

Introduction au Deep Learning

Jour 2: Data Science, Machine Learning et IA

Aziza Merzouki

Université de Genève

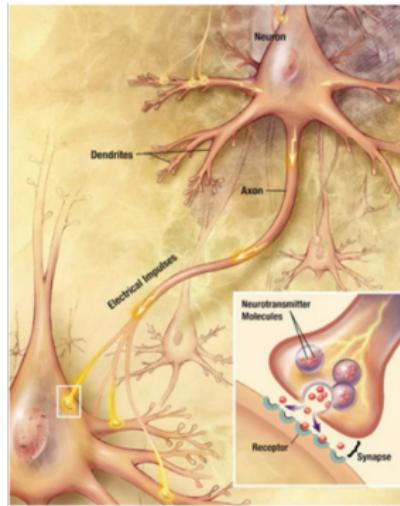
05 Décembre, 2025

Outline

- 1 Introduction
- 2 Réseaux de Neurones
- 3 Applications
- 4 Types de Deep Neural Networks
- 5 Transfert d'Apprentissage
- 6 Conclusion

Qu'est-ce que le Deep Learning?

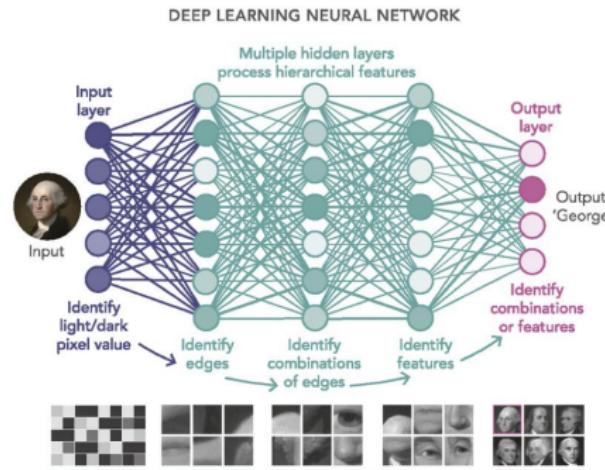
- Le Deep Learning s'inspire de la **structure du cerveau humain.**



Structures des neurones

Qu'est-ce que le Deep Learning?

- Composé de plusieurs couches : **Couche d'Entrée**, **Couches Cachées** et **Couche de Sortie**.
- Il apprend à créer des **motifs** à partir de données d'entrée pour les utiliser dans la **prise de décision**.



[Neural Networks and Deep Learning](#), a free online book written by Michael Nielsen.

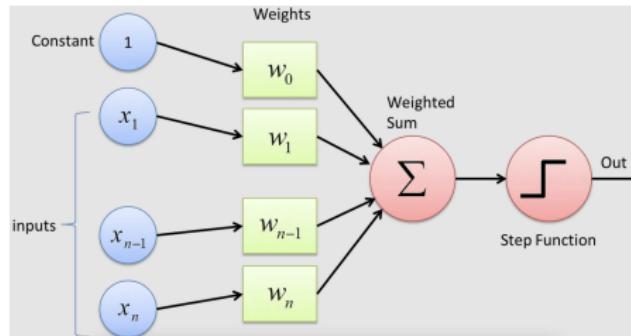
Réseaux de Neurones

- Les réseaux de neurones sont la pierre angulaire du Deep Learning.
- Ils sont composés de **couches de neurones** (artificiels) interconnectées.
- Chaque **neurone est connecté à d'autres neurones** pour le traitement de l'information.
- Chaque neurone reçoit des **entrées**, effectue un **calcul** et transmet la **sortie** aux neurones de la couche suivante.

Perceptron

- Le bloc de construction fondamental des réseaux de neurones.
- Prend des **entrées pondérées**, les **somme** et les passe à travers une **fonction d'activation**.
- Il peut être utilisé pour la **classification binaire**.

$$\text{Sortie } y = \sigma(w_0 + w_1x_1 + w_2x_2 + w_3x_3)$$

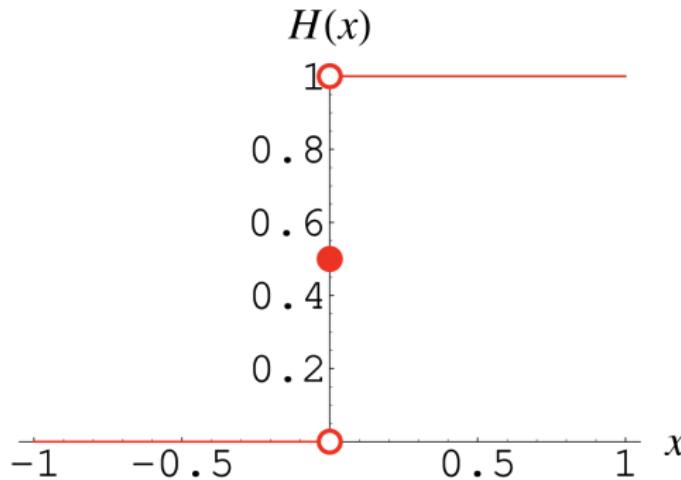


(1958, Rosenblatt)

Perceptron - Fonction d'Activation

Fonction de Heaviside :

$$H(u) = \begin{cases} 1, & \text{si } u \geq 0 \\ 0, & \text{sinon} \end{cases}$$



Fonctions d'Activation

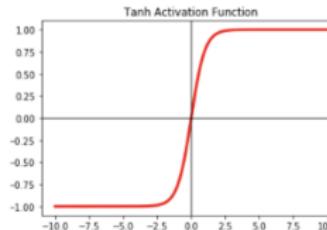
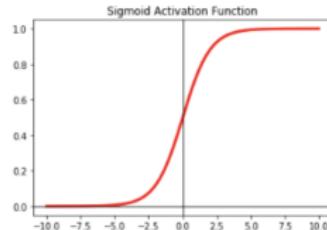
- Le Perceptron peut utiliser différentes fonctions d'activation.
- Deux exemples populaires de fonctions d'activation sont la fonction sigmoïde et la fonction tangente hyperbolique (\tanh).

Fonction Sigmoïde :

$$\sigma(u) = \frac{1}{1 + e^{-u}}$$

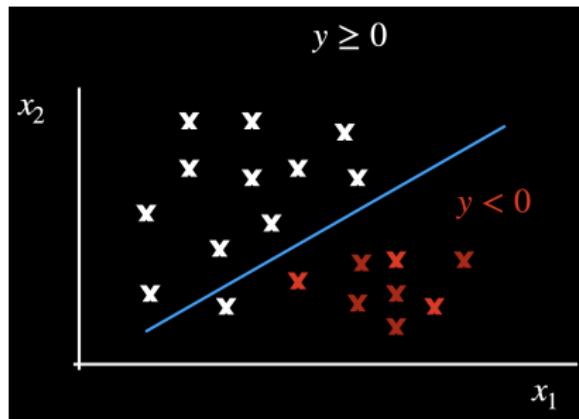
Fonction Tangente Hyperbolique (\tanh) :

$$\tanh(u) = \frac{e^u - e^{-u}}{e^u + e^{-u}}$$



Limites du Perceptron

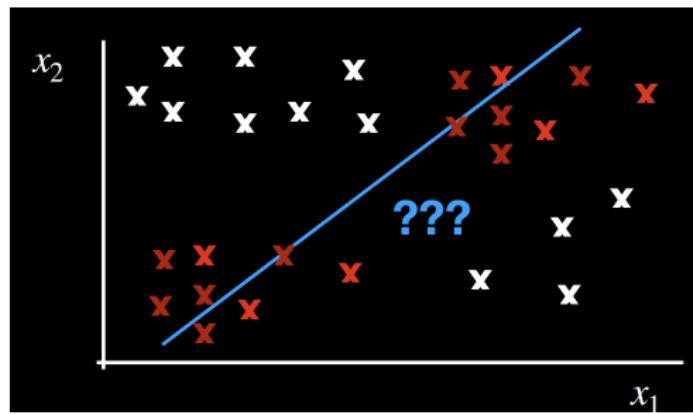
- Le Perceptron ne peut apprendre que des tâches qui sont **linéairement séparables.**



[Perceptron Algorithm for Classification in Python.](#)

Limites du Perceptron

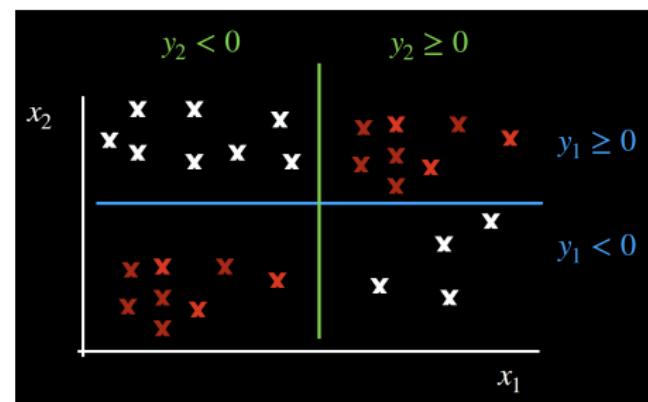
- Le Perceptron ne peut pas apprendre des tâches qui ne sont pas **linéairement séparables**, e.g. classification XOR.



Neural Representation of AND, OR, NOT, XOR and XNOR Logic Gates (Perceptron Algorithm).

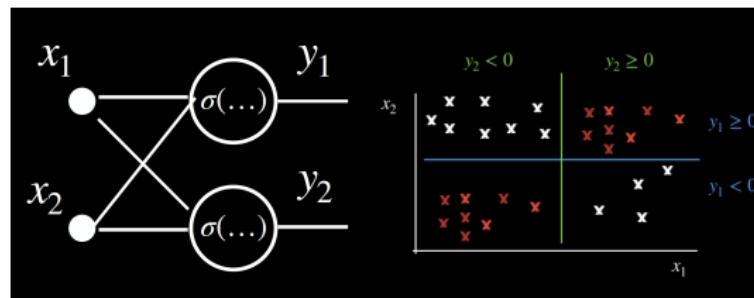
Surmonter la Limitation de Séparation Non Linéaire

- Pour résoudre un problème non linéaire, on peut le décomposer en plusieurs problèmes linéaires.



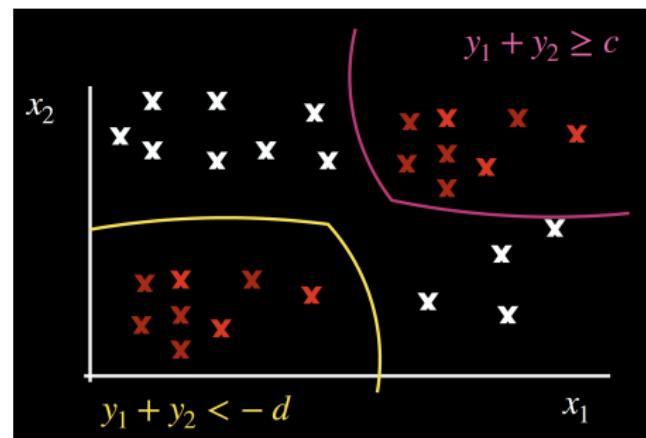
Surmonter la Limitation de Séparation Non Linéaire

- Chaque nouveau problème linéaire peut être résolu par un neurone.



Surmonter la Limitation de Séparation Non Linéaire

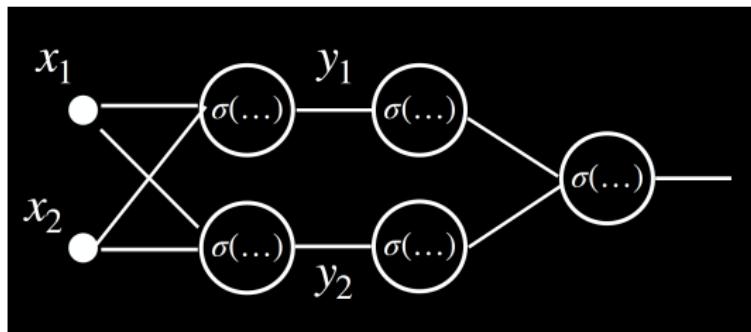
- La solution du problème non-linéaire peut être une combinaison des solutions des problèmes linéaires.



How to solve XOR Classification with one hidden layer of neurons.

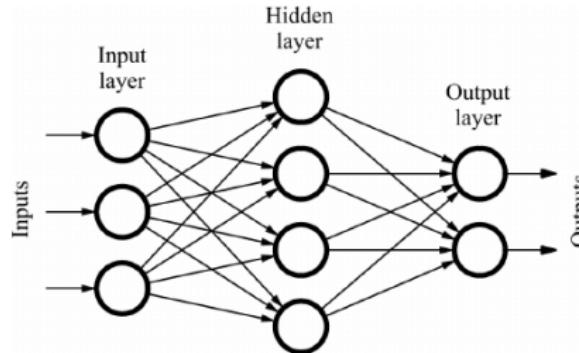
Multilayer Perceptron (MLP)

- Pour résoudre des problèmes non linéaires, on peut ajouter des **neurones et des couches supplémentaires**.

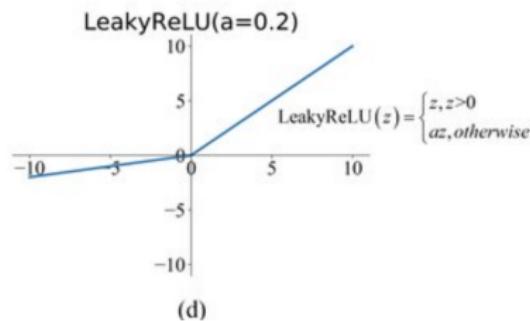
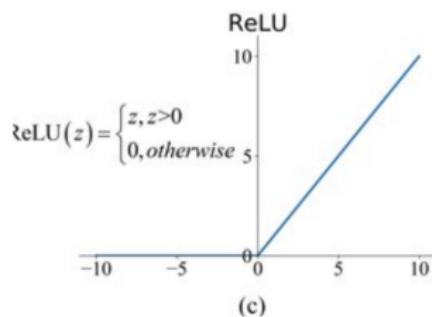
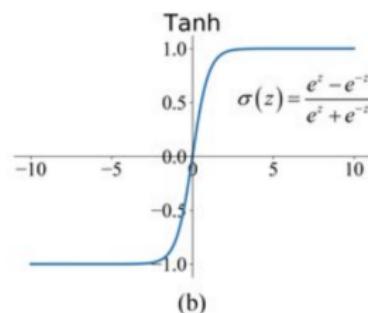
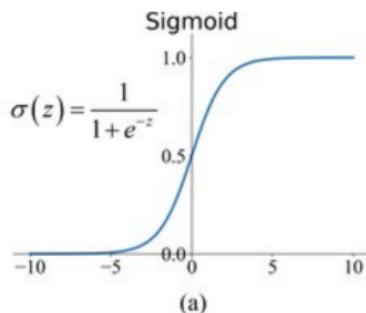


Feed Forward

- Le **Feed Forward** est le processus de **propagation des données** à travers le réseau de neurones
- Chaque couche du réseau "extrait" des caractéristiques (features) de la couche précédente.
- Les couches supérieures du réseau représentent des caractéristiques de plus haut niveau.

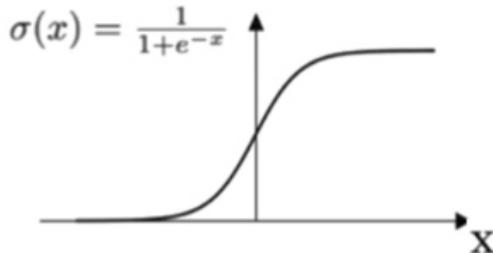
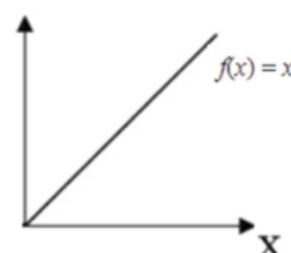


Fonctions d'activation - Couches cachées

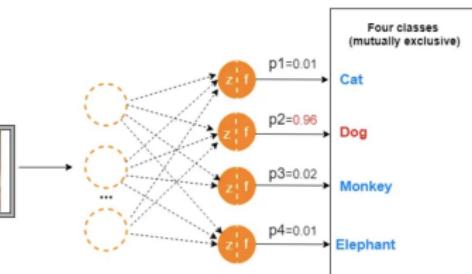


Towards Data Science; How to Choose the Right Activation Function for Neural Networks

Fonctions d'activation - Couches de sorties

Sigmoid**Linear**

$$\text{Softmax}(z_i) = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}}$$



Towards Data Science; How to Choose the Right Activation Function for Neural Networks

Optimisation - Rétropropagation (Backpropagation)

- La **rétropropagation** est utilisé pour **entraîner** les réseaux de neurones.
- Elle consiste à **ajuster les poids** des connexions en fonction de l'**erreur** calculée entre les **prédictions** du réseau et les **valeurs réelles** (étiquettes).
- La rétropropagation suit l'étape de la **propagation** avant (Feed Forward).

Ajustement des Poids

- Ajuster les poids du réseau pour minimiser la fonction de coût, notée $J(w, x)$.
- Descente de gradient (Gradient Descent) ou variante nécessite le calcul du gradient $\nabla_w J(w, x)$.

Mise-à-jour des poids (Gradient Descent) :

$$w_{new} = w_{old} - \alpha \cdot \nabla_w J(w, x)$$

Où :

- α est le taux d'apprentissage (learning rate).

Dérivées de la Fonction de Coût par Rapport aux Poids

Pour calculer les dérivées de la fonction de coût J par rapport aux poids $w_{ij}^{(l)}$, on utilise le théorème de dérivation des fonctions composées (**chain rule**) :

$$\frac{\partial J}{\partial w_{ij}^{(l)}} = \frac{\partial J}{\partial a_j^{(l)}} \cdot \frac{\partial a_j^{(l)}}{\partial z_j^{(l)}} \cdot \frac{\partial z_j^{(l)}}{\partial w_{ij}^{(l)}}$$

Où :

- $a_j^{(l)}$: activation du neurone j à la couche l
- $z_j^{(l)} = \sum_i w_{ij}^{(l)} a_i^{(l-1)} + b_j^{(l)}$: somme pondérée en entrée du neurone
- $\frac{\partial a_j^{(l)}}{\partial z_j^{(l)}}$ dépend de la fonction d'activation σ : $a_j^{(l)} = \sigma(z_j^{(l)})$
- $\frac{\partial z_j^{(l)}}{\partial w_{ij}^{(l)}} = a_i^{(l-1)}$

Khan Academy; Chain Rule

The Chain Rule of Calculus for Univariate and Multivariate Functions

Exemples d'applications

- Traitement

- Traitement d'images (e.g., reconnaissance d'objet)
- Traitement du langage/texte (e.g., analyse des sentiments)
- Traitement audio (e.g., transcription)

- Génération

- Génération d'images (e.g., transfert de style)
- Génération de texte (e.g. génération de poèmes)
- Génération d'audio (e.g., génération de musique)

Quizz

Perceptron

Ana aime aider son jeune frère avec ses devoirs d'informatique pendant son temps libre. Il suit un cours d'introduction au Machine Learning, et ses devoirs portent sur les Perceptrons. Il devait construire un Perceptron pour un problème de classification binaire, mais c'est sa première fois, et un Perceptron n'est pas une chose simple à mémoriser quand on n'a pas beaucoup d'expérience. Ana a décidé de concevoir quelques questions pour aider son frère. Ils ont un jeu de données avec 4 caractéristiques. Ils construisent le Perceptron et effectuent la passe avant en utilisant un vecteur contenant les valeurs d'un seul échantillon.

Quelle est la taille de la sortie si ce vecteur ressemble à ceci [1.0, 2.0, 1.0, 2.0] ?

- La sortie contiendra une seule valeur.
- La sortie contiendra deux valeurs.
- La sortie contiendra trois valeurs.
- La sortie contiendra quatre valeurs.

Quizz

Réseaux Neuronaux Profonds

Joanna est une chercheuse sur le changement climatique. Elle veut analyser des images satellites de la forêt amazonienne pour repérer les points chauds de déforestation afin d'aider les décideurs à agir de manière préventive. Joanna a construit un réseau neuronal peu profond pour prédire les schémas de déforestation, mais il n'est pas aussi précis qu'elle le souhaiterait. Un réseau neuronal profond pourrait être meilleur.

Que devrait faire Joanna pour transformer son réseau neuronal peu profond en un réseau profond ?

- Joanna devrait changer la fonction d'activation dans les couches existantes.
- Joanna devrait changer le type de données d'entrée qu'elle utilise.
- Joanna devrait ajouter plus de couches au réseau.
- Joanna devrait utiliser un algorithme d'optimisation différent.

Quizz

Couches de Réseau Neuronal

Lauren est intriguée par le fonctionnement interne des réseaux neuronaux profonds et veut comprendre comment ils traitent les images pour faire des prédictions précises. Elle s'intéresse particulièrement à la manière dont ces réseaux gèrent les images, alors elle a pris un jeu de données de photos de chiens et a commencé à faire des expériences.

Laquelle des affirmations suivantes simplifie raisonnablement le fonctionnement du réseau ?

- Les premières couches d'un réseau neuronal calculent des caractéristiques plus complexes que les couches profondes.
- Les couches profondes d'un réseau neuronal calculent des caractéristiques plus complexes que les premières couches.
- La complexité des caractéristiques calculées par les couches du réseau neuronal est proportionnellement répartie parmi toutes les couches.
- La complexité des caractéristiques calculées par les couches du réseau neuronal n'a aucun rapport avec la position de la couche.

Quizz

Entropie Croisée Binaire

Adriana est une passionnée de Machine Learning. Elle a récemment entendu parler de l'entropie croisée binaire lors d'une discussion avec un collègue. Bien qu'elle soit familière avec diverses techniques de Machine Learning, l'entropie croisée binaire était nouvelle pour elle. Elle a ouvert un livre et a commencé à lire sur cette fonction de perte. Malheureusement, il y avait une question à laquelle elle ne pouvait pas répondre.

Pour quels problèmes de réseau neuronal devrait-on utiliser l'entropie croisée binaire comme fonction de perte ?

- L'entropie croisée binaire est la fonction de perte utilisée pour les problèmes de classification multi-étiquettes.
- L'entropie croisée binaire est la fonction de perte utilisée pour les problèmes de classification binaire.
- L'entropie croisée binaire est la fonction de perte utilisée pour les problèmes de classification multi-classes.
- L'entropie croisée binaire est la fonction de perte utilisée pour les problèmes de régression.

Quizz

Classification Multi-classes vs Multi-étiquettes

Camille travaille dans une entreprise qui produit des téléphones portables, et son équipe veut classifier les défauts en trois catégories : rayures visuelles, bogues logiciels et défauts matériels. Chaque téléphone portable peut avoir un ou plusieurs de ces défauts. L'équipe de Camille utilise un réseau neuronal pour résoudre ce problème, et elle essaie de déterminer la meilleure façon de le concevoir.

Quelle serait la meilleure approche pour l'équipe de Camille afin de concevoir le réseau pour classifier les défauts ?

- La couche de sortie du réseau devrait avoir une fonction d'activation softmax, et la fonction de perte devrait être l'entropie croisée catégorielle.
- La couche de sortie du réseau devrait avoir une fonction d'activation sigmoïde, et la fonction de perte devrait être l'entropie croisée binaire.
- La couche de sortie du réseau devrait avoir une fonction d'activation softmax, et la fonction de perte devrait être l'entropie croisée binaire.
- La couche de sortie du réseau devrait avoir une fonction d'activation sigmoïde, et la fonction de perte devrait être l'entropie croisée catégorielle.

Quizz

Règle de la Chaîne

Amy est une data scientist qui travaille dans une grande entreprise technologique. Elle est passionnée par la compréhension des principes fondamentaux du machine learning et cherche toujours à améliorer ses compétences. Récemment, Amy s'est plongée plus profondément dans le sujet des réseaux neuronaux et a découvert le concept de la règle de la chaîne en calcul. Malgré le temps passé à essayer de comprendre comment cela fonctionne, Amy n'est pas claire sur son utilité.

Lesquelles des caractéristiques suivantes sont propres à la règle de la chaîne ?

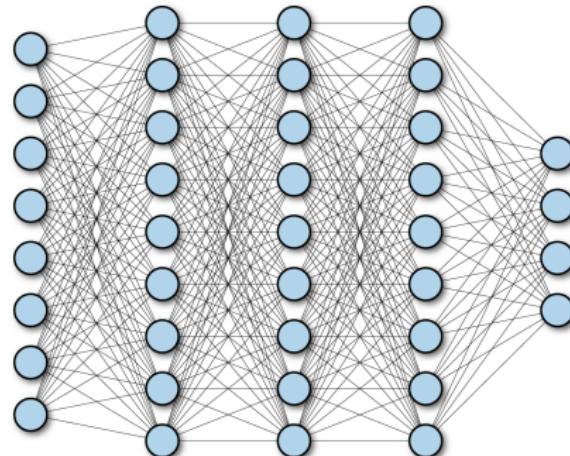
- La règle de la chaîne permet de calculer les dérivées des fonctions composées.
- La règle de la chaîne est basée sur l'idée que la dérivée d'une fonction composée est le produit de la dérivée de la fonction externe et de la dérivée de la fonction interne.
- La règle de la chaîne est utilisée pour calculer le gradient de la fonction de perte par rapport aux poids du réseau.
- La règle de la chaîne est utilisée dans l'algorithme de rétropropagation.

Architecture de Deep Neural Networks

- Dense
- Convolutional Neural Networks
- Recurrent Neural Networks
- Transformer Networks

Réseaux de Neurones Denses (Fully Connected)

- Architecture classique où chaque neurone d'une couche est connecté à tous les neurones de la couche précédente.
- Utilisés pour de la régression ou classification sur des données structurées, ou des caractéristiques élaborées (engineered features).



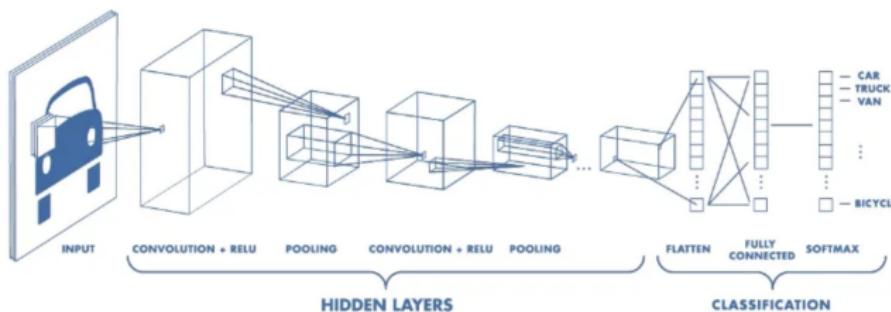
Réseaux de Neurones Denses (Fully Connected) - Limitations

Les réseaux de neurones denses, bien que puissants, présentent certaines limitations :

- Nombre de paramètres très grand
- Requiert énormément de données
- Temps d'apprentissage très long
- Risque de rester coincé dans un minimum local

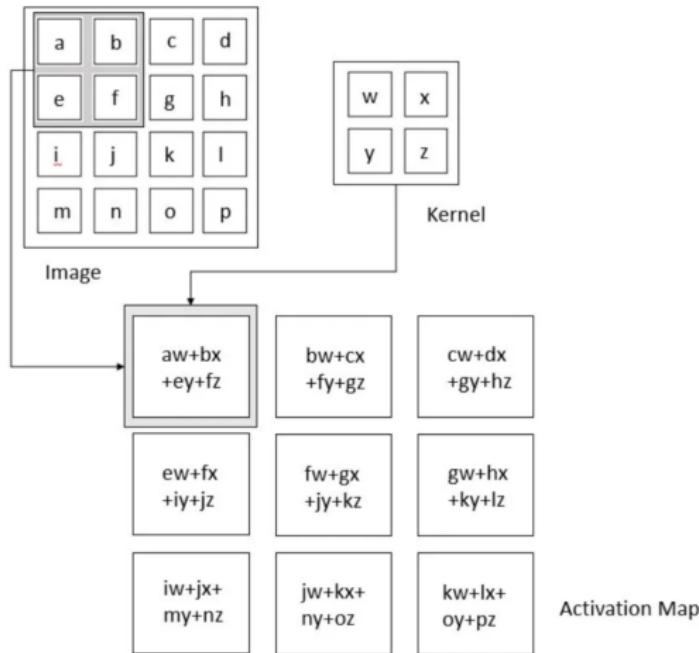
Réseaux de Neurones Convolutionnels (CNN)

- Conçus pour le traitement des **données en grille** telles que les images.
- Utilisent des couches de convolution pour extraire des "**caractéristiques locales**".
- Utilisés dans la vision par ordinateur.

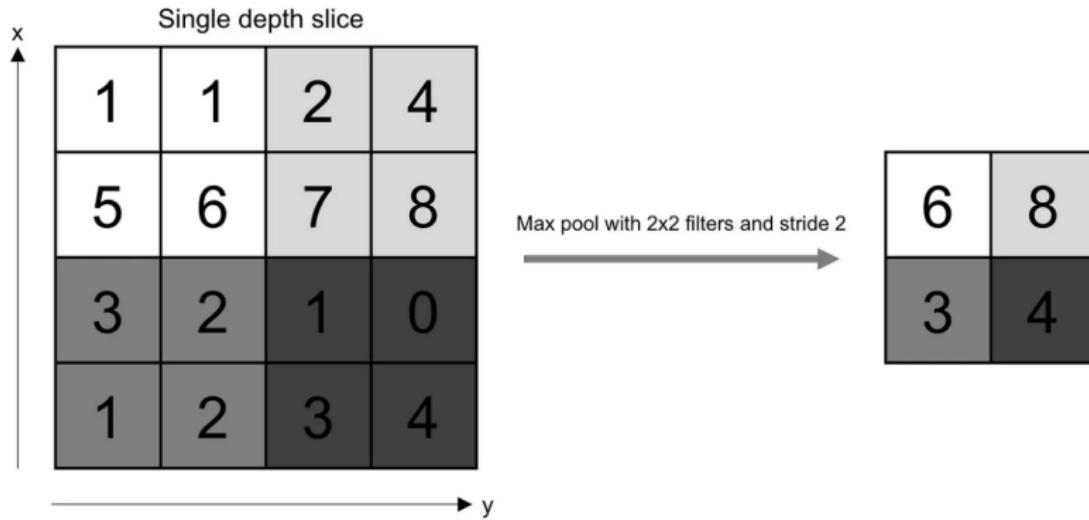


A Beginner's Guide to Convolutional Neural Networks (CNNs)

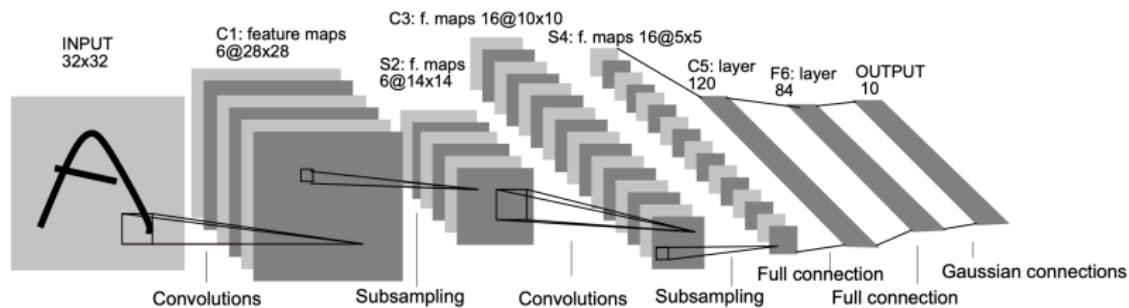
CNN - Convolution



CNN - Pooling



CNN - LeNet-5



(LeCunn - 1998; GradientBased Learning Applied to Document Recognition)

Quizz

Réseaux de Neurones Convolutionnels

Nina discute de la meilleure approche pour leur nouveau projet avec son collègue. Leur tâche est de développer un modèle pour identifier des objets dans des images de drones. Nina pense qu'un Réseau de Neurones Convolutionnel est le choix idéal, mais son collègue pense qu'ils devraient s'en tenir à un réseau neuronal entièrement connecté simple. Nina doit fournir des raisons convaincantes pour choisir un Réseau de Neurones Convolutionnel plutôt qu'un réseau entièrement connecté.

Lesquels des arguments suivants Nina pourrait-elle utiliser pour étayer son point de vue ?

- L'utilisation d'un Réseau de Neurones Convolutionnel nécessite moins de paramètres qu'un réseau entièrement connecté.
- Les Réseaux de Neurones Convolutionnels peuvent apprendre une hiérarchie de caractéristiques visuelles similaire à celle du cerveau humain, ce qui se traduit par de meilleures performances.
- Les Réseaux de Neurones Convolutionnels sont généralement moins profonds que les réseaux entièrement connectés, ce qui rend le processus d'entraînement plus facile et plus rapide.
- La structure locale de l'image peut être utilisée plus efficacement par un Réseau de Neurones Convolutionnel, ce qui se traduit par des caractéristiques et des performances bien meilleures.

Réseaux de Neurones Récurrents (RNN)

- Conçus pour travailler avec des **données séquentielles**.
- **Connexions récurrentes** qui leur permettent de conserver des **informations antérieures**.
- Utilisés dans le traitement du **langage naturel** et la **reconnaissance vocale**.

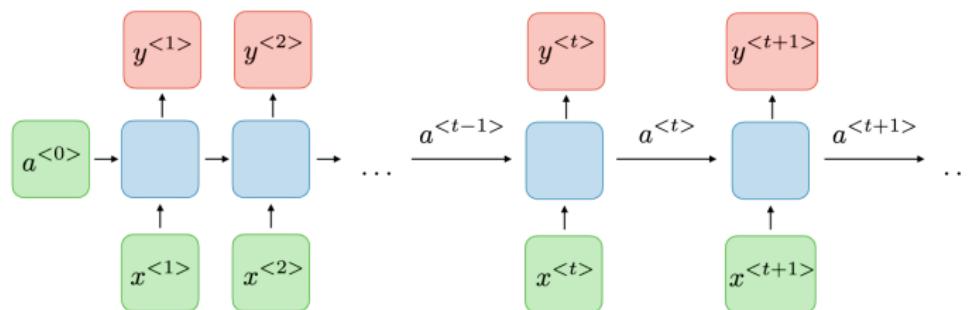
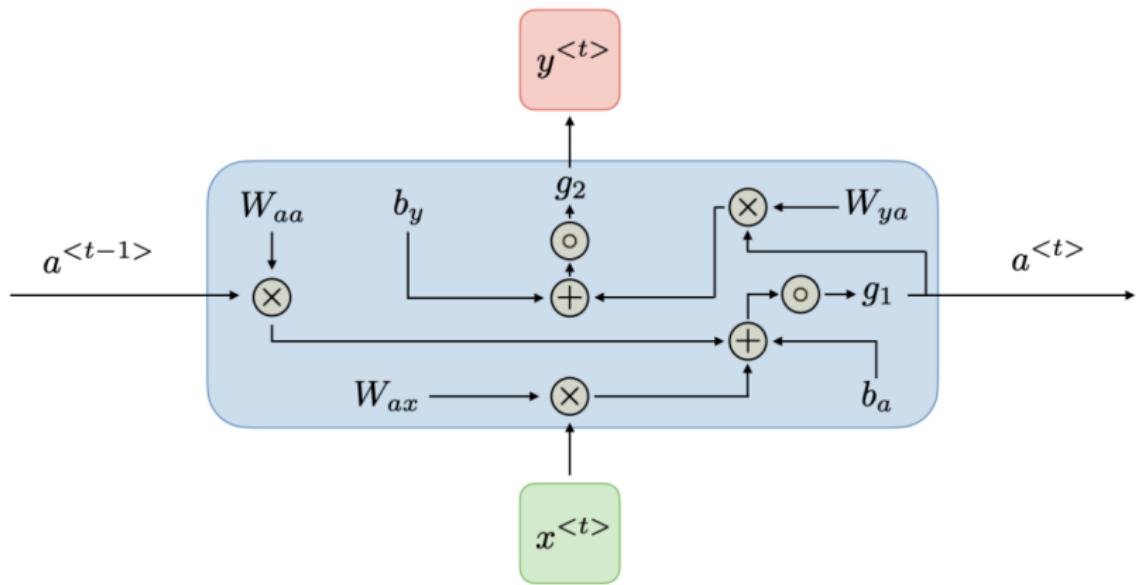


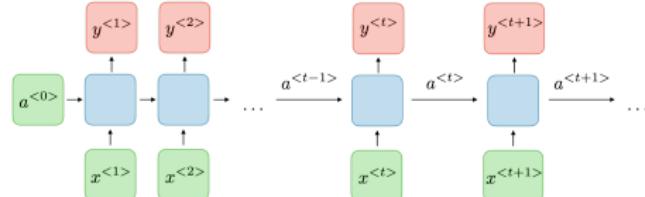
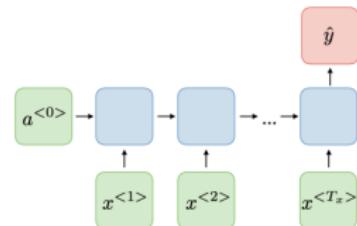
Figure: Exemple d'architecture Many-to-Many

Unité RNN

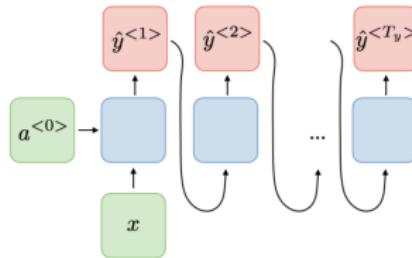
- Les poids sont indépendants du temps.



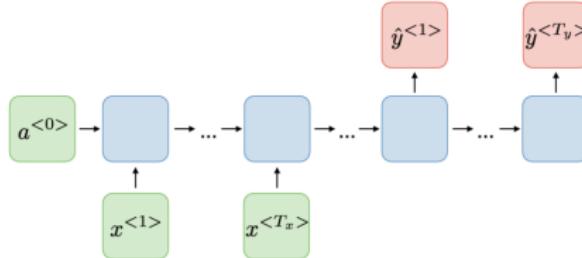
Architectures RNN

(a) Many-to-Many ($T_x = T_y$)

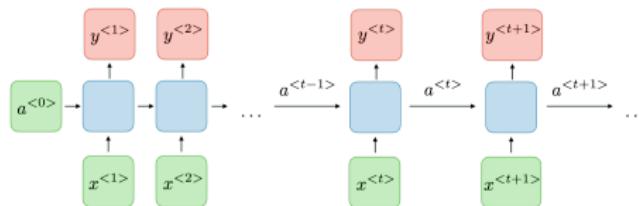
(b) Many-to-One



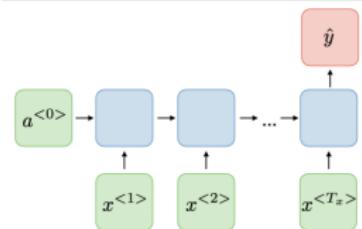
(c) One-to-Many

(d) Many-to-Many ($T_x \neq T_y$)

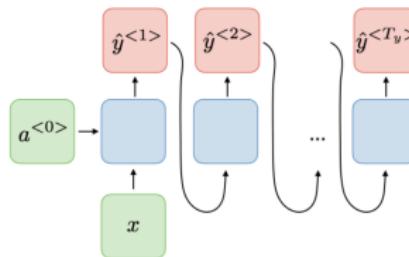
Architectures RNN - Applications



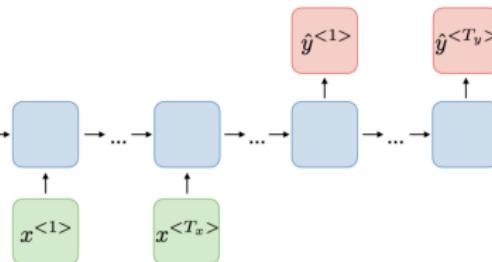
(a) Reconnaissance d'entités



(b) Analyse de sentiments



(c) Génération de musique



(d) Traduction automatique

Modèle de Langage (Language Model)

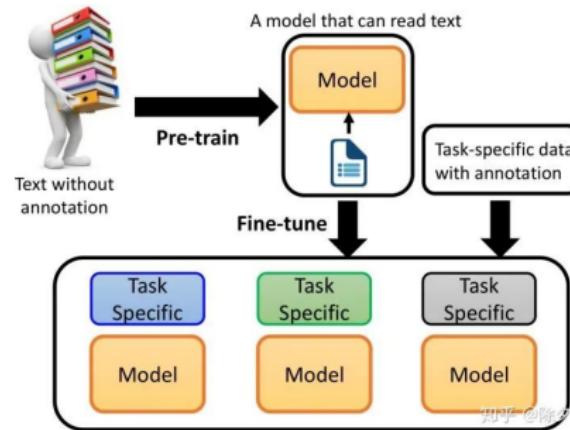
- Les réseaux de neurones récurrents (RNN) peuvent être utilisés pour former des **modèles de langage**.
 - Un modèle de langage calcule la **probabilité du prochain mot** dans une séquence de mots.
 - Il est utilisé dans la **génération de texte**, reconnaissance vocale, la traduction automatique, la correction automatique, etc.

S = Where are we going

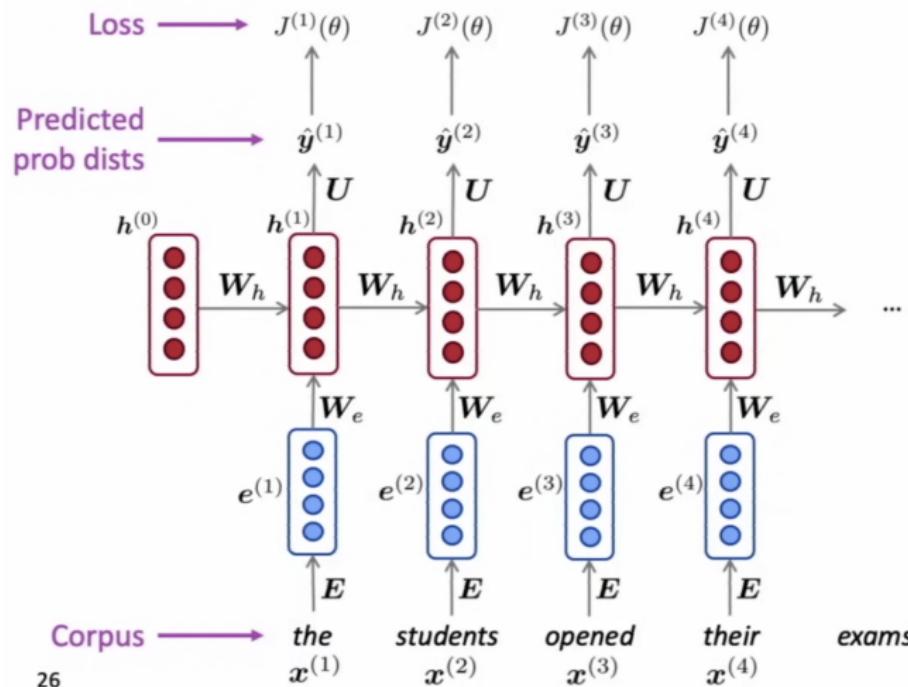
$$P(S) = P(\text{Where}) \times P(\text{are} | \text{Where}) \times P(\text{we} | \text{Where are}) \times P(\text{going} | \text{Where are we})$$

Formation d'un Modèle de Langage

- Le modèle est alimenté avec de nombreuses **séquences de texte** pour apprendre les **probabilités conditionnelles** entre les mots ⇒ Apprentissage Auto-supervisé.
 - Le processus d'entraînement consiste à ajuster les poids du RNN pour **maximiser la probabilité des mots réels observés** dans les données d'entraînement.



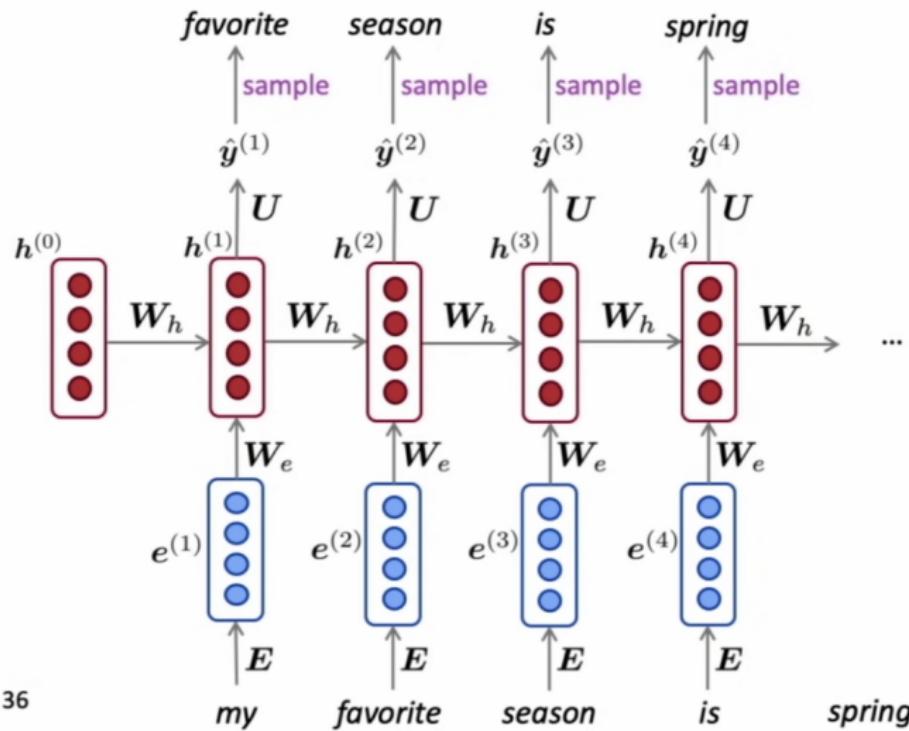
Modèle de Langage et RNN



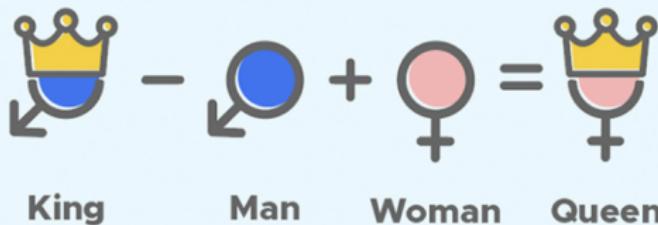
Génération de Texte avec un Modèle de Langage

- Un modèle de langage peut être utilisé pour **générer de nouvelles séquences de mots**.
- Le modèle prend un mot initial comme point de départ et **échantillonne** le mot suivant en fonction de ses probabilités conditionnelles apprises.
- Ce processus est **répété** pour générer des séquences de texte complètes.

Génération de Texte et RNN



Embedding - Propriétés



Google - Machine Learning Crash Course - Embeddings

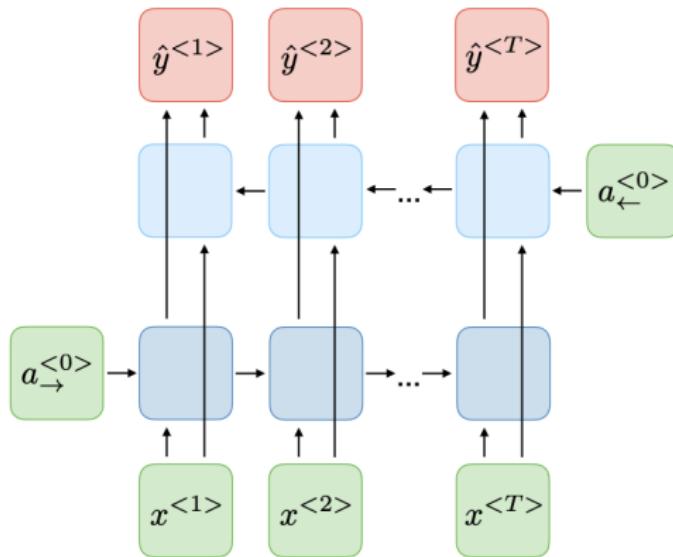
Limitations des Réseaux de Neurones Récurrents (RNN)

Bien que les RNN soient puissants pour traiter des séquences, ils présentent certaines limitations importantes :

- **Incapacité à considérer les entrées futures** pour la prise de décision.
- **Difficulté d'accès à l'information très ancienne**, ce qui limite leur capacité à traiter des dépendances à long terme.

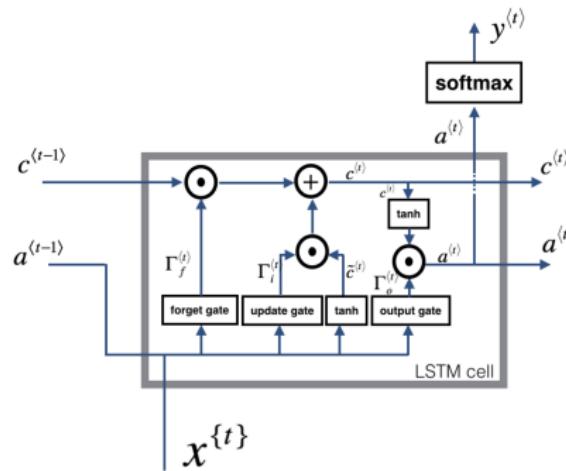
Ces limitations ont conduit au développement de variantes de RNN, telles que les **RNN Bi-directionnels** (BRNN) et les réseaux **Long Short-Term Memory** (LSTM).

Réseaux de neurones récurrents bi-directionnels (BRNN)



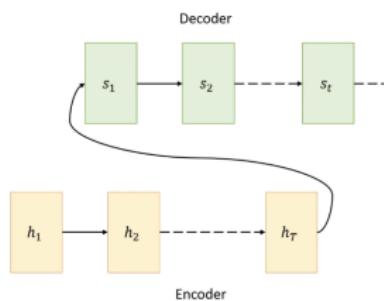
Réseaux LSTM (Long Short-Term Memory)

- Variante des RNN qui permet la modélisation séquentielle à long terme.
- Etat supplémentaire c de la cellule qui joue le rôle de la mémoire.

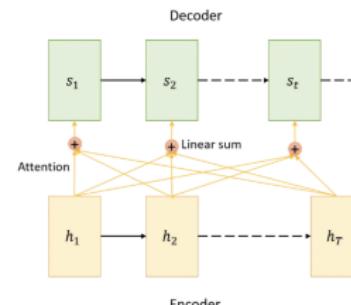


Modèles d'Attention

- Les modèles d'attention permettent au modèle de se **concentrer sur des parties spécifiques** de la séquence d'entrée lors de la prise de décision.
- Ils sont capables de capturer des **relations à longue distance** dans la séquence



(a) Vanilla Encoder Decoder Architecture



(b) Attention Mechanism

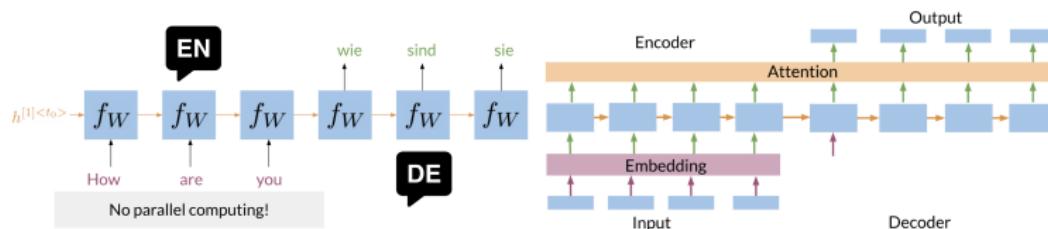
(Bahdanau, 2014; [Neural Machine Translation by Jointly Learning to Align and Translate](#))

Fonctionnement des Modèles d'Attention

- Un modèle d'attention attribue des **poids** aux différentes parties de la séquence d'entrée en fonction de leur pertinence pour la tâche.
- Ces poids sont calculés dynamiquement en fonction du contexte et de la tâche, ce qui permet au modèle de se concentrer sur les parties les plus importantes.

Les Transformers

- Les Transformers sont une **architecture** de réseau de neurones basée sur **l'attention**.
- Ils ont révolutionné le domaine du NLP depuis leur introduction en 2017.
- Les Transformers ont été à la base de nombreux modèles à succès, y compris GPT

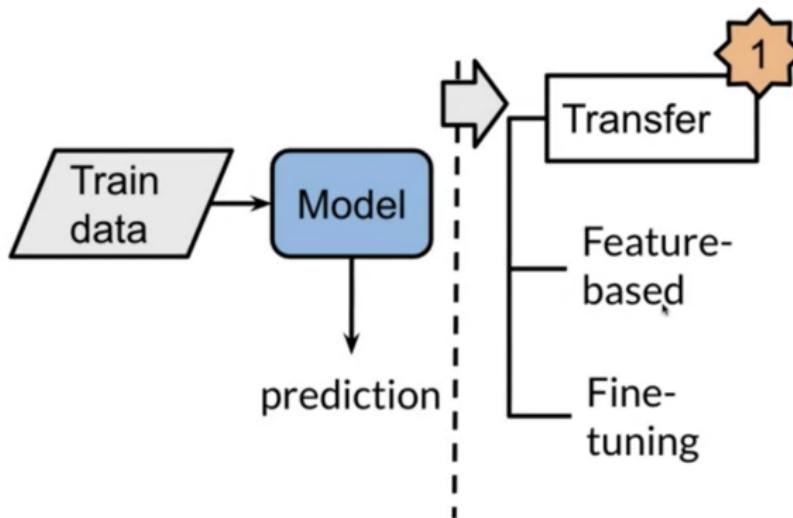


- 1- Deeplearning.ai - Short Course "Attention in Transformers"
- 2- Coursera - Natural Language Processing with Attention Models
- 3- Attention is all you need; Attentional Neural Network Models — Łukasz Kaiser

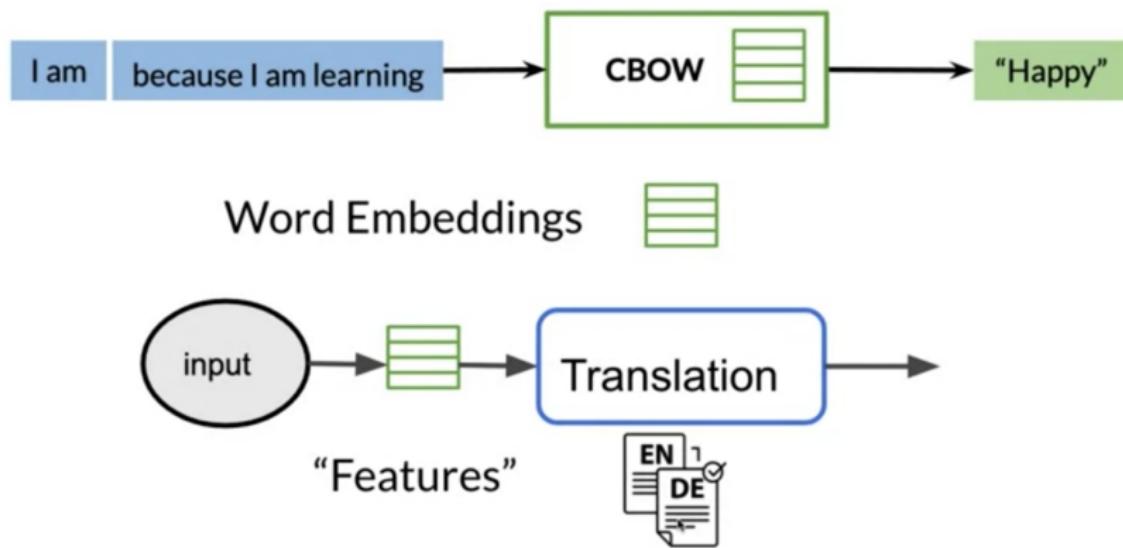
Transfert d'Apprentissage (Transfer Learning)

- Le transfert d'apprentissage implique l'adaptation d'un modèle pré-entraîné à un nouveau domaine ou à un nouveau problème.
- Les connaissances acquises en résolvant un problème sont appliquées à un problème différent mais lié.
- Le transfert d'apprentissage réduit le besoin de recommencer à zéro pour chaque nouveau problème.

Transfert d'Apprentissage - Two options

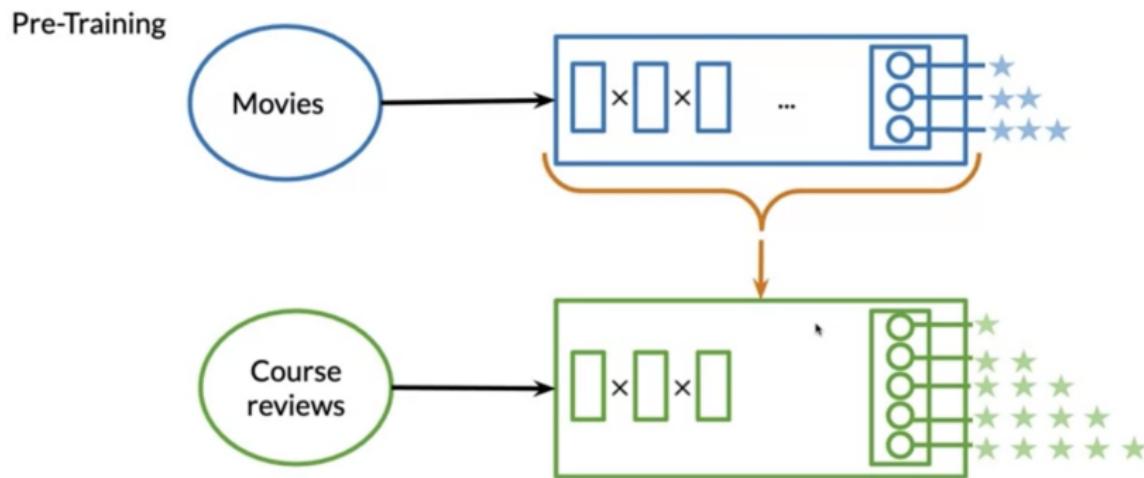


Transfert d'Apprentissage - Feature-based



Continuous bag of words (CBOW) in NLP - GeeksforGeeks

Transfert d'Apprentissage - Fine Tuning



Transfer Learning with Fine-tuning - GeeksforGeeks

Conclusion

- Réseaux de neurones: Du perceptron aux réseaux de neurones profonds
- Fonctions d'activation
- Feed-Forward
- Optimisation - Rétropropagation
- Architectures de Deep-Learning, e.g., Denses, Convolutionnels, Récurrents
- Transfert d'apprentissage

Travaux Pratiques

- Word embedding et traduction automatique de mot
- Réseau de neurones convolutionnel pour la classification d'images