

SOMMAIRE DE L'EXPOSE SUR TENSORFLOW

Table des matières

I. Introduction.....	2
1. Présentation de TensorFlow : bibliothèque open-source créée par Google pour le Machine Learning et le Deep Learning.	2
II. Concepts fondamentaux.....	2
1. Définition de TensorFlow.	2
2. A quoi sert TensorFlow ?	2
3. Principe des graphes de calcul et auto-différentiation.	3
III. Architecture de TensorFlow.....	3
1. APIs : haut niveau (Keras), bas niveau (Core), et écosystème (TFX, TF Hub).	3
2. Nouveautés principales de TensorFlow 2.x.....	3
IV. Applications en Data Mining.....	3
1. Étapes clés : prétraitement, extraction, modélisation, validation.....	3
V. Exemples pratiques.....	4
1. Classification d'images (CNN).	4
2. Analyse prédictive (régression).....	4
VI. Évolutions récentes	4
1. TensorFlow Decision Forests.....	4
1. Prédiction de churn (attrition client)	4
2. Détection de fraude.....	5
3. Scoring de crédit ou de risque	5
Avantages clés de TF-DF pour ces cas.....	5
2. TensorFlow Lite (appareils mobiles).	5
3. TensorFlow Quantum (IA quantique).....	5
4. TensorFlow.js (applications web).	6
VII. Avantages et limites	6
1. Force : communauté, multiplateforme, accélération GPU/TPU.....	6
2. Limite : complexité initiale, besoin de ressources matérielles.....	7
VIII. Conclusion.....	7
1. Rappel des points essentiels.....	7
2. Perspectives futures (IA embarquée, générative, quantique).....	8

I. Introduction

1. Présentation de TensorFlow : bibliothèque open-source créée par Google pour le Machine Learning et le Deep Learning.

TensorFlow est une bibliothèque logicielle open-source créée par Google Brain en 2015, spécialisée dans les calculs numériques et l'apprentissage automatique, en particulier l'apprentissage profond (Deep Learning). Elle offre une structure flexible permettant de développer, entraîner et déployer facilement divers modèles de Machine Learning sur des plateformes multiples.

II. Concepts fondamentaux

1. Définition de TensorFlow.

TensorFlow est une bibliothèque open-source développée par google, principalement utilisée pour le **Machine Learning** (apprentissage automatique) et le **Deep Learning** (apprentissage profond).

2. A quoi sert TensorFlow ?

TensorFlow permet de :

- Concevoir et entraîner des modèles d'intelligence artificielle (IA)
- Traiter de grandes quantités de données
- Créer des réseaux de neurones complexes pour la reconnaissance d'images, de sons, de textes, etc.

Fonctionnement :

TensorFlow s'appuie sur un système de graphiques de calculs :

- Les Tensors sont des structures de données (comme des matrices ou des vecteurs)
- Les opérations mathématiques sont représentées sous forme de graphes
- Le calcul peut être reparti sur des CPU, GPU, TPU pour de meilleures performances

Avantages :

- Compatible avec python, mais aussi d'autre langages (C++, JavaScript, etc)
- Très utilisé en recherche et en industrie (Google Translate, Google Photos, etc)
- Dispose de nombreuses extensions :
 - **Keras** (interface simplifiée)
 - **TensorFlow** Lite (pour mobiles)
 - **TensorFlow.js** (pour le web)
 - **TensorBoard** (Visualisation de modèles et performances)

Exemple d'usage :

- Reconnaissance Faciale ou Vocale
- Analyse de sentiment sur des textes
- Détection de fraudes
- Prédiction des ventes

- Voitures autonomes

3. Principe des graphes de calcul et auto-différentiation.

TensorFlow utilise des graphes de calcul pour représenter les opérations mathématiques. Un graphe est composé de nœuds (opérations) et d'arêtes (tenseurs). L'auto-différentiation permet à TensorFlow d'effectuer automatiquement le calcul du gradient nécessaire à l'entraînement des réseaux neuronaux via la rétropropagation.

Pourquoi c'est important :

Dans le Deep Learning, on entraîne les réseaux neuronaux avec une méthode appelée descente de gradient, qui nécessite le calcul des dérivées de la fonction de perte (LOSS) par rapport aux poids du modèle.

III. Architecture de TensorFlow

1. APIs : haut niveau (Keras), bas niveau (Core), et écosystème (TFX, TF Hub).

- **API de haut niveau** : Keras offre une interface simple et intuitive pour la création et l'entraînement rapide des modèles.
- **API de bas niveau** : TensorFlow Core offre davantage de flexibilité et de contrôle, adaptée à des usages plus avancés.
- **Écosystème complémentaire** : TensorFlow Extended (TFX) et TensorFlow Hub facilitent l'intégration et le déploiement des modèles.

2. Nouveautés principales de TensorFlow 2.x.

TensorFlow 2.x introduit une exécution immédiate (Eager Execution) par défaut, simplifiant ainsi l'écriture et le débogage des codes. Elle améliore l'intégration de Keras et facilite l'utilisation des accélérations matérielles comme GPU et TPU.

IV. Applications en Data Mining

1. Étapes clés : prétraitement, extraction, modélisation, validation.

- **Prétraitement des données** : Cette phase inclut le nettoyage des données (suppression ou imputation des valeurs manquantes), la normalisation (mise à l'échelle des variables numériques), l'encodage des variables catégorielles (One-Hot Encoding) et parfois l'augmentation des données.
- **Extraction des caractéristiques** : Sélectionner les variables les plus pertinentes (feature selection) en utilisant des méthodes statistiques ou des modèles préliminaires, puis transformer les données (feature engineering) pour améliorer les performances.

- **Modélisation** : Construire des modèles supervisés (régression, classification) ou non supervisés (clustering, détection d'anomalies). TensorFlow propose des API facilitant la définition de réseaux neuronaux profonds (MLP, CNN, RNN) selon le problème à résoudre.
- **Entraînement et optimisation** : Ajuster les paramètres du modèle avec des algorithmes d'optimisation (par ex. Adam, SGD), régler l'apprentissage (learning rate schedules) et contrôler l'overfitting (early stopping, régularisation).
- **Validation et évaluation** : Utiliser la validation croisée, générer des métriques de performance (accuracy, f1-score, AUC) et interpréter les résultats avec TensorBoard ou d'autres outils de visualisation.

V. Exemples pratiques

1. Classification d'images (CNN).

Utilisation des réseaux neuronaux convolutifs (CNN) pour classer des images.
Exemple classique : Reconnaissance de chiffres manuscrits avec la base de données MNIST, en utilisant Keras intégré à TensorFlow.

2. Analyse prédictive (régression).

Application de TensorFlow pour prédire des valeurs continues, comme le prix des maisons (régression linéaire ou réseaux de neurones pour des prédictions avancées).

VI. Évolutions récentes

1. TensorFlow Decision Forests.

TensorFlow Decision Forests est une extension qui permet d'entraîner et d'utiliser des **modèles à base d'arbres de décision** dans l'environnement TensorFlow. Ces modèles incluent :

- Forêts aléatoires (Random Forests)
- Boosted Trees (Gradient Boosting)
- Cartes de décision (Decision Trees)

Avantages :

- Fonctionne bien avec des données tabulaires (non-images).
- Nécessite moins de réglages que les réseaux de neurones.
- Donne des résultats facilement interprétables.

TF-DF est particulièrement utile pour des cas classiques de data mining comme la prédiction de churn, la détection de fraude, etc.

Pour plus de détails. En Data Mining, on travaille souvent sur des **données structurées** (colonnes + lignes comme dans Excel). TF-DF est très efficace dans ce contexte, pour des tâches typiques comme :

1. Prédiction de churn (attrition client)

- **Objectif** : prédire si un client va quitter un service (abonnement téléphonique, banque, streaming, etc.).

- **Variables typiques** : ancienneté, fréquence d'utilisation, historique de paiement, service utilisé, etc.
- **TF-DF** excelle ici car il peut gérer des données tabulaires avec peu de prétraitement, et donne une bonne performance.

2. Détection de fraude

- **Objectif** : identifier des transactions anormales (fraude à la carte, faux paiements...).
- **Variables** : montant, lieu, fréquence, type de paiement, heure...
- TF-DF est capable de détecter **des schémas complexes dans les données catégorielles et numériques**, très utiles pour repérer des fraudes.

3. Scoring de crédit ou de risque

- Estimer si un client est solvable ou à risque.
- Données utilisées : revenus, historique de crédit, dettes, âge, etc.
- Très utilisé par les banques, les assureurs, etc.

Avantages clés de TF-DF pour ces cas

- **Pas besoin de normaliser les données** comme avec les réseaux neuronaux.
- **Résultats explicables** : on peut voir pourquoi une décision a été prise.
- **Excellente performance** sur de nombreux benchmarks en Data Mining.

2. TensorFlow Lite (appareils mobiles).

TensorFlow Lite est une version légère de TensorFlow destinée à **exécuter des modèles sur des appareils mobiles ou embarqués**, comme Android, Raspberry Pi, ou microcontrôleurs.

Caractéristiques principales :

- **Taille réduite des modèles** grâce à la quantification.
- **Exécution rapide et optimisée** sur CPU mobiles ou GPU embarqués.
- Compatible avec Android (via Android Studio) et iOS.

Pipeline typique :

1. Entraîner le modèle avec TensorFlow.
2. Le convertir avec TFLiteConverter.
3. L'utiliser dans une application mobile.

3. TensorFlow Quantum (IA quantique).

TensorFlow Quantum est une extension développée en collaboration avec Google Quantum AI. Elle permet de **concevoir, simuler et entraîner des modèles hybrides quantique-classique**.

Objectif :

- Intégrer les circuits quantiques (Qubits + portes logiques) dans les réseaux neuronaux.
- Explorer des applications comme la chimie quantique, l'optimisation, ou le Machine Learning avancé.

Fonctionnement :

- Représentation des circuits quantiques avec Cirq.
- Entraînement avec TensorFlow comme pour un modèle classique.

TFQ reste expérimental, mais représente une avancée importante dans le domaine de l'IA quantique.

4. TensorFlow.js (applications web).

TensorFlow.js permet d'exécuter des modèles **directement dans le navigateur Web**, sans serveur, en JavaScript.

Deux cas d'usage principaux :

- **Utilisation de modèles pré entraînés** (vision, NLP, pose humaine, etc.).
- **Entraînement de modèles directement dans le navigateur.**

Avantages :

- Pas besoin de backend ni de serveur Python.
- Facile à déployer pour des démonstrations ou des applis interactives.
- Compatible avec WebGL pour accélération GPU côté client.

VII. Avantages et limites

1. Force : communauté, multiplateforme, accélération GPU/TPU.

- **Large communauté et écosystème riche**
TensorFlow bénéficie d'un vaste soutien de la communauté open-source, ainsi que d'une documentation abondante, de nombreux tutoriels, forums et exemples. Cela facilite la montée en compétence et la résolution rapide de problèmes.
- **Multiplateforme et multi-langage**
TensorFlow est compatible avec divers systèmes d'exploitation (Windows, Linux, macOS) et fonctionne sur de nombreuses plateformes : serveurs, navigateurs (avec TensorFlow.js), mobiles (avec TensorFlow Lite), objets connectés, etc. Il est aussi utilisable avec plusieurs langages (Python, C++, JavaScript...).
- **Accélération matérielle avec GPU et TPU**
TensorFlow est optimisé pour l'exécution sur des GPU (carte graphique) et TPU

(Tensor Processing Unit, développée par Google), ce qui permet d'entraîner des modèles très complexes en un temps réduit. C'est un atout majeur pour le Deep Learning.

- **Intégration avec l'écosystème Google et outils professionnels**

TensorFlow s'intègre parfaitement avec des outils comme Google Colab, TensorBoard (pour la visualisation), TensorFlow Serving (pour le déploiement), ou encore TFX pour les pipelines de production.

2. Limite : complexité initiale, besoin de ressources matérielles.

- **Courbe d'apprentissage parfois abrupte**

Pour les débutants, l'utilisation de TensorFlow peut sembler complexe, notamment lorsqu'on utilise l'API de bas niveau ou que l'on souhaite construire des architectures personnalisées. Cela nécessite une bonne compréhension des concepts de Machine Learning.

- **Nécessite des ressources matérielles puissantes**

L'entraînement de modèles complexes (réseaux neuronaux profonds, CNN, RNN) peut être très gourmand en ressources. Sur une machine sans GPU, les performances peuvent être très limitées, surtout avec de grands ensembles de données.

- **Verbo­sité du code avec l'API bas niveau**

Bien que l'API Keras ait beaucoup simplifié l'utilisation, l'API TensorFlow Core (plus bas niveau) peut encore générer du code long, complexe, et difficile à déboguer.

VIII. Conclusion

1. Rappel des points essentiels.

TensorFlow est aujourd'hui l'une des bibliothèques les plus puissantes et les plus utilisées dans le domaine du **Machine Learning** et du **Deep Learning**. Créée par Google Brain, elle offre un large éventail de fonctionnalités permettant de concevoir, entraîner, évaluer et déployer des modèles intelligents sur des plateformes variées : serveurs, navigateurs, appareils mobiles, objets connectés, etc.

Tout au long de cet exposé, nous avons vu :

- Les **fondements théoriques** de TensorFlow (tenseurs, graphes de calcul, auto-différentiation).
- Les **API disponibles**, avec une mention spéciale pour Keras qui simplifie grandement la création de modèles.
- Les **applications concrètes** dans le Data Mining (prétraitement, modélisation, validation).
- Des **cas pratiques**, notamment la classification d'images et la régression.
- Les **évolutions récentes** qui étendent l'usage de TensorFlow au web (TensorFlow.js), aux appareils mobiles (TensorFlow Lite), à l'informatique quantique (TensorFlow Quantum), et aux modèles explicables sur données tabulaires (TensorFlow Decision Forests).

- Enfin, nous avons souligné ses **atouts** (performance, flexibilité, accélération matérielle) ainsi que ses **limites** (complexité initiale, consommation de ressources).

2. Perspectives futures (IA embarquée, générative, quantique).

TensorFlow continue d'évoluer à un rythme soutenu, porté par l'essor de l'intelligence artificielle dans tous les secteurs. Plusieurs tendances fortes se dessinent pour les années à venir :

- **L'IA embarquée** : avec TensorFlow Lite, l'IA s'invite dans les smartphones, drones, montres connectées et objets intelligents. Cela permet des applications en temps réel, sans connexion internet.
- **L'IA générative** : TensorFlow est déjà utilisé pour entraîner des modèles génératifs (GAN, VAE, diffusion) capables de créer des images, du texte, ou même du code, avec des résultats impressionnants.
- **L'IA quantique** : avec TensorFlow Quantum, la fusion entre l'informatique quantique et l'apprentissage machine devient accessible aux chercheurs et développeurs, ouvrant des perspectives innovantes dans la simulation de phénomènes complexes.

En somme, TensorFlow est non seulement un outil de référence aujourd'hui, mais aussi une plateforme tournée vers l'avenir, capable de s'adapter aux défis technologiques de demain.