venir, dans les limites imposées par le format sélectionné. Ce questionnaire nous permet également d'anticiper certaines difficultés de connexion ou de sécurité interne en entreprise (intraentreprise ou classe virtuelle) qui pourraient être problématiques pour le suivi et le bon déroulement de la session de formation.

Méthodes pédagogiques

Stage Pratique : 60% Pratique, 40% Théorie. Support de la formation distribué au format numérique à tous les participants.

Organisation

Le cours alterne les apports théoriques du formateur soutenus par des exemples et des séances de réflexions, et de travail en groupe.

Validation

À la fin de la session, un questionnaire à choix multiples permet de vérifier l'acquisition correcte des compétences.

Sanction

Une attestation sera remise à chaque stagiaire qui aura suivi la totalité de la formation.