

Mis à jour le 01/10/2026

S'inscrire

Formation PyTorch avancé

3 jours (21 heures)

Présentation

PyTorch fournit des outils avancés pour optimiser l'entraînement et l'inférence des modèles de Deep Learning, exploiter efficacement les GPU et distribuer les calculs sur des infrastructures adaptées aux modèles de grande taille.

Notre formation PyTorch avancé : optimisation et industrialisation vous permettra de passer d'un modèle fonctionnel à un entraînement performant et adapté aux contraintes de production.

Vous apprendrez à profiler les workloads CPU et GPU, analyser l'utilisation de la mémoire et identifier les goulets d'étranglement. Vous exploiterez torch.compile et TorchInductor pour accélérer vos modèles et diagnostiquer les graph breaks et recompilations.

Vous mettrez en œuvre le Mixed Precision Training et optimiserez vos pipelines de données avant de distribuer vos entraînements sur plusieurs GPU avec DDP et FSDP.

La formation couvre également la quantification, le pruning et l'optimisation de l'inférence. Vous apprendrez enfin à exporter vos modèles avec torch.export et ONNX et à intégrer leur suivi dans une démarche MLOps.

À l'issue de cette formation, vous serez capable de profiler, accélérer, distribuer et industrialiser des modèles PyTorch tout en maîtrisant les compromis entre performances, mémoire, précision et coûts.

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

Objectifs

- Profiler et diagnostiquer les performances CPU et GPU
- Accélérer les modèles avec torch.compile et TorchInductor
- Optimiser la mémoire, les DataLoaders et l'entraînement en Mixed Precision
- Distribuer les entraînements avec DDP et FSDP
- Optimiser l'inférence avec la quantification et le pruning
- Exporter et industrialiser les modèles avec torch.export et ONNX

Public visé

- Machine Learning Engineers
- AI Engineers
- Data Scientists expérimentés
- Ingénieurs MLOps

Pré-requis

- Bonne maîtrise de Python et PyTorch
- Expérience dans la conception et l'entraînement de réseaux de neurones
- Maîtrise des Tensors, Autograd, Dataset et DataLoader
- Connaissances des principes du Deep Learning

Pré-requis logiciels

- 8 Go de RAM au minimum, 16 Go recommandés
- Accès à un GPU compatible CUDA ou à un environnement Cloud GPU
- Environnement Python avec Jupyter Notebook ou VS Code
- Accès à plusieurs GPU ou à une infrastructure adaptée pour les ateliers DDP et FSDP

Programme de notre formation PyTorch avancé

[Jour 1 - Matin]

Profiler et diagnostiquer les performances PyTorch

- Analyser les performances CPU et GPU avec PyTorch Profiler
- Identifier les goulets d'étranglement liés au calcul, aux transferts et aux entrées/sorties
- Analyser l'utilisation et l'allocation de la mémoire GPU
- Diagnostiquer les problèmes de gradients, synchronisations et opérations coûteuses
- Utiliser les hooks et outils de debugging pour inspecter l'exécution des modèles
- Atelier pratique : profiler un modèle pré-entraîné, identifier ses principaux goulets d'étranglement et proposer un plan d'optimisation.

[Jour 1 - Après-midi]

Accélérer les modèles avec torch.compile et TorchInductor

- Comprendre le fonctionnement de torch.compile et de la compilation de graphes PyTorch
- Découvrir le rôle de TorchInductor dans l'optimisation de l'exécution
- Identifier et diagnostiquer les graph breaks et recompilations
- Comparer les différents modes de compilation selon les contraintes de performance
- Mesurer les gains de compilation sur l'entraînement et l'inférence
- Atelier pratique : compiler un modèle, analyser les graph breaks puis mesurer les gains obtenus par rapport à l'exécution eager.

[Jour 2 - Matin]

Optimiser l'entraînement, la mémoire et les pipelines de données

- Mettre en œuvre le Mixed Precision Training pour réduire la consommation mémoire et accélérer les calculs
- Comprendre les formats numériques et arbitrer entre précision, stabilité et performance
- Optimiser les Dataset et DataLoader : workers, prefetching, pinned memory et transferts vers le GPU
- Réduire les copies, synchronisations et allocations mémoire inutiles
- Diagnostiquer les problèmes de gradients et stabiliser l'entraînement de modèles complexes
- Atelier pratique : optimiser un pipeline d'entraînement en combinant mixed precision, DataLoader et gestion mémoire.

[Jour 2 - Après-midi]

Distribuer l'entraînement avec DDP et FSDP

- Comprendre les architectures d'entraînement multi-GPU et les coûts de communication
- Distribuer l'entraînement avec DistributedDataParallel (DDP)
- Comprendre le sharding des paramètres, gradients et états de l'optimiseur avec FSDP
- Choisir entre DDP et FSDP selon la taille du modèle, la mémoire disponible et l'infrastructure
- Diagnostiquer les problèmes de synchronisation, mémoire et performance dans un entraînement distribué
- Atelier pratique : adapter un entraînement PyTorch au multi-GPU puis comparer une stratégie DDP et FSDP.

[Jour 3 - Matin]

Optimiser les modèles et l'inférence

- Exploiter des modèles pré-entraînés avancés avec PyTorch et Hugging Face
- Appliquer des techniques de quantification pour réduire la mémoire et accélérer l'inférence
- Utiliser le pruning pour réduire la complexité des modèles

- Optimiser les batches, la mémoire et la latence selon les contraintes d'inférence
- Mesurer les compromis entre précision, throughput, latence, mémoire et coût
- Atelier pratique : quantifier et optimiser un modèle pré-entraîné puis comparer ses performances avant et après optimisation.

[Jour 3 - Après-midi]

Exporter, monitorer et industrialiser les modèles PyTorch

- Exporter les modèles avec torch.export pour préparer leur exécution hors de l'environnement d'entraînement
- Exporter et valider les modèles avec ONNX et contrôler la cohérence des résultats
- Préparer les modèles pour leur intégration dans une chaîne d'inférence en production
- Versionner les modèles, configurations, métriques et expériences
- Intégrer le monitoring, la reproductibilité et les pratiques MLOps au cycle de vie PyTorch
- Atelier pratique : optimiser et exporter un modèle, valider son inférence puis construire un workflow d'industrialisation et de suivi des performances.

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.