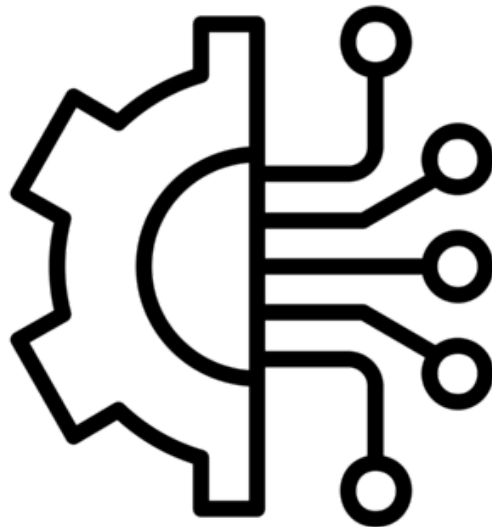


Spé IA - Deep Learning



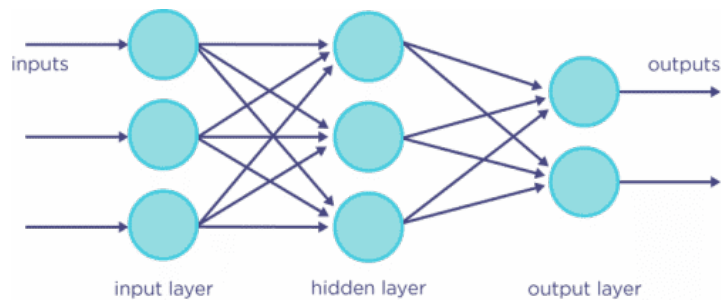
Sommaire :

I. Introduction.....	3
A. Définition du Deep Learning.....	3
B. Importance dans l'intelligence artificielle.....	3
C. Applications principales.....	4
1. Vision par ordinateur.....	4
2. Traitement du langage naturel (NLP).....	4
3. Reconnaissance vocale et audio.....	4
4. Automatisation et robotique.....	4
5. Santé et biotechnologie.....	4
II. Principes Fondamentaux du Deep Learning.....	5
A. Réseaux de neurones artificiels.....	5
1. Structure d'un neurone artificiel.....	5
2. Architecture des réseaux.....	5
3. Types de réseaux.....	5
B. Apprentissage supervisé VS non supervisé.....	6
1. Apprentissage supervisé.....	6
2. Apprentissage non supervisé.....	6
3. Principales différences.....	7
C. Rétropropagation et optimisation.....	7
1. Rétropropagation.....	7
2. Méthodes d'optimisation.....	8
III. Les Transformers : Révolution dans le Traitement du Langage Naturel (NLP).....	9
A. Problèmes des modèles séquentiels classiques (RNN, LSTM).....	9
B. Présentation de l'architecture Transformer (Self-Attention, Multi-Head Attention).....	10
1. Self-Attention.....	10
2. Multi-Head Attention.....	10
IV. Étude de Cas : Création d'un Outil de Traduction avec Apprentissage de Lettres et de Mots.....	12
A. Objectif du projet : Construire un modèle capable de traduire des mots d'une langue à une autre.....	12
B. Préparation des données.....	12
1. Collecte d'un corpus bilingue (exemple : anglais-français).....	12
2. Tokenisation : lettres VS mots.....	13
C. Modélisation.....	13
D. Entraînement et Évaluation.....	14
1. Entraînement.....	14
2. Évaluation.....	14
V. Conclusion et Perspectives.....	16
VI. Sources.....	17

I. Introduction

A. Définition du Deep Learning

Le Deep Learning est une branche de l'intelligence artificielle qui repose sur l'utilisation de réseaux de neurones artificiels à plusieurs couches pour apprendre des représentations complexes à partir des données de masse. Il s'agit d'une sous-discipline du Machine Learning, qui vise à entraîner des modèles, à reconnaître des motifs et à prendre des décisions de manière autonome.



L'apprentissage profond repose sur des architectures neuronales inspirées en partie du fonctionnement du cerveau humain, notamment les réseaux de neurones convolutifs (CNN) pour l'analyse d'images et les réseaux de neurones récurrents (RNN, LSTM, Transformers) pour le traitement du langage naturel (NLP).

Grâce à la puissance de calcul croissante et à la disponibilité de grandes quantités de données, le Deep Learning a permis des avancées majeures dans divers domaines :

- Reconnaissance d'images et de vidéos (ex : classification, détection d'objets)
- Traitement du langage naturel (ex : traduction automatique, génération de texte)
- Reconnaissance vocale (ex : assistants vocaux comme Siri, Alexa)
- Médecine et biologie (ex : diagnostic assisté par IA, découverte de médicaments)

L'un des principaux avantages du Deep Learning est sa capacité à extraire automatiquement des caractéristiques pertinentes des données, sans nécessiter une ingénierie manuelle avancée. Toutefois, il requiert d'importantes ressources de calcul et un grand volume de données pour obtenir des performances optimales.

B. Importance dans l'intelligence artificielle

Le Deep Learning est un élément clef de l'intelligence artificielle, car il permet aux machines d'analyser des données complexes et de prendre des décisions de manière autonome. Contrairement aux autres approches classiques basées sur des règles, le Deep Learning apprend directement à partir des données, ce qui améliore la précision et l'efficacité des modèles.

Grâce à sa capacité à traiter de grands volumes d'informations, il est utilisé dans de nombreux domaines : vision par ordinateur (reconnaissance faciale, voitures autonomes), traitement du langage naturel (traduction automatique, assistants vocaux) et santé (diagnostic médical assisté par IA). Son développement a été accéléré par la puissance des GPU et des avancées algorithmiques comme les Transformers, qui ont révolutionné le NLP.

C. Applications principales

Le Deep Learning a révolutionné de nombreux domaines grâce à sa capacité à traiter et analyser de grandes quantités de données avec une précision remarquable. Voici quelques-unes de ses principales applications.

1. Vision par ordinateur

- Reconnaissance faciale : utilisée dans la sécurité (ex : déverrouillage des smartphones, surveillance).
- Détection d'objets et classification d'images : identification d'éléments dans des images (ex : Google Photos, voitures autonomes).
- Imagerie médicale : détection de maladies à partir de radiographies ou d'IRM (ex : diagnostic du cancer).

2. Traitement du langage naturel (NLP)

- Traduction automatique : modèles comme Google Translate utilisent des réseaux neuronaux pour améliorer la qualité des traductions.
- Assistants vocaux et chatbots : Siri, Alexa, ChatGPT utilisent le Deep Learning pour comprendre et générer du texte.
- Analyse des sentiments : détection des émotions dans les avis clients ou sur les réseaux sociaux.

3. Reconnaissance vocale et audio

- Transcription automatique : conversion de la parole en texte (ex : YouTube, outils de sous-titrage automatique).
- Synthèse vocale : génération de voix artificielle (ex : assistants vocaux, IA comme VALL-E).

4. Automatisation et robotique

- Voitures autonomes : traitement en temps réel des données issues des capteurs (ex : Tesla, Waymo).
- Robots intelligents : utilisés dans l'industrie et la logistique (ex : Amazon Robotics).

5. Santé et biotechnologie

- Diagnostic assisté par IA : analyse de données médicales pour améliorer les diagnostics.
- Découverte de médicaments : accélération du développement de nouveaux traitements grâce à l'IA.

II. Principes Fondamentaux du Deep Learning

A. Réseaux de neurones artificiels

Ils se composent de neurones artificiels interconnectés, organisés en couches, qui traitent l'information en appliquant des transformations successives.

1. Structure d'un neurone artificiel

Un neurone artificiel reçoit des signaux d'entrée, les pondère, effectue une somme pondérée et applique une fonction d'activation pour produire une sortie. Cette sortie est ensuite transmise aux neurones des couches suivantes.

2. Architecture des réseaux

Les réseaux de neurones sont généralement organisés en plusieurs couches :

- Couche d'entrée : reçoit les données initiales.
- Couches cachées : effectuent des transformations intermédiaires.
- Couche de sortie : fournit le résultat final.

Cette organisation permet aux réseaux d'apprendre des représentations complexes et de modéliser des relations non linéaires.

3. Types de réseaux

Il existe différents types de réseaux de neurones, adaptés à diverses applications :

- Perceptron multicouche (MLP) : composé d'une couche d'entrée, de couches cachées et d'une couche de sortie, il est utilisé pour des tâches de classification et de régression.
- Réseaux de neurones convolutifs (CNN) : spécialisés dans le traitement des données structurées en grille, comme les images, ils sont efficaces pour la reconnaissance d'images et la vision par ordinateur.
- Réseaux de neurones récurrents (RNN) : adaptés au traitement des données séquentielles, comme le texte ou la parole, ils sont utilisés dans le traitement du langage naturel et la reconnaissance vocale.

Ces réseaux sont entraînés à l'aide d'algorithmes d'apprentissage supervisé ou non supervisé, leur permettant d'ajuster les poids synaptiques pour améliorer leurs performances sur des tâches spécifiques.

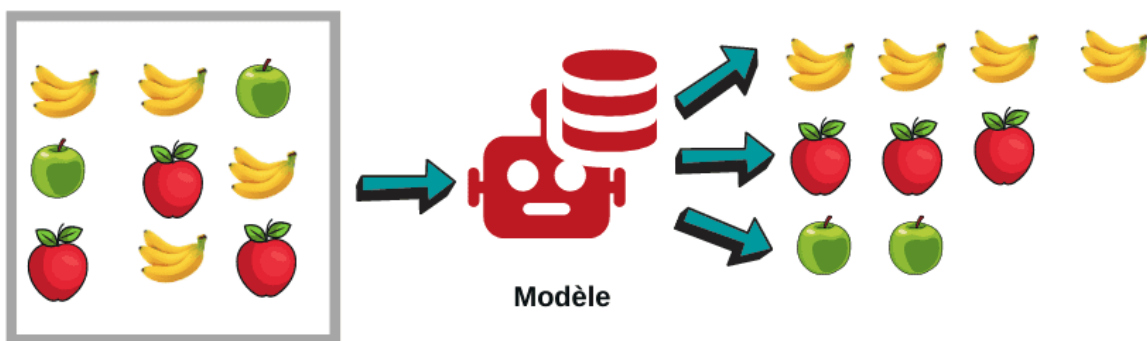
Les réseaux de neurones artificiels ont conduit à des avancées majeures dans divers domaines, notamment la vision par ordinateur, le traitement du langage naturel et la reconnaissance vocale, en permettant aux machines d'apprendre à partir de données et de s'adapter à des tâches diverses.

B. Apprentissage supervisé VS non supervisé

Dans le domaine de l'apprentissage automatique, deux approches principales sont utilisées pour entraîner les modèles : l'apprentissage supervisé et l'apprentissage non supervisé.

1. Apprentissage supervisé

L'apprentissage supervisé consiste à entraîner un modèle à partir de données étiquetées, c'est-à-dire des ensembles de données où chaque entrée est associée à une sortie connue. L'objectif est que le modèle apprenne à prédire la sortie correcte pour de nouvelles entrées similaires. Par exemple, en fournissant à un modèle des images de fruits avec leurs labels correspondants, il peut apprendre à distinguer ces catégories.



815 x 244

Caractéristiques :

- Données étiquetées : Nécessite un ensemble de données avec des labels connus.
- Objectif : Prédire des labels pour de nouvelles données.
- Exemples d'applications : Classification d'images, reconnaissance vocale, prédiction de valeurs numériques.

Cependant, l'apprentissage supervisé peut être limité par la disponibilité de données étiquetées de haute qualité et peut nécessiter une expertise pour structurer correctement les modèles.

2. Apprentissage non supervisé

L'apprentissage non supervisé, en revanche, travaille sur des données non étiquetées. L'algorithme tente de découvrir des structures ou des motifs cachés dans les données sans guidance explicite. Par exemple, il peut regrouper des clients en segments similaires en fonction de leurs comportements d'achat sans connaître à l'avance les catégories existantes.

Caractéristiques :

- Données non étiquetées : Ne nécessite pas de labels préexistants.
- Objectif : Découvrir des structures sous-jacentes, comme des regroupements ou des associations.
- Exemples d'applications : Segmentation de clients, réduction de dimensionnalité, détection d'anomalies.

L'apprentissage non supervisé peut traiter de grands volumes de données en temps réel, mais les résultats peuvent nécessiter une validation humaine pour assurer leur pertinence.

3. Principales différences

- Nature des données : L'apprentissage supervisé utilise des données étiquetées, tandis que l'apprentissage non supervisé travaille avec des données non étiquetées.
- Objectif : Le premier vise à prédire des résultats spécifiques, tandis que le second cherche à identifier des structures ou des motifs cachés.
- Complexité : L'apprentissage supervisé peut nécessiter une expertise pour structurer les modèles et des données étiquetées, tandis que l'apprentissage non supervisé peut traiter des données en temps réel mais nécessite souvent une validation humaine des résultats.

Comprendre ces distinctions est essentiel pour choisir la méthode appropriée en fonction des objectifs et de la nature des données disponibles.

C. Rétropropagation et optimisation

Dans le domaine de l'apprentissage profond, l'entraînement efficace des réseaux de neurones repose sur deux concepts fondamentaux : la rétropropagation et les méthodes d'optimisation.

1. Rétropropagation

La rétropropagation est un algorithme essentiel qui permet d'ajuster les poids des connexions dans un réseau de neurones en fonction de l'erreur observée entre la sortie prédite et la sortie souhaitée. Elle repose sur l'application de la règle de la chaîne pour calculer les dérivées partielles de la fonction de perte par rapport à chaque poids du réseau. Ce processus se déroule en deux phases principales :

1. Propagation avant : Les données d'entrée traversent le réseau, couche par couche, jusqu'à produire une sortie.
2. Propagation arrière : L'erreur entre la sortie prédite et la sortie réelle est propagée en sens inverse dans le réseau, permettant le calcul des gradients nécessaires à la mise à jour des poids.

Ce mécanisme d'ajustement est crucial pour minimiser l'erreur globale du réseau et améliorer sa capacité à généraliser sur des données non vues.

2. Méthodes d'optimisation

Les algorithmes d'optimisation jouent un rôle central dans l'ajustement des poids du réseau en fonction des gradients calculés lors de la rétropropagation. L'objectif est de trouver les valeurs de poids qui minimisent la fonction de perte du réseau. Parmi les méthodes d'optimisation couramment utilisées :

- Descente de gradient stochastique (SGD) : Met à jour les poids en utilisant le gradient de la fonction de perte par rapport à un échantillon de données, ce qui permet une convergence plus rapide et une meilleure généralisation.
- Méthodes adaptatives : Des algorithmes tels qu'AdaGrad, RMSProp et Adam ajustent dynamiquement le taux d'apprentissage en fonction des itérations et des caractéristiques des données, offrant ainsi une convergence plus stable et efficace.

Ces méthodes d'optimisation sont essentielles pour assurer un entraînement efficace et performant des réseaux de neurones profonds.

III. Les Transformers : Révolution dans le Traitement du Langage Naturel (NLP)

A. Problèmes des modèles séquentiels classiques (RNN, LSTM)

Les réseaux de neurones récurrents (RNN) et leurs variantes comme les Long Short-Term Memory (LSTM) ont été largement utilisés pour traiter des données séquentielles. Cependant, ces modèles présentent certaines limitations qui peuvent affecter leurs performances.

Problèmes des RNN classiques

- **Problème de gradient** : Les RNN souffrent souvent du problème de gradients évanescents ou explosifs lors de l'entraînement, rendant difficile l'apprentissage de dépendances à long terme.
- **Capacité de mémoire limitée** : Les RNN ont une capacité restreinte à mémoriser des informations sur de longues séquences, ce qui limite leur efficacité pour des tâches nécessitant une mémoire à long terme.
- **Problèmes de parallélisation** : En raison de leur nature séquentielle, les RNN traitent les données étape par étape, ce qui complique la parallélisation et peut ralentir l'entraînement et l'inférence.

Limites des LSTM

Les LSTM ont été conçus pour atténuer certains des problèmes des RNN, notamment en introduisant des mécanismes de mémoire pour mieux gérer les dépendances à long terme. Cependant, ils présentent également des limitations :

- **Complexité computationnelle** : Les LSTM possèdent des architectures plus complexes avec des mécanismes de portes sophistiqués, ce qui les rend plus lents à entraîner et à exécuter.
- **Traitement séquentiel** : Comme les RNN, les LSTM nécessitent de lire entièrement une séquence pour produire une prédiction, ce qui peut être inefficace pour certaines applications.

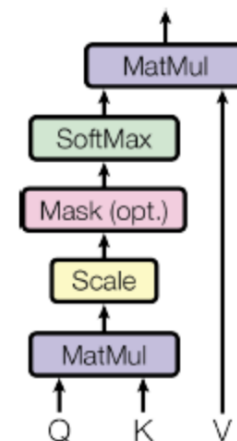
Ces limitations ont conduit au développement de nouvelles architectures, comme les Transformers, qui offrent des solutions plus efficaces pour capturer les dépendances à long terme et permettre une meilleure parallélisation, que nous allons voir par la suite.

B. Présentation de l'architecture Transformer (Self-Attention, Multi-Head Attention)

L'architecture Transformer a révolutionné le traitement du langage naturel en introduisant des mécanismes d'attention qui permettent de modéliser efficacement les dépendances au sein des données séquentielles, sans recourir à des structures récurrentes. Les deux composantes clés de cette architecture sont le Self-Attention et le Multi-Head Attention.

1. Self-Attention

Le mécanisme de Self-Attention permet à chaque élément d'une séquence de données de considérer tous les autres éléments de cette même séquence pour générer sa représentation. Concrètement, pour chaque mot d'une phrase, le modèle évalue l'importance des autres mots afin de capturer les relations contextuelles. Cette approche est particulièrement efficace pour saisir les dépendances à longue distance et les nuances contextuelles dans les données textuelles.



Le processus de Self-Attention implique la création de trois vecteurs pour chaque mot :

- Query (requête)
- Key (clé)
- Value (valeur)

Ces vecteurs sont obtenus en multipliant l'embedding du mot par des matrices de poids apprises. Ensuite, pour déterminer l'attention qu'un mot doit porter aux autres mots de la séquence, on calcule le produit scalaire entre le vecteur Query du mot en question et les vecteurs Key des autres mots. Les scores obtenus sont normalisés via une fonction softmax pour obtenir des coefficients d'attention, qui sont ensuite utilisés pour pondérer les vecteurs Value correspondants. La somme pondérée de ces vecteurs constitue la nouvelle représentation du mot.

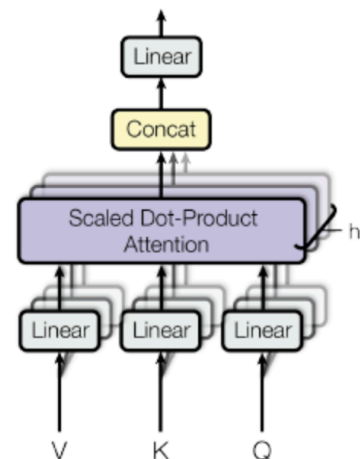
Ce mécanisme permet au modèle de focaliser son attention sur les parties pertinentes de la séquence lors du traitement de chaque mot, améliorant ainsi la qualité des représentations contextuelles.

2. Multi-Head Attention

Le Multi-Head Attention est une extension du mécanisme de Self-Attention. Au lieu de n'utiliser qu'une seule instance de Self-Attention, le modèle en exécute plusieurs en parallèle, appelées "têtes" d'attention. Chaque tête dispose de ses propres matrices de poids, ce qui lui permet de capturer différents types de relations contextuelles au sein de la séquence.

Les sorties de ces têtes d'attention sont ensuite concaténées et passées à une couche linéaire pour obtenir la représentation finale. Cette approche multi-têtes permet au modèle de :

- Capturer diverses relations contextuelles : chaque tête peut se concentrer sur des aspects différents des relations entre les mots.
- Améliorer la capacité d'expression du modèle : en combinant les informations de multiples têtes, le modèle obtient une représentation plus riche et plus robuste.



Ainsi, le Multi-Head Attention enrichit la capacité du Transformer à comprendre et à modéliser des structures complexes dans les données séquentielles.

En combinant ces mécanismes d'attention, l'architecture Transformer offre une flexibilité et une efficacité accrues dans le traitement des séquences, surpassant souvent les modèles séquentiels traditionnels tels que les RNN et LSTM vu précédemment.

IV. Étude de Cas : Création d'un Outil de Traduction avec Apprentissage de Lettres et de Mots

L'architecture Transformer a révolutionné le traitement du langage naturel en introduisant des mécanismes d'attention qui permettent de modéliser efficacement les dépendances au sein des données séquentielles, sans recourir à des structures récurrentes. Les deux composantes clés de cette architecture sont le Self-Attention et le Multi-Head Attention.

A. Objectif du projet : Construire un modèle capable de traduire des mots d'une langue à une autre

Le mécanisme de Self-Attention permet à chaque élément d'une séquence de données de considérer tous les autres éléments de cette même séquence pour générer sa représentation. Concrètement, pour chaque mot d'une phrase, le modèle évalue l'importance des autres mots afin de capturer les relations contextuelles. Cette approche est particulièrement efficace pour saisir les dépendances à longue distance et les nuances contextuelles dans les données textuelles.

Le processus de Self-Attention implique la création de trois vecteurs pour chaque mot :

- Query (requête)
- Key (clé)
- Value (valeur)

Ces vecteurs sont obtenus en multipliant l'embedding du mot par des matrices de poids apprises. Ensuite, pour déterminer l'attention qu'un mot doit porter aux autres mots de la séquence, on calcule le produit scalaire entre le vecteur Query du mot en question et les vecteurs Key des autres mots. Les scores obtenus sont normalisés via une fonction softmax pour obtenir des coefficients d'attention, qui sont ensuite utilisés pour pondérer les vecteurs Value correspondants. La somme pondérée de ces vecteurs constitue la nouvelle représentation du mot.

Ce mécanisme permet au modèle de focaliser son attention sur les parties pertinentes de la séquence lors du traitement de chaque mot, améliorant ainsi la qualité des représentations contextuelles.

B. Préparation des données

1. Collecte d'un corpus bilingue (exemple : anglais-français)

La collecte d'un corpus bilingue est une étape essentielle dans le développement d'outils de traduction automatique. Un corpus bilingue est constitué de paires de textes dans deux langues différentes, généralement des phrases ou des documents alignés, permettant au modèle d'apprendre les correspondances entre les langues.

Méthodes de collecte :

- Sources parallèles : Utilisation de documents déjà traduits, tels que les publications officielles, les sous-titres de films ou les sites web multilingues.
- Alignement automatique : Appariement de phrases de textes bilingues en utilisant des techniques d'appariement basées sur des statistiques de co-occurrences de mots.
- Collecte ciblée : Création de corpus spécifiques à un domaine en traduisant manuellement des documents pertinents ou en utilisant des outils de traduction assistée par ordinateur.

La qualité et la taille du corpus bilingue influencent directement les performances du modèle de traduction. Il est donc crucial de disposer de données représentatives et bien alignées.

2. Tokenisation : lettres VS mots

La tokenisation est le processus de découpage d'un texte en unités appelées "tokens". Ces tokens peuvent être des mots, des sous-mots ou des caractères individuels. Le choix du niveau de tokenisation a un impact significatif sur le traitement du langage naturel.

Tokenisation par mots :

- Avantages :
 - Représentation intuitive du texte.
 - Capture des unités sémantiques complètes.
- Inconvénients :
 - Sensibilité aux mots rares ou inconnus.
 - Vocabulaire de grande taille nécessitant plus de ressources.

Tokenisation par lettres :

- Avantages :
 - Vocabulaire réduit.
 - Gestion efficace des mots inconnus ou rares.
- Inconvénients :
 - Perte d'information sémantique liée aux mots entiers.
 - Nécessite des séquences plus longues pour capturer le contexte.

C. Modélisation

La modélisation en apprentissage profond consiste à concevoir et entraîner des réseaux de neurones artificiels capables d'apprendre des représentations hiérarchiques à partir de données brutes. Ces modèles, composés de multiples couches de neurones, permettent d'extraire des caractéristiques complexes et abstraites, facilitant ainsi la résolution de tâches variées telles que la classification d'images, la reconnaissance vocale ou la traduction automatique.

Principales étapes de la modélisation :

1. Conception de l'architecture : Sélectionner une structure adaptée au problème à résoudre, qu'il s'agisse de réseaux convolutifs pour le traitement d'images, de réseaux récurrents pour les données séquentielles ou de transformeurs pour le traitement du langage naturel.
2. Préparation des données : Collecter et prétraiter les données d'entraînement, en veillant à leur qualité et à leur pertinence.
3. Entraînement du modèle : Ajuster les poids du réseau en minimisant une fonction de coût à l'aide d'algorithmes d'optimisation tels que la rétropropagation du gradient.
4. Évaluation et ajustement : Tester les performances du modèle sur des données inédites et affiner ses hyperparamètres pour améliorer sa généralisation.

Une modélisation efficace en apprentissage profond repose sur une compréhension approfondie des architectures de réseaux, une préparation rigoureuse des données et une optimisation méthodique des paramètres du modèle.

D. Entraînement et Évaluation

L'entraînement et l'évaluation d'un modèle de traduction automatique basé sur les Transformers suivent un processus bien défini pour garantir une bonne compréhension des langues et une traduction fluide.

1. Entraînement

L'entraînement consiste à apprendre les correspondances entre les phrases de la langue source (ex français) et de la langue cible (ex anglais).

Étapes principales :

- Prétraitement des données : Normalisation du texte, suppression des caractères spéciaux, tokenisation (mots ou sous-mots).
- Encodage et décodage : Le modèle apprend à représenter les phrases d'entrée sous forme de vecteurs et à générer des phrases en sortie.
- Optimisation des paramètres : Utilisation de l'algorithme backpropagation et d'un optimiseur comme Adam pour ajuster les poids du réseau.
- Gestion du surapprentissage : Régularisation via dropout et utilisation d'un ensemble de validation pour ajuster les hyperparamètres.

2. Évaluation

Une fois entraîné, le modèle est testé sur des phrases inédites pour mesurer sa performance.

Métriques utilisées :

- BLEU Score : Compare la traduction générée avec une traduction de référence.
- Perplexité : Évalue la cohérence des phrases générées.

- Erreur humaine : Analyse qualitative des traductions pour identifier les erreurs sémantiques.

Un bon modèle de traduction doit être capable de produire des phrases naturelles, grammaticalement correctes et fidèles au sens d'origine.

V. Conclusion et Perspectives

L'apprentissage profond, et en particulier l'architecture Transformer, a révolutionné la traduction automatique en permettant une meilleure compréhension du contexte et des dépendances entre les mots. Grâce à des techniques comme le Self-Attention et le Multi-Head Attention, ces modèles offrent des performances bien supérieures aux approches classiques comme les RNN ou LSTM.

L'étude de cas sur la création d'un outil de traduction montre l'importance de chaque étape, depuis la collecte des données jusqu'à l'entraînement et l'évaluation du modèle.

L'optimisation des paramètres et l'utilisation de corpus de qualité sont essentielles pour obtenir des traductions précises et naturelles.

Avec les avancées continues en IA, les modèles de traduction automatique continueront de s'améliorer, rendant la communication entre langues de plus en plus fluide et accessible à tous.

VI. Sources

<https://ai.stackexchange.com/questions/20075/why-does-the-transformer-do-better-than-rnn-and-lstm-in-long-range-context-depen>

<https://arxiv.org/abs/2006.03860>

<https://perso.limsi.fr/pap/inalco/grefenstette.pdf>

<https://www.alteryx.com/fr/glossary/supervised-vs-unsupervised-learning>

<https://blog.octo.com/les-reseaux-de-neurones-recurrents-des-rnn-simples-aux-lstm>

<https://www.geeksforgeeks.org/rnn-vs-lstm-vs-gru-vs-transformers>

<https://huggingface.co/learn/nlp-course/fr/chapter6/6>

<https://www.ibm.com/fr-fr/think/topics/backpropagation>

<https://www.ibm.com/fr-fr/topics/neural-networks>

<https://www.ibm.com/fr-fr/topics/supervised-learning>

<https://medium.com/%40indrajitbarat9/recurrent-neural-networks-rnns-challenges-and-limitations-4534b25a394c>

https://www.researchgate.net/publication/384008011_Backpropagation_and_Optimization_in_Deep_Learning_Tutorial_and_Survey

https://fr.wikipedia.org/wiki/R%C3%A9seau_de_neurones_artificiels