



ÉCOLE  
**CENTRALE**LYON

# ÉCOLE CENTRALE LYON

## RAPPORT

# Deep Learning et Réseaux de Neurones

### *Élèves :*

Chaymae DAHHASSI  
Iheb HARHOURI  
Loqmen ANANI

### *Enseignant :*

Alexandre SAIDI

28 avril 2024

## Table des matières

<b>1 Introduction</b>	<b>3</b>
<b>2 Fondements théoriques des réseaux de neurones</b>	<b>4</b>
2.1 Historique et évolution des réseaux de neurones artificiels . . . . .	4
2.2 Structure de base d'un neurone artificiel . . . . .	5
2.3 Les réseaux de neurones multicouches . . . . .	6
2.4 Types d'architectures de réseaux de neurones . . . . .	7
2.4.1 Réseaux de neurones convolutifs (CNN) : . . . . .	7
2.4.2 Réseaux de neurones récurrents (RNN) : . . . . .	10
2.4.3 Réseaux de neurones récurrents à mémoire à court et long terme (LSTM) : . . . . .	12
2.4.4 LSTM Vs RNN . . . . .	17
2.4.5 Les Réseaux Gated Recurrent Unit (GRU) . . . . .	17
<b>3 Principes du Deep Learning</b>	<b>19</b>
3.1 Définition du Deep Learning . . . . .	19
3.2 Différences entre le Deep Learning et le Machine Learning traditionnel . . . . .	19
3.3 Avantages et inconvénients du Deep Learning . . . . .	20
<b>4 Entraînement des réseaux de neurones</b>	<b>21</b>
4.1 Fonctionnement de l'apprentissage supervisé et non supervisé . . . . .	21
4.2 Techniques d'optimisation des poids . . . . .	24
4.2.1 Descente de Gradient . . . . .	25
4.2.2 Rétropropagation (Backpropagation) . . . . .	27
4.3 Le surapprentissage . . . . .	28
4.3.1 Définition . . . . .	28
4.3.2 Pourquoi se produit-il ? . . . . .	28
4.3.3 Méthodes pour l'éviter . . . . .	29
<b>5 Applications du Deep Learning</b>	<b>29</b>
5.1 Vision par ordinateur . . . . .	29
5.2 Traitement du langage naturel . . . . .	31
5.3 Autres domaines : jeux vidéo, recommandation de contenu, santé . . . . .	32
<b>6 Recherche et tendances futures dans le Deep Learning</b>	<b>33</b>
6.1 Avancées récentes dans le domaine du Deep Learning . . . . .	33
6.2 Tendances émergentes et domaines de recherche prometteurs . . . . .	34
<b>7 Défis et limites du Deep Learning</b>	<b>36</b>
7.1 Problèmes liés à la taille des données et au temps de calcul . . . . .	36
7.2 Interprétabilité des modèles . . . . .	37
7.3 Éthique et questions de confidentialité . . . . .	37
<b>8 Étude de cas : Classification du cancer du sein par CNN</b>	<b>38</b>
8.1 Introduction et motivations . . . . .	38
8.2 Revue de littérature . . . . .	39
8.3 Jeu de données utilisé . . . . .	42

8.4	Architecture CNN proposée . . . . .	43
8.5	Code Python . . . . .	43
8.6	Résultats et performance . . . . .	44
8.7	Conclusion . . . . .	46
<b>9</b>	<b>Conclusion</b>	<b>48</b>

# 1 Introduction

L'intelligence artificielle (IA) a captivé l'imaginaire collectif depuis des décennies, alimentant à la fois l'excitation et l'appréhension quant à son potentiel et à son impact sur nos vies. Au cœur de cette révolution technologique se trouve le domaine du machine learning (ML), une branche de l'IA qui cherche à développer des algorithmes capables d'apprendre à partir des données et de prendre des décisions autonomes. Dans cette quête incessante vers des systèmes intelligents, le Deep Learning (DL) émerge comme une puissante sous-discipline du ML, révolutionnant la manière dont nous abordons les problèmes complexes de classification, de reconnaissance de motifs, de traitement du langage naturel et bien plus encore.[1]

Le DL se distingue par sa capacité à apprendre des représentations de données de plus en plus abstraites à partir de couches de neurones artificiels. Cette approche, inspirée par le fonctionnement du cerveau humain, permet aux réseaux neuronaux profonds d'analyser des données brutes avec une précision et une sophistication sans précédent. Alors que le ML traditionnel exige souvent des ingénieurs de concevoir manuellement des caractéristiques pertinentes pour résoudre un problème, le DL offre la possibilité d'apprendre automatiquement ces caractéristiques à partir des données elles-mêmes, réduisant ainsi la dépendance à l'expertise humaine.

Dans cette étude, nous plongeons dans l'univers fascinant du DL, explorant ses fondements théoriques, ses architectures de réseau, ses techniques d'entraînement et ses applications concrètes. Notre objectif est d'acquérir une compréhension approfondie des mécanismes sous-jacents du DL et de son potentiel à résoudre des problèmes du monde réel. En nous appuyant sur des exemples concrets et des cas d'utilisation variés, nous illustrons la puissance et la polyvalence du DL dans des domaines tels que la santé, la finance, l'automobile, la robotique et bien d'autres encore.

## 2 Fondements théoriques des réseaux de neurones

### 2.1 Historique et évolution des réseaux de neurones artificiels

L'IA est un domaine qui a été exploré depuis plusieurs décennies, connaissant des périodes de progrès et de stagnation en raison de contraintes techniques et financières. Depuis les années 50, de nombreux chercheurs se sont intéressés à la nature de l'intelligence et à sa représentation synthétique, cherchant à s'inspirer du fonctionnement cognitif du cerveau humain pour résoudre une variété de problèmes. La recherche en intelligence artificielle est étroitement liée aux neurosciences, progressant conjointement tout en étant distinctes, mais contribuant ensemble à l'avancement de la compréhension de l'intelligence.



FIGURE 1 – Le perceptron de Frank Rosenblatt (Getty Images)

C'est en 1957 que le premier modèle de neurone artificiel a été développé par Frank Rosenblatt. C'est l'un des premiers modèles de neurones artificiels et il a joué un rôle important dans le développement des réseaux de neurones et de l'IA en général.

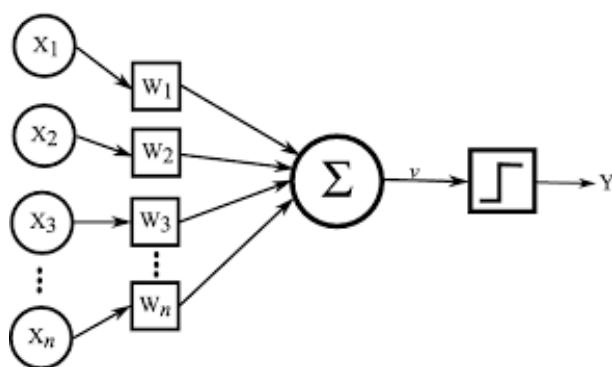


FIGURE 2 – Schéma du perceptron

Le perceptron est un algorithme d'apprentissage supervisé utilisé pour la classification binaire. Ses composantes comprennent :

- **Les entrées et les poids** : Chaque perceptron reçoit plusieurs entrées qui représentent généralement les caractéristiques des données à classer. Chaque entrée est associée à un poids, indiquant son importance relative pour la décision finale.

- **La fonction d'activation :** La somme pondérée des entrées passe par une fonction d'activation, souvent une fonction seuil, qui renvoie 1 si la somme pondérée dépasse un seuil donné, et 0 sinon.
- **Le biais :** Ce paramètre supplémentaire permet de décaler la fonction d'activation vers la gauche ou la droite, améliorant ainsi la flexibilité du modèle.
- **La sortie :** C'est le résultat final produit par le perceptron. Dans sa forme simple, la sortie est une classification binaire, généralement 0 ou 1.

Initialement, le perceptron initialise aléatoirement ses poids et son biais, qui servent de point de départ pour le processus d'apprentissage. Ensuite, les entrées, représentant les caractéristiques des données à classer, sont introduites dans le perceptron. La première opération réalisée par le perceptron sur ces entrées est une somme pondérée : pour chaque entrée, le perceptron multiplie l'entrée par son poids associé, puis somme ces produits. Le résultat de cette somme pondérée est ensuite soumis à une fonction d'activation, généralement une fonction seuil ou une fonction échelon unité. La sortie de cette fonction d'activation constitue la prédiction finale du perceptron. Pendant la phase d'apprentissage, cette prédiction est comparée à la vraie valeur pour calculer l'erreur. Si la prédiction est correcte, aucun ajustement n'est effectué. Cependant, si la prédiction est incorrecte, le perceptron ajuste ses poids et son biais dans le but de minimiser l'erreur lors des prédictions futures.

Bien que le perceptron de Rosenblatt ait été initialement considéré comme un progrès significatif dans le domaine de l'intelligence artificielle, ses capacités étaient limitées par sa structure simple. Il ne pouvait pas résoudre des problèmes non linéaires et a été rapidement supplanté par des modèles plus complexes, tels que les réseaux de neurones multicouches. Cependant, il trouve des applications variées dans plusieurs domaines clés à savoir le traitement de l'image et de la parole, la détection de spam (Entraîné sur un ensemble de courriels étiquetés, il apprend à classer les nouveaux courriels comme "spam" ou "non-spam" en se basant sur des caractéristiques spécifiques), la reconnaissance de caractères manuscrits...

## 2.2 Structure de base d'un neurone artificiel

En IA, l'objectif n'est pas de reproduire exactement le fonctionnement du cerveau humain, mais plutôt de s'en inspirer pour développer des solutions novatrices. Notre cerveau est constamment en processus d'apprentissage, influencé par les influx électriques échangés entre les neurones, qui dictent nos actions. Ces neurones sont activés en permanence par une multitude de signaux, internes et externes, tels que les **stimuli sensoriels** ou les besoins physiologiques comme la faim, et leurs connexions neuronales assurent la transmission de ces informations.

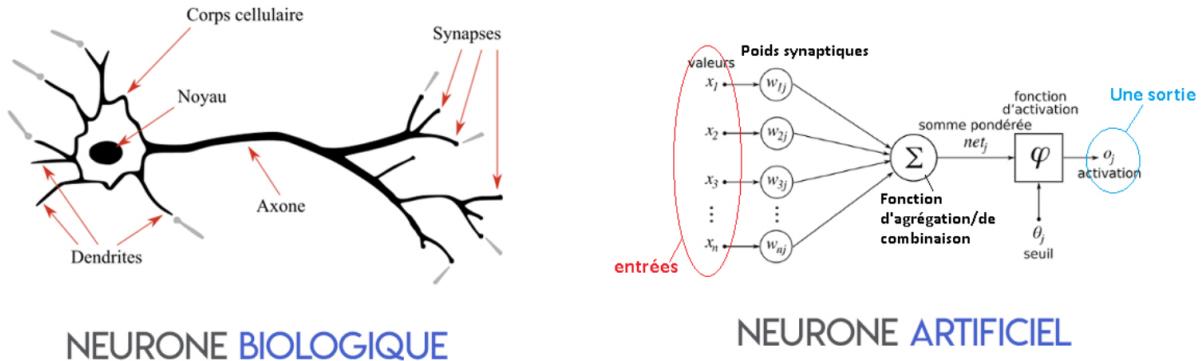


FIGURE 3 – Représentation d'un neurone biologique et artificiel

Le neurone humain est composée d'un corps cellulaire, d'un axone et de dendrites. Les dendrites reçoivent les signaux provenant d'autres neurones sous forme d'influx nerveux, tandis que l'axone transmet les signaux électriques à d'autres neurones. Le corps cellulaire traite ces signaux et détermine si le neurone doit s'activer et transmettre l'information.

En comparaison, le neurone artificiel est une abstraction mathématique inspirée du neurone biologique. Il est composé d'entrées, de poids associés à ces entrées, d'une fonction d'activation et d'une sortie. Les entrées représentent les données d'entrée, les poids déterminent l'importance relative de chaque entrée, la fonction d'activation modifie la sortie en fonction de la somme pondérée des entrées, et la sortie représente la prédiction ou la décision du neurone.[2]

Bien que les neurones humains et artificiels partagent certaines similitudes dans leur fonctionnement de base, les neurones artificiels sont simplifiés et conçus pour être traités par des algorithmes informatiques, tandis que les neurones humains sont incroyablement complexes et interagissent dans un réseau neuronal dynamique et adaptable.

### 2.3 Les réseaux de neurones multicouches

Les réseaux de neurones multicouches sont une classe de modèles de réseaux de neurones artificiels qui comprennent plusieurs couches de neurones, chacune étant connectée à la suivante. Contrairement aux perceptrons simples, qui ne comportent qu'une seule couche de neurones, les réseaux de neurones multicouches ont au moins une couche cachée située entre la couche d'entrée et la couche de sortie.

Chaque neurone dans une couche donnée est connecté à tous les neurones de la couche suivante par des poids, qui sont ajustés pendant l'apprentissage du réseau. L'information se propage à travers le réseau de neurones depuis la couche d'entrée, où les données sont introduites, jusqu'à la couche de sortie, qui produit les prédictions ou les classifications finales.

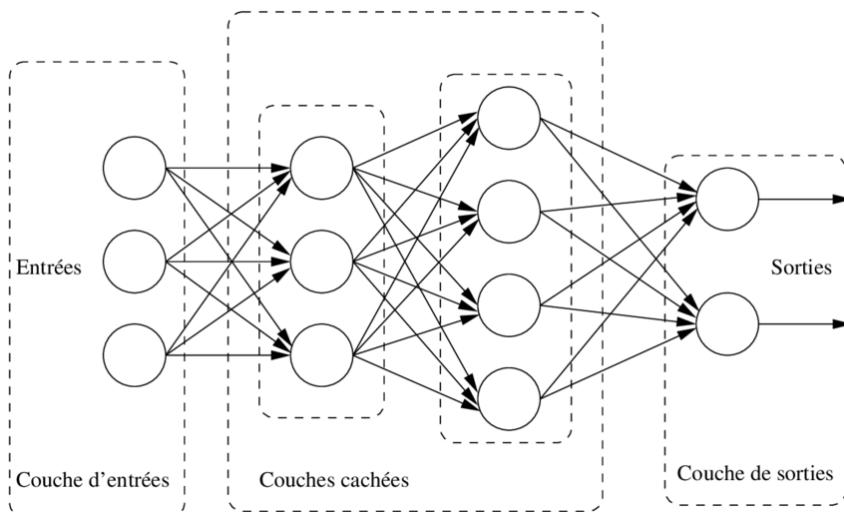


FIGURE 4 – Réseaux de neurones multicouches

Ces réseaux de neurones sont capables d'apprendre des représentations complexes et hiérarchiques des données, ce qui leur permet de modéliser des fonctions non linéaires et d'effectuer des tâches complexes telles que la reconnaissance d'images, la traduction automatique et la prédiction de séries chronologiques. Ils sont largement utilisés dans de nombreux domaines, notamment l'informatique, la finance, la médecine et la recherche.

## 2.4 Types d'architectures de réseaux de neurones

En plus du perceptron et des réseaux de neurones multicouches, il existe d'autres types de réseaux de neurones, chacun ayant ses propres caractéristiques et applications spécifiques :

### 2.4.1 Réseaux de neurones convolutifs (CNN) :

Les réseaux de convolution (LeCun, 1989), également connus sous le nom de réseaux neuronaux convolutifs ou CNN, sont la version étendue des réseaux de neurones artificiels (ANN) spécialement conçus pour le traitement de données présentant une topologie de type **grille connue**. Des exemples incluent les données séquentielles, qui peuvent être considérées comme une grille à une dimension prenant des échantillons à intervalles de temps réguliers, et les données d'image, qui peuvent être considérées comme une grille bidimensionnelle de pixels.[3]

Le terme "réseau neuronal convolutif" indique que le réseau utilise une opération mathématique appelée **convolution**. La convolution est une forme spécialisée d'opération linéaire. Les réseaux de convolution sont simplement des réseaux neuronaux qui utilisent la convolution à la place de la multiplication de matrices générale dans au moins l'une de leurs couches.

Les CNN sont spécialement conçus pour le traitement des images et des données spatiales. Ils utilisent des couches de convolution pour extraire des caractéristiques des

données d'entrée et sont largement utilisés dans la reconnaissance d'images, la détection d'objets et la classification d'images.

*Pour mieux comprendre la convolution :*

La convolution est une opération sur deux fonctions d'un argument à valeur réelle. Pour motiver la définition de la convolution, commençons par des exemples de deux fonctions que nous pourrions utiliser. Supposons que nous suivons la position d'un vaisseau spatial avec un capteur laser. Notre capteur laser fournit une seule sortie  $x(t)$ , la position du vaisseau spatial à l'instant  $t$ . Tant  $x$  que  $t$  sont des valeurs réelles, c'est-à-dire que nous pouvons obtenir une lecture différente du capteur laser à n'importe quel instant. Supposons maintenant que notre capteur laser soit quelque peu bruyant. Pour obtenir une estimation moins bruitée de la position du vaisseau spatial, nous aimeraisons faire la moyenne de plusieurs mesures. Bien sûr, les mesures plus récentes sont plus pertinentes, donc nous voudrons que ce soit une moyenne pondérée qui donne plus de poids aux mesures récentes. Nous pouvons le faire avec une fonction de pondération  $w(a)$ , où  $a$  est l'âge d'une mesure. Si nous appliquons une telle opération de moyenne pondérée à chaque instant, nous obtenons une nouvelle fonction  $s$  fournissant une estimation lissée de la position du vaisseau spatial :

$$s(t) = \int x(a)w(t-a)da.$$

Cette opération est appelée convolution. L'opération de convolution est généralement représentée par un astérisque :

$$s(t) = (x * w)(t).$$

Dans notre exemple,  $w$  doit être une fonction de densité de probabilité valide, sinon la sortie ne sera pas une moyenne pondérée. De plus,  $w$  doit être nul pour tous les arguments négatifs, sinon il regardera vers l'avenir, ce qui est vraisemblablement au-delà de nos capacités. Ces limitations sont propres à notre exemple. En général, la convolution est définie pour toutes les fonctions pour lesquelles l'intégrale ci-dessus est définie et peut être utilisée à d'autres fins que la prise de moyennes pondérées.

En terminologie de réseau de convolution, le premier argument (dans cet exemple, la fonction  $x$ ) de la convolution est souvent désigné comme l'entrée, et le deuxième argument (dans cet exemple, la fonction  $w$ ) comme le noyau ou le kernel. La sortie est parfois appelée la carte des caractéristiques (*Feature Map* en anglais).

*Architecture des CNNs :*

Le réseau neuronal convolutif se compose de plusieurs couches telles que la couche d'entrée, la couche convulsive, la couche de pooling et les couches entièrement connectées (Fully connected layers).

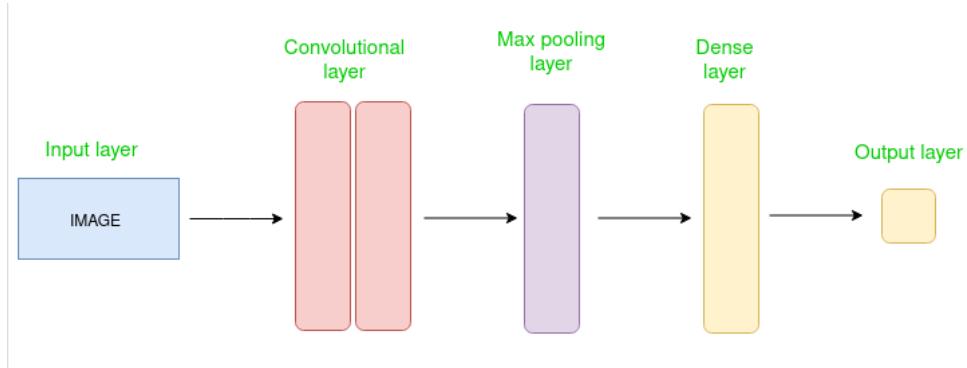


FIGURE 5 – Architecture CNN simple

La couche convulsive applique des filtres à l'image d'entrée pour extraire les caractéristiques, la couche Pooling sous-échantillonne l'image pour réduire les calculs et la couche entièrement connectée effectue la prédiction finale. Le réseau apprend les filtres optimaux par **rétropropagation** et **descente de gradient**, deux termes qui seront présentés et expliqués par la suite. [4]

*Couches utilisées pour créer des ConvNets :*

Considérons l'exécution d'un réseau de convolution sur une image de dimension  $32 \times 32 \times 3$ .

1. **Couches d'entrée** : Cette couche reçoit l'entrée brute de l'image, avec une largeur de 32, une hauteur de 32 et une profondeur de 3, représentant les canaux RGB.
2. **Couches convolutives** : Ces couches extraient des **caractéristiques** de l'image en appliquant des filtres apprenables sur les images d'entrée. Les filtres, souvent de formes  $2 \times 2$ ,  $3 \times 3$  ou  $5 \times 5$ , parcourrent l'image en calculant le produit scalaire avec le patch d'image correspondant. La sortie est appelée **cartes de fonctionnalités**, et avec par exemple 12 filtres, elle serait de dimension  $32 \times 32 \times 12$ .
3. **Couche d'activation** : Ajoutant de la non-linéarité au réseau, cette couche applique une fonction d'activation comme **ReLU**, **Tanh** ou **Leaky ReLU** sur la sortie de la couche précédente. Les dimensions de sortie restent inchangées, soit  $32 \times 32 \times 12$ . Une étude a montré que l'entraînement supervisé de réseaux neuro-naux très profonds est beaucoup plus rapide si les couches cachées sont composées de ReLU.[5]
4. **Couche de pooling** : Insérée périodiquement, elle réduit la taille du volume, accélérant les calculs et prévenant le surajustement. Les types communs sont le **pooling maximum** et le **pooling moyen**. Avec un pool max de  $2 \times 2$  filtres et une foulée de 2, la dimension résultante serait de  $16 \times 16 \times 12$ .

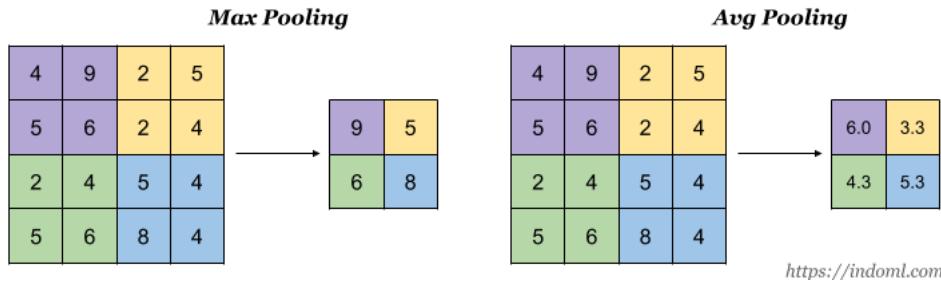


FIGURE 6 – Max Pooling et Average Pooling

5. **Aplatissement** : Les cartes de caractéristiques résultantes sont aplatis en un vecteur unidimensionnel après les couches de convolution et de regroupement.
6. **Couches entièrement connectées** : Elles reçoivent les entrées de la couche précédente et réalisent la classification ou la régression finale.
7. **Couche de sortie** : Les sorties des couches entièrement connectées sont introduites dans une fonction logistique, telle que sigmoïde ou softmax, pour convertir les scores en probabilités pour chaque classe dans le cas de la classification.

#### *Avantages des CNNs*

- Bons pour détecter des patterns et des caractéristiques dans les images, les vidéos et les signaux audio.
- Robustes à l'in-variance de translation, de rotation et de mise à l'échelle.
- Entraînement de bout en bout (*end-to-end training* en anglais), pas besoin d'extraction manuelle de fonctionnalités.
- Peuvent gérer de grandes quantités de données et atteindre une grande précision.

#### *Inconvénients des CNNs*

- L'entraînement est coûteux en calcul et nécessite beaucoup de mémoire.
- Peut être sujet au surajustement si suffisamment de données ou si une régularisation appropriée n'est pas utilisée.
- Nécessite de grandes quantités de données étiquetées.
- L'interprétabilité est limitée, il est difficile de comprendre ce que le réseau a appris.

#### 2.4.2 Réseaux de neurones récurrents (RNN) :

Les Réseaux de neurones récurrents (RNN) sont conçus pour traiter des données séquentielles, telles que des séquences de mots dans un texte ou des séquences temporelles dans des données chronologiques. Ils sont utilisés dans des applications telles que la traduction automatique, la génération de texte et la prédiction de séries chronologiques.

Dans un RNN, la sortie de l'étape précédente est introduite comme entrée de l'étape en cours. Dans les réseaux de neurones traditionnels, toutes les entrées et sorties sont indépendantes les unes des autres. Néanmoins, dans les cas où il est nécessaire de **prédir** le mot suivant d'une phrase par exemple, les mots précédents sont requis et il est donc nécessaire de se souvenir des mots précédents. C'est ainsi que RNN a vu le jour,

qui a résolu ce problème à l'aide d'une **couche cachée**. La caractéristique principale et la plus importante de RNN est son état caché , qui mémorise certaines informations sur une séquence. L'état est également appelé état mémoire (*Memory State* en anglais) car il mémorise l'entrée précédente sur le réseau. Il utilise les mêmes paramètres pour chaque entrée car il effectue la même tâche sur toutes les entrées ou couches cachées pour produire la sortie. Cela réduit la complexité des paramètres, contrairement aux autres réseaux de neurones.[6]

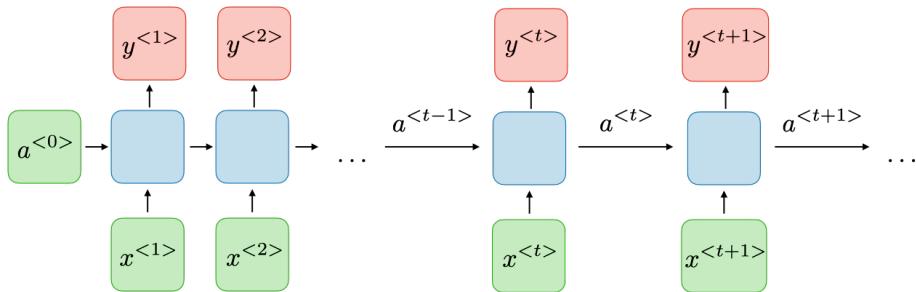


FIGURE 7 – Forme des RNNs

À l'instant  $t$ , l'activation  $a_t$  et la sortie  $y_t$  sont de la forme suivante :

$$a_t = g_1(W_{aa}a_{t-1} + W_{ax}x_t + b_a)$$

$$y_t = g_2(W_{ya}a_t + b_y)$$

où  $W_{ax}$ ,  $W_{aa}$ ,  $W_{ya}$ ,  $b_a$ ,  $b_y$  sont des coefficients indépendants du temps et où  $g_1$ ,  $g_2$  sont des fonctions d'activation.

#### Types de RNN

- **One to One** : Ce type de RNN se comporte de la même manière que n'importe quel réseau neuronal simple, il est également connu sous le nom de *Vanilla Neural Network*. Dans ce réseau neuronal, il n'y a qu'une seule entrée et une seule sortie.
- **One to Many** : Dans ce type de RNN, il y a une entrée et de nombreuses sorties qui lui sont associées.
- **Many to One** : Dans ce type de réseau, de nombreuses entrées sont introduites dans le réseau à plusieurs états du réseau, générant une seule sortie.
- **Many to Many** : Dans ce type de réseau de neurones, il existe plusieurs entrées et plusieurs sorties correspondant à un problème. Un exemple de ce problème sera la traduction linguistique. Dans la traduction linguistique, nous fournissons plusieurs mots d'une langue en entrée et prédisons plusieurs mots de la deuxième langue en sortie.

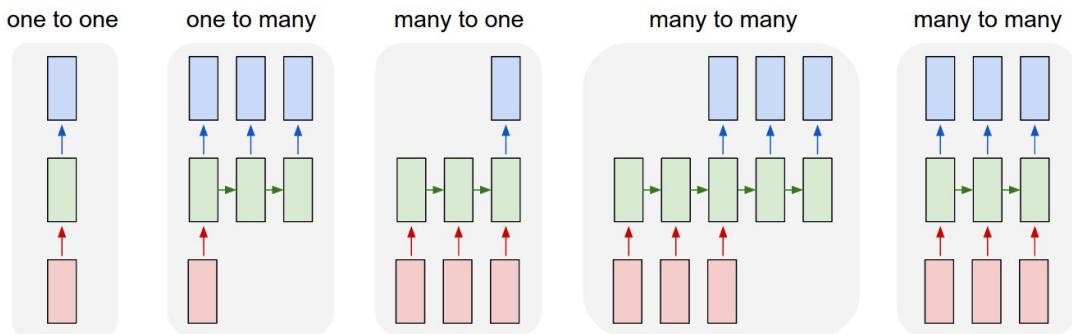


FIGURE 8 – Types de RNN

#### *Avantages des RNNs*

- La taille du modèle n'augmente pas avec la taille de l'entrée.
- Les coefficients sont indépendants du temps.
- Possibilité de prendre en compte des entrées de toute taille.
- Les calculs prennent en compte les informations antérieures.

#### *Inconvénients des RNNs*

- Difficile d'accéder à des informations d'un passé lointain.
- Impossible de prendre en compte des informations futures d'un état donné.
- Temps de calcul long.

### 2.4.3 Réseaux de neurones récurrents à mémoire à court et long terme (LSTM) :

Les LSTMs sont une version améliorée des RNNs conçus par Hochreiter & Schmidhuber. Les LSTMs sont bien adaptés aux tâches de prédiction de séquence et excellent dans la capture des dépendances à long terme [7]. Leurs applications s'étendent aux tâches impliquant des séries temporelles et des séquences. La force des LSTMs réside dans leur capacité à saisir la dépendance d'ordre cruciale pour résoudre des problèmes complexes, tels que la traduction automatique et la reconnaissance vocale.

#### *Architecture et fonctionnement du LSTM*

L'architecture LSTM possède une structure en chaîne qui contient quatre réseaux de neurones et différents blocs de mémoire appelés cellules.

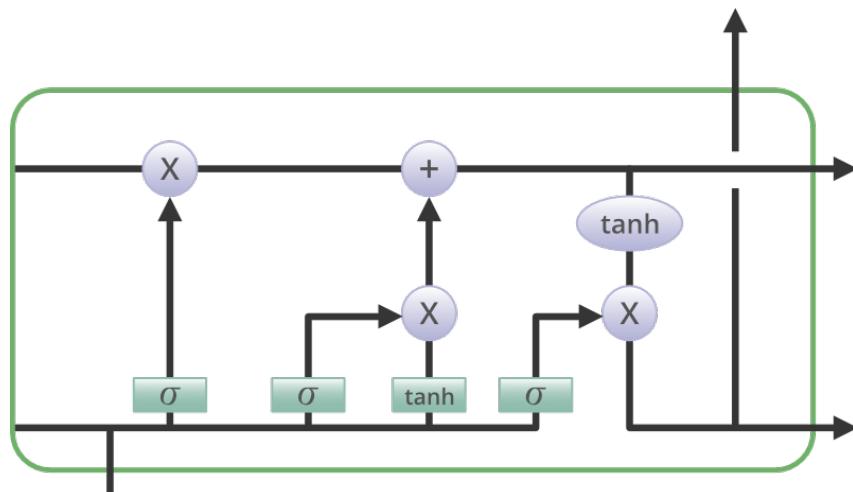


FIGURE 9 – Architecture dun LSTM

L’information est conservée par les cellules et les manipulations de la mémoire sont effectuées par les portes. On distingue trois types de portes :

**Porte d’oubli :**

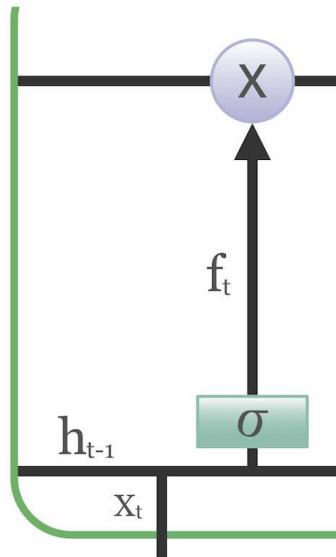


FIGURE 10 – La porte d’oubli d’un LSTM

L’information qui n’est plus utile dans l’état de la cellule est supprimée avec la porte d’oubli (*Forget gate* en anglais). Deux entrées  $x_t$  (entrée au temps particulier) et  $h_{t-1}$  (sortie de la cellule précédente) sont alimentées à la porte et multipliées par des matrices de poids suivies de l’addition d’un biais. Le résultat est ensuite passé à travers une fonction d’activation qui donne une sortie binaire. Si pour un état de cellule particulier la sortie est 0, la pièce d’information est oubliée et pour la sortie 1, l’information est conservée pour une utilisation future. L’équation pour la porte d’oubli est :

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$

où :

$W_f$  représente la matrice de poids associée à la porte d'oubli.

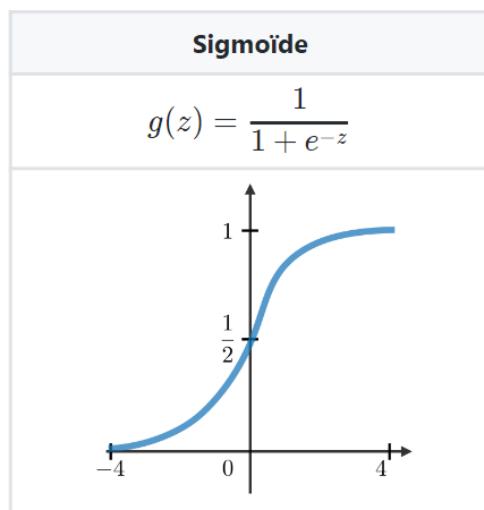
$[h_{t-1}, x_t]$  représente la concaténation de l'entrée actuelle et de l'état caché précédent.

$b_f$  est le biais avec la porte d'oubli.

$\sigma$  est la fonction d'activation **sigmoïde**.

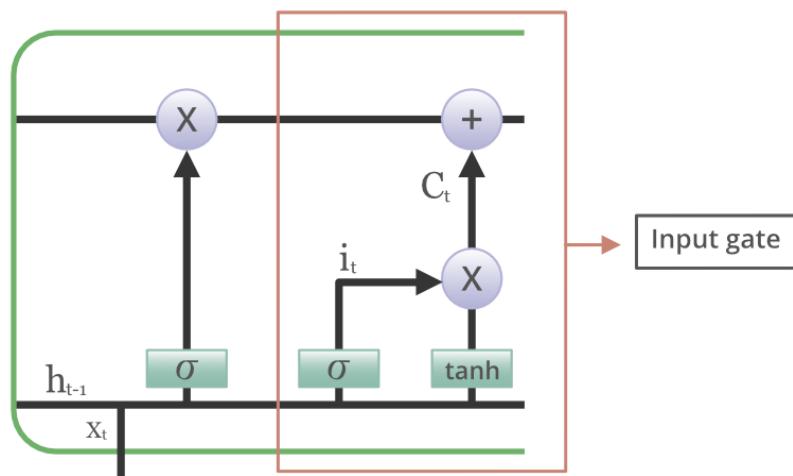
*La fonction sigmoïde :*

La fonction d'activation sigmoïde transforme les valeurs d'entrée en valeurs dans l'intervalle  $(0, 1)$ . Elle est définie comme suit :



Cette fonction est également appelée fonction logistique. Elle est utilisée pour introduire de la non-linéarité dans les réseaux de neurones et est couramment utilisée dans les couches cachées des réseaux de neurones artificiels.

### Porte d'entrée



L'ajout d'informations utiles à l'état de la cellule est effectué par la porte d'entrée (*input gate* en anglais). Tout d'abord, l'information est régulée à l'aide de la fonction sig-

moïde et filtre les valeurs à mémoriser de manière similaire à la porte d'oubli en utilisant les entrées  $h_{t-1}$  et  $x_t$ . Ensuite, un vecteur est créé à l'aide de la fonction **tangente hyperbolique** qui donne une sortie de -1 à +1, contenant toutes les valeurs possibles de  $h_{t-1}$  et  $x_t$ . Enfin, les valeurs du vecteur et les valeurs régulées sont multipliées pour obtenir les informations utiles. L'équation de la porte d'entrée est la suivante :

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad \hat{C}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c)$$

Nous multiplions l'état précédent par  $f_t$ , en ignorant l'information que nous avions précédemment choisie d'ignorer. Ensuite, nous incluons  $i_t \cdot \hat{C}_t$ . Cela représente les valeurs de candidat mises à jour, ajustées en fonction de la quantité que nous avons choisie de mettre à jour pour chaque valeur d'état.

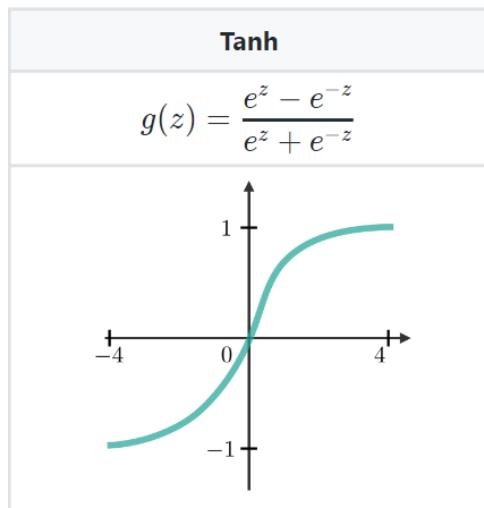
$$C_t = f_t \odot C_{t-1} + i_t \odot \hat{C}_t$$

où

$\odot$  représente la multiplication élément par élément tanh est la fonction d'activation tangente hyperbolique.

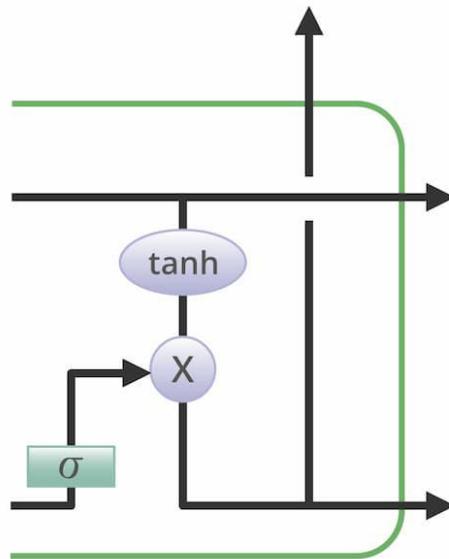
*la fonction tangente hyperbolique*

Elle est similaire à la fonction sigmoïde mais a une plage de valeurs allant de -1 à 1, ce qui la rend centrée autour de zéro.



En plus d'être utilisée dans les couches cachées des réseaux de neurones, la fonction tangente hyperbolique est également utilisée comme fonction d'activation dans les couches de sortie pour des tâches telles que la régression où des valeurs continues sont nécessaires.

### Porte de sortie



La porte de sortie (*output gate* en anglais) est responsable d'extraire des informations utiles de l'état cellulaire actuel pour les présenter en tant que sortie. Tout d'abord, un vecteur est généré en appliquant la fonction tangente hyperbolique sur la cellule. Ensuite, les informations sont régulées en utilisant la fonction sigmoïde et filtrées par les valeurs à retenir en utilisant les entrées  $h_{t-1}$  et  $x_t$ . Enfin, les valeurs du vecteur et les valeurs régulées sont multipliées pour être envoyées en tant que sortie et entrée à la cellule suivante. L'équation pour la porte de sortie est :

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$$

#### 2.4.4 LSTM Vs RNN

Caractéristique	LSTM	RNN
Mémoire	Possède une unité de mémoire spéciale lui permettant d'apprendre les dépendances à long terme dans les données séquentielles	Ne possède pas d'unité de mémoire
Directionnalité	Peut être entraîné à traiter des données séquentielles dans les deux sens, avant et arrière	Peut seulement être entraîné à traiter des données séquentielles dans un seul sens
Entraînement	Plus difficile à entraîner que les RNN en raison de la complexité des portes et de l'unité de mémoire	Plus facile à entraîner que les LSTM
Apprentissage de dépendances à long terme	Oui	Limité
Capacité à apprendre des données séquentielles	Oui	Oui
Applications	Traduction automatique, reconnaissance vocale, résumé de texte, traitement du langage naturel, prévision de séries temporelles	Traitement du langage naturel, traduction automatique, reconnaissance vocale, traitement d'images, traitement vidéo

TABLE 1 – Comparaison entre LSTM et RNN

#### 2.4.5 Les Réseaux Gated Recurrent Unit (GRU)

Les Réseaux Gated Recurrent Unit (GRU) sont un type de réseau neuronal récurrent (RNN) qui a été introduit par Cho et al. en 2014 comme une alternative plus simple aux réseaux de mémoire à court et long terme (LSTM). Comme les LSTM, les GRU peuvent traiter des données séquentielles telles que du texte, de la parole et des séries temporelles.

L'idée de base derrière les GRU est d'utiliser des mécanismes de portes pour mettre à jour sélectivement l'état caché du réseau à chaque pas de temps. Les mécanismes de portes sont utilisés pour contrôler le flux d'informations dans et hors du réseau. Les GRU ont deux mécanismes de portes, appelés la porte de réinitialisation et la porte de mise à jour.

La porte de réinitialisation détermine quelle partie de l'état caché précédent doit être oubliée, tandis que la porte de mise à jour détermine quelle partie de la nouvelle entrée doit être utilisée pour mettre à jour l'état caché. La sortie du GRU est calculée en fonction de l'état caché mis à jour.

Les équations utilisées pour calculer la porte de réinitialisation, la porte de mise à jour et l'état caché d'un GRU sont les suivantes [8] :

$$\begin{aligned} r_t &= \sigma(W_r \cdot [h_{t-1}, x_t] + b_r) \\ z_t &= \sigma(W_z \cdot [h_{t-1}, x_t] + b_z) \\ \tilde{h}_t &= \tanh(W \cdot [r_t \odot h_{t-1}, x_t] + b) \\ h_t &= (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t \end{aligned}$$

Où :

- $r_t$  est la porte de réinitialisation,
- $z_t$  est la porte de mise à jour,
- $\tilde{h}_t$  est l'état caché mis à jour temporairement,
- $h_t$  est l'état caché final,
- $W_r$ ,  $W_z$ , et  $W$  sont les matrices de poids,
- $b_r$ ,  $b_z$ , et  $b$  sont les biais,
- $\sigma$  est la fonction sigmoïde, et
- $\odot$  représente la multiplication élément par élément.

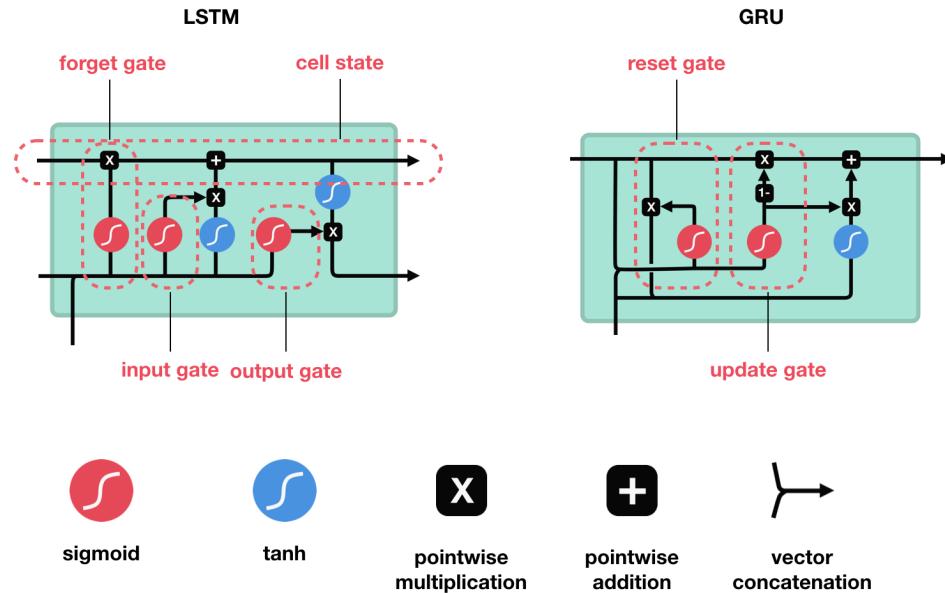


FIGURE 11 – Différence entre l'architecture d'un LSTM et d'un GRU

En résumé, les réseaux GRU sont un type de RNN qui utilisent des mécanismes de portes pour mettre à jour sélectivement l'état caché à chaque pas de temps, ce qui leur permet de modéliser efficacement des données séquentielles. Ils se sont révélés efficaces dans diverses tâches de traitement du langage naturel, telles que la modélisation de langage, la traduction automatique et la reconnaissance vocale.

## 3 Principes du Deep Learning

### 3.1 Définition du Deep Learning

Le Deep Learning (apprentissage profond en français), est un domaine en plein essor de l'intelligence artificielle (IA) qui s'inspire du fonctionnement du cerveau humain pour apprendre à partir de données [9]. Il utilise des réseaux de neurones artificiels (RNA) composés de plusieurs couches cachées, permettant de traiter des informations complexes et de réaliser des tâches de haut niveau [10].

Les réseaux de Deep Learning sont capables de capturer des relations non linéaires complexes entre les données, ce qui les rend particulièrement adaptés à des tâches telles que la reconnaissance d'images, la traduction automatique et le traitement du langage naturel [11]. On peut voir un exemple démonstratif avec la figure 12 dans laquelle les couches cachées permettent de mieux exploiter les parties de l'animal pour déterminer l'espèce.

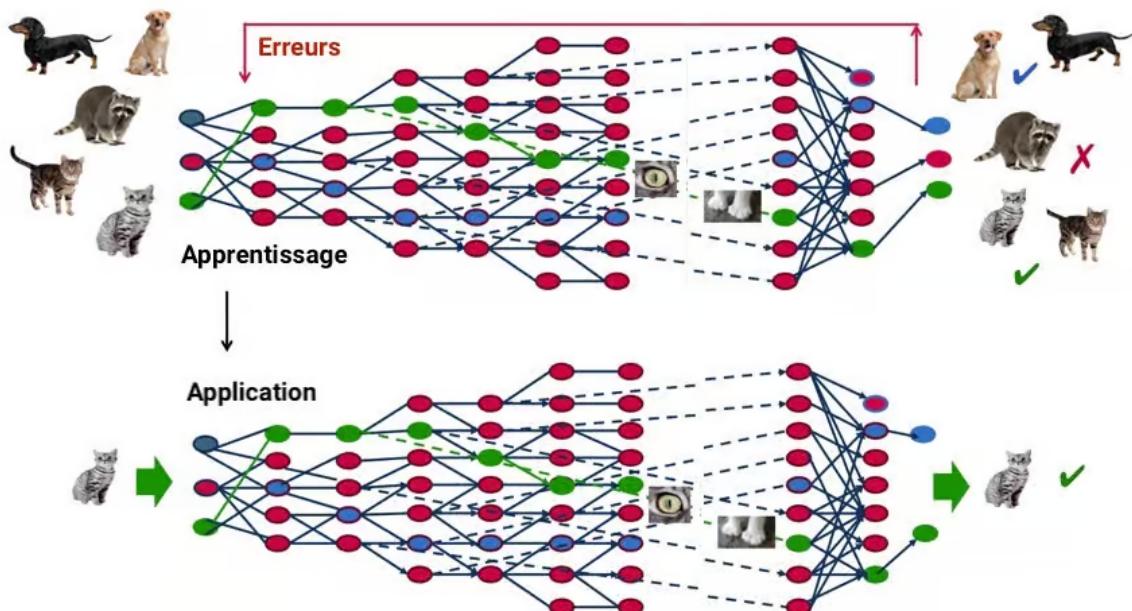


FIGURE 12 – Exemple conceptuel de l'application du deep learning à la reconnaissance d'image [12]

### 3.2 Différences entre le Deep Learning et le Machine Learning traditionnel

Contrairement aux algorithmes de Machine Learning traditionnels qui s'appuient sur des règles et des caractéristiques définies manuellement, les réseaux Deep Learning apprennent de manière automatique à partir de données brutes [10] (voir figure 13).

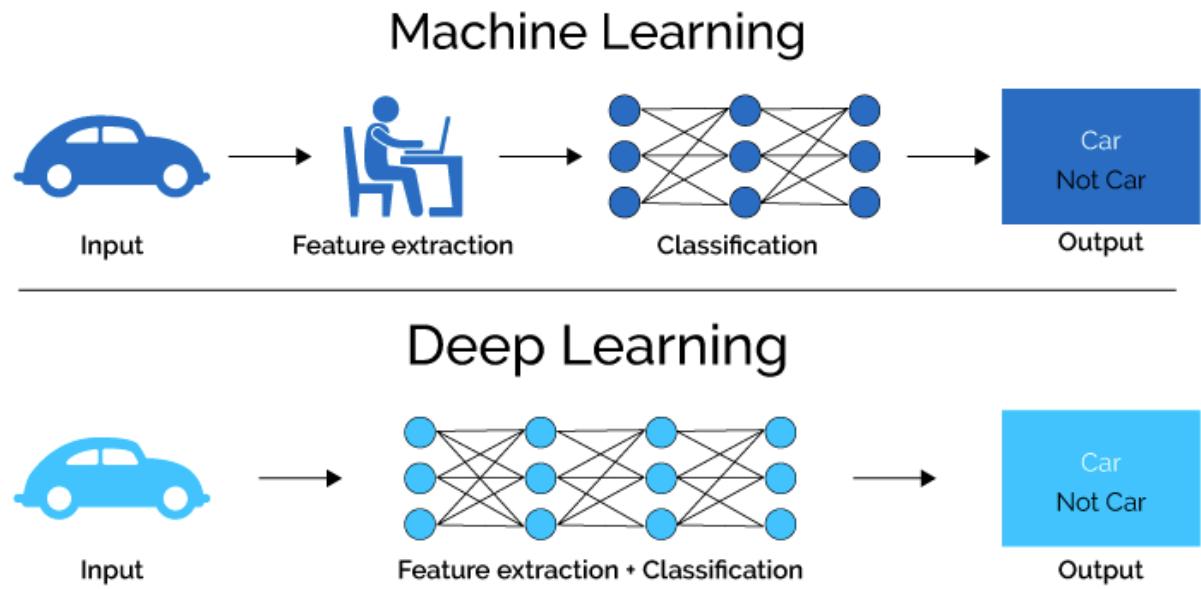


FIGURE 13 – Différence schématique entre le Deep Learning et le Machine Learning traditionnel [13].

D'autre part, le Deep Learning se distingue du Machine Learning traditionnel par plusieurs aspects clés :

- **Profondeur des réseaux** : Les réseaux de neurones du Deep Learning possèdent plusieurs couches cachées, ce qui leur permet d'extraire des informations plus riches et plus abstraites des données [14].
- **Apprentissage automatique** : Les réseaux Deep Learning apprennent automatiquement à partir de données brutes, sans nécessiter d'ingénierie de caractéristiques manuelle [10].
- **Grande quantité de données** : L'apprentissage des réseaux Deep Learning nécessite de grandes quantités de données pour s'entraîner efficacement, souvent plus que le machine learning traditionnel [10].
- **Tâches complexes** : Le Deep Learning est particulièrement adapté aux tâches complexes qui impliquent des données non structurées et des relations non linéaires [11].

### 3.3 Avantages et inconvénients du Deep Learning

Le Deep Learning présente de nombreux avantages qui ont révolutionné le domaine de l'intelligence artificielle :

- **Haute performance** : Le Deep Learning a permis d'obtenir des résultats de pointe dans de nombreux domaines, surpassant souvent les approches traditionnelles du Machine Learning [14].
- **Apprentissage automatique** : L'apprentissage automatique des réseaux Deep Learning permet de gagner du temps et d'éviter des tâches manuelles fastidieuses liées à l'ingénierie de caractéristiques [10].
- **Flexibilité** : Les réseaux Deep Learning peuvent être adaptés à une grande variété de tâches et de problèmes grâce à leur architecture modulaire et à leur capacité à apprendre à partir de données diverses [10].

Cependant, le Deep Learning présente également certains inconvénients qu'il est important de prendre en compte :

- **Besoin de données** : La grande quantité de données nécessaire à l'entraînement des réseaux Deep Learning peut constituer un obstacle pour certains projets [10].
- **Boîte noire** : Le fonctionnement interne des réseaux Deep Learning peut être difficile à comprendre, ce qui peut limiter leur interprétation.

## 4 Entraînement des réseaux de neurones

### 4.1 Fonctionnement de l'apprentissage supervisé et non supervisé

L'apprentissage supervisé se définit par la présence d'un objectif d'apprentissage clairement établi, spécifié par une cible précise à prédire. Cet objectif se concrétise par un ensemble de  $n$  paires d'entrées  $X_t$  et des cibles correspondantes  $Y_t$ . Habituellement, cet ensemble est constitué en soumettant un ensemble d'entrées à un groupe d'individus qui assignent à chaque entrée une cible adaptée, en fonction du problème spécifique à résoudre. La fonction principale d'un algorithme d'apprentissage dans ce contexte est de développer un modèle capable de reproduire ce processus de labellisation humain, c'est-à-dire de prédire la cible  $y$  pour une nouvelle entrée  $x$ , de la manière dont un humain l'aurait déterminée. Toutefois, les applications des algorithmes d'apprentissage ne se limitent pas uniquement à la simulation des comportements humains ; ils peuvent également servir à modéliser les relations entre des paires d'entrées et de cibles issues d'autres phénomènes, comme par exemple la relation entre une action et son prix sur le marché boursier. La nature de l'ensemble  $Y$  d'où proviennent les cibles dépendra du type de problème à résoudre. Deux types de problèmes fréquents dont les problèmes de **classification** et de **régression**. [15]

- **La régression** est utilisée lorsque la sortie à prédire peut prendre des valeurs continues, il s'agit d'une variable réelle. Exemples :
- **Prédiction du poids** : dans ce cas  $x \in X$  correspond au caractéristiques physiques d'une personnes( âge, sexe, ...) et  $y \in Y$  correspond à son poids.
- **Prédiction de la consommation électrique d'une installation** : dans ce cas  $x \in X$  correspond aux données météorologiques et aux données sur l'installation ( température, humidité, type de l'installation, ...) et  $y \in Y$  correspond à la consommation de l'installation.

La figure 14 illustre un problème de régression en une dimension. La figure à gauche présente le cas où la prédiction est produite par un modèle linéaire et celle à droite présente le cas où la régression est produite par un modèle quadratique.

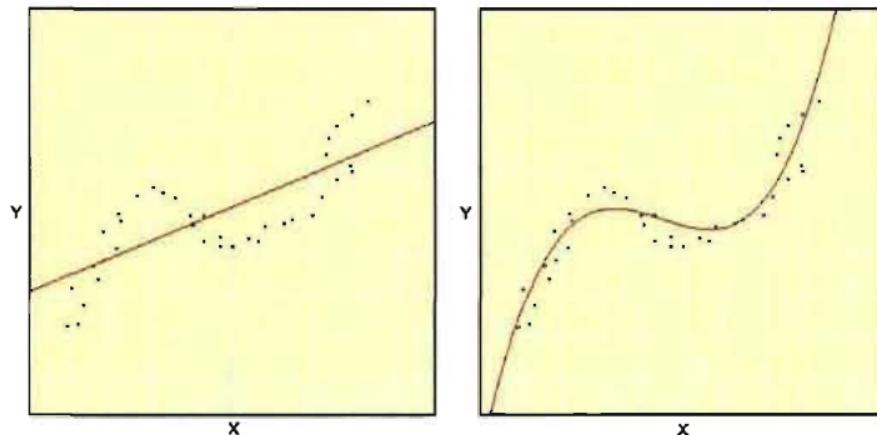


FIGURE 14 – Régression en une dimension [15]

Pour évaluer la performance d'un modèle de régression, on calcule généralement la différence au carré entre la sortie du modèle et la cible attendue.

- **La classification** est une tâche consistant à choisir une classe (valeur) parmi toutes celles possibles. Y correspond donc à l'ensemble de classes possibles auxquelles peuvent appartenir les différentes entrées  $x \in X$ . Exemples : Un algorithme prédisant le chiffre manuscrit sur l'image d'entrée ou un algorithme classifiant une tumeur comme « bénigne » ou « maligne ». La figure 15 présente un ensemble de données étiquetées où chaque image d'entrée est associée avec la prédiction voulue.

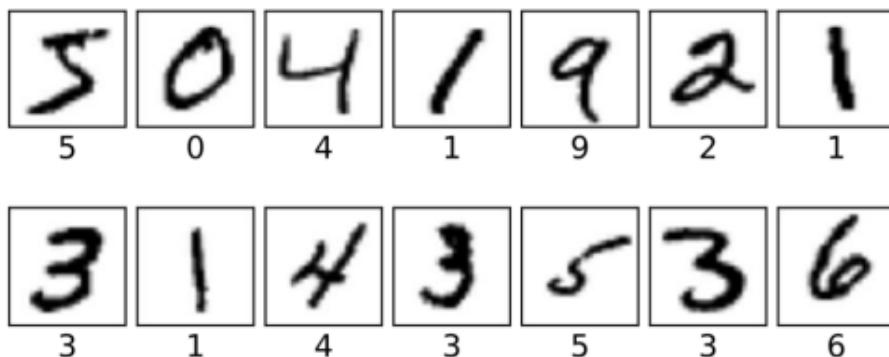


FIGURE 15 – Classification : Ensemble de données étiquetées [16]

Pour évaluer la performance d'un modèle de classification, on fait généralement appel au taux d'erreur de classification, qui représente la proportion des cas où la sortie prédite par le modèle diffère de la valeur cible attendue..

On note qu'on plus de ces deux familles (régression et classification), il existe des algorithmes qui peuvent faire des prédictions parmi des vecteurs de grandes dimensions en combinant la classification et la régression. Ces types d'algorithmes sont couramment utilisés dans le traitement automatique du langage, où les mots peuvent être représentés par des vecteurs de grande dimension, résultant de combinaisons de lettres. [16]

L'apprentissage non supervisé est une branche de l'apprentissage automatique qui vise à identifier des structures cachées dans des données non étiquetées. Contrairement à l'apprentissage supervisé où les données d'entrée sont associées à des étiquettes ou des réponses spécifiques, l'apprentissage non supervisé travaille sur des données sans annotations préalables, ce qui le rend utile dans des situations où les annotations sont coûteuses ou impossibles à obtenir.

Les algorithmes d'apprentissage non supervisé s'appuient sur divers principes et techniques mathématiques pour extraire des informations des données. Parmi ces techniques, on trouve :

- **1. Analyse en composantes principales (ACP)** : Cette technique réduit la dimensionnalité des données en transformant les variables originales en un nombre plus petit de variables non corrélées appelées composantes principales. Mathématiquement, l'ACP cherche à résoudre le problème suivant :

$$\max_{\|u\|=1} \{u^T \Sigma u\}$$

où  $\Sigma$  est la matrice de covariance des données et  $u$  est le vecteur des poids de la composante principale. L'objectif est de maximiser la variance des données projetées sur  $u$ .

- **2. Clustering (regroupement)** : Il s'agit de partitionner l'ensemble de données en groupes (ou clusters) tels que les données dans chaque groupe soient plus similaires entre elles qu'avec celles des autres groupes. Le k-means est l'un des algorithmes de clustering les plus utilisés. Il minimise la somme des distances au carré entre les points et le centroïde de leur cluster attribué. La fonction objective est :

$$\min_S \sum_{i=1}^k \sum_{x \in S_i} \|x - \mu_i\|^2$$

où  $S_i$  sont les clusters et  $\mu_i$  sont les centres des clusters.

Parmi les algorithmes clés en apprentissage non supervisé on trouve :

- **1. K-means** : K-means est un algorithme de clustering qui partitionne les données en  $K$  clusters distincts en minimisant la variance intra-cluster. L'algorithme fonctionne en quatre étapes principales :
    - Initialisation : Choix aléatoire de  $K$  points comme centres de clusters initiaux (centroïdes).
    - Attribution : Chaque point de donnée est assigné au cluster dont le centroïde est le plus proche.
    - Mise à jour : Les centroïdes de chaque cluster sont recalculés en prenant la moyenne de tous les points attribués au cluster.
    - Répétition : Les étapes d'attribution et de mise à jour sont répétées jusqu'à convergence, généralement lorsque les affectations de cluster ne changent plus.
- L'algorithme cherche à minimiser la fonction objective suivante :

$$J = \sum_{i=1}^K \sum_{x \in S_i} \|x - \mu_i\|^2$$

où  $\mu_i$  est le centroïde du cluster  $S_i$ , et  $\|x - \mu_i\|^2$  est le carré de la distance euclidienne entre le point  $x$  et le centroïde  $\mu_i$ . [17]

- **2. t-SNE (t-distributed Stochastic Neighbor Embedding)** : t-SNE est une technique puissante pour la visualisation de données à haute dimension en réduisant les dimensions tout en préservant les structures locales des données. Il convertit les distances entre les points en probabilités conditionnelles qui représentent des similarités.

Les probabilités dans l'espace d'origine sont définies de telle sorte que les points similaires ont une forte probabilité de sélection, et les points non similaires une faible probabilité. Dans l'espace de faible dimension, t-SNE minimise la divergence de Kullback-Leibler entre les distributions de probabilités jointes dans les espaces original et réduit. Mathématiquement, cette minimisation est effectuée via une technique de descente de gradient. [17]

- **3. DBSCAN (Density-Based Spatial Clustering of Applications with Noise)** : DBSCAN est un algorithme de clustering basé sur la densité qui peut identifier des clusters de formes variées en considérant les densités locales des points. L'algorithme classe chaque point comme un point de base, un point de bordure ou un bruit, en fonction de la densité des voisins à proximité.

DBSCAN définit un rayon  $\epsilon$  et un seuil minimum de points  $MinPts$ . Un point est considéré comme un point central si au moins  $MinPts$  points se trouvent à l'intérieur d'un rayon  $\epsilon$  de ce point. Les clusters sont formés par la connexion de points centraux qui sont accessibles les uns aux autres par des points denses, et les points de bordure sont ceux qui sont à portée de  $\epsilon$  d'un point central mais qui n'ont pas suffisamment de voisins pour être eux-mêmes des points centraux.

## 4.2 Techniques d'optimisation des poids

Dans un réseau de neurones, chaque connexion entre deux neurones est associée à un poids. Ce poids modifie l'amplitude du signal transmis entre ces neurones. Lorsqu'un réseau est formé sur un jeu de données, ces poids sont ajustés pour minimiser l'erreur entre la sortie prédite du réseau et les valeurs réelles observées. Les poids sont donc essentiels pour la précision des prédictions du modèle.

L'optimisation des poids est essentielle pour plusieurs raisons :

- **Réduction de l'erreur** : Les poids optimisés permettent de réduire l'erreur de prédiction, ce qui améliore la performance du modèle sur des données nouvelles et non vues pendant l'entraînement.
- **Éviter le surajustement** : Une bonne stratégie d'optimisation aide à prévenir le surajustement, où le modèle apprend les détails et le bruit dans les données d'entraînement à un tel point qu'il échoue à généraliser à de nouvelles données.
- **Efficacité computationnelle** : Des techniques d'optimisation efficaces peuvent réduire le temps nécessaire pour entraîner un modèle, ce qui est important lors de l'utilisation de grands ensembles de données.

Les principales techniques d'optimisation de poids utilisées sont :

### 4.2.1 Descente de Gradient

La recherche de l'amélioration et du développement de nouveaux algorithmes d'optimisation a joué un rôle important dans l'amélioration des architectures d'apprentissage profond. Les réseaux de neurones sont, dans une très large mesure, un problème d'optimisation qui cherche à trouver l'optimum global à travers une trajectoire d'entraînement robuste et une convergence rapide en utilisant des algorithmes de descente de gradient. Le problème d'optimisation consiste à trouver des valeurs de paramètres de modèle qui sont cohérentes avec les informations préalables et qui offrent le meilleur ajustement avec l'erreur de prédiction la plus faible.

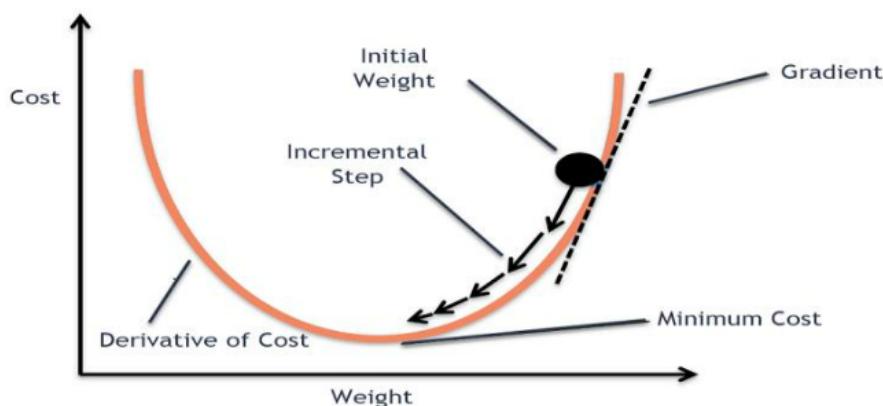


FIGURE 16 – Descente de Gradient [18]

Mathématiquement, le problème d'optimisation peut s'écrire de la façon suivante :

Soit  $f : R^n \rightarrow R$ , trouver  $\hat{x} = \operatorname{argmin} f(\mathbf{x}), x \in R^n$

$f$  est appelée la fonction objectif ou fonction de coût et  $x$  est le minimiseur de la fonction objectif  $f$ .

La descente de gradient est une méthode pour minimiser une fonction objectif  $f(\mathbf{x})$  paramétrée par les paramètres  $\mathbf{x} \in R$  en mettant à jour les paramètres dans la direction opposée au gradient de la fonction objectif  $\nabla f(x)$ . Si on considère  $\mathbf{X}$  comme l'ensemble des paramètres du modèle et  $f$  comme la fonction de coût, la mise à jour des poids se fait selon la règle :

$$\mathbf{X} = \mathbf{X} - \alpha \nabla f(\mathbf{X}) , \text{ où } \alpha \text{ est le taux d'apprentissage}$$

En général, il existe trois variantes de la descente de gradient : 1) La descente de gradient par lot (**Batch Gradient Descent, BGD**) qui utilise l'ensemble des données pour l'entraînement en même temps, puis ajuste les poids, ce qui en fait une approche déterministe, 2) La descente de gradient stochastique (**Stochastic Gradient Descent, SGD**) – utilise une seule donnée de l'ensemble d'entraînement à la fois (une ligne après l'autre) et ajuste les poids à chaque itération pour chaque ligne, et 3) un hybride de BGD et SGD appelé descente de gradient par mini-lots (**Mini-batch Gradient Descent**) qui utilise plus d'un exemple d'entraînement à la fois [14]. Elles diffèrent selon la quantité

de données utilisées pour calculer le gradient de la fonction objectif, faisant généralement un compromis entre la précision de la mise à jour des paramètres et le temps nécessaire pour effectuer la mise à jour [19]. La BGD fonctionne bien dans des scénarios convexes où trouver un minimum local assure d'atteindre un minimum global, cependant, les scénarios hautement non convexes sont courants pour les réseaux de neurones et le défi est d'éviter de rester piégé dans des minima locaux sous-optimaux. Pour cette raison et d'autres, telles que le choix et l'ajustement des taux d'apprentissage, des algorithmes d'optimisation de la descente de gradient ont été développés pour l'apprentissage profond afin de pallier les défis associés à la descente de gradient. Parmi ces algorithmes on peut citer :

- **Momentum** Une technique d'optimisation courante est la descente de gradient stochastique (SGD), mais lors de l'entraînement de l'algorithme, le temps d'exécution est relativement élevé. Le **Momentum** est conçu pour apprendre rapidement, surtout en présence de grandes courbures, de gradients petits mais bruyants ou de gradients stables. L'utilisation de cet algorithme est une approche qui peut aider le réseau à éliminer les minima locaux. Parfois le gradient ne pointe pas vers le minimum, et les étapes successives de la descente de gradient oscillent d'un côté à l'autre, avançant donc très lentement vers le minimum. La figure 17 ainsi illustre comment l'incorporation du **momentum**, en amortissant ces oscillations, tend à favoriser la convergence vers le minimum.

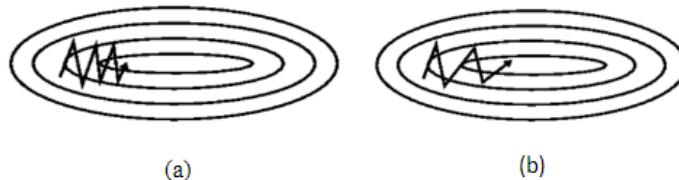


FIGURE 17 – (a) SGD sans Momentum (b) SGD avec Momentum [18]

- **Descente de Gradient Adaptative (AdaGrad)** Ceci est une méthode qui sélectionne le taux d'apprentissage selon la situation [18]. Comme le taux est calculé à partir des paramètres, les taux d'apprentissage ont tendance à s'adapter. Un gradient avec des paramètres plus élevés aura un taux d'apprentissage réduit et vice versa . La théorie d'AdaGrad est similaire à celle de l'algorithme AdaDelta en ce qu'elle mesure différents taux d'apprentissage pour différents éléments de paramètres. Cependant, elle utilise l'agrégation des carrés des gradients : contrairement à AdaDelta, elle utilise la moyenne mobile des carrés des gradients. [18]
- **Adaptive Delta (AdaDelta)** AdaDelta est une extension de AdaGrad. AdaDelta fonctionne en utilisant plusieurs fenêtres de taille fixe au lieu d'accumuler les gradients. Il surveillera uniquement les gradients disponibles dans la fenêtre [18]. Étant donné que l'algorithme SGD nécessite une sélection manuelle du taux d'apprentissage, un taux d'apprentissage inapproprié pourrait conduire à une faible précision de prédiction. Cependant, en tant qu'optimisation de l'algorithme SGD, l'Adadelta peut ajuster automatiquement le taux d'apprentissage et augmenter la précision des prédictions.

### 4.2.2 Rétropropagation (Backpropagation)

La rétropropagation, ou backpropagation, est une technique utilisée dans l'apprentissage supervisé pour permettre aux réseaux de neurones d'apprendre à partir de leurs erreurs. En pratique, elle permet de mettre à jour les poids des connexions du réseau de manière itérative afin de minimiser la différence entre les sorties prédictes et les sorties attendues.

#### Fonction de Coût

La première étape dans la rétropropagation est de définir une fonction de coût  $C$ , qui mesure l'erreur entre la sortie prédictée du réseau  $y_{pred}$  et la sortie réelle attendue  $y_{vrai}$ . Un exemple courant de fonction de coût est l'erreur quadratique moyenne :

$$C = \frac{1}{2} \sum (y_{pred} - y_{vrai})^2$$

#### Propagation Avant

Durant la propagation avant, les entrées du réseau sont passées à travers les différentes couches pour produire une sortie. Pour chaque couche  $l$ , l'activation  $a^{(l)}$  est calculée à partir de l'activation de la couche précédente  $a^{(l-1)}$  comme suit :

$$a^{(l)} = \sigma(W^{(l)}a^{(l-1)} + b^{(l)})$$

où  $W^{(l)}$  et  $b^{(l)}$  sont respectivement les poids et les biais de la couche  $l$ , et  $\sigma$  est une fonction d'activation non-linéaire, comme la fonction sigmoïde. [20]

#### Calcul du Gradient

Le but de la rétropropagation est de calculer le gradient de la fonction de coût par rapport à chaque poids et biais dans le réseau. Ce calcul se fait en deux étapes principales : la propagation arrière de l'erreur et le calcul des gradients.

**1. Propagation de l'Erreur :** L'erreur est calculée à la sortie du réseau et propagée vers l'arrière à travers chaque couche. L'erreur delta  $\delta^{(l)}$  pour la couche  $l$  est calculée comme : Pour la dernière couche,

$$\delta^{(l)} = \frac{\partial C}{\partial a^{(l)}} \sigma'(z^{(l)})$$

Pour les couches cachées,

$$\delta^{(l)} = ((W^{(l+1)})^T \delta^{(l+1)}). \sigma'(z^{(l)})$$

où  $z^{(l)} = W^{(l)}a^{(l-1)} + b^{(l)}$  et  $\sigma'$  est la dérivée de la fonction d'activation.

**2. Calcul des Gradients :** Les gradients des poids et des biais sont calculés en utilisant les erreurs delta [20] :

$$\frac{\partial C}{\partial W^{(l)}} = \delta^{(l)}(a^{(l-1)})^T, \quad \frac{\partial C}{\partial b^{(l)}} = \delta^{(l)}$$

### Mise à jour des Poids

Les poids sont ensuite mis à jour en utilisant un algorithme d'optimisation comme la descente de gradient [20], où  $\alpha$  est le taux d'apprentissage :

$$W^{(l)} = W^{(l)} - \alpha \frac{\partial C}{\partial W^{(l)}}, b^{(l)} = b^{(l)} - \alpha \frac{\partial C}{\partial b^{(l)}}$$

## 4.3 Le surapprentissage

### 4.3.1 Définition

Le surapprentissage (*overfitting* en anglais) survient lorsqu'un modèle d'apprentissage automatique est trop ajusté aux données d'entraînement spécifiques sur lesquelles il a été formé, au point où il devient inefficace pour généraliser et faire des prédictions précises sur de nouvelles données. Cela se produit souvent lorsque le modèle devient trop complexe par rapport à la taille et à la diversité des données d'entraînement, ce qui lui permet de mémoriser les données plutôt que de les comprendre. En conséquence, le modèle peut présenter une performance apparemment élevée sur les données d'entraînement, mais lorsqu'il est confronté à des données qu'il n'a jamais vues auparavant, il peut avoir du mal à généraliser ses prédictions.

Le surapprentissage peut être problématique car il compromet la capacité du modèle à prendre des décisions précises sur de nouvelles données, ce qui limite son utilité dans des applications réelles. Pour atténuer le surapprentissage, les data scientists utilisent des techniques telles que la régularisation, qui ajoutent des contraintes supplémentaires au modèle pour l'empêcher de devenir trop complexe, et la validation croisée, qui évalue les performances du modèle sur des ensembles de données distincts pour vérifier sa capacité de généralisation. En identifiant et en traitant le surapprentissage, les modèles d'apprentissage automatique peuvent devenir plus robustes et fiables dans leur capacité à faire des prédictions précises sur de nouvelles données.

### 4.3.2 Pourquoi se produit-il ?

Vous ne pouvez obtenir des prévisions précises que si le modèle de machine learning parvient à généraliser à tous les types de données de son domaine. Le surapprentissage survient lorsque le modèle ne peut pas se généraliser et s'ajuste trop étroitement aux données d'entraînement. Plusieurs facteurs peuvent contribuer au surapprentissage, notamment :

- La taille des données d'entraînement est insuffisante et ne représente pas de manière exhaustive toutes les valeurs possibles des données d'entrée.
- Les données d'entraînement contiennent des éléments non pertinents en grande quantité, souvent appelés données bruitées.
- Le modèle est entraîné pendant une durée excessive sur un ensemble de données d'entraînement spécifique.
- La complexité du modèle est trop élevée, ce qui l'amène à apprendre le bruit présent dans les données d'entraînement.

### 4.3.3 Méthodes pour l'éviter

Pour éviter le surapprentissage, différentes stratégies de science des données peuvent être utilisées pour diversifier et adapter le jeu de données d'entraînement. Voici quelques-unes de ces stratégies :

#### 1. Arrêt anticipé :

L'arrêt anticipé consiste à interrompre la phase d'entraînement avant que le modèle ne puisse apprendre le bruit des données. Il est crucial de choisir le bon moment pour arrêter l'entraînement afin d'obtenir des résultats précis.

#### 2. Élagage :

Le processus d'élagage identifie les caractéristiques ou paramètres ayant le plus d'impact sur les prédictions finales du modèle. En sélectionnant les caractéristiques les plus importantes et en éliminant les moins pertinentes, le modèle peut être amélioré. Par exemple, dans la prédiction d'images, la forme du visage peut être priorisée par rapport à d'autres caractéristiques telles que la forme des yeux.

#### 3. Régularisation :

La régularisation comprend diverses techniques visant à réduire le surapprentissage en éliminant les facteurs ayant un impact négligeable sur les résultats des prédictions. Par exemple, elle peut appliquer des pénalités à certaines caractéristiques pour réduire leur influence sur les prédictions. Parmi les techniques de régularisation on trouve : le **Dropout**, qui permet de réduire l'overfitting lors de l'entraînement du modèle. Le terme » Dropout » fait référence à la suppression de neurones dans les couches d'un modèle de Deep Learning. En fait, on désactive temporairement certains neurones dans le réseau, ainsi que toutes ses connexions entrantes et sortantes. **Le choix des neurones à désactiver est aléatoire.**

*Pour plus d'information sur le Dropout, ce papier de recherche l'explique en détail :)*

— Research paper about Dropout

#### 4. Assemblage :

L'assemblage combine les prédictions de plusieurs algorithmes distincts pour obtenir des résultats plus précis. En utilisant des méthodes telles que le bagging et le boosting, les prédictions des "apprenants faibles" sont combinées pour améliorer la précision globale du modèle.

#### 5. Augmentation des données :

Cette technique modifie légèrement les exemples de données à chaque fois qu'ils sont traités par le modèle. Par des transformations comme la translation, le retournement et la rotation, l'augmentation des données permet de rendre chaque échantillon d'entraînement unique, évitant ainsi au modèle d'apprendre des caractéristiques spécifiques à un ensemble de données particulier.

## 5 Applications du Deep Learning

### 5.1 Vision par ordinateur

Depuis le début des années 2000, les réseaux convolutifs, ou ConvNets, ont été appliqués avec un grand succès dans la détection, la segmentation et la reconnaissance d'objets

dans les images. Ces applications ont principalement concerné des domaines où les données étaient abondantes, comme la reconnaissance des panneaux de signalisation, la segmentation d'images biologiques, ainsi que la détection de visages, de texte, de piétons et de corps humains dans les images. Un succès pratique récent notable des ConvNets concerne la reconnaissance faciale.

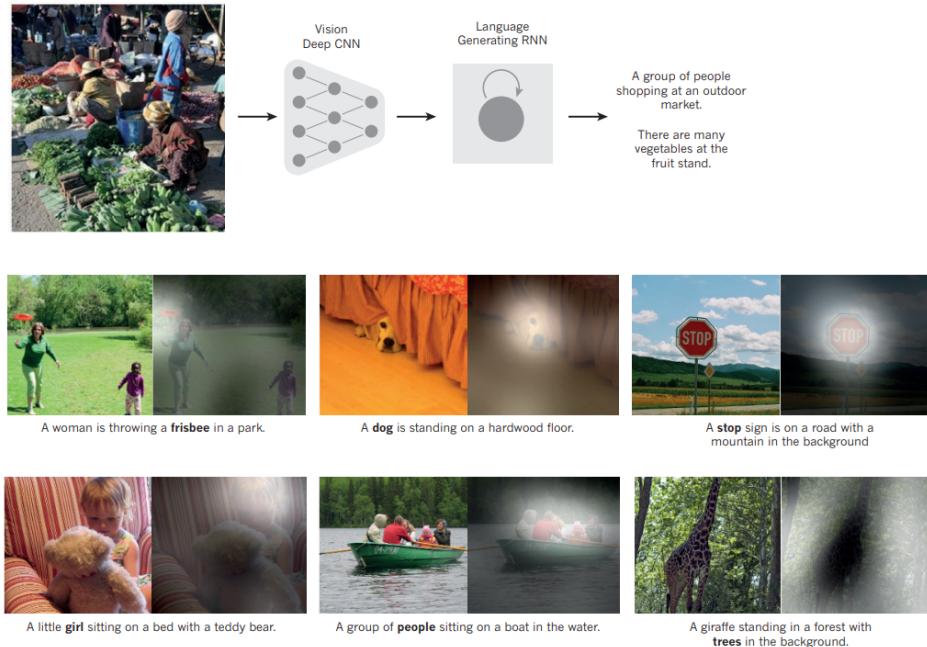


FIGURE 18 – Légendes générées par un réseau neuronal récurrent (RNN) qui prend comme entrée supplémentaire la représentation extraite par un réseau neuronal convolutif profond (CNN) à partir d'une image de test [10]

Malgré ces succès, les ConvNets ont été largement délaissés par les communautés de vision par ordinateur et d'apprentissage automatique jusqu'à la compétition ImageNet en 2012. Lorsque des réseaux convolutionnels profonds ont été appliqués à un ensemble de données d'environ un million d'images du web, réparties en 1 000 classes différentes, ils ont obtenu des résultats spectaculaires, réduisant presque de moitié les taux d'erreur des meilleures approches concurrentes. Ce succès est dû à l'utilisation efficace des GPU, des ReLUs, d'une nouvelle technique de régularisation appelée « dropout », et des techniques pour générer plus d'exemples d'entraînement en déformant les existants. Cette réussite a provoqué une révolution dans le domaine de la vision par ordinateur ; les ConvNets sont maintenant l'approche dominante pour presque toutes les tâches de reconnaissance et de détection, et atteignent des performances comparables à celles des humains dans certains domaines.

Les architectures récentes de ConvNet comprennent de 10 à 20 couches de ReLUs, des centaines de millions de poids et des milliards de connexions entre les unités. Alors que l'entraînement de tels réseaux pouvait prendre des semaines il y a seulement deux ans, les progrès de matériel, de logiciel et des algorithmes ont réduit les temps d'entraînement à quelques heures.

La performance des systèmes de vision basés sur les ConvNets a poussé ainsi la plupart des

grandes entreprises technologiques, y compris Google, Facebook, Microsoft, IBM, Yahoo!, Twitter et Adobe, ainsi qu'un nombre croissant de start-ups, à initier des projets de recherche et de développement et à déployer des produits et des services de compréhension d'images basés sur les ConvNets.

## 5.2 Traitement du langage naturel

Les réseaux profonds possèdent des avantages distincts par rapport aux algorithmes d'apprentissage classiques qui ne font pas usage de représentations distribuées. Ces avantages découlent du pouvoir de la composition et dépendent de la structure composante appropriée de la distribution génératrice des données. Premièrement, l'apprentissage des représentations distribuées permet une généralisation à de nouvelles combinaisons des valeurs des caractéristiques apprises au-delà de celles observées pendant l'entraînement (par exemple,  $2^n$  combinaisons sont possibles avec  $n$  caractéristiques binaires).

Les couches cachées d'un réseau neuronal multicouche apprennent à représenter les entrées du réseau de manière à faciliter la prédiction des sorties cibles. Cela est bien démontré par l'entraînement d'un réseau neuronal multicouche pour prédire le mot suivant dans une séquence à partir d'un contexte local de mots précédents. Chaque mot du contexte est présenté au réseau sous forme de vecteur "one-of-N" [10], c'est-à-dire qu'un composant est à 1 et les autres sont à 0. Dans la première couche, chaque mot crée un modèle différent d'activations, ou vecteurs de mots. Dans un modèle de langue, les autres couches du réseau apprennent à convertir les vecteurs de mots entrants en un vecteur de mot de sortie pour le mot suivant prédict, ce qui peut être utilisé pour prédire la probabilité de n'importe quel mot du vocabulaire d'apparaître comme le mot suivant. Le réseau apprend des vecteurs de mots contenant de nombreux composants actifs, chacun pouvant être interprété comme une caractéristique distincte du mot, comme cela a été démontré pour la première fois dans le contexte de l'apprentissage de représentations distribuées pour des symboles. Ces caractéristiques sémantiques n'étaient pas explicitement présentes dans l'entrée. Elles ont été découvertes par la procédure d'apprentissage comme un bon moyen de factoriser les relations structurées entre les symboles d'entrée et de sortie en de multiples « micro-règles ». [10] L'apprentissage des vecteurs de mots s'est également avéré très efficace lorsque les séquences de mots proviennent d'un large corpus de texte réel et que les micro-règles individuelles sont peu fiables. Lorsqu'il est entraîné à prédire le mot suivant dans un article de journal, par exemple, les vecteurs de mots appris pour « mardi » et « mercredi » sont très similaires, tout comme les vecteurs de mots pour « Suède » et « Norvège ». Ces représentations sont appelées représentations distribuées parce que leurs éléments (les caractéristiques) ne sont pas mutuellement exclusifs et leurs nombreuses configurations correspondent aux variations observées dans les données. Ces vecteurs de mots sont composés de caractéristiques apprises qui n'ont pas été déterminées à l'avance par des experts, mais découvertes automatiquement par le réseau neuronal. Les représentations vectorielles des mots apprises à partir de textes sont maintenant très largement utilisées dans les applications de langage naturel.

Dans la figure 19, on a à gauche une illustration des représentations de mots apprises pour la modélisation du langage, projetées de manière non linéaire en 2D à l'aide de l'algorithme t-SNE.[10] À droite, une représentation en 2D des phrases apprises par un réseau neuronal récurrent encodeur-décodeur anglais-français. On peut observer que les

mots ou les séquences de mots sémantiquement similaires sont mappés à des représentations proches. Les représentations distribuées des mots sont obtenues en utilisant la rétropropagation pour apprendre une représentation pour chaque mot et une fonction qui prédit une quantité cible telle que le mot suivant dans une séquence (pour la modélisation du langage) ou une séquence entière de mots traduits (pour la traduction automatique).

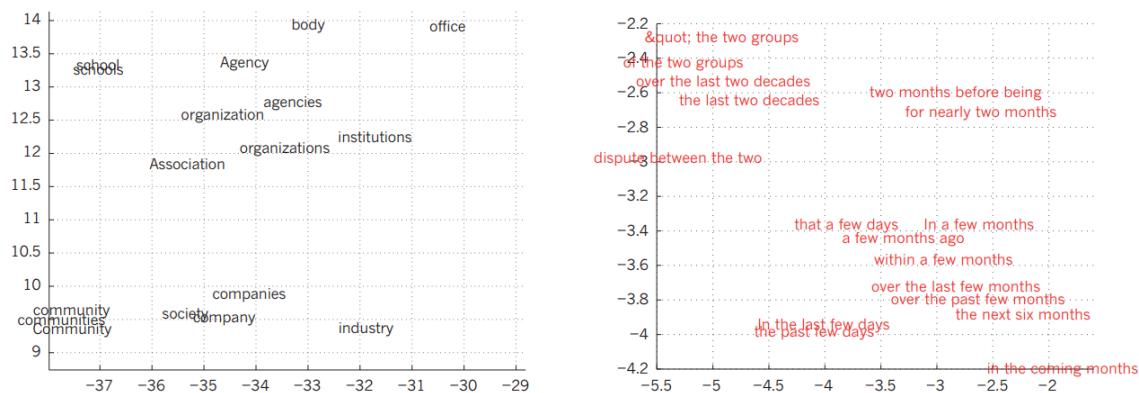


FIGURE 19 – Visualisation des vecteurs de mots appris [10]

Avant l'introduction des modèles de langage neuronaux, l'approche standard pour la modélisation statistique du langage ne profitait pas des représentations distribuées : elle était basée sur le comptage des fréquences d'occurrences de courtes séquences de symboles d'une longueur allant jusqu'à  $N$  (appelées N-grammes). Le nombre possible de N-grammes est de l'ordre de  $V^N$ , où  $V$  est la taille du vocabulaire, donc prendre en compte un contexte de plus de quelques mots nécessiterait des corpus d'entraînement très grands. Les N-grammes traitent chaque mot comme une unité atomique, donc ils ne peuvent pas généraliser à travers des séquences de mots sémantiquement liées, alors que les modèles de langue neuronaux le peuvent car ils associent chaque mot à un vecteur de caractéristiques réelles, et les mots sémantiquement liés se retrouvent proches les uns des autres dans cet espace vectoriel.

### 5.3 Autres domaines : jeux vidéo, recommandation de contenu, santé

Dans l'industrie des jeux vidéo, le deep learning a permis de créer des expériences utilisateur plus immersives et réalistes. Les personnages contrôlés par l'IA, grâce au deep learning, peuvent réagir de manière plus naturelle aux actions des joueurs, apprendre de leurs comportements et même développer des stratégies complexes. Par exemple, les techniques de deep reinforcement learning, où les agents apprennent à maximiser une fonction de récompense à travers des interactions avec leur environnement, ont été utilisées pour entraîner des agents à jouer et à exceller dans des jeux complexes comme ceux présentés par Mnih et al. dans leur étude de 2015 sur l'apprentissage profond par renforcement [21]

Le secteur de la recommandation de contenu bénéficie également largement des avancées en deep learning, notamment à travers des systèmes de recommandation personnalisée sur des plateformes comme Netflix, Spotify et YouTube. Ces systèmes utilisent des modèles de deep learning pour analyser les préférences des utilisateurs, leurs interactions

précédentes et même le contenu lui-même pour recommander des articles qui sont le plus susceptibles de plaire à l'utilisateur. Les réseaux de neurones profonds permettent une compréhension fine des caractéristiques des contenus, améliorant ainsi la précision des recommandations [22].

Dans le domaine de la santé, le deep learning a fait des progrès significatifs, particulièrement dans le diagnostic médical, l'analyse d'images médicales et la prédition de maladies. Les réseaux de neurones convolutifs (CNNs) sont au cœur de cette révolution, permettant par exemple de détecter avec précision des anomalies dans des images radiologiques ou des scans CT. Une étude notable de Esteva et al. (2017) [23] a démontré que les CNNs peuvent égaler la précision des dermatologues dans la détection du cancer de la peau, ouvrant la voie à des diagnostics plus rapides et accessibles [23]. Dans le domaine de la santé, le deep learning a fait des progrès significatifs, particulièrement dans le diagnostic médical, l'analyse d'images médicales et la prédition de maladies. Les réseaux de neurones convolutifs (CNNs) sont au cœur de cette révolution, permettant par exemple de détecter avec précision des anomalies dans des images radiologiques ou des scans CT. Une étude notable de Esteva et al. (2017) a démontré que les CNNs peuvent égaler la précision des dermatologues dans la détection du cancer de la peau, ouvrant la voie à des diagnostics plus rapides et accessibles [23].

## 6 Recherche et tendances futures dans le Deep Learning

Le domaine du Deep Learning connaît une évolution fulgurante et suscite un intérêt croissant de la part des chercheurs et des entreprises. De nouvelles avancées et tendances émergentes promettent de révolutionner encore davantage cette technologie et d'ouvrir de nouvelles perspectives dans de nombreux domaines.

### 6.1 Avancées récentes dans le domaine du Deep Learning

Parmi les avancées récentes les plus notables dans le domaine du Deep Learning, on peut citer :

1. **Apprentissage par renforcement profond (Deep Reinforcement Learning)** : Cette approche permet aux modèles Deep Learning d'apprendre par interaction avec leur environnement, en essayant différentes actions et en maximisant une récompense à long terme. Cela ouvre de nouvelles possibilités pour le développement de robots autonomes et d'agents intelligents capables d'apprendre et de s'adapter à des situations complexes. Exemples :
  - Le développement de robots capables d'apprendre à jouer à des jeux vidéo complexes comme Dota 2 ou StarCraft II en surpassant les meilleurs joueurs humains [24].
  - La création de drones autonomes capables de naviguer dans des environnements inconnus et d'accomplir des missions complexes [25].
  - L'entraînement de systèmes d'intelligence artificielle pour optimiser des processus industriels ou financiers.

2. **Apprentissage multi-tâches (Multi-Task Learning)** : Cette technique permet à un modèle Deep Learning d'apprendre à accomplir plusieurs tâches simultanément, en partageant les connaissances et les paramètres entre les différentes tâches. Cela peut améliorer les performances des modèles et les rendre plus adaptables à des environnements variés. Exemples :
  - La conception de modèles Deep Learning pour la reconnaissance d'images qui peuvent identifier plusieurs objets dans une image en même temps [26].
  - Le développement de systèmes de traduction automatique capables de traduire entre plusieurs langues avec un haut niveau de précision [27].
  - [28].
3. **Apprentissage métta (Meta-Learning)** : Cette approche permet aux modèles Deep Learning d'apprendre à apprendre plus rapidement et efficacement. Cela signifie qu'ils peuvent s'adapter à de nouvelles tâches avec peu de données ou d'instructions, ce qui les rend plus polyvalents et plus robustes. Exemples :
  - La création de systèmes d'intelligence artificielle capables d'apprendre à résoudre de nouveaux problèmes en transférant des connaissances acquises sur des tâches précédentes [29].
  - La conception de modèles Deep Learning capables de s'auto-améliorer en identifiant leurs propres faiblesses et en ajustant leurs paramètres en conséquence [30].
4. **Apprentissage fédéré (Federated Learning)** : Cette technique permet de collaborer sur des modèles Deep Learning sans partager les données sensibles des utilisateurs. Cela est particulièrement utile dans des domaines tels que la santé ou les finances, où la confidentialité des données est essentielle. Exemples :
  - Le développement de modèles de diagnostic médical qui peuvent apprendre à partir des données de plusieurs hôpitaux sans compromettre la confidentialité des patients [31].
  - La création de systèmes de recommandation personnalisés qui peuvent tirer parti des données d'utilisateurs provenant de différentes plateformes tout en protégeant leur vie privée [32].
  - La conception de modèles de détection de fraude financière qui peuvent apprendre à partir des données de plusieurs institutions financières sans révéler des informations sensibles [33].

## 6.2 Tendances émergentes et domaines de recherche prometteurs

Le domaine du Deep Learning continue d'évoluer rapidement et de nouvelles tendances émergent constamment. Parmi les domaines de recherche prometteurs qui retiennent l'attention des chercheurs, on peut citer :

1. **Apprentissage automatique explicite (Explainable AI)** : Cette approche vise à rendre les modèles Deep Learning plus transparents et interprétables, afin de mieux comprendre leurs décisions et de les déboguer plus facilement. Cela est crucial pour gagner la confiance des utilisateurs et garantir une utilisation responsable de cette technologie. Domaines de recherche :
  - La création d'outils permettant aux utilisateurs de comprendre comment les modèles Deep Learning prennent des décisions [34].

- La conception de modèles Deep Learning intrinsèquement explicables, qui fournissent des explications par défaut pour leurs prédictions [35].
- 2. Apprentissage neuronal neuromorphique (Neuromorphic Learning) :** Cette approche s'inspire du fonctionnement du cerveau humain pour concevoir des architectures de réseaux de neurones plus efficaces et plus économies en énergie. Cela pourrait révolutionner l'intelligence artificielle en la rendant plus adaptable et plus efficiente. Domaines de recherche :
- Le développement de puces neuromorphiques capables de simuler le fonctionnement du cerveau humain [36].
  - La conception d'algorithmes d'apprentissage inspirés des processus d'apprentissage biologique [37].
  - La création de systèmes d'intelligence artificielle capables d'apprendre et de s'adapter en temps réel, comme le cerveau humain [38].
- 3. Apprentissage automatique quantique (Quantum Machine Learning) :** Cette approche combine l'apprentissage automatique avec les principes de la mécanique quantique pour développer des algorithmes plus puissants et plus efficaces. Cela pourrait ouvrir de nouvelles possibilités pour résoudre des problèmes complexes dans des domaines tels que la chimie, la physique des matériaux et la cryptographie. Domaines de recherche :
- Le développement d'algorithmes d'apprentissage automatique quantiques pour la simulation de systèmes quantiques complexes [39].
  - La conception de systèmes d'intelligence artificielle quantiques capables de résoudre des problèmes NP-complets, qui sont actuellement considérés comme impossibles à résoudre par des ordinateurs classiques [40].
- 4. Apprentissage automatique à grande échelle (Large-Scale Machine Learning) :** Cette approche vise à développer des systèmes d'apprentissage automatique capables de traiter et d'analyser des ensembles de données massifs, tels que les données des réseaux sociaux, les images satellites ou les données génomiques . Cela pourrait révolutionner de nombreux domaines en permettant d'extraire des informations précieuses de ces vastes sources de données. Domaines de recherche :
- Le développement d'algorithmes d'apprentissage automatique distribués capables de traiter des données sur des infrastructures cloud ou des réseaux de calcul haute performance [41].
  - La création de techniques de réduction de dimensionnalité pour compresser et analyser des ensembles de données massifs [42].
  - La conception de systèmes d'intelligence artificielle capables d'apprendre en ligne à partir de flux de données en temps réel [43].
- 5. Apprentissage automatique pour le bien social (AI for Social Good) :** Cette approche vise à utiliser l'apprentissage automatique pour résoudre des problèmes sociaux et environnementaux tels que la pauvreté, la faim, les maladies et le changement climatique. Cela pourrait contribuer à améliorer la vie des populations dans le monde entier. Domaines de recherche :
- Le développement d'outils d'apprentissage automatique pour la cartographie de la pauvreté et l'aide aux populations vulnérables [44].
  - La création de systèmes d'intelligence artificielle pour la gestion des ressources naturelles et la lutte contre le changement climatique [45].

- La conception de modèles de Deep Learning pour la prévention des maladies et l'amélioration des soins de santé [46].

Ces tendances émergentes et les domaines de recherche prometteurs illustrent le dynamisme et le potentiel immense du domaine du Deep Learning. Au cours des prochaines années, nous pouvons nous attendre à voir des avancées encore plus spectaculaires qui révolutionneront de nombreux aspects de notre vie et ouvriront de nouvelles perspectives pour l'avenir de l'humanité.

## 7 Défis et limites du Deep Learning

Malgré ses nombreux atouts et son potentiel révolutionnaire dans le domaine de l'intelligence artificielle, le Deep Learning n'est pas exempt de défis et de limites qu'il est crucial de prendre en compte avant de l'intégrer dans un projet. Voici quelques points clés à examiner attentivement :

### 7.1 Problèmes liés à la taille des données et au temps de calcul

1. **Besoin de données volumineuses et de haute qualité** : L'entraînement efficace des réseaux Deep Learning nécessite des ensembles de données conséquents et de qualité irréprochable. Cela peut constituer un obstacle majeur pour les projets qui n'ont pas accès à des sources de données suffisamment vastes ou qui ne disposent pas de données adéquates en termes de qualité. La collecte et le prétraitement de ces données peuvent s'avérer des processus longs et coûteux, engendrant des difficultés supplémentaires.
  - **Exemples :**
    - Un modèle de reconnaissance d'images pour le diagnostic médical peut nécessiter des millions d'images de radiographies étiquetées par des professionnels de la santé, ce qui peut être difficile à obtenir en raison de restrictions de confidentialité et de la nécessité d'une expertise médicale [47].
    - Un système de traduction automatique pour une langue rare peut manquer de données textuelles suffisantes pour un apprentissage efficace, ce qui peut limiter sa performance et sa généralisation [48].
2. **Temps de calcul conséquents** : L'entraînement des modèles Deep Learning peut s'avérer gourmand en ressources informatiques, particulièrement pour les modèles complexes et les ensembles de données volumineux. Cela peut se traduire par des durées d'entraînement longues et des coûts élevés en termes de puissance de calcul, ce qui peut limiter l'applicabilité de cette technologie dans certains contextes.
  - **Exemples :**
    - L'entraînement d'un modèle Deep Learning pour la recherche de médicaments peut prendre plusieurs semaines ou mois sur des supercalculateurs coûteux, ce qui peut freiner les efforts de recherche et développement [49].
    - Le déploiement de modèles Deep Learning sur des appareils mobiles ou embarqués peut être contraint par les limitations de puissance de calcul et de stockage, nécessitant des optimisations et des adaptations spécifiques [50].

## 7.2 Interprétabilité des modèles

1. **"Boîte noire" et manque de transparence :** Le fonctionnement interne des réseaux Deep Learning peut être difficile à comprendre, ce qui limite leur interprétabilité. Il peut être ardu de cerner les raisons pour lesquelles un modèle Deep Learning prend une décision particulière, ce qui peut poser des problèmes dans des situations où une explication est nécessaire, comme dans le domaine médical ou juridique, ou pour déboguer des erreurs.

— **Exemples :**

- Un modèle de prédiction de risque de crédit peut être incapable d'expliquer les raisons pour lesquelles il a refusé une demande de prêt, ce qui peut frustrer le demandeur et limiter les possibilités de recours [51].
- Un système de détection d'anomalies dans les images médicales peut identifier une anomalie sans fournir d'informations sur la nature du problème ou les caractéristiques spécifiques de l'image qui ont déclenché l'alerte [52].

2. **Difficultés d'audit et de responsabilisation :** Le manque d'interprétabilité peut également soulever des inquiétudes concernant la transparence et la responsabilisation des systèmes Deep Learning. Il peut être difficile de déterminer si un modèle Deep Learning est biaisé ou discriminatoire, et il peut être ardu de contester ses décisions, ce qui peut engendrer des problèmes d'ordre éthique et juridique.

— **Exemples :**

- Un système de reconnaissance faciale utilisé par les forces de l'ordre peut être difficile à auditer et à responsabiliser en cas d'erreurs d'identification ou de discrimination, ce qui peut porter atteinte à la confiance du public et aux droits individuels [53].

## 7.3 Éthique et questions de confidentialité

1. **Biais et discrimination :** Les modèles Deep Learning peuvent être biaisés, reflétant les biais présents dans les données d'entraînement. Cela peut entraîner des décisions discriminatoires, en particulier pour les groupes minoritaires ou défavorisés.

— **Exemples :**

- Un système de prédiction de la récidive criminelle peut être biaisé contre les personnes ayant déjà été incarcérées, ce qui peut perpétuer des cycles d'incarcération et limiter les opportunités de réinsertion sociale [54].
- Des algorithmes de modération de contenu sur les réseaux sociaux peuvent discriminer à l'encontre de certaines opinions ou expressions, ce qui peut limiter la liberté d'expression [55].

2. **Confidentialité des données :** L'entraînement des modèles Deep Learning nécessite souvent l'utilisation de vastes ensembles de données personnelles. Il est crucial de garantir la confidentialité et la sécurité de ces données, en respectant les réglementations sur la protection des données et en obtenant le consentement éclairé des utilisateurs.

— **Exemples :**

- L'utilisation de données de santé pour entraîner des modèles de diagnostic médical doit se faire dans le respect du secret médical et de la vie privée des patients.

- La collecte de données pour la reconnaissance faciale dans les espaces publics soulève des questions concernant la surveillance de masse et la liberté individuelle.
3. **Manque de contrôle sur les systèmes Deep Learning :** Les systèmes Deep Learning complexes peuvent devenir difficiles à contrôler une fois déployés. Il est important de développer des mécanismes de surveillance et de contrôle pour s'assurer qu'ils fonctionnent comme prévu et qu'ils ne causent pas de dommages involontaires.
- **Exemples :**
    - Un système de trading algorithmique basé sur le Deep Learning pourrait prendre des décisions financières à grande vitesse et à grande échelle, ce qui pourrait provoquer des krachs boursiers si des mécanismes de contrôle appropriés ne sont pas mis en place.
    - Des robots autonomes équipés de Deep Learning pourraient commettre des erreurs ou causer des dommages s'ils ne sont pas dotés de systèmes de sécurité et de contrôle adéquats.

Le Deep Learning est donc un domaine en plein essor qui présente un immense potentiel pour de nombreuses applications. Cependant, il est essentiel de prendre en compte les défis et les limites associés à cette technologie, notamment les problèmes liés à la taille des données et au temps de calcul, le manque d'interprétabilité des modèles, ainsi que les questions d'éthique et de confidentialité. En développant des approches responsables et en mettant en place des garde-fous appropriés, nous pouvons maximiser les avantages du Deep Learning tout en minimisant ses risques et en garantissant une utilisation équitable et responsable de cette technologie puissante.

## 8 Étude de cas : Classification du cancer du sein par CNN

### 8.1 Introduction et motivations

Lorsque l'on aborde la classification du cancer du sein par les réseaux de neurones convolutifs (CNN), il est indéniable que le domaine du deep learning (DL) a révolutionné la manière dont nous diagnostiquons et traitons les maladies, en particulier dans le domaine de la santé. Cette approche innovante a ouvert de nouvelles voies pour améliorer la précision des diagnostics et la prise en charge des patients, tout en permettant une collaboration plus étroite entre les médecins et les technologies avancées.[56]

Le cancer du sein est l'une des formes de cancer les plus courantes chez les femmes à travers le monde, et son diagnostic précoce est essentiel pour un traitement efficace et des taux de survie accrus [57]. Les CNN, en tant qu'outil de deep learning, offrent un moyen prometteur d'améliorer la précision du diagnostic en analysant les images médicales, telles que les mammographies, pour détecter les signes précoce de la maladie[58]. Leur capacité à extraire des caractéristiques pertinentes à partir d'images complexes permet aux professionnels de la santé d'identifier les anomalies avec une précision accrue, souvent avant même qu'elles ne soient détectées par des méthodes traditionnelles.

L'une des motivations clés derrière l'utilisation des CNN dans la classification du cancer du sein est leur capacité à traiter de grandes quantités de données et à reconnaître des motifs complexes qui pourraient échapper à l'œil humain. En analysant des milliers, voire des millions d'images médicales annotées, les réseaux de neurones peuvent apprendre à identifier les caractéristiques subtiles associées aux tumeurs malignes, améliorant ainsi la sensibilité et la spécificité du diagnostic.

Cependant, il est important de noter que le DL ne vise pas à remplacer les médecins, mais plutôt à les soutenir dans leur pratique clinique. En fournissant des outils puissants pour l'analyse et l'interprétation des données médicales, le Machine Learning en général [59], et les CNNs en particulier peuvent aider les médecins à prendre des décisions plus éclairées et à personnaliser les traitements en fonction des besoins spécifiques de chaque patient. Cette approche collaborative entre l'intelligence artificielle et les professionnels de la santé ouvre la voie à une médecine plus précise et plus efficace, où la technologie agit comme un allié précieux dans la lutte contre les maladies.

## 8.2 Revue de littérature

Le tableau suivant synthétise les résultats de différentes études de recherche que nous avons examinées concernant la classification du cancer du sein. Il met en évidence les ensembles de données utilisés, le nombre de caractéristiques extraites, les méthodes de machine learning employées, ainsi que les résultats d'accuracy obtenus pour chaque étude :

Référence	Dataset	Features	Méthode	Accuracy
[60]	Wisconsin Breast Cancer Dataset (WBCD) (original)	11 attributs, 699 instances	SVM, C4.5, NB, k-NN	Meilleure accuracy pour SVM (97.13%)
[61]	WBCD + autre dataset	Dataset 1 : 11 attributs, 699 instances. Dataset 2 : 117 attributs, 102294 instances	SV avec trois fonctions et 2 caractéristiques : bagging et boosting	96,85%
[62]	WBCD	30 attributs, 699 instances	Random forest, Naive Bayes, SVM, KNN	97,9%
[63]	Wisconsin Prognostic Breast Cancer Dataset	32 attributs 194 instances	SVM	97%
[64]	Wisconsin Diagnostic Breast Cancer Dataset	30 attributs de 569 patients, 569 instances	Comparer l'apprentissage supervisé (SL) avec le semi-supervisé (SSL) pour 9 algorithmes	KNN (SL=98%, SSL=97%), logistics regression (SL=97%, SSL=98%)
[65]	WBCD (originale)	11 attributs avec 699 instances	Boosting Artificial Neural Network (BANN) avec deux SVMs	100%
[66]	WBCD	32 attributs, 569 instances	SVM avec optimisation par descente de gradient stochastique, regression logistique simple et réseau de neurones à perceptrons multicouches	99,44%

Référence	Dataset	Features	Méthode	Accuracy
[67]	DDSM, IN breast et BCDR	DDSM : 5316 images, 641 cas de patients. IN breast : 200 images pour 50 cas. BCDR : 600 images de 300 patients	CNN	97,35%, 95,50% et 96,67% pour les trois datasets respectivement
[68]	WBCD	9 attributs, 5699 instances	Deep Neural Network (DNN)	98,62%
[69]	Les datasets ont été collectées de deux institutions médicales	990 images, 540 masses malignes et 450 lésions bénignes	CNN	87,68%
[70]	La dataset a été collectée entre 2010 et 2016 dans cinq sites d'imagerie médicale	1,001,093 images de 141,473 patients	CNN	L'accuracy a été calculée en se basant sur l'aire sous la courbe (AUC)
[71]	Les données ont été collectées indépendamment	1200 cas avec 4000 échantillons prouvés cancéreux	CNN	L'accuracy a été calculée en se basant sur l'aire sous la courbe (AUC)
[72]	Dataset privée	67,520 images mammographiques de 16,968 femmes	CNN	95%

### 8.3 Jeu de données utilisé

Pour notre problème de classification du cancer du sein, nous avons opté pour le Breast Cancer Wisconsin (Diagnostic) Data Set, largement utilisé dans le domaine de l'apprentissage automatique et de la recherche biomédicale. Il contient des caractéristiques calculées à partir d'une image numérisée d'une aspiration à l'aiguille fine (AAF) d'une masse mammaire, qui peuvent être utilisées pour prédire si la masse est bénigne ou maligne.

Le jeu de données, présenté sous forme tabulaire, comprend des mesures de diverses caractéristiques des noyaux cellulaires présents dans les échantillons d'AAF, telles que le rayon, la texture, le périmètre, la surface, la régularité, la compacité, la concavité, les points concaves, la symétrie et la dimension fractale.

Chaque instance dans le jeu de données représente une masse mammaire, avec des caractéristiques calculées à partir d'une image numérisée et un diagnostic correspondant indiquant si la masse est maligne (cancéreuse) ou bénigne (non cancéreuse).

Le jeu de données a été initialement créé par le Dr William H. Wolberg de l'Université du Wisconsin Hospitals, Madison, Wisconsin, USA, et a été largement utilisé pour la recherche en apprentissage automatique, notamment dans le développement et l'évaluation d'algorithmes de classification pour le diagnostic du cancer du sein.

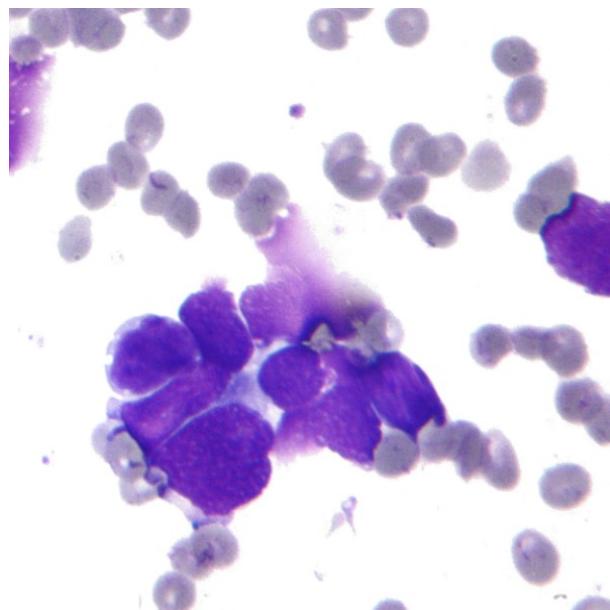


FIGURE 20 – Image histopathologique obtenue par biopsie à l'aiguille fine (FNA)

Le fichier .csv de ce dataset peut être trouvé suivant ce lien Kaggle :

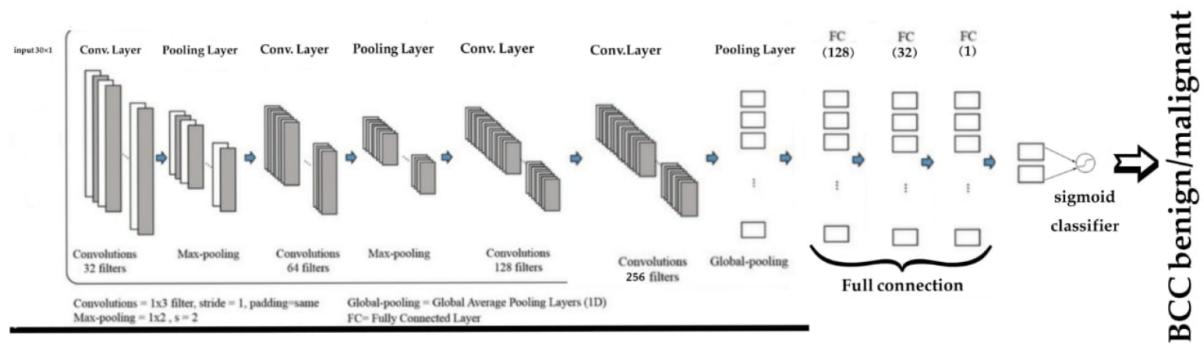
— Breast Cancer Wisconsin Data-KAGGLE

ou suivant :

— Breast Cancer Wisconsin Data-UCI Machine Learning Repository

## 8.4 Architecture CNN proposée

L'architecture CNN que nous proposons pour ce problème de classification est la suivante :



Les couches utilisées par ordre sont :

- Couche de convolution (32 filtres)
- Max pooling
- Couche de convolution (64 filtres)
- Max pooling
- Couche de convolution (128 filtres)
- Couche de convolution (256 filtres)
- Couche de Global Average pooling
- 3 couches entièrement connectées

A la fin, on utilise un "sigmoïd classifier", c'est-à-dire que les entrées seront pondérées et sommées, puis passées à travers une fonction sigmoïde qui produit une probabilité de classe. Si cette probabilité est supérieure à un seuil spécifié, la classe correspondante est prédite. Dans notre cas, nous avons deux classes : classe des cas médicaux bénins et la classe des cas médicaux malins.

Dans notre cas, on a utilisé des fonctions d'activation de type **LeakyReLU**, pour accélérer et stabiliser l'apprentissage, nous avons utilisé La **batch normalization**, et pour prévenir le surajustement (*overfitting* en anglais) et améliorer la généralisation du modèle, nous avons utilisé la technique du **dropout**.

## 8.5 Code Python

Le fichier .py accompagné de ce rapport contient le code utilisé pour implémenter notre modèle CNN.

Après avoir importé les bibliothèques nécessaires, et les données du fichier data.csv, nous avons effectué un prétraitement de ces données :

- supprimer les données inutiles
- diviser les données en ensembles d'entraînement et de test : 80% pour l'entraînement et 20% pour le test.
- normaliser les données d'entraînement : cette technique vise à mettre à l'échelle les valeurs des fonctionnalités pour qu'elles se situent dans une plage commune,

généralement entre 0 et 1 ou -1 et 1. Cela aide le modèle à converger plus rapidement pendant l'entraînement et à mieux généraliser sur de nouvelles données.

Ensuite, nous avons défini l'architecture du modèle CNN :

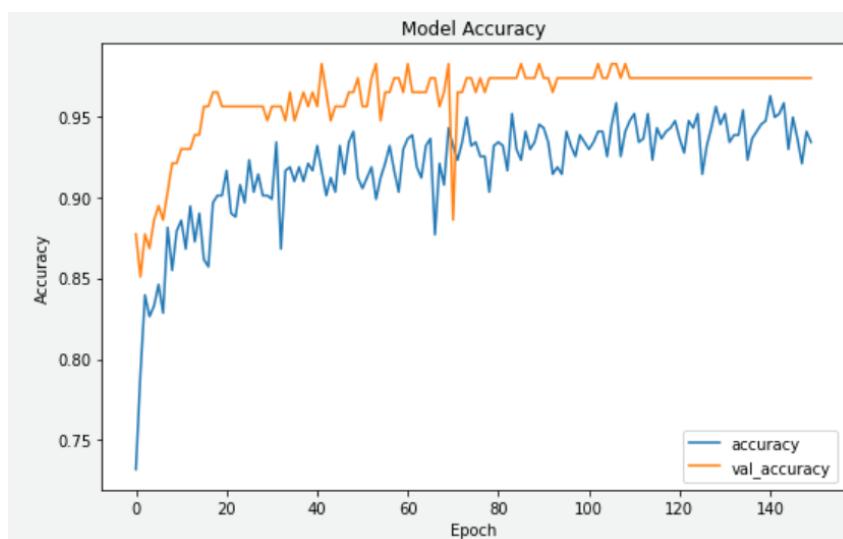
- Utilisation de couches convolutionnelles (Conv1D), suivies de fonctions d'activation LeakyReLU, de normalisation par lots (BatchNormalization) et de couches de régularisation (Dropout).
- Utilisation de couches de regroupement maximal (MaxPool1D) pour réduire la dimensionnalité.
- Utilisation de couches de densité pour la classification finale.

Après, nous avons précédé à l'entraînement du modèle sur Google Colaboratory, et avons obtenu les résultats présentés dans la partie suivante.

## 8.6 Résultats et performance

Après l'entraînement de notre modèle CNN, nous avons évalué sa performance en utilisant plusieurs KPI (Key Performance Indicators), à savoir :

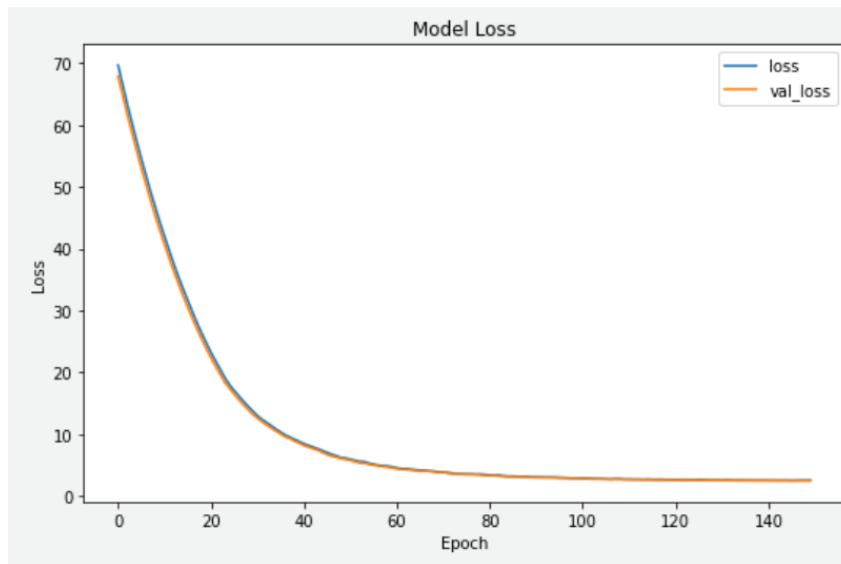
- **Précision (accuracy)** : Il s'agit du pourcentage de prédictions correctes parmi toutes les prédictions effectuées par le modèle. C'est l'un des KPI les plus basiques et les plus importants pour évaluer la performance globale d'un modèle CNN.



Notre modèle a réussi à atteindre une accuracy de presque 98% sur les données de validation (en orange) et 94% sur les données de test (en bleu). D'après la courbe ci-dessous, on peut remarquer que notre modèle **n'a pas d'overfitting** puisque la courbe bleue, représentant l'accuracy du modèle après entraînement sur les données de validation, augmente avec les époques jusqu'à atteindre des valeurs d'accuracy satisfaisantes.

- **Perte (loss)** : la loss est une fonction objectif que le modèle cherche à minimiser lors de l'entraînement. Dans notre étude, nous avons opté pour la **Binary Cross-Entropy Loss** qui est utilisée dans les problèmes de classification binaire. Elle mesure la divergence entre la distribution de probabilité prédite par le modèle et

la distribution réelle des étiquettes de classe binaires. Ci-dessous est la courbe de perte de notre problème de classification :



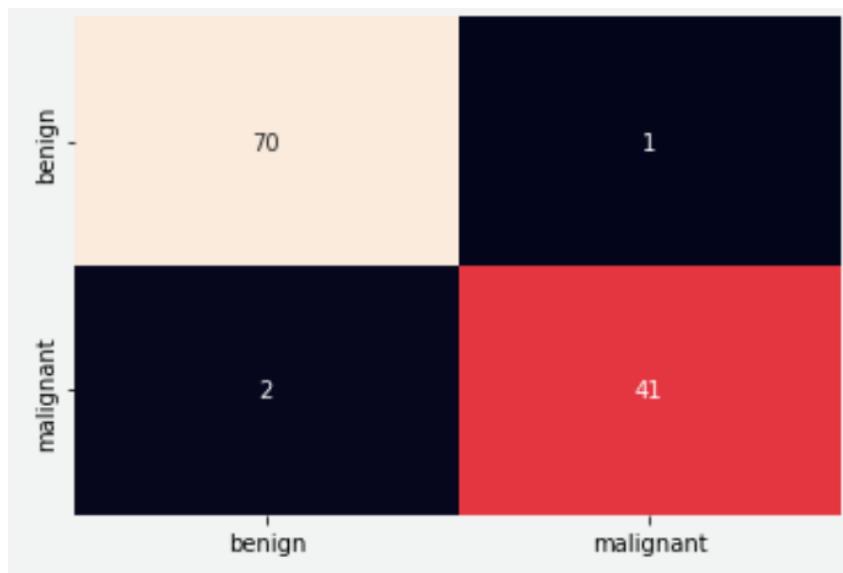
On remarque que la loss pour le test et la validation a diminué significativement avec chaque vingtaine d'époque. Ce qui peut montrer aussi que notre modèle a été bien entraîné sur les données test et a arrivé à prédire correctement plusieurs données de validation, et tout cela sans overfitting.

- **Précision (precision)** : La précision mesure la capacité du modèle à identifier correctement les instances positives parmi toutes les instances identifiées comme positives.
- **Rappel (recall)** : Le rappel mesure la capacité du modèle à identifier correctement toutes les instances positives dans un ensemble de données. Il est particulièrement important dans les problèmes de classification où la détection des cas positifs est critique.
- **F1-score** : Le F1-score est une mesure de la précision d'un modèle qui prend en compte à la fois le rappel et la précision. Il est calculé comme la moyenne harmonique de la précision et du rappel et est souvent utilisé comme un compromis entre les deux mesures.

Le tableau suivant résume les résultats de ces trois derniers KPIs :

	précision	recall	f1-score	support
0 (bénigne)	0.97	0.99	0.98	71
1 (maligne)	0.98	0.95	0.96	43
Précision	-	-	0.97	114

Pour avoir une vue plus claire sur la performance de notre modèle CNN, nous avons modélisé sa matrice de confusion :



Les lignes représentent les prédictions de notre modèle (cas malin ou bénin), alors que les colonnes représentent les vraies labels des données (cas malin ou bénin).

La matrice de confusion nous permet de visualiser les performances de notre modèle de classification du cancer du sein en comparant ses prédictions avec les véritables étiquettes des données. Les cellules de la matrice représentent le nombre d'échantillons classés correctement ou incorrectement par le modèle. En examinant les valeurs situées le long de la diagonale principale de la matrice, nous pouvons voir le nombre d'échantillons correctement classés pour chaque classe. Par exemple, le modèle a réussi à prédire correctement 70 cas bénins (True Positives), mais a échoué pour deux cas (False Positives) et les a classés en tant que cas malins alors qu'ils sont en vérité bénins. Les valeurs hors diagonale indiquent les erreurs de classification, où le modèle a prédit une classe mais les données réelles appartiennent à une autre classe.

## 8.7 Conclusion

La classification du cancer du sein par CNN représente un domaine de recherche crucial dans le domaine de la santé, offrant des perspectives prometteuses pour la détection précoce et précise de cette maladie. À travers notre étude, nous avons déployé un modèle de classification utilisant des CNNs qui a produit des résultats remarquables, atteignant une précision de près de 98%. Ces résultats démontrent l'efficacité et le potentiel des approches de deep learning pour la classification de données médicales complexes comme celles du cancer du sein.

L'utilisation de CNN dans la classification du cancer du sein permet non seulement une détection précise des tumeurs malignes et bénignes, mais offre également une méthode automatisée et reproductible, réduisant ainsi le temps et les coûts associés aux diagnostics traditionnels. Cependant, malgré ces performances impressionnantes, il reste encore des défis à relever, notamment en termes d'interprétabilité des modèles et de généralisation à

différentes populations et types de données.

En bref, notre étude met en lumière le potentiel considérable des CNNs dans la classification du cancer du sein, ouvrant la voie à des applications cliniques plus larges et à une amélioration continue de la précision diagnostique. Toutefois, des recherches supplémentaires sont nécessaires pour mieux comprendre les mécanismes sous-jacents et pour garantir la fiabilité et la généralisabilité des modèles dans des contextes cliniques réels.

## 9 Conclusion

Dans cette étude, nous avons exploré en profondeur les fondements théoriques, les principes d'entraînement, les applications et les défis associés au Deep Learning, en mettant en lumière ses avancées récentes et ses tendances futures. Nous avons examiné les différentes architectures de réseaux de neurones, y compris les réseaux convolutifs (CNN), les réseaux récurrents (RNN) et les réseaux récurrents à mémoire à court et long terme (LSTM), ainsi que les techniques d'entraînement et d'optimisation des poids.

Nous avons également exploré les applications du Deep Learning dans divers domaines, tels que la vision par ordinateur, le traitement du langage naturel et la santé, en mettant en évidence ses contributions significatives à ces domaines. En outre, nous avons examiné les recherches émergentes et les tendances futures dans le Deep Learning, ainsi que les défis et les limites auxquels il est confronté, tels que la taille des données, l'interprétabilité des modèles et les questions éthiques.

Enfin, nous avons présenté une étude de cas sur la classification du cancer du sein par CNN, où nous avons discuté de l'introduction et des motivations de l'étude, de la revue de littérature, du jeu de données utilisé, de l'architecture CNN proposée, ainsi que des résultats et de la performance obtenus. Notre modèle a produit des résultats prometteurs, avec une précision de près de 98%, démontrant ainsi le potentiel du Deep Learning dans la classification médicale. Toutefois, des recherches supplémentaires sont nécessaires pour explorer davantage les capacités et les limites des modèles de Deep Learning dans ce domaine.

## Références

- [1] O. Ezratty. Les usages de l'intelligence artificielle 2018, 2018. Consulté le 15 avril 2024.
- [2] Coursera. Apprentissage automatique supervisé : Régression et classification. Consulté le 9 avril 2024.
- [3] Yann LeCun and et al. Handwritten digit recognition with a back-propagation network. In *Advances in Neural Information Processing Systems*, pages 396–404, 1990.
- [4] Coursera. Deep learning. Consulté le 2 avril 2024.
- [5] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. In *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics*, pages 315–323, 2011.
- [6] Stanford University. Enseignement - cs 229. Consulté le 12 avril 2024.
- [7] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9 :1735–1780, 1997.
- [8] R. Dey and F. M. Salem. Gate-variants of gated recurrent unit (gru) neural networks. In *2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS)*, pages 1597–1600, Boston, MA, August 2017. IEEE.
- [9] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. <http://www.deeplearningbook.org>.
- [10] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553) :436–444, 2015.
- [11] Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout : A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1) :1929–1958, 2014.
- [12] Céline Deluzarche. Définition | Deep Learning - Apprentissage profond | Futura Tech — futura-sciences.com. <https://www.futura-sciences.com/tech/definitions/intelligence-artificielle-deep-learning-17262/>. [Accessed 28-04-2024].
- [13] Uchenna Odi and Thomas Nguyen. Geological facies prediction using computed tomography in a machine learning and deep learning environment. 07 2018.
- [14] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, London, England, 2016.
- [15] Hugo Larochelle. *Étude de techniques d'apprentissage non-supervisé pour l'amélioration de l'entraînement supervisé de modèles connexionnistes*. Ph.d. dissertation, Université de Montréal, 2009.
- [16] L. De Matteis, S. Janny, S. Nathan, and W. Shu-Quartier. Introduction à l'apprentissage automatique, 2022. Ce document donne un aperçu général de différents types d'apprentissage machine, y compris supervisé et non supervisé, avec des exemples pratiques et des explications théoriques.
- [17] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.
- [18] SH Haji and AM Abdulazeez. Examine divers algorithmes de descente de gradient pour l'optimisation dans les réseaux de neurones, y compris leur application dans des contextes médicaux tels que la détection du cancer. *PalArch's Journal of Archaeology of Egypt/Egyptology*, 2021. Liendel' article.

- [19] Sebastian Ruder. An overview of gradient descent optimization algorithms. <http://arxiv.org/abs/1609.04747>, 2016. Accessed : 29 October 2018.
- [20] Simon Haykin. *Neural Networks and Learning Machines*. Pearson, 2009.
- [21] Volodymyr Mnih et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540) :529, 2015.
- [22] Paul Covington, Jay Adams, and Emre Sargin. Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems*. ACM, 2016.
- [23] Andre Esteva, Brett Kuprel, Roberto A. Novoa, Justin Ko, Susan M. Swetter, Helen M. Blau, and Sebastian Thrun. Dermatologist-level classification of skin cancer with deep neural networks. *Nature*, 542(7639) :115, 2017.
- [24] Morgane Tual. Intelligence artificielle : un programme se hisse parmi les meilleurs joueurs de « Starcraft II » — lemonde.fr. <https://www.lemonde.fr/pixels/article/2019/10/30/intelligence-artificielle-un-programme-se-hisse-parmi-les-meilleurs-joueurs-de-st> 6017484\_4408996.html. [Accessed 28-04-2024].
- [25] Quand l'IA révolutionne les drones — linkedin.com. <https://www.linkedin.com/pulse/quand-lia-r%C3%A9volutionne-les-drones-arnault-ioualalen/>. [Accessed 28-04-2024].
- [26] Touraya El Hassani. You only look once - un réseau de neurones pour la détection d'objets. <https://blog.octo.com/you-only-look-once-un-reseau-de-neurones-pour-la-detection-dobjets>, 2021.
- [27] Tristan. Qu'est-ce que la traduction automatique neuronale ? <https://www.linguise.com/fr/blog/guide/quelle-est-la-traduction-automatique-neuronale/>, 2021.
- [28] Botpress Communauté. La puissance de l'ia conversationnelle chatbots. <https://botpress.com/fr/blog/le-pouvoir-des-chatbots-de-l-ai-conversationnelle>, février 17 2023.
- [29] Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. A comprehensive survey on transfer learning. *arXiv preprint arXiv:1911.02685*, 2019.
- [30] Paris Singularity. Quand l'intelligence artificielle pourra s'auto-améliorer, 2015.
- [31] PostDICOM. L'essor de l'imagerie médicale pilotée par l'ia : une nouvelle ère pour les diagnostics médicaux, Year.
- [32] Digital Test Lab. Recommandation personnalisée par ia : comment transformer l'expérience utilisateur, Year.
- [33] Alexandre Alfocea. Application : Comment le machine learning permet de détecter la fraude bancaire ?, 2021.
- [34] Aurélia Léon. Interprétabilité des modèles de machine learning, octobre 2020.
- [35] Alexandre Duval. L'intelligence artificielle explicative, enjeu majeur pour ouvrir les "boîtes noires", Juin 2021.

- [36] Philippe Richard. Des puces neuromorphiques imitent le cerveau pour être plus performantes, Août 2023.
- [37] Communauté LinkedIn. Quelles sont les meilleures techniques de conception d'algorithmes inspirées par la nature ?, Year.
- [38] Laurie Henry. Nouveau système d'ia s'auto-organise comme le cerveau humain, 2023.
- [39] Simulation quantique émulation de systèmes complexes avec q.
- [40] Joseph Mikael. Informatique quantique : Et si c'Était devenu un sujet de court terme ?, 2022.
- [41] M. Dehghani and Z. Yazdanparast. From distributed machine to distributed deep learning : a comprehensive survey. *Journal of Big Data*, 10 :158, 2023.
- [42] Nilesh Barla. Dimensionality reduction for machine learning, 2023.
- [43] Mariam Barry, Jacob Montiel, Albert Bifet, Sameer Wadkar, Nikolay Manchev, Max Halford, Raja Chiky, Saad El Jaouhari, Katherine Shakman, Joudi Fehaily, Fabrice Deit, Vinh-Thuy Tran, and Eric Guerizec. Streammllops : Operationalizing online learning for big data streaming real-time applications. pages 3508–3521, 04 2023.
- [44] Nandan Banerji, Sreenivasulu Ballem, Siva Mala Munnangi, and Sandeep Mittal. *A Machine Learning Approach for Poverty Estimation Using Aerial Images*. February 07 2024.
- [45] Unknown. The potential role of artificial intelligence in combating climate change and natural resources management : Political, legal, and ethical challenges. *21 Pages Posted : 18 Nov 2021*, 2021.
- [46] D Bordoloi, V Singh, S Sanober, SM Buhari, JA Ujjan, and R Boddu. Deep learning in healthcare system for quality of service. *J Healthc Eng*, 2022 :8169203, 2022.
- [47] Luciano M Prevedello, Safwan S Halabi, George Shih, Carol C Wu, Marc D Kohli, Fei H Chokshi, Bradley J Erickson, Jayashree Kalpathy-Cramer, Katherine P Andriole, and Adam E Flanders. Challenges related to artificial intelligence research in medical imaging and the importance of image analysis competitions. *Radiol Artif Intell*, 1(1) :e180031, Jan 2019.
- [48] Barry Haddow, Rachel Bawden, Antonio Valerio Miceli Barone, Jindřich Helcl, and Alexandra Birch. Survey of low-resource machine translation. *Computational Linguistics*, 48(3) :673–732, September 2022.
- [49] T Liu, D Lu, H Zhang, M Zheng, H Yang, Y Xu, C Luo, W Zhu, K Yu, and H Jiang. Applying high-performance computing in drug discovery and molecular simulation. *Natl Sci Rev*, 3(1) :49–63, 2016.
- [50] Yanjiao Chen, Baolin Zheng, Zihan Zhang, Qian Wang, Chao Shen, and Qian Zhang. Deep learning on mobile and embedded devices : State-of-the-art, challenges and future directions. *ACM Computing Surveys*, 53, 05 2020.
- [51] Francis Gichere. Credit decision making : The role of predictive models and regulations in consumer credit risk management, 2023.
- [52] Yizhou Wang, Dongliang Guo, Sheng Li, Octavia Camps, and Yun Fu. Explainable anomaly detection in images and videos : A survey. 2023.
- [53] UN Interregional Crime and Justice Research Institute and World Economic Forum (Organisation : Switzerland). *A Policy Framework for Responsible Limits on Facial*

- Recognition Use Case : Law Enforcement Investigations.* UNICRI, Turin, Italy, 2022.  
 Insight report : revised November 2022.
- [54] Kelly E Moore, Jeffrey B Stuewig, and June P Tangney. The effect of stigma on criminal offenders' functioning : A longitudinal mediational model. *Deviant Behav*, 37(2) :196–218, Feb 2016.
- [55] Anastasia Kozyreva, Stefan M. Herzog, Stephan Lewandowsky, and Jason Reifler. Resolving content moderation dilemmas between free speech and harmful misinformation. Edited by Timothy Wilson, University of Virginia, Charlottesville, VA, Feb 2023.
- [56] Z. Gao, Z. Lu, J. Wang, S. Ying, and J. Shi. A convolutional neural network and graph convolutional network based framework for classification of breast histopathological images. *IEEE Journal of Biomedical and Health Informatics*, 26 :3163–3173, 2022.
- [57] MD Ali, A Saleem, H Elahi, MA Khan, MI Khan, MM Yaqoob, U Farooq Khattak, and A Al-Rasheed. Breast cancer classification through meta-learning ensemble technique using convolution neural networks. *Diagnostics (Basel)*, 13(13) :2242, 2023.
- [58] Y. Li, J. Wu, and Q. Wu. Classification of breast cancer histology images using multi-size and discriminative patches based on deep learning. *IEEE Access*, 7 :21400–21408, 2019.
- [59] S. Ara, A. Das, and A. Dey. Malignant and benign breast cancer classification using machine learning algorithms. In *Proceedings of the 2021 International Conference on Artificial Intelligence (ICAI)*, pages 97–101, Islamabad, Pakistan, 2021.
- [60] H. Asri, H. Mousannif, H. Al Moatassime, and T. Noel. Using machine learning algorithms for breast cancer risk prediction and diagnosis. *Procedia Comput. Sci.*, 83(Fams) :1064–1069, 2016.
- [61] M. W. Huang, C. W. Chen, W. C. Lin, S. W. Ke, and C. F. Tsai. Svm and svm ensembles in breast cancer prediction. *PLoS One*, 12(1) :1–14, 2017.
- [62] Y. K. and M. Bahaj. Applying best machine learning algorithms for breast cancer prediction and classification. In *International Conference on Electronics, Control, Optimization and Computer Science (ICECOCS)*, pages 1–5, 2018.
- [63] R. Rawal. Breast cancer prediction using machine learning. *J. Emerg. Technol. Innov. Res.*, 7(5), 2020.
- [64] N. Al-Azzam and I. Shatnawi. Comparing supervised and semi-supervised machine learning models on diagnosing breast cancer. *Ann. Med. Surg.*, 62(December 2020) :53–64, 2021.
- [65] M. Abdar and V. Makarenkov. Cwv-bann-svm ensemble learning classifier for an accurate diagnosis of breast cancer. *Meas. J. Int. Meas. Confed.*, 146(May) :557–570, 201.
- [66] A. S. Assiri, S. Nazir, and S. A. Velastin. Breast tumor classification using an ensemble machine learning method. *J. Imaging*, 6(6) :39, May 2020.
- [67] A. O. Chougrad and H. Zouaki. Deep convolutional neural networks for breast cancer screening. *Comput Methods Prog Biomed*, 157 :19–30, 2018.
- [68] P. V. S. S. R. C. M. S. Karthik and R. Srinivasa Perumal. Breast cancer classification using deep neural networks. *Knowl Comput Its Appl Knowl Manip Process Tech*, 1 :227–241, 2018.

- [69] H. Cai, Q. Huang, W. Rong, Y. Song, J. Li, J. Wang, J. Chen, and L. Li. Breast microcalcification diagnosis using deep convolutional neural network from digital mammograms. *Comput. Math. Methods Med.*, 2019(January 2020), 2019.
- [70] et Al NW. Deep neural networks improve radiologists' performance in breast cancer screening. *IEEE Trans. Med. Imaging*, 39(4) :1184–1194, 2020.
- [71] E. F. Conant, A. Y. Toledano, S. Periaswamy, S. V. Fotin, J. Go, J. E. Boatsman, and J. W. Hoffmeister. Improving accuracy and efficiency with concurrent use of artificial intelligence for digital breast tomosynthesis. 2020.
- [72] G. V. Ionescu, M. Fergie, M. Berks, E. F. Harkness, J. Hulleman, A. R. Brentnall, J. Cuzick, D. G. Evans, and S. M. Astley. Prediction of reader estimates of mammographic density using convolutional neural networks. *J. Med. Imaging*, 1584(1) :012005, 2020.