

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université 8 Mai 1945 – Guelma
Faculté des Sciences et de la Technologie
Département de Génie Electrotechnique et Automatique

Réf:...../2023



MEMOIRE

Présenté pour l'obtention du **diplôme de MASTER Académique**

Domaine: Sciences et Technologie

Filière: Automatique

Spécialité: Automatique et Informatique industrielle

Par: (Djemaa Mahir)

Thème

Classification des images par CNN

Soutenu publiquement, le 18/06/2023, devant le jury composé de :

Babouri Abdesselam	Professeur	Univ. Guelma	Président
Boucerredj Leila	Docteur/MCA	Univ. Guelma	Examineur
Debeche Mehdi	Docteur/MAA	Univ. Guelma	Encadreur

Année Universitaire: 2022/2023

Dédicace:

à mon père qui est le meilleur père dans ce monde, grâce à son encouragement, sa confiance et son soutien moral et matériel et pour son amour infini j'exprime une profonde gratitude

à ma mère mon profond amour et ma passion Pour tous les sacrifices que vous faites pour moi, pour toute la confiance que vous placez en moi et pour tout l'amour qui m'entoure

pour toute la famille

A mes chers amis

A toutes les personnes que je porte dans mon cœur.

Nous dédions cet humble travail à :

En premier lieu à ceux que personne ne peut remplacer

Le sacrifice qu'ils ont fait pour notre éducation et notre bien-être.

pour nos parents qui se sacrifient pour subvenir à nos besoins

Tout au long de notre période de formation, qui est la base de notre travail

Que Dieu les bénisse et les protège.

A notre chère qui nous a apporté son soutien

Dans les moments les plus difficiles.

Remerciement :

Nous remercions Dieu Tout-Puissant de nous avoir donné la santé et le bien-être
je Désir de commencer et de terminer cette lettre. Tout d'abord, nous tenons à
remercier notre encadreur Monsieur Debeche mehdi pour son Aide et encadrement,
nous le remercions aussi pour la qualité exceptionnelle de son encadrement

Sa patience, sa précision et sa disponibilité lors de la préparation
de notre mémoire.

Nous tenons à le remercier

aussi pour son aide pratique , son soutien moral et son
encouragement. Nos remerciements vont également à tous
Nos enseignants pour leur générosité et leur grande patience
Ils ont pu démontrer malgré leurs charges académiques
professionnels.

Résumé :

L'objectif principal de ce mémoire est de construire un modèle de classification basé sur les techniques d'apprentissage automatique, Nous utiliserons à cette effet un modèle de réseaux de neurones convolutifs (CNN) pour identifier et classer correctement des images de chiffres appartenant à la base de données MNIST, cette base de données contient des images de chiffres manuscrits en noir et blanc, de taille 28 x 28 pixels.

Dans ce travail on mettra en évidence l'efficacité des modèles à base de CNN dans la reconnaissance et la classification de chiffres manuscrits à partir de leur image brute ce qui sera l'objet de notre partie application.

.

Summary :

The main objective of this thesis is to build a classification model based on machine learning techniques, We will use for this purpose a model of convolutional neural networks (CNN) to identify and correctly classify images of digits belonging to the MNIST database, this database contains images of black and white handwritten digits of size 28 x 28 pixels.

In this work we will highlight the effectiveness of CNN-based models in the recognition and classification of handwritten digits from their raw image, which will be the subject of our application part.

.

ملخص

الهدف الرئيسي من هذه الأطروحة هو بناء نموذج تصنيف يعتمد على تقنيات التعلم الآلي ، و لهذا الغرض سوف نستخدم نموذج الشبكة العصبية التلافيفية **CNN** وهذا لتحديد وتصنيف بشكل صحيح صور أرقام تنتمي إلى قاعدة بيانات **MNIST** تحتوي قاعدة البيانات هذه على صور لأرقام مكتوبة بخط اليد بالأبيض والأسود، بحجم 28×28 بكسل.

في هذا العمل ، سنسلط الضوء على فعالية النماذج المستندة إلى **CNN** في التعرف على الأرقام المكتوبة بخط اليد وتصنيفها ابتداءً من صورتها الخامة ، والتي ستكون موضوع جزء التطبيق لدينا.

Sommaire

Introduction générale

CHAPITRE I *L'intelligence artificielle*

<i>I.1 - Introduction.....</i>	2
<i>I.2 - Définition.....</i>	3
<i>I.3 - Définition</i>	3
<i>I.4 - Historique de l intelligence artificiel.....</i>	3
<i>I.5 - L'utilisation de l'intelligence artificielle.....</i>	5
<i>I.6 - Les domaines d'applications de L'IA</i>	5
<i>I.7 - Les avantages, inconvénients et les limites de l'intelligence artificielle</i>	6
<i>I - Apprentissage Profond (Deep Learning)</i>	7
<i>I.1 - Introduction</i>	7
<i>I.2 - Définition</i>	8
<i>I.3 - Les différentes Architectures du Deep Learning.....</i>	9
<i>I.3-1 Les réseaux de neurones convolutifs.....</i>	9
<i>I. 3-2 - Réseau de neurones récurrents (RNN).....</i>	9
<i>I.3-2-1 Les types de réseaux de neurones récurrents</i>	10
<i>I.3-2-2 L' Architectures RNN</i>	11
<i>I. 3-3 Generative Adversarial Networks</i>	13
<i>I. 3-4 ResNets</i>	13
<i>I. 3-5 Auto-encoders</i>	14
<i>I. 4 - Domaines d'application de l'apprentissage profonde.....</i>	15
<i>I. 5 - Conclusion</i>	16

CHAPITRE II *L'algorithme CNN Convolutional Neural Net Work*

<i>I.1- Introduction.....</i>	17
<i>I.2- Les réseaux de neurones artificiels</i>	17
<i>I.2-1 Définition</i>	17
<i>I.2-2 Architecture d'un réseau de neurones artificiel</i>	17
<i>I.2-3 Types des réseaux de neurones</i>	18
<i>I.2-3-1 Les réseaux de neurones à propagation avant (feed-forward).....</i>	18
<i>I.2-3-2 Les réseaux de neurones récurrents.....</i>	18
<i>I.2-3-3 Les réseaux de neurones convolutifs.....</i>	18
<i>II - Convolutional neural net work</i>	19
<i>II.1 - Définition</i>	19
<i>II.2 - Principe d'architecture d'un CNN</i>	20
<i>II.3 - Architecture de réseaux de neurone convolutif.....</i>	21

<i>II.4 - Les couches de CNN</i>	22
<i>II.4-1 Couche de convolution (CONV)</i>	22
<i>II.4-2 Couche Pooling</i>	25
<i>II.4-3 Couche de correction Relu</i>	26
<i>II.4-4 Couche entièrement connectée(FC)</i>	27
<i>II.4-5 La couche de perte (LOSS)</i>	27
<i>II.5 - Architecture générale d'un CNN en 2D et en 3D</i>	28
<i>II.5-1 Architecture générale d'un CNN en 2D</i>	28
<i>II.5-2 Architecture générale d'un CNN en 3D</i>	30
<i>II.6 - Example of 2D Convolution</i>	32
<i>II.7 - Convolution 3D</i>	35
<i>II.8 - Conclusion</i>	37

Chapitre III *Classification*

<i>III.1 - Introduction</i>	38
<i>III.2 - Définition</i>	38
<i>III.3 - Les étapes d'une classification</i>	38
<i>III.4 - Différent type de classification</i>	39
<i>III.4-1 Classification binaire</i>	39
<i>III.4-2 Classification multi classe</i>	39
<i>III.5 - Différentes approches de classification</i>	39
<i>III.5-1 Apprentissage supervisé</i>	40
<i>III.5-2 Apprentissage non supervisé</i>	40
<i>III.5-3 Apprentissage par renforcement</i>	40
<i>III.5-4 Apprentissage semi-supervisé</i>	40
<i>III.6 - Principe de fonctionnement de la classification</i>	40
<i>III.7 - Domaine d'applications des algorithmes de classification</i>	41
<i>III.8 - Quelle est la précision des modèles d'apprentissage</i>	42
<i>III.8-1 Métriques de mesure de performance d'un classifieur</i>	42
<i>III.8-2 Les avantages de l'utilisation de la précision comme métrique</i>	44
<i>III.8-3 Over all Accuracy (Précision global)</i>	44
<i>III.8-4 Le coefficient Kappa</i>	45
<i>III.8-5 Algorithme d'optimisation</i>	46
<i>III.8-5-1 Descente du gradient (Gradient Descent)</i>	46
<i>III.8-5-2 Descente de gradient par lots (Batch Gradient Descent)</i>	47
<i>III.8-5-3 Descente de gradient stochastique</i>	47
<i>III.8-5-4 Descente de gradient en mini-lot (Mini-Batch Gradient Descent)</i>	47
<i>III.8-5-5 Descente de gradient en mini-lot avec inertie (Momentum)</i>	47
<i>III.9 - Calcul des poids dans les réseaux de neurones multicouche</i>	51
<i>III.10 - Ingénierie des réseaux de neurones</i>	52
<i>III.11 - Conclusion</i>	58

Chapitre IV
L'implémentation

<i>IV.1 - Introduction</i>	<i>59</i>
<i>IV.2 - Logiciels utilisés : MATLAB.....</i>	<i>59</i>
<i>IV.3 - Optimisateur utiliser : sgd (signifie Stochastic Gradient Descent with Momentum)</i>	<i>59</i>
<i>IV.4 - Partie d'application</i>	<i>59</i>
<i>IV.5 - Structure model CNN.....</i>	<i>62</i>
<i>IV.6 - Conclusion.....</i>	<i>68</i>

Conclusion générale

Liste des figures :

Chapitre I : L'intelligence artificiel		
Figure I.1	Les types de l'intelligence artificielles	3
Figure I.2	La relation entre l'intelligence artificielle et le ML et le deep learning	8
Figure I.3	Réseau de neurone profond de type convolutionnel	8
Figure I.4	(à gauche) Un RNN (à droite) Sa version déroulé Source	10
Figure I.5	Couche de neurones récurrents devant une couche de neurones traditionnels.	11
Figure I.6	Récurrent Neural Network (RNN) Tutorial: Types and Examples	12
Figure I.7	General architecture of GAN	13
Figure I.8	Apprentissage résiduel	14
Figure I.9	Representation Basic d'un Auto encoder	15
Chapitre II : L'algorithme CNN convolutional Neural Net Work		
Figure II.1	Architecture d'un réseau de neurones	18
Figure II.2	Architecture générale d'un réseau de neurones convolutif (CNN/ConvNet)	20
Figure II.3	fonction d'activation relu	27
Figure II.4	Une couche fully-connected (FC)	27
Figure II.5	Opération de convolution 2D	28
Figure II.6	2D pooling opération	29
Figure II.7	Structure Dun réseau CNN 2D	29
Figure II.8	Architecture générale d'un CNN en 3D	30
Figure II.9	Couche de convolution 3D	31
Figure I.10	Opération de convolution 3d à l'aide d'une image 3d	31
Figure I.11	3D pooling opération	32
Figure I.12	3D Fully Connected Layer in cnn classification	32
Chapitre III : Classification		
Figure III.1	Types de classification	39
Figure III.2	Les étapes effectuées dans un réseau de neurones pour chaque couche	52
Figure III.3	La structure d'un réseau de neurones.	53
Chapitre IV : Implémentation		
Figure IV.1	Exemple d'image de chiffres appartenant à notre base de données	60
Figure IV.2	Evolution des valeurs des métriques de performance en fonction du nombre de block convolutif	61
Figure IV.3	Structure d'un Model CNN	62
Figure IV.4	3l'évolution de la précision global (Over all Accuracy) pour le model à 3 block de convolution	62
Figure IV.5	Evolution des valeurs des métriques de performance en fonction de la taille du Mini Lot (Mini Batch)	63
Figure IV.6	Le graphe suivant représente l'évolution de la précision global et de la fonction de perte pour un mini lot de 32	64
Figure IV.7	Evolution des valeurs des métriques de performance en fonction de la taille de l'ensemble d'apprentissage	65
Figure IV.8	Exemple de reconnaissance du chiffre 8 à partir de son image brute	66

Liste des Tableaux :

Figure IV.1	Tableau 1 : les différentes valeurs des 3 métriques obtenus pour différentes profondeurs de modèle	61
Figure IV.2	Tableau 2 : les différentes valeurs des 3 métriques obtenus pour différentes tailles de mini lot	63
Figure IV.3	Tableau 3 : les différentes valeurs des 3 métriques obtenues pour validation du modèle choisi pour différentes tailles d'ensemble d'apprentissage	64

Liste des acronymes utilisés dans le mémoire:

CNN	: Convolutional Neural Network
RNN	: Réseau de neurones récurrent
LSTM	: Long Short-Term Memory
GRU	: Gated Recurrent Unit
GAN	: Generative Adversarial Networks
AE	: Auto-Encodeur
ReLU	: Unités Rectifié linéaires
RN	: Réseaux de neurones
AI	: Artificielle Intelligence
ML	: Machine learning
DL	: Deep Learning
GPU	: Graphics processing unit

Introduction général

Au cours des dernières années, le deep learning et en particulier les réseaux de neurones ont connu une popularité croissante, notamment grâce à l'émergence des algorithmes convolutifs (CNN) qui ont permis de résoudre efficacement des problèmes complexes en apprentissage automatique. Les CNN sont devenus l'une des structures de réseau les plus représentatives dans le domaine du traitement et de la reconnaissance d'images.

La technologie d'apprentissage en profondeur, dont font partie les CNN, a connu un grand succès dans le traitement et la reconnaissance d'images en raison de sa capacité à apprendre des caractéristiques de manière automatique et à exploiter des modèles complexes.

Les performances des CNN ont souvent dépassé celles des méthodes traditionnelles, ouvrant de nouvelles perspectives dans le domaine de la vision par ordinateur.

- Dans le premier chapitre : Nous nous intéresseront aux notions d'intelligence artificielle et d'apprentissage en profondeur, en mentionnant les caractéristiques, les relations et les domaines de leur utilisation.
- Dans Le deuxième chapitre : Nous définirons l'architecture des réseaux convolutifs CNN et la constitution de leurs différentes couches.
- Dans Le troisième chapitre : une définition de la classification mentionnant les types de classification et les critères de classification sera donnée.

Quant au dernier chapitre nous le consacrerons à la partie pratique (classification d'image brut de chiffre à l'aide d'un CNN) moyennant le logiciel MATLAB, et à la fin nous terminerons par une conclusion générale et un point de vue.

Chapitre I
L'intelligence artificielle

I - L 'Intelligence Artificielle :

1 - Introduction :

L'Intelligence Artificielle (IA) est un domaine en constante évolution et il est difficile de le définir de manière précise. Les premiers chercheurs dans ce domaine avaient pour objectif de concevoir des machines intelligentes capables de reproduire les tâches intellectuelles complexes accomplies par les êtres humains. Cependant, ils ont rapidement réalisé les limites de leurs ambitions. Par la suite, les chercheurs ont continué leurs travaux en explorant de nouvelles techniques informatiques, que ce soit au niveau logiciel ou matériel. Ils ont également développé des langages de programmation orientés objet et des machines spécifiquement conçues pour la programmation symbolique, comme le langage LISP. [1]

Au fil des années de recherche, l'IA a quitté les laboratoires pour devenir partie intégrante de notre vie quotidienne. Par exemple, elle nous aide à naviguer dans les villes en évitant les embouteillages grâce à des systèmes de navigation intelligents. De plus, nous utilisons des assistants virtuels pour nous aider à accomplir diverses tâches, que ce soit la gestion de nos calendriers, la recherche d'informations ou l'automatisation de certaines actions.

L'IA continue de se développer et de trouver de nouvelles applications dans de nombreux domaines, tels que la médecine, la finance, l'industrie, l'agriculture et bien d'autres. Elle offre des possibilités d'automatisation, d'optimisation et de prise de décision basée sur des données à une échelle sans précédent.

2 - Définition :

L'intelligence artificielle, également connue sous le nom d'IA, est une discipline scientifique et technologique qui cherche à permettre aux machines, telles que les ordinateurs et les programmes informatiques, d'exécuter des processus cognitifs qui sont généralement associés aux capacités du cerveau humain. Ces processus incluent la compréhension, la communication (tant entre les machines qu'avec les humains), l'adaptation et l'apprentissage autonome grâce à des techniques telles que le deep Learning. [2]

3 - Définition :

L'intelligence artificielle (IA) est un domaine de l'informatique qui est étroitement lié à d'autres disciplines. Il s'appuie sur des fondements théoriques tels que les mathématiques, la logique et les statistiques. Il tire également des enseignements des sciences humaines telles que les sciences cognitives, la psychologie, la philosophie et la linguistique. De plus, l'IA s'inspire de la neurobiologie pour reproduire certaines caractéristiques de l'intelligence humaine par le biais du bio mimétisme. Enfin, les avancées dans les technologies matérielles fournissent le support physique nécessaire à l'exécution des logiciels d'IA. [4]

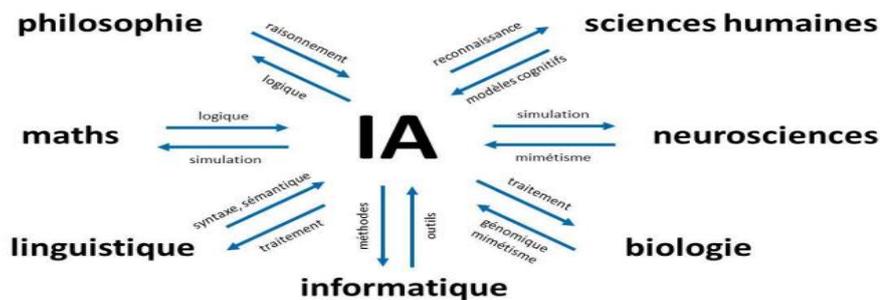


Figure I.1: les types de l'intelligence artificielles

4 - Historique de l'intelligence artificielle [5] :

1-1950 :

Alan Turing a publié un article intitulé "Computing Machinery and Intelligence" dans lequel il a présenté un test appelé "Imitation Game", qui est maintenant connu sous le nom de test de Turing. Ce test consiste à soumettre un candidat, humain ou machine, à une série de questions, et un juge doit déterminer si les réponses proviennent d'un être humain ou d'une

machine. Aujourd'hui, le test de Turing est un concours où des participants tentent de convaincre le juge qu'ils sont des humains plutôt que des machines, en utilisant leur capacité à communiquer et à répondre de manière convaincante.

2-1974-1980 :

Entre 1974 et 1980, l'IA a traversé ce qui est connu sous le nom de "l'hiver de l'IA". Ce terme fait référence à une période où le financement gouvernemental et l'intérêt pour le domaine de l'IA ont considérablement diminué

3-1980 :

En août 1980, l'Université de Stanford a accueilli la première conférence nationale de l'American Society for Artificial Intelligence.

4-1997 :

En 1997, le champion du monde d'échecs Garry Kasparov a été vaincu par le système IBM Deep Blue

5-2002 :

En 2002, l'IA est entrée dans la maison pour la première fois sous la forme d'un aspirateur Roomba.

6- 2011 :

Watson d'IBM a triomphé dans Jeopardy, où il a dû faire face à des questions et des énigmes complexes, et donner des réponses exactes

7-2014 :

Eugene Goostman a remporté un concours dans le célèbre "test de Turing", qui suscite de nombreuses controverses. Bien que l'on débâte de la réussite réelle d'Eugene au test, les rumeurs ont réussi à convaincre 33% des juges du panel qu'il était un véritable être humain. Ce résultat est survenu lors du 60e anniversaire de la mort d'Alan Turing, qui avait prédit qu'à l'approche de l'an 2000, les ordinateurs seraient suffisamment intelligents pour tromper les humains en les faisant croire qu'ils sont réels environ 30% du temps.

8-2017 :

Chaque jour, des avancées significatives dans le domaine de l'intelligence artificielle se produisent, et cette chronologie ne fait qu'effleurer la surface de ce vaste domaine. Elle constitue un point de départ pour un livre blanc qui explorera plus en détail l'impact de l'IA sur le monde des arts à but non lucratif et pourquoi il est important de s'en préoccuper. Restez à l'écoute pour découvrir comment l'IA vous permettra d'améliorer votre productivité et de travailler de manière plus intelligente

5 - L'utilisation de l'intelligence artificielle:

Ces dernières années, l'intelligence artificielle a trouvé sa place dans notre vie quotidienne et continue de progresser à grands pas. Voici quelques exemples d'utilisation courante de

l'intelligence artificielle[2]:

- Les systèmes de messagerie électronique
- Les réseaux sociaux
- Les moteurs de recherche
- Les outils de traduction
- Les plateformes de commerce en ligne
- Les applications de navigation
- Reconnaissance d'image

6 - Les domaines d'applications de L'IA :

Comme l'Intelligence Artificielle, s'est développée en parallèle avec le calcul numérique et non en concurrence avec celui-ci. Elle s'attaque donc, à des secteurs d'application réputés difficiles ou impossibles à résoudre par le calcul numérique. [1] Nous pouvons citer entre autres :

- Le calcul formel.
- la vision : analyse de texte, d'images.
- La robotique : génération de plans.
- Les machines autonomes : perception, interprétation, décision, action.
- Le langage naturel : traduction, compréhension, synthèse.

- La démonstration de théorèmes,
- Les jeux,
- La représentation des connaissances dans diverses disciplines (droit, médecine, automatique, etc.).

7 - Les avantages, inconvénients et les limites de l'intelligence artificielle :

7-1 Les avantages :

- L'intelligence artificielle a le potentiel de remplacer les humains dans leurs tâches quotidiennes, permettant ainsi d'effectuer des travaux pénibles ou dangereux tels que le ménage, les courses, la cuisine, le jardinage, sans aucune contrainte physique comme le besoin de repos ou de nourriture[2].
- Grâce aux algorithmes de l'IA, les calculs sur les ordinateurs peuvent être réalisés de manière plus rapide et efficace, réduisant ainsi les erreurs par rapport aux calculs effectués par les êtres humains.
- L'IA facilite les déplacements grâce à des véhicules autonomes équipés de caméras et de capteurs, capables de se déplacer sans intervention humaine.
- Dans le domaine médical, l'IA présente de nombreux avantages, tels que le suivi à distance des patients, les prothèses intelligentes et les traitements personnalisés

7-2 Les inconvénients :

- Un inconvénient majeur qui peut être envisagé est la possibilité d'erreurs dans la programmation des robots, ce qui pourrait compromettre leur bon fonctionnement. Les machines telles que les ordinateurs, les robots et les véhicules intelligents ne sont pas capables de détecter les erreurs de programmation. Les conséquences d'une telle erreur pourraient être catastrophiques à grande échelle, mais le risque reste très faible.
- L'augmentation du taux de chômage est un autre effet potentiel, car les entreprises peuvent préférer remplacer les travailleurs par des robots dotés d'intelligence artificielle, qui ne se fatiguent pas et ne nécessitent qu'une maintenance occasionnelle, entraînant ainsi des suppressions d'emplois.

- Les coûts élevés de recherche et développement dans le domaine de l'IA constituent également un défi. La création de robots autonomes capables de fonctionner dans la vie quotidienne serait extrêmement coûteuse.

II - Apprentissage Profond (Deep Learning) :

1 Introduction :

L'apprentissage en profondeur est un domaine de recherche récent dans le domaine de l'apprentissage automatique (ML) qui s'inspire de la structure et du fonctionnement du cerveau. Les algorithmes utilisés dans l'apprentissage en profondeur peuvent apprendre des niveaux de représentation multiples pour modéliser des relations complexes entre les données. Ces dernières années, le deep learning a attiré une grande attention en raison de ses performances exceptionnelles, en particulier dans la classification d'images, grâce à l'utilisation d'algorithmes basés sur des réseaux de neurones qui imitent le cerveau.

Le deep learning est une forme d'intelligence artificielle dérivée de l'apprentissage automatique, où la machine est capable d'apprendre et de s'améliorer par elle-même en fonction du volume de données disponibles. Les progrès réalisés dans le domaine du deep learning ont été rendus possibles grâce à l'augmentation de la puissance de calcul des ordinateurs et au développement de vastes bases de données.

L'apprentissage en profondeur repose sur des réseaux neuronaux artificiels composés de milliers d'unités, appelées neurones, qui effectuent des opérations simples. Les résultats d'une couche de neurones sont ensuite utilisés comme entrées pour les calculs de la couche suivante, et ainsi de suite. L'illustration suivante montre la relation entre les trois concepts : intelligence artificielle (AI), apprentissage automatique (ML) et apprentissage en profondeur (DL).

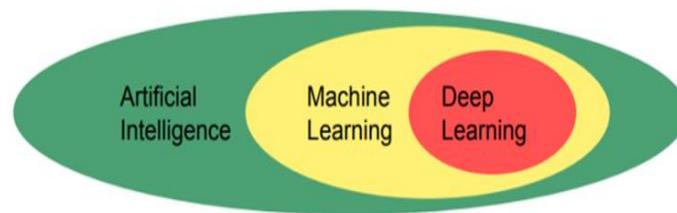


Figure I.2: La relation entre l'intelligence artificielle et le ML et le deep learning

2- Définition :

Le deep learning est une branche spécifique des techniques d'apprentissage automatique (machine learning) qui repose sur l'utilisation de réseaux de neurones à plusieurs couches, également appelées couches cachées [7]. Ces couches permettent de décomposer de manière hiérarchique le contenu de données complexes telles que la voix ou les images, afin de les classer par la suite. Par exemple, cela peut inclure l'identification des mots dans la voix ou l'association de tags descriptifs aux images.

Le deep learning peut être considéré comme un sous-domaine de l'apprentissage automatique, utilisant des algorithmes qui tirent parti de l'apprentissage à plusieurs niveaux pour créer un modèle capable de représenter les relations complexes entre les données. Cela permet à la machine d'apprendre de manière autonome et de former un réseau de neurones profond de type convolutionnel (Figure 3). Le deep learning se situe à l'intersection des réseaux de neurones, de la modélisation graphique, de l'optimisation, de l'intelligence artificielle, et de la reconnaissance de formes et du traitement du signal.

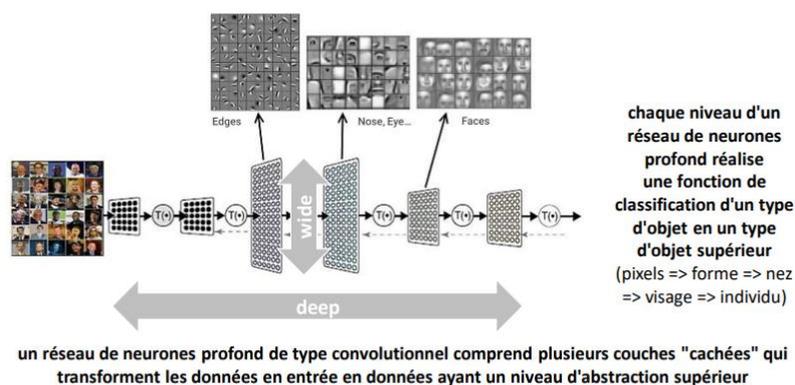


Figure I.3: Réseau de neurone profond de type convolutionnel

3- Les différentes Architectures du Deep Learning :

Le Deep Learning repose sur le concept des réseaux de neurones artificiels et est spécialement conçu pour traiter de grandes quantités de données en ajoutant des couches au réseau. Un modèle de Deep Learning est capable d'extraire des caractéristiques à partir des données brutes en utilisant plusieurs couches de traitement qui comprennent des transformations linéaires et non linéaires. Il peut apprendre progressivement à partir de ces caractéristiques à travers chaque couche, avec une intervention humaine minimal [10].

3-1 - Les réseaux de neurones convolutifs :

Réseau de neurones spécialisé dans le traitement de données présentant une structure de grille. Ils se sont avérés extrêmement efficaces dans des domaines tels que la reconnaissance et la classification d'images et de vidéos. Les CNN ont réussi à identifier des visages, des objets, des panneaux de signalisation et ont même été utilisés dans le domaine de la conduite autonome. Plus récemment, les CNN ont également montré leur efficacité dans diverses tâches de traitement du langage naturel, telles que la classification de phrases [10].

Dans le domaine de l'apprentissage automatique (Machine Learning), un réseau convolutif CNN est un type de réseau de neurones à propagation avant (feed forward) qui s'inspire des processus biologiques, Il met en œuvre cinq principales couches :

- La couche de convolution.
- La couche Rectified Linear Unit.
- La couche pooling.
- La couche entièrement connectée FC.
- La couche de perte.

3-2 Réseau de neurones récurrents (RNN) :

Un réseau de neurones récurrent (RNN) est une catégorie de réseaux de neurones qui comprend des connexions pondérées au sein d'une même couche, contrairement aux réseaux anticipatifs traditionnels où seules les connexions anticipatives sont établies vers les couches suivantes. Les RNN sont dotés de boucles, ce qui leur permet de stocker des informations lors

du traitement de nouvelles entrées. Cette capacité de mémoire les rend idéaux pour des tâches de traitement où les entrées précédentes, telles que des séries chronologiques, doivent être prises en compte. C'est pourquoi les réseaux de neurones profonds actuels sont largement basés sur les RNN [8] .

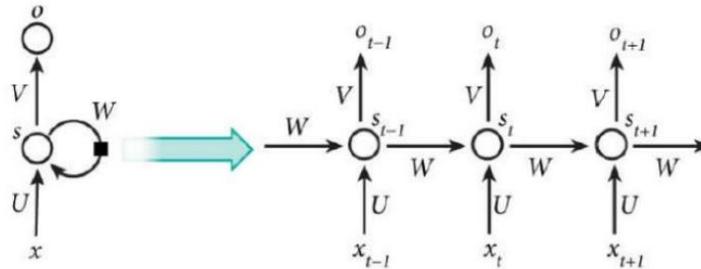


Figure I.4 : (à gauche) Un RNN (à droite) Sa version déroulé Source

Le schéma ci-dessus montre un RNN déroulé. En déroulant, nous signifions simplement qu'on montre le réseau pour la séquence complète.

Par exemple, si la séquence qui nous intéresse est une phrase de 5 mots, le réseau serait déroulé en un réseau de neurones de 5 couches, une couche pour chaque mot[9] .

Les formules qui régissent les calculs dans un RNN sont les suivantes :

- x_t est l'entrée au moment t .
- u et w sont les paramètres que le réseau va apprendre des données d'apprentissage.
- s_t est l'état caché au moment t , C'est la « mémoire » du réseau. s_t est calculé en fonction de l'état Caché précédent et de l'entrée à l'étape actuelle :

$$(w \cdot s_{t-1} + x_t \cdot u) = s_t \dots \dots \dots (1.I)$$

Les types de réseaux de neurones récurrents :

Les réseaux de neurones traditionnels analysent des données étiquetées mais ne sont pas conçus pour faire des prédictions sur des séries temporelles, c'est-à-dire des données qui évoluent dans le temps. Pour traiter ce type de données, il existe trois principaux types de réseaux de neurones récurrents : le RNN simple, le LSTM et le GRU.

Le RNN simple, également appelé "Vanilla RNN", est la forme la plus basique de RNN. Il ne possède pas de portes (Gates), ce qui signifie que le flux d'informations n'est pas contrôlé. Les informations essentielles à la tâche peuvent être écrasées par des informations redondantes ou non pertinentes. En pratique, les RNN simples ne sont pas largement utilisés et sont principalement étudiés à des fins d'enseignement.

Les LSTM (Long Short-Term Memory) et les GRU (Gated Recurrent Unit) sont des améliorations des RNN simples. Ils comprennent plusieurs portes (respectivement 3 et 2) qui permettent de contrôler sélectivement l'oubli ou la mémorisation des informations de la séquence temporelle précédente dans une mémoire dynamique. Ces portes permettent aux LSTM et aux GRU de mieux gérer les dépendances à long terme dans les données séquentielles, ce qui les rend plus adaptés pour les tâches de prédiction de séries temporelles.

Comme les réseaux de neurones traditionnels, les réseaux de neurones récurrents peuvent également comporter plusieurs couches. Cela leur permet de capturer davantage de non-linéarités dans les données, mais cela augmente également le temps de calcul lors de la phase d'apprentissage. De plus, il est possible de combiner des couches récurrentes avec des couches classiques telles que des couches denses (MLP) ou des couches de convolution (CNN) pour bénéficier des avantages des deux architectures.

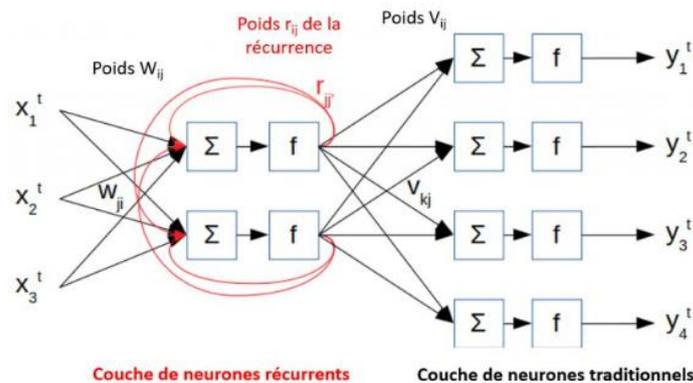


Figure I.5: Couche de neurones récurrents devant une couche de neurones traditionnels.

3-2-1 Architectures RNN :

L'architecture d'un réseau de neurones récurrents (RNN) se compose généralement de plusieurs couches récurrentes empilées. Chaque couche récurrente peut être un RNN simple,

un LSTM ou un GRU, comme mentionné précédemment. Voici une représentation générale de l'architecture d'un RNN :

- **Entrée (Input):** Les données séquentielles sont introduites dans le réseau par l'intermédiaire de l'entrée. Chaque séquence de données est représentée par une série d'éléments (par exemple, des mots dans une phrase ou des instants temporels dans une série temporelle).
- **Couche récurrente (Recurrent Layer):** Les informations séquentielles sont traitées dans la couche récurrente. Cette couche possède des connexions de rétroaction qui permettent aux informations de circuler d'une étape à l'autre dans la séquence. Cela permet au réseau de capturer les dépendances temporelles et de modéliser les relations complexes entre les éléments séquentiels.
- **Optionnel :** Stacking de couches récurrentes (Stacking Recurrent Layers) : Il est possible d'empiler plusieurs couches récurrentes les unes sur les autres pour former un réseau de neurones récurrents profond. Chaque couche récurrente traite les informations provenant de la couche précédente, permettant ainsi au réseau de capturer des niveaux de représentation plus abstraits et complexes.
- **Couche de sortie (Output Layer):** La couche de sortie génère les prédictions ou les sorties souhaitées en fonction des informations traitées par les couches récurrentes. La nature de la tâche détermine le type de couche de sortie utilisée. Par exemple, pour la classification, une couche de sortie dense avec une fonction d'activation appropriée peut être utilisée..

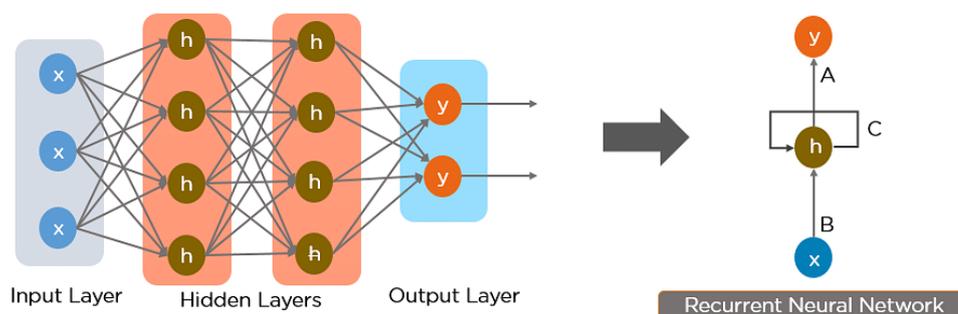


Figure I.6: Recurrent Neural Network (RNN) Tutorial: Types and Examples

3-3 Generative Adversarial Networks :

La prémisse de base des réseaux adverses génératifs (GAN) est la formation simultanée de deux modèles d'apprentissage profond. Dans les réseaux adverses génératifs (GAN), deux modèles d'apprentissage en profondeur s'affrontent : le générateur, dont le rôle est de produire de nouvelles instances ou exemples, et le discriminateur, dont l'objectif est de déterminer si une instance donnée provient des données d'apprentissage ou a été générée par le générateur. Ainsi, ces deux modèles sont en compétition pour accomplir leurs tâches respectives[10].

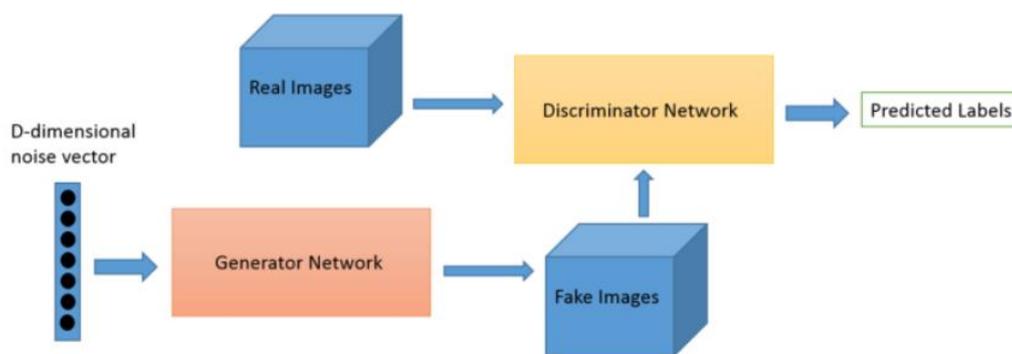


Figure I.7: General architecture of GAN

3-4 ResNets :

Depuis leur popularité croissante en 2015, les réseaux ResNets, également connus sous le nom de Deep Residual Networks, ont été largement adoptés et utilisés par de nombreux scientifiques des données et chercheurs en IA. Comme vous le savez déjà, les CNN (Convolutional Neural Networks) sont extrêmement utiles pour résoudre des problèmes de classification d'images et de reconnaissance visuelle. Cependant, à mesure que ces tâches deviennent plus complexes, la formation d'un réseau neuronal devient plus difficile, car il nécessite des couches profondes supplémentaires pour calculer et améliorer la précision du modèle[10].

L'apprentissage résiduel a été conçu pour résoudre ce problème, et l'architecture résultante est couramment appelée ResNet. Cette approche permet de faciliter la formation de réseaux profonds en introduisant des connexions résiduelles, qui permettent aux informations d'être transmises directement d'une couche à une autre sans subir de transformations

complexes. Cela permet aux réseaux ResNet d'apprendre plus efficacement et d'atteindre de meilleures performances, même avec un nombre élevé de couches.

En résumé, les réseaux ResNet sont devenus une méthode populaire dans le domaine de l'apprentissage en profondeur en raison de leur capacité à résoudre des problèmes de classification d'images complexes grâce à leur architecture qui facilite la formation de réseaux profonds.

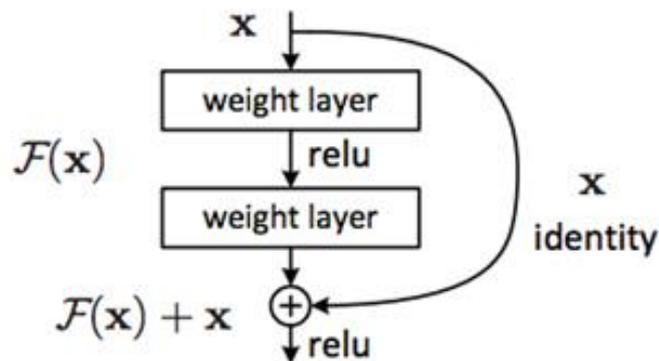


Figure I.8: Apprentissage résiduel

3-5 Auto-encoders :

Un auto-encodeur (AE) est un modèle qui réalise à la fois la compression de l'entrée (via un encodeur) et la décompression de celle-ci (via un décodeur). Son objectif principal est de reproduire les données d'entrée à travers les données de sortie. Les auto-encodeurs sont des réseaux de neurones qui opèrent en compressant l'entrée dans une représentation latente, puis en reconstruisant la sortie à partir de cette représentation.

Ce type de réseau se compose généralement de trois parties distinctes :

L'encodeur : Cette partie du réseau se charge de compresser l'entrée en une représentation dans un espace latent. Il peut être représenté par une fonction de codage, par exemple :

$$h = f(x) \dots \dots \dots (2.1)$$

L'espace latent : C'est l'espace où la partie compressée des données d'entrée est représentée.

Le décodeur : Cette partie du réseau a pour objectif de reconstruire l'entrée à partir de la représentation dans l'espace latent. Elle peut être représentée par une fonction de décodage, par exemple :

$$y = g(h) \dots \dots \dots (3.1)$$

Ainsi, les auto-encodeurs permettent de compresser les données d'entrée en une représentation latente, puis de les reconstruire avec le décodeur, visant à reproduire fidèlement les données d'origine [10]

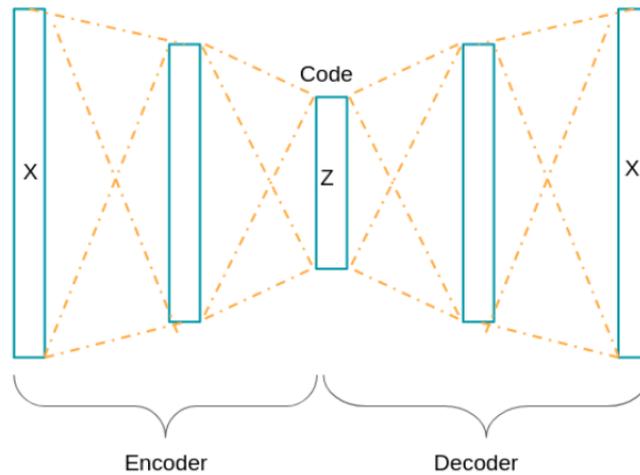


Figure I.9: A basic representation of Auto encoder

4 -Domaines d'application de l'apprentissage profonde :

Le deep learning est utilisé dans de nombreux domaines [11] :

- Reconnaissance d'image,
- Traduction automatique,
- Voiture autonome,
- Diagnostic médical,
- Recommandations personnalisées,
- Modération automatique des réseaux sociaux,
- Prédiction financière et trading automatisé,
- Identification de pièces défectueuses,
- Détection de malwares ou de fraudes,
- *Chatbots* (agents conversationnels),
- Exploration spatiale,
- Robots intelligents.
- Deep Boltzmann Machines

5- Conclusion :

Dans ce chapitre, nous avons défini le concept d'intelligence artificielle et nous avons expliqué la relation entre les trois concepts d'intelligence artificielle (IA), d'apprentissage machine (ML) et d'apprentissage profond (DL) qui font partie de l'IA.

Le chapitre suivant définit le réseau neuronal, qui est le composant de base du CNN

Chapitre II

L'algorithme CNN Convolutional Neural Net Work

1-Introduction :

Les réseaux de neurones convolutifs sont conçus pour fonctionner avec des entrées structurées en grille, qui ont de fortes dépendances. Les images sont un exemple de données structurées sous forme de grille qui se prêtent généralement bien à l'utilisation de réseaux de neurones. Le Deep Learning s'est révélé être un outil puissant au cours des dernières décennies en raison de sa capacité à traiter de grandes quantités de données. L'introduction de couches cachées a dépassé les approches traditionnelles, notamment en ce qui concerne la reconnaissance de formes. Parmi les réseaux de neurones profonds les plus populaires, les réseaux de neurones convolutifs (également connus sous le nom de CNN ou ConvNet) jouent un rôle central dans l'apprentissage en profondeur, notamment dans les applications de vision par ordinateur.

Depuis les années 1950, les premiers jours de l'intelligence artificielle, les chercheurs ont rencontré des difficultés pour développer des systèmes capables de comprendre les données visuelles. Au fil des années, ce domaine est devenu connu sous le nom de vision par ordinateur. En 2012, la vision par ordinateur a connu une avancée majeure lorsque des chercheurs de l'Université de Toronto ont développé un modèle d'IA qui surpassait de loin les meilleurs algorithmes de reconnaissance d'images disponibles à l'époque[12].

2-Les réseaux de neurones artificiels :***2-1 Définition :***

Un réseau de neurones artificiels est un système conçu à l'origine en s'inspirant du fonctionnement des neurones biologiques, mais qui s'est ensuite rapproché des méthodes statistiques. Ces réseaux sont composés de processeurs élémentaires fortement connectés, qui fonctionnent en parallèle. Chaque processeur élémentaire, ou neurone artificiel, effectue des calculs pour générer une sortie spécifique en se basant sur les informations qu'il reçoit[13].

2-2 Architecture d'un réseau de neurones artificiel :

En général, un réseau de neurones est composé d'un ensemble de couches successives, où chaque couche reçoit ses entrées à partir des sorties de la couche précédente. Cela signifie que l'ensemble du réseau est entièrement connecté. Chaque couche est constituée de neurones qui ne sont pas connectés entre eux, mais qui reçoivent des informations numériques des

neurones voisins. L'ensemble des couches comprend une couche d'entrée qui lit les valeurs d'entrée, une couche de sortie qui fournit les résultats du système, et entre ces deux se trouvent une ou plusieurs couches cachées qui participent au processus de transfert des informations[2].

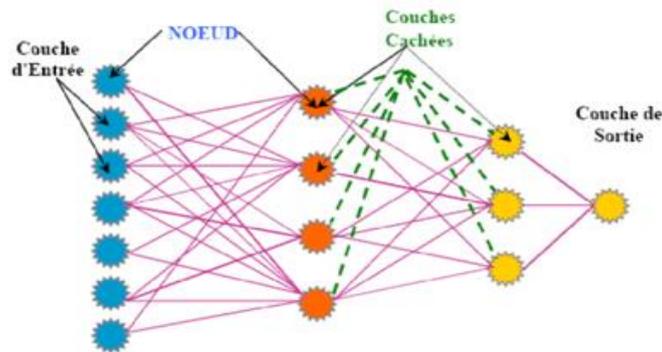


Figure II.1: Architecture d'un réseau de neurones

2-3 Types des réseaux de neurones :

Il existe différents types de réseaux de neurones, et ils sont classés en fonction du nombre de nœuds cachés du modèle ou encore du nombre d'entrées et de sorties de chaque nœud.

La propagation des informations entre les différents neurones peut varier et dépend du type de réseaux de neurones.

Il existe plusieurs types de réseaux de neurones :

2-3-1 Les réseaux de neurones à propagation avant (feed-forward) : C'est la variante la plus simple, où l'information se déplace uniquement dans une direction. Elle traverse directement les nœuds de traitement (couches cachées) à partir de l'entrée, puis se dirige vers les sorties. Dans ce type de réseau, il n'y a pas de cycles ou de boucles.

2-3-2 Les réseaux de neurones récurrents: Ce mode d'apprentissage est un peu plus complexe. Ces réseaux comportent au moins un cycle dans leur structure, ce qui leur permet de sauvegarder les résultats produits par les nœuds de traitement et de les utiliser pour nourrir le modèle. Les réseaux de neurones récurrents sont utilisés dans des domaines tels que la reconnaissance automatique de formes et la traduction automatique de la parole.

2-3-3 Les réseaux de neurones convolutifs: Inspirés par le fonctionnement du cortex visuel des animaux, ces réseaux traitent les données en les faisant passer à travers plusieurs couches, de la couche d'entrée à la couche de sortie. Les réseaux de neurones convolutifs sont utilisés

dans des domaines tels que la reconnaissance faciale, la reconnaissance d'images et de vidéos, ainsi que le traitement du langage naturel.

II - Convolutional neural net work:

1- Definition:

Dans le domaine de l'apprentissage en profondeur, les réseaux de neurones convolutifs (CNN/ConvNet) sont une classe de réseaux de neurones profonds largement utilisés pour l'analyse d'images visuelles. Contrairement à ce que l'on pourrait penser généralement lorsqu'on évoque les réseaux de neurones, les CNN ne se contentent pas d'utiliser des multiplicateurs matriciels. Ils font appel à une technique spéciale appelée "bypass" ou "skip connection".

La convolution, en mathématiques, est une opération arithmétique qui permet de combiner deux fonctions pour en produire une troisième, exprimant ainsi la manière dont la forme de l'une est modifiée par l'autre. Dans le contexte des CNN, la convolution est utilisée pour extraire des caractéristiques des images en appliquant des filtres spécifiques sur des régions locales. Cela permet aux CNN de capturer des motifs et des informations significatives au sein des images[14].

En résumé, les réseaux de neurones convolutifs (CNN/ConvNet) sont une classe de réseaux de neurones profonds largement utilisés dans l'analyse d'images. Ils se distinguent par l'utilisation de la convolution, une opération mathématique qui permet de capturer des caractéristiques visuelles importantes. Les CNN utilisent également des connexions spéciales appelées "bypass" ou "skip connection" pour améliorer la performance et la capacité d'apprentissage du réseau

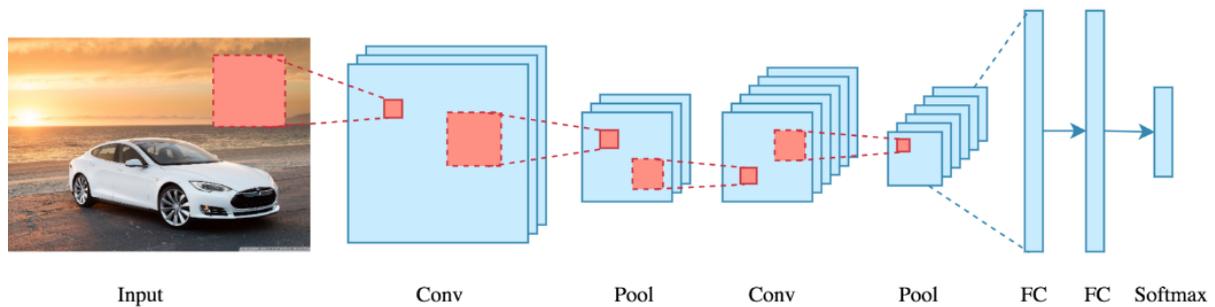


Figure II.2: Architecture générale d'un réseau de neurones convolutif (CNN/ConvNet)

2- Principe d'architecture d'un CNN :

Un réseau de neurones convolutif n'est pas seulement un réseau neuronal profond avec de nombreuses couches cachées. Il s'agit plutôt d'un réseau profond qui simule le fonctionnement du cortex visuel du cerveau pour reconnaître et classifier des images ou des vidéos, et pour découvrir un objet ou même une partie dans une image.

Le concept et le fonctionnement des réseaux de neurones convolutifs est différent des autres réseaux de neurones, en effet un réseau neuronal convolutif comporte deux parties distinctes avec une entrée dans laquelle une image en forme de matrice de pixels bidimensionnelle (avec 2 dimensions, noir et blanc), ou une image couleur avec 3 dimensions (couleurs : rouge, vert et bleu) ou une image multidimensionnelle (image satellitaire) .

La première partie d'un réseau de neurones convolutif est la partie convolutionnelle qui sert à extraire les caractéristiques de l'image. Ensuite, l'image passe par le fichier de séquence de filtre, ou le noyau d'enroulement, ce qui conduit à la création d'une nouvelle image appelée cartes de convolution. Généralement, les filtres intermédiaires réduisent la résolution de l'image.

Ensuite, les cartes des caractéristiques sont aplaties dans un vecteur de caractéristiques pour former les données d'entrée de la partie de couche entièrement connectée. Le rôle principal de cette couche (complètement connectée) est de combiner les caractéristiques contenues dans le vecteur de son entrée pour la classification des images[15].

3- Architecture de réseaux de neurone convolutif :

Les réseaux de neurones convolutifs (CNN), également appelés Convolutional Neural Networks, sont constitués de trois types de couches principales : les couches convolutives, les couches de regroupement et les couches entièrement connectées. Lorsqu'elles sont empilées dans un ordre spécifique, ces couches forment une architecture CNN. Chacune de ces couches est traitée indépendamment et joue un rôle spécifique dans le processus de traitement des données :

- **La couche convolutive (CONV)** : traite les données en utilisant des filtres ou des champs récepteurs pour détecter des motifs et des caractéristiques spécifiques dans les données d'entrée. Elle applique des opérations de convolution pour extraire les informations pertinentes.
- **La couche de regroupement (POOL)** : permet de réduire la taille des cartes de caractéristiques générées par les couches convolutives. Cela est généralement réalisé par des techniques de sous-échantillonnage, telles que le max pooling ou le averagepooling, qui réduisent la résolution spatiale tout en préservant les informations essentielles.
- **La couche de correction (ReLU)** : est souvent appelée "ReLU" en référence à la fonction d'activation utilisée, l'Unité de rectification linéaire. Elle introduit de la non-linéarité dans le modèle en activant les neurones avec des valeurs positives et en supprimant les valeurs négatives. Cela permet d'apprendre des caractéristiques complexes et d'améliorer la capacité du modèle à capturer des modèles non linéaires.
- **La couche entièrement connectée (FC)** : est une couche de type perceptron, où chaque neurone est connecté à tous les neurones de la couche précédente. Elle agit en combinant les caractéristiques apprises précédemment pour effectuer des prédictions ou des classifications sur les données d'entrée.
- **La couche de perte (LOSS)** : Est utilisée pour évaluer l'écart entre les prédictions du modèle et les valeurs réelles. Elle mesure l'erreur du modèle et est utilisée pour ajuster les poids et les paramètres du réseau lors de l'apprentissage, afin d'optimiser les performances du modèle.

En résumé, les CNN utilisent ces différentes couches pour traiter les données de manière séquentielle et extraire des caractéristiques significatives. Chaque couche joue un rôle spécifique dans la transformation des données et contribue à la capacité du modèle à apprendre et à effectuer des tâches telles que la classification d'images, la détection d'objets, la segmentation, etc[16].

4- Les couches de CNN :

4-1 Couche de convolution (CONV) :

La couche de convolution est un élément essentiel des réseaux neuronaux convolutifs (CNN) et est généralement la première couche de ces réseaux.

Son objectif est de détecter la présence d'un ensemble de caractéristiques dans les images en entrée. Pour cela, elle effectue une opération de filtrage par convolution : elle fait "glisser" une fenêtre représentant la caractéristique souhaitée sur l'image et calcule le produit de convolution entre la fenêtre et chaque région balayée de l'image. Dans ce contexte, une caractéristique est considérée comme un filtre, les deux termes étant équivalents.

La couche de convolution reçoit plusieurs images en entrée et effectue la convolution de chaque image avec chaque filtre. Les filtres correspondent exactement aux caractéristiques que l'on souhaite détecter dans les images.

Pour chaque paire (image, filtre), on obtient une carte d'activation, également appelée carte de caractéristiques, qui indique l'emplacement des caractéristiques dans l'image : plus la valeur est élevée, plus la région correspondante de l'image ressemble à la caractéristique.

Contrairement aux méthodes traditionnelles, les caractéristiques ne sont pas prédéfinies selon un formalisme particulier, mais sont apprises par le réseau pendant la phase d'entraînement. Les noyaux des filtres représentent les poids de la couche de convolution. Ils sont initialisés puis mis à jour à l'aide de la rétro propagation du gradient.

C'est là toute la puissance des réseaux neuronaux convolutifs : ils sont capables de déterminer automatiquement les éléments discriminants d'une image en s'adaptant au problème donné. Par exemple, si le problème consiste à distinguer les chats des chiens, les caractéristiques automatiquement apprises peuvent décrire la forme des oreilles ou des pattes[17].

Lors de l'utilisation d'un CNN, il existe trois hyper paramètres importants que nous devons choisir :

Profondeur de la couche : il s'agit du nombre de noyaux de convolution utilisés, ce qui correspond également au nombre de neurones associés à un même champ récepteur. Une profondeur plus élevée permet au réseau d'apprendre des caractéristiques plus complexes.

Le pas (Stride) : il contrôle le chevauchement des champs récepteurs lors de la convolution. Un pas plus petit entraîne un chevauchement plus important et génère un volume de sortie plus grand.

La marge à 0 (Rembourage) ou zéro padding : parfois, il est utile d'ajouter des zéros autour du volume d'entrée. La taille de ce "zéro padding" est le troisième hyper paramètre. Cette marge permet de contrôler la dimension spatiale du volume de sortie. En conservant la même surface que celle du volume d'entrée, elle peut aider à préserver les informations sur les bords de l'image.

En résumé, la couche de convolution est essentielle dans un CNN. Elle effectue la convolution des images avec des filtres pour extraire des caractéristiques importantes. Les hyper paramètres tels que la profondeur, le pas et la marge jouent un rôle crucial dans la manière dont la convolution est appliquée et dans la dimension du volume de sortie[16].

1	1	1	3
4	6	4	8
30	0	1	5
0	2	2	4

En commençant par le premier filtre, l'opération de convolution commence au coin supérieur gauche de la matrice de la même taille que le filtre de convolution. L'opération de convolution est la somme des produits des éléments qui sont situés sur les mêmes positions des deux matrices. Le résultat de 7 dans la matrice de résultat est trouvé comme suit: $(1 \times 1) + (1 \times 0) + (4 \times 0) + (6 \times 1) = 7$

1	1	1	3
4	6	4	8
30	0	1	5
0	2	2	4

$$* \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} =$$

7			

Une autre opération de convolution est menée pour la prochaine sous-matrice :

1	1	1	3
4	6	4	8
30	0	1	5
0	2	2	4

$$* \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} =$$

7	5		

De la même manière, la troisième opération de convolution est menée, comme il est montré dans la figure

1	1	1	3
4	6	4	8
30	0	1	5
0	2	2	4

$$* \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} =$$

7	5	9	

Une fois la rangée du haut terminée, la rangée suivante recommence à partir de la gauche .

1	1	1	3
4	6	4	8
30	0	1	5
0	2	2	4

$$* \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} =$$

7	5	9	
4			

Le même processus se répète jusqu'à ce que la carte des caractéristiques du filtre donné soit produite, comme le montre la figure

1	1	1	3
4	6	4	8
30	0	1	5
0	2	2	4

$$* \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} =$$

7	5	9	
4	7	9	
32	2	5	

1	1	1	3
4	6	4	8
30	0	1	5
0	2	2	4

$$* \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} =$$

7	5	9
4	7	9
32	2	5

4-2 Couche Pooling :

Le regroupement de couches est une opération couramment utilisée après une couche convolutive dans un réseau de neurones convolutif (CNN). Cette opération permet de réduire la dimensionnalité des caractéristiques extraites et d'introduire une certaine invariance aux translations dans les données.

Il existe principalement deux types de regroupement couramment utilisés : le regroupement maximum (max pooling) et le regroupement moyen (averagepooling).

Dans le regroupement maximum, une fenêtre (généralement de taille 2x2 ou 3x3) glisse sur la carte d'activation générée par la couche convolutive, et la valeur maximale dans chaque fenêtre est sélectionnée pour former une nouvelle carte d'activation réduite. Cela permet de conserver les caractéristiques les plus dominantes tout en réduisant la taille spatiale.

Dans le regroupement moyen, la même fenêtre glisse sur la carte d'activation, mais cette fois-ci, la valeur moyenne des activations dans chaque fenêtre est calculée pour former la nouvelle carte d'activation réduite. Cette opération permet de prendre en compte l'information globale de chaque région et de produire une représentation plus lisse.

Le regroupement de couches permet donc de réduire la dimension spatiale des caractéristiques tout en conservant les informations les plus importantes. Cela aide à réduire le nombre de paramètres et de calculs nécessaires dans le réseau, tout en introduisant une certaine invariance aux translations dans les données [20].

Type	Max pooling	Average pooling
But	Chaque opération de pooling sélectionne la valeur maximale de la surface	Chaque opération de pooling sélectionne la valeur moyenne de la surface
Illustration		
Commentaires	<ul style="list-style-type: none"> • Garde les caractéristiques détectées • Plus communément utilisé 	<ul style="list-style-type: none"> • Sous-échantillonne la <i>feature map</i> • Utilisé dans LeNet

- **Types de pooling :**

- Le « max pooling » :** qui revient à prendre la valeur maximale de la sélection. C'est le type le plus utilisé car il est rapide à calculer (immédiat), et permet de simplifier efficacement l'image
- Le « meanpooling » (ou averagepooling) :** soit la moyenne des pixels de la sélection : on calcule la somme de toutes les valeurs et on divise par le nombre de valeurs. On obtient ainsi une valeur intermédiaire pour représenter ce lot de pixels
- Le « sumpooling » :** c'est la moyenne sans avoir divisé par le nombre de valeurs (on ne calcule que leur somme) [19]

4-3 Couche de correction Relu :

Il est possible d'améliorer l'efficacité du traitement en intercalant entre les couches de traitement une couche qui va opérer une fonction mathématique (fonction d'activation) sur les signaux de sortie.

La fonction ReLU ($F(x) = \max(0, x)$) Cette fonction force les neurones à retourner des valeurs positives [21].

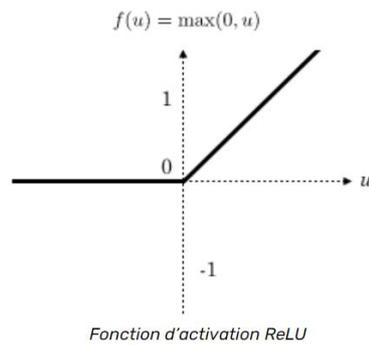


Figure II.3: Fonction d'activation Relu

4-4 Couche entièrement connectée(FC) :

Après plusieurs couches de convolution et de max -pooling, le raisonnement de haut niveau dans le réseau neuronal se fait via des couches entièrement connectées. Les neurones dans une couche entièrement connectée ont des connexions vers toutes les sorties de la couche précédente (comme on le voit régulièrement dans les réseaux réguliers de neurones). Leurs fonctions d'activations peuvent donc être calculées avec une multiplication matricielle suivie d'un décalage de polarisation [22].

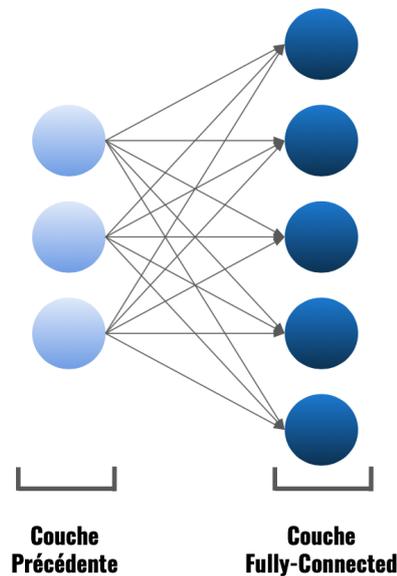


Figure II.4: Une couche fully-connected (FC)

4-5 La couche de perte (LOSS) :

La couche de perte est la dernière couche d'un réseau neuronal, responsable de calculer l'erreur entre les prédictions du réseau et les valeurs réelles. Lorsqu'il s'agit d'une tâche de classification où la variable aléatoire est discrète, avec des valeurs 0 ou 1 représentant

l'appartenance ou non à une classe, la fonction de perte la plus courante et appropriée est l'entropie croisée (cross-entropy en anglais).

L'entropie croisée est une mesure de la différence globale entre deux distributions de probabilité, dans ce cas, celle de la prédiction du modèle et celle des valeurs réelles. Cette fonction est issue du domaine de la théorie de l'information. Formellement, elle est exprimée par l'équation suivante[23] :

$$loss(x, class) = - \sum_{class=1}^c y_{x,class} \log(p_{x,class}) \dots \dots \dots (1.II)$$

5- Architecture générale d'un CNN en 2D et en 3D :

Les réseaux de neurones convolutifs (CNN) sont couramment utilisés pour le traitement d'images en deux dimensions (2D) et en trois dimensions (3D). Voici une description générale de l'architecture des CNN en 2D et 3D :

5-1 Architecture générale d'un CNN en 2D :

- **Couche d'entrée** : Cette couche reçoit les images en 2D en tant qu'entrée pour le réseau.
- **Couches de convolution** : Les couches de convolution sont responsables de l'extraction des caractéristiques de l'image. Chaque couche de convolution applique plusieurs filtres pour détecter des motifs spécifiques dans l'image.

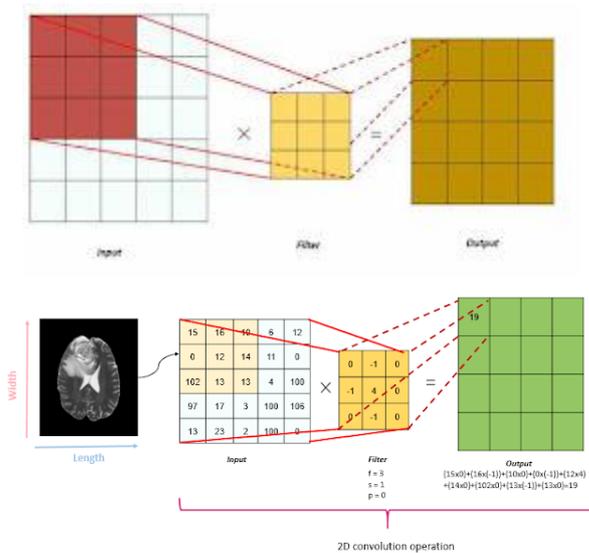


Figure II.5:Opération de convolution 2D

- **Couches de pooling**: Les couches de pooling réduisent la dimension spatiale de la sortie des couches de convolution en effectuant une opération de sous-échantillonnage. Cela permet de réduire le nombre de paramètres du réseau et d'obtenir une représentation plus compacte des caractéristiques.

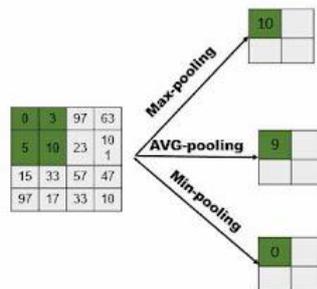


Figure II.6 : 2D pooling opération

- **Couches entièrement connectées** : Les couches entièrement connectées prennent en entrée les caractéristiques extraites par les couches de convolution et les utilisent pour effectuer la classification ou la régression.
- **Couche de sortie** : Cette couche produit la sortie finale du réseau, généralement sous la forme de probabilités pour chaque classe dans le cas d'un problème de classification[9].

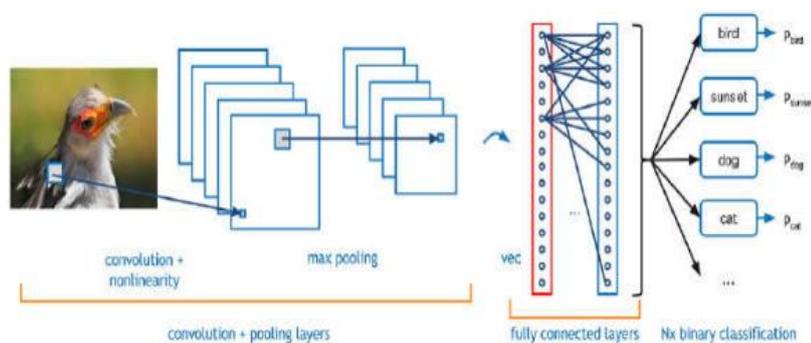


Figure II.7: Structure d'un réseau CNN 2D

5-2 Architecture générale d'un CNN en 3D :

Un réseau neuronal convolutif 3D (CNN) est un type de CNN conçu pour traiter des données d'entrée 3D. Il partage une structure similaire à un CNN 2D mais fonctionne sur des volumes tridimensionnels au lieu d'images bidimensionnelles. Alors qu'un CNN 3D peut fournir des résultats plus précis qu'un CNN 2D en raison de l'inclusion d'informations spatiales et temporelles supplémentaires, il nécessite généralement plus d'espace mémoire et une durée d'exécution plus longue en raison des calculs impliqués dans les convolutions 3D.

La structure d'un CNN 3D se compose de couches convolutives, de couches de regroupement et de couches entièrement connectées, similaires à un CNN 2D. Cependant, au lieu de convoluer des filtres 2D sur des cartes d'entités d'entrée 2D, les filtres 3D sont convolutés sur des volumes d'entrée 3D. Cela permet au réseau de capturer des caractéristiques spatiales dans différentes dimensions, telles que la largeur, la hauteur et la profondeur, ainsi que des caractéristiques temporelles au fil du temps.

La dimensionnalité accrue des données d'entrée dans un CNN 3D conduit à un plus grand nombre de paramètres et de calculs par rapport à un CNN 2D. Par conséquent, la formation et l'inférence avec un CNN 3D peuvent nécessiter plus de ressources de mémoire et de calcul. Cependant, l'avantage d'utiliser un CNN 3D est qu'il peut tirer parti des données d'entrée riches pour fournir des résultats plus précis et détaillés, en particulier lorsqu'il s'agit de données volumétriques ou spatio-temporelles, telles que des séquences vidéo, des analyses médicales ou des représentations 3D d'objets[24].

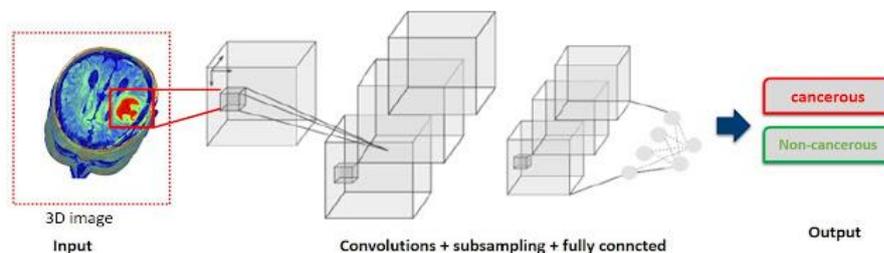


Figure II.8: Architecture générale d'un CNN en 3D

a. Couche de convolution 3D :

La couche de convolution utilise le produit de convolution pour retrouver les caractéristiques incluses dans les signaux de l'image. La sortie de la couche convolutive est

appelée carte de caractéristiques ou carte d'activation. Comme le montre la figure suivante, l'opération de convolution 3D est plus sophistiquée que le processus de convolution 2D. Nécessite plus d'espace et de temps de fonctionnement.

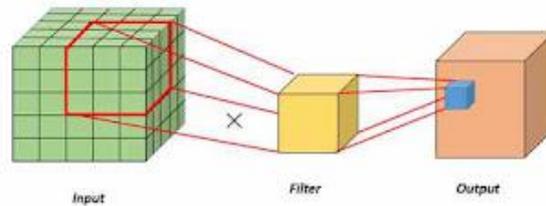


Figure II.9: Couche de convolution 3D

Le résultat de la convolution est calculé selon un filtre (f), un rembourrage (p) et une foulée (s) [24].

- Le filtre : le filtre permet d'analyser l'image zone par zone.
- Le padding représente les pixels (de valeur nulle) à ajouter autour de l'image afin d'éviter la perte d'informations.
- Le paramètre de foulée indique le nombre de pixels à sauter à chaque étape pour procéder au processus de convolution

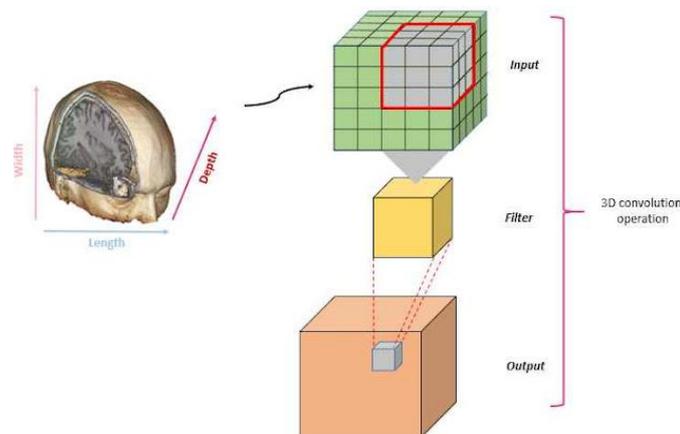


Figure II.10 : opération de convolution 3d à l'aide d'une image 3d

b. Couche de regroupement 3D (pooling):

La couche de regroupement sert à réduire la dimension spatiale de l'image en ne gardant que les pixels les plus descriptifs. Il existe 3 méthodes courantes à utiliser :

- Max-pooling (sélectionnez la valeur la plus élevée)
- Min-pooling (sélectionnez la valeur la plus basse)
- Average-pooling (la moyenne des valeurs).

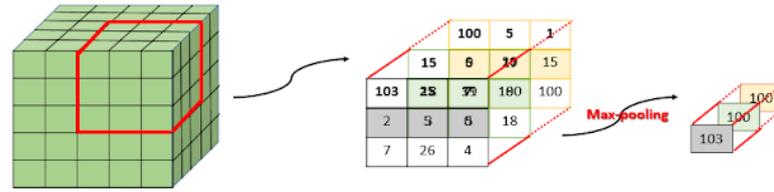


Figure II.11 3D pooling operation

c. Couche entièrement connectée 3D(Fully Connected Layer)

La couche entièrement connectée s'applique à une entrée précédemment aplatie. Il connecte chaque neurone d'une couche à tous les neurones de l'autre couche, il fonctionne comme le réseau de neurones multi-perceptrons. Dans le cas de la classification, le résultat de cette couche est un vecteur qui contient les valeurs de probabilité de la classification.

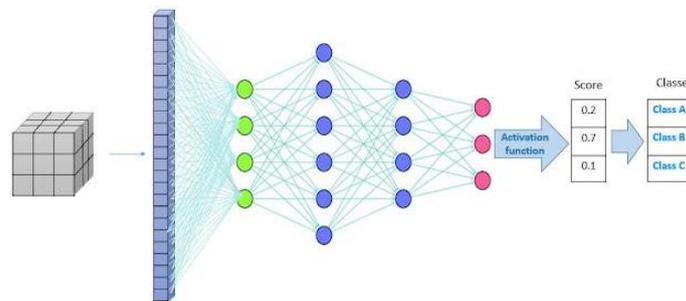


Figure II.123D Couche Entièrement Connectée (Fully Connected Layer) dans un CNN

6 - Exemple de Convolution 2D :

Voici un exemple simple de convolution du signal d'entrée 3x3 et de la réponse impulsionnelle (noyau) dans l'espace 2D. La définition de la convolution 2D et la méthode de convolution en 2D sont expliquées ci-dessous [25].

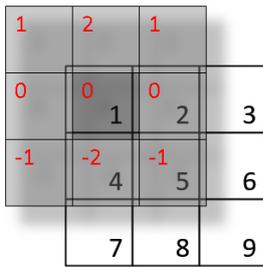
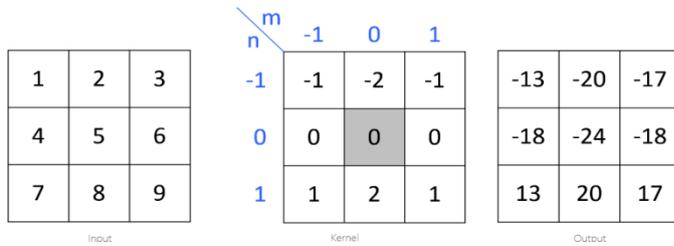
Équation de convolution 2D :

$$y[m, n] = x[m, n] * h[m, n] \dots \dots \dots (2. II)$$

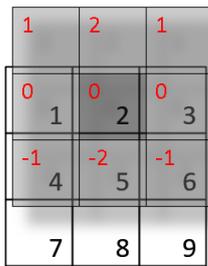
$$= \sum_{j=-a}^a \sum_{i=-b}^b x[i + m, j + n]. h[m, n] \dots \dots \dots (3. II)$$

$$= \sum_{j=-a}^a \sum_{i=-b}^b x[i, j]. h[m - i, n - j] \dots \dots \dots (4. II)$$

-Soit l'exemple de la convolution de la matrice d'entrée (Input) avec un noyau (Kernel) suivant :



$$\begin{aligned}
 y[0,0] &= \sum_j \sum_i x[i,j] \cdot h[0-i,0-j] \\
 &= x[-1,-1] \cdot h[1,1] + x[0,-1] \cdot h[0,1] + x[1,-1] \cdot h[-1,1] \\
 &\quad + x[-1,0] \cdot h[1,0] + x[0,0] \cdot h[0,0] + x[1,0] \cdot h[-1,0] \\
 &\quad + x[-1,1] \cdot h[1,-1] + x[0,1] \cdot h[0,-1] + x[1,1] \cdot h[-1,-1] \\
 &= 0 \cdot 1 + 0 \cdot 2 + 0 \cdot 1 \\
 &\quad + 0 \cdot 0 + 1 \cdot 0 + 2 \cdot 0 \\
 &\quad + 0 \cdot (-1) + 4 \cdot (-2) + 5 \cdot (-1) \\
 &= -13
 \end{aligned}$$



$$\begin{aligned}
 y[1,0] &= \sum_j \sum_i x[i,j] \cdot h[1-i,0-j] \\
 &= x[0,-1] \cdot h[1,1] + x[1,-1] \cdot h[0,1] + x[2,-1] \cdot h[-1,1] \\
 &\quad + x[0,0] \cdot h[1,0] + x[1,0] \cdot h[0,0] + x[2,0] \cdot h[-1,0] \\
 &\quad + x[0,1] \cdot h[1,-1] + x[1,1] \cdot h[0,-1] + x[2,1] \cdot h[-1,-1] \\
 &= 0 \cdot 1 + 0 \cdot 2 + 0 \cdot 1 \\
 &\quad + 1 \cdot 0 + 2 \cdot 0 + 3 \cdot 0 \\
 &\quad + 4 \cdot (-1) + 5 \cdot (-2) + 6 \cdot (-1) \\
 &= -20
 \end{aligned}$$

		1	2	1	
	1	0	2	0	3
	4	-1	5	-2	6
	7		8		9

$$\begin{aligned}
 y[2,0] &= \sum_j \sum_i x[i,j] \cdot h[2-i,0-j] \\
 &= x[1,-1] \cdot h[1,1] + x[2,-1] \cdot h[0,1] + x[3,-1] \cdot h[-1,1] \\
 &\quad + x[1,0] \cdot h[1,0] + x[2,0] \cdot h[0,0] + x[3,0] \cdot h[-1,0] \\
 &\quad + x[1,1] \cdot h[1,-1] + x[2,1] \cdot h[0,-1] + x[3,1] \cdot h[-1,-1] \\
 &= 0 \cdot 1 + 0 \cdot 2 + 0 \cdot 1 \\
 &\quad + 2 \cdot 0 + 3 \cdot 0 + 0 \cdot 0 \\
 &\quad + 5 \cdot (-1) + 6 \cdot (-2) + 0 \cdot (-1) \\
 &= -17
 \end{aligned}$$

1	2	1		
	1	2		3
0	0	4	0	5
-1	-2	7	-1	8

$$\begin{aligned}
 y[0,1] &= \sum_j \sum_i x[i,j] \cdot h[0-i,1-j] \\
 &= x[-1,0] \cdot h[1,1] + x[0,0] \cdot h[0,1] + x[1,0] \cdot h[-1,1] \\
 &\quad + x[-1,1] \cdot h[1,0] + x[0,1] \cdot h[0,0] + x[1,1] \cdot h[-1,0] \\
 &\quad + x[-1,2] \cdot h[1,-1] + x[0,2] \cdot h[0,-1] + x[1,2] \cdot h[-1,-1] \\
 &= 0 \cdot 1 + 1 \cdot 2 + 2 \cdot 1 \\
 &\quad + 0 \cdot 0 + 4 \cdot 0 + 5 \cdot 0 \\
 &\quad + 0 \cdot (-1) + 7 \cdot (-2) + 8 \cdot (-1) \\
 &= -18
 \end{aligned}$$

1	2	1
1	2	3
0	0	0
4	5	6
-1	-2	-1
7	8	9

$$\begin{aligned}
 y[1,1] &= \sum_j \sum_i x[i,j] \cdot h[1-i, 1-j] \\
 &= x[0,0] \cdot h[1,1] + x[1,0] \cdot h[0,1] + x[2,0] \cdot h[-1,1] \\
 &\quad + x[0,1] \cdot h[1,0] + x[1,1] \cdot h[0,0] + x[2,1] \cdot h[-1,0] \\
 &\quad + x[0,2] \cdot h[1,-1] + x[1,2] \cdot h[0,-1] + x[2,2] \cdot h[-1,-1] \\
 &= 1 \cdot 1 + 2 \cdot 2 + 3 \cdot 1 \\
 &\quad + 4 \cdot 0 + 5 \cdot 0 + 6 \cdot 0 \\
 &\quad + 7 \cdot (-1) + 8 \cdot (-2) + 9 \cdot (-1) \\
 &= -24
 \end{aligned}$$

1	2	3
4	5	6
7	8	9

$$\begin{aligned}
 y[2,1] &= \sum_j \sum_i x[i,j] \cdot h[2-i, 1-j] \\
 &= x[1,0] \cdot h[1,1] + x[2,0] \cdot h[0,1] + x[3,0] \cdot h[-1,1] \\
 &\quad + x[1,1] \cdot h[1,0] + x[2,1] \cdot h[0,0] + x[3,1] \cdot h[-1,0] \\
 &\quad + x[1,2] \cdot h[1,-1] + x[2,2] \cdot h[0,-1] + x[3,2] \cdot h[-1,-1] \\
 &= 2 \cdot 1 + 3 \cdot 2 + 0 \cdot 1 \\
 &\quad + 5 \cdot 0 + 6 \cdot 0 + 0 \cdot 0 \\
 &\quad + 8 \cdot (-1) + 9 \cdot (-2) + 0 \cdot (-1) \\
 &= -18
 \end{aligned}$$

7- Convolution 3D selon l'équation :

Dans un réseau de neurones convolutif (CNN) 3D, les équations clés sont similaires à celles d'un CNN 2D, mais elles sont étendues à la troisième dimension pour prendre en compte le volume de données 3D. Voici les équations principales utilisées dans un CNN 3D :

a. Convolution 3D :

La convolution 3D est l'opération de base dans une couche de convolution d'un CNN 3D. Elle consiste à appliquer un ensemble de filtres 3D sur un volume d'entrée 3D pour produire des cartes d'activation. Mathématiquement, la convolution 3D est définie comme suit:

$$y[i, j, k] = \sum_{m=-a}^a \sum_{n=-b}^b \sum_{p=-c}^c x[i + m, j + n, k + p]. h[m, n, p] \dots \dots \dots (5.II)$$

où $y[i, j, k]$ représente la valeur de la carte d'activation à la position $[i, j, k]$, x est le volume d'entrée, h est le filtre 3D, et a, b, c sont les demi-tailles du filtre 3D.

b. Fonction d'activation :

Après la convolution, une fonction d'activation non linéaire est généralement appliquée à chaque carte d'activation pour introduire des non-linéarités dans le modèle. Parmi les fonctions d'activation couramment utilisées dans les CNN, on trouve la fonction ReLU (Rectified Linear Unit) :

$$f(x) = \max(0, x) \dots \dots \dots (6.II) \text{ où } x \text{ est la valeur d'entrée.}$$

c. Pooling 3D :

Le pooling 3D est une opération de sous-échantillonnage qui réduit la taille spatiale du volume tout en conservant les caractéristiques les plus importantes. Le max pooling 3D est une méthode couramment utilisée, où la valeur maximale est sélectionnée dans chaque région de pooling. Mathématiquement, le max pooling 3D est défini comme suit :

$$Y[i, j, k] = \max_{m,n,p} X[rs.i + m, cs.j + n, ds.k + p] \dots \dots \dots (7.II)$$

où $Y[i, j, k]$ représente la valeur après pooling, X est le volume en entrée, et rs, cs, ds sont les facteurs de sous-échantillonnage pour chaque dimension.

Ces équations représentent les opérations principales dans une couche de convolution 3D d'un CNN. En combinant ces opérations avec d'autres couches, telles que les couches de mise en commun, les couches entièrement connectées et les fonctions d'activation, on construit un réseau de neurones convolutif capable de traiter des volumes de données 3D et d'apprendre des caractéristiques complexes à partir de ces données.

8-Conclusion :

Nous avons consacré ce chapitre à l'introduction du concept de réseaux neuronaux convolutifs et à leur rôle dans la reconnaissance d'images.

Nous avons également abordé les méthodes de classification et les types de couches des réseaux neuronaux, ainsi que l'architecture générale des réseaux neuronaux convolutifs, en mentionnant leurs caractéristiques et leurs avantages en 2D et en 3D.

Dans le troisième chapitre, nous aborderons le sujet de la classification de ses types, et des critères de performances d'un classifieur.

Chapitre 3
Classification

1-Introduction :

La classification est le processus de classification des éléments en groupes spécifiques, et dans le contexte de l'apprentissage automatique, cette classification est effectuée par un ordinateur. Pensez à quel point ce serait génial si votre ordinateur pouvait faire la différence entre vous et un étranger, ou distinguer une pomme de terre d'une tomate, ou décider si une certaine performance méritait un A ou un F. Soudain, le concept devient beaucoup plus intéressant. Dans l'apprentissage automatique supervisé, la classification est une tâche essentielle qui implique l'utilisation de données étiquetées pour l'apprentissage. Dans les domaines de l'apprentissage automatique et des statistiques, la classification fait référence au problème consistant à attribuer une nouvelle observation à l'une de plusieurs classes ou sous-populations prédéfinies. Cette détermination est basée sur un ensemble de données de d'entraînement, où les observations sont connues pour appartenir à des catégories spécifiques[26]

2- Définition :

La classification est une discipline largement utilisée dans de nombreux domaines. Elle est souvent connue sous différents termes tels que classification, segmentation et regroupement. Pour donner une définition précise de la classification, il est nécessaire de comprendre ses racines, qui dérivent du verbe "classer", englobant plus qu'un seul domaine.

En mathématiques, la classification est la catégorisation des objets. Elle consiste à attribuer une classe à chaque objet ou individu à classer, sur la base de données d'apprentissage. Les méthodes d'apprentissage sont couramment utilisées pour accomplir cette tâche [27].

3- Les étapes d'une classification[27] :

- Sélection des données à utiliser.
- Calcul de la similarité entre les n individus en se basant sur les données initiales.
- Choix d'un algorithme de classification et exécution de celui-ci.
- Interprétation des résultats obtenus.
- Évaluation de la qualité de la classification.
- Description des classes obtenues.

4-Différent type de classification[26] :

4-1 Classification binaire : la classification binaire consiste à catégoriser les données en deux classes ou catégories distinctes. C'est la forme de classification la plus simple, où le but est d'affecter chaque point de données à l'une des deux classes prédéfinies. Par exemple, déterminer si un e-mail est un spam ou non, classer une transaction comme frauduleuse ou légitime, ou prédire si un patient a ou non une certaine condition médicale.

4-2 Classification multi classe : la classification multi classe implique la catégorisation des données en plus de deux classes ou catégories. Dans ce type de classification, l'objectif est d'affecter chaque point de données à l'une de plusieurs classes prédéfinies. Par exemple, classer des images en différents types d'animaux (chat, chien, oiseau, etc.), reconnaître des chiffres manuscrits (0-9) ou identifier le genre d'une chanson (rock, pop, jazz, etc.). Les problèmes de classification multi classe peuvent être résolus à l'aide de divers algorithmes tels que les arbres de décision, la régression logistique, les machines à vecteurs de support ou les modèles d'apprentissage en profondeur comme les réseaux de neurones profond.

Ces deux types de classification fournissent une base pour organiser et analyser les données en fonction de leurs catégories ou classes distinctes

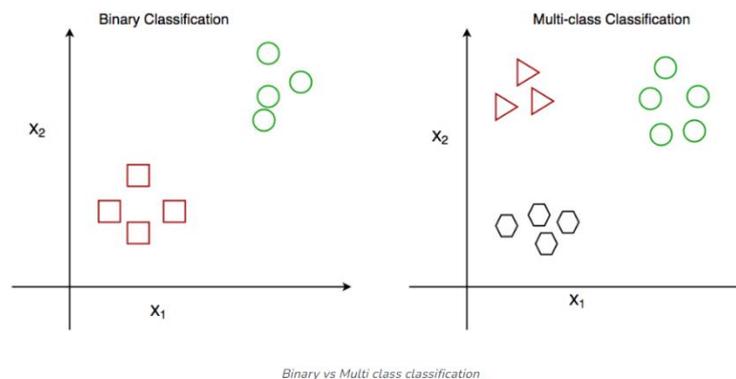


Figure III.1: Types de classification

5 - Différentes approches de classification :

Il existe différentes approches en matière de classification et d'apprentissage automatique.

Voici quelques distinctions courantes [28] :

5-1 Apprentissage supervisé : Dans l'apprentissage supervisé, un modèle est entraîné à partir d'un ensemble de données étiquetées, où chaque exemple est associé à une étiquette ou à un résultat correct. L'objectif est de prédire les étiquettes des nouvelles données non étiquetées. Cela est couramment utilisé pour la classification, la régression et d'autres tâches de prédiction.

5-2 Apprentissage non supervisé : Contrairement à l'apprentissage supervisé, l'apprentissage non supervisé utilise un ensemble de données non étiquetées. L'objectif est de découvrir des structures, des patterns ou des relations inhérentes aux données. Les algorithmes de regroupement (clustering) et les techniques de réduction de dimensionnalité font partie de l'apprentissage non supervisé.

5-3 Apprentissage par renforcement : Dans l'apprentissage par renforcement, un agent interagit avec un environnement et apprend à prendre des actions afin de maximiser une récompense cumulative. L'agent explore l'environnement, reçoit des récompenses ou des pénalités en fonction de ses actions, et ajuste sa politique pour prendre des décisions optimales.

5-4 Apprentissage semi-supervisé : L'apprentissage semi-supervisé est une combinaison d'apprentissage supervisé et non supervisé. Il utilise à la fois des données étiquetées et non étiquetées pour entraîner le modèle. Cela peut être utile lorsque l'obtention de données étiquetées est coûteuse ou difficile, mais que des données non étiquetées sont disponibles en abondance.

6 - Principe de fonctionnement de la classification :

L'objectif principal est de construire un modèle en apprenant les relations entre les données d'entrée et les étiquettes de sortie correspondantes. Cette phase de formation permet au modèle de comprendre les caractéristiques de chaque classe et de développer un processus de prise de décision basé sur les exemples étiquetés fournis.

Une fois le modèle formé, il peut être utilisé pour faire des prédictions pour de nouvelles instances, même si leurs étiquettes n'étaient pas initialement connues. Le but ultime de la classification est de généraliser les connaissances acquises à partir des données d'entraînement et de faire des prédictions précises pour les données non étiquetées, élargissant ainsi la capacité du modèle au-delà de l'ensemble de données étiqueté fourni.

7 - Domaine d'applications des algorithmes de classification :

Les algorithmes de classification trouvent de nombreuses applications pratiques dans divers domaines, tels que :

Filtrage des courriers indésirables : Les techniques de classification sont utilisées pour distinguer les courriers indésirables des courriers légitimes, permettant ainsi de filtrer efficacement les spams.

Évaluation du risque de crédit : Les banques et les institutions financières utilisent des modèles de classification pour évaluer le risque de défaut de remboursement d'un emprunteur, aidant ainsi à prendre des décisions éclairées en matière de prêt.

Diagnostic médical : Les algorithmes de classification sont appliqués dans le domaine médical pour aider au diagnostic de maladies. Ils peuvent être utilisés pour prédire la présence de certaines affections ou pour classer les patients en fonction de leur probabilité de développer une maladie spécifique.

Classement des images : Les techniques de classification sont utilisées pour classer et organiser des images en fonction de leur contenu, ce qui facilite la recherche et l'accès aux images pertinentes.

Analyse des sentiments : Les modèles de classification sont utilisés dans l'analyse des sentiments pour classer les textes selon des catégories telles que positif, négatif ou neutre. Cela permet d'extraire des informations sur les opinions et les sentiments exprimés dans les médias sociaux, les commentaires des clients, etc.

Détection de fraude : Les algorithmes de classification sont utilisés pour détecter les comportements frauduleux dans les transactions financières, tels que les transactions suspectes sur les cartes de crédit ou les activités de blanchiment d'argent.

Contrôle de qualité : Les techniques de classification sont utilisées dans le domaine industriel pour classer les produits en fonction de leur qualité, en identifiant les défauts et en garantissant la conformité aux normes de qualité.

Systèmes de recommandation : Les modèles de classification sont utilisés dans les systèmes de recommandation pour classer les produits ou les éléments selon les préférences et les comportements des utilisateurs, offrant ainsi des recommandations personnalisées.

Ces applications démontrent la polyvalence et l'importance des algorithmes de classification dans de nombreux domaines, contribuant à l'automatisation des tâches et à la prise de décision basée sur des modèles prédictifs[26].

8 - Métriques de mesure de performance d'un classifieur:

8 -1-Précision des model d'apprentissage [29] :

La précision est une mesure couramment utilisée pour évaluer les performances des modèles de classification en apprentissage automatique. Elle offre une indication de la capacité du modèle à classer correctement les instances positives et négatives. La simplicité de la précision en fait une métrique largement comprise par les utilisateurs finaux et les scientifiques des données.

Cependant, il est important de reconnaître que la précision présente certaines limites. En se concentrant uniquement sur cette métrique, on peut négliger les nuances des erreurs commises par les modèles d'apprentissage automatique.

La précision ne tient compte que du nombre de prédictions correctes par rapport au nombre total de prédictions, sans considérer les faux positifs et les faux négatifs. Par conséquent, un modèle avec une précision élevée peut encore présenter des erreurs significatives, telles que des faux positifs ou des faux négatifs.

Il est donc important de prendre en compte d'autres métriques d'évaluation, telles que le rappel, qui fournit une vision plus complète des performances du modèle. Ces métriques permettent de considérer à la fois les erreurs positives et négatives, offrant une évaluation plus approfondie et nuancée des performances du modèle de classification.

En résumé, bien que la précision soit une métrique simple et largement utilisée, il est nécessaire de l'interpréter avec prudence et de compléter son évaluation par d'autres métriques pour obtenir une vision plus complète des performances du modèle d'apprentissage automatique

➤ La précision :

La précision indique combien du nombre total de prédictions spécifiées comme positives sont correctement affectées. Cette métrique est également connue sous le nom de valeur prédictive positive[30].

Exemple :

		Predicted	
		1	0
Actual	1	TP	FN
	0	FP	TN

Dans la matrice de confusion ces métriques sont calculées de la manière suivante :

$$\text{Précision} = \frac{TP}{(TP + FP)} \dots \dots \dots (1. III)$$

➤ **Le Rappel :** Le rappel est le nombre total de cas positifs réels qui ont été correctement prédits

$$\text{Rappel} = \frac{TP}{(TP + FN)} \dots \dots \dots (2. III)$$

➤ **La mesure F1 :** La mesure F1 combine précision et rappel. Le résultat est la moyenne harmonique des deux valeurs, il est calculé comme suit :

$$F1 = 2 \times (\text{Précision} \times \text{Rappel}) \div (\text{Précision} + \text{Rappel}) \dots \dots \dots (3. III)$$

Soit un exemple ci dessous d'un modèle qui prédit sur la base de certains attributs physiques et comportementaux si un animal était un chien ou un chat.

		Predicted		
		Dog	Cat	
Actual	Dog	24	6	30
	Cat	2	18	20

Voici les résultats si "Chien" est utilisé comme réponse positive.

- Précision = $24 \div (24 + 2) = 0,9231$
- Rappel = $24 \div (24 + 6) = 0,8$
- F1 = $2 \times (0,9231 \times 0,8) \div (0,9231 + 0,8) = 0,8572$

Comme on peut le voir, la valeur F1 se situe entre les valeurs de précision et de rappel.

Bien que la précision F1 ne soit pas aussi facile à comprendre, elle ajoute de la nuance au nombre de précision de base[30].

8-2 - Les avantages de l'utilisation de la précision comme métrique sont les suivants :

❖ **Facilité de mise en œuvre** : La précision est une métrique simple à calculer, ce qui en facilite l'implémentation.

❖ **Compréhension aisée** : La précision est largement comprise par de nombreux utilisateurs finaux et scientifiques des données, ce qui facilite la communication des résultats.

Cependant, il existe également des inconvénients à l'utilisation de la précision comme métrique :

❖ **Performances sur des ensembles de données déséquilibrés** : Lorsque les classes sont déséquilibrées, c'est-à-dire lorsque l'une des classes est beaucoup plus fréquente que l'autre, la précision peut être trompeuse. Elle peut donner une fausse impression de bonnes performances en se concentrant uniquement sur la classe majoritaire, négligeant ainsi les erreurs commises sur la classe minoritaire.

❖ **Manque de distinction entre précision et rappel** : La précision mesure la capacité d'un modèle à classer correctement les instances positives, mais elle ne tient pas compte du nombre d'instances positives réellement identifiées. Par conséquent, elle ne fournit pas d'informations sur la capacité du modèle à détecter toutes les instances positives, ce qui est mesuré par le rappel.

8-3 - Over all Accuracy (précision global): La précision globale est la probabilité qu'un individu soit correctement classé par un test, c'est-à-dire la somme des vrais positifs et des vrais négatifs divisée par le nombre total d'individus testés[32].

$$OA = \frac{\text{nombre d'échantillons correctement classés}}{\text{nombre d'échantillons}} \dots \dots \dots (4. III)$$

Exemple :

		Validation		Sum
		Forest	Non-For.	
Class.	Forest	6	0	6
	Non-For.	2	1	3
Sum		8	1	9

$$OA = \frac{6 + 1}{9} = 0.78 = 78\%$$

8-4 - Le coefficient Kappa :

Le coefficient Kappa est calculé à partir d'un test statistique utilisé pour évaluer l'exactitude d'une classification. Il mesure essentiellement la performance de la classification par rapport à une attribution aléatoire des valeurs, c'est-à-dire s'il y a une amélioration par rapport au hasard. Le coefficient Kappa peut varier de -1 à 1. Une valeur de 0 indique que la classification n'est pas meilleure qu'une classification aléatoire. Une valeur négative indique que la classification est significativement pire que le hasard. Une valeur proche de 1 indique que la classification est significativement meilleure que le hasard.

Ainsi, le coefficient Kappa est une mesure de la concordance entre les prédictions de la classification et les valeurs réelles, en tenant compte de la possibilité de résultats aléatoires. Il fournit une indication de la fiabilité de la classification et permet de déterminer si la performance est significativement meilleure ou pire que ce qui serait attendu par hasard. Plus la valeur de Kappa est proche de 1, plus la classification est considérée comme fiable et meilleure que le hasard.

Il convient de noter que le coefficient Kappa doit être utilisé avec prudence et dans le contexte approprié. Il est recommandé de l'interpréter conjointement avec d'autres métriques d'évaluation pour obtenir une évaluation complète des performances du modèle de classification[32]

$$Kappa = \frac{N \sum_{i=1}^n m_{i,i} - \sum_{i=1}^n (G_i C_i)}{N^2 - \sum_{i=1}^n (G_i C_i)} \dots \dots \dots (5. III)$$

Où :

- *i* est le numéro de classe

- N est le nombre total de valeurs classer
- $m_{i,i}$ est le nombre de valeurs appartenant à la classe de vérité i qui ont également été classées dans la classe i (c'est-à-dire les valeurs trouvées le long de la diagonale de la matrice de confusion)
- C_i est le nombre total de valeurs prédites appartenant à la classe i
- G_i est le nombre total de valeurs de vérité appartenant à la classe i

Dans l'exemple de la matrice de confusion, le coefficient kappa est de 0.5915 [31].

Exemple [33]:

		Predicted class			Total
		C1	C2	C3	
Correct class	C1	15	2	3	20
	C2	7	15	8	30
	C3	2	3	45	50
Total		24	20	56	100

$$Kappa = \frac{100 \cdot (15 + 15 + 45) - (20 \cdot 24 + 30 \cdot 20 + 50 \cdot 56)}{100^2 - (20 \cdot 24 + 30 \cdot 20 + 50 \cdot 56)} = 0.5915$$

8-5 - Algorithme d'optimisation :

8-5-1- Descente du gradient (Gradient Descent) :

La descente de gradient est un algorithme d'optimisation utilisé en apprentissage automatique pour minimiser la fonction de coût $J(\theta)$ d'un modèle en ajustant itérativement ses paramètres θ dans la direction opposée du gradient.

Le gradient $\nabla J(\theta)$ représente la pente de la fonction de coût, et en mettant à jour les paramètres dans la direction du gradient négatif, l'algorithme peut converger vers l'ensemble optimal de paramètres qui correspond le mieux aux données d'apprentissage.

L'équation générale de mise à jour des paramètres pour la descente de gradient est [35] :

$$\theta_{t+1} = \theta_t - \alpha \nabla J(\theta_t) \dots \dots \dots (6. III)$$

θ_t : représente les paramètres du modèle à l'itération t.

α : est le taux d'apprentissage (taille du pas) .

$\nabla J(\theta_t)$: est le gradient de la fonction de coût par rapport aux paramètres.

Il existe différentes variantes de descente de gradient :

8-5-2 Descente de gradient par lots (Batch Gradient Descent) : Dans la descente de gradient par lots, le gradient est calculé à l'aide de l'ensemble des données d'apprentissage. L'équation de mise à jour des paramètres devient :

$$\theta_{t+1} = \theta_t - \alpha \frac{1}{N} \sum_{i=1}^N \nabla J(\theta_t, \mathbf{x}^{(i)}, \mathbf{y}^{(i)}) \dots \dots \dots (7.III)$$

où N est le nombre total d'exemples d'apprentissage et $(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})$ représente le *ième* exemple d'apprentissage[35] .

8-5-3 Descente de gradient stochastique (Stochastic Gradient Descent) : dans la descente de gradient stochastique, le gradient est calculé en utilisant un seul exemple d'apprentissage à la fois. L'équation de mise à jour des paramètres devient :

$$\theta_{t+1} = \theta_t - \alpha \nabla J(\theta_t; \mathbf{x}^{(i)}, \mathbf{y}^{(i)}) \dots \dots \dots (8.III)$$

Où $(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})$ est un exemple d'apprentissage sélectionné au hasard.

8-5-4 Descente de gradient en mini-lot (Mini-Batch Gradient Descent) : La descente de gradient en mini-lot est un compromis entre la descente de gradient par lot et stochastique. Il calcule le gradient à l'aide d'un petit lot d'exemples de formation[35] . L'équation de mise à jour des paramètres devient :

$$\theta_{t+1} = \theta_t - \alpha \frac{1}{B} \sum_{i=1}^B \nabla J(\theta_t, \mathbf{x}^{(i)}, \mathbf{y}^{(i)}) \dots \dots \dots (9.III)$$

où B est la taille du mini-lot et $\{(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})\}_{i=1}^B$ représente le mini-lot d'exemples d'apprentissage.

8-5-5 Descente de gradient en mini-lot avec inertie (Momentum) :

La descente de gradient avec moment, également connue sous le nom d'optimisation avec moment ou simplement moment, est une technique utilisée dans les algorithmes d'optimisation, en particulier dans la formation des réseaux neuronaux profonds. Elle permet

d'accélérer la convergence du processus d'optimisation en ajoutant une fraction du gradient précédent à l'étape de mise à jour actuelle.

Dans la descente de gradient standard, les paramètres d'un modèle sont mis à jour en fonction du gradient négatif de la fonction de perte par rapport à ces paramètres. Toutefois, cette approche peut conduire à une convergence lente, voire à un blocage dans des optima locaux.

Le moment aborde ces problèmes en introduisant un terme de "Momentum" qui accumule les gradients passés et influence l'étape de mise à jour actuelle. Au lieu de considérer uniquement le gradient actuel, le moment prend en compte la direction et l'amplitude des gradients précédents.

Cela permet à l'algorithme d'optimisation de continuer à progresser même lorsque le gradient actuel est faible ou bruité [34].

$$v(t) = \beta \times v(t - 1) + (1 - \beta) \times \delta(t) \dots \dots \dots (10. III)$$

Décomposons-le.

$v(t)$: Le nouveau poids qui est mis à jour à l'itération t

β : Quantité de mouvement constante

$\delta(t)$: Est le gradient à l'itération t

Supposons que la mise à jour du poids à l'itération zéro $t = 0$ est nulle

$$v(0) = 0 \dots \dots \dots (11. III)$$

$$v(1) = \beta \times v(0) + (1 - \beta) \times \delta(1) \dots \dots \dots (12. III)$$

$$v(1) = (1 - \beta) \times \delta(1) \dots \dots \dots (13. III)$$

$$v(2) = \beta \times v(1) + (1 - \beta) \times \delta(2) \dots \dots \dots (14. III)$$

$$v(2) = \beta \times \{(1 - \beta) \times \delta(1)\} + (1 - \beta) \times \delta(2) \dots \dots \dots (15. III)$$

$$v(2) = (1 - \beta)\{\beta \times \delta(1) + \delta(2)\} \dots \dots \dots (16. III)$$

$$v(3) = \beta \times v(2) + (1 - \beta) \times \delta(3) \dots \dots \dots (17. III)$$

$$v(3) = \beta \times \{(1 - \beta)\{\beta \times \delta(1) + \delta(2)\}\} + (1 - \beta) \times \delta(3) \dots \dots \dots (18. III)$$

$$v(3) = (1 - \beta)\{\beta^2 \times \delta(1) + \beta \times \delta(2) + \delta(3)\} \dots \dots \dots (19. III)$$

$$v(n) = (1 - \beta) \sum_{t=1}^n \beta^{n-t} \delta(t) \dots \dots \dots (20. III)$$

Remarque : Dans de nombreux textes le terme $(1 - \beta)$ est remplacé par η le taux d'apprentissage (Le pas d'apprentissage).

Focalisons-nous sur la constante β et ignorons le terme $(1 - \beta)$ dans l'équation ci-dessus.

Exemple :

- pour $\beta = 0,1$ et $n = 3$ un gradient à $t = 3$ contribuera à **100 %** de sa valeur, un gradient à $t = 2$ contribuera à **10 %** de sa valeur et un gradient à $t = 1$ ne contribuera qu'à **1 %** de sa valeur on remarque que la contribution des gradients précédents diminue rapidement.
- pour $\beta = 0,9$ et $n = 3$ un gradient à $t = 3$ contribuera à **100 %** de sa valeur à $t = 2$ contribuera à **90 %** de sa valeur et un gradient à $t = 1$ contribuera à **81 %** de sa valeur.

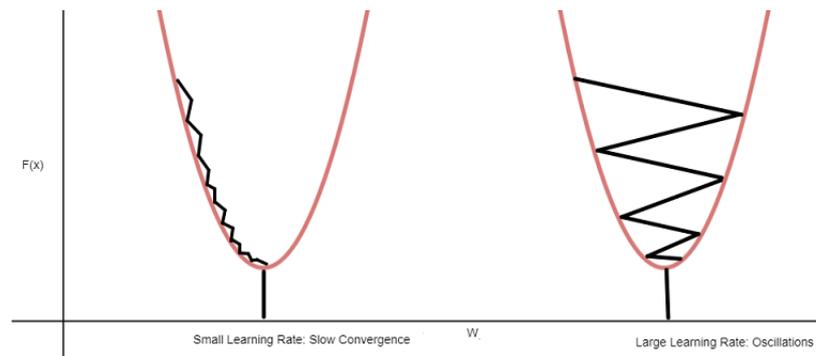
Cela nous amène à conclure que le sommet absorbera plus de gradations du passé. Ainsi, en général, il est maintenu autour de **0,9** dans la plupart des cas.

Remarque : la contribution réelle de chaque gradient de mise à jour des poids dépend du taux d'apprentissage.

Cela répond au premier point que nous avons soulevé : lorsque le gradient actuel est minimal ou nul, l'apprentissage devient nul. En utilisant le *Momentum* avec la descente de gradient, les gradients passés pousseront le coût plus loin pour se déplacer autour du point de selle.

Dans la surface de coût illustrée ci-dessus, zoomons sur le point C.

Avec la descente de gradient, si le taux d'apprentissage est trop faible, les poids seront mis à jour très lentement, de sorte que la convergence prend beaucoup de temps, même lorsque le gradient est élevé. Ceci est illustré dans l'image de gauche ci-dessous. Si le taux d'apprentissage est trop élevé, le coût fluctue autour d'un minimum, comme le montre l'image de droite ci-dessous [34].



- Comment le **Momentum** résout-il cela ?

Regardons à nouveau notre dernière équation d'addition pour la quantité de mouvement.

Cas 1 : Lorsque toutes les gradations précédentes ont le même signe

Le terme positif va être important et nous ferons de grands pas en mettant à jour les pondérations. Le long de la courbe BC, même si le taux d'apprentissage est faible, tous les gradients le long de la courbe auront la même direction (signe) augmentant ainsi l'élan et accélérant la descente.

Cas 2 : Lorsque certains gradients ont un signe ($+v\mathbf{g}$) tandis que d'autres ont un signe ($-v\mathbf{g}$)

Le terme de sommation deviendra petit et les mises à jour de poids seront petites. Si le taux d'apprentissage est élevé, le gradient à chaque itération autour de la vallée C changera de signe entre ($+v\mathbf{g}$) et ($-v\mathbf{g}$) et après quelques oscillations la somme des gradients précédents deviendra petite. Ainsi, faire de petites mises à jour dans les poids à partir de là et amortir les oscillations.

Cela répond un peu à notre deuxième problème. Un gradient avec impulsion fait de petits pas dans les directions où les gradients oscillent et fait de grands pas le long de la direction où les gradients précédents ont la même direction (même signe).

- **Conclusion :**

En ajoutant un terme d'impulsion dans la descente de gradient, les gradients accumulés des itérations précédentes entraîneront davantage le coût pour se déplacer autour du point de selle même lorsque le gradient actuel est minimal ou nul.

Bien que l'élan avec la régression de gradient converge mieux et plus rapidement, cela ne résout toujours pas tous les problèmes. Tout d'abord, l'hyper paramètre η (taux d'apprentissage) doit être réglé manuellement. Deuxièmement, dans certains cas où, même si le taux d'apprentissage est faible, le terme de quantité de mouvement et le gradient de courant peuvent à eux seuls conduire à des oscillations[34].

Premièrement, le problème du taux d'apprentissage peut être résolu en utilisant d'autres formes de descente de gradient telles que Adaptive Gradient et RMSprop. Deuxièmement, le problème de grande quantité de mouvement peut être résolu davantage en utilisant une variation de descente de gradient basée sur la quantité de mouvement appelée Nesterov Accelerated Gradient Descent.

9-Calcul des poids dans les réseaux de neurones multicouche :

Dans un réseau de neurones artificiels, chaque neurone effectue les étapes suivantes[36] :

- ✓ **Entrées et poids :** Le neurone reçoit des entrées provenant de la couche précédente ou d'autres sources. Chaque entrée, représentée par e_i , est associée à un poids w_i correspondant. Les poids déterminent l'influence de chaque entrée sur le neurone et sont ajustés lors de l'apprentissage du réseau.
- ✓ **Sommation pondérée :** Le neurone calcule la somme pondérée de toutes les entrées en les multipliant par leurs poids respectifs. Cette opération est réalisée en ajoutant le produit de chaque entrée et poids. Ensuite, un seuil w_0 , qui est une constante, est ajouté à la somme pondérée ;

$$A = w_0 + \sum_{i=0}^n e_i \times w_i \dots \dots \dots (21.III)$$

Dans cette formule, $\sum_{i=0}^n$ représente la somme, e_i représente les entrées, w_i représente les poids correspondants et w_0 est le (biais).

- ✓ **Fonction d'activation** : La somme pondérée est ensuite introduite dans une fonction d'activation f . Cette fonction est non linéaire et permet d'introduire de la complexité et de la capacité de représentation dans le neurone. Des exemples de fonctions d'activation courantes sont la sigmoïde, la ReLU (Rectified Linear Unit), la tangente hyperbolique, etc. La fonction d'activation transforme la somme pondérée en une sortie : $S = f(A) \dots \dots \dots (22. III)$
- ✓ **Sortie** : La sortie du neurone, notée A, est calculée par la fonction d'activation. Cette sortie peut être transmise à d'autres neurones dans la couche suivante du réseau ou utilisée à d'autres fins en fonction de la configuration du réseau.

Ces étapes sont répétées pour chaque neurone dans une couche donnée du réseau. Le processus est itéré à travers les différentes couches jusqu'à ce que la sortie finale du réseau soit obtenue.

Il est important de souligner que les réseaux de neurones peuvent avoir des architectures, des fonctions d'activation et des méthodes d'apprentissage variées. La description fournie ici représente le fonctionnement de base, mais il existe de nombreuses variations et améliorations possibles en fonction du problème à résoudre[36].

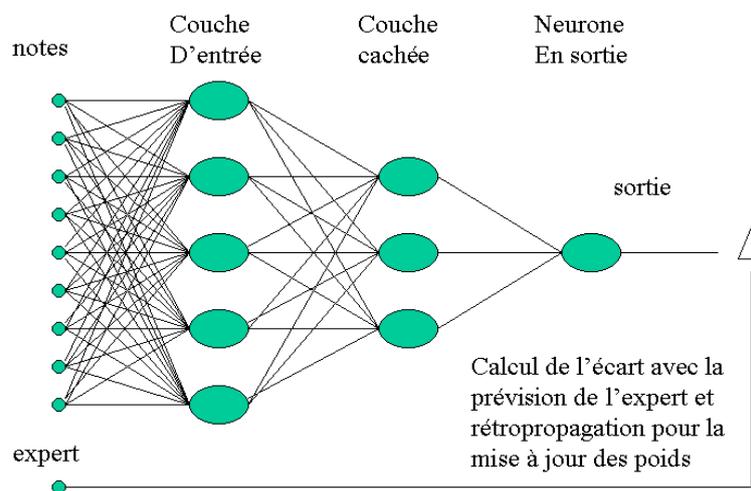


Figure III.2 Les étapes effectuées dans un réseau de neurones pour chaque couche

10 - Ingénierie des réseaux de neurones :

Définissons la structure d'un réseau de neurones. Ici, à des fins de démonstration, un réseau de neurones à deux couches est considéré deux caractéristiques d'entrée $x1$ et $x2$ sont entrées

dans la couche d'entrée. La couche cachée contient trois neurones. Chaque neurone s'est vu attribuer un paramètre de poids ($w_{11}, w_{12}, w_{13}, w_{21}, w_{22}, w_{23}, w_{31}, w_{41}, w_{51}$) comme le montre la figure ci-dessous Figure III.3. Généralement, ces paramètres de poids sont initialisés aléatoirement, b_1 et b_2 sont respectivement le paramètre de polarisation (biais) de la couche d'entrée et de la couche cachée[3].

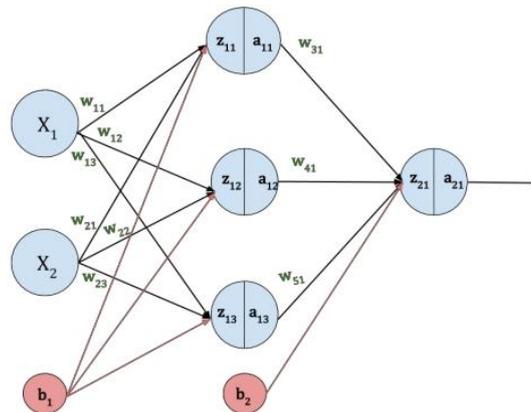


Figure III.3 la structure d'un réseau de neurones.

Il y a 5 étapes pour la mise à jour des paramètres de pondération et de biais dans un réseau de neurones:

- ✓ Initialisation des paramètres de pondération et de biais.
- ✓ Propagation de l'entrée vers l'avant.
- ✓ Calculer de la perte (fonction de coût).
- ✓ retro propagation de l'erreur.
- ✓ Mettre à jour les paramètres de pondération et de biais.

a. Propagation vers l'avant :

Dans la propagation vers l'avant, les données d'entrée sont transmises au réseau dans le sens direct. Chaque couche masquée obtient les données, effectue le calcul et transmet le résultat à la couche suivante. Enfin, la couche de sortie calcule la sortie du modèle.

Examinons les équations mathématiques de la propagation vers l'avant. z_{11}, z_{12}, z_{13} et z_{21} sont les valeurs des neurones intermédiaires, calculées à partir des poids, des biais et des valeurs des neurones de la couche précédente. La fonction d'activation sera appliquée à la valeur des neurones intermédiaires pour déduire des modèles non linéaires entre l'entrée et la variable de

sortie cible. a_{11}, a_{12}, a_{13} et a_{21} sont la sortie de la fonction d'activation appliquée à z_{11}, z_{12}, z_{13} et z_{21} respectivement.

Les équations ci-dessous présentes les équations mathématiques de la propagation vers l'avant et sa représentation vectorielle.

❖ *Equation de propagation vers l'avant :*

$$z_{11} = x_1 w_{11} + x_2 w_{21} + b_1 \dots \dots \dots (23. III)$$

$$z_{12} = x_1 w_{12} + x_2 w_{22} + b_1 \dots \dots \dots (24. III)$$

$$z_{13} = x_1 w_{13} + x_2 w_{23} + b_1 \dots \dots \dots (25. III)$$

$$a_{11} = \sigma(z_{11}) \dots \dots \dots (26. III)$$

$$a_{12} = \sigma(z_{12}) \dots \dots \dots (27. III)$$

$$a_{13} = \sigma(z_{13}) \dots \dots (\sigma - \text{fonction d'activation}) \dots \dots \dots (28. III)$$

$$z_{21} = a_{11} w_{31} + a_{12} w_{41} + a_{13} w_{51} + b_2 \dots \dots \dots (29. III)$$

$$a_{21} = \sigma(z_{21}) \dots \dots \dots (30. III)$$

Représentation vectorielle :

$$\begin{bmatrix} z_{11} \\ z_{12} \\ z_{13} \end{bmatrix} = \begin{bmatrix} w_{11} & w_{21} \\ w_{12} & w_{22} \\ w_{13} & w_{23} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} \dots \dots \dots (31. III)$$

$$z^{[1]} = w^{[1]}x + b^{[1]} \dots \dots \dots (32. III)$$

$$\begin{bmatrix} a_{11} \\ a_{12} \\ a_{13} \end{bmatrix} = \sigma \begin{bmatrix} z_{11} \\ z_{12} \\ z_{13} \end{bmatrix} \dots \dots \dots (33. III)$$

$$A^{[1]} = \sigma(Z^{[1]}) \dots \dots \dots (34. III)$$

$$[z_{21}] = [w_{31} \quad w_{41} \quad w_{51}] \begin{bmatrix} a_{11} \\ a_{12} \\ a_{13} \end{bmatrix} + [b_2] \dots \dots \dots (35. III)$$

$$z^{[2]} = w^{[2]}A^{[1]} + b^{[2]} \dots \dots \dots (36. III)$$

$$\begin{bmatrix} a_{21} \end{bmatrix} = \sigma \begin{bmatrix} z_{21} \end{bmatrix} \dots \dots \dots (37. III)$$

$$a_{21} = \sigma[z_{21}] \dots \dots \dots (38. III)$$

$$A^{[2]} = \sigma(Z^{[2]}) \dots \dots \dots (39. III)$$

Calcul des pertes (erreur) :

Fonction de perte (Loss function) :

$$J = -\frac{1}{m} \sum_{i=1}^m L(a^{[2](i)}, y^{(i)}) \dots \dots \dots (40. III)$$

$$L(a^{[2]}, y) = -y \log a^{[2]} - (1 - y) \log(1 - a^{[2]}) \dots \dots \dots (41. III)$$

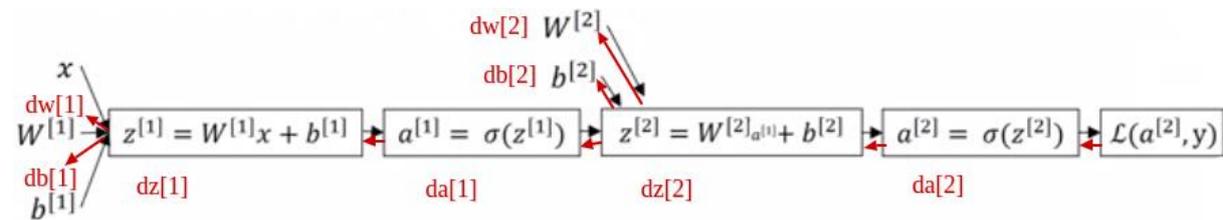
b. Calcul de rétro propagation :

La rétro propagation est le processus de propagation d'une erreur (perte) dans le réseau neuronal et la mise à jour ultérieure des poids de chaque neurone en ajustant les paramètres de poids et de biais. La rétro propagation joue un rôle important dans un réseau neuronal. Elle effectue diverses opérations de calcul pour découvrir des modèles entre la covariable et la variable cible.

L'objectif principal d'un réseau neuronal est d'obtenir l'erreur minimale (perte). Nous pouvons obtenir l'erreur minimale entre la valeur cible réelle et la valeur cible prédite si nous obtenons la valeur correcte pour les paramètres de pondération et de biais.

L'erreur de paramètre a été modifiée. Le taux de variation de l'erreur est déterminé en calculant la dérivation partielle de la fonction de perte par rapport à chaque paramètre.

En effectuant la dérivation, nous pouvons déterminer la sensibilité de la fonction de perte pour chaque paramètre de pondération et de biais. Cette méthode est également connue sous le nom de méthode d'amélioration du gradient[37].



Dans la figure ci-dessus, la ligne rouge représente le processus de rétro propagation où $da[2]$, $dz[2]$, $dw[2]$, $db[2]$, $da[1]$, $dz[1]$, $dw[1]$ et $db[1]$ est la dérivation partielle de la fonction de perte par rapport à $a[2]$, $z[2]$, $w[2]$, $b[2]$, $a[1]$, $z[1]$, $w[1]$ et $b[1]$, respectivement.

Tout en effectuant le calcul de dérivation, nous devons également calculer la dérivation de la fonction d'activation. Prenons l'exemple dérivation de la fonction d'activation sigmoïde.

$$\sigma(x) = \frac{1}{1 + e^{-x}} = (1 + e^{-x})^{-1} \dots\dots\dots (42. III)$$

$$\sigma'(x) = (1 + e^{-x})^{-2} e^{-x} \dots\dots\dots (43. III)$$

$$= \frac{e^{-x}}{(1 + e^{-x})^2} \dots\dots\dots (44. III)$$

$$= \frac{1}{(1 + e^{-x})} \times \frac{1 + e^{-x} - 1}{(1 + e^{-x})} \dots\dots\dots (45. III)$$

$$= \frac{1}{(1 + e^{-x})} \times \left(1 - \frac{1}{(1 + e^{-x})}\right) \dots\dots\dots (46. III)$$

$$= \sigma(x)(1 - \sigma(x)) \dots\dots\dots (47. III)$$

Calculons les dérivées partielle des paramètres de poids et de biais par rapport à ces paramètres.

$$d\mathbf{a}^{[2]} = \frac{\partial}{\partial \mathbf{a}^{[2]}} \left(-y \log \mathbf{a}^{[2]} - (1 - y) \log(1 - \mathbf{a}^{[2]}) \right) \dots\dots\dots (48. III)$$

$$= \frac{-y}{\mathbf{a}^{[2]}} + \frac{(1 - y)}{1 - \mathbf{a}^{[2]}} \dots\dots\dots (49. III)$$

$$d\mathbf{z}^{[2]} = \frac{\partial L}{\partial \mathbf{z}^{[2]}} = \frac{\partial L}{\partial \mathbf{a}^{[2]}} \times \frac{\partial \mathbf{a}^{[2]}}{\partial \mathbf{z}^{[2]}} \dots\dots\dots (50. III)$$

$$= \frac{-y}{\mathbf{a}^{[2]}} + \frac{(1 - y)}{1 - \mathbf{a}^{[2]}} \times \frac{\partial \sigma(\mathbf{z}^{[2]})}{\partial \mathbf{z}^{[2]}} \dots\dots\dots (51. III)$$

$$= \frac{-y(1 - \mathbf{a}^{[2]}) + (1 - y)\mathbf{a}^{[2]}}{\mathbf{a}^{[2]}(1 - \mathbf{a}^{[2]})} \times \mathbf{a}^{[2]}(1 - \mathbf{a}^{[2]}) \dots\dots\dots (52. III)$$

$$= -y + y\mathbf{a}^{[2]} + \mathbf{a}^{[2]} - y\mathbf{a}^{[2]} \dots\dots\dots (53. III)$$

$$= \mathbf{a}^{[2]} - y \dots\dots\dots (54. III)$$

Où: $(\mathbf{a}^{[2]} = \sigma(\mathbf{z}^{[2]}))$

$$d\mathbf{w}^{[2]} = \frac{\partial L}{\partial \mathbf{w}^{[2]}} = \frac{\partial L}{\partial \mathbf{z}^{[2]}} \times \frac{\partial \mathbf{z}^{[2]}}{\partial \mathbf{w}^{[2]}} \dots\dots\dots (55. III)$$

$$= d\mathbf{z}^{[2]} \times \frac{\partial (\mathbf{w}^{[2]}\mathbf{a}^{[2]} + \mathbf{b}^{[2]})}{\partial \mathbf{w}^{[2]}} \dots\dots\dots (56. III)$$

$$= d\mathbf{z}^{[2]} \times \mathbf{a}^{[1]} \dots\dots\dots (57. III)$$

$$d\mathbf{b}^{[2]} = \frac{\partial L}{\partial \mathbf{b}^{[2]}} = \frac{\partial L}{\partial \mathbf{z}^{[2]}} \times \frac{\partial \mathbf{z}^{[2]}}{\partial \mathbf{b}^{[2]}} \dots \dots \dots (58.III)$$

$$= d\mathbf{z}^{[2]} \times \frac{\partial (\mathbf{w}^{[2]}\mathbf{a}^{[1]} + \mathbf{b}^{[2]})}{\partial \mathbf{b}^{[2]}} \dots \dots \dots (59.III)$$

$$= d\mathbf{z}^{[2]} \dots \dots \dots (60.III)$$

$$d\mathbf{a}^{[2]} = \frac{\partial L}{\partial \mathbf{a}^{[1]}} = \frac{\partial L}{\partial \mathbf{z}^{[2]}} \times \frac{\partial \mathbf{z}^{[2]}}{\partial \mathbf{a}^{[1]}} \dots \dots \dots (61.III)$$

$$= d\mathbf{z}^{[2]} \times \frac{\partial (\mathbf{w}^{[2]}\mathbf{a}^{[1]} + \mathbf{b}^{[2]})}{\partial \mathbf{a}^{[1]}} \dots \dots \dots (62.III)$$

$$= d\mathbf{z}^{[2]} \times \mathbf{w}^{[2]} \dots \dots \dots (63.III)$$

$$d\mathbf{z}^{[1]} = \frac{\partial L}{\partial \mathbf{z}^{[1]}} = \frac{\partial L}{\partial \mathbf{a}^{[1]}} \times \frac{\partial \mathbf{a}^{[1]}}{\partial \mathbf{z}^{[1]}} \dots \dots \dots (64.III)$$

$$= (d\mathbf{z}^{[2]} \times \mathbf{w}^{[2]}) \times \frac{\partial \sigma(\mathbf{z}^{[1]})}{\partial \mathbf{z}^{[1]}} \dots \dots \dots (65.III)$$

$$= (d\mathbf{z}^{[2]} \times \mathbf{w}^{[2]}) \times \sigma'(\mathbf{z}^{[1]}) \dots \dots \dots (66.III)$$

$$d\mathbf{w}^{[1]} = \frac{\partial L}{\partial \mathbf{w}^{[1]}} = \frac{\partial L}{\partial \mathbf{z}^{[1]}} \times \frac{\partial \mathbf{z}^{[1]}}{\partial \mathbf{w}^{[1]}} \dots \dots \dots (67.III)$$

$$= d\mathbf{z}^{[1]} \times \frac{\partial (\mathbf{w}^{[1]}\mathbf{x} + \mathbf{b}^{[1]})}{\partial \mathbf{w}^{[1]}} = d\mathbf{z}^{[1]} \times \mathbf{X} \dots \dots \dots (68.III)$$

$$d\mathbf{b}^{[1]} = \frac{\partial L}{\partial \mathbf{b}^{[1]}} = \frac{\partial L}{\partial \mathbf{z}^{[1]}} \times \frac{\partial \mathbf{z}^{[1]}}{\partial \mathbf{b}^{[1]}} \dots \dots \dots (69.III)$$

$$= d\mathbf{z}^{[1]} \times \frac{\partial (\mathbf{w}^{[1]}\mathbf{x} + \mathbf{b}^{[1]})}{\partial \mathbf{b}^{[1]}} = d\mathbf{z}^{[1]} \dots \dots \dots (70.III)$$

11 - Conclusion :

Dans ce chapitre, nous avons discuté des concepts de classification, en explorant les divers types de classifications ainsi que les mesures de performance utilisées pour évaluer un classifieur.

Dans le dernier chapitre, sera consacré à l'application d'un classifieur profond (CNN) à un cas concret celui de la base MNSIT d'image de chiffres Manuscrit afin de les reconnaître et de les classer.

Chapitre 4
Implémentation

1-Introduction :

Dans ce chapitre, nous allons utiliser un modèle profond (**CNN**) pour l'apprentissage et la classification d'images numériques, dans notre cas les images de chiffres. Nous chercherons à construire un modèle capable d'extraire automatiquement les caractéristiques les plus pertinentes à partir des images brutes et de réaliser des prédictions précises en minimisant une fonction de coût basée sur la cross-entropie

2 - Logiciels utilisés : MATLAB

Définition : MATLAB est à la fois un langage de programmation et un environnement qui offre des fonctionnalités avancées pour effectuer des calculs numériques. Il permet d'accomplir diverses tâches, notamment la manipulation de matrices, le tracé de fonctions, la visualisation de données, le traitement d'images, l'implémentation de nouveaux algorithmes et la création d'interfaces graphiques. [2]

3 - Optimisateur utiliser : Sgdm (signifie Stochastic Gradient Descent With Momentum)

C'est un type de Descente de Gradient comme nous l'avons mentionné plus tôt et défini dans le chapitre précédent.

4- Partie d'application :

Notre base de donnée **MNIST** pour **Modified** ou **Mixed National Institute of Standards and Technology**, est une base de données de chiffres écrits à la main. Ce sont des images en noir et blanc, normalisées centrées de 28 x 28 pixels elle est constituer d'un lot de 10000 images de différent chiffre de 0 à 9.Cette basse est organier sous forme de 10 lot contenant 1000 images représentant différente image d'un seul chiffre dont voici(Figure IV.1)quelque exemple d' image appartenant à notre base de donnée :

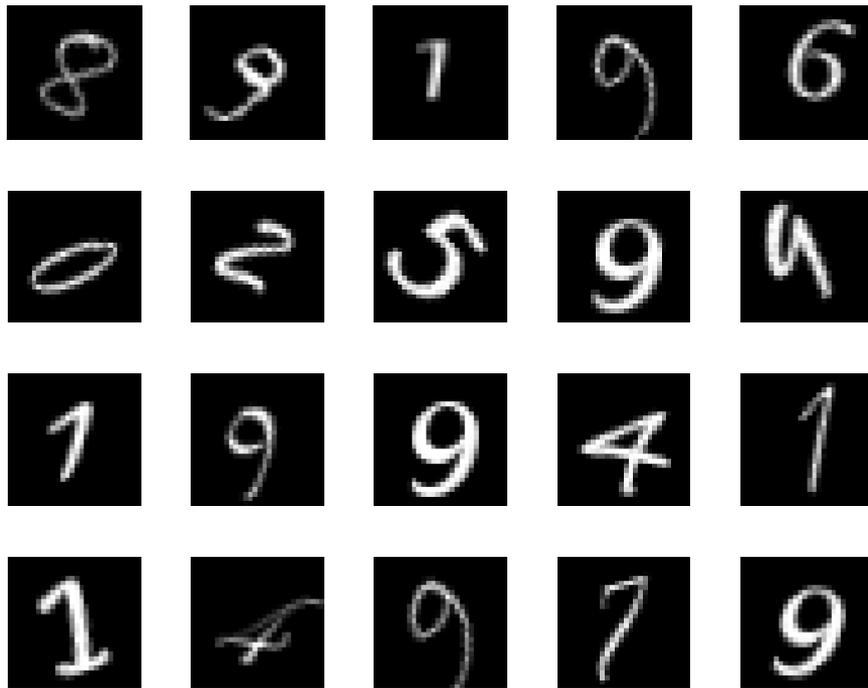


Figure IV.1 Exemple d'image de chiffres appartenant à notre base de données

Notre application comptera deux étapes essentielles :

- 1- Choix du model selon deux critères la profondeur du model et la taille du mini lot
- 2- Validation du model choisi dans l'étape 1 sur différente taille d'ensemble d'apprentissage.

1- Choix du modèle :

On va choisir un modèle linéaire de classifieur CNN se basant sur la profondeur (nombre de block de convolution) et la taille du mini lot (mini batch) moyennant 3 métriques pour juger de la performance du modèle sélectionné.

a- Influence du nombre de block de convolution sur les performances du modèle :

Dans notre expérience en va tester les performances de 7 modèles de CNN en fonction de leur profondeur, pour cela en va fixer la taille du mini lot à 128 échantillons, le pas d'apprentissage à 0.01, la valeur du paramètre d'inertie à 0.9 (Momentum) et 12 époques le but étant de choisir le modèle qui fournit les meilleures performances pour un même ensemble d'apprentissage teste dans notre cas en prendra 75% de l'ensemble d'image de chiffre à classer pour la phase d'apprentissage et 25% pour la phase de test .

Le tableau ci-dessous résume les différentes valeurs des 3 métriques obtenus pour différentes profondeur de model :

Nombre de block De convolution	1	2	3	4	5	6	7
Over All Accuracyx100	0.992	0.998	0.9996	0.9996	0.9988	0.9984	0.9992
AverageAccuracyx100	0.992	0.998	0.9996	0.9996	0.9988	0.9984	0.9992
KAPPA	0.991	0.9978	0.9996	0.9996	0.9978	0.9982	0.9987

Tableau 1 : Les différentes valeurs des 3 métriques obtenus pour différentes profondeur de model

Pour plus de clarté on à reporter les valeurs contenus dans le tableau sur le graphe ci-dessous (Figure IV.2) qui représente l'évolution des valeurs des métriques de performance en fonction du nombre de block convolutif :

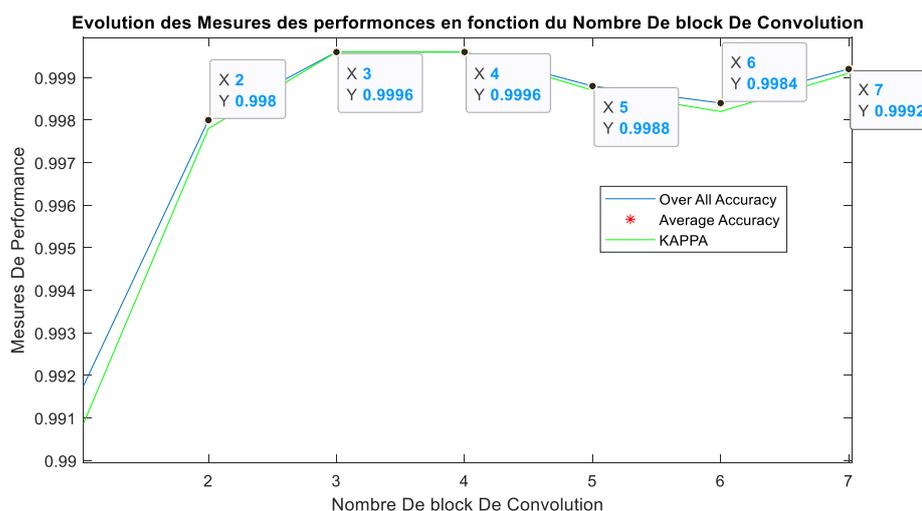


Figure IV.2 Evolution des valeurs des métriques de performance en fonction du nombre de block Convolutif

D'après les mesures des métriques on constate que les model avec trois et quatre blocks de convolutions donnent les meilleurs performances pour les choix décrit précédemment dans la suite dans notre expérience nous allons utiliser le model à trois block, puisque qui dit moins de block dit moins de complexité et moins de temps de calcul donc pour des performances équivalentes en prendra le model le plus simple possible et le moins couteux en temps de calcul.

5-Structure du model CNN choisit :

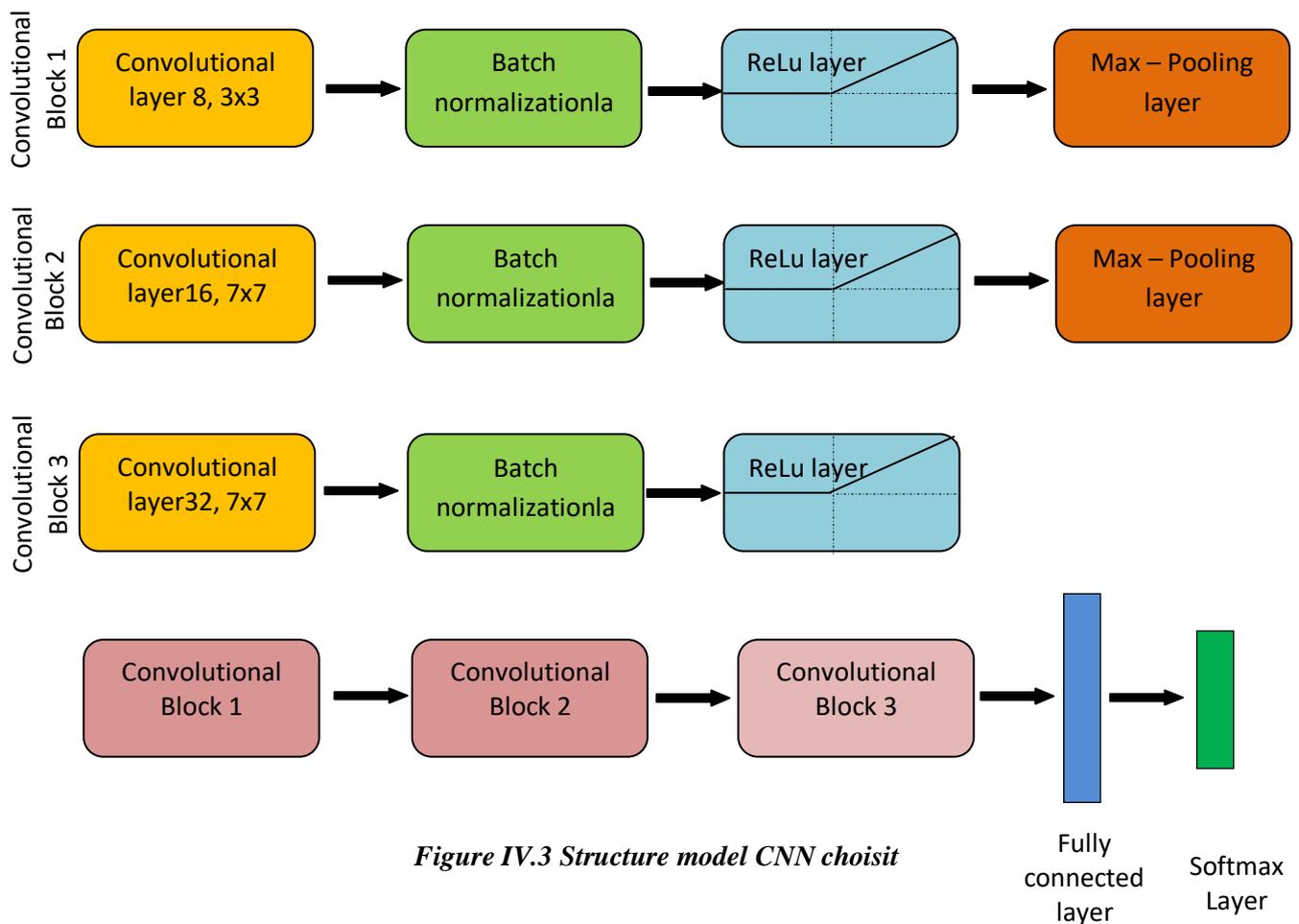


Figure IV.3 Structure model CNN choisit

- Le graphe suivant (Figure IV.4) représente l'évolution de la précision global (Over all Accuracy) et de la fonction de perte (cross entropy) pour le model à 3 block de convolution pour un pas d'apprentissage à 0.01, la valeur du paramètre d'inertie à 0.9 (**Momentum**) et 12 époques

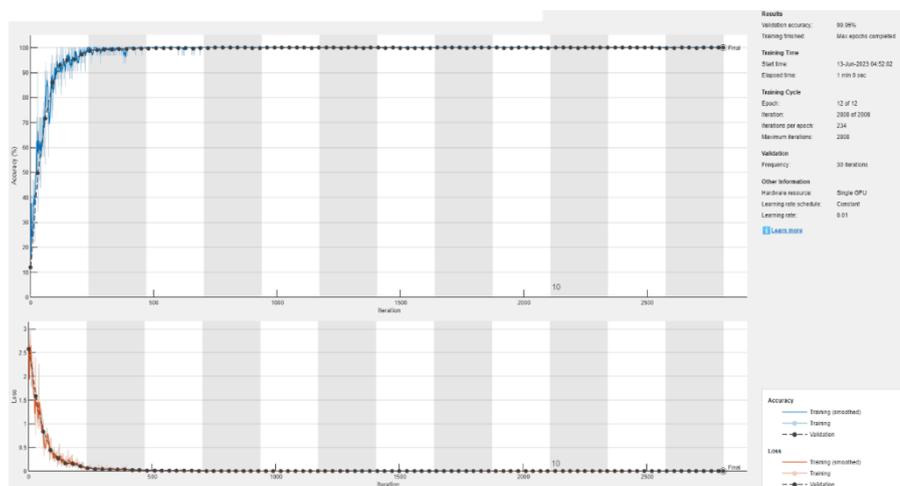


Figure IV.4 l'évolution de la précision global (Over all Accuracy) pour le model à 3 block de convolution

b- Influence de la taille du Mini Lot sur les performances du modèle :

Dans cette expérience on va tester les performances du modèle CNN choisis précédemment pour différentes tailles de mini lot (mini batch) dans notre cas on prendra 32, 64, 128, 256, 512 et 1024 comme taille de mini lot, on garde les mêmes valeurs pour le pas d'apprentissage à 0.01, du paramètre d'inertie à 0.9 (*Momentum*) et 12 époque le but étant de choisir la taille du mini lot qui fournit les meilleures performances pour un même ensemble d'apprentissage teste dans notre cas 75% de l'ensemble d'image de chiffre à classer pour la phase d'apprentissage et 25% pour la phase de test. Le tableau ci-dessous résume les différentes valeurs des 3 métriques obtenus pour différentes taille de mini Lot :

Taille du mini lot	32	64	128	256	512	1024
Over All Accuracy x100	1	0.9996	0.9984	0.9984	0.9952	0.9948
Average Accuracy x100	1	0.9996	0.9984	0.9984	0.9952	0.9948
KAPPA	1	0.9996	0.9982	0.9982	0.9946	0.9916

Tableau 2 : les différentes valeurs des 3 métriques obtenus pour différentes taille de mini lot

Pour plus de clarté on a reporté les valeurs contenues dans le tableau sur le graphe ci-dessous (Figure IV.5) qui représente l'évolution des valeurs des métriques de performance en fonction de la taille du mini lot :

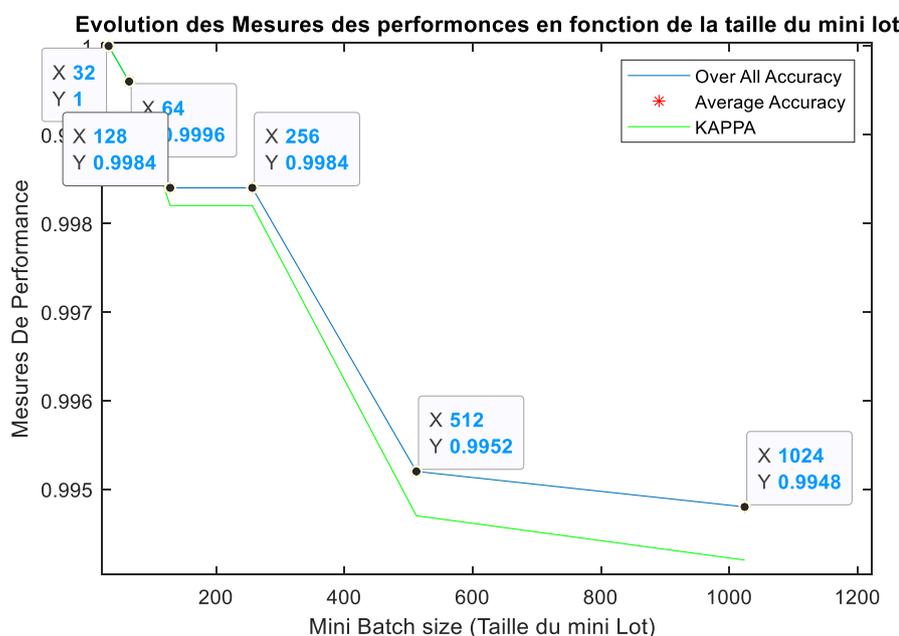


Figure IV.5: Evolution des valeurs des métriques de performance en fonction de la taille du Mini Lot (Mini Batch)

- D'après le graphe on constate que plus la taille de mini batch augmente plus les performances

du model choisi diminues, et la meilleurs performance en terme de métriques utiliser et obtenus

pour la taille de Mini Lot 32 puisqu'on obtient 100 pour cent de reconnaissance pour les 25 % d'images de testes

- Le graphe suivant représente l'évolution de la précision global (Over all Accuracy) et de la fonction de perte (cross entropy) du model sélectionner précédemment pour la taille du mini lot 32 ,pour un pas d'apprentissage à 0.01, la valeur du paramètre d'inertie à 0.9 (*Momentum*) et 12 époques.

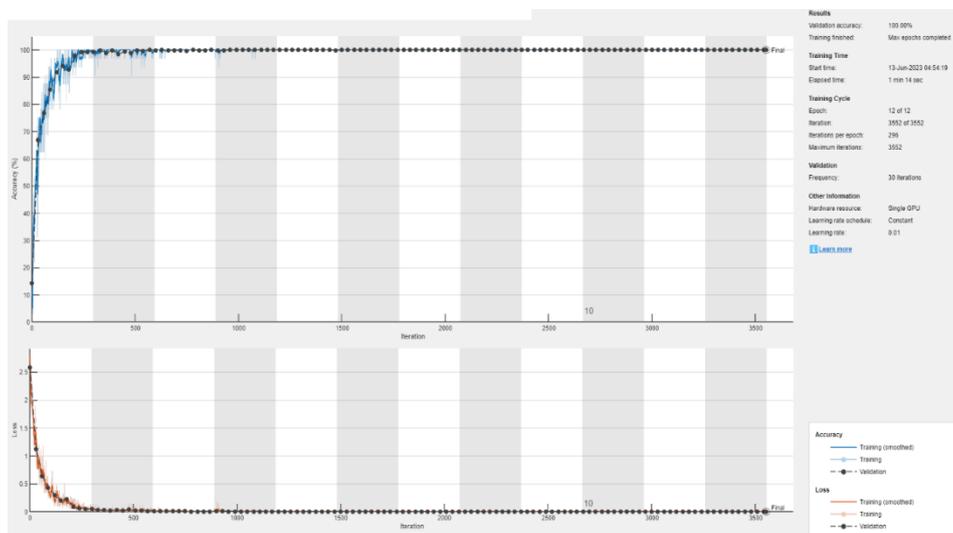


Figure IV.6 : Le graphe suivant représente l'évolution de la précision global et de fonction de perte pour un mini lot de 32

2 –validation du model choisi pour diffèrent taille d'ensemble d'apprentissage :

Pour justifier des bonnes performances du model face à différente portion d'ensemble d'apprentissage, test nous allons utiliser les portions d'ensemble apprentissage résumés dans le tableau ci-dessous et valider le model sur le reste de l'ensemble d'image.

Taille de l'ensemble d'apprentissage (%)	5	15	25	35	45	50	55	65	75	85	95
Over All Accuracy x100	0.8499	0.9775	0.9897	0.9958	0.9989	0.9984	0.9993	0.9994	0.9996	0.9993	1.0000
Average Accuracy x100	0.8499	0.9775	0.9897	0.9958	0.9989	0.9984	0.9993	0.9994	0.9996	0.9993	1.0000
KAPPA	0.8332	0.9750	0.9886	0.9954	0.9988	0.9982	0.9993	0.9994	0.9996	0.9993	1.0000

Tableau 3 : Les différentes valeurs des 3 métriques obtenues pour validation du model choisi pour diffèrent taille d'ensemble d'apprentissage

A partir du tableau nous obtenons le graphe suivant :

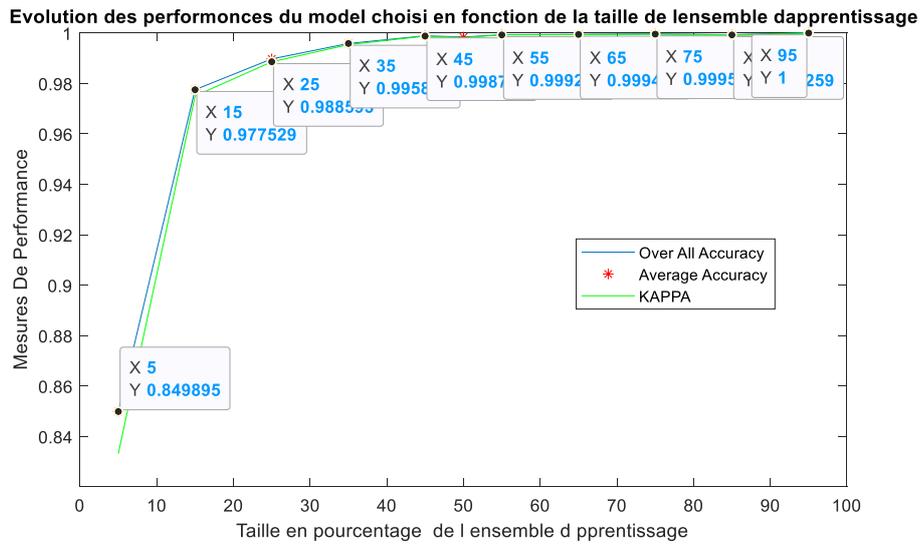


Figure IV.7: Evolution des valeurs des métriques de performance en fonction de la taille de l'ensemble d'apprentissage

- Selon le graphe obtenu Globalement toutes les métriques indiquent et confirment que quelque soit la taille de l'ensemble d'apprentissage notre modèle restera performant proportionnellement à la taille de l'ensemble d'apprentissage, aussi il est évident que plus la taille de l'ensemble d'apprentissage est large plus on aura d'exemples représentatifs d'une classe donnée et plus le modèle aura un plus grand pouvoir de généralisation ce qui est confirmé par les valeurs sur le graphe représentant les mesures des métriques en fonction de la taille de l'ensemble d'apprentissage

- Exemple de reconnaissance du chiffre 8 à partir de son image brute (figureIV.8) :

Voici un exemple de reconnaissance du chiffre 8 à partir de son image brute par le model sélectionner (3 block de convolution et 32 pour la taille du mini lot) dans le quel en va représenter les différentes Sorties avec leur taille respectives pour chaque couche du réseau jusqu'à la reconnaisse du chiffre par le model

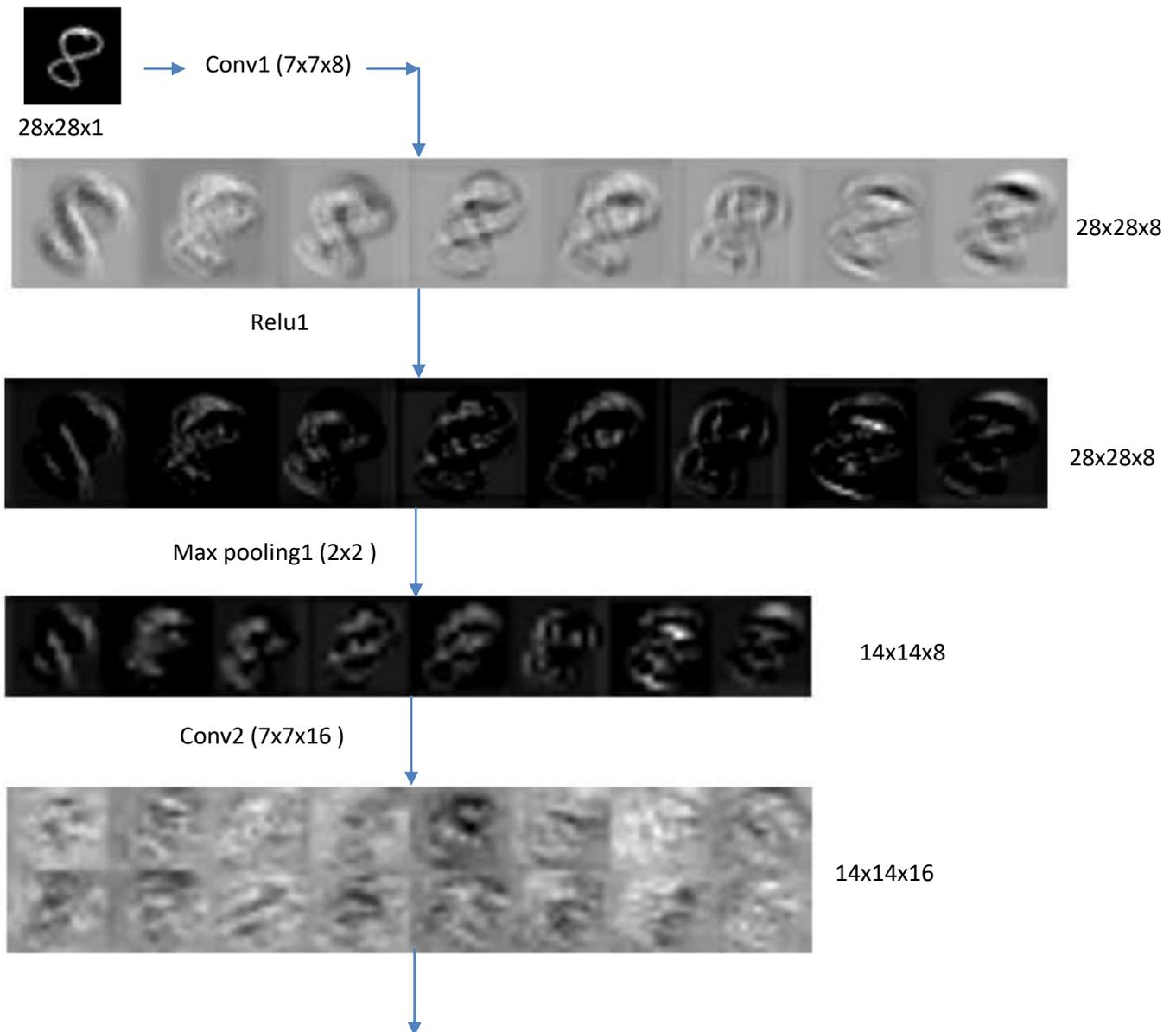


Figure IV.8

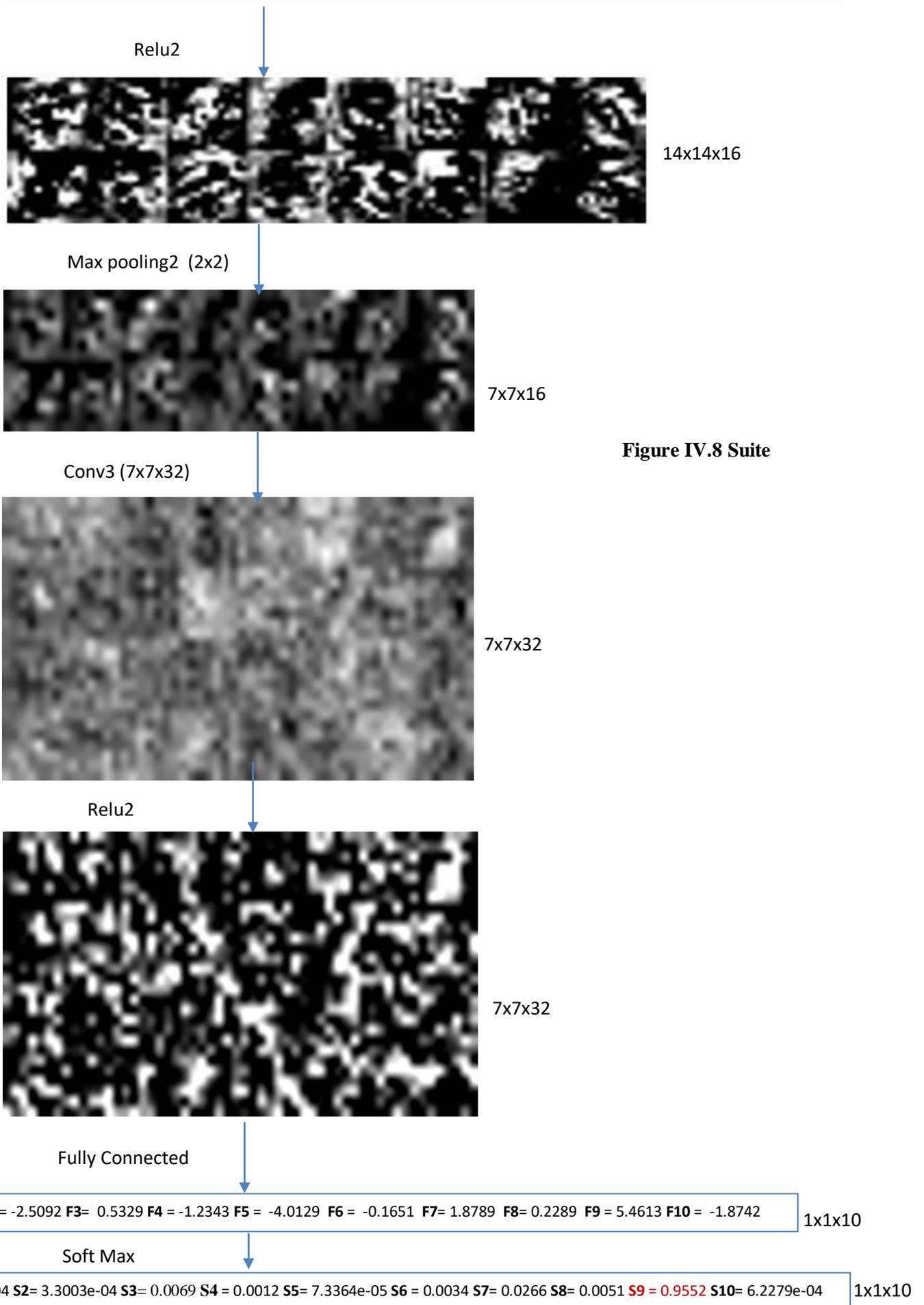


Figure IV.8 Suite

D'après la sortie de la fonction Soft Max qui assigne une probabilité d'appartenance à l'image brut d'entrée, on constate que notre model à bien reconnue le chiffre **8** en assignant la probabilité la plus élevé qui est de **0.9552** à la neuvième sortie **S9** qui correspond effectivement au chiffre **8**

6- Conclusion :

En conclusion en utilisant un model simple pour la classification d'image brut de chiffre Il est possible d'obtenir de bon résultats en se basant sur les critères de mesures de Performance tel que la précision globale, la précision moyenne, et le coefficient de cohen Kappa.

- Aussi les performance du model CNN à choisir son influencer par la profondeur du réseaux (nombre de block convlutif) et la taille du mini batch.
- Globalement un model bien choisi donne de bonne performance quel que soit la taille de l'ensemble d'apprentissage (proportionnellement à la taille de l'ensemble d'apprentissage).

Conclusion général

En conclusion, nous avons exploré en détail les concepts de base de l'intelligence artificielle (IA), de l'apprentissage automatique (ML), de l'apprentissage profond (DL) et de leur relation dans le domaine de la reconnaissance d'objets. Nous nous sommes intéressés en particulier aux réseaux de neurones convolutifs (CNN) et de leur rôle dans cette tâche.

Nous avons également présenté les différents types, différentes approches, et différents modèles de classification tels que les réseaux de neurones convolutifs CNN, en soulignant leurs avantages en 2D et en 3D. Nous avons aussi exposés les bases de la classification, et les différentes métriques de mesures de performance utilisées pour évaluer les classificateurs.

En utilisant un modèle simple de classification d'images numériques brutes, nous avons montré que de bons résultats peuvent être obtenus sur la base de critères de référence tels que la précision globale, la précision moyenne et le coefficient de corrélation de Cohen Kappa.

Il est important de noter que les performances du modèle CNN sont affectées par des facteurs tels que la profondeur du réseau (le nombre de blocs convolutionnels) et la taille du mini Lot utilisé pendant la phase d'entraînement. Ainsi, il est nécessaire de faire une sélection judicieuse de ces paramètres pour améliorer les performances du modèle.

En résumé, ces chapitres soulignent l'importance cruciale de l'intelligence artificielle, de l'apprentissage automatique et de l'apprentissage en profondeur dans le domaine de la reconnaissance d'images. Avec un modèle CNN approprié et en utilisant les critères de mesures de performance appropriés, de bons résultats peuvent être obtenus. Cependant, des facteurs tels que la profondeur du réseau et la petite taille des lots doivent être pris en compte lors de la sélection et de la configuration du modèle.

Liste Des Références

- [1] Introduction à l'Intelligence Artificielle GI 2020: M. Benbrahim
- [2] La Classification d'images d'insectes ravageurs en utilisant le Deep Learning Réalisé par : LOUNIS Katia MOUSSI Dahbia Encadré par Promotrice:MmeAOUDJIT. R Co-promotrice: MmeAIT ISSAD. H 2019-2020
- [3] <https://studymachinelearning.com/mathematics-behind-the-neural-network/>
- [4] Olivier Ezratty, Les usages de l'intelligence artificielle Novembre 2018
- [5] <https://amt-lab.org/blog/2017/3/a-brief-history-of-artificial-intelligence>
- [6] mémoire CONVOLUTIONAL NEURAL NETWORK POUR LA DETECTION DU PORT DU MASQUE Préparé par : MAZIR MELISSA AMRIOU HANANE Spécialité : Réseaux et Télécoms 2021/2022
- [7] Apprentissage profond | Psychomédia (psychomedia.qc.ca)
- [8] <https://developer.ibm.com/learningpaths/supervised-deep-learning/recurrent-neural-networks/introduction-recurrent-neural-networks>
- [9] A. HABBA et O. ISHAK. Année Universitaire 2018/2019. La classification des images satellitaires par l'apprentissage profonde (deeplearning). Mémoire de fin d'étude. Université Ahmed Draia – Adrar
- [10] <https://hub.packtpub.com/top-5-deep-learning-architectures>
- [11] <https://www.futura-sciences.com/tech/definitions/intelligence-artificielle-deep-learning-17262/> PAR CÉLINE DELUZARCHE JOURNALISTE (1979 - 2021) LE 7 FÉVRIER 2021
- [12] <https://www.analyticsvidhya.com/blog/2021/05/convolutional-neural-networks-cnn/> ManavMandal — Published On May 1, 2021 and Last Modified On April 28th, 2023
- [13] GASMI Lynda Deep Learning for face Recognition Deep Learning for face Recognition /2020 devant le jury composé de : Dr. AKHROUF Samir
- [14] <https://www.analyticsvidhya.com/blog/2021/05/convolutional-neural-networks-cnn/#be44> ManavMandal — Published On May 1, 2021 and Last Modified On April 28th, 2023
- [15] Modélisation et classification avec Deep Learning Application à la détection du Coronavirus Covid-19 Présenté par : Ounissi Mohammed Harnane Zahra Ilham Dirigé par : Pr. Lamamra Kheireddine 2020
- [16] LILIA, D. R. (2020). La Détection de la colère chez le conducteur en utilisant le Deep Learning. Algérie/BISKRA.

- [17] <https://openclassrooms.com/fr/courses/4470531-classez-et-segmentez-des-donnees-visuelles/5083336-decouvrez-les-differentes-couches-dun-cnn>
- [18] Calculate the Output Size of a Convolutional Layer | Baeldung on Computer Science Last updated: April 14, 2023 Written by: Panagiotis Antoniadis
- [19] https://penseeartificielle.fr/focus-reseau-neurones-convolutifs/#4_Le_flattening_ou_mise_a_pla
- [20] <https://stanford.edu/~shervine/l/fr/teaching/cs-230/pense-bete-reseaux-neurones-convolutionnels>
- [21] Classification automatique des textes par Les réseaux de neurones à convolution MALKI NARIMENE 2018/2019
- [22] La classification des images satellitaires par l'apprentissage profonde (deeplearning) Préparé par Abdelaziz HABBA et Omar ISHAK Encadré par Mr. OUAHAB Abdelwhab Année Universitaire 2018/2019
- [23] Sarah Madeleine 2021/02/16 <https://www.imaios.com/fr/ressources/blog/classification-des-images-medicales-comprendre-le-reseau-de-neurones-convolutifs-cnn>
- [24] June 22, 2021 <https://www.reachiteasily.com/2021/06/3d-convolutional-neural-network-pytorch.html>
- [25] http://www.songho.ca/dsp/convolution/convolution2d_example.html
- [26] <https://www.geeksforgeeks.org/getting-started-with-classification/>
- [27] Classification des images avec les réseaux de neurones Convolutionnels Dirigé par : Réalisé par : Promotion 2017-2018 Madame FELLAG. Mr BELHADJER Hakim. Mr SAROUER Brahim
- [28] <https://course.elementsofai.com/fr/4/1>
- [29] Stephen Allwright 14 May 2022 <https://stephenallwright.com/good-accuracy-score/>
- [30] Contributeurs pour cet article ShivamPandey-msft ,k fend ,roschlom, RobinARH ,ShivamPandeyMSFT 08/03/2023 <https://learn.microsoft.com/fr-fr/dynamics365/finance/finance-insights/confusion-matrix>
- [31] <https://www.l3harrisgeospatial.com/docs/calculatingconfusionmatrices.html>
- [32] http://gsp.humboldt.edu/olm/Courses/GSP_216/lessons/accuracy/metrics.html
- [33] https://www.researchgate.net/figure/The-accuracy-and-Cohens-kappa-of-the-confusion-matrix-example-for-the-3-class-problem-is_tbl4_289029388
- [34] :RaufBhat Data Scientist Published in Towards Data Science Oct 3, 2020 <https://towardsdatascience.com/gradient-descent-with-momentum-59420f626c8f>

[35] https://www.analyticsvidhya.com/blog/2020/10/how-does-the-gradient-descent-algorithm-work-in-machine-learning/?utm_source=blog&utm_medium=variants-of-gradient-descent-algorithm

[36] <http://sn1.chez-alice.fr/dedoubl/reseau.html>