

الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne démocratique et populaire

وزارة التعليم العالي والبحث العلمي
Ministère de l'enseignement supérieur et de la recherche scientifique

جامعة سعد دحلب البلدية
Université SAAD DAHLAB de BLIDA

كلية التكنولوجيا
Faculté de Technologie

قسم الإلكترونيك
Département d'Électronique



Mémoire de Master

Filière Électronique
Spécialité Instrumentation

Présenté par

DIBOUNE Khadidja

&

FENNICHE Rayane Ferial

Classification des défauts dans les réseaux électriques à l'aide de réseaux de neurones convolutifs CNN

Proposé par : Mr. YKHLEF Farid

Année Universitaire 2022-2023

Remerciements

A ALLAH le tout puissant qui nous a donné le courage, la patience et la santé pour arriver à ce qu'on est maintenant.

Nous voudrions, en premier lieu, adresser nos sincères remerciements aux nos

Professeurs de nous avoir accompagné et soutenir durant toute la période de notre formation.

Nous remercions autant Pr. Ykhlef Farid qui a pleinement participé à la réalisation de ce travail par son aide, ses précieux conseils, sa volonté de transmettre son savoir et tout le temps consacré pour nous.

Nous tenons à exprimer notre reconnaissance aux membres de jury pour l'intérêt qu'ils ont porté à notre recherche en acceptant d'examiner notre modeste travail et de l'enrichir par leurs recommandations.

Enfin, notre profond respect va à toute personne qui, de près ou de loin, a participé à la réalisation de ce travail.

Dédicaces

A mes chers parents, pour tous leurs sacrifices, leur amour, leur tendresse, leur soutien et leurs prières tout au long de mes études.

A mon très cher mari Mr. Fethi qui me donne toujours le courage et l'aide.

A mon très cher frère Mohamed pour leur appui et leur encouragement.

A mes chères sœurs Imen et Belkisse pour leurs encouragements permanents, et leur soutien moral.

A mon cher binôme Rayan.

A mes amies Asma Merabet et Dounia Bounab et Tachouche Hayet pour l'aide et la patience.

A toute ma famille et mes amies pour leur soutien tout au long de mon parcours universitaire.

Que ce travail soit l'accomplissement de vos vœux tant allégués, et le fruit de votre soutien infallible. Merci d'être toujours là pour moi.

Khadidja

A mes chers parents, pour tous leurs sacrifices, leur amour, leur tendresse, Leur soutien et leurs prières tout au long de mes études,

A mes chères sœurs pour leurs encouragements permanents, et leur soutien Moral surtout AMIRA.

A mes chers frères pour leur appui et leur encouragement.

A mon cher binôme KHADIDJA.

A toute ma famille pour leur soutien tout au long de mon parcours Universitaire.

Que ce travail soit l'accomplissement de vos vœux tant allégués, et le fruit de Votre soutien infallible.

Merci d'être toujours là pour moi.

RAYANE FERIEL

ملخص:

في الوقت الحاضر، تعد المعدات المستخدمة في المرافق الكهربائية أكثر عرضة لأنواع مختلفة من الأعطال والاضطرابات، ولا يمكن ضمان استمرارية الخدمات. يمكن أن تكون هذه الأعطال دائمة، فمن الضروري اكتشافها وتصنيفها لإصلاح واستعادة مصدر الطاقة في أسرع وقت ممكن.

الهدف من هذا العمل هو استخدام فعالية طريقة التعلم العميق لتصنيف المشاكل على الشبكة الكهربائية.

في هذا العمل، نستخدم الشبكات العصبية التلافيفية حيث قمنا بتدريبها واختبارها باستخدام المكتبات التي تدعم التعلم الآلي من أجل الحصول على النتائج المطلوبة.

الكلمات المفتاحية: الشبكة الكهربائية، التصنيف، المشاكل، تعلم العميق.

Résumé :

De nos jours, les équipements utilisés dans les services d'électricité sont plus sensibles à divers types de défauts et de perturbations, et la continuité des services ne peut être garantie. Ces défauts pouvant être permanents, il est nécessaire de les détecter, de les classer pour réparer et restaurer l'alimentation dès que possible.

L'objectif de ce travail est l'utilisation de l'efficacité de la méthode Deep Learning pour la classification des problèmes sur un réseau électrique.

Dans ce travail, on utilise des réseaux neuronaux convolutifs, que nous avons entraîné et testé à l'aide de la bibliothèque Tensorflow qui prend en charge l'apprentissage automatique en Python afin d'obtenir les résultats requis.

Mots clés : Réseau électrique, classification, problèmes, Deep Learning.

Abstract:

Nowadays, equipment used in electrical utilities is more susceptible to various types of faults and disturbances, and the continuity of services cannot be guaranteed. These faults can be permanent, it is necessary to detect them, classify them to repair and restore the power supply as soon as possible

The objective of this work is the use of the effectiveness of the Deep Learning method for the classification of problems on the electrical network.

In this work, we use convolutional neural networks where we trained and tested them using tensorflow libraries which supports machine learning in Python in order to obtain the required results.

Keywords: Electrical network, classification, problems, deep learning.

Listes des acronymes et abréviations

BT : Base Tension.

CNN : Convolution Neural Network.

CSV : Comma-Separated Value.

DL : Deep Learning.

FFT : Fast Fourier Transformation.

GPU : Graphical Processing Unit.

GRU : Gated Recurrent Unit.

HT : Haute Tension.

IA : Intelligence Artificielle.

LSTM : Long Short-Term Memory.

ML : Machine Learning.

MLP : Multi Layered

MT : Moyenne Tension.

RNA : Réseaux de Neurones Artificiels.

SVM : Support Vector Machine Perceptron.

THT : Très Haute Tension.

Table des matières

Introduction générale	1
Chapitre 1 État de l'art	4
1.1 Introduction	4
1.2 Définition d'un réseau électrique	4
1.3 Hiérarchisation du réseau électrique	5
1.3.1 Production	5
1.3.2 Transport	5
1.3.3 Répartition	5
1.3.4 Distribution	5
1.4 Production de l'énergie électrique	6
1.4.1 Types de centrales électriques	6
1.4.2 Production d'électricité à partir de ressources non Conventionnelles (Énergie renouvelable)	6
1.5 Défauts dans les réseaux électriques	7
1.6 Définition des défauts dans les réseaux électriques	8
1.6.1 Normal (signal pur)	8
1.6.2 Les harmoniques	8
1.6.3 Chute de tension	9
1.6.4 Transitoire	10
1.7 Origines des défauts	10
1.8 Les caractéristiques des défauts	11
1.9 Les conséquences des défauts	11
1.10 L'impact de l'intelligence artificielle sur la classification	12
1.11 Classification	12
1.11.1 Méthodes de classification	12
a. Méthode basée sur les caractéristiques du signal	12
b. Méthode basée sur les statistiques	13
c. Méthode basée sur les réseaux de neurones convolutifs (CNN)	13
d. Méthode basée sur les techniques de traitement de signal	14
1.12 Conclusion	14
Chapitre 2 : Les réseaux de neurones	16
2.1 Introduction	16
2.2 Notion de base	16
2.2.1 L'intelligence artificielle	16
2.2.2 Science de données (Data science)	17
2.2.3 Apprentissage automatique (Machine Learning)	17
2.2.4 Apprentissage profond (Deep learning)	17
2.2.5 Machine learning vs Deep learning	18
2.2.6 Le neurone biologique	19
2.2.7 Le neurone formel	19
2.2.8 Méthodes d'apprentissage	21
a. Apprentissage supervisé	21
b. Apprentissage non-supervisé	21
2.3 Réseau de neurones	21

2.3.1	Fonctionnement de réseaux de neurones	21
2.3.2	Types de réseaux de neurones.....	22
a.	Réseaux de neurones feed-forward (réseaux de neurones non bouclés)	22
b.	Réseaux de neurones récurrent (réseaux de neurones bouclés).....	22
c.	Réseaux de neurones convolutifs (CNN)	23
2.4	Réseaux de neurones convolutifs	23
2.4.1	Définition.....	23
2.4.2	Application	23
2.4.3	Architecture de CNN	24
2.4.4	Les couches d'un réseau de neurone convolutifs	24
a.	Couches convolutives	24
b.	Couche de correction (RELU).....	25
c.	Couche de regroupement (pooling)	25
d.	Couche d'aplanissement (flattening).....	26
e.	Couche totalement connectées (Fully connected ou Dense).....	27
f.	Couche de perte (Loss)	27
2.4.5	Exemples de modèles de CNN	28
2.4.6	Rétropropagation du gradient (Optimisation)	28
2.4.7	Découpage de données.....	29
2.4.8	Les fonctions d'activation	30
a.	Relu	30
b.	Softmax.....	30
2.5	Conclusion	31
Chapitre 3 Implémentation et interprétations des résultats.....		33
3.1	Introduction	33
3.2	Contexte expérimental.....	33
3.2.1	Google colab.....	33
a.	Définition	33
b.	Avantages	34
3.2.2	Python	34
3.2.3	Matériels utilisés	35
3.2.4	Librairies utilisées dans l'implémentation	35
a.	Pandas	35
b.	Matplotlib	36
c.	Numpy	36
d.	Tensorflow	36
3.2.5	Bases de données (Data set)	36
3.3	Extraction des paramètres	37
3.4	Métriques	38
3.4.1	Précision	38
3.4.2	Perte	39
3.5	La description d'architecture de modèle CNN	39
3.6	Résultat de classification.....	42
3.6.1	Visualiser les FFT des signaux (les classes).....	42
3.6.2	Précision et perte	43

3.6.3	Matrice de confusion	44
a.	Définition	44
b.	Interprétation de la matrice de confusion	44
3.7	Conclusion	46
	Conclusion générale	47
	Bibliographie	48

Liste des figures

Figure 1: Structure générale d'un réseau électrique.	6
Figure 2 : Graphe représente un signal pur.	8
Figure 3 : Graphe représente la troisième onde harmonique.	9
Figure 4 : Graphe représente la cinquième onde harmonique.....	9
Figure 5: Graphe représente la chute de tension.	10
Figure 6: Graphe représente le transitoire.	10
Figure 7 : Graphe représente les SVM	13
Figure 8: Convolutional neural network.	14
Figure 9: L'interaction entre intelligence artificielle et machine learning et deep learning.....	18
Figure 10 : Machine learning vs Deep learning.	18
Figure 11 : Modèle d'un neurone artificiel.	19
Figure 12 : mise en correspondance neurone biologique / artificiel.	20
Figure 13 : Exemple de Fonctionnement de réseau neurones.	22
Figure 14 : Convolutional Neural Network.....	24
Figure 15 : Schéma du glissement de la fenêtre de filtre sur l'image d'entrée.	25
Figure 16 : Schéma représentant la couche Relu.....	25
Figure 17: Type de la couche de regroupement.	26
Figure 18: Schéma représentant la couche de flattening.	26
Figure 19: Schéma représentant l'importance de la couche flatting.....	27
Figure 20 : Schéma représentant la couche de Fully connected.	27
Figure 21: Découpage des données.	29
Figure 22 : Tracés des fonctions Relu et Softmax.	30
Figure 23: logo de google colab	34
Figure 24: Logo de langage Python	35
Figure 25 : affichage de base de données.....	37
Figure 26 : affichage de base de données (suite).....	37
Figure 27 : Diagramme montre les étapes de code.	38
Figure 28 : Graphe représentant couche convolution (conv1d).	39
Figure 29 : Graphe représentant couche BatchNormalization.	39
Figure 30 : Graphe représentant couche convolution (conv1d_1).	40
Figure 31 : Graphe représentant couche BatchNormalization_1.	40
Figure 32 : Graphe représentant couche dense.....	41
Figure 33 : Graphe représentant couche dense_1.....	41
Figure 34 : Modèle séquentiel de CNN.	42
Figure 35 : code pour ploter les classes.	42
Figure 36 : Graphes représentent la transformation de fourrier des signaux.	43
Figure 37 : Graphe représentant la précision du modèle.	43
Figure 38 :Graphe représentant l'erreur du modèle.	44
Figure 39 : Matrice de confusion.	44
Figure 40 : Résultat la matrice de confusion.....	45

Liste des tableaux

Tableau 1 : La suite les classes (défauts) sur le réseau électrique.....	7
Tableau 2: transition entre le neurone biologique et artificiel.	20

Introduction générale

La gestion et l'optimisation de la distribution de l'énergie électrique sont deux enjeux majeurs pour les entreprises d'électricité et les industries du secteur. Dans ce contexte, la classification des problèmes sur un réseau électrique est une tâche critique pour assurer un approvisionnement électrique sûr et fiable.

Dans ce mémoire de fin d'étude, nous nous intéressons à l'utilisation de l'algorithme Convolutional Neural Network (CNN) pour la classification des problèmes dans un réseau électrique. Les CNN sont des modèles d'apprentissage en profondeur qui ont prouvé leur efficacité dans de nombreux domaines, notamment la vision par ordinateur.

Nous allons explorer comment les CNN peuvent être utilisés pour classer les différentes pannes et les dysfonctionnements qui peuvent survenir sur un réseau électrique, tels que les surtensions, les sous-tensions, les courts-circuits, les pertes de phase, les pertes de continuité.

Enfin, nous allons décrire l'implémentation de notre approche de classification de défauts à l'aide de l'algorithme CNN sur un jeu de données de test, et évaluer les performances de cette approche en termes de précision. Ce travail sert essentiel à aider les ingénieurs en électricité à améliorer la fiabilité et l'efficacité des réseaux électriques, ce qui est crucial pour répondre aux besoins énergétiques croissants dans les sociétés.

Ce mémoire est constitué de trois chapitres et organisé comme suit :

Dans le premier chapitre, nous présentons un état de l'art sur le réseau électrique, sa structure, et sa production, et les causes, les conséquences des défauts et la classification et ses modèles, ainsi enfin l'impact de l'intelligence artificielle sur la classification.

Ensuite, dans le second chapitre, nous nous consacrons sur la présentation des réseaux de neurones, des notions de base sur l'apprentissage automatique et profond, les types de réseaux de neurones où nous donnerons aussi plus de détails sur les réseaux de neurones convolutifs.

Nous nous penchons dans le dernier chapitre à la description et l'implémentation de notre système de classification des défauts basée sur les CNN dans un réseau électrique. Dans ce chapitre, nous présentons notre modèle et la base de données choisie pour l'implémentation. Après nous exposons les différents outils utilisés dans le développement de notre système, ainsi que les différents résultats obtenus.

Et enfin, nous terminerons ce mémoire par une conclusion générale qui résumera notre travail.

Chapitre 1

État de l'art



Chapitre 1 État de l'art

1.1 Introduction

Le réseau électrique est un système complexe et essentiel pour la distribution de l'énergie électrique à travers les zones résidentielles, commerciales et industrielles.

Cependant, ce système peut faire face à plusieurs problèmes tels que les pannes de courant, les surtensions, les sous-tensions, les déséquilibres de charge, les perturbations harmoniques, les interférences électromagnétiques, les pertes de transmission, etc.

Pour mieux comprendre ces problèmes, il est important de les classifier en fonction de leur nature et de leur impact sur le système.

Dans ce chapitre, on va décrire le réseau électrique et ses types ainsi que sa structure. Par la suite on discutera les défauts des réseaux électriques ainsi que leurs méthodes de classification utilisées afin d'assurer un approvisionnement en énergie électrique fiable et efficace.

1.2 Définition d'un réseau électrique

Un réseau électrique est un ensemble d'infrastructures permettant d'acheminer l'énergie électrique des centres de production vers les consommateurs d'électricité. Il est constitué de lignes électriques exploitées à différents niveaux de tension, connectées entre elles dans des postes électriques. Ces dernières permettent de distribuer l'électricité et de la transformer d'une grandeur à une autre grâce à des transformateurs.

Un réseau électrique doit également assurer la gestion dynamique de l'ensemble de la production, du transport, de la distribution et de la consommation, en mettant en œuvre des réglages visant à assurer la stabilité de l'ensemble [1].

1.3 Hiérarchisation du réseau électrique

La grille est divisée en quatre sections principales [2] :

1.3.1 Production

La production d'énergie électrique consiste à convertir l'énergie mécanique en énergie électrique à l'aide de turbo-alternateurs fonctionnant avec différentes sources primaires telles que le gaz, le pétrole, l'hydraulique ou le nucléaire. Chaque centrale électrique, en fonction du pays et de sa politique énergétique, utilise une ou plusieurs de ces sources primaires pour produire de l'électricité

1.3.2 Transport

Un alternateur produit de l'énergie électrique en moyenne tension (12 à 15 kV), et celle-ci est injectée dans le réseau de transport par des postes de transformation pour être transmise en haute ou très haute tension afin de réduire les pertes dans les lignes. Le niveau de la tension de transport varie selon les distances et les puissances transportées, plus les distances sont grandes plus la tension doit être élevée. Par exemple, le réseau de transport en Algérie utilise une tension de 220 kV (voir 400 kV pour certaines lignes dans le sud notamment), le réseau européen utilise 400 kV, et le réseau nord-américain 735 kV.

1.3.3 Répartition

Le réseau de répartition prend sa source dans le réseau de transport à partir des postes d'interconnexion THT/HT (MT) et sert à approvisionner les gros consommateurs industriels sous haute ou moyenne tension, et à répartir les puissances dans différentes régions rurales ou urbaines. Ce type de réseau utilise des tensions de 60 et 30 kV.

1.3.4 Distribution

La distribution sert à alimenter les consommateurs en moyenne ou en basse tension (typiquement 400 V), grâce à des postes de transformation MT/BT.

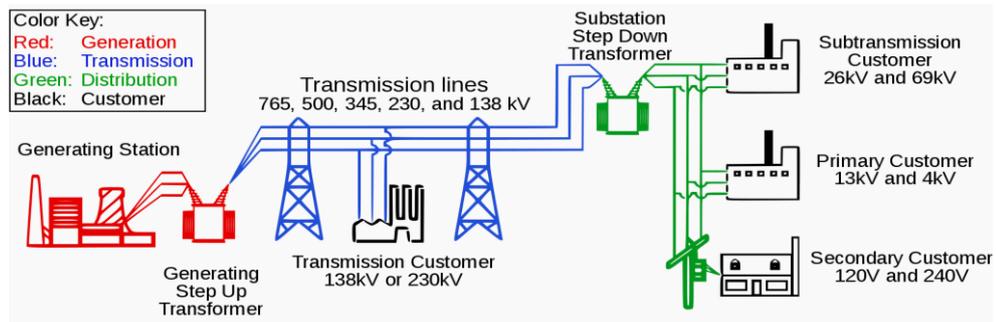


Figure 1: Structure générale d'un réseau électrique.

1.4 Production de l'énergie électrique

L'énergie électrique est produite dans des centrales qui disposent d'éléments indispensables à la génération de courants électriques qui sont [3] :

- Une turbine en mouvement.
- Un alternateur qui transforme l'énergie mécanique en énergie électrique.

1.4.1 Types de centrales électriques

Il existe plusieurs types de centrales électriques [3]:

- Les centrales à combustibles fossiles (charbon, pétrole et gaz naturel) dites centrales thermiques classiques.
- Les centrales nucléaires qui sont également des centrales que l'on peut qualifier de thermiques.
- Les centrales hydroélectriques.
- Les centrales solaires (photovoltaïques, thermique).
- Les centrales éoliennes.

1.4.2 Production d'électricité à partir de ressources non Conventionnelles (Énergie renouvelable)

Le terme "énergie renouvelable" est utilisé pour désigner l'échelle la moins inépuisable et abondante. Il existe cinq principaux types d'énergies renouvelables : l'énergie solaire, éolienne, hydraulique, la biomasse et la géothermie. Leur point commun est de produire (ou rarement) des émissions polluantes en phase d'exploitation, contribuant ainsi à la lutte contre les effets de l'errance et du réchauffement climatique. Les différents types d'énergies renouvelables sont listés ci-dessous :

- L'énergie solaire photovoltaïque et thermique.
- L'énergie éolienne.
- L'énergie géothermique.

1.5 Défauts dans les réseaux électriques

Les réseaux de distribution d'énergie peuvent être le siège d'un certain nombre d'incidents qui sont généralement dus à l'apparition de défauts. Ces défauts donnent lieu à l'établissement d'un courant de court-circuit soit entre un ou plusieurs conducteurs et la terre.

Il existe plusieurs défauts. Dans notre étude nous nous intéressons à cinq classes :

- Normal (signal pur).
- 3^{ème} onde harmonique.
- 5^{ème} onde harmonique.
- Chute de tension.
- Transitoire.

Classe	Equation	Paramètre
Normal	$s(t) = A \sin(\omega t)$	$A = 1 \text{ pu},$ $f = 60 \text{ Hz ou } 50\text{Hz}$
Chute de tension	$s(t) = A(1 - \alpha(u(t - t_1) - u(t - t_2))) \sin(\omega t)$	$0.1 \leq \alpha \leq 0.9,$ $T < t_2 - t_1 < 9T$
Transitoire	$S(t) = A \sin(\omega t) + \alpha \exp(-(t-t_1) \tau) \sin(\omega (t - t_1))$	$0.1 \leq \alpha \leq 0.8,$ $0.2 \text{ } 0.008\text{s} < \tau < 0.04\text{s},$ $100\text{Hz} < \omega < 400\text{Hz}$
3 ^{em} onde harmonique	$S(t) = A (\alpha_1 \sin(\omega t) + \alpha_3 \sin(3\omega t))$	$0.05 \leq \alpha_3 \leq 0.15,$ $\sum \alpha_i^2 = 1$
5 ^{em} onde harmonique	$S(t) = A (\alpha_1 \sin(\omega t) + \alpha_5 \sin(5\omega t))$	$0.05 \leq \alpha_5 \leq 0.15,$ $\sum \alpha_i^2 = 1$

Tableau 1 : La suite les classes (défauts) sur le réseau électrique. [4]

1.6 Définition des défauts dans les réseaux électriques

1.6.1 Normal (signal pur)

Un signal pur est un signal qui ne contient qu'une seule fréquence et qui est donc périodique. Cela signifie qu'il est représenté mathématiquement par une sinusoïde unique.

Le concept de signal pur est souvent utilisé en théorie et en traitement du signal pour simplifier l'analyse de signaux complexes en les décomposant en signaux purs de fréquences différentes.

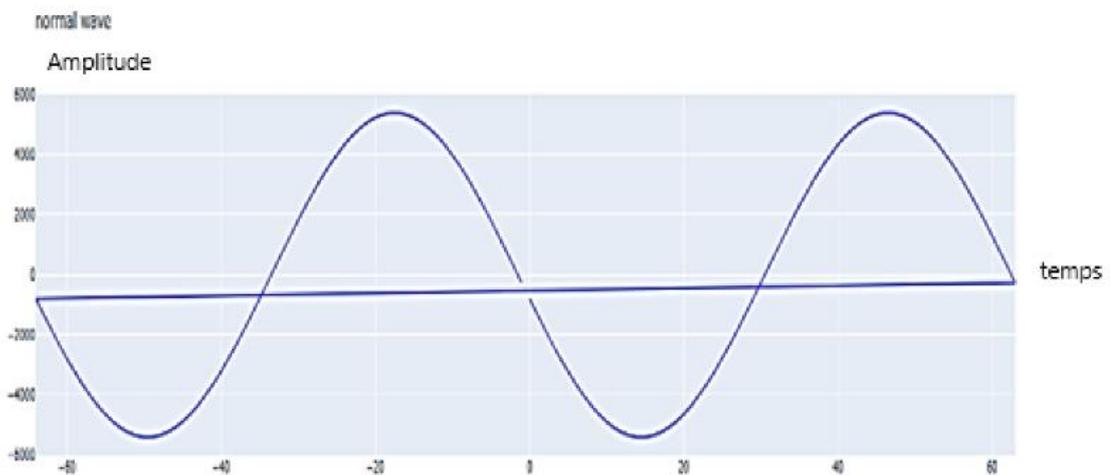


Figure 2 : Graphe représente un signal pur.

1.6.2 Les harmoniques

Les harmoniques peuvent être définies comme des formes d'onde sinusoïdales avec des fréquences qui sont multiples de la fréquence à laquelle la tension d'alimentation doit être délivrée. Ils sont généralement produits en raison des caractéristiques non linéaires de la charge et des appareils. Un paramètre utilisé pour calculer l'harmonique est la distorsion harmonique totale. Il existe plusieurs types d'harmoniques telles que la 3^{em}, la 5^{em}, la 7^{em} et la 9^{em} onde. Nous nous intéressons aux deux premières harmoniques [5]. Les harmoniques peuvent être présentes dans différents types de signaux périodiques, tels que les signaux électriques, les signaux sonores ou les signaux lumineux.

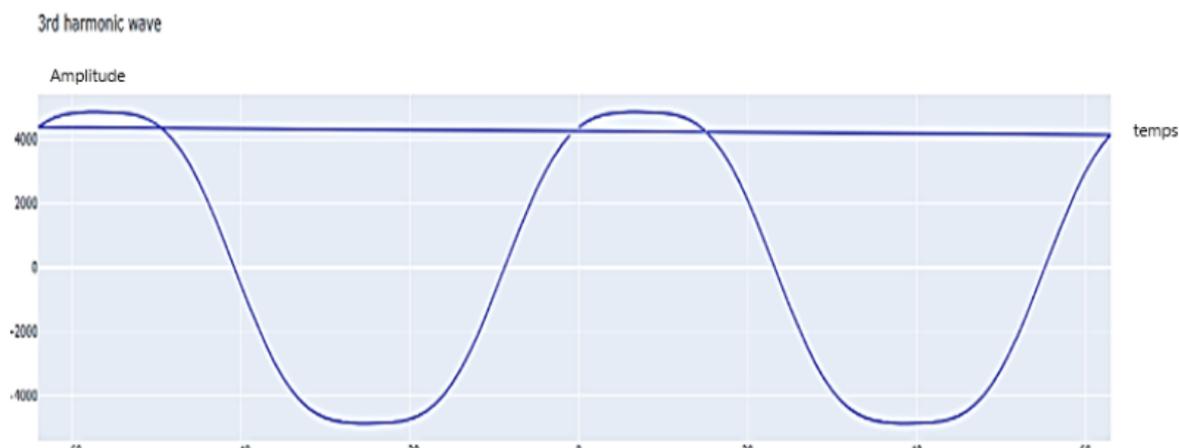


Figure 3 : Graphe représente la troisième onde harmonique.

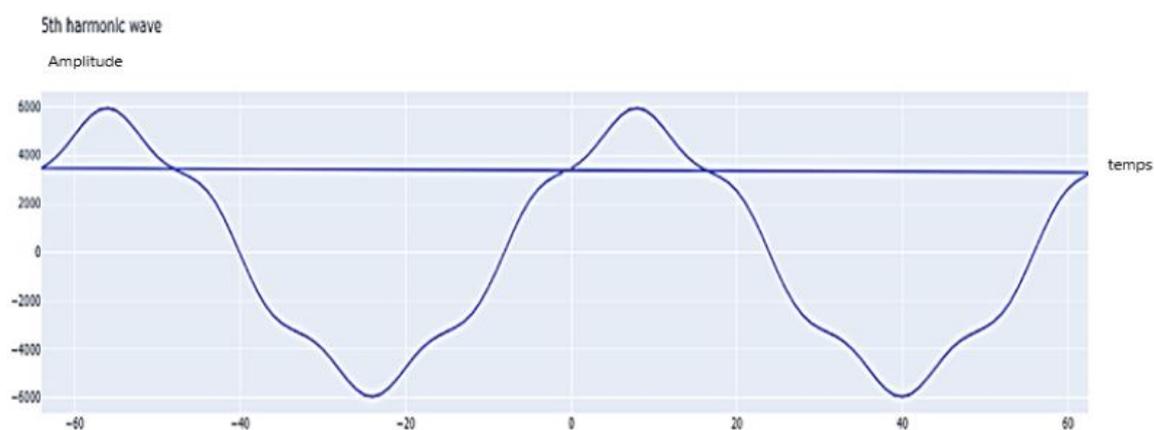


Figure 4 : Graphe représente la cinquième onde harmonique

1.6.3 Chute de tension

Une chute de tension est une baisse de tension brève (courte) et temporaire de l'amplitude de la tension dans le réseau de distribution ou le système électrique du client. Elle peut être causée par divers défauts dans les réseaux de transport et de distribution, des défauts dans les équipements connectés ou des courants d'appel et de commutation élevés dans l'installation du client.

Une chute de tension, également appelée baisse de tension, peut affecter le fonctionnement d'équipements tels que l'électronique et d'autres équipements sensibles connectés au réseau. L'effet varie d'un équipement à un autre et dépend de la sévérité du creux, de sa durée, de son ampleur et de la capacité de l'équipement à supporter la baisse de tension [6].

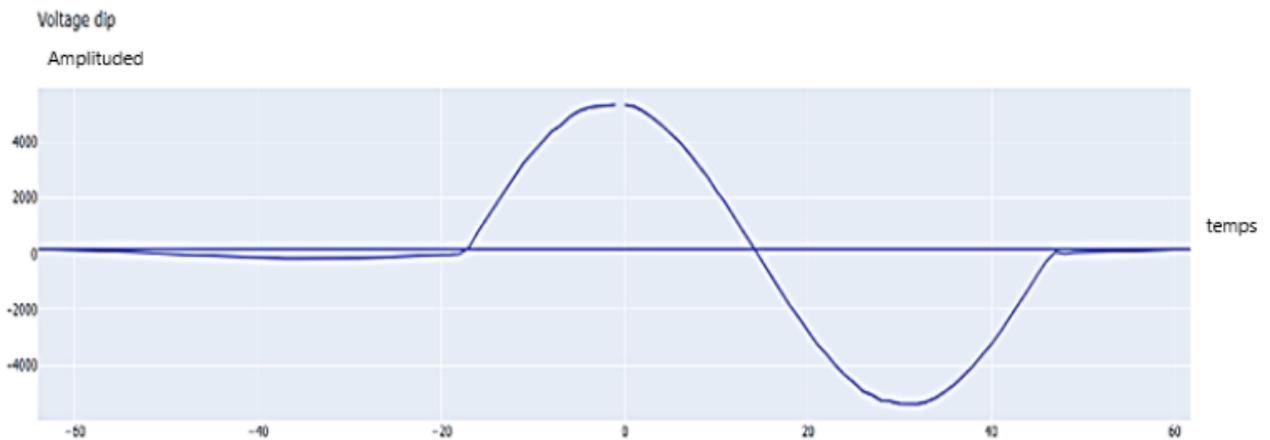


Figure 5: Graphe représente la chute de tension.

1.6.4 Transitoire

Le transitoire est un signal dont la perturbation meurt à zéro à un moment spécifié. Les transitoires peuvent être divisés en transitoires impulsionnels et transitoires oscillatoires.

Les transitoires impulsionnels sont un changement soudain de fréquence sans puissance dans l'état d'équilibre du signal de puissance, qui est généralement unidirectionnel en polarité, tandis que les transitoires oscillatoires sont un changement soudain de fréquence dans l'état d'équilibre du signal de puissance et cela inclut généralement des valeurs de polarité positives et négatives [5].



Figure 6: Graphe représente le transitoire.

1.7 Origines des défauts

Les défauts liés aux lignes de transport prennent leurs racines dans :

- **Électriques** : C'est l'altération des isolants des matériels de réseau, par exemple. En effet, les matériels électriques que l'on trouve sur le réseau ou dans les postes comportent des isolants (solides, liquides ou gazeux) constitués d'assemblages plus ou

moins complexes placés entre les parties sous tension et la masse. Ces isolants subissent des dégradations au cours du temps qui conduisent à des défauts d'isolement et donc des courts-circuits [7].

- **Atmosphériques** : Les lignes aériennes sont soumises aux perturbations extérieures telles que la foudre, les tempêtes ou le givre [7].
- **Mécaniques** : C'est la chute d'un corps sur un conducteur ou la dégradation mécanique de conducteurs consécutive à des agressions extérieures par des engins de terrassement [7].
- **Humaines** : Ce sont les fausses manœuvres telles l'ouverture d'un sectionneur en charge [7].

1.8 Les caractéristiques des défauts

Ils sont principalement caractérisés par :

- **Leurs durées** : Auto-extincteur, fugitif ou permanent.
- **Leurs origines** : Electrique, atmosphérique, mécanique et humain.
- **Leurs localisations** : Interne ou externe à une machine ou à un tableau électrique.

Outre ces caractéristiques, les courts-circuits peuvent être :

Monophasés : 80% des cas.

Biphasés : 15% des cas. Ces défauts dégénèrent souvent en défauts triphasés :

Triphasés : 5% seulement dès l'origine. [8]

1.9 Les conséquences des défauts

- **Les surintensités** : On appelle surintensité, tout courant supérieur à une valeur assignée. Cette valeur garantit un bon fonctionnement ainsi que la sécurité des personnes et des biens. On distingue habituellement dans les surintensités, les surcharges et les courts circuits [9].
- **La surcharge** : Les dégâts causés par les surcharges plus particulièrement aux matières isolantes en contact avec les conducteurs de courant, sont la conséquence de l'effet thermique du courant. La durée de cet effet thermique est relativement longue (de quelques secondes à quelques heures) [9].
- **Le court-circuit** : Echauffement important pouvant entraîner la fusion des parties actives (soudure du contact, projection de particule), l'effet thermique est tellement

rapide que les dégâts dans l'installation se produisent en quelques millisecondes, provoquant également un arc électrique [9].

- **Les surtensions** : L'augmentation soudaine de la tension, généralement à la foudre au contact entre HTA et BTA, provoque un claquage des isolants et créations des courts-circuits [9].

1.10 L'impact de l'intelligence artificielle sur la classification

L'intelligence artificielle a un impact significatif sur la classification des problèmes et des perturbations sur le réseau électrique. Les systèmes de distribution électrique modernes sont de plus en plus complexes et interconnectés, ce qui rend la gestion des problèmes de plus en plus difficile pour les opérateurs de réseau électrique.

L'IA peut aider à résoudre ce problème en fournissant des solutions automatisées pour la classification des problèmes et des perturbations sur le réseau électrique. Les techniques d'apprentissage automatique peuvent être utilisées pour analyser les données de surveillance du réseau électrique, telles que les mesures de tension et de courant, pour détecter les anomalies et les perturbations.

De plus, l'IA peut être utilisée pour la classification des problèmes de manière plus efficace et précise. Par exemple, les algorithmes de clustering peuvent être utilisés pour regrouper des problèmes similaires.

1.11 Classification

Le processus de classification des signaux de distribution d'énergie électrique, connu sous le nom de classification des signaux électriques, consiste à regrouper les signaux en différentes classes ou catégories en fonction de leurs caractéristiques ou de leurs propriétés. Cette classification trouve son application dans plusieurs domaines, tels que la surveillance des réseaux électriques, la gestion de l'énergie, la détection d'anomalies, et bien d'autres

1.11.1 Méthodes de classification

a. *Méthode basée sur les caractéristiques du signal*

Cette méthode implique l'extraction de caractéristiques pertinentes du signal électrique, telles que l'amplitude, la fréquence, la puissance, la forme d'onde, etc., et l'utilisation de

ces caractéristiques pour classer les signaux en différentes catégories à l'aide de techniques d'apprentissage supervisé, tels que les arbres de décision, les réseaux de neurones, les machines à vecteurs de support (SVM), etc. [10].

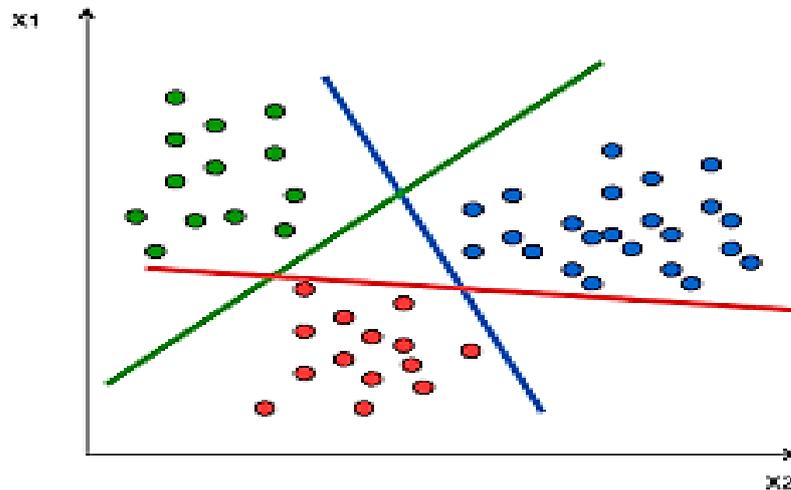


Figure 7 : Graphe représente les SVM

b. **Méthode basée sur les statistiques**

Cette méthode implique l'utilisation de statistiques descriptives, telles que la moyenne, la médiane, l'écart-type, etc., pour caractériser les signaux électriques, et la comparaison de ces statistiques entre les signaux pour les classer en différentes catégories.

c. **Méthode basée sur les réseaux de neurones convolutifs (CNN)**

Les réseaux de neurones convolutifs sont une classe d'algorithmes d'apprentissage profond spécialement conçus pour traiter des données en forme de grille, tels que des images ou des signaux temporels.

Les CNN peuvent être utilisés pour extraire automatiquement des caractéristiques pertinentes des signaux électriques et les utiliser pour la classification. [11]

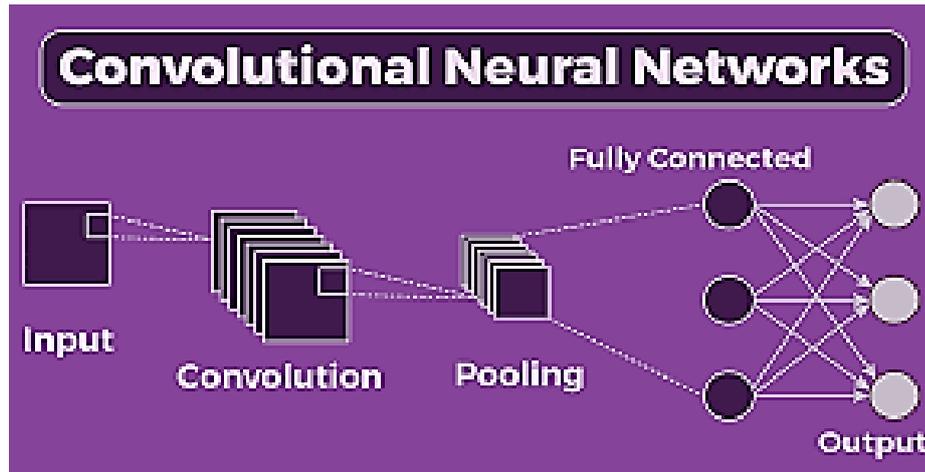


Figure 8: Convolutional neural network.

d. ***Méthode basée sur les techniques de traitement de signal***

Cette méthode implique l'utilisation de techniques de traitement du signal, telles que la transformée de Fourier, la transformée en ondelettes, etc., pour extraire des informations du signal électrique, et l'utilisation de ces informations pour la classification [12].

1.12 Conclusion

En conclusion, nous pouvons affirmer que le réseau électrique est un système complexe qui peut être soumis à diverses perturbations et défauts. Les perturbations peuvent être causées par des événements naturels tels que les tempêtes, les orages ou les séismes, ainsi que par des facteurs humains tels que la surcharge, les erreurs de manipulation, etc. Les défauts peuvent être des courts-circuits, des pannes d'équipements, des chutes de tension.

La classification des perturbations et des défauts est une étape cruciale dans la surveillance et la gestion du réseau électrique. Les méthodes de classification basées sur l'intelligence artificielle ont montré une grande efficacité dans la détection et la classification automatique de ces phénomènes. Ces méthodes sont basées sur l'utilisation de techniques telles que les réseaux de neurones, les algorithmes de clustering.

Au total, la classification des perturbations et des défauts sur le réseau électrique est un domaine de recherche en pleine expansion qui offre de nombreuses opportunités pour améliorer la qualité de l'énergie électrique et réduire les coûts de maintenance. Il est donc essentiel de continuer à explorer et développer de nouvelles méthodes pour améliorer la classification et la surveillance du réseau électrique.

Chapitre 2

Les réseaux de

neurones



Chapitre 2 : Les réseaux de neurones

2.1 Introduction

Les réseaux de neurones sont une classe de modèles d'apprentissage automatique (ML) qui ont révolutionné de nombreux domaines tels que la vision par ordinateur, la reconnaissance vocale, le traitement d'image et bien d'autres encore. Inspirés par la structure du cerveau humain, les réseaux de neurones sont constitués de multiples neurones interconnectés, organisés en couches.

Ces couches peuvent être entraînées sur des données d'apprentissage pour apprendre à effectuer des tâches telles que la classification d'images et de signal, la prédiction de séries temporelles, ou encore la reconnaissance de la parole.

Les réseaux de neurones ont connu une forte popularité ces dernières années, grâce à l'augmentation de la puissance de calcul et la disponibilité de grands ensembles de données.

Dans ce chapitre, nous allons explorer les principaux types de réseaux de neurones, leurs architectures, leur fonctionnement, leur entraînement et leurs applications pratiques.

2.2 Notion de base

2.2.1 L'intelligence artificielle

L'intelligence artificielle (IA) est un processus d'imitation de l'intelligence humaine qui repose sur la mise en œuvre et l'application d'algorithmes exécutés dans un environnement informatique dynamique. Son but est de permettre à des ordinateurs de penser et d'agir comme des êtres humains [13]. Elle est utilisée dans divers domaines tels que la médecine, les transports, le traitement de signal, et bien d'autres.

2.2.2 **Science de données (Data science)**

La data science s'est imposée dans de nombreuses actions. Le data scientist est l'un des rôles les plus recherchés par les organisations désormais « data driven ». Les applications de data science reposent sur des technologies telles que le Machine Learning et les capacités du Big Data. Elles doivent extirper des données un flot d'indicateurs, et ouvrir le champ des possibles en se basant sur des analyses prédictives, de la reconnaissance d'objets et d'images ou encore le traitement du langage naturel [14].

2.2.3 **Apprentissage automatique (Machine Learning)**

Le Machine Learning (ML) peut être défini comme étant une technologie d'intelligence artificielle permettant aux machines d'apprendre sans avoir été au préalable programmées spécifiquement à cet effet. Le Machine Learning est explicitement lié au Big Data, étant donné que pour apprendre et se développer, les ordinateurs ont besoin de flux de données à analyser, sur lesquelles s'entraîner.

De ce fait, le Machine Learning, issu par essence du Big Data, a précisément besoin de ce dernier pour fonctionner. Le Machine Learning et le Big Data sont donc interdépendants [15].

2.2.4 **Apprentissage profond (Deep learning)**

Le Deep Learning ou apprentissage profond (DL) est un type d'intelligence artificielle dérivé du *Machine Learning* (apprentissage automatique) où la machine est capable d'apprendre par elle-même, contrairement à la programmation où elle se contente d'exécuter à la lettre des règles prédéterminées [16].

Le Deep Learning est utilisé dans de nombreux domaines, tels que la reconnaissance d'image, la traduction automatique, la reconnaissance vocale, la robotique, traitement de signal, etc.

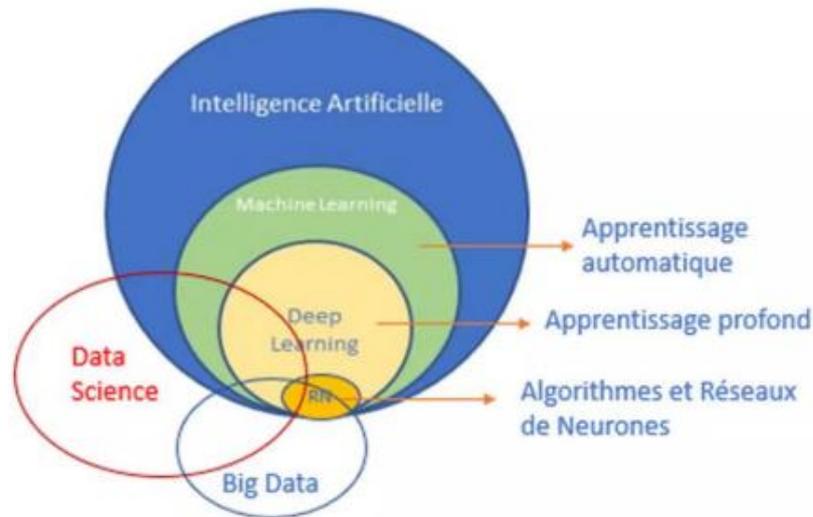


Figure 9: L'interaction entre intelligence artificielle et machine learning et deep learning.

2.2.5 Machine learning vs Deep learning

La majeure différence qu'on note entre ses 2 concepts provient de la manière dont les données sont présentées au système (modèle).

- Les algorithmes de ML nécessitent presque toujours des données structurées, alors que les réseaux d'apprentissage approfondis reposent sur des couches de réseaux de neurones artificiels (RNA).
- On voit aussi une différence au sein de l'architecture des modèles qui les composent, on note que les modèles type DL sont plus profonds que les modèles type ML.
- Deep learning n'utilise que les réseaux de neurones, alors que pour le ML les réseaux de neurones sont qu'une approche de conception des modèles parmi tant d'autres. En considérant le fait que le DL est la prochaine étape de l'évolution du ML inculquant aux machines la manière de prendre leurs décisions de façon précise sans l'intervention de l'expert humain [17].

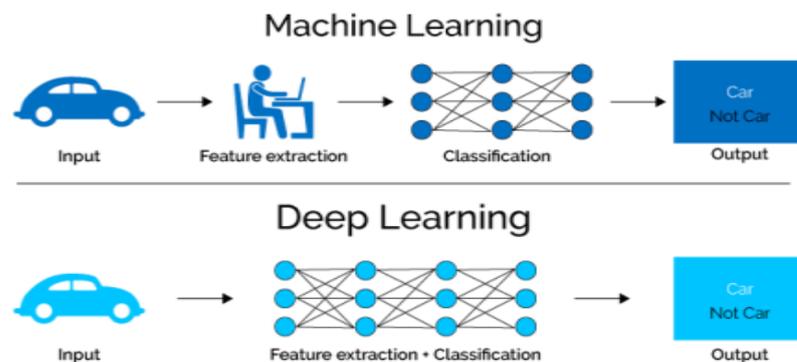


Figure 10 : Machine learning vs Deep learning.

2.2.6 Le neurone biologique

Un neurone biologique reçoit des entrées ou signaux transmis par d'autres neurones (interaction dendrites- synapse). Au niveau du corps (soma), le neurone analyse et traite ces signaux en les sommant. Si le résultat obtenu est supérieur au seuil d'activation (ou d'excitabilité), il envoie une décharge nommée potentiel d'action le long de son axone vers d'autres neurones biologiques [18].

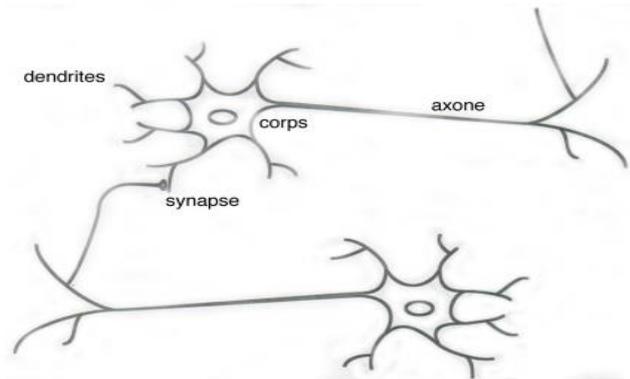


Figure 11 : schéma d'un neurone biologique [18].

2.2.7 Le neurone formel

Chaque neurone artificiel est un opérateur mathématique, on peut dire processeur élémentaire. Il reçoit un nombre variable d'entrées en provenance de neurones amonts. A chacune de ces entrées est associée un poids w (weight : poids en anglais) représentatif de la force de la connexion. Chaque processeur élémentaire est doté d'une sortie unique, qui se ramifie ensuite pour alimenter un nombre variable de neurones avals. A chaque connexion est associé un poids [19].

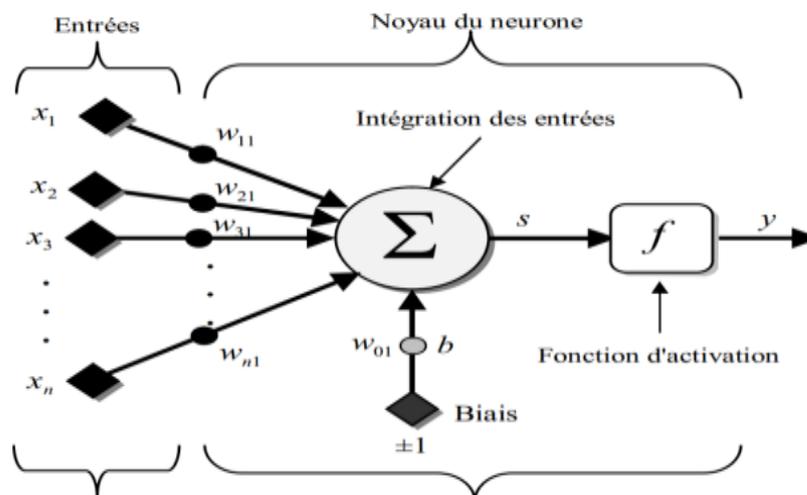


Figure 11 : Modèle d'un neurone artificiel [18].

- Les x_i représentent les vecteurs d'entrées.
- Les w_{ij} sont les poids synaptiques du neurone j .
- Biais : entrée prend souvent les valeurs -1 ou +1 qui permet d'ajouter de la flexibilité au réseau en permettant de varier le seuil de déclenchement du neurone par l'ajustement des poids et du biais lors de l'apprentissage.
- Noyau : intègre toutes les entrées et le biais et calcul la sortie du neurone selon une fonction d'activation qui est souvent non linéaire pour donner une plus grande flexibilité d'apprentissage.

Le tableau ci-dessous indique la transition entre le neurone biologique et le neurone formel (artificiel) :

Neurone biologique	Neurone formel
Synapses	Poids des connexions
Axones	Signal de sortie
Dendrites	Signal d'entrée
Noyau ou Somma	Fonction d'activation

Tableau 2: transition entre le neurone biologique et artificiel.

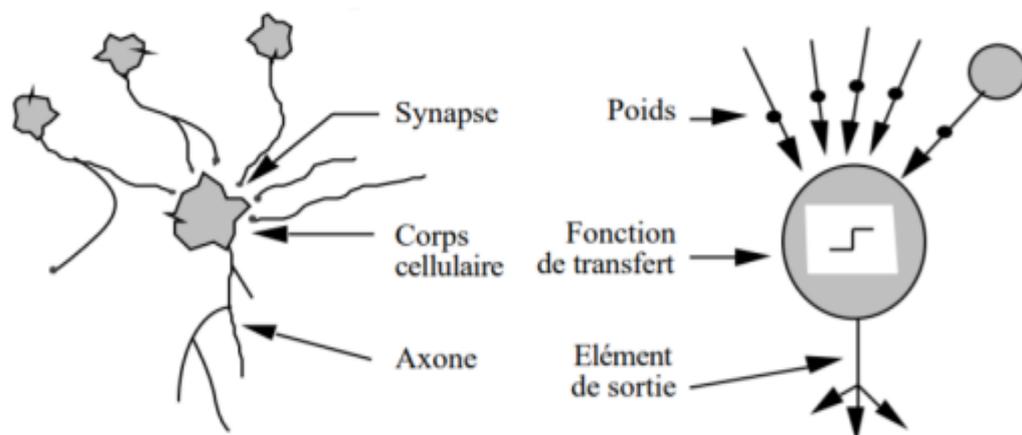


Figure 12 : mise en correspondance neurone biologique / artificiel [18].

2.2.8 Méthodes d'apprentissage

On appelle apprentissage le processus d'acquisition de connaissances sur l'environnement. On peut distinguer deux procédures de base : l'apprentissage supervisé et l'apprentissage non supervisé.

a. ***Apprentissage supervisé***

Les algorithmes d'apprentissage supervisé utilisent un jeu de données où à chaque échantillon est associé à une étiquette ou catégorie ou classe (label, Target). Cela implique d'avoir des bases de données annotées en fonction de ces classes. Un algorithme d'apprentissage supervisé se base sur la correction des erreurs de prédiction en comparant la classe prédite d'une donnée par rapport à la classe à laquelle appartient effectivement cette donnée [19].

b. ***Apprentissage non-supervisé***

L'apprentissage non supervisé n'a pas de données d'entraînement étiquetées à sa disposition. Dans ce cas-là, l'algorithme tente de trouver dans les données des caractéristiques qui permettent de les différencier et de mettre en évidence l'existence de classes ou groupes [19].

2.3 Réseau de neurones

Le concept de réseau de neurones est issu de l'étude des neurones biologiques à partir desquels un mécanisme d'apprentissage et de restitution d'informations a été déterminé. Les réseaux de neurones font partie du domaine des sciences cognitives qui cherche à développer des modèles de systèmes capables de présenter des capacités d'apprentissage et d'adaptation à leur environnement, ils permettent aux ordinateurs de résoudre des problèmes de manière autonome et d'améliorer leurs capacités de manière générale [19].

2.3.1 Fonctionnement de réseaux de neurones

Un réseau de neurones repose sur un grand nombre de processeurs fonctionnant en parallèle et organisés par tiers. Le premier tiers reçoit les entrées d'informations brutes.

Par la suite, chaque tiers reçoit les sorties d'informations du tiers précédent. Le dernier tiers, en revanche, produit les résultats du système [20].

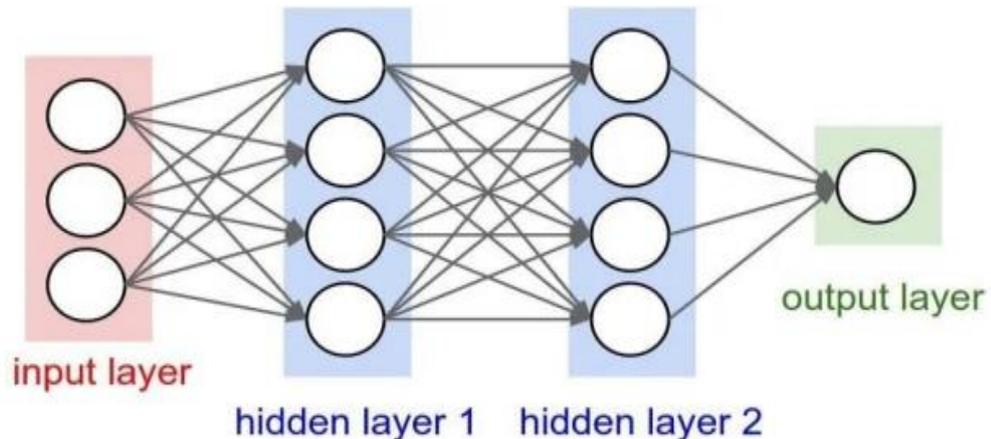


Figure 13 : Exemple de Fonctionnement de réseau neurones.

2.3.2 Types de réseaux de neurones

On distingue différents types de réseaux de neurones en fonction du nombre d'épaisseurs qui séparent l'entrée de données de la production du résultat, en fonction du nombre de nœuds cachés du modèle, ou encore du nombre d'entrées et de sorties de chaque nœud. En fonction du type de réseau, la propagation des informations entre les différents tiers de neurones peut varier :

a. **Réseaux de neurones feed-forward (réseaux de neurones non bouclés)**

Les réseaux de neurones feed-forward sont des réseaux de neurones où les connexions entre les neurones ne forment pas de boucles. Les entrées sont transmises à travers le réseau de neurones couche par couche, avec chaque couche effectuant une transformation non linéaire sur les données. Les réseaux de neurones feed-forward sont couramment utilisés pour des tâches telles que la classification d'images, la détection d'objets et la reconnaissance de visages.

Exemples de réseaux de neurones feed-forward :

- Multilayer Perceptron (MLP) [21].
- Convolutional Neural Network (CNN) [22].

b. **Réseaux de neurones récurrent (réseaux de neurones bouclés)**

Les réseaux de neurones récurrents sont conçus pour traiter des données séquentielles, telles que des séquences de texte ou des signaux temporels. Ils sont appelés récurrents car ils ont des connexions bouclées qui permettent aux sorties de chaque étape de la

séquence d'être utilisées comme entrées pour la prochaine étape. Les réseaux de neurones récurrents sont capables de modéliser des relations temporelles complexes et ont été largement utilisés dans des applications telles que la reconnaissance de la parole, la traduction automatique et la prédiction de séries temporelles.

Exemples :

- Long Short-Term Memory (LSTM) [23].
- Gated Recurrent Unit (GRU) [24].

c. **Réseaux de neurones convolutifs (CNN)**

Ces réseaux sont conçus pour traiter des données en grille, comme des images, en prenant en compte la structure spatiale des données. Les réseaux convolutifs sont souvent utilisés pour la reconnaissance d'images, la classification d'images et la segmentation d'images [25].

2.4 Réseaux de neurones convolutifs

2.4.1 Définition

Dans le domaine de l'apprentissage profond, un réseau neuronal convolutif (CNN) est une classe de réseaux neuronaux artificiels d'apprentissage profond et de type feed-forward, et un type particulier de perceptrons multicouches supervisés.

Les réseaux neuronaux convolutifs (CNN) sont une classe de modèles profonds inspirés du traitement de l'information dans le cerveau humain. Dans le système visuel du cerveau, chaque neurone possède un champ réceptif qui capte les données d'un certain voisinage local dans l'espace visuel. Ils sont spécifiquement conçus pour reconnaître des données multi-dimensionnelles avec un degré élevé de variance à la mise à l'échelle et à la distorsion [26].

2.4.2 Application

- Reconnaissance d'objets, segmentation, détection, classification.
- Traitement du langage naturel, traitement de signal.
- Découverte de médicaments ,évaluation des risques pour la santé et découverte de biomarqueurs du vieillissement.
- Analyse vidéo.

2.4.3 Architecture de CNN

L'architecture du CNN se compose d'une couche d'entrée, de plusieurs types de couches cachées et d'une couche de sortie. Le premier type de couches cachées est responsable de la convolution et l'autre du calcul de la moyenne locale, du sous-échantillonnage et de la réduction de la résolution. Les troisièmes couches cachées agissent comme un classificateur perceptron multicouche traditionnel [26].

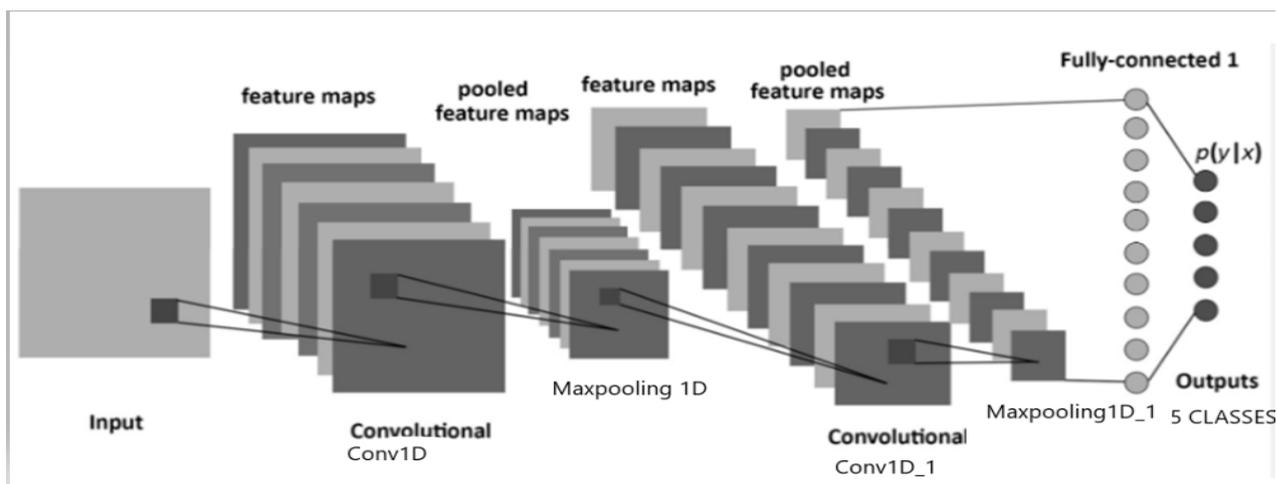


Figure 14 : Convolutional Neural Network.

2.4.4 Les couches d'un réseau de neurone convolutifs

a. Couches convolutives

Les couches convolutives constituent le noyau du réseau de neurones convolutifs. Une convolution extrait les tuiles de la carte de caractéristiques d'entrée (featuremap : est une représentation abstraite de l'image).

Ses valeurs dépendent des paramètres du noyau de convolution appliqué et des valeurs de pixels de l'image d'entrée) et leur applique des filtres pour calculer de nouvelles caractéristiques, produisant ainsi une carte de caractéristiques de sortie, ou caractéristique convoluée (dont la taille et la profondeur peuvent être différentes de celles de la carte de caractéristiques d'entrée). Les convolutions sont définies par trois paramètres [27] :

- **La profondeur de la couche** : c'est à dire le nombre de noyaux de convolution (ou nombre de neurones associés à un même champ récepteur).
- **Le pas (Stride)** : il contrôle le chevauchement des champs récepteurs. Plus le pas est petit, plus les champs récepteurs se chevauchent et plus le volume de sortie sera grand.

- **Le remplissage par zéro ou la marge (à 0) (zeropadding)** : parfois, il est commode de mettre des zéros à la frontière du volume d'entrée. Cela contrôle la dimension spatiale du volume de sortie. En particulier, il est parfois souhaitable de conserver la même surface que celle du volume d'entrée.

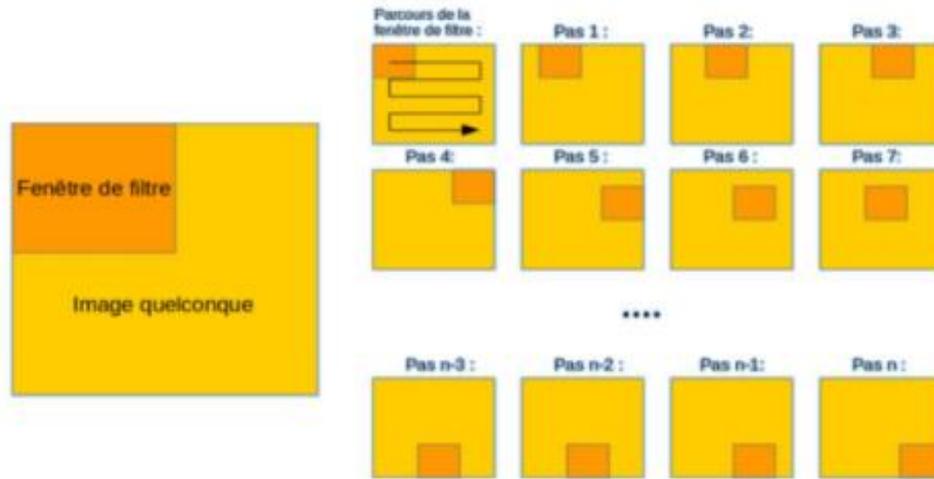


Figure 15 : Schéma du glissement de la fenêtre de filtre sur l'image d'entrée.

b. **Couche de correction (RELU)**

L'unité linéaire rectifiée est la fonction d'activation la plus couramment utilisée dans les modèles d'apprentissage profond. La fonction renvoie 0 si elle reçoit une entrée négative, mais pour toute valeur positive x , elle renvoie cette valeur. Positive x , elle renvoie cette valeur. Elle peut donc s'écrire $R(z) = \max(0, z)$ [28].

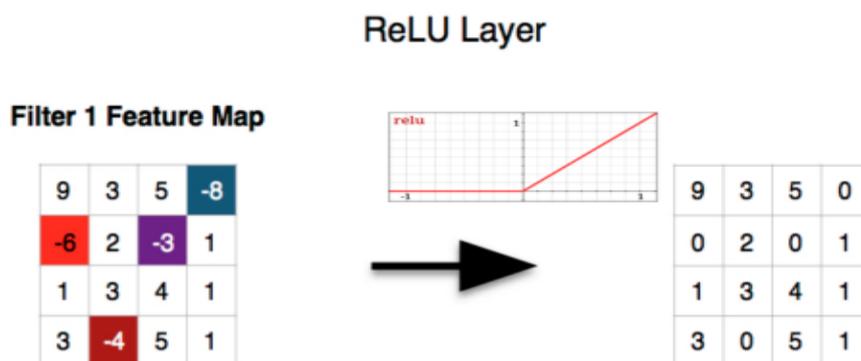


Figure 16 : Schéma représentant la couche Relu.

c. **Couche de regroupement (pooling)**

La couche de mise en commun effectuera une opération de sous-échantillonnage le long des dimensions spatiales (largeur, hauteur).

Le pooling est une opération simple qui consiste à remplacer un carré de pixels (généralement 2*2 ou 3*3) par une seule valeur.

On peut considérer 3 types de regroupement :

1. Pooling moyen (average pooling) : prend la valeur moyenne des pixels dans le carré sélectionné.
2. Mise en commun maximale (max pooling) : prend la valeur maximale des pixels dans le carré sélectionné.
3. Sum pooling : prend la valeur de la somme des pixels dans le carré sélectionné [28].

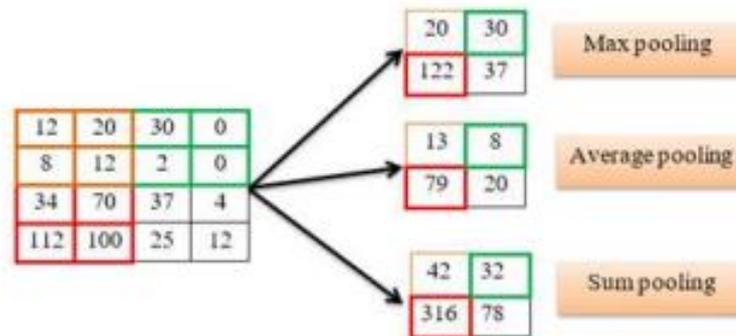


Figure 17: Type de la couche de regroupement.

d. **Couche d'aplanissement (flattening)**

L'aplatissement consiste à convertir tous les tableaux bidimensionnels résultants en un seul vecteur linéaire long et continu vecteur linéaire continu [27].

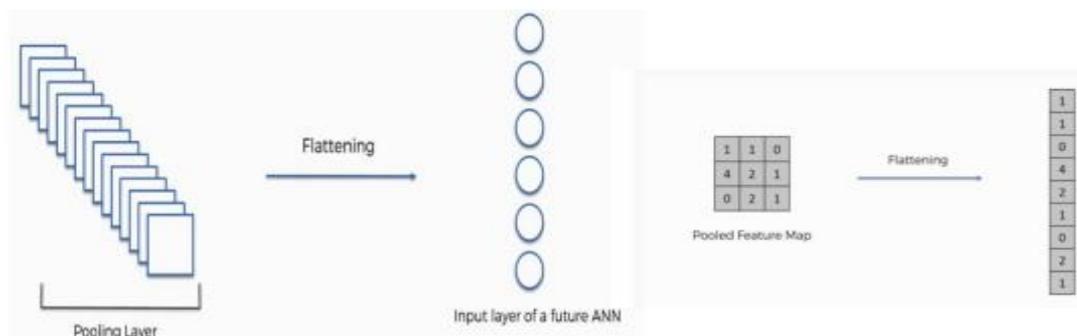


Figure 18: Schéma représentant la couche de flattening.

Le processus de construction d'un CNN comporte quatre grandes étapes :

1. Convolution(conv1D)
2. Mise en commun (pooling)
3. Aplatissement (Flatten)
4. Connexion complète (dense)

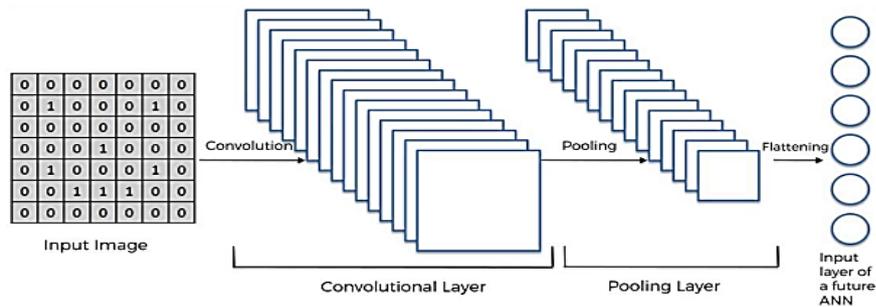


Figure 19: Schéma représentant l'importance de la couche flattig

Le flatten devient donc l'entrée RNA qui est utilisé pour la méthode de rétropropagation.

e. **Couche totalement connectées (Fully connected ou Dense)**

Après plusieurs couches de convolution et de mise en commun, le raisonnement de haut niveau dans les réseaux de neurones se fait au travers de couches entièrement connectées, dont le rôle est de classer en fonction des caractéristiques extraites par convolution. En règle générale, la couche finale entièrement connectée contient une fonction d'activation softmax qui fournit une valeur de probabilité comprise entre 0 et 1 pour chaque étiquette de classe que le modèle tente de prédire.

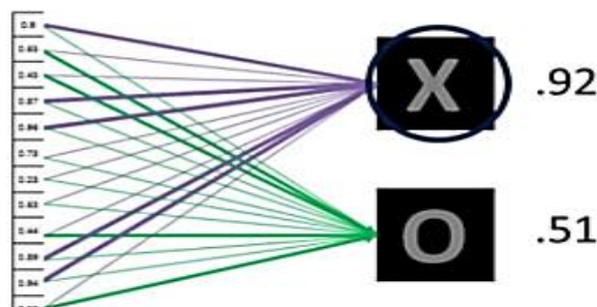


Figure 20 : Schéma représentant la couche de Fully connected.

f. **Couche de perte (Loss)**

La couche de perte spécifie comment la formation du réseau pénalise les écarts entre le signal attendu et le signal réel.

Il s'agit généralement de la dernière couche du réseau, et diverses fonctions de perte adaptées à différentes tâches peuvent être utilisées ici. La fonction "Soft max" permet de calculer la distribution de probabilité des classes de sortie.

2.4.5 Exemples de modèles de CNN

Les CNN utilisent plus de paramètres qu'un MLP standard. Même si les règles habituelles pour les taux d'apprentissage et des constantes de régularisation s'appliquent toujours, il faut prendre en considération les notions de nombre de filtres, leur forme et la forme du max pooling. La forme la plus commune d'une architecture CNN empile quelques couches Conv-ReLU, les suit avec des couches Pool, et répète ce schéma jusqu'à ce que l'entrée soit réduite dans un espace d'une taille suffisamment petite. À un moment, il est fréquent de placer des couches entièrement connectées (FC). La dernière couche entièrement connectée est reliée vers la sortie. Voici quelques architectures communes CNN qui suivent ce modèle :

- INPUT -> CONV -> RELU -> FC
- INPUT -> [CONV -> RELU -> POOL] * 2 -> FC -> RELU -> FC

Ici, il y a une couche de CONV unique entre chaque couche POOL

- INPUT -> [CONV -> RELU -> CONV -> RELU -> POOL] * 3 -> [FC -> RELU] * 2 -> FC

Ici, il y a deux couches CONV empilées avant chaque couche POOL.

L'empilage des couches CONV avec de petits filtres de pooling (plutôt un grand filtre) permet un traitement plus puissant, avec moins de paramètres. Cependant, avec l'inconvénient de demander plus de puissance de calcul (pour contenir tous les résultats intermédiaires de la couche CONV). On dispose maintenant dans la littérature de plusieurs modèles d'architecture dont l'efficacité varie selon la tâche, car ils n'ont pas tous le même nombre de convolutions (ni la même structure), citons notamment [29] :

- ❖ LeNet (le plus simple, pour découvrir les CNN).
- ❖ GoogLeNet (parent du célèbre Inception, un algorithme de reconnaissance d'image avec d'autres subtilités, au même titre que YOLO, MobileNet, etc...).
- ❖ VGGNet (encore très utilisé et puissant).
- ❖ ResNet (pareil, même s'il s'éloigne d'un CNN traditionnel).

2.4.6 Rétropropagation du gradient (Optimisation)

L'entraînement d'un CNN consiste à optimiser les coefficients du réseau, à partir d'une initialisation aléatoire, pour minimiser l'erreur de classification en sortie. Les deux parties des CNN sont entraînées simultanément : on apprend à la fois les coefficients des

noyaux de convolutions pour extraire des caractéristiques pertinentes, et la bonne combinaison de ces caractéristiques.

En pratique, les coefficients du réseau sont modifiés de façon à corriger les erreurs de classification rencontrées, selon une méthode de descente de gradient. Ces gradients sont rétropropagés dans le réseau depuis la couche de sortie, d'où le nom rétropropagation du gradient (back propagation en anglais) donné aux algorithmes d'entraînement des réseaux de neurones.

L'entraînement par batch (lot) consiste à rétropropager l'erreur (mise à jour du gradient) de classification par groupes d'images (Ensemble d'exemples utilisés dans une itération). Cette méthode est plus rapide qu'en calculant l'erreur sur tout le jeu d'entraînement à chaque itération. Elle est plus stable qu'en travaillant image par image, car les gradients d'erreurs ont moins de variance. À noter qu'un nombre trop important d'images par batch peut engendrer des problèmes de mémoire lors de l'exécution du code [29].

2.4.7 Découpage de données

L'ensemble des données ou Dataset à disposition est découpé selon 2 voire 3 parties :

- **Données d'apprentissage** : ou training set qui représente environ 60% des données mise à disposition, elles serviront à entraîner notre modèle.
- **Données de test** : ou Test set représente 20% des données globales elles serviront durant la phase de test du modèle.
- **Données de validation** : représente une portion 20% des données qui vont servir à tester le modèle, pour valider les résultats de l'entraînement. Ses données ne servent pas évaluer l'efficacité d'un modèle, mais plutôt des algorithmes entre eux.

Remarque : La validation de l'entraînement n'est pas obligatoire.

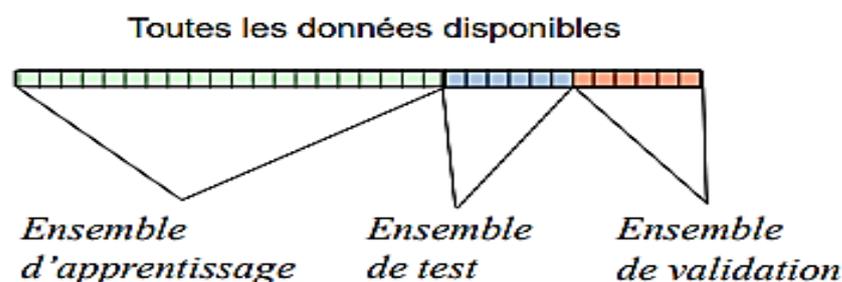


Figure 21: Découpage des données.

2.4.8 Les fonctions d'activation

Les fonctions d'activation sont des fonctions mathématiques utilisées dans les réseaux de neurones artificiels, qui sont une composante clé de l'apprentissage en profondeur (Deep Learning).

Il existe plusieurs types de fonctions d'activation couramment utilisées en deep learning, notamment :

a. **Relu**

La fonction d'activation Relu (RectifiedLinear Unit) est une fonction simple mais très efficace dans les réseaux de neurones. Elle renvoie la valeur de l'entrée si celle-ci est positive, et 0 sinon. L'avantage de la fonction Relu est qu'elle est facile à calculer et qu'elle permet de résoudre le problème de la disparition du gradient (vanishing gradient) qui peut se produire avec d'autres fonctions d'activation. Relu est donnée par l'équation [30] :

$$f(x) = \max(0, x)$$

b. **Softmax**

La fonction d'activation Softmax est souvent utilisée dans les réseaux de neurones pour la classification multi-classe. Elle prend en entrée un vecteur de valeurs réelles et renvoie un vecteur de probabilités normalisées (c'est-à-dire que la somme des probabilités est égale à 1). La fonction softmax est définie par l'équations [31] :

$$f(x) = \frac{e^x}{\sum e^x}$$

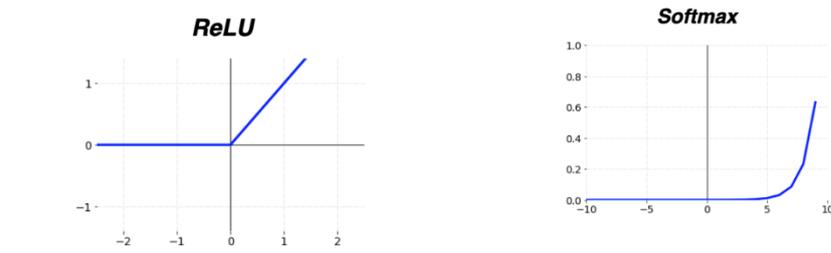


Figure 22 : Tracés des fonctions Relu et Softmax.

2.5 Conclusion

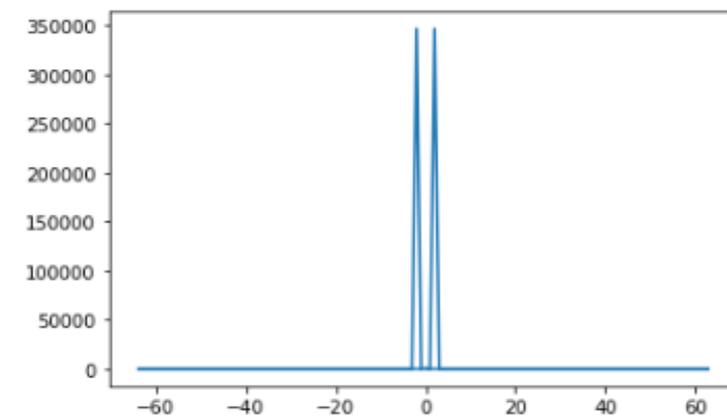
Dans ce chapitre, nous avons présenté les réseaux de neurones et nous avons détaillé le type de réseaux de neurones convolutifs. Ces réseaux sont capables d'extraire des caractéristiques des images présentées en entrée et de classer ces caractéristiques.

Ils sont basés sur la notion de « champs récepteurs » (receptive fields), ils mettent également en œuvre l'idée de partage de poids qui permet de réduire de nombreux paramètres libres de l'architecture. Ce partage de poids permet également de réduire les temps de calcul, l'espace mémoire nécessaire, et aussi d'améliorer les capacités de généralisation du réseau.

Chapitre 3

Implémentation et interprétations des résultats

```
yf = np.abs(fft(data_arr[8][0:128]))
xf = fftfreq(128,1/128)
plt.plot(xf, yf)
plt.show()
print("class",data_arr[8][128], "Normal wave")
```



class 1.0 Normal wave

Chapitre 3 Implémentation et interprétations des résultats

3.1 Introduction

Ce troisième chapitre traite du contexte expérimental ainsi que des différents éléments clés utilisés pour réaliser nos classifications. Ce chapitre aborde notamment les aspects suivants : le matériel utilisé, les bibliothèques utilisées, la base de données utilisée, les métriques utilisées pour évaluer nos résultats, une description détaillée du modèle CNN utilisé, ainsi que les résultats de nos classifications.

Dans ce chapitre, nous détaillerons chaque élément afin de fournir un aperçu complet du cadre expérimental dans lequel notre étude a été menée.

3.2 Contexte expérimental

Pour arriver à mettre en œuvre notre système, nous commencerons d'abord par des simulations de modèles. Le matériel et le logiciel de programmation nécessaires sont répertoriés. la base de données utilisée est ensuite présentée.

3.2.1 Google colab

a. *Définition*

Colab (short for Colaboratory), un outil d'apprentissage automatique et de recherche, conçu dans le cadre d'un projet de recherche Google. Générer et exécuter Jupyter Notebook sur Colab est super facile et gratuit.

Nous pouvons enregistrer notre ordinateur portable sur Google Drive ou GitHub et même entraîner notre apprentissage en profondeur sur GPU [29].



Figure 23: logo de google colab

b. **Avantages**

1. Cartographier Google Drive.
2. Travailler avec des fichiers de manière transparente sur un ordinateur.
3. Réduire les interactions manuelles à chaque exécution.
4. Ne pas compiler les bibliothèques à chaque exécution, juste une fois.
5. Nettoyer le dossier racine sur Google Drive.
6. Copier les ensembles de données sur le système de fichiers local VM pour améliorer les vitesses d'entraînement.
7. Travailler avec les fichiers de configuration directement depuis l'ordinateur.
8. Obtenir les poids entraînés directement synchronisés sur l'ordinateur en temps réel pendant l'entraînement.
9. Formation.

3.2.2 Python

Python est un langage de programmation puissant et facile à apprendre. Il possède des structures de données de haut niveau efficaces et une approche simple mais efficace de la programmation orientée objet. La syntaxe élégante et le typage dynamique de Python, ainsi que sa nature interprétée, en font un langage idéal pour le développement de scripts et d'applications rapides dans de nombreux domaines. Nature interprétée, en font un langage idéal pour le développement de scripts et d'applications rapides dans de nombreux domaines et sur la plupart des plates-formes.

L'interpréteur Python et la bibliothèque standard étendue sont disponibles gratuitement sous forme de source ou de binaire pour toutes les principales plates-formes sur le site Web de Python, <http://www.python.org/>, et peuvent être distribués librement. Le même site contient également des distributions et des pointeurs vers de nombreux modules,

programmes et outils Python tiers gratuits, ainsi qu'une documentation supplémentaire [32].



Figure 24: Logo de langage Python

3.2.3 Matériels utilisés

On a réussi à implémenter notre code sur notre ordinateur avec des configurations suivant :

- Pc portable HP EliteBook
- CPU 2,50 GHZ
- Mémoire installée 4 GO
- Système : Windows 10 64 bits
- Système d'exploitation Windows 10 Professionnel 64 bit
- Processeur 1
- Disque dur (SSD) 128GB

3.2.4 Bibliothèques utilisées dans l'implémentation

a. ***Pandas***

Pandas est une bibliothèque open source sous licence BSD fournissant des structures de données hautes performances et faciles à utiliser, ainsi que des outils d'analyse des données pour le langage de programmation Python. Pandas est un projet sponsorisé par NumFOCUS. Cela contribuera au succès du développement de pandas en tant que projet open source de classe mondiale et permettra de faire un don au projet [33].

b. ***Matplotlib***

C'est une bibliothèque de traçage pour le langage de programmation Python et son extension mathématique numérique NumPy. Il fournit une API orientée objet permettant d'incorporer des graphiques dans des applications à l'aide de kits d'outils d'interface graphique à usage général tels que Tkinter, Python, Qt ou GTK [34].

c. ***Numpy***

C'est une bibliothèque permettant d'effectuer des calculs numériques avec Python. Elle introduit une gestion facilitée des tableaux de nombres, des fonctions sophistiquées (diffusion), on peut aussi l'intégrer le code C / C ++ et Fortran [35].

d. ***Tensorflow***

Tensorflow est un Framework de programmation pour le calcul numérique qui a été rendu Open Source par Google en Novembre 2015. Depuis sa parution, TensorFlow n'a cessé de gagner en popularité, pour devenir très rapidement l'un des frameworks les plus utilisés pour le Deep Learning et donc les réseaux de neurones.

Son nom est notamment inspiré du fait que les opérations courantes sur des réseaux de neurones sont principalement faites via des tables de données multidimensionnelles, appelées Tenseurs (Tensor). Un Tensor à deux dimensions est l'équivalent d'une matrice. Aujourd'hui, les principaux produits de Google sont basés sur TensorFlow : Gmail, Google Photos, Reconnaissance de voix, et de caractères [36].

3.2.5 **Bases de données (Data set)**

Dataset (ou jeux de données) est couramment utilisé en Machine Learning. Il regroupe un ensemble de données cohérents qui peuvent se présenter sous différents formats (textes, chiffres, images, vidéos, etc.) [37]. Techniquement, un data set peut avoir une structure tabulaire, par exemple un fichier CSV (Comma-Separated Values), une structure d'arbre, comme dans un fichier JSON ou XML, ou encore une structure de graphe, comme dans le RDF (Reconnaissance De Formes).

Dans notre cas on s'intéresse par la dataset de type csv. Cette dataset contient 129 colonnes et 11998 lignes de données relatives à des défauts sur le signal de réseau

électrique. Les données sont réparties en cinq classes distinctes, chacune représentant un type de défaut spécifique. Chaque colonne contient des informations sur une caractéristique particulière du signal, telles que la fréquence, l'amplitude ou la phase. Les lignes représentent des échantillons de signal spécifiques, capturés à différents moments et endroits du réseau électrique.

On utilise ce data set pour entraîner un modèle de Deep Learning afin de classer automatiquement les différents types de défauts dans le réseau électrique.

Le dataset (jeux de données) utilisé pour ce projet a été fourni par Kaggle. On peut la trouver ici sur ce site <https://www.kaggle.com/code/jaideepreddykotla/power-quality-classification-using-cnn-1/>

	Col1	Col2	Col3	Col4	Col5	Col6	Col7	Col8	Col9	Col10
0	-797.962914	-1320.199586	-1828.741445	-2319.935857	-2789.218649	-3231.111864	-3641.962908	-4018.119193	-4355.929647	-4650.829055
1	4387.885674	4567.747164	4693.741356	4776.824895	4824.300958	4850.777519	4862.646175	4867.211521	4866.298624	4858.081691
2	-4050.074299	-4340.408587	-4576.875995	-4769.518913	-4918.338175	-5025.159834	-5100.026323	-5141.111052	-5153.892968	-5137.458861
3	3458.450134	3686.700705	4000.773209	4392.451258	4834.343975	5269.845213	5636.871457	5876.078192	5950.944421	5841.384319
4	-910.262100	-312.245917	298.552078	896.568469	1462.629392	1978.474763	2439.540162	2847.652177	3212.852886	3547.924521

5 rows x 129 columns

Figure 25 : affichage de base de données

...	Col1120	Col1121	Col1122	Col1123	Col1124	Col1125	Col1126	Col1127	Col1128	output
...	3632.834002	3221.982426	2780.090228	2311.720161	1821.438861	1311.983509	790.659456	262.031836	-269.335205	1
...	-165.252287	524.062959	1199.684611	1842.437384	2437.714882	2972.733259	3437.450839	3828.215741	4143.201102	2
...	563.322919	-47.475988	-656.447463	-1252.637741	-1825.089177	-2364.673580	-2861.346591	-3309.630078	-3706.785897	2
...	-722.183210	67.562203	850.005032	1561.232918	2152.858410	2601.141643	2912.474970	3123.378574	3286.805684	3
...	-4506.574476	-4196.154858	-3881.168803	-3560.705563	-3225.633496	-2860.433401	-2452.321914	-1991.255895	-1474.497200	3

Figure 26 : affichage de base de données (suite)

3.3 Extraction des paramètres

Après avoir obtenu une base de données relatives aux défauts sur le signal du réseau électrique, nous avons entrepris une étude approfondie en utilisant l'environnement de collaboration (Colab). Pour analyser ces données, nous avons suivi les étapes suivantes :

- Étape 1 : Nous téléchargeons la base de données sur Google Drive.
- Étape 2 : décompresser le fichier zip a fichier csv.
- Étape 3 : chargement et affichage de la base de données.

- Étape 4 : Dans ce segment, nous traçons une vague de chaque classe après application de la transformation de Fourier.
- Étape 5 : nous divisons l'ensemble de données dans un rapport de 60 %, 20 %, 20 % (ensemble d'entraînement, ensemble de validation, ensemble de test).
- Étape 6 : training le modèle après utilisé le tensorboard pour donner les histogrammes des couches de Cnn et les graphes précision et perte.
- Étape 7 : on évalue le modèle avec matrice de confusion

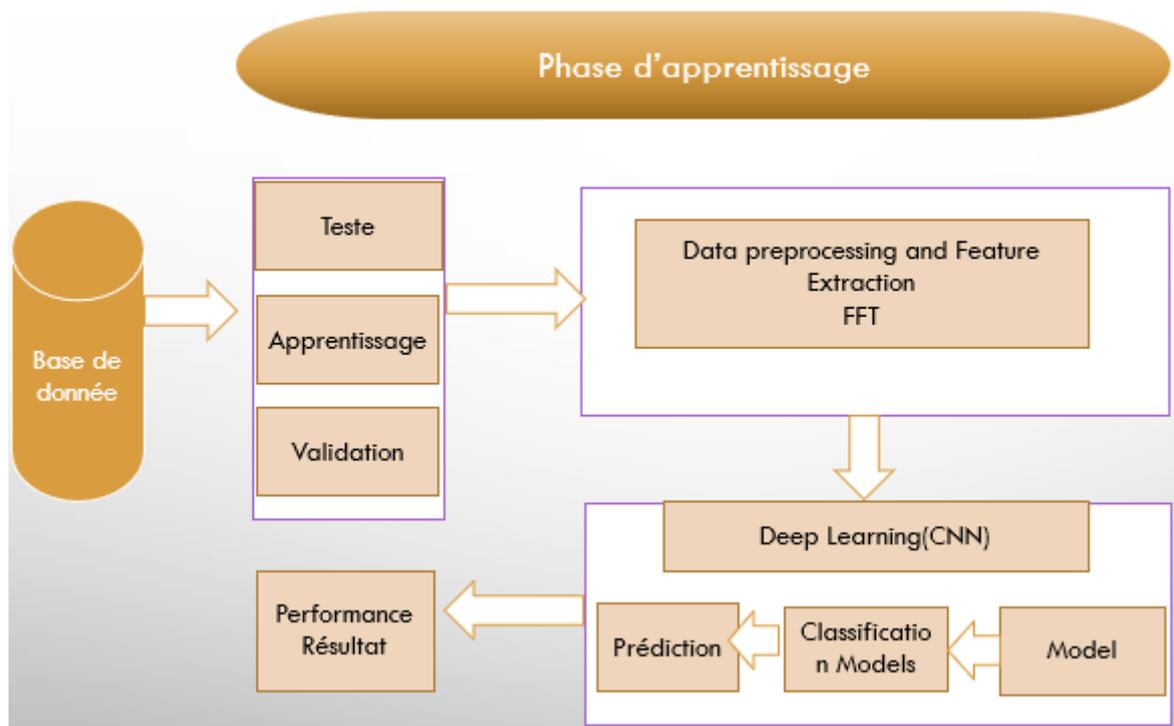


Figure 27 : Diagramme montre les étapes de code.

3.4 Métriques

3.4.1 Précision

L'une des métriques les plus évidentes, c'est la mesure de tous les cas correctement identifiés. Il est surtout utilisé lorsque toutes les classes sont d'égale importance [29].

$$\text{Accuracy} = \frac{\text{true positive} + \text{true negative}}{(\text{true positive} + \text{false positive} + \text{true negative} + \text{false negative})}$$

3.4.2 Perte

Perte est une mesure qui fournit l'écart entre les prédictions du modèle et les valeurs réelles des données d'entraînement. Utilisée pour quantifier à quel point le modèle est performant dans la tâche qu'il doit accomplir.

3.5 La description d'architecture de modèle CNN

Le modèle présenté est un réseau de neurones convolutif (CNN) avec plusieurs couches.

- 1^{ère} couche est une couche de convolution (Conv1d) qui comprend 128 filtres de taille 3, une fonction d'activation Relu et un rembourrage ("same"). L'entrée du modèle est conçue avec la forme "input Shape". La couche de convolution extrait des caractéristiques importantes de l'entrée.

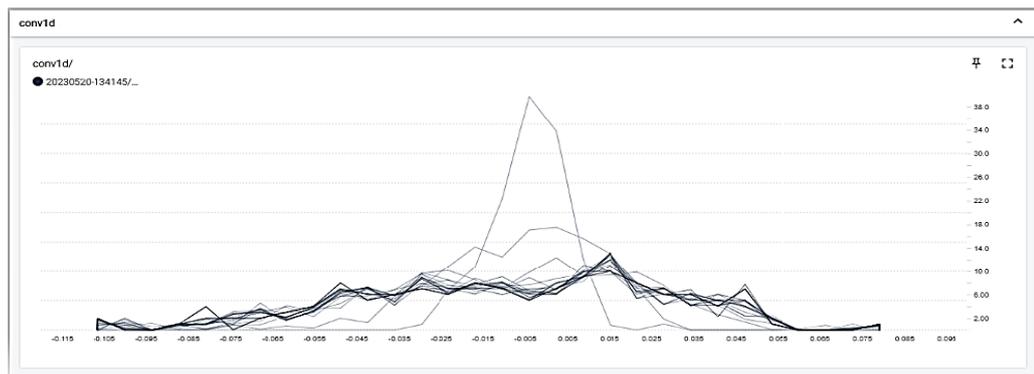


Figure 28 : Graphe représentant couche convolution (conv1d).

- 2^{em} couche est une normalisation de lots (BatchNormalization) qui normalise les activations de la couche précédente. Cette normalisation est effectuée pour accélérer l'apprentissage et améliorer la stabilité du modèle.

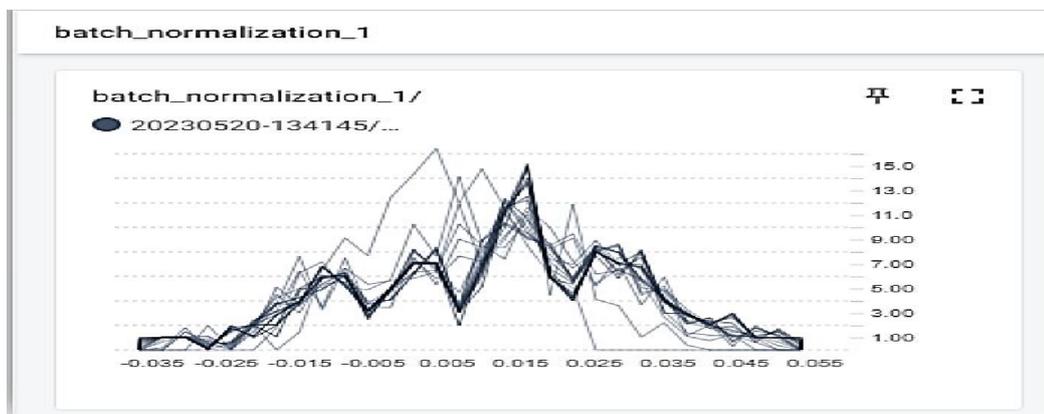


Figure 29 : Graphe représentant couche BatchNormalization.

- 3^{em} couche est une couche de regroupement maximal (MaxPooling1D) qui réduit la taille de la sortie de la couche précédente de moitié en prenant la valeur maximale de chaque fenêtre de taille 2. Cette couche réduit la dimensionnalité de la sortie et permet une généralisation efficace.
- 4^{em} et 5^{em} couches sont similaires à la première et à la 2^{em} couches, respectivement, avec des filtres de convolution de taille 128 et des noyaux de taille 3, suivis d'une normalisation de lots et d'un regroupement maximal.

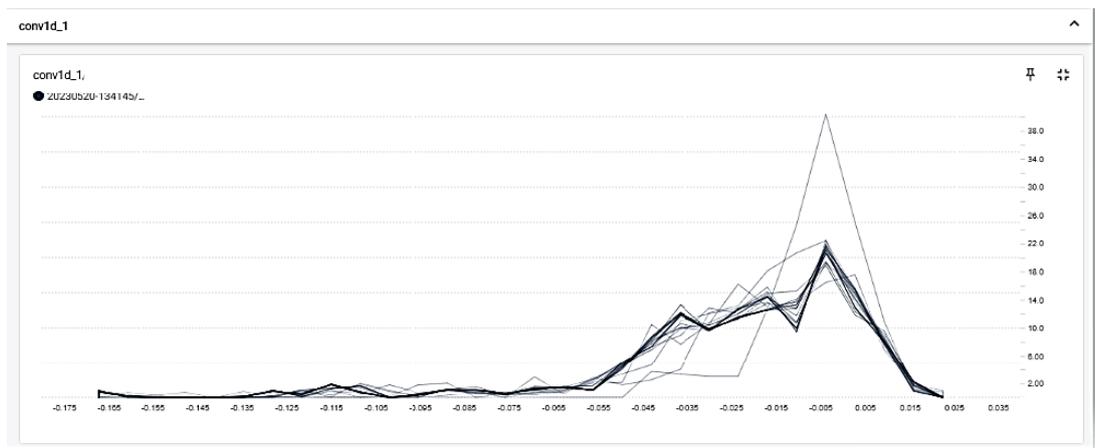


Figure 30 : Graphe représentant couche convolution (conv1d_1).

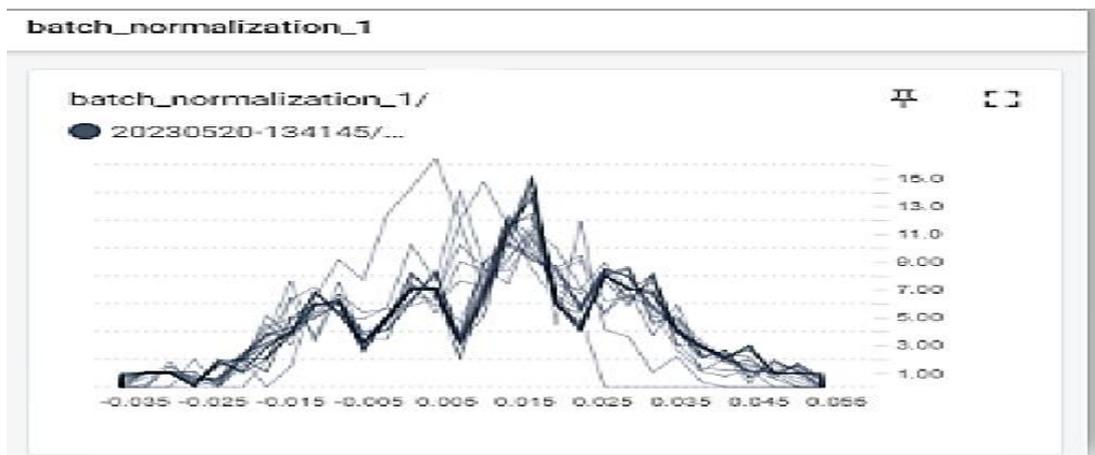


Figure 31 : Graphe représentant couche BatchNormalization_1.

- 6^{em} couche est une couche de nivellement (Flatten) qui transforme la sortie de la dernière couche de convolution en un vecteur plat.

- 7^{em} couche est une couche entièrement connectée (Dense) avec 16 neurones et une fonction d'activation ReLU. Cette couche permet une combinaison non linéaire des caractéristiques extraites des couches de convolution précédentes.

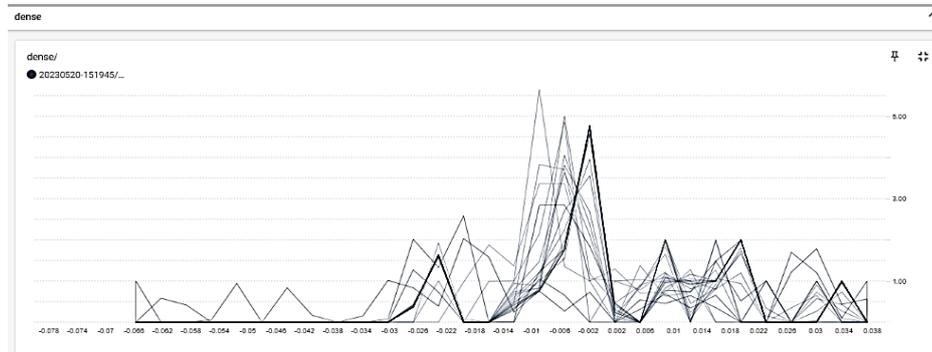


Figure 32 : Graphe représentant couche dense.

- 8^{em} et dernière couche est une couche Dense avec un nombre de neurones égal au nombre de classes dans le jeu de données (num_classes). Cette couche utilise une fonction d'activation softmax pour produire une distribution de probabilité sur les classes de sortie.

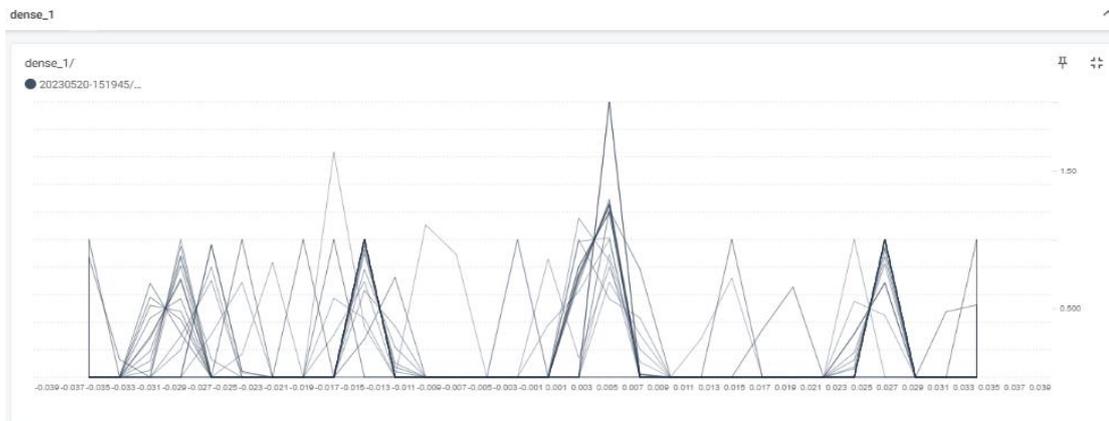


Figure 33 : Graphe représentant couche dense_1.

Le modèle a donc une architecture de type CNN avec plusieurs couches de convolution, de normalisation de lots, de regroupement maximal et de couches entièrement connectées pour produire une sortie de classification. Il est couramment utilisé pour la classification d'images ou de signaux séquentiels.

Model: "sequential"

Layer (type)	Output Shape	Param #
conv1d (Conv1D)	(None, 128, 128)	512
batch_normalization (Batch Normalization)	(None, 128, 128)	512
max_pooling1d (MaxPooling1D)	(None, 64, 128)	0
conv1d_1 (Conv1D)	(None, 64, 128)	49280
batch_normalization_1 (Batch Normalization)	(None, 64, 128)	512
max_pooling1d_1 (MaxPooling1D)	(None, 32, 128)	0
flatten (Flatten)	(None, 4096)	0
dense (Dense)	(None, 16)	65552
dense_1 (Dense)	(None, 5)	85

Total params: 116,453
Trainable params: 115,941
Non-trainable params: 512

Figure 34 : Modèle séquentiel de CNN.

3.6 Résultat de classification

3.6.1 Visualiser les FFT des signaux (les classes)

```

yf = np.abs(fft(data_arr[0][0:128]))
xf = fftfreq(128,1/128)
plt.plot(xf, yf)
plt.show()
print("class",data_arr[0][128], "Normal wave")

yf = np.abs(fft(data_arr[1][0:128]))
xf = fftfreq(128,1/128)
plt.plot(xf, yf)
plt.show()
print("class",data_arr[1][128], "3rd harmonic wave")

yf = np.abs(fft(data_arr[3][0:128]))
xf = fftfreq(128,1/128)
plt.plot(xf, yf)
plt.show()
print("class",data_arr[3][128], "5th harmonic wave")

yf = np.abs(fft(data_arr[6][0:128]))
xf = fftfreq(128,1/128)
plt.plot(xf, yf)
plt.show()
print("class",data_arr[6][128], "Voltage dip")

yf = np.abs(fft(data_arr[8][0:128]))
xf = fftfreq(128,1/128)
plt.plot(xf, yf)
plt.show()
print("class",data_arr[8][128], "Transient wave")

```

Figure 35 : code pour ploter les classes.

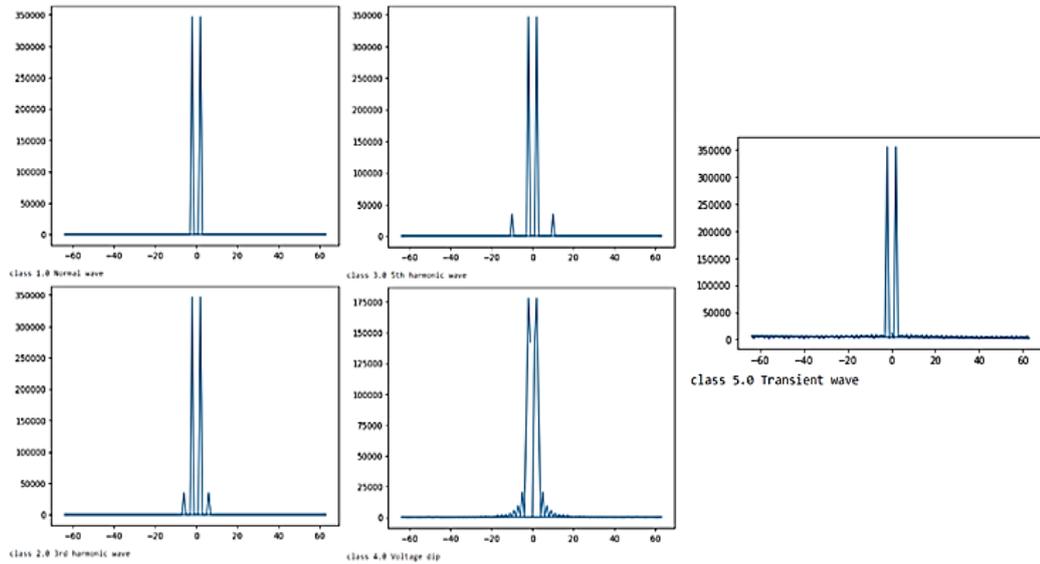


Figure 36 : Graphes représentent la transformation de fourier des signaux.

On a appliqué la FFT sur les signaux des classes pour en ressortir l'amplitude, on obtient donc l'histogramme qui devient image (input) sur le modèle du CNN.

3.6.2 Précision et perte

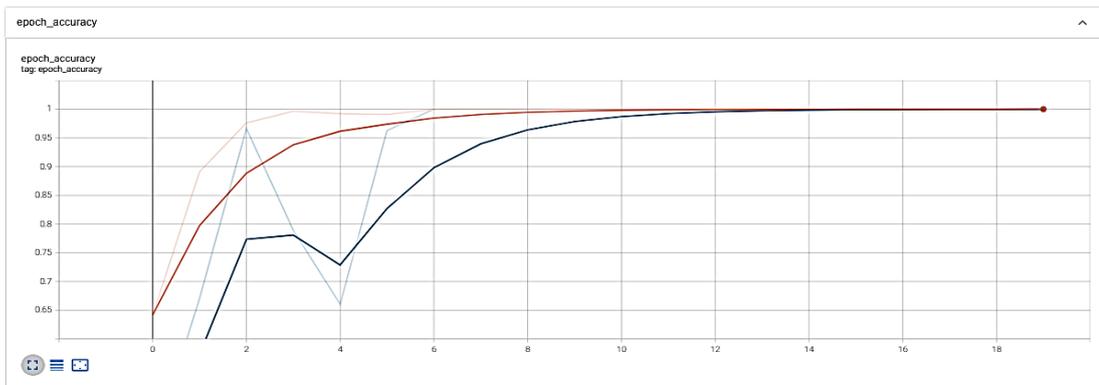


Figure 37 : Graphe représentant la précision du modèle.

On a la précision (accuracy) augmente jusqu'à accuracy =1.0000

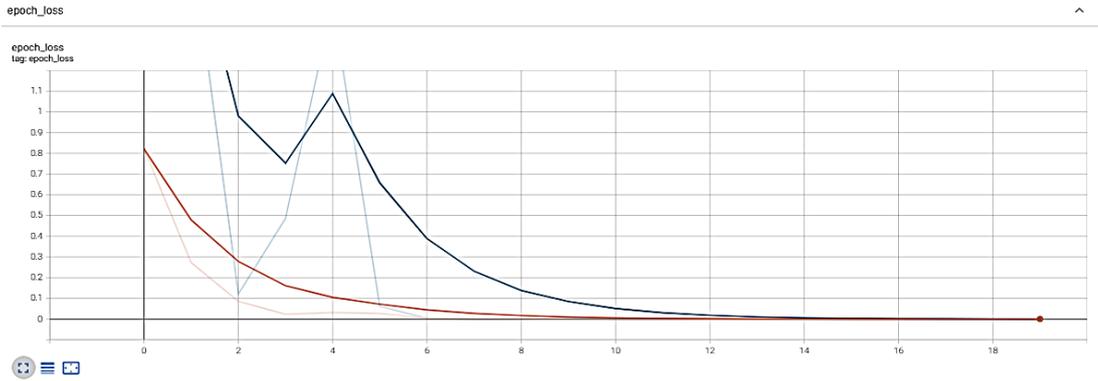


Figure 38 : Graphe représentant l'erreur du modèle.

On a perte (loss) descente jusqu'à $\text{loss} = 2.6585e-05$.

3.6.3 Matrice de confusion

a. Définition

La matrice de confusion est un outil simple mais puissant qui fournit une vue d'ensemble des performances d'un modèle de classification en mettant en évidence les prédictions correctes et incorrectes pour chaque classe.

		Réponse de l'expert	
		p	n
Réponse du classifieur	Y	Vrai Positif	Faux Positif
	N	Faux Négatif	Vrai Négatif

Figure 39 : Matrice de confusion.

b. Interprétation de la matrice de confusion

- La classe réelle 1 a été prédite correctement pour tous les échantillons, car il y a 410 prédictions correctes pour cette classe (dans la première ligne).
- La classe réelle 2 (signal pur) a également été prédite correctement pour tous les échantillons, car il y a 390 prédictions correctes pour cette classe (dans la deuxième ligne).

- La classe réelle 3 (3em onde harmonique) a également été prédite correctement pour tous les échantillons, car il y a 600 prédictions correctes pour cette classe (dans la troisième ligne).
- La classe réelle 4 (5em onde harmonique) a également été prédite correctement pour tous les échantillons, car il y a 420 prédictions correctes pour cette classe (dans la quatrième ligne).
- La classe réelle 5 (transitoire) a également été prédite correctement pour tous les échantillons, car il y a 580 prédictions correctes pour cette classe (dans la cinquième ligne).
- Cette matrice de confusion indique que le modèle a réalisé des prédictions parfaites pour toutes les classes, avec un nombre d'échantillons correctement classés égal au nombre total d'échantillons pour chaque classe. Cela suggère que le modèle a une performance très élevée et qu'il est capable de distinguer avec précision chaque classe dans notre problème de classification.

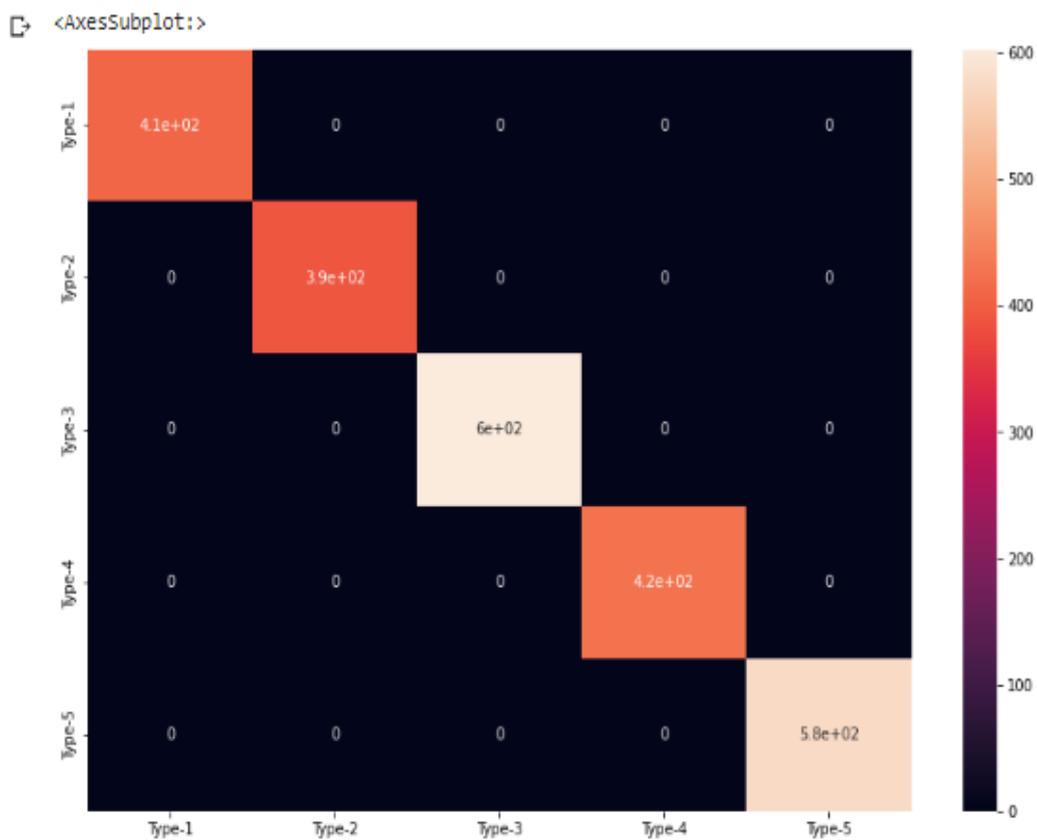


Figure 40 : Résultat la matrice de confusion.

3.7 Conclusion

Dans ce chapitre on a présenté la classification avec Deep Learning sur Python ou on a utilisé modèle d'apprentissage. Le système proposé repose sur modèle de Deep Learning à savoir le CNN. Ce modèle montre des performances très importantes pour classification défauts sur le signal du réseau électrique, et ce modèle donne un très bon résultat à partir des données de matrice de confusion, perte et précision.

Conclusion générale

En conclusion, ce projet de fin d'études intitulé "Classification des défauts dans les réseaux électriques à l'aide des réseaux de neurones convolutifs CNN" a abordé en profondeur les différentes étapes nécessaires à la mise en place d'un système de classification efficace pour les problèmes rencontrés sur un réseau électrique.

Le premier chapitre, "État de l'art", a présenté une revue complète sur les réseaux électriques et l'impact de l'intelligence artificielle sur la classification et ensuite les méthodes de classification existantes. Cela a permis de mettre en évidence les défis spécifiques liés à la classification des problèmes sur un réseau électrique.

Le deuxième chapitre, "Réseau de neurones", s'est concentré sur l'algorithme CNN (Convolutional Neural Network) et ses différentes couches. Une explication détaillée du fonctionnement du CNN a été fournie, mettant en évidence sa capacité à extraire automatiquement des caractéristiques pertinentes à partir des données.

Le dernier chapitre, "Implémentation et interprétation des résultats", a décrit le contexte expérimental et le matériel utilisé pour l'implémentation du modèle CNN. Une base de données appropriée a été utilisée, et le modèle CNN a été entraîné et évalué sur ces données. Les résultats de classification obtenus ont démontré une précision très élevée, ce qui indique que le modèle CNN est performant pour la classification des problèmes sur un réseau électrique.

En somme, ce projet a réussi à développer un modèle de classification robuste basé sur l'algorithme CNN, en se basant sur une recherche approfondie et une mise en œuvre soignée. Les résultats obtenus sont prometteurs et ouvrent la voie à des applications potentielles dans le domaine de la gestion et de la maintenance des réseaux électriques. Il est important de souligner que des travaux futurs pourraient encore améliorer les performances du modèle et explorer d'autres techniques d'apprentissage automatique pour une classification encore plus précise et efficace des problèmes sur les réseaux électriques.

1.1 Travaux cites

- [1] « Technico-Science.net, » 17 05 2023. [En ligne]. Available: <https://www.techno-science.net/glossaire-definition/Reseau-electrique.html>.
- [2] « Centre Universitaire Nour El Bachir El Bayadh Institut des Sciences Département de Technologie »
- [3] « Production, transport et distribution de l'énergie électrique, » 25 05 2023. [En ligne]. Available: https://fac.umc.edu.dz/fstech/cours/G%20Transport/M1-TDH_Cours%20Equipement%20Electrique.pdf.
- [4] F. Y. e. Y. Simhamed, «Automatic Classification Mechanism for the Two Most Common Power Quality Disturbances: Sag and swell, » LATSI laboratory Department of Electronics University of Blida, Algeria
- [5] P. Settipalli, «AUTOMATED CLASSIFICATION OF POWER QUALITY DISTURBANCES USING SIGNAL PROCESSING TECHNIQUES AND, » University of Kentucky, USK, 2007.
- [6] V. DIP, « SUNPOWER, » 09 05 2023. [En ligne]. Available: https://www.sunpower-uk.com/glossary/what-is-a-voltage-dip/?fbclid=IwAR1fjPTNU1kgjS4VSPI1WHEVLBPst_INdeHs_JYiLXVhQ-WS_KGKAhXNxSA.
- [7] C. D. PHAM, « Détection et localisation de défauts dans les réseaux de distribution HTA en présence de génération d'énergie dispersée ». Hanoï 19 09 2005.
- [8] F. D. G. T. B. de Metz-Noblat, «"Calcul des courants de courts circuits», Schneider Electric, Cahier technique, » n° %1158, 2000.
- [9] a. MAHJOUBY, « effet de l'intégration des énergies renouvelables sur les méthodes de locations des défauts dans les réseaux de distribution, » university sidi Mohamed ben Abdellah, 14-06-2016.
- [10] U. S. Sambit Dash, «Multiple Power Quality Event Detection and, » Department of Electrical Engineering, Odisha, India.
- [11] J. Krywyk, « Classification d'images les réseaux de neurones convolutifs en toute simplicité, » Le Blog des octos , 25/10/2016.

- [12] R. H. a. Lyon, «Traitement du signal et étude des réseaux : application à l'étude du système de vélos en libre-service à Lyon,» Traitement du signal, 29/02/2016.
- [13] «NetAPP,» Qu'est-ce que l'intelligence artificielle ?, [En ligne]. Available: <https://www.netapp.com/fr/artificial-intelligence/what-is-artificial-intelligence/>. [Accès le 25 05 2023].
- [14] R. SCH Melzer, « Le mag it,» 17 12 2020. [En ligne]. Available: <https://www.lemagit.fr/conseil/Les-15-techniques-de-data-science-a-connaître>. [Accès le 19 04 2023].
- [15] 20 06 2020. [En ligne]. Available: <https://ia-data-analytics.fr/machine-learning/>.
- [16] C. DELUZARCHE, «Deep Learning : qu'est-ce que c'est ?,» 07 02 2021. [En ligne]. Available: <https://www.futura-sciences.com/tech/definitions/intelligence-artificielle-deep-learning-17262/>.
- [17] S. Senani, «Réseau de neurones convolutionnels pour la détection précoce de la rétiopathie diabétique,» Tizi-Ouzou, 2019.
- [18] I. C. Kaadoud, «POUR LA SCIENCE BLOGS,» 08 10 2018. [En ligne]. Available: <https://scilogs.fr/intelligence-mecanique/reprenons-bases-neurone-artificiel-neurone-biologique/>. [Accès le 19 04 2023].
- [19] I. Ghania, «« Régression et modélisation par les réseaux de neurones »,» UNIVERSITE MENTOURI -CONSTANTINE, 30 juin 2009 .
- [20] B. L, «Lebigdata,» 05 04 2019. [En ligne]. Available: <https://www.lebigdata.fr/reseau-de-neurones-artificiels-definition>. [Accès le 29 03 2023].
- [21] Y. LeCun, B. Boser et J. S. Denker, «Back propagation Applied to Handwritten Zip Code Recognition,» Décembre 1989. [En ligne]. Available: <https://ieeexplore.ieee.org/document/6795724>. [Accès le 17 04 2023].
- [22] [En ligne]. Available: <https://openclassrooms.com/fr/courses/4470531-classez-et-segmentez-des-donnees-visuelles/5082166-quest-ce-quun-reseau-de-neurones-convolutif-ou-cnn>.
- [23] S. H. e. J. Schmid Huber, «ACM (ASSOCIATION FOR COMPUTING MACHINERY),» ACM DEGITAL LYBRARY , 01 11 1997. [En ligne]. Available: <https://dl.acm.org/doi/10.1162/neco.1997.9.8.1735#sec-comments>.
- [24] G. H. E. G. D. E. J. Martens, «ACM (ASSOCIATION FOR COMPUTING MACHINERY),» Juin 2013. [En ligne]. Available: <https://dl.acm.org/doi/10.5555/3042817.3043064>.
- [25] F. chollot, « Apprentissage en profondeur avec python,» manning Publication , 2018.

- [26] E.-B. H. a. M. L. Ahmed El-Sawy, «Cnn for handwritten arabic digits recognition based on lenet-5,» chez In International Conference on Advanced, SPRINGER, 2016, pp. 566-575.
- [27] M. Z. A. Amir Al, «A Briefly Explanation of Convolutional Neural Network with Practical Implementation in Keras & TensorFlow,» chez Machine Learning & Deep Learning Researcher.
- [28] A. Lahreche, «Deep learning for Arabic letters recognition,» UNIVERSITY KASDI-MERBAH OUARGLA, Faculty of New Technologies of Information and Communication ,Department of Computer Science and Information Technologies, 2018/2019.
- [29] A. houda Benzeghda, « La reconnaissance des caractères Arabes manuscrits par le mécanisme d'Attention,» UNIVERSITE LARBI BEN M'HIDI, 2020/2021.
- [30] G. Hinton, «A Practical Guide to Training Restricted Boltzmann Machines, » Technical Report UTML TR 2010-003, University of Toronto, 2010.
- [31] I. B. Y. C. A Goodfellow, «deep learning, » MIT Press, 2016.
- [32] J. Fred L. Drake, «Python tutorial, » p. 1, 18 06 2012.
- [33] « pandas,» [En ligne]. Available: <https://pandas.pydata.org/>. [Accès le 13 05 2023].
- [34] «matplotlib,» [En ligne]. Available: <https://matplotlib.org/>. [Accès le 13 05 2023].
- [35] «numpy,» [En ligne]. Available: <https://www.numpy.org/>.. [Accès le 13 05 2023].
- [36] «tensorflow,» [En ligne]. Available: <https://www.tensorflow.org/>. [Accès le 14 05 2023].
- [37] « DATASIENCETEST,» Qu'est-ce qu'un dataset ? Comment le manipuler ? 23 08 2005. [En ligne]. Available: [https://datascientest.com/dataset-definition#:~:text=Les%20datasets%20\(ou%20jeux%20de,images%2C%20vid%C3%A9os%20etc%E2%80%A6\)..](https://datascientest.com/dataset-definition#:~:text=Les%20datasets%20(ou%20jeux%20de,images%2C%20vid%C3%A9os%20etc%E2%80%A6)..)