

Mis à jour le 05/05/2026

S'inscrire

Formation deep Learning avec PyTorch

3 jours (21 heures)

Présentation

PyTorch est devenu le framework de référence pour la recherche et l'industrie en Intelligence Artificielle. Sa flexibilité, son approche par graphe dynamique et son intégration native avec Python en font l'outil privilégié pour concevoir des architectures de Deep Learning innovantes.

Notre formation Deep Learning avec PyTorch vous permettra de maîtriser les concepts fondamentaux des réseaux de neurones, de la manipulation des Tensors à la création d'architectures complexes comme les CNN ou les Transformers.

Vous apprendrez à structurer vos phases d'entraînement, à optimiser la convergence de vos modèles et à utiliser des outils avancés pour le débogage et la visualisation. La formation met l'accent sur la pratique pour vous rendre autonome dans la résolution de problématiques concrètes de vision par ordinateur ou de traitement du langage.

À l'issue de la formation, vous serez en mesure de concevoir, d'entraîner et de déployer des modèles de Deep Learning performants, tout en adoptant les meilleures pratiques de développement liées à l'écosystème PyTorch.

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

Objectifs pédagogiques

- Construire des réseaux de neurones
- Entraîner et évaluer des modèles
- Optimiser et déboguer les architectures
- Déployer des modèles DL

Public visé

- Data scientists
- AI engineers

Pré-requis

- Maîtrise du langage Python
- Connaissances en Machine Learning
- Notions de mathématiques (algèbre linéaire, dérivées)

Pré-requis logiciels

- 8 Go de RAM au minimum, 16 Go recommandés
- Accès à un GPU (via Google Colab, Kaggle ou machine locale avec CUDA)
- Environnement Python avec Jupyter Notebook ou VS Code

Programme de notre formation Deep Learning avec PyTorch

[Jour 1 - Matin]

Fondamentaux de PyTorch et Tensors

- Comprendre l'écosystème PyTorch et ses avantages
- Manipulation des Tensors : création, opérations et broadcasting
- Le moteur d'autodifférenciation : Autograd
- Gestion des périphériques (CPU vs GPU/CUDA)
- Introduction au module torch.nn
- Atelier pratique : Implémentation d'une régression linéaire "from scratch" avec Autograd.

[Jour 1 - Après-midi]

Construction de réseaux de neurones (MLP)

- Structure d'un Module PyTorch : méthode `__init__` et `forward`
- Fonctions d'activation (ReLU, Sigmoid, Softmax)
- Initialisation des poids et biais
- Les couches denses (Linear layers)
- Architecture des réseaux de neurones multi-couches
- Atelier pratique : Création d'un Perceptron Multicouche pour la classification de données tabulaires.

Entraînement et Boucles d'optimisation

- Choix des fonctions de coût (Loss Functions)
- Algorithmes d'optimisation (SGD, Adam, RMSprop)
- Cycle d'entraînement : Forward pass, Loss calculation, Backward pass, Optimizer step
- Suivi des métriques de performance
- Gestion des DataLoaders et Datasets
- Atelier pratique : Entraînement complet d'un modèle sur le dataset MNIST.

[Jour 2 - Matin]

Deep Learning pour la Vision (CNN)

- Principes des réseaux de neurones convolutifs (CNN)
- Couches de convolution, pooling et dropout
- Architectures célèbres (ResNet, VGG) et Transfer Learning
- Augmentation de données avec torchvision
- Fine-tuning de modèles pré-entraînés
- Atelier pratique : Classification d'images avec un modèle CNN personnalisé et pré-entraîné.

[Jour 2 - Après-midi]

Optimisation et Régularisation

- Prévenir le sur-apprentissage (Overfitting)
- Techniques de Batch Normalization
- Stratégies de Learning Rate Scheduling
- Utilisation de TensorBoard ou Weights & Biases pour la visualisation
- Sauvegarde et chargement de modèles (state_dict)
- Atelier pratique : Optimisation d'un modèle complexe et visualisation de la convergence.

Débogage et Profiling

- Identifier les problèmes de gradients (vanishing/exploding)
- Vérification des dimensions des Tensors
- Utilisation des hooks PyTorch pour inspecter les couches
- Profiling des performances mémoire et calcul
- Atelier pratique : Session de débogage sur une architecture volontairement défailante.

[Jour 3 - Matin]

Introduction aux modèles Séquentiels et Transformers

- Réseaux récurrents (RNN, LSTM, GRU)
- Mécanismes d'Attention
- Introduction à l'architecture Transformer
- Utilisation de la bibliothèque Hugging Face avec PyTorch
- Applications au NLP (Traitement du langage naturel)
- Atelier pratique : Analyse de sentiments ou génération de texte simple.

[Jour 3 - Après-midi]

Déploiement et Industrialisation

- Optimisation des modèles pour l'inférence
- Conversion vers TorchScript et ONNX
- Introduction à TorchServe pour le service de modèles
- Quantification et élagage (Pruning) de modèles
- Bonnes pratiques MLOps avec PyTorch
- Atelier pratique : Export d'un modèle et simulation d'une API d'inférence.

Projet de synthèse final

- Définition d'une problématique métier
- Choix et conception de l'architecture
- Entraînement optimisé et évaluation rigoureuse
- Présentation des résultats et pistes d'amélioration
- Atelier pratique : Challenge final de groupe sur un dataset réel.

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.