REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE YAHIA FARES DE MEDEA



Faculté de Technologie

Mémoire de fin d'études de Master

Filière : Génie électrique

Spécialité : Systèmes des télécommunications

Classification des modulations par l'utilisation des réseaux neuronaux profonds

---

Présenté par :

ATCHI Anes

HAMDANE Abdelmalek

Proposé et Dirigé par :

Dr. DALI Mohammed

Dr LAIDI Kamel

Promotion 2024-2025

Remerciements

Avant d'entamer la présentation de ce travail, il nous tient à cœur d'exprimer nos remerciements les plus sincères à toutes les personnes qui ont contribué, de près ou de loin, à la réalisation de ce mémoire.

Nous remercions tout d'abord Allah, le Tout-Puissant et le Très Miséricordieux, de nous avoir accordé la santé, la force, la patience et la persévérance nécessaires à l'aboutissement de ce projet. Sans Sa guidance, rien n'aurait été possible.

Nos remerciements vont également à l'ensemble du corps enseignant et administratif de l'Université Yahia Farès de Médéa, pour la qualité de leur encadrement tout au long de notre formation. Leur disponibilité, leur rigueur et leur engagement ont grandement enrichi notre parcours académique.

Nous adressons notre plus profonde gratitude à Mr DALI Mohamed, notre encadrant, pour sa confiance, sa bienveillance, ses conseils précieux, ainsi que pour son accompagnement tout au long de la réalisation de ce mémoire. Son expertise et sa rigueur scientifique ont été d'un apport inestimable.

Nous exprimons aussi notre reconnaissance envers nos familles, pour leur amour inconditionnel, leur soutien moral et leur patience tout au long de notre parcours universitaire. Leur présence et leurs encouragements ont été essentiels dans chaque étape de notre chemin.

Enfin, nous remercions les membres du jury pour l'intérêt porté à notre travail, ainsi que toutes les personnes, connues ou anonymes, qui ont contribué d'une manière ou d'une autre à la réussite de ce projet.

Trouvez dans ces mots l'expression de notre profonde reconnaissance.

Dédicace

Ce mémoire représente bien plus qu'un simple travail académique. Il est le reflet d'un parcours fait de défis, d'efforts constants, de remises en question, de nuits blanches et de moments de doute, mais aussi d'apprentissage, de satisfaction et de petites victoires. Je le dédie, avec tout mon respect et ma gratitude, à celles et ceux qui ont contribué, de près ou de loin, à sa réalisation.

À ma chère maman,
Toi qui es la lumière de ma vie, le cœur battant de notre foyer, et la source de tant de bénédictions. Ton amour inconditionnel, ta patience dans les moments difficiles, ton soutien discret mais toujours présent, et tes prières silencieuses ont été mon refuge dans les périodes d'incertitude. Tu es mon exemple de force, de courage et de générosité. Ce mémoire est le fruit de ton éducation et de tes sacrifices, et je te le dédie avec tout l'amour que tu m'as donné.

À mon cher père,
Ton calme, ta sagesse et ta manière de toujours croire en moi même lorsque je doutais ont été une véritable force. Tu m'as transmis des valeurs de rigueur, d'honnêteté et de responsabilité que j'ai essayé d'appliquer tout au long de mon parcours. Merci pour ton appui constant, tes encouragements, et ta discrète mais précieuse présence.

À mes frères,
Vous êtes mes complices de toujours. Merci pour votre soutien sans condition, pour vos gestes d'encouragement, vos paroles simples mais toujours sincères, et pour la force tranquille que vous m'avez transmise. Vos parcours, vos efforts, vos conseils ont été pour moi une source d'inspiration. Chacun de vous a contribué à sa manière à ce que ce mémoire voie le jour.

À ma sœur,
Merci pour ton écoute précieuse, ton regard bienveillant et ta capacité à me motiver même dans mes pires moments de doute. Ta sensibilité, ton attention et ton amour fraternel ont eu un rôle important dans mon équilibre tout au long de cette aventure.

À mes amis,
Merci pour votre présence, vos encouragements, vos discussions qui m'ont permis de souffler, vos conseils toujours avisés, et votre humour salvateur. Dans les moments de pression ou de découragement, votre amitié a été une véritable bouffée d'air. Ce mémoire porte aussi les traces de ces échanges humains sincères et précieux.

À tous les professeurs
qui ont marqué mon parcours, par leur passion, leur rigueur et leur bienveillance. Merci pour les connaissances transmises, les échanges inspirants et les moments marquants que je n'oublierai jamais.

Anes ATCHI

Dédicace

Je dedie ce travail :

A mes parents,
Pour leur amour inestimable, leurs sacrifices silencieux, leur patience infinie, et leur soutien constant qui m'ont permis d'arriver jusque-la. Merci d'avoir toujours cru en moi.

A mon frere,
Pour sa presence, ses conseils et ses encouragements dans les moments difficiles.

A mes soeurs,
Pour leur affection, leur joie communicative et leur soutien sincere.

A toute ma famille,
Pour leur bienveillance, leurs prieres et leur accompagnement durant tout mon parcours.

A mon binome,
Pour la collaboration harmonieuse, l'engagement et l'esprit d'equipe tout au long de ce projet.

A tous mes amis,
Pour leurs encouragements, leur ecoute et leur presence precieuse.

Abdelmalek HAMDANE

ملخص:

يتناول هذا البحث مشكلة *التصنيف الآلي لأنماط التضمين* بالاعتماد على تقنيات *التعلم العميق* .تم اعتماد نهجين: الأول باستخدام *شبكات عصبونية التفافية* (*CNN*) مصممة يدويًا، والثاني من خلال *التعلم بالنقل* (*Transfer Learning*) انطلاقًا من نموذج *SqueezeNet*.

تم إجراء التجارب على قاعدة البيانات *RML2016a* باستخدام ثلاث تمثيلات مختلفة :*مركبتا الطور والتربيع(I/Q)* ، *صور الطيف الزمني*(*Spectrogram*) ، *وصور ناتجة عن تحويل Wigner-Ville (WVD).* أظهرت النتائج أن البنى المقترحة كانت فعّالة بشكل خاص عند تطبيقها على بيانات *I/Q* و *WVD* مقارنة بتمثيلات *الطيف الزمني*.

الكلمات المفتاحية: التصنيف الآلي لأنماط التضمين، التعلم العميق، *شبكات عصبونية التفافية*، *مركبتا الطور والتربيع*، الطيف *الزمني*.

Résumé :

Ce mémoire traite de la problématique de *classification automatique de modulations* en s'appuyant sur les techniques de *Deep Learning*. Deux approches ont été explorées : des *réseaux de neurones convolutifs (CNN)* conçus manuellement et l'utilisation du *transfert learning* à partir du modèle *SqueezeNet*. Les expériences ont été menées sur la base de données *RML2016a*, exploitée sous trois représentations différentes : les composantes en phase et en quadrature *I/Q*, les *images de spectrogrammes*, et les images issues de la transformée de *Wigner-Ville (WVD)*. Les résultats montrent que les architectures sont particulièrement efficaces pour les données I/Q et WVD, par rapport au spectrogramme.

Mots-clés : Deep Learning, classification automatique de modulations, CNN, Transfert Learning, RML2016a, I/Q, Spectrogramme, WVD, RML2016a.

Abstract :

This thesis addresses the problem of *automatic modulation classification (AMC)* using *Deep Learning* techniques. Two approaches were explored: manually designed *Convolutional Neural Networks (CNNs)* and *transfer learning* based on the *SqueezeNet* model. Experiments were conducted using the *RML2016a* dataset, which was processed in three different representations: *in-phase and quadrature (I/Q) components, spectrogram images*, and images derived from the *Wigner-Ville Distribution (WVD*). The results show that the proposed architectures are particularly effective when applied to I/Q and WVD data, compared to spectrogram-based representations.

Keywords : Automatic Modulation Classification (AMC) Deep Learning, CNN, (I/Q) Wigner-Ville Distribution (WVD), spectrogram, RML2016a.

Sommaire

Remerciements II
Dédicace III
Résumé :..... V
Liste des figures..... IX
Liste des tableaux XII
Liste des abréviations XIII
Liste des symboles..... XIV

INTRODUCTION GENERALE 15

I. CHAPITRE I : MODULATIONS NUMERIQUES ET LEURS CLASSIFICATIONS 16
I.1. Introduction..... 17
I.2. Chaine de transmission numérique..... 17
I.2.1. Au niveau d'émetteur 18
I.2.2. Au niveau du récepteur..... 19
I.3. La modulation numérique 20
I.3.1. Principe de la modulation numérique..... 20
I.3.2. Caractéristiques et critères d'évaluation d'une modulation 22
I.4. Types de base de la modulation numérique 24
I.4.1. Modulation par déplacement amplitude (M-ASK - Amplitude Shift Keying)..... 24
I.4.2. Modulation d'amplitude sur deux porteuses en quadrature (MAQ) 26
I.4.3. Modulation par déplacement de fréquence (M-FSK - Frequency Shift Keying) 27
I.4.4. Modulation par déplacement de Phase (M-PSK - Phase Shift Keying)..... 29
I.5. Systèmes multistandard et problématique de classification de modulation 31
I.5.1. Radio logicielle (Software Defined Radio - SDR) 32
I.5.2. De la SDR à la Radio Cognitive..... 33
I.5.3. Radio cognitive 33
I.6. Méthodes classiques de classification (sans Deep Learning) 35
I.6.1. Les méthodes basées sur la vraisemblance (Likelihood-Based : LB)..... 36
I.6.2. Les méthodes basées sur les caractéristiques (Feature-Based : FB) 37
I.6.3. Caractéristiques spectrales 38

I.6.4. La transformée en ondelettes 39
I.6.5. Caractéristiques cyclo-stationnaires 40
I.6.6. Les statistiques d'ordre supérieur 41
I.7. Conclusion 42

II. CHAPITRE II : DEEP LEARNING ET CLASSIFICATION DES MODULATIONS 44
II.1. Introduction 45
II.2. Intelligence artificielle et réseaux de neurones artificiels 46
II.2.1. L'intelligence Artificielle (IA) 46
II.2.2. Réseaux neurones Artificiels 49
II.3. Deep Learning 56
II.3.1. Principe de fonctionnement du Deep Learning 56
II.3.2. Avantages du Deep Learning : 57
II.3.3. Contraintes d'utilisation du Deep Learning 58
II.4. Réseaux de neurones convolutifs 58
II.4.1. Les couches du CNN 60
II.4.2. Etapes de classification des images par réseaux CNN 64
II.5. Classification de modulation par Deep Learning 69
II.5.1. Limitation des approches traditionnelles de la classification automatique de modulation 69
II.5.2. Évolution vers les approches Deep Learning 70
II.5.3. Prétraitement et représentation des signaux 70
II.5.4. Architectures de réseaux de neurones pour la classification de modulation 72
II.6. Conclusion 75

III. CHAPITRE III : RESULTATS DE SIMULATION : PRESENTATION ET INTERPRETATION 76
III.1. Introduction : 77
III.2. Présentation de la base de données RML2016a 77
III.2.1. RML2016a en format image en utilisant le spectrogramme : 79
III.2.2. RML2016a en format image en utilisant la transformée de Wigner-Ville (WVD) 80
III.2.3. Classification par un modèle CNN proposé : 81
III.2.4. Classification par l'utilisation du "transfert learning" 93

III.3. Modèle de réseaux utilisant la représentation I.Q 97
III.4. Conclusion : 103

IV. CONCLUSION GENERALE 104

REFERENCES BIBLIOGRAPHIQUES 106

Liste des figures

Chapitre I

Figure I:1: Chaîne de transmission numérique. Les deux lignes de la chaîne correspondent respectivement à l'émetteur et au récepteur. 18

Figure I:2 : Principe de la modulation numérique. 21

Figure I:3: Représentation de la constellation d'une modulation. 22

Figure I:4: Constellations des modulations 2-ASK, 4-ASK et 8-ASK.....25

Figure I:5:Représentation temporelle du signal modulant et du signal modulé pour une modulation 4-ASK. 25

Figure I:6: Constellation de la modulation 8-QAM et 16 QAM. 26

Figure I:7: Représentation temporelle des deux signaux modulants et du signal modulé pour une modulation 8-QAM. 27

Figure I:8 : Représentation temporelle du signal modulant et du signal modulé pour la modulation BFSK.....28

Figure I:9 : Constellations des modulations BPSK, QPSK et 8 PSK. 30

Figure I:10 : Allure temporelle du signal modulant et du signal modulé pour la modulation 8-PSK. 30

Figure I:11 : Eléments de base de la radio logicielle. 32

Figure I:12 : Architecture simplifiée d'un système RC. 34

Figure I:13 : Etapes de classification des méthodes basées sur la vraisemblance (LB). 36

Figure I:14 : Etapes de classification des méthodes basées sur les caractéristiques (FB). ...38

Chapitre II

Figure II:1 : Schéma bloc de l'apprentissage supervisé d'un réseau de neurone.....47

Figure II:2 : Schéma bloc de l'apprentissage non supervisé d'un réseau de neurone.....48

Figure II:3 : Neurone formel de Mac Culloch et Pitts.[14] 50

Figure II:4 : Fonction d'activation ReLU. 51

Figure II:5 : Fonction d'activation sigmoïde. 52

Figure II:6 : Fonction d'activation tanh. 52

Figure II:7 : Fonction d'activation ELU. 53

Figure II:8 : Un résumé des types d'architecture de réseaux de neurones. 55

Figure II:9 : Schéma explicatif du principe de fonctionnement du Deep Learning. 57

Figure II:10 : Architecture de base d'un réseau CNN. 59

Figure II:11 : Illustration du principe de fonctionnement d'une couche de convolution à trois entrées et deux sorties. 61

Figure II:12 : Principe des deux types de sous échantillonnage (Pooling). 62

Figure II:13 : Principe du drop-out durant l'entraînement d'un modèle. 63

Figure II:14 : Couche entièrement connectée (FullyConnected layer). 64

Figure II:15 : Utilité de chacun des trois sous-ensembles d'une base de données : training, validation et test. 66

Chapitre III

Figure III:1: Organisation hiérarchique des images de spectrogramme de la base RML2016a 79

Figure III:2: Exemples de spectrogrammes des signaux de la base RML2016a 80

Figure III:3: Exemples des images de transformée WVD des signaux de la base RML2016a 81

Figure III:4: Evolution de l'entrainement du premier modèle proposé appliqué à des image WVD de la base RML2016a 84

Figure III:5: Matrices de confusion pour les différentes valeurs du SNR obtenu par le premier modèle proposé appliqué à des images WVD. 87

Figure III:6: Evolution de l'entrainement du premier modèle proposé appliqué à des images de spectrogramme de la base RML2016a. 89

Figure III:7: Matrices de confusion pour les différentes valeurs du SNR obtenu par le premier modèle proposé appliqué à des images de spectrogramme. 92

Figure III:8: Evolution de l'entrainement du réseau SqueezeNet appliqué à des image WVD de la base RML2016a. 95

Figure III:9 Evolution de l'entrainement du réseau proposé appliqué à des données en format IQ de la base RML2016a. 99

Figure III:10: Matrices de confusion pour les différentes valeurs du SNR obtenu par le premier réseau proposé appliqué à des données en format IQ de la base RML2016a.....102

Liste des tableaux

Chapitre I

Tableau I:1: Caractéristiques spectrales utilisées pour la classification de modulation. 38

Tableau I:2: Cumulants théorique des modulations M-PAM, M-PSK et M-QAM. 42

Chapitre II

Tableau II:1: Tableau comparatif des bases de données AMR. 75

Chapitre III

Tableau III:1: Architecture du premier réseau proposé pour la classification automatique de la modulation.....82

Tableau III:2: Les options d'apprentissage du réseau.....83

Tableau III:3: Justesse de test du premier modèle proposé appliqué aux images de spectrogramme pour les valeurs positifs du SNR 85

Tableau III:4: justesse de test du premier modèle proposé appliqué aux images de spectrogramme pour les valeurs positifs du SNR 89

Tableau III:5: caractéristiques les différentes couches du réseaux SqueezeNet. 93

Tableau III:6: Justesse de test du réseau SqueezeNet appliqué aux images WVD pour les valeurs positives du SNR 95

Tableau III:7: Comparaison entre le réseau proposé et le réseau SqueezeNet en termes de précision de classification par classe de modulation et pour chaque valeur de SNR.....96

Tableau III:8: Justesse d'apprentissage des réseaux essayés avec la Base RML2016.10a…97

Tableau III:9: Caractéristiques les différentes couches du réseau proposé pour les données IQ.....98

Tableau III:10: justesses de test du réseau proposé appliqué à des données en format IQ de la base RML2016a. 100

Liste des abréviations

AI : Artificial Intelligence
ALRT : Average Likelihood Ratio Test
ANN : Artificial Neural Network
ASK : Amplitude Shift Keying
AMC : Automatic Modulation Classification
BPSK : Binary Phase Shift Keying
CAN : Convertisseur Analogique Numérique
CNN : Convolutional Neural Network
CPFSK : Continuous Phase Frequency Shift Keying
CR : Cognitive Radio
CWT : Continuous Wavelet Transform
DL : Deep Learning
DNN : Deep Neural Network
FB : Feature-Based
FFT : Fast Fourier Transform
FSK : Frequency Shift Keying
GFSK : Gaussian Frequency Shift Keying
GLRT : Generalized Likelihood Ratio Test
GPU : Graphics Processing Unit
HLRT : Hybrid Likelihood Ratio Test
HOS : Higher Order Statistics
IA : Intelligence Artificielle
IQ : In-phase and Quadrature
LB : Likelihood-Based
LMS : Least Mean Square
MAC : Modulation Automatic Classification
ML : Machine Learning
MLE : Maximum Likelihood Estimation
M-PSK : M-ary Phase Shift Keying
M-QAM : M-ary Quadrature Amplitude Modulation
NN : Neural Network
PAM : Pulse Amplitude Modulation
PDF : Probability Density Function
PSK : Phase Shift Keying
QAM : Quadrature Amplitude Modulation
QPSK : Quadrature Phase Shift Keying
ReLU : Rectified Linear Unit
RF : Radio Frequency
RML : RadioML (Radio Machine Learning dataset)

RNN : Recurrent Neural Network
SCF : Spectral Correlation Function
SDR : Software Defined Radio
SNR : Signal-to-Noise Ratio
SVM : Support Vector Machine
TEB : Taux d'Erreur Binaire
TES : Taux d'Erreur Symbole
WBFM : Wide Band Frequency Modulation
WVD : Wigner-Ville Distribution
WT : Wavelet Transform

Liste des symboles

x_i : Entrée i-ème d'un neurone
w_i : Poids synaptique associé à l'entrée x_i
b : Biais ou seuil d'activation
S : Somme pondérée des entrées $S = \sum w_i x_i + b$
$\hat{y}$: Sortie du neurone
f(S) : Fonction d'activation
α : Paramètre de la fonction ELU
e : Constante d'Euler (base du logarithme naturel)
θ : Paramètres du modèle (poids et biais)
L : Fonction de perte
$\nabla_\theta L$: Gradient de la fonction de perte par rapport aux paramètres
D : Ensemble de données
K : Noyau ou filtre convolutif
* : Symbole de convolution
r : Récompense (reinforcement learning)
t : Temps ou index temporel
$\varphi(x)$: Représentation ou caractéristique extraite de l'entrée x

INTRODUCTION GENERALE

Dans le contexte actuel des télécommunications modernes, l'évolution rapide des technologies sans fil et la multiplication des standards de communication ont conduit à une complexification croissante des systèmes de transmission.

La classification automatique des modulations (AMC) est devenue un élément crucial dans les systèmes de communication modernes, particulièrement dans le contexte de la radio cognitive (CR) et de la radio logicielle (SDR), permettant l'identification automatique du type de modulation d'un signal reçu, une étape essentielle pour l'adaptation dynamique des paramètres de transmission et la gestion efficace des ressources spectrales.

Traditionnellement, deux approches principales ont été utilisées pour la classification des modulations : les méthodes basées sur la vraisemblance (LB) et celles basées sur les caractéristiques (FB), mais ces approches présentent des limitations significatives en présence de bruit ou d'interférences.

L'émergence du Deep Learning offre de nouvelles perspectives prometteuses pour surmonter ces limitations. Notre travail s'inscrit dans le cadre de la classification automatique des modulations, en s'appuyant sur des réseaux de neurones convolutifs **(CNN)**.

Notre mémoire s'articule en trois chapitres :

Le premier présente les fondements théoriques des modulations numériques et leurs méthodes de classification traditionnelles, établissant une base solide pour comprendre les enjeux et défis du domaine.

Le second explore l'application du Deep Learning à la classification des modulations, en détaillant les architectures neuronales et leurs principes de fonctionnement.

Le troisième chapitre présente les résultats de la classification des onze types de modulation issus de la base RML2016a, en utilisant à la fois un réseau CNN conçu spécifiquement et la technique de Transfert Learning basée sur le modèle SqueezeNet.Afin d'évaluer l'impact du format de représentation des données sur les performances de classification, la base RML2016a a été exploitée sous trois formes distinctes : les composantes en phase et en quadrature (I/Q), les images de spectrogrammes, et les images issues de la transformée de Wigner-Ville (WVD).

CHAPITRE I :

MODULATIONS NUMERIQUES ET LEURS CLASSIFICATIONS

I.1. **Introduction**

Un système de communication numérique assure la transmission fiable et efficace des informations malgré les perturbations du canal, tout en optimisant les ressources disponibles. Une chaine de communication numérique est le socle des systèmes de télécommunication modernes, reliant émetteurs et récepteurs à travers divers canaux de transmission [1]. Elle comprend plusieurs modules ou blocs, chacun jouant un rôle crucial dans la fiabilité et l'efficacité du transfert d'information [2].

La modulation numérique est considérée comme l'un des blocs les plus importants de la chaine de communication, les modulations numériques se déclinent en plusieurs formes selon les besoins et les contraintes du système [1][3]. Chaque type présente des compromis entre plusieurs paramètres et critères d'évaluation.

La souplesse de traitement et la reconfigurabilité sont des critères essentiels des systèmes de communication basés sur la radios logicielles (Software Defined Radio, SDR) et la radio cognitive (CR) [4][5]. En effet, ces systèmes peuvent, par simple modification logicielle, commuter entre différentes formes de modulations ou d'accès radio, ce qui les rend indispensables dans les environnements dynamiques. Cette capacité d'adaptation implique la détection et la classification automatique des signaux modulés pour identifier l'environnement spectral et choisir la stratégie de communication la plus appropriée [6].

Dans ce contexte, la classification automatique des modulations devient une brique essentielle pour la radio cognitive et les systèmes adaptatifs. Ainsi, la nécessité d'une classification fiable et robuste des modulations numériques est au cœur des défis des chaînes de communication modernes [1][6].

Ce chapitre se propose d'explorer ces concepts, en commençant par une revue détaillée de la chaîne de communication numérique, des modulations numériques et de leurs principales catégories, avant d'aborder l'importance stratégique de la classification automatique des modulations et un panorama des méthodes classiques utilisées à cet effet.

I.2. **Chaine de transmission numérique**

La chaîne de transmission numérique représente les différentes étapes de traitement de l'information. Elle relie l'émetteur au récepteur par l'intermédiaire d'un canal de transmission

(Channel). Le canal est le milieu dans lequel est transmis ou stocké l'information du message (comme un câble électrique ou une fibre optique) [1][2][7].

Ainsi, comme indiqué dans la fig.1, le message m, qu'il soit analogique (m(t)) ou déjà sous forme numérique (m[n]), est transformé via les différents blocs de l'émetteur pour finalement devenir un signal analogique x(t) qui est émis. Le récepteur reçoit quant à lui un signal analogique y(t) qui diffère de x(t) à cause des perturbations sur le canal de transmission. Les opérations inverses de l'émetteur permettent d'obtenir un message reçu $\hat{m}$, qu'on espère être exactement le message émis [1][2][7].

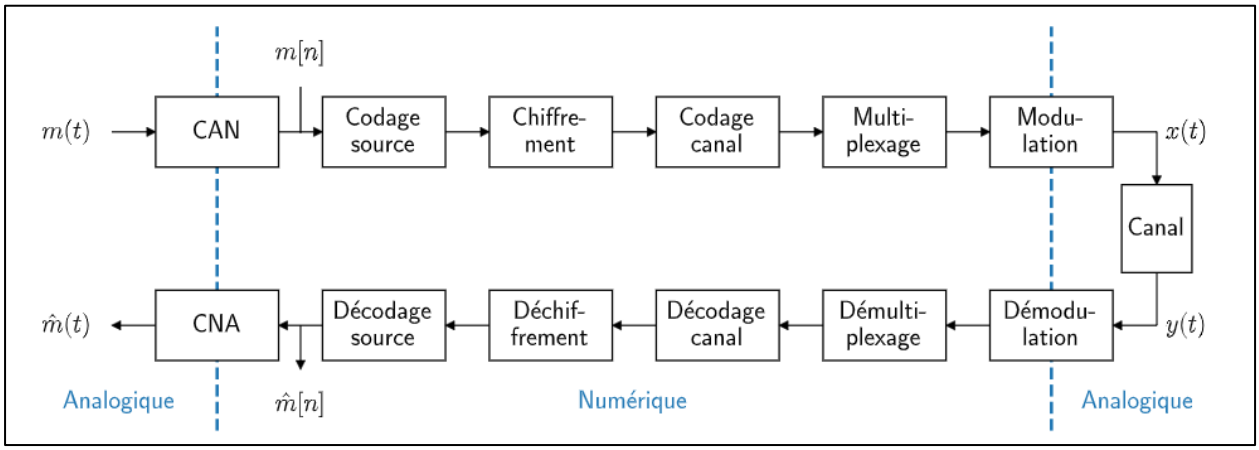


Figure I:1: Chaîne de transmission numérique. Les deux lignes de la chaîne correspondent respectivement à l'émetteur et au récepteur.

I.2.1. **Au niveau d'émetteur**

- **Le convertisseur analogique -numérique (CAN) :** transforme le message analogique en message numérique. Il effectue donc un échantillonnage et une quantification du signal portant le message analogique. Le CAN n'est bien sûr pas utile si le message est déjà sous forme numérique [1][2].
- **Le codage source** : effectue une mise en correspondance entre l'alphabet du message et celui du canal. Par exemple, un texte dont les symboles sont les lettres et les signes de ponctuation est transformé en message binaire dont les symboles sont 0 et 1. Le codage source peut également effectuer une compression des données, c'est-à-dire réduire le plus possible la taille du message qui sera effectivement transmis [1][2].

- **Le chiffrement (cryptage)** : consiste à modifier le message selon une certaine règle pour garantir son authentification ou pour le rendre incompréhensible si on ne possède pas la règle [2].
- **Le codage canal** : ajoute de la redondance dans le message, le message est donc rallongé. L'objectif est de protéger le message des erreurs de transmission. Les codes utilisés permettent de détecter la présence d'erreurs dans le message reçu, et parfois même de les corriger [1][2].
- **Le multiplexage** : regroupe plusieurs messages différents pour qu'ils puissent être transmis sur le même canal [1][2].
- **La modulation :** peut être définie comme le processus par lequel le signal est transformé de sa forme originale en une forme adaptée au canal de transmission, par exemple en faisant varier les paramètres (Amplitude, fréquence ou la phase) d'une onde sinusoïdale appelée porteuse [1][3]. Le dispositif qui effectue cette modulation, C'est le modulateur. L'opération inverse qui permet d'extraire le signal (original) de la porteuse est la démodulation.
- **Le canal :** représente le milieu physique ou électromagnétique par lequel le signal transite de l'émetteur vers le récepteur. Il peut s'agir d'un câble, de l'air (ondes radio), de la fibre optique …etc. Le canal peut altérer le signal à cause de divers phénomènes comme le bruit, l'atténuation, la distorsion ou les interférences, ce qui peut entraîner des erreurs de transmission [1][2].

I.2.2. **Au niveau du récepteur**

- **La démodulation :** permet d'extraire le signal numérique à partir du signal analogique modulé reçu. Elle consiste à éliminer la modulation (amplitude, fréquence, phase…) appliquée à l'émission afin de retrouver la suite de bits initiale transmise [1][3].
- **Le démultiplexage :** consiste à séparer les différents messages qui avaient été regroupés lors du multiplexage à l'émission. Chaque message est ainsi récupéré individuellement pour être traité [1][2].
- **Le décodage canal :** vise à détecter et corriger les erreurs introduites pendant la transmission. Il repose sur l'utilisation de codes de correction d'erreurs. Ces codes

permettent de repérer les erreurs éventuelles dans le message reçu, et parfois même de les corriger automatiquement [1][2].

- **Le déchiffrement (décryptage) :** consiste à transformer le message chiffré en son contenu lisible. Il utilise une clé de déchiffrement afin de restaurer le message original, garantissant la confidentialité et l'intégrité des données [2].
- **Le décodage source :** permet de restituer l'information originale à partir de sa version compressée ou codée. Il consiste à inverser le codage source effectué à l'émission afin de retrouver le message dans une forme exploitable (texte, image, son...) [1][2].
- **Le convertisseur numérique – analogique (CNA) :** transforme le message numérique obtenu en un signal analogique. Il reconstitue un signal continu à partir d'échantillons numériques pour une restitution finale exploitable (par exemple : un son audible ou une image visible) [1][2].

I.3. **La modulation numérique**

I.3.1. **Principe de la modulation numérique**

Afin de pouvoir transmettre sans fil des informations qui étaient à l'origine analogiques (comme la voix) mais qui ont été numérisées, il a fallu mettre au point des techniques spécifiques. Ces techniques reposent sur le principe des modulations numériques. La figure I.2 montre le principe de la modulation numérique. Tout comme pour les modulations analogiques, le système d'émission utilise une onde porteuse, de forme sinusoïdale et de fréquence fc, qui sert à transporter le message binaire à travers l'espace hertzien [1][3].

Les modulations numériques sont largement privilégiées grâce à une meilleure compatibilité avec les données numériques et une plus grande immunité contre les interférences.

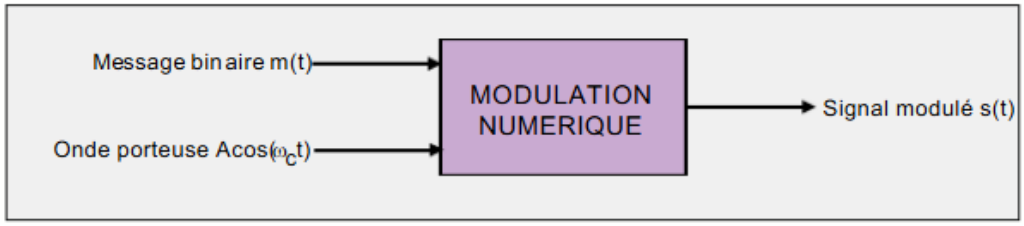


Figure I:2 : Principe de la modulation numérique.

La forme générale d'un signal modulant (le signal de l'information obtenu après l'étape de codage) est :

$$m(t) = \sum_k c_k g(t - kT) = \sum_k c_k(t) = \sum_k a_k(t) + j\sum_k b_k(t) = a(t) + jb(t) \quad \text{I.1}$$

Avec : g(t) est une forme d'impulsion dans l'intervalle [0, T[

C_k est un symbole complexe donnée par

$$c_k = a_k + jb_k \quad \text{I.2}$$

a_k et **b_k** prennent leurs valeurs dans l'alphabet **($A_1, A_2, \dots, A_M$)** et dans l'alphabet (**$B_1, B_2, \dots, B_M$**).

$a_k(t)$ et **$b_k(t)$** sont des impulsions de forme :

$$a_k(t) = a_k g(t - kT) \;\; et \;\; b_k(t) = b_k g(t - kT) \quad \text{I.3}$$

Pour une porteuse ayant la forme :

$$p(t) = \cos(2\pi f_c t) \quad \text{I.4}$$

La forme complexe du signal modulé en bande passante peut s'écrire :

$$s(t) = \Re\{\sum_k c_k(t) \cdot e^{j2\pi f_c t}\} \quad \text{I.5}$$

Où :

$$s(t) = \sum_k a_k(t)\cos(2\pi f_c t) - \sum_k b_k(t)\sin(2\pi f_c t) \quad \text{I.6}$$

$$s(t) = a(t)\cos(2\pi f_c t) - b(t)\sin(2\pi f_c t) \quad \text{I.7}$$

L'équation I.7 montre que a(t) module la porteuse en phase $\cos(2\pi f_c t)$ et b(t) module la porteuse en quadrature $\sin(2\pi f_c t)$ ce qui signifie que la modulation est bidimensionnelle

Les principaux types de modulation numérique incluent le déplacement d'amplitude (ASK), le déplacement de fréquence (FSK), le déplacement de phase (PSK) et le déplacement d'amplitude et de phase (QAM) [1][3].

I.3.2. **Caractéristiques et critères d'évaluation d'une modulation**

I.3.2.1. **Constellation**

La constellation est une représentation graphique dans le plan complexe (I/Q) des symboles possibles pour une modulation donnée [1][3]. C'est la représentation de tous les symboles $c_k = a_k + jb_k$ dans le plans complexe comme le montre la figure I.3. La constellation permet de visualiser la modulation et d'analyser la distance entre symbole pour avoir son impact sur les erreurs en fonction du bruit.

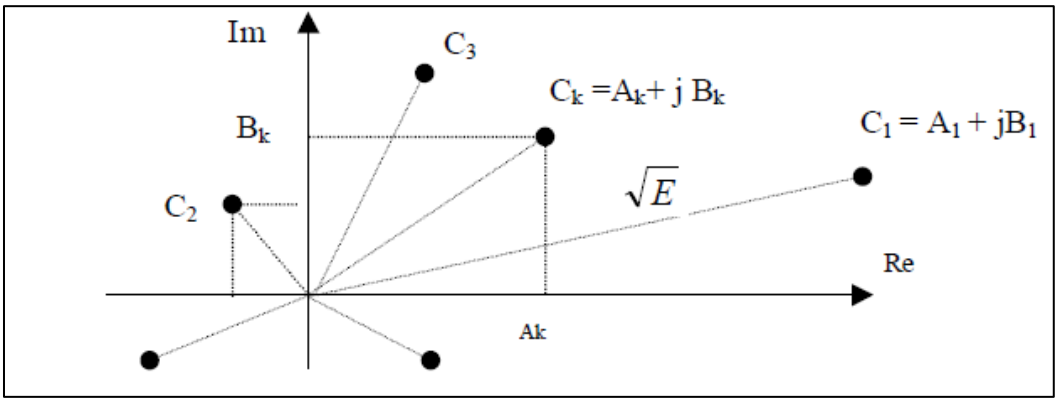


Figure I:3: Représentation de la constellation d'une modulation.

I.3.2.2. **Probabilité d'erreur**

La probabilité d'erreur par bit Pb est la probabilité de recevoir un bit erroné (transmission de 0 et réception de 1 ou transmission de 1 et réception de 0) [1][3].

La probabilité d'erreur par symbole Ps est la probabilité de recevoir un symbole erroné.

Pb et Ps sont des valeurs théoriques, dans la pratique et dans les simulations on utilise une estimation de Pb et Ps qui sont: le Taux d'Erreur par Bit TEB (Bit Error Rate : BER) et le taux d'erreur par symbole TES (Symbol Error rate SER) qui sont calculés par :

$$BER=\frac{nombre\ de\ bits\ erronés}{nombre\ total\ de\ bits} \qquad I.8$$

$$SER=\frac{nombre\ de\ symboles\ erronés}{nombre\ total\ de\ symboles} \quad \text{I.9}$$

La probabilité de l'erreur ou taux d'erreur sont liées à la distance minimale d_{min} entre les points de la constellation. Plus d_{min} est grande et plus la probabilité d'erreur sera faible.

I.3.2.3. **L'efficacité spectrale**

C'est le rapport entre le débit binaire ***D*** et la largeur de bande spectrale du signal modulé **B** :

$$\boldsymbol{\eta = \frac{D}{B}} \text{ En bit/sec/Hz} \quad \text{I.10}$$

Pour une modulation **M-aire** :

$$\boldsymbol{\eta = \frac{log_2(M)}{TB}} \text{ En bit/sec/Hz} \quad \text{I.11}$$

L'efficacité spectrale augmente, avec le nombre de bit/symbole ***n***.

I.3.2.4. **Energie d'émission moyenne**

L'énergie d'émission moyenne d'un signal modulé est une grandeur importante qui dépend directement des symboles de la constellation utilisée et de leur probabilité d'apparition [1][3]. Elle correspond à l'énergie moyenne transmise par symbole ou par unité de temps. Elle dépend de la géométrie de la constellation (position des points), de la probabilité d'apparition des symboles (souvent équiprobables) et et du type de modulation (PSK, QAM, etc.).

L'énergie d'émission moyenne est donnée par :

$$E_{avg} = \mathbb{E}[|s|^2] = \sum_{i=1}^{M} P(s_i) \cdot |s_i|^2 \quad \text{I.12}$$

Avec : M : nombre de symboles dans la constellation, $s_i \in \mathbb{C}$: symbole complexe, $|s_i|^2$: énergie instantanée du symbole s_i et $P(s_i)$: probabilité d'occurrence de s_i

Si tous les symboles sont équiprobables l'expression précédente devient :

$$E_{avg} = \frac{1}{M}\sum_{i=1}^{M} |\, s_i|^2 \quad \text{I.13}$$

I.4. **Types de base de la modulation numérique**

I.4.1. **Modulation par déplacement amplitude (M-ASK - Amplitude Shift Keying)**

La modulation par déplacement d'amplitude (ASK), où l'information est codée en faisant varier l'amplitude d'un signal porteur sinusoïdal [1][3]. La fréquence et la phase de la porteuse restent constantes. Dans sa forme la plus simple, appelée modulation On-Off Keying (OOK) ou 2-ASK, la présence d'une amplitude représente un bit '1' et son absence (ou une amplitude très faible) représente un bit '0'. Des variantes M-ASK utilisent M niveaux d'amplitude distincts.

Le signal modulé peut s'écrire comme une somme de symboles pondérant une fonction de base modulée en fréquence :

$$s(t) = \sum_n a_n \cdot g(t - nT_s) \cdot \cos(2\pi f_c t) \qquad \text{I.14}$$

Où :

- $a_n \in \mathcal{A} \subset \mathbb{R}$: symboles de la constellation ASK (ex : $\{0, A\}$ pour 2-ASK, $\{-3A, -A, A, 3A\}$ pour 4-ASK)
- $g(t)$: forme d'impulsion de base (ex : fonction porte rectangulaire, racine de cosinus surélevé, etc.)
- T_s : durée d'un symbole
- f_c : fréquence de la porteuse
- $\cos(2\pi f_c t)$: porteuse sinusoïdale

Pour une modulation M-ASK symétrique, les M symboles sont uniformément répartis sur une droite comme le montre la figure I.4

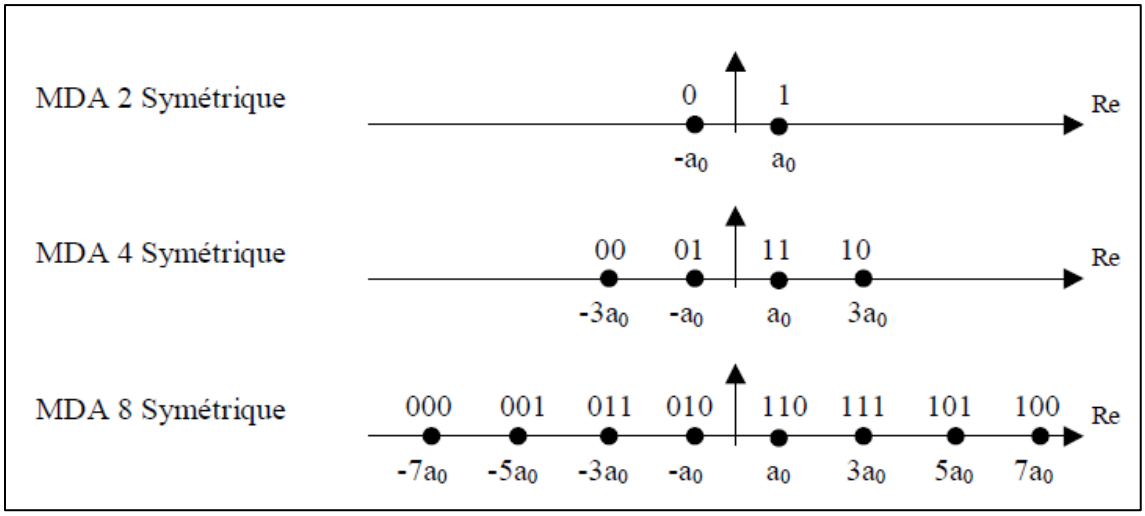


Figure I:4: Constellations des modulations 2-ASK, 4-ASK et 8-ASK.

La figure I.5 donne la représentation temporelle du signal modulant et la représentation du

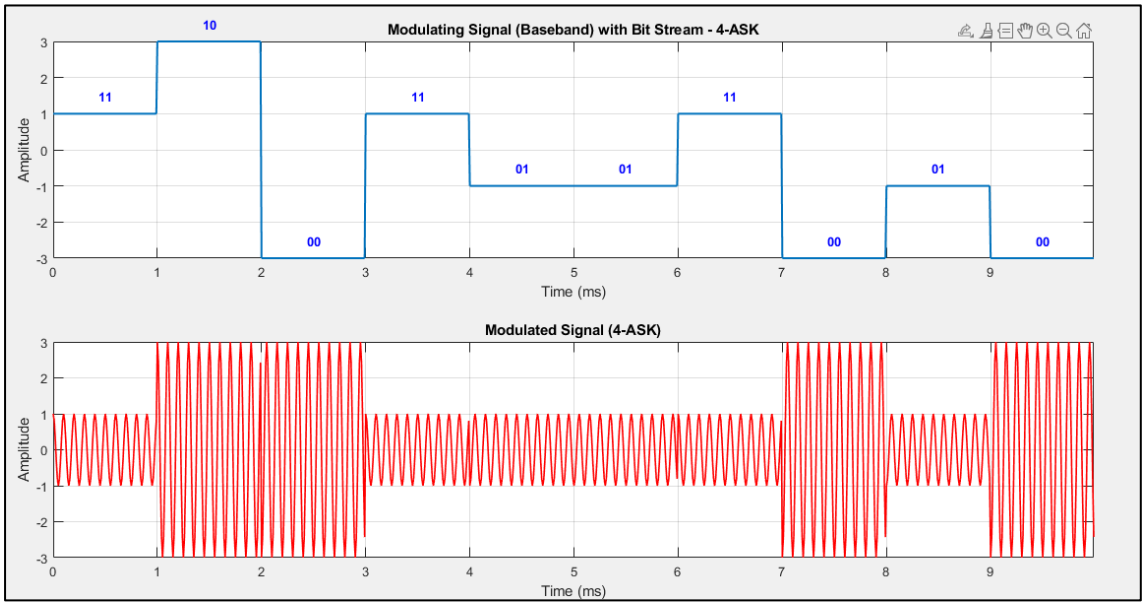


Figure I:5:Représentation temporelle du signal modulant et du signal modulé pour une modulation 4-ASK.

signal modulé pour une séquence binaire : 11 10 00 11 01 01 11 00 01 00

La valeur approximative de la probabilité de l'erreur binaire est donnée par :

$$p_b \approx \frac{2(M-1)}{M \log_2 M} \cdot Q\left(\sqrt{\frac{6 \log_2 M \cdot E_b}{(M^2-1)N_0}}\right) \quad \text{I.15}$$

Avec : $Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^{\infty} e^{-t^2/2}\, dt$: fonction Q de Gauss, E_b : énergie par bit, N_0 : densité spectrale de puissance du bruit (en W/Hz) et M : nombre de symboles.

I.4.2. **Modulation d'amplitude sur deux porteuses en quadrature (MAQ)**

La modulation MAQ (Quadrature Amplitude modulation QAM) est une modulation bidimensionnelle qui permet d'utiliser efficacement l'énergie émise lorsque le nombre de points *M* est grand [1][3]. Toute en maximisant la distance minimale entre symboles par répartition des points de la constellation uniformément dans le plan.

Le signal modulé *s(t)* est la somme de deux porteuses en quadrature, modulées en amplitude par les deux signaux *a(t)* et *b(t)* comme le montre l'équation I.7.

La valence **M** de la modulation peut être exprimé par : $\boldsymbol{M = M_1.M_2}$

Avec $M_1 = 2^{n_a}$ et $M_2 = 2^{n_b}$, $\boldsymbol{n_a}$ est le nombre de bits par symbole $\boldsymbol{a_k}$ et $\boldsymbol{n_b}$ le nombre de bits par symboles $\boldsymbol{b_k}$

Les éléments $\boldsymbol{a_k}$ sont : $\mp a_0, \mp 3a_0, \ \mp 5a_0, \ \ldots \mp (M_1 - 1)a_0$ et les éléments $\boldsymbol{b_k}$ sont : $\mp a_0, \mp 3a_0, \ \mp 5a_0, \ \ldots \mp (M_2 - 1)a_0$

Pour une modulation M-QAM carrée $M_1 = M_2$ et (n_a=n_b) donc $M = {M_1}^2 = {M_2}^2$ et $n = 2n_a$=$2n_b$ la constellation prend la forme d'un carré et M peut être exprimé sous forme $\boldsymbol{M = 4^i}$ **(**Exemples MAQ-4, MAQ-16, MAQ-64)

La figure I.6 montre deux exemples de la constellation 8 QAM et 16 QAM.

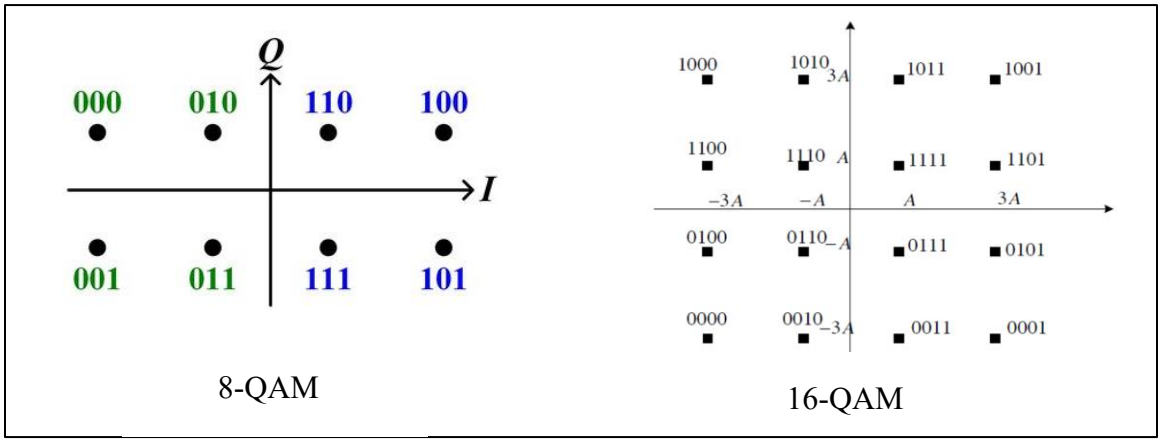


Figure I:6: Constellation de la modulation 8-QAM et 16 QAM.

Pour la modulation QAM, la valeur approximative de la probabilité d'erreur par bit pour un canal AWGN est exprimée par :

$$TEB \approx \frac{4\left(1-\frac{1}{\sqrt{M}}\right)}{\log_2 M} \cdot Q\left(\sqrt{\frac{3\log_2 M \cdot \frac{E_b}{N_0}}{M-1}}\right) \quad I.16$$

La figure I.7 illustre l'allure temporelle des deux signaux modulants a(t) (composante en phase) et b(t) (composante en quadrature) et du signal modulé s(t) pour une modulation 8-QAM et pour n train binaire de 120 bits.

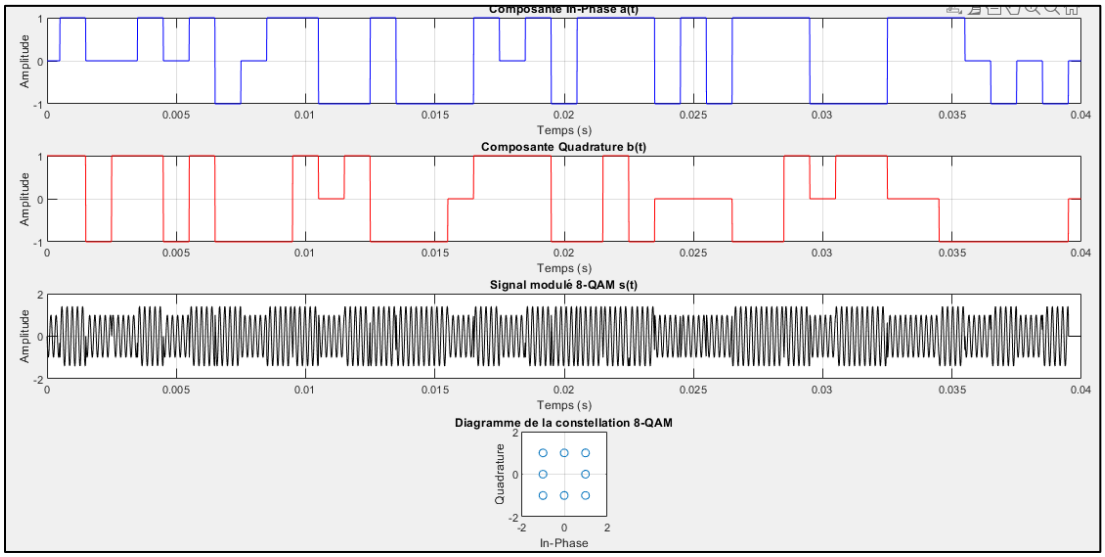


Figure I:7: Représentation temporelle des deux signaux modulants et du signal modulé pour une modulation 8-QAM.

I.4.3. **Modulation par déplacement de fréquence (M-FSK - Frequency Shift Keying)**

Pour la modulation par déplacement de fréquence (FSK) l'information est codée en faisant varier la fréquence d'un signal porteur entre plusieurs valeurs discrètes prédéterminées [1][3]. L'amplitude et la phase de la porteuse restent constantes. Dans sa forme la plus simple, la FSK binaire (BFSK ou 2-FSK), deux fréquences distinctes sont utilisées pour représenter les bits '0' et '1'. Des variantes M-FSK utilisent M fréquences distinctes.

Pour la modulation FSK, le signal modulé s(t) a une enveloppe constante, C'est la dérivée de la phase qui est reliée de façon linéaire à la valeur des symboles représentant l'information.

Le signal FSK s'écrit :

$$s(t) = A\cos(2\pi f_c t + \phi(t)) \quad I.17$$

La fréquence instantanée est donnée par :

$$f_i(t) = f_c + \frac{1}{2\pi}\frac{d\phi(t)}{dt} \quad \text{I.18}$$

La première partie de l'équation I.18 représente la fréquence centrale du signal modulé (fréquence de la porteuse) et la deuxième partie représente la déviation de fréquence qui est proportionnelle aux symboles information.

Sachant que la déviation de fréquence est proportionnelle aux symboles d'information, la fréquence peut s'exprimer comme suivant :

$$f_i(t) = f_c + \frac{\Delta f}{2} a_k \quad \text{I.19}$$

Où : Δf est la différence de la *fréquence instantanée* correspondant à l'émission de deux symboles adjacents.

Et a_k est un symbole de l'information appartenant à l'ensemble { ±1, ±3, … ±(M-1) }.

La figure I.8 donne la représentation temporelle du signal modulant et celle du signal modulé pour une modulation BFSK (M=2). La figure montre que le signal modulé possède deux fréquences différentes correspondant aux deux valeurs du symbole (0 et 1).

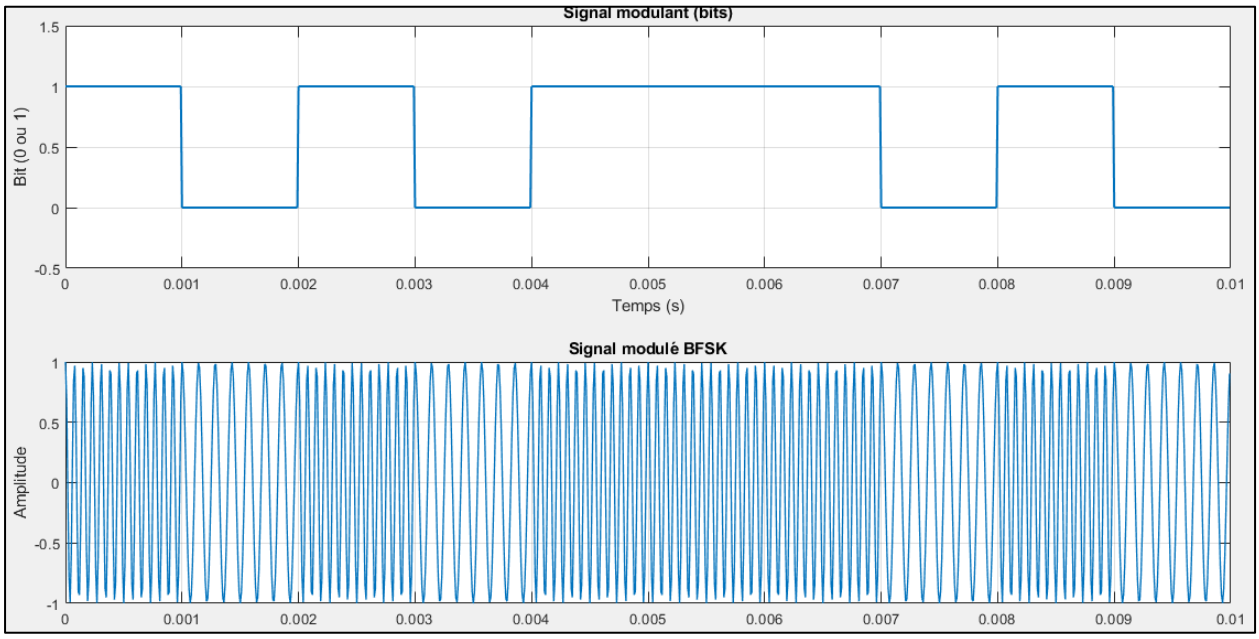


Figure I:8 : Représentation temporelle du signal modulant et du signal modulé pour la modulation BFSK.

La limité supérieure de la probabilité d'erreur par bit est donnée par l'expression :

$$P_b \leq \frac{M-1}{2} \cdot Q\left(\sqrt{\log_2 M \cdot \frac{E_b}{N_0}}\right) \quad \text{I.20}$$

I.4.4. **Modulation par déplacement de Phase (M-PSK - Phase Shift Keying)**

Pour la modulation par déplacement de phase (PSK) l'information est codée en modifiant la phase d'un signal porteur sinusoïdal [1][3]. La phase de la porteuse prend un nombre fini de valeurs discrètes, chaque valeur représentant un symbole (un ou plusieurs bits). L'amplitude et la fréquence de la porteuse restent constantes.

Pour la modulation PSK les points de la constellation (c_k) sont répartis sur un cercle, donc :

$$c_k = a_k + jb_k = e^{j\varphi_k} \quad \text{I.21}$$

Ce qui donne :

$$a_k = \cos(\varphi_k)\ et\ b_k = \sin(\varphi_k) \quad \text{I.22}$$

Pour améliorer la probabilité de l'erreur, les symboles sont répartis régulièrement sur le cercle selon l'expression :

$$\varphi_k = \frac{\pi}{M} + k\frac{2\pi}{M}\ pour\ k = 0,1, \dots M-1 \text{ Pour M>2 ou } \varphi_k = 0\ ou\ \pi \text{ pour M=2} \quad \text{I.23}$$

La figure I.9 illustre des exemples de la constellation pour les modulations BPSK (M=2), QPSK (M=4) et 8-PSK

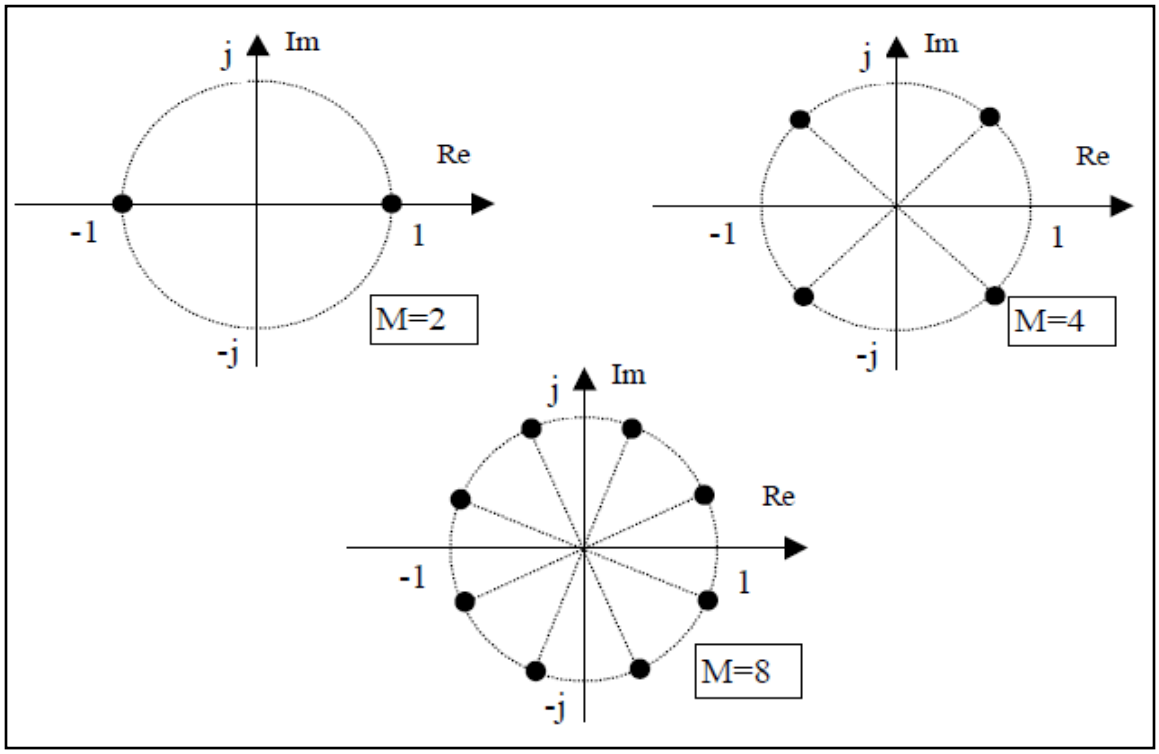


Figure I:9 : Constellations des modulations BPSK, QPSK et 8 PSK.

Le signal modulé présente des sauts de phase qui sont proportionnelles aux passages entre les symboles consécutifs comme le montre la figure I.10.

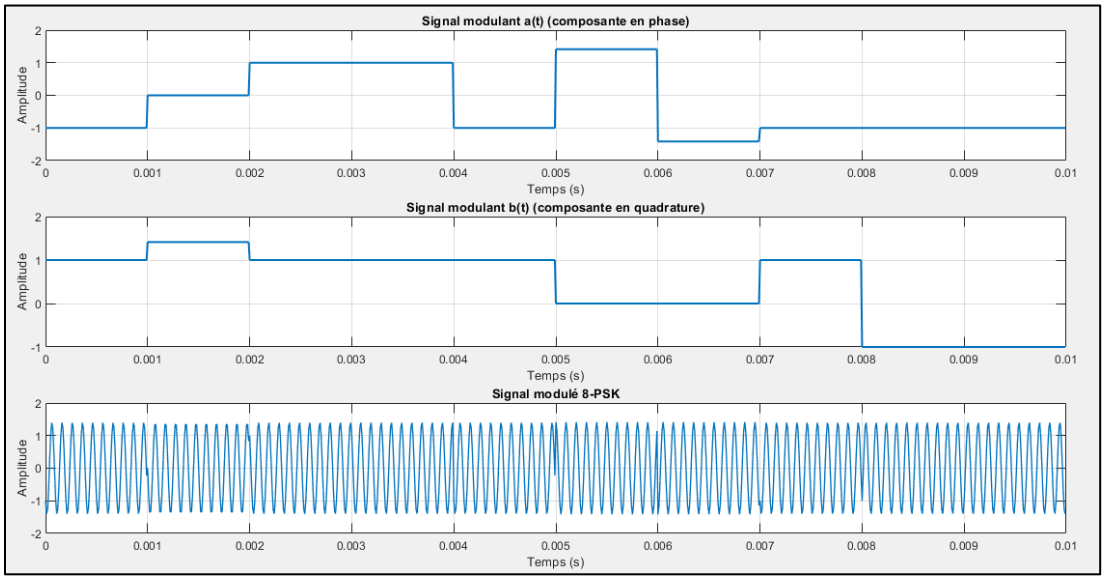


Figure I:10 : Allure temporelle du signal modulant et du signal modulé pour la modulation 8-PSK.

La valeur approximative de la probabilité de l'erreur par bit pour la modulation M-PSK est donnée par :

$$\text{TEB} \approx \frac{2}{\log_2 M} \cdot Q\left(\sqrt{2\log_2 M \cdot \sin^2\left(\frac{\pi}{M}\right) \cdot \frac{E_b}{N_0}}\right) \qquad \text{I.24}$$

I.5. **Systèmes multistandard et problématique de classification de modulation**

Un système multistandard peut être défini comme un système de communication capable de s'adapter à différentes normes de transmission, en modifiant ses paramètres de fonctionnement tels que la fréquence, la bande passante, le schéma de modulation, le codage, et les protocoles de communication.

L'évolution rapide des technologies de communication a conduit à une prolifération de normes et standards : - Réseaux cellulaires (2G/GSM, 3G/UMTS, 4G/LTE, 5G) – Réseaux locaux sans fil (Wi-Fi 802.11a/b/g/n/ac/ax) - Communications à courte portée (Bluetooth, Zigbee, NFC) - Réseaux d'objets connectés (LoRa, SigFox, NB-IoT) - Communications par satellite - Radiodiffusion (DAB, DVB) [1][7].

Cette diversité crée le besoin de dispositifs capables de s'adapter à différents environnements et applications.

Dans un contexte multi-standards, la classification automatique des modulations est essentielle pour permettre aux récepteurs intelligents (comme les radios logicielles ou cognitives) de s'adapter dynamiquement à des environnements hétérogènes [6][7]. Elle permet l'identification du type de modulation utilisé sans connaissance préalable du signal transmis, ce qui est crucial lorsque plusieurs standards coexistent (Wi-Fi, LTE, Bluetooth, etc.).

Cela améliore l'efficacité spectrale en facilitant la détection et l'exploitation opportuniste des bandes disponibles. Elle est aussi indispensable pour les récepteurs universels, capables de communiquer avec divers protocoles sans intervention humaine.

En outre, cette classification permet des applications en surveillance du spectre, détection d'intrusion, et guerre électronique, où il est nécessaire d'identifier rapidement des signaux inconnus. Elle constitue donc un pilier fondamental de la radio cognitive et des systèmes de communication adaptatifs.

I.5.1. **Radio logicielle (Software Defined Radio - SDR)**

La SDR (Software Defined Radio) est un système de communication radio qui peut s'adapter à n'importe quelle bande de fréquence et recevoir n'importe quelle modulation en utilisant le même matériel. Les opportunités qu'offre la SDR lui permettent de résoudre des problèmes de la gestion dynamique du spectre [4][5]. Les équipements SDR peuvent fonctionner dans des réseaux sans fil hétérogènes. Les systèmes de communication à base de la SDR peuvent émettre et recevoir des signaux à différentes fréquences pour implémenter des normes de télécommunications allant de la radio FM à la 5G, au LTE et au WLAN. C'est-à-dire qu'une SDR idéale peut s'adapter automatiquement aux nouvelles fréquences et aux nouvelles modulations.

Comme le montre la figure I.11, une radio logicielle se compose généralement d'un dispositif frontal RF configurable et d'une carte FPGA ou d'un SoC (System-on-Chip) programmable permettant d'exécuter des fonctions numériques.

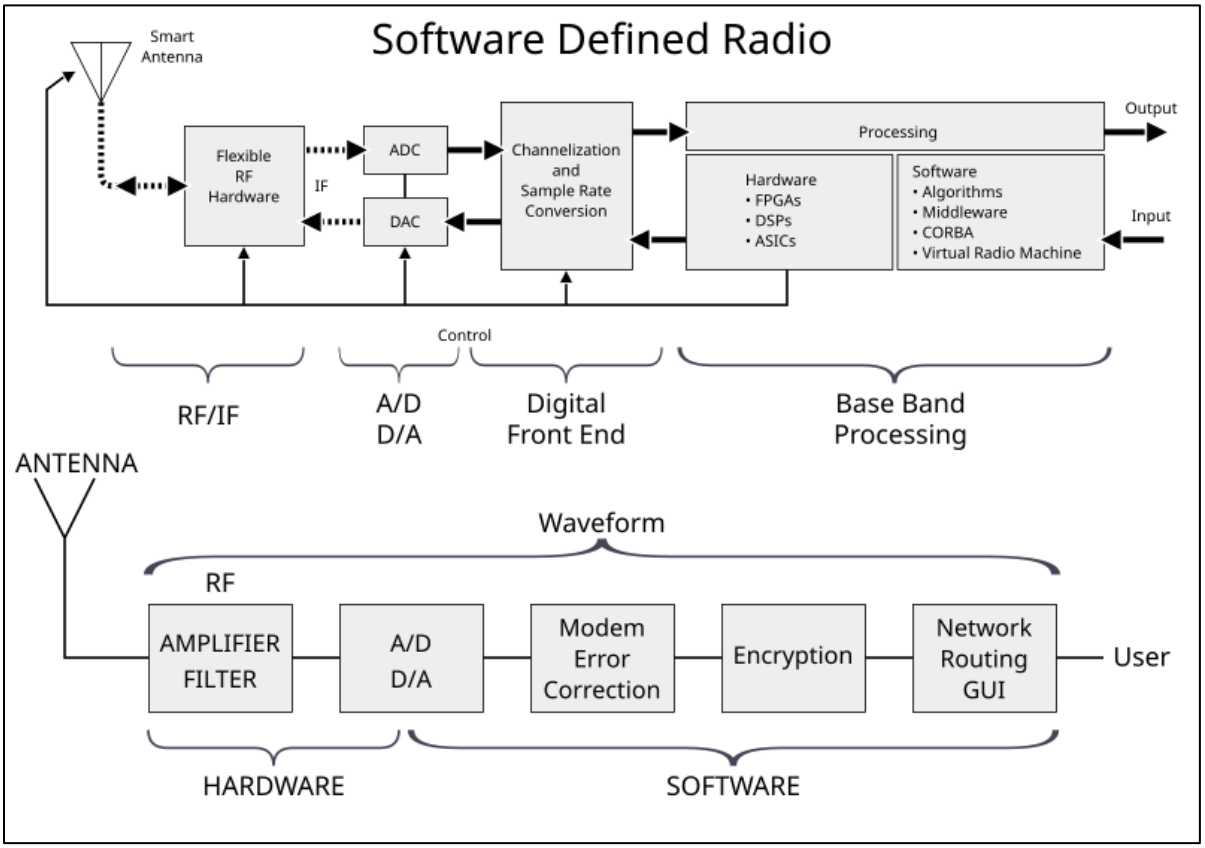


Figure I:11 : Eléments de base de la radio logicielle.

I.5.2. **De la SDR à la Radio Cognitive**

La radio cognitive (CR) est née en réponse aux limites de la radio logicielle (SDR), confrontée à des défis croissants dans un environnement spectral de plus en plus encombré et complexe [4][5][9]. Alors que la SDR permet une reconfiguration logicielle des paramètres radio (fréquence, modulation, puissance), elle reste passive et incapable de s'adapter intelligemment aux conditions dynamiques du spectre. Ce manque d'autonomie a engendré plusieurs problèmes majeurs : un gaspillage des ressources spectrales (bandes inutilisées mais réservées), des interférences croissantes entre systèmes hétérogènes (réseaux civils/militaires/urgence), et une incapacité à optimiser les performances dans des environnements variables.

Pour pallier ces insuffisances, la radio cognitive a introduit une couche d'intelligence artificielle couplée à la SDR, permettant une perception active de l'environnement RF et une prise de décision autonome [4][5][9]. Grâce à des techniques comme le spectrum sensing, l'apprentissage automatique (optimisation des paramètres en temps réel) et le partage dynamique du spectre (accès opportuniste sans interférence), la RC transforme les radios en systèmes auto-adaptatifs. Ces capacités sont devenues indispensables pour des applications critiques comme les réseaux militaires (évitement de brouillage), la 5G (efficacité spectrale), ou les communications d'urgence (priorisation du trafic). Ainsi, la radio cognitive représente l'évolution naturelle de la SDR, combinant flexibilité matérielle et intelligence logicielle pour relever les défis des communications futures.

I.5.3. **Radio cognitive**

L'ITU-R définie la radio cognitive comme suivant :

Une radio cognitive est un système radio employant une technologie permettant au système :

– D'obtenir la connaissance de son environnement opérationnel et géographique, des politiques pouvant être établies, de son état interne,

– D'ajuster dynamiquement et de manière autonome ses paramètres opérationnels et ses protocoles en se fondant sur la connaissance acquise dans le but d'atteindre des objectifs prédéfinis,

– D'apprendre de l'expérience acquise [9].

L'architecture simplifiée d'un système RC est basée sur six éléments comme le montre la figure I.12.

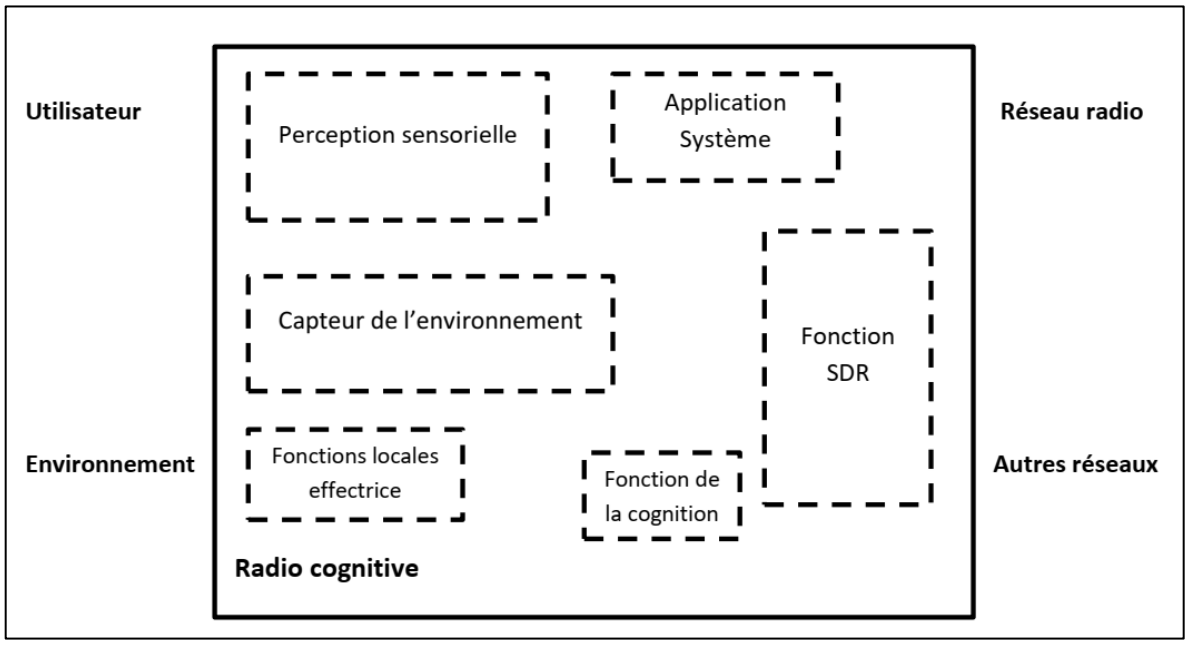


Figure I:12 : Architecture simplifiée d'un système RC.

Les six composantes fonctionnelles de l'architecture d'une radio cognitive (RC) sont :

1. Perception sensorielle de l'utilisateur : Cette composante permet à la radio cognitive d'interagir avec l'utilisateur via des interfaces multimodales. Elle inclut des capteurs haptiques (toucher), acoustiques (voix, sons) et vidéo pour capter les intentions, émotions ou préférences de l'utilisateur. Elle collecte des informations sensorielles pour adapter l'expérience utilisateur en temps réel. Grâce à ces données, le système comprend mieux le contexte d'usage. Cela permet une personnalisation fine des services et une adaptation dynamique aux besoins perçus de l'utilisateur.

2. Capteurs de l'environnement local : Ils permettent au système de percevoir son environnement physique immédiat. Ces capteurs incluent la localisation GPS, la température, l'accéléromètre, la luminosité, etc. Les données collectées aident à adapter les paramètres de transmission (comme la puissance ou la bande de fréquence) aux conditions réelles. Par exemple, un changement de position ou d'orientation peut déclencher une réévaluation du canal radio. Cette perception environnementale est cruciale pour assurer une communication fiable et efficiente.

3. Applications du système : Cette composante regroupe les services que la RC doit supporter : navigation GPS, streaming multimédia, messagerie, etc. Elle fournit des indications sur

les exigences en matière de qualité de service (QoS) comme la latence, le débit, ou la fiabilité. En comprenant la nature de l'application en cours, la RC peut adapter dynamiquement ses ressources. Par exemple, une vidéo HD nécessitera plus de bande passante qu'un message texte. Cette composante guide les décisions d'optimisation du système radio.

4. Fonctions SDR (Software Defined Radio) : Elles assurent la flexibilité du système radio par la reconfiguration logicielle des paramètres physiques : fréquence, modulation, puissance, etc. Ces fonctions permettent la détection de signaux RF, le changement de bande, la sélection de canaux, ou encore l'identification du type de modulation. Grâce à la SDR, la radio peut s'adapter en temps réel aux variations du spectre et aux standards multiples. Elle constitue la couche matérielle programmable indispensable à la mise en œuvre de la cognition.

5. Fonctions de la cognition : Cette unité est le « cerveau » du système. Elle prend en charge les fonctions de contrôle, de planification et d'apprentissage. Elle utilise les données de l'environnement, de l'utilisateur et des applications pour prendre des décisions intelligentes : par exemple, quel canal utiliser, à quelle puissance émettre, ou quand basculer vers un autre protocole. Elle peut intégrer des algorithmes d'intelligence artificielle ou de machine learning. Ce moteur décisionnel rend la radio réellement cognitive et adaptative.

6. Fonctions locales effectrices : Elles traduisent les décisions du système en actions visibles ou audibles pour l'utilisateur. Cela inclut la synthèse vocale (text-to-speech), l'affichage graphique, les alertes sonores ou les messages multimédias. Elles permettent une interaction intuitive et efficace avec l'utilisateur final. Par exemple, une RC peut afficher une carte ou une instruction vocale selon le contexte détecté. Ces fonctions garantissent une communication bidirectionnelle fluide entre la machine et l'être humain.

I.6. **Méthodes classiques de classification (sans Deep Learning)**

Les méthodes classiques de classification de modulation sont celles qui n'utilisent pas le deep learning. Elles se basent sur des techniques traditionnelles d'analyse du signal, comme l'extraction de caractéristiques (features) suivie de l'utilisation d'algorithmes de classification classiques (K-NN, SVM, arbres de décision, etc.) [6][9][10], contrairement aux méthodes modernes qui apprennent directement à partir des données brutes via des réseaux de neurones. Elles sont traditionnellement été regroupées en deux grandes catégories, les méthodes basées sur la vraisemblance LB (Likelihood-Based), et celles basées sur les caractéristiques FB (Features-

Based). Les classifieurs basés sur la vraisemblance formulent le processus de reconnaissance de la modulation sous la forme de problèmes de test d'hypothèses multiples, dans lesquels chaque schéma de modulation est représenté par une hypothèse. L'intérêt pour les classifieurs LB est motivé par l'optimalité de sa précision de classification lorsque le modèle et les paramètres du canal sont parfaitement connus des classifieurs. Les méthodes basées sur les caractéristiques forment le groupe le plus large des algorithmes de classification de modulation. Ces groupes d'algorithmes utilisent des caractéristiques du signal telles les statistiques de d'ordre supérieur (moments, cumulants), caractéristiques cyclo-stationnaires, Transformée en ondelettes (WT), caractéristiques spectrales, etc... Ces caractéristiques permettent de distinguer les différents types de modulation. Bien que les approches LB puissent aboutir à la solution optimale, les approches FB sont généralement préférées en raison de leur faible complexité et de leurs performances satisfaisantes.

I.6.1. **Les méthodes basées sur la vraisemblance (Likelihood-Based : LB)**

Les a méthodes basées sur la vraisemblance LB comprend deux étapes comme le montre la figure I.13. Dans la première étape, la vraisemblance est évaluée pour chaque hypothèse de modulation avec des échantillons de signal observés. Dans la deuxième étape, les probabilités des différentes hypothèses de modulation sont comparées pour conclure la décision de classification [6][10].

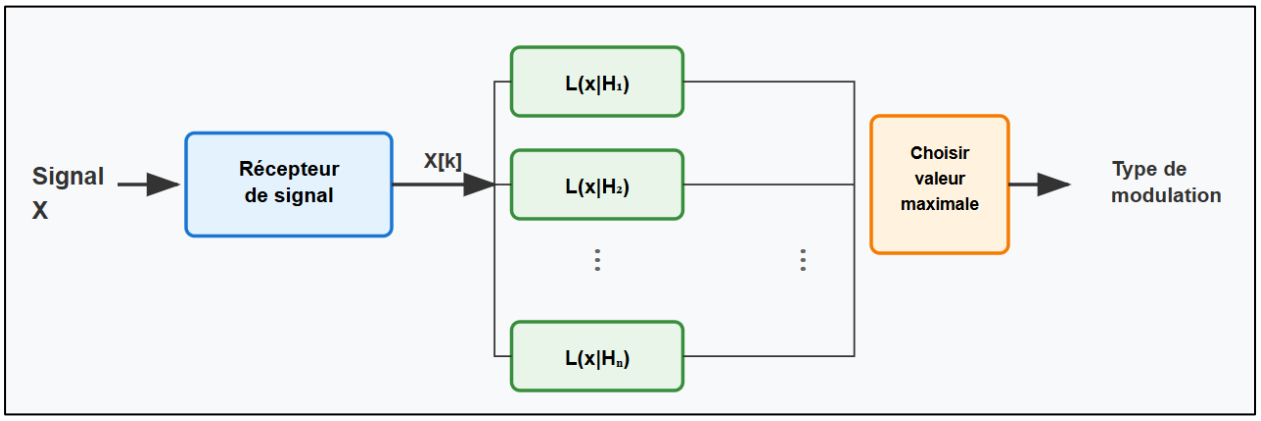


Figure I:13 : Etapes de classification des méthodes basées sur la vraisemblance (LB).

En fonction de la manière dont les paramètres et les données inconnus sont traités dans le calcul de la fonction de vraisemblance, les algorithmes basés sur MLE (Maximum Likelihood Estimate) peuvent être divisées en trois catégories principales :

- Tests de ratio de vraisemblance moyenne (ALRT)
- Tests de ratio de vraisemblance généralisée (GLRT)
- Tests de ratio de vraisemblance hybride (HLRT)

Dans ALRT, les quantités inconnues du signal telles que la constellation de signal et la puissance de bruit sont traitées comme des variables aléatoires. La fonction de densité de probabilité (PDF) du signal reçu est calculée en faisant la moyenne de ces variables aléatoires en supposant que leur distribution soit connue. Cette méthode conduit à la meilleure précision de classification si cette hypothèse est exacte.

Pour la GLRT, la PDF du signal reçu est calculée en utilisant l'estimation de vraisemblance maximale (MLE) des valeurs inconnues, où elles sont comme des variables déterministes. Tandis que ALRT nécessite des calculs lourds, en particulier lorsque le nombre de variables aléatoires augmente, GLRT est moins complexe, mais présente un inconvénient lorsqu'elle est utilisée pour classer des constellations de signaux imbriquées telles que (16-QAM et 64-QAM), ce qui entraîne une classification incorrecte.

HLRT traite les paramètres inconnus en tant que variables aléatoires ou déterministes, et traite les données en tant que variables aléatoires afin d'éliminer le problème de constellation de signaux imbriqués du GLRT. Avec les estimations ML des paramètres, HLRT est également complexe à mettre en œuvre.

En général, l'approche LB est sensible aux inadéquations des modèles, tel que le décalage temporel et les performances d'identification peuvent se dégrader de manière significative en raison de telles imperfections.

I.6.2. **Les méthodes basées sur les caractéristiques (Feature-Based : FB)**

Un algorithme FB bien conçu peut offrir les mêmes performances que l'algorithme LB, tout en ayant une complexité de calcul moindre. Dans les méthodes basées sur les caractéristiques (FB), le type de modulation est identifié à l'aide d'un processus en deux étapes. Premièrement, des caractéristiques représentatives sont utilisées pour représenter le signal reçu au lieu de traiter le signal comme un flux de symboles. Ensuite, les caractéristiques sélectionnées sont utilisées par un classifieur d'apprentissage automatique afin de prendre une décision concernant la classe de modulation [6][9][10]. Le processus général de la méthode FB est illustré dans la Figure I.14.

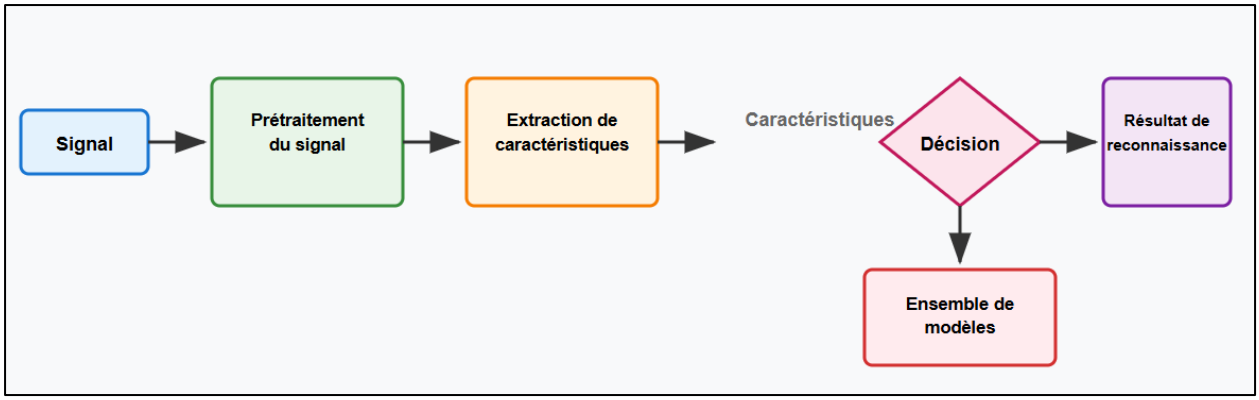


Figure I:14 : Etapes de classification des méthodes basées sur les caractéristiques (FB).

Les méthodes d'extraction de caractéristiques sont plus simples à mettre en œuvre et peuvent conduire à des solutions sous-optimales par rapport à l'approche basée sur la vraisemblance. L'efficacité de cette approche dépend de la puissance de classification des caractéristiques sélectionnées et de la simplicité des classifieurs appliqués. Nous consacrerons cette section pour énumérer certaines des caractéristiques bien connues de la classification de modulation.

I.6.3. **Caractéristiques spectrales**

Ces caractéristiques exploitent les caractères spectraux uniques de différentes modulations de signal sous trois aspects essentiels : l'amplitude, la phase et la fréquence. Dans la plupart des références ont proposé neuf caractéristiques pour la reconnaissance de modulations analogiques et numériques classiques. Ces caractéristiques sont dérivées de la densité spectrale de puissance, de l'amplitude instantanée, de la fréquence instantanée et de la phase du signal [6][9][10]. Ces caractéristiques sont résumées dans le tableau suivant :

Tableau I:1: Caractéristiques spectrales utilisées pour la classification de modulation.

Caractéristiques	Description
γ_{max}	La valeur maximale de la densité spectrale de puissance de l'amplitude instantanée normalisée et centré du signal reçu.
σ_{ap}	L'écart type de la valeur absolue de la composante non linéaire de la phase instantanée.

σ_{dp}	L'écart type de la composante non linéaire de la phase instantanée directe.
P	La symétrie du spectre autour de la fréquence porteuse.
σ_{ap}	L'écart type de la valeur absolue de l'amplitude instantanée normalisée et centrée des échantillons de signal.
σ_{af}	L'écart type de la valeur absolue de la fréquence instantanée normalisée et centrée.
σ_a	L'écart type de l'amplitude instantanée normalisée et centrée.
μ^a_{42}	Le kurtosis de l'amplitude instantanée normalisée et centrée.
μ^f_{42}	Le kurtosis de la fréquence instantanée normalisée et centrée.

I.6.4. **La transformée en ondelettes**

La transformation en ondelettes continue du signal reçu r(t) est définie par :

$$CWT(a,\tau) = \int_{-\infty}^{+\infty} r(t)\Psi^*_{a,\tau}dt \qquad \text{I.25}$$

Parmi les différentes fonctions d'ondelettes mères, à savoir Morlet, Haar et Shannon, la fonction d'ondelettes de Haar est privilégiée en raison de sa forme simple et de la commodité de son calcul. L'ondelette de Haar est donnée par :

$$\Psi(t) = \begin{cases} 1, & \text{si } 0 \le t < \frac{T}{2} \\ -1, & \text{si } \frac{T}{2} \le t < T \\ 0, & \text{sinon} \end{cases} \qquad \text{I.26}$$

Avec :

$$\Psi_{a,\tau}(t) = \frac{1}{\sqrt{a}}\Psi\left(\frac{t-\tau}{a}\right) \qquad \text{I.27}$$

La technique de transformation en ondelettes (WT) du signal peut analyser efficacement les formes d'onde avec des discontinuités et des pics nets. En outre, le WT peut être calculé à l'aide d'algorithmes rapides et efficaces, un fait important pour la mise en œuvre d'un algorithme en temps réel. Cependant, peu de formes d'onde réelles présentent des discontinuités aussi marquées,

à l'exception de la période des symboles et des temps de changement de phase. L'estimation de ces points de synchronisation rend la technique difficile à mettre en œuvre. De plus, la sélection du type de WT approprié pour la discrimination de caractéristiques peut être un défi en fonction des formes d'onde des modulations utilisées. Cette technique n'est pas largement représentée dans la littérature et les modulations couvertes semblent très limitées, un exemple de discrimine uniquement les formes d'onde M-PSK et M-FSK.

I.6.5. **Caractéristiques cyclo-stationnaires**

Les techniques qui s'appuient sur les caractéristiques cyclo-stationnaires d'un signal peuvent être appliquées à la reconnaissance de nombreuses modulations et à des niveaux faibles de SNR avec un bruit considérable. Elles tirent parti du fait que de nombreuses formes d'onde peuvent être modélisées comme étant cyclo-stationnaires plutôt que stationnaires en raison des périodicités sous-jacentes des signaux. Pour de tels processus, leurs moyennes et leurs autocorrélations sont périodiques [6][9][10].

Étant donné un signal sinusoïdal x(t), on dit que le signal présente des propriétés cyclo-stationnaires ou contient une périodicité d'ordre 2 si l'autocorrélation cyclique présentée dans l'équation

$$R_x^{\alpha}(\tau) = \lim_{T\to\infty} \frac{1}{T} \int_{-T/2}^{T/2} x\left(t+\frac{\tau}{2}\right)\left(t-\frac{\tau}{2}\right) e^{-i2\pi f \alpha t} dt \qquad \text{I.28}$$

Existe et n'est pas identique à zéro en fonction de τ, avec la fréquence $\alpha \neq 0$.

Le spectre cyclique, qui peut être interprété comme une fonction de corrélation spectrale (SCF) est donné par la transformée de Fourier de l'autocorrélation cyclique

$$S_x^{\alpha}(f) = \int_{-\infty}^{+\infty} R_x^{\alpha}(\tau) e^{-i2\pi f \tau} d\tau \qquad \text{I.29}$$

Il est démontré que le plan SCF théorique de différents signaux de modulation sur un domaine de α et f présente des différences distinctives et peut être utilisé pour la classification des modulations de signal. Cependant, l'utilisation de la SCF est limitée à certaines modulations d'ordre inférieur, elles ne convient pas aux modulations d'ordre supérieur. Pour cette raison, l'utilisation des cumulants cycliques pour la classification des modulations d'ordre élevé a été proposée.

I.6.6. **Les statistiques d'ordre supérieur**

Les Statistiques d'Ordre Supérieur à 2 (High Order Statistics), autrement dit les moments et cumulants d'ordre supérieur à 2, sont utilisées essentiellement en complément aux statistiques d'ordre 2. Elles donnent une description plus complète des données et de leurs propriétés. Le moment d'un signal est égal à l'espérance du signal élevée à une puissance déterminée par l'ordre du moment. La moyenne d'un signal est le moment du premier ordre, tandis que le moment du deuxième ordre indique généralement la puissance du signal [6][9][10]. Pour un processus aléatoire stationnaire à valeurs complexes, y, les moments d'ordre supérieur (HOM) sont définis comme suit :

$$M_{pq}(y) = E[y^p (y^*)^q] \qquad \text{I.30}$$

Les moments peuvent être estimés comme suit :

$$\widehat{M}_{pq}(y) = \frac{1}{N}\sum_{i=1}^{N} y^p(i) y^{*q}(i) \qquad \text{I.31}$$

Où N est le nombre d'échantillons du signal y.

Les cumulants d'ordre p du signal y de moyenne nulle sont donnés par :

$$C_{pq}(y) = [\underbrace{y, \dots, y}_{(p-q)\text{ fois}}, \underbrace{y^*, \dots, y^*}_{(q)\text{ fois}}] \qquad \text{I.32}$$

En règle générale, les HOCs (High Order Cumulant) sont exprimés en fonction des moments d'ordre supérieur (HOMs) d'un signal. Les relations entre certains des HOC les plus couramment utilisés et les HOM sont données par les équations suivantes :

$$\begin{aligned}
C_{20} &= M_{20} \\
C_{21} &= M_{21} \\
C_{40} &= M_{40} - 3M_{20}^2 \\
C_{41} &= M_{40} - 3M_{20}M_{21} \\
C_{42} &= M_{42} - |M_{20}|^2 - 2M_{21}^2 \\
C_{60} &= M_{60} - 15M_{20}M_{40} + 30M_{20}^3 \\
C_{61} &= M_{61} - 5M_{21}M_{40} - 10M_{20}M_{41} + 30M_{20}^2 M_{21} \\
C_{62} &= M_{62} - 6M_{20}M_{42} - 8M_{21}M_{41} - M_{22}M_{40} + 6M_{20}^2 M_{22} + 24M_{21}^2 M_{20} \\
C_{63} &= M_{63} - 9M_{21}M_{42} + 12M_{21}^3 - 3M_{20}M_{43} - 3M_{22}M_{41} + 18M_{20}M_{21}M_{22}
\end{aligned} \qquad \text{I.33}$$

Les valeurs des cumulants pour chaque type de modulation est unique et peut donc être utilisée comme caractéristique de la classification de modulation. Les valeurs théoriques du cumulant pour certains des types de modulation sont présentées dans le tableau suivant :

Tableau I:2: Cumulants théorique des modulations M-PAM, M-PSK et M-QAM.

Modulation	C_{20}	C_{21}	C_{40}	C_{41}	C_{42}
2-PAM	1.0000	1.0000	- 2.0000	- 2.0000	- 2.0000
4-PAM	1.0000	1.0000	- 1.3600	- 1.3600	- 1.3600
8-PAM	1.0000	1.0000	- 1.2381	- 1.2381	- 1.2381
BPSK	1.0000	1.0000	- 2.0000	- 2.0000	- 2.0000
QPSK	0.0000	1.0000	0.0000	0.0000	- 1.0000
8-PSK	0.0000	1.0000	0.0000	0.0000	- 1.0000
4-QAM	0.0000	1.0000	- 0.6800	0.0000	- 0.6800
16-QAM	0.0000	1.0000	- 0.6800	0.0000	- 0.6800
64-QAM	0.0000	1.0000	- 0.6191	0.0000	- 0.6191

I.7. **Conclusion**

Dans ce chapitre, nous avons commencé par les principaux éléments d'une chaîne de communication numérique, en expliquant comment l'information est traitée, modulée, transmise puis démodulée à travers un canal. Chaque bloc joue un rôle essentiel pour assurer une transmission efficace et fiable dans des environnements souvent perturbés.

Nous avons ensuite abordé les modulations numériques, qui consistent à adapter le signal porteur en fonction des données à transmettre. Parmi les techniques étudiées figurent la modulation M-ASK, M-QAM, M-PSK et M-FSK. Pour chaque type nous avons expliqué le principe de la

modulation, nous avons donné l'expression de la probabilité de l'erreur un exemple de la représentation temporelle des composantes du signal modulant et la représentation temporelle du signal modulé.

Les systèmes de communication multistandard imposent une adaptabilité croissante. La radio logicielle (SDR) et la radio cognitive (CR) illustrent cette évolution vers des systèmes capables de changer dynamiquement leur configuration en fonction de leur environnement ou du standard en cours d'utilisation.

La problématique de la classification automatique des modulations devient alors centrale, notamment dans les SDR et CR, où il est nécessaire d'identifier le type de modulation d'un signal inconnu sans information préalable, pour pouvoir le traiter correctement.

Enfin, nous avons présenté les méthodes classiques de classification, en distinguant celles basées sur la vraisemblance, qui comparent le signal reçu à des modèles statistiques, et celles basées sur des caractéristiques, qui extraient des paramètres représentatifs du signal pour les classer via des algorithmes de classification.

CHAPITRE II :
DEEP LEARNING ET CLASSIFICATION DES MODULATIONS

II.1. **Introduction**

L'intelligence artificielle et plus particulièrement l'apprentissage profond (Deep Learning) ont révolutionné de nombreux domaines d'application, notamment celui des télécommunications. La classification automatique des modulations numériques, traditionnellement basée sur des méthodes statistiques et des approches conventionnelles, bénéficie aujourd'hui des avancées significatives du Deep Learning.

Ce chapitre présente les fondements théoriques de l'apprentissage profond et son application dans le contexte de la classification des modulations. Nous nous concentrerons sur les architectures de réseaux de neurones profonds, leurs principes de fonctionnement et leur capacité à extraire automatiquement les caractéristiques pertinentes des signaux modulés.

Nous commencerons par introduire les concepts fondamentaux du Deep Learning, en explorant les différentes couches qui composent un réseau de neurones profond, les mécanismes d'apprentissage et les techniques d'optimisation. Une attention particulière sera portée aux réseaux de neurones convolutifs (CNN), qui ont démontré leur efficacité dans le traitement des signaux et la reconnaissance de motifs.

Nous aborderons également les aspects pratiques de l'implémentation, notamment le choix des hyperparamètres, les stratégies d'entraînement et les méthodes d'évaluation des performances. Cette base théorique nous permettra de comprendre comment le Deep Learning peut être exploité efficacement pour la tâche complexe de classification automatique des modulations numériques.

II.2. **Intelligence artificielle et réseaux de neurones artificiels**

II.2.1. **L'intelligence Artificielle (IA)**

II.2.1.1. **Concept de base**

L'intelligence artificielle (IA) est une technologie mise au point pour simuler l'intelligence humaine à travers des algorithmes. Pour se rapprocher le plus possible du fonctionnement du cerveau humain, **l'IA** fait appel à un certain nombre d'éléments dont les plus importants sont :

- Des algorithmes informatiques.
- De grandes bases de données.
- Des systèmes et matériels informatiques performants.

L'utilisation de ces trois éléments permet aux systèmes d'intelligence artificielle de raisonner, apprendre, s'améliorer de manière itérative en analysant et en intégrant les informations qui leur sont fournies, et même de prendre des décisions.

Le concept d'intelligence artificielle a vu le jour dans les années 1950, lorsque des chercheurs comme Alan Turing, John McCarthy et Marvin Lee Minsky ont posé les premières bases de ce qui allait devenir un domaine révolutionnaire [11][12] Le célèbre test de Turing, imaginé en 1950, cherchait à déterminer si une machine pouvait penser comme un humain. Quelques années plus tard, John McCarthy a inventé le terme « Artificial Intelligence » lors d'une conférence qui marquera un tournant dans l'histoire de l'informatique. Depuis ces premières expérimentations, l'IA a très bien évolué, passant des premiers systèmes experts aux technologies actuelles comme le Deep Learning et les réseaux neuronaux. Ces avancées permettent aujourd'hui à des plateformes d'intégrer des solutions innovantes pour optimiser le travail d'équipe. Près de 75 ans après, l'IA fait aujourd'hui partie intégrante de notre quotidien et elle ne cesse de se développer.

II.2.1.2. **Modèles d'entrainement de L'IA**

L'entraînement d'un modèle d'intelligence artificielle (IA) est le processus par lequel un modèle apprend à effectuer une tâche en analysant un grand ensemble de données. Durant cette phase, le modèle ajuste ses paramètres internes (comme les poids dans un réseau de neurones) pour minimiser l'erreur entre ses prédictions et les résultats attendus.

On distingue trois types de modèles d'entrainement :

L'apprentissage supervisé est un modèle de Machine Learning qui mappe une entrée spécifique à une sortie à l'aide de données d'entraînement étiquetées (données structurées). En d'autres termes, pour entraîner l'algorithme à reconnaître des images de chats par exemple, il faut lui fournir des images étiquetées comme étant des chats.

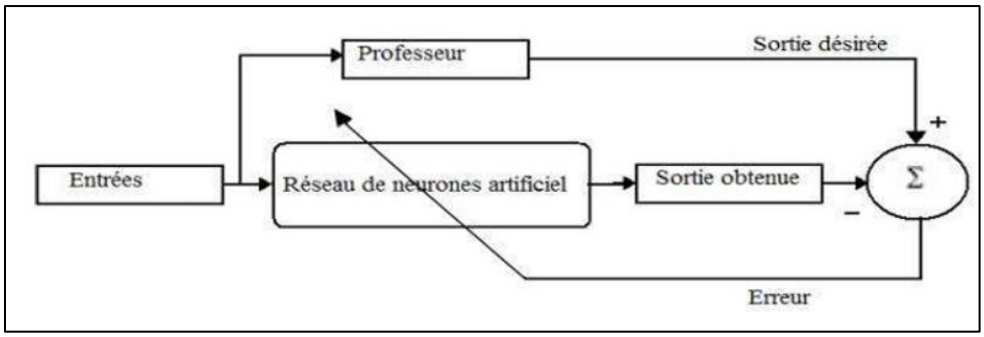


Figure II:1 : Schéma bloc de l'apprentissage supervisé d'un réseau de neurone.

L'apprentissage non supervisé est un modèle de Machine Learning qui apprend les tendances à partir de données sans étiquette (données non structurées). Contrairement à l'apprentissage supervisé, le résultat final n'est pas connu à l'avance. En effet, l'algorithme apprend à partir des données et les classe par groupes en fonction d'attributs. Par exemple, l'apprentissage non supervisé est efficace pour établir des correspondances de tendances et la modélisation descriptive.

En plus de l'apprentissage supervisé et non supervisé, une approche mixte appelée apprentissage partiellement supervisé est souvent utilisée. Lors de celle-ci, seules certaines données sont étiquetées. Dans l'apprentissage partiellement supervisé, le résultat final est connu, mais l'algorithme doit trouver comment organiser et structurer les données afin d'obtenir les résultats souhaités.

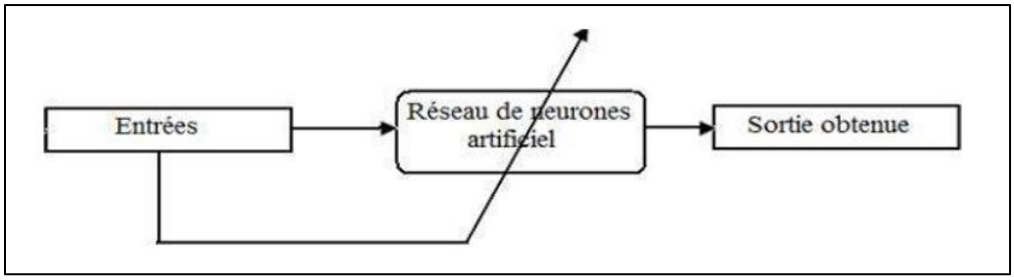


Figure II:2 : Schéma bloc de l'apprentissage non supervisé d'un réseau de neurone.

L'apprentissage par renforcement est un modèle de Machine Learning qui peut être décrit globalement comme "l'apprentissage par la pratique". Un "agent" apprend à exécuter une tâche définie de manière empirique (boucle de rétroaction) jusqu'à ce que ses performances se situent dans une plage souhaitée. L'agent reçoit un renforcement positif lorsqu'il effectue bien la tâche, et un renforcement négatif lorsqu'il se trompe. Un exemple d'apprentissage par renforcement consiste à apprendre à un robot à ramasser une balle.

II.2.1.3. **Domaines de L'IA**

L'IA englobe plusieurs domaines spécialisés qui constituent ses piliers fondamentaux :

Le calcul formel permet la manipulation symbolique et la résolution d'équations mathématiques complexes par des systèmes informatiques.

La représentation des connaissances s'attache à modéliser et structurer l'information de manière à ce qu'elle puisse être utilisée efficacement par les systèmes intelligents.

Le raisonnement et la résolution de problèmes concernent la capacité des systèmes à analyser des situations, à tirer des conclusions logiques et à élaborer des solutions adaptées.

L'apprentissage automatique (Machine Learning) constitue un domaine central où les systèmes améliorent leurs performances à partir de données reçues et des résultats obtenues sans être explicitement programmés pour chaque tâche.

Les réseaux neuronaux artificiels sont des modèles primitifs du cerveau humain. Une cellule neuronale possède une seule sortie et des entrées reliées à d'autres neurones. Ces réseaux utilisent plusieurs couches pour traiter les informations, imitant la structure et le fonctionnement du cerveau humain. Cela exige une programmation non explicite et de la répartition d'information sur l'ensemble du réseau.

II.2.2. **Réseaux neurones Artificiels**

Les réseaux de neurones artificiels (RNA) sont des modèles inspirés du fonctionnement du cerveau humain, capables d'apprendre à partir de données. Ils sont constitués de couches de neurones interconnectés qui traitent l'information de manière hiérarchique. Grâce à des mécanismes d'apprentissage comme la rétropropagation, les RNA ajustent leurs connexions pour minimiser l'erreur de prédiction. Ils sont largement utilisés dans des domaines variés tels que la reconnaissance d'images, le traitement du langage ou la classification de signaux. Leur capacité à modéliser des relations complexes en fait un outil central de l'intelligence artificielle moderne.

II.2.2.1. **L'évolution des réseaux neurones Artificiels**

L'histoire des réseaux de neurones artificiels remonte à 1943, lorsque Warren McCulloch et Walter Pitts ont conçu le premier modèle de neurone inspiré de la biologie. En 1949, Donald Hebb a approfondi cette idée en démontrant que les connexions neuronales se renforcent par l'usage, jetant ainsi les bases de l'apprentissage automatique. En 1957, Frank Rosenblatt a introduit le perceptron, un réseau à une seule couche aux capacités limitées, car incapable de traiter des données non linéairement séparables. Ce n'est que dans les années 1980, avec l'avènement d'ordinateurs plus puissants et l'invention de la rétropropagation, que les réseaux multicouches (MLP) ont pris leur essor. Puis, en 1998, Yann LeCun a révolutionné le domaine avec LeNet, un réseau convolutif (CNN) capable de reconnaître les chiffres manuscrits [13]. Depuis, l'apprentissage profond s'est imposé grâce à des architectures toujours plus performantes, intégrant des couches automatiques d'extraction de caractéristiques et éliminant le besoin d'une étape manuelle préalable. Ces avancées ont ouvert la voie aux systèmes modernes d'IA, capables de traiter des problèmes de plus en plus complexes.

II.2.2.2. **Fonctionnement du neurone artificiel**

Le neurone artificiel ou neurone formel, cherche à reproduire le fonctionnement d'un neurone biologique. Il existe différents niveaux d'abstraction, suivant la précision de la modélisation voulue.

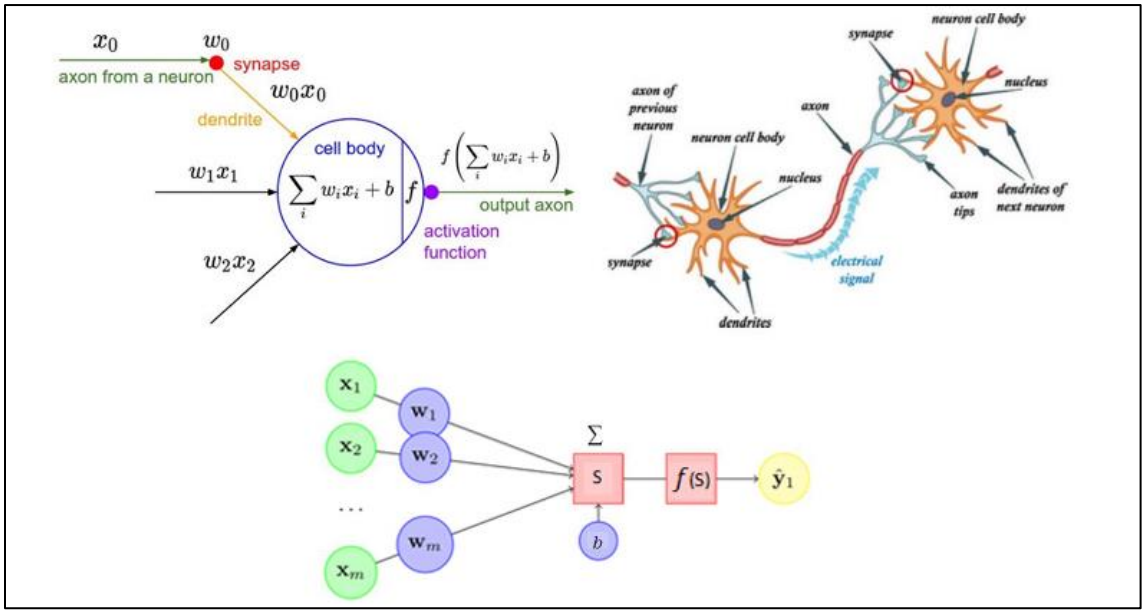


Figure II:3 : Neurone formel de Mac Culloch et Pitts.[14]

Comme le montre la figure II.3, chaque neurone est caractérisé par une fonction de combinaison et une fonction d'activation.

Tels que :

- **(X_1, X_2, …, X_m)** : sont les entrées du neurone (signaux qui lui parviennent).
- **(W_1, W_2, ..., W_n) :** les poids associés à chaque connexion.
- **b :** le seuil d'activation (biais).
- S est la somme pondérée des entrées (potentiel d'activation).
- **$\hat{Y} = f(S)$:** la sortie du neurone

La fonction de combinaison réalise une combinaison linaire de l'ensembles des signaux d'entrée x_i en utilisant un vecteur de poids (Wi) selon l'équation suivante :

$$S = \sum_{i=1}^{n} WiXi + b \quad \text{II.1}$$

II.2.2.3. **Fonction d'activation**

Le terme de fonction d'activation est synonyme de potentiel d'activation pour les neurones biologiques. La fonction d'activation applique une opération non-linaire sur la sortie de la fonction de combinaison ce qui est similaire au fonctionnement des neurones biologiques.

Il existe un grand nombre de fonctions d'activation, Voici quelques fonctions d'activation courantes :

La fonction Rectified Linear Unit (ReLU) : Principalement appliquée dans les réseaux de neurones convolutifs après les couches de convolutions. Elle attribut toutes les valeurs négatives à 0 et garde les valeurs positives inchangées. Elle est utilisée afin d'augmenter la non-linéarité du réseau. Elle est décrite par l'équation suivante :

$$f(s) = max(0, s) = \begin{cases} s \; si \; s \geq 0 \\ 0 \; sinon \end{cases} \quad \text{II.2}$$

Et dont l'allure est la suivante :

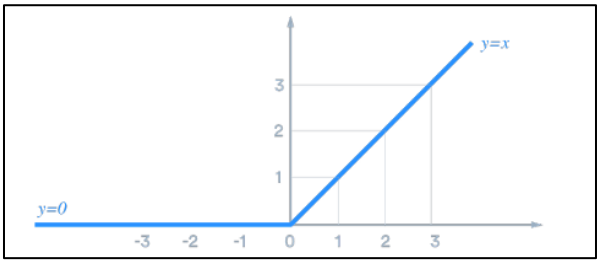


Figure II:4 : Fonction d'activation ReLU.

La fonction Sigmoïde : Elle représente la fonction de répartition de la loi logistique. Elle est souvent utilisée dans les réseaux de neurones parce qu'elle est dérivable. La forme de la dérivée de sa fonction inverse est extrêmement simple et facile à calculer, ce qui améliore les performances des algorithmes. Elle est décrite par la fonction suivante :

$$f(s) = \frac{1}{1+e^{-s}} \forall s \in \mathbb{R} \quad \text{II.3}$$

Et dont l'allure est la suivante :

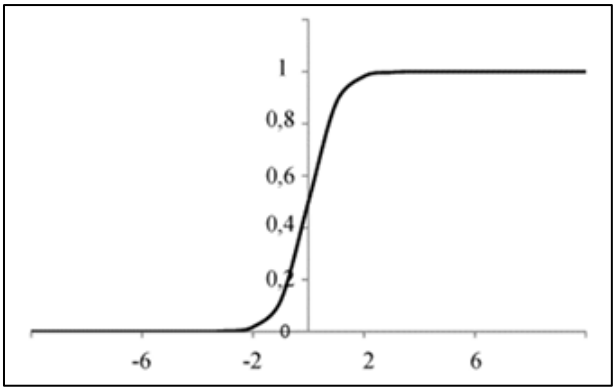


Figure II:5 : Fonction d'activation sigmoïde.

La fonction tangente hyperbolique (tanh) : La fonction tanh donne un résultat entre-1 et 1

L'avantage de la fonction tanh est que les entrées négatives seront bien répertoriées comme négatives. Elle a de meilleurs résultats que la fonction sigmoïde et la fonction ReLU dans un certain nombre de problèmes. Elle est définie par l'équation suivante :

$$\tanh(s) = \frac{\sinh(x)}{\cosh(x)} = \frac{e^{s} - e^{-s}}{e^{s} + e^{-s}} \quad \text{II.4}$$

Et voici son allure :

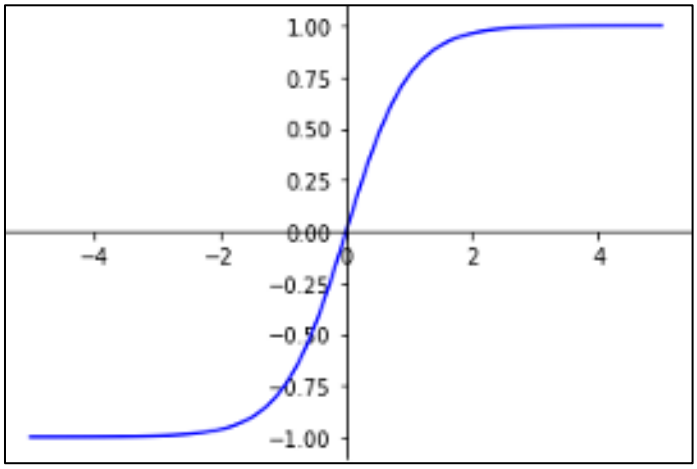


Figure II:6 : Fonction d'activation tanh.

La fonction Exponential Linear Unit (ELU) :

Les ELU résultent de valeurs négatives et positives, permettant de pousser l'activation de l'unité moyenne vers zéro, dans une procédure similaire à la normalisation mais avec une

complexité de calcul plus faible, accélérant ainsi l'apprentissage du modèle. Elle est décrite par l'équation suivante :

$$f(s) = \begin{cases} s & si \quad s \geq 0 \\ \alpha(e^s - 1) sinon, \alpha \epsilon \mathbb{R} \end{cases} \quad \text{II.5}$$

Qui donne l'allure suivante :

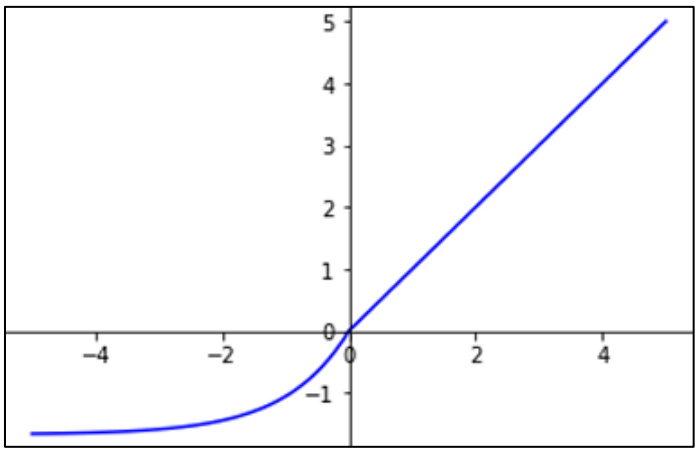


Figure II:7 : Fonction d'activation ELU.

II.2.2.4. **Structure de réseaux de neurones**

Les réseaux de neurones se structurent de manière variée selon leurs types et applications, oscillant entre architectures simples et complexes.

Les réseaux de neurones à propagation avant (feedforward) se composent d'une succession de couches linéaires et non linéaires, facilitant le passage de l'information d'une entrée vers une sortie sans rétroaction.

Les réseaux convolutifs (CNN) introduisent des filtres dans leurs couches cachées pour extraire automatiquement des caractéristiques hiérarchisées à partir d'images ou de signaux, en mettant en évidence localement des motifs répétitifs.

De leur côté, les réseaux récurrents (RNN) et leurs variantes telles que LSTM ou GRU intègrent des connexions récurrentes qui permettent de modéliser des dépendances temporelles en « mémorisant » des informations sur une séquence, et ce, en traitant l'information de manière séquentielle.

Les architectures modernes à base d'attention, comme les Transformers, offrent une structure qui alloue des poids variables à différentes parties des données d'entrée, optimisant ainsi la gestion des dépendances contextuelles sur de longues séquences, ce qui démontre la riche diversité dans la conception des réseaux de neurones pour répondre aux besoins spécifiques de chaque domaine d'application.

Il existe, donc, une multitude de structures de réseaux de neurones. La figure II.8 donne un aperçu des structures courantes couramment utilisées.

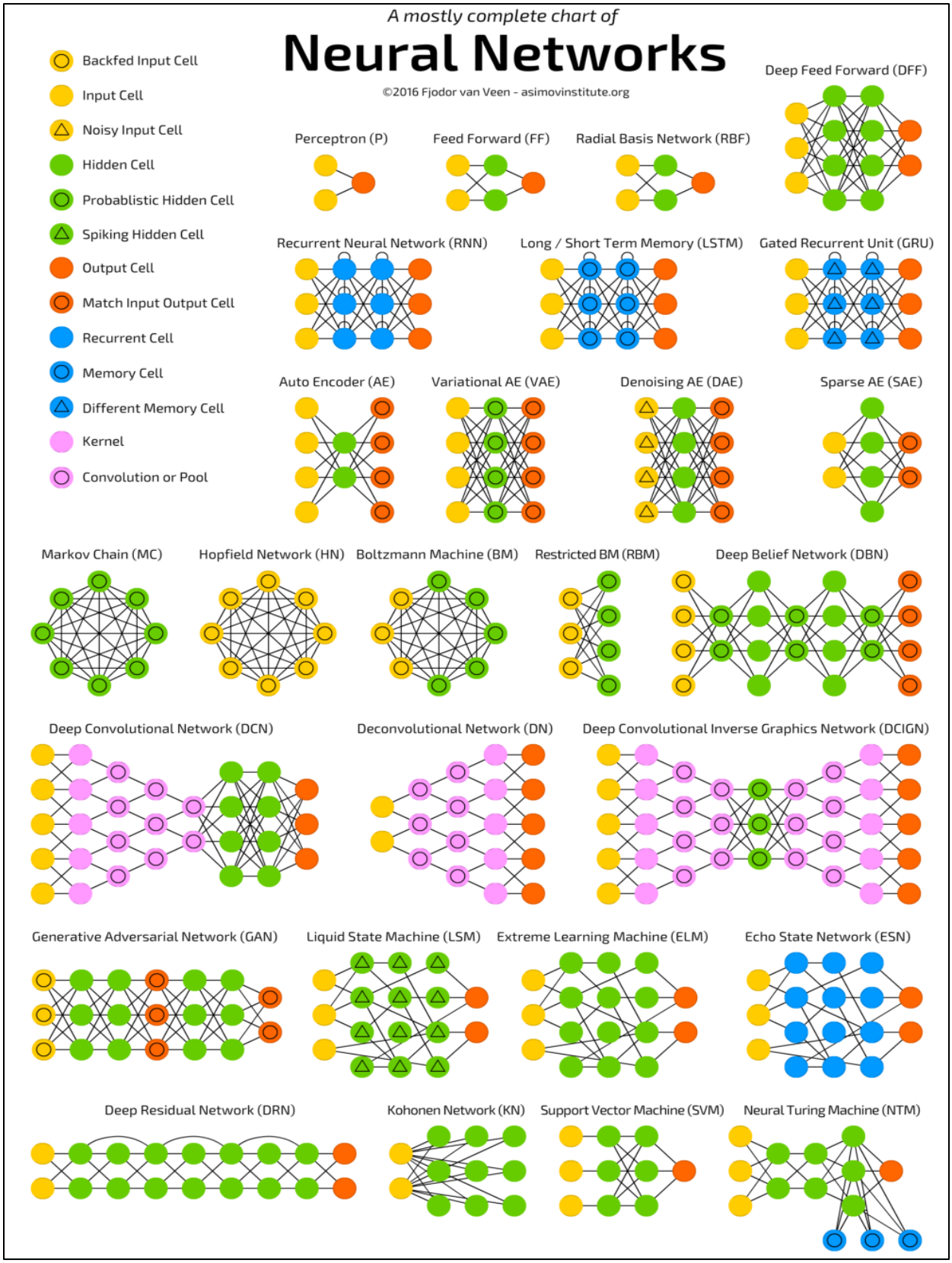


Figure II:8 : Un résumé des types d'architecture de réseaux de neurones.

II.3. **Deep Learning**

Le Deep Learning ou apprentissage profond, est une technique d'intelligence artificielle qui utilise des réseaux de neurones artificiels pour résoudre des problèmes complexes. Ces réseaux sont composés d'un nombre important de couches interconnectées qui permettent d'extraire des informations et des caractéristiques à partir de grandes quantités de données. Le Deep Learning est utilisé pour des tâches comme la reconnaissance d'images (traitement d'image), la traduction automatique, la synthèse vocale, et bien d'autres.

II.3.1. **Principe de fonctionnement du Deep Learning**

Les modèles de Deep Learning sont fondés sur des architectures de réseaux de neurones. Chaque neurone reçoit et interprète les informations de la couche précédente. Les neurones situés entre les couches d'entrée et de sortie d'un réseau de neurones sont appelés couches cachées. Le terme « Deep » fait généralement référence au nombre de couches cachées (jusqu'à des milliers) dans le réseau de neurones.

La figure II.9 illustre le principe du Deep Learning. Le système apprendra par exemple à reconnaître les lettres avant de s'attaquer aux mots dans un texte, ou détermine s'il y a un visage sur une photo avant de découvrir de quelle personne il s'agit. À chaque étape, les « mauvaises » réponses sont éliminées et renvoyées vers les niveaux en amont pour ajuster le modèle mathématique. Au fur et à mesure, le programme réorganise les informations en blocs plus complexes. Lorsque ce modèle est par la suite appliqué à d'autres cas, il est normalement capable de reconnaître un chat sans que personne ne lui ait jamais indiqué qu'il n'ait jamais appris le concept de chat. Les données de départ sont essentielles : plus le système accumule d'expériences différentes, plus il sera performant.

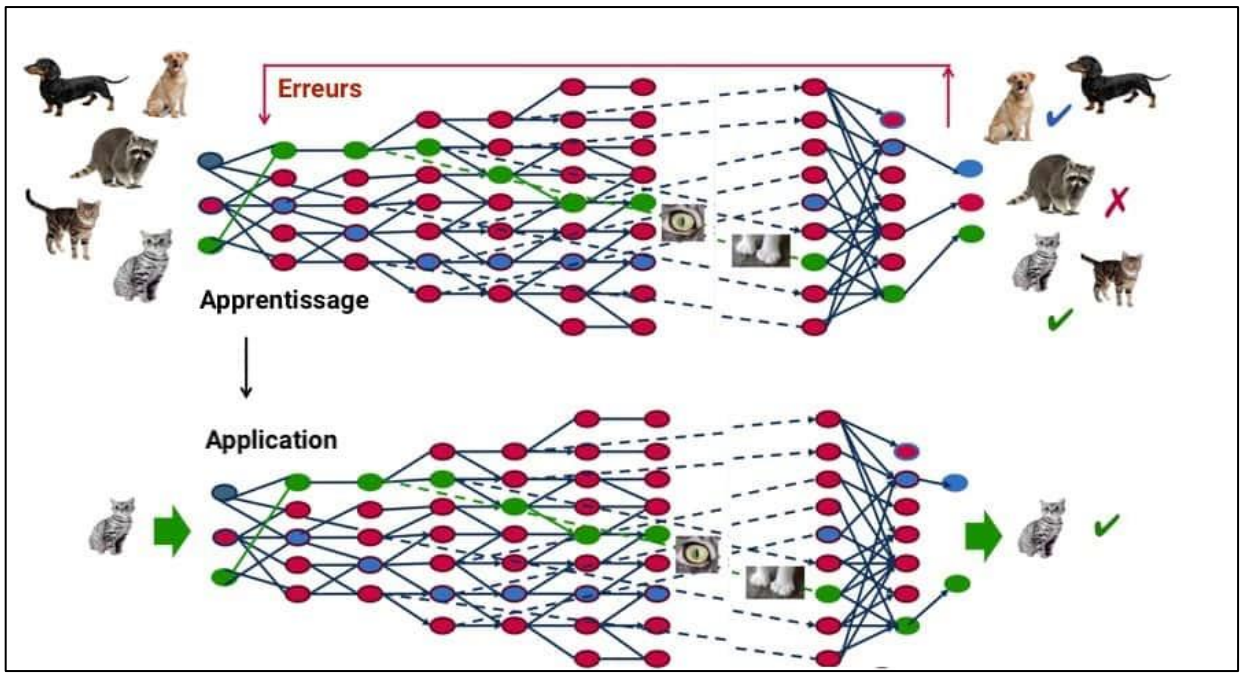


Figure II:9 : Schéma explicatif du principe de fonctionnement du Deep Learning.

II.3.2. **Avantages du Deep Learning :**

Performance élevée sur les grandes quantités de données : Les réseaux de neurones profonds assure de meilleures performances lorsqu'ils sont entraînés sur de vastes ensembles de données. Leur architecture leur permet de capturer des relations complexes non linéaires. Ils surpassent souvent les méthodes traditionnelles dès que les données sont abondantes. Cela les rend particulièrement puissants pour les applications industrielles à grande échelle.

L'Automatisation de l'extraction des caractéristiques : Un workflow de ML commence par l'extraction manuelle des caractéristiques pertinentes à partir des données. Les caractéristiques sont ensuite utilisées pour créer un modèle capable de prédire l'évolution de nouvelles données. Avec un workflow de DL, dans des applications telles que la reconnaissance d'images ou la Computer Vision, les caractéristiques pertinentes sont automatiquement extraites des données de l'image. Lorsqu'il traite des signaux ou des données textuelles, le DL peut réduire le prétraitement des données nécessaire.

Adaptabilité à différents types de données : Le deep learning peut être utilisé avec des images, du texte, des signaux, des vidéos, etc., sans modification profonde du modèle de base. Les architectures peuvent être adaptées pour chaque type de donnée avec des blocs spécialisés (CNN,

RNN, etc.). Cette flexibilité dépasse les capacités des méthodes classiques souvent limitées à un format spécifique.

Capacité de généralisation et transfert d'apprentissage : Les modèles de deep learning peuvent être préentraînés sur un domaine et réutilisés dans un autre via le transfert learning. Cela réduit les besoins en données et en temps d'entraînement pour les nouvelles tâches. C'est une avancée majeure par rapport aux techniques classiques, souvent conçues pour une seule tâche. Elle ouvre la voie à des systèmes intelligents plus souples et économiques.

II.3.3. **Contraintes d'utilisation du Deep Learning**

Complexité des modèles : Les réseaux de neurones profonds comportent souvent des millions de paramètres à ajuster. Cette complexité rend leur conception, réglage et interprétation difficiles. Ils fonctionnent souvent comme des "boîtes noires", ce qui limite leur transparence.Cela complique leur adoption dans des domaines exigeant une forte explicabilité (santé, droit...).

Temps d'apprentissage long : L'entraînement des modèles profonds peut prendre plusieurs heures, voire plusieurs jours. Cela dépend du volume de données, de la profondeur du réseau et des ressources disponibles.Les itérations sont nombreuses et les algorithmes d'optimisation nécessitent un grand nombre d'époques. Cela ralentit le développement et l'expérimentation des solutions.

Besoin de processeurs puissants : Le Deep Learning nécessite des ressources matérielles avancées, comme des GPU ou TPU (Tensor Processing Unit). Les CPU classiques sont généralement insuffisants pour des tâches complexes. Cela peut représenter un coût élevé, surtout pour les chercheurs qui ne possèdent pas de ressources financières suffisantes. L'infrastructure doit aussi gérer la mémoire, le stockage et la parallélisation.

Nécessité de grandes bases de données : Les réseaux profonds ont besoin de beaucoup de données annotées pour bien généraliser. Une base de données insuffisante peut provoquer le surapprentissage (overfitting). Cela limite leur utilisation dans les domaines où les données sont rares ou sensibles.La collecte, l'étiquetage et le nettoyage de ces données demandent aussi du temps et des ressources.

II.4. **Réseaux de neurones convolutifs**

Les réseaux de neurones convolutifs (CNN ou ConvNet) constituent une évolution majeure par rapport aux perceptrons multicouches (MLP), permettant de pallier certaines de leurs

limitations. C'est LeCun et son équipe qui ont marqué un tournant important dans l'évolution des CNN en proposant le réseau LeNet, une architecture spécialisée dans la reconnaissance de chiffres manuscrits.

Les réseaux CNN comportent deux parties bien distinctes comme le montre la figure II.10.

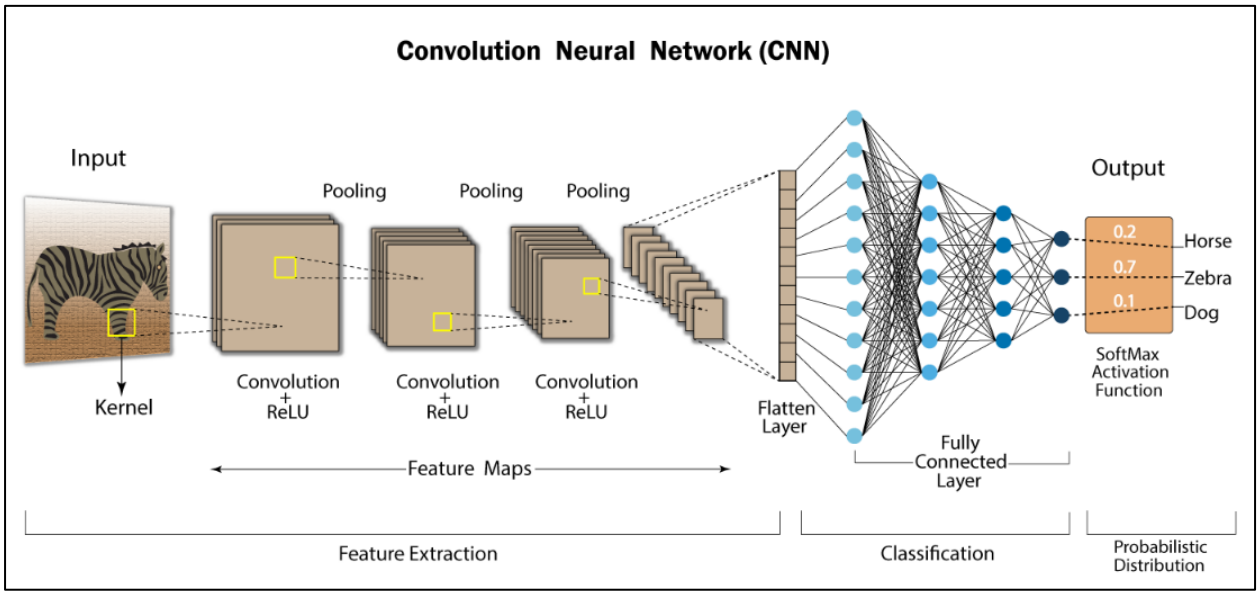


Figure II:10 : Architecture de base d'un réseau CNN.

La première partie d'un réseau de neurones convolutif (CNN) agit comme un extracteur automatique de caractéristiques à partir des images. L'image est traitée successivement par plusieurs filtres (ou noyaux de convolution), générant des cartes de convolution représentant différentes composantes visuelles. Certaines de ces couches appliquent une réduction de dimension, comme le *max pooling*, pour diminuer la résolution tout en conservant l'information essentielle. À la fin de cette phase, les cartes sont aplaties et combinées en un vecteur de caractéristiques appelé **code CNN**, qui sert d'entrée à la seconde partie du réseau.

La seconde partie du CNN assure la tâche de classification, c'est-à-dire la prédiction de la classe à laquelle appartient l'image. Elle repose sur une ou plusieurs couches entièrement connectées (*Fully Connected layers*), qui exploitent le vecteur de caractéristiques extrait précédemment pour produire la décision finale. [15]

II.4.1. **Les couches du CNN**

II.4.1.1. **Les couches de la partie d'extraction des caractéristiques**

Ce premier bloc fait la particularité de ce réseau, puisqu'il fonctionne comme un extracteur de caractéristiques. La première couche de ce bloc filtre l'image avec plusieurs noyaux de convolution, et renvoie des caractéristiques qui sont normalisées ou redimensionnées via une fonction d'activation, ce qui donne une carte de caractéristiques. Ce processus peut être répété plusieurs fois. Les cartes de caractéristiques obtenues sont filtrées en utilisant de nouveaux noyaux à chaque fois, ce qui donne de nouvelles caractéristiques à normaliser et à filtrer à nouveau, et ainsi de suite. Enfin, les valeurs des dernières caractéristiques sont concaténées et aplaties en un vecteur colonne. Ce vecteur définit la sortie du premier bloc, et l'entrée du second. Plus précisément, il existe essentiellement deux types de couches dans cette partie du réseau. Les couches convolutionnelles et les couches de Pooling.

Couches convolutionnelles : La convolution consiste à appliquer un filtre (ou kernel) sur une image en le déplaçant selon un pas appelé stride. Pour conserver la taille d'origine de l'image, on utilise un padding, qui ajoute des valeurs (souvent des zéros) aux bords. À chaque sortie de convolution, on ajoute un biais, une valeur constante propre à chaque filtre.

La figure II.11 illustre une couche de convolution avec trois entrées et deux sorties, calculées à l'aide de deux groupes de filtres. Chaque groupe contient trois filtres 3×3, chacun étant appliqué à une entrée. Les résultats sont ensuite sommés et un biais est ajouté à chaque sortie. Le stride est de 2, ce qui réduit la taille des cartes de sortie.

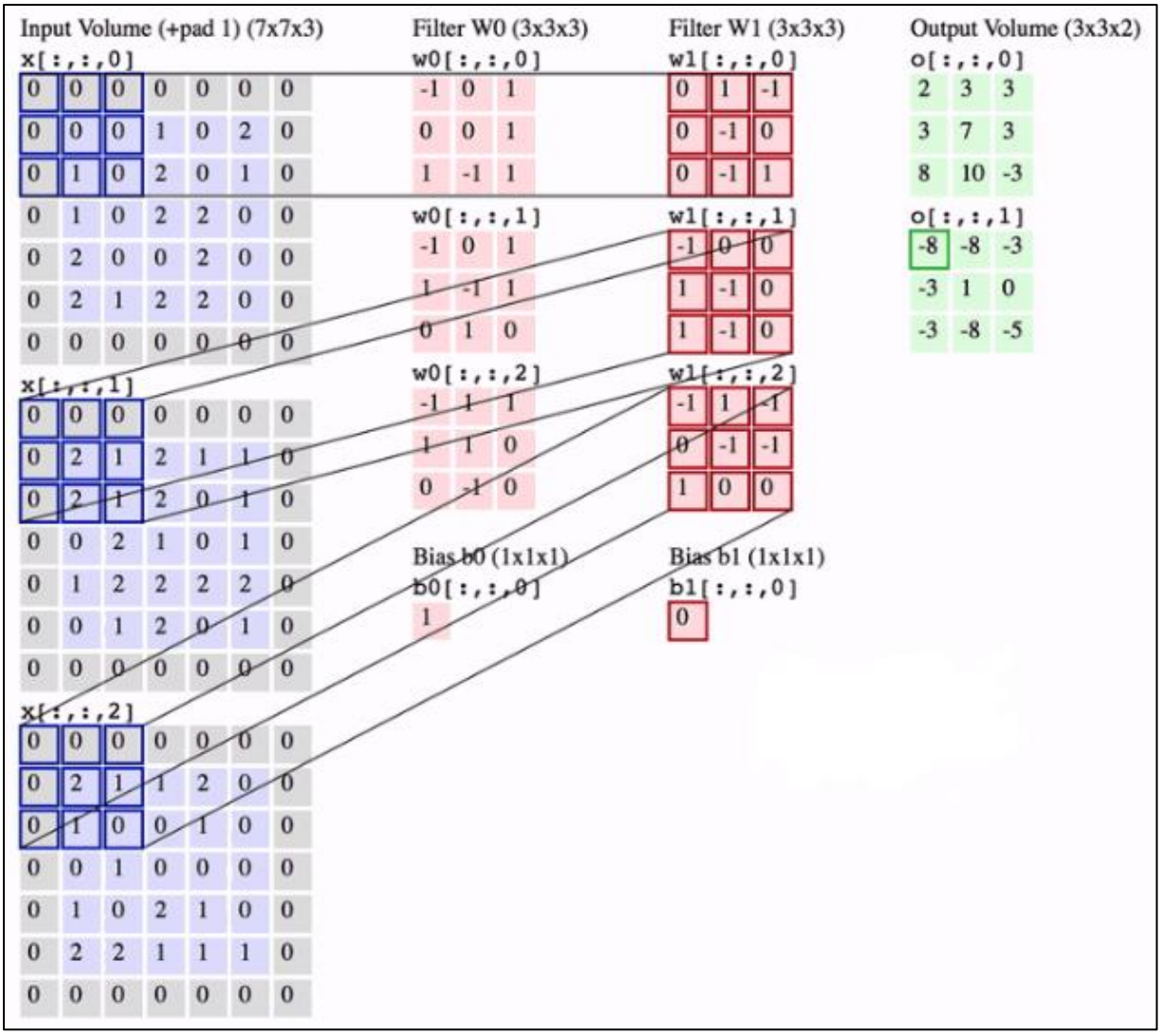


Figure II:11 : Illustration du principe de fonctionnement d'une couche de convolution à trois entrées et deux sorties.

Couche de correction ReLU (Rectified Linear Unit) : On fait passer les cartes de convolution à travers une couche d'activation non linéaire telle que Rectified Linear Unit (ReLu), qui consiste à remplacer les nombres négatifs des images filtrées par des zéros.

Couches de Pooling : Cette couche consiste à réduire progressivement la taille de l'image en ne gardant que les informations les plus importantes, par exemple pour chaque groupe de 4 pixels, le pixel ayant la valeur maximales (*Max Pooling, le plus populaire*) ou la moyenne des pixels (Average pooling).

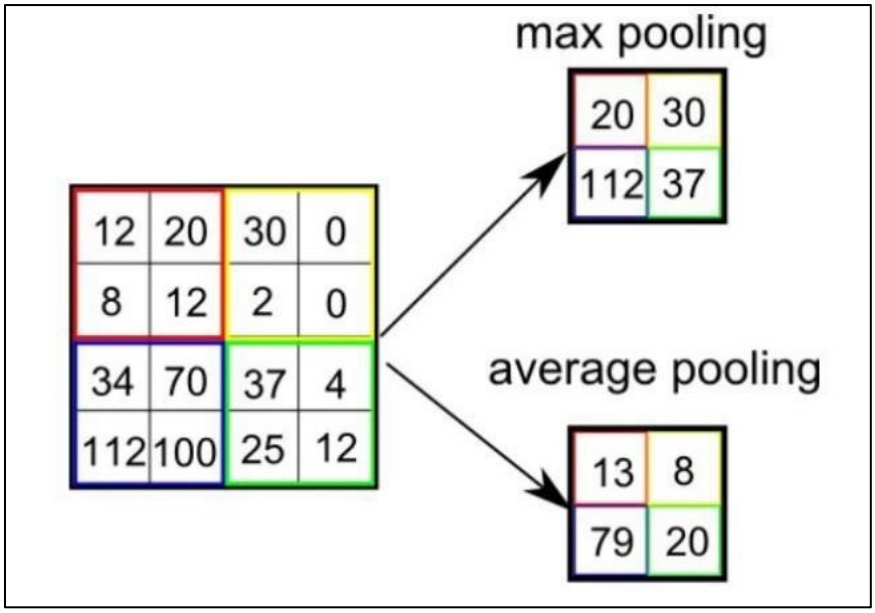


Figure II:12 : Principe des deux types de sous échantillonnage (Pooling).

La Batch Normalization (normalisation par lots) est une technique de normalisation des entrées d'un réseau de neurones artificiels qui a été proposée en 2015 par Ioffe et Szegedy et qui permet d'accélérer l'apprentissage. Elle consiste à normaliser les activations d'une couche en soustrayant la moyenne et en divisant par l'écart type des activations dans un mini-lot d'exemples d'entraînement. De plus, la normalisation par lots agit comme une régularisation et peut réduire le besoin d'autres formes de régularisation, telles que la désactivation aléatoire et le lissage des poids.

Drop-out : Le Drop-Out est une technique de régularisation simple mais efficace. Pendant l'entraînement, le Drop-Out élimine aléatoirement une fraction des neurones du réseau, généralement entre **20% et 50%**, dans chaque couche. Cela signifie que chaque neurone a une chance d'être temporairement “dropped” (éliminé) au cours d'une étape d'entraînement particulière. Cela oblige le réseau à compter sur plusieurs neurones pour représenter des caractéristiques, plutôt que de dépendre d'un petit sous-ensemble de neurones. Le principe d’utilisation du drop-out est de désactiver aléatoirement des neurones afin que le réseau soit plus réactif et puisse donc apprendre plus rapidement. On l’utilise généralement après la couche convolutionnelle et entièrement connectée. [16]

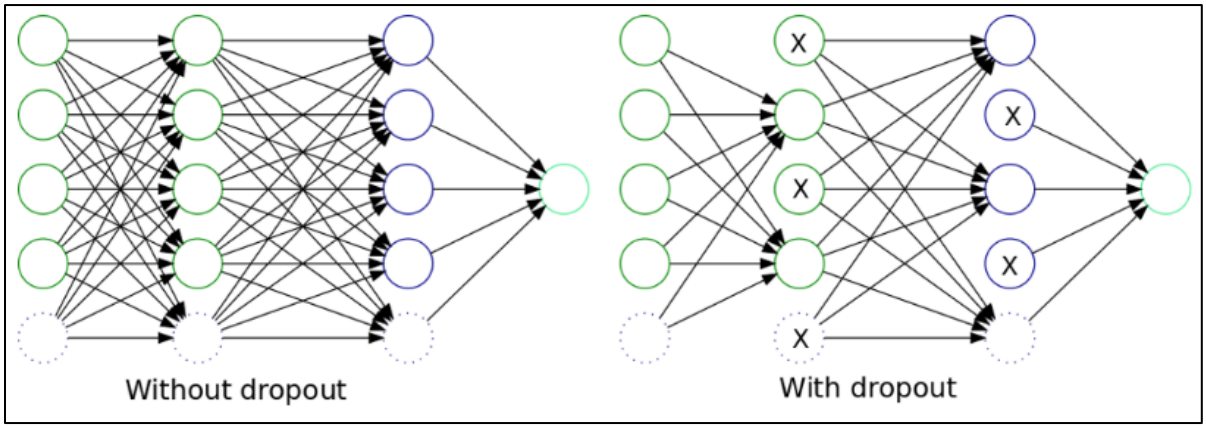


Figure II:13 : Principe du drop-out durant l'entraînement d'un modèle.

Couche d'aplatissement (Flatten layer) : Il s'agit de la dernière étape du bloc d'extraction de caractéristiques. L'aplatissement consiste à concaténer les lignes des matrices en un vecteur unidimensionnel que nous pouvons connecter au bloc de prédiction.

II.4.1.2. **Bloc des couches de classification**

Couches complètement connectées (FullyConnected FC) : Chaque neurone dans cette couche est connecté à tous les neurones de la couche précédente, suivant le principe de base du réseau neuronal multicouche conventionnel.

Les poids de chaque couche complètement connectée sont représentés sous forme d'une matrice unique à deux dimensions, dont une dimension représente le nombre d'entrées de la couche et l'autre dimension représente le nombre de sorties de la couche. La sortie d'une couche complètement connectée est obtenue en faisant le produit matrice-vecteur de la matrice des poids et du vecteur des caractéristiques d'entrée, suivie de l'ajout d'un vecteur de biais.

L'objectif de la couche complètement connectée est d'utiliser les caractéristiques pour classer l'image d'entrée du réseau en différentes classes en fonction.

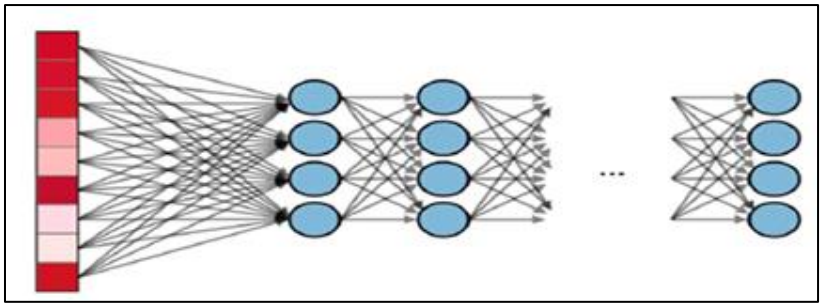


Figure II:14 : Couche entièrement connectée (FullyConnected layer).

La fonction softmax : La fonction softmax est une fonction d'activation souvent utilisée dans les réseaux de neurones pour produire des sorties normalisées et interprétables. Elle prend un vecteur de scores réels en entrée et produit un vecteur de probabilités normalisées en sortie. La valeur de la probabilité est calculée comme suit :

$$f(s_j) = \frac{e^{s_j}}{\Sigma_i e^{si}} \quad \text{II.6}$$

Où s_j est l'élément considéré j du vecteur d'entrée. La classe ayant la plus grande probabilité sera prise comme sortie du réseau de neurones.

II.4.2. **Etapes de classification des images par réseaux CNN**

La classification d'images par réseaux de neurones convolutifs (CNN) est une méthode puissante qui permet de reconnaître automatiquement le contenu visuel des images. Grâce à leur capacité à extraire et apprendre des caractéristiques discriminantes, les CNN sont devenus la référence dans les tâches de vision par ordinateur.

Ce processus se déroule en plusieurs étapes successives, depuis le traitement initial de l'image jusqu'à la prédiction finale. Dans ce qui suit, nous détaillerons les principales phases de cette classification incluant la préparation de la base de données, l'apprentissage du réseau et finalement l'évaluation du réseau.

II.4.2.1. **Préparation des Données**

Lors de l'entraînement d'un réseau neuronal pour des tâches de classification, il est crucial de disposer d'un ensemble de données vaste et diversifié qui représente fidèlement le problème à

résoudre. L'efficacité et la robustesse du modèle dépendent fortement de la qualité et de la structuration de ces données. L'ensemble de données est généralement divisé en trois sous-ensembles distincts, chacun ayant un rôle spécifique dans le processus d'apprentissage et d'évaluation du modèle :

Données d'entraînement (Training Data) : Ce sont les données principales utilisées pour enseigner le modèle. Le réseau neuronal ajuste ses poids et ses biais de manière itérative en fonction de ces données, cherchant à minimiser l'erreur entre ses prédictions et les étiquettes réelles. C'est sur cet ensemble que le modèle apprend les motifs et les caractéristiques nécessaires à la classification.

Données de validation (Validation Data) : Contrairement aux données de test, les données de validation ne servent pas à évaluer la performance finale, mais à surveiller et ajuster le modèle durant l'apprentissage. Elles permettent de détecter le surapprentissage (*overfitting*) en comparant les performances sur l'ensemble d'entraînement et de validation. De plus, elles sont utilisées pour choisir les hyperparamètres optimaux (comme le taux d'apprentissage, la taille du réseau, ou le nombre d'époques). En résumé, les données de validation aident à contrôler et améliorer la généralisation du modèle avant son évaluation finale sur les données de test.

Données de test (Testing Data) : Une fois que le modèle est entièrement entraîné et que ses hyperparamètres ont été optimisés à l'aide des données de validation, l'ensemble de test est utilisé pour une évaluation finale et impartiale. Ces données n'ont jamais été vues par le modèle pendant l'entraînement ou la validation, ce qui permet d'estimer sa capacité réelle à généraliser à des données nouvelles et inconnues. Les métriques de performance obtenues sur cet ensemble sont les plus représentatives de la performance du modèle en conditions réelles.

La figure II.14 illustre le rôle de chacun des trois sous-ensembles de la base de données.

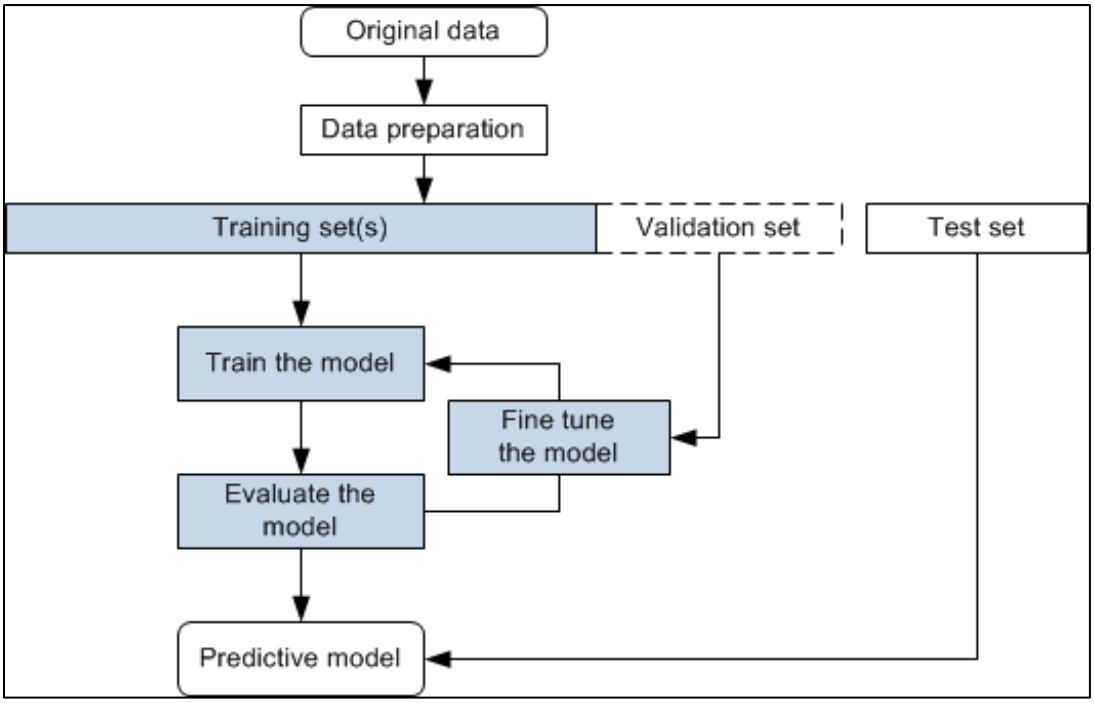


Figure II:15 : Utilité de chacun des trois sous-ensembles d'une base de données : training, validation et test.

II.4.2.2. **Entrainement du modèle**

Le processus d'entraînement d'un modèle de Deep Learning pour la classification suit généralement les étapes suivantes :

Initialisation des poids et biais : Au début de l'entraînement, les poids et les biais de toutes les connexions du réseau neuronal sont initialisés aléatoirement. Cette initialisation est cruciale car elle brise la symétrie et permet au réseau d'apprendre des caractéristiques différentes.

Propagation avant (Forward Propagation) : Les données d'entrée sont transmises à travers le réseau, de la couche d'entrée vers la couche de sortie. À chaque neurone, la somme pondérée de ses entrées est calculée, puis une fonction d'activation est appliquée pour produire une sortie. Cette sortie est ensuite transmise aux neurones de la couche suivante, et ce processus se poursuit jusqu'à ce que la couche de sortie produise la prédiction finale du modèle.

Calcul de la fonction de perte (Loss Function) : La prédiction du modèle (la sortie de la propagation avant) est comparée à la vérité terrain (l'étiquette réelle des données d'entrée) à l'aide d'une fonction de perte (ou fonction de coût). Cette fonction quantifie l'écart entre la prédiction et la réalité, c'est-à-dire l'erreur du modèle. L'objectif de l'entraînement est de minimiser cette perte.

Rétropropagation (Backpropagation) : L'erreur calculée par la fonction de perte est ensuite propagée en arrière à travers le réseau, de la couche de sortie vers la couche d'entrée. Pendant cette phase, les gradients de la fonction de perte par rapport à chaque poids et biais du réseau sont calculés. Ces gradients indiquent la direction et l'ampleur des ajustements nécessaires pour réduire l'erreur.

Optimisation (Optimization) : Un algorithme d'optimisation, tel que la descente de gradient (ou ses variantes comme Adam, RMSprop, etc.), utilise les gradients calculés lors de la rétropropagation pour ajuster les poids et les biais du réseau. Ces ajustements sont effectués de manière à réduire progressivement la valeur de la fonction de perte, améliorant ainsi la précision des prédictions du modèle.

Les étapes de propagation avant, calcul de la perte, rétropropagation et optimisation sont répétées de nombreuses fois sur l'ensemble des données d'entraînement. Chaque passage complet sur l'ensemble des données d'entraînement est appelé une époque (epoch). Ce processus itératif se poursuit jusqu'à ce que le modèle converge (c'est-à-dire que la fonction de perte ne diminue plus significativement) ou que les performances sur l'ensemble de validation cessent de s'améliorer, indiquant un risque de surapprentissage.

II.4.2.3. **Évaluation du Modèles de Classification**

Une fois le modèle de Deep Learning entraîné, son évaluation est une étape critique pour s'assurer de sa performance et de sa capacité à généraliser sur des données inconnues. Cette évaluation se fait principalement sur l'ensemble de données de test, qui n'a pas été utilisé pendant l'entraînement ou la validation. Plusieurs métriques sont couramment utilisées pour quantifier la performance d'un modèle de classification [16] :

Précision : elle est calculée comme le rapport entre le nombre d'échantillons positifs correctement classés et le nombre total d'échantillons classés comme positifs (correctement ou incorrectement). La précision mesure l'exactitude du modèle à classer un échantillon comme positif.

$$précision = \frac{TP}{TP+FP} \quad \text{II.7}$$

Avec : TP = vrais positifs (True Positives), FP = faux positifs (False Positives)

Justesse (Accuracy) : c'est une mesure qui décrit généralement les performances du modèle dans toutes les classes. Il est utile lorsque toutes les classes sont d'égale importance. Il est calculé comme le rapport entre le nombre de prédictions correctes et le nombre total de prédictions.

$$Justesse = \frac{TP+TN}{TP+FP+TN+FN} \quad \text{II.8}$$

- Avec : TN = vrai négatifs (True Negatives), FN = faux négatifs (False Negatives)

Rappel (recall) : Aussi appelé sensibilité, il mesure la proportion de vrais positifs correctement identifiés parmi tous les positifs réels.

$$Rappel = \frac{TP}{TP+FN} \quad \text{II.9}$$

Score F1 : La moyenne harmonique de la précision et du rappel. Il est particulièrement utile lorsque l'on recherche un équilibre entre précision et rappel, surtout avec des classes déséquilibrées.

$$F_1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad \text{II.10}$$

Matrice de Confusion : La matrice de confusion est un outil de mesure de la performance des modèles de classification dans laquelle on spécifie pour chaque classe le nombre d'élément classés correctement et le nombre d'éléments classés d'une manière erronée.

Pour un système de classification à plus de deux classes les étiquettes réelles sont représentées sur les lignes (ou colonnes) de la matrice de confusion et les étiquettes prédites (résultats de classification) sont représentés sur les colonnes (ou lignes). Les éléments diagonaux représentent le nombre de points pour lesquels l'étiquette prédite est égale à la véritable étiquette, tandis que les éléments hors diagonale sont ceux qui sont mal étiquetés par le système. Plus les valeurs diagonales de la matrice de confusion sont élevées, mieux est le système de classification.

Courbe ROC (Receiver Operating Characteristic) et AUC (Area Under the Curve) : Une courbe ROC est obtenue en traçant le *taux de vrais positifs* (TPR) en fonction du *taux de faux positifs* (FPR). Le taux de vrais positifs est la proportion d'observations qui ont été correctement prédites comme étant positives sur toutes les observations positives :

$$TPR = TP / (TP + FN) \quad \text{II.11}$$

De même, le taux de faux positifs est la proportion d'observations qui sont incorrectement prédites positives sur toutes les observations négatives :

$$FPR = FP / (TN + FP) \qquad \text{II.12}$$

Les classificateurs qui donnent des courbes plus proches du coin supérieur gauche indiquent une meilleure performance La courbe ROC est un graphique qui illustre la performance d'un modèle de classification binaire à tous les seuils de classification. L'AUC représente l'aire sous cette courbe, fournissant une mesure agrégée de la performance du modèle.

II.5. **Classification de modulation par Deep Learning**

II.5.1. **Limitation des approches traditionnelles de la classification automatique de modulation**

Nous avons vu dans la fin du chapitre 1 que l'approche classique de l'AMC a été abordée selon deux méthodes classiques de classification : les méthodes basées sur la vraisemblance (Likelihood-Based Methods) et les méthodes basées sur l'extraction de caractéristiques (Feature-Based Methods).

Malgré leur succès relatif, les aproches traditionnelles présentent plusieurs limitations lorsqu'elles sont confrontées aux défis réels des communications sans fil modernes :

- Sensibilité significative au bruit et aux interférences
- Performances dégradées en présence de canaux sélectifs en fréquence ou à évanouissements rapides
- Difficulté à généraliser à de nouveaux types de modulations ou à des conditions de canal non rencontrées précédemment
- Nécessité d'une expertise considérable pour la conception manuelle des extracteurs de caractéristiques.

C'est dans ce contexte que les approches basées sur l'apprentissage profond (Deep Learning) ont émergé comme une alternative prometteuse pour résoudre les défis de la classification automatique de modulation.

II.5.2. **Évolution vers les approches Deep Learning**

L'application du Deep Learning à la classification de modulation représente un changement de paradigme significatif par rapport aux méthodes conventionnelles. Cette transition est motivée par plusieurs facteurs :

- **Robustesse accrue :** Les modèles profonds ont démontré une meilleure tolérance au bruit et aux distorsions de canal par rapport aux approches classiques [17].
- **Performances supérieures :** Des gains significatifs de précision ont été observés, particulièrement dans les scénarios à faible rapport signal/bruit (SNR) [18].
- **Capacité d'extraction automatique de caractéristiques :** Les architectures de Deep Learning peuvent apprendre directement à partir des données brutes, éliminant ainsi la nécessité d'une conception manuelle d'extracteurs de caractéristiques [20][21].
- **Adaptabilité :** Capacité à s'adapter à de nouveaux types de modulations et à des conditions de canal variables par simple réentraînement [22].

L'évolution des approches Deep Learning pour l'AMC peut être résumée en trois générations principales :

1. Première génération : Application de réseaux profonds traditionnels (DNN) à des caractéristiques extraites manuellement [23].
2. Deuxième génération : Introduction des réseaux de neurones convolutifs (CNN) et récurrents (RNN) appliqués directement aux signaux bruts ou à des représentations temps-fréquence [21][24].
3. Troisième génération : Architectures hybrides et avancées incluant mécanismes d'attention, réseaux adverses, et apprentissage par transfert [25][26].

II.5.3. **Prétraitement et représentation des signaux**

Une étape critique dans la conception d'un système AMC basé sur l'apprentissage profonde est le choix de la représentation du signal. Ce choix influence directement l'efficacité avec laquelle les réseaux neuronaux peuvent extraire des caractéristiques discriminantes.

Les principales représentations utilisées sont :

II.5.3.1. **Représentation IQ**

La représentation IQ (In-phase/Quadrature) est la forme la plus directe d'un signal de communication numérique. Un signal complexe $x(t) = I(t) + jQ(t)$ est représenté comme une séquence de paires de valeurs réelles $(I[n], Q[n])$ correspondant aux composantes en phase et en quadrature [17]. Pour un signal échantillonné de longueur N, la représentation peut être formulée comme :

$$X_{IQ} = \begin{pmatrix} I[0]\, I[1] & \dots & I[N-1] \\ Q[0] Q[1] \dots & & Q[N-1] \end{pmatrix} \quad \text{II.13}$$

Cette représentation est particulièrement adaptée aux CNN qui peuvent traiter $\boldsymbol{X_{IQ}}$ comme une image à 2 canaux, chaque canal correspondant aux composantes I et Q [20].

II.5.3.2. **Représentation en constellation**

La représentation en constellation visualise les symboles complexes dans un plan bidimensionnel, où l'axe horizontal représente la composante en phase (I) et l'axe vertical représente la composante en quadrature (Q). Cette visualisation révèle des motifs géométriques distinctifs pour différentes modulations (par exemple, une disposition quadratique pour 16-QAM, circulaire pour PSK) [19].

Pour un ensemble de symboles complexes $s_k = I_k + jQ_k$, la constellation est définie comme l'ensemble de points :

$$C = (I_k, Q_k) | k = 1,2, \dots, K \quad \text{II.14}$$

II.5.3.3. **Représentations temps-fréquence**

Les représentations temps-fréquence capturent l'évolution du contenu fréquentiel d'un signal au cours du temps [26]. Parmi celles-ci, le spectrogramme, dérivé de la transformée de Fourier à court terme (STFT), est particulièrement populaire :

$$S(t,f) = \left| \int_{\{-\infty\}}^{\{\infty\}} x(\tau) w(\tau - t) e^{\{-j\, 2\, \pi f\, \tau\},} d\tau \right|^2 \quad \text{II.15}$$

Où $w(t)$ est une fenêtre d'analyse.

D'autres représentations temps-fréquence incluent les scalogrammes (basés sur la transformée en ondelettes), les transformées de Wigner-Ville, et les cyclo-stationnarités qui capturent la périodicité des statistiques du signal.

II.5.4. **Architectures de réseaux de neurones pour la classification de modulation**

Diverses architectures de réseaux neuronaux ont été appliquées à l'AMC, chacune présentant des avantages spécifiques. Cette section examine les principales catégories, avec une attention particulière aux CNN qui se sont révélés particulièrement efficaces pour cette tâche.

II.5.4.1. **Réseaux de neurones convolutifs (CNN)**

Les CNN ont émergé comme l'architecture dominante pour l'AMC en raison de leur capacité à exploiter efficacement les structures spatiales et les corrélations locales présentes dans les données de signal [21][24].

En fonction des types de données d'entrée, les approches AMR basées sur des CNN existants peuvent être regroupées en deux catégories : les modèles CNN utilisant des entrées I/Q brutes et les modèles CNN utilisant des entrées prétraitées.

Modèles CNN basés sur des entrées de données I/Q brutes : En utilisant un modèle CNN simple à quatre couches prenant des données I/Q comme entrées, le potentiel des CNN appliqués à la reconnaissance automatique de la modulation (AMR) a été exploré pour la première où une précision de reconnaissance plus élevée que de nombreuses méthodes classiques a été obtenue. Quelques modèles CNN améliorés, intégrant des couches complexes ou des opérations de transformation spécifiques, ont été appliqués à l'AMR. Par exemple, des modèles inspirés par les ResNet et DenseNet) ont été proposés pour permettre aux caractéristiques extraites de multiples couches d'être efficacement transmises au module de détection. Des modèles de classification de la modulation des signaux en temps réel pour les systèmes OFDM ont été proposés par des spécialistes.

Modèles CNN classiques utilisant des entrées prétraitées : ces modèles utilisent les caractéristiques importantes facilement extraites par des méthodes classiques de traitement du signal, telles que les cumulants d'ordre supérieur, les images spectrales ou les diagrammes de constellation. Combiner les méthodes classiques AMR basées sur des caractéristiques (FB-AMR) avec des modèles CNN a montré un potentiel considérable pour surmonter ces limitations. Par

exemple, [27] applique une transformée de Fourier discrète à court terme pour convertir des signaux radio 1D en images spectrales, puis applique un filtre gaussien pour réduire le bruit. Le modèle proposé atteint ainsi une précision de reconnaissance élevée démontrant l'efficacité de cette stratégie. Dans [28], les séquences I/Q ainsi que leur transformée Stockwell orthonormée discrète (DOST) sont combinées pour représenter le signal reçu à des fins de reconnaissance de modulation par CNN.

Par ailleurs, le diagramme de constellation est une autre méthode de représentation du signal largement utilisée pour l'AMR ; les caractéristiques peuvent ainsi en être extraites directement afin d'identifier le schéma de modulation. Ce diagramme peut être converti en image à 3 canaux dans pour tirer parti des modèles CNN tels qu'AlexNet et GoogLeNet adaptés à la classification d'images couleur.

Le CNN basé sur le diagramme de constellation est principalement utilisé pour résoudre le problème de confusion entre 16QAM et 64QAM.

Pour tirer parti de la forte capacité de résistance au bruit des images spectrales cycliques ainsi que de la capacité de reconnaissance des schémas de modulation d'ordre supérieur offertes par les diagrammes de constellation, un CNN à deux branches est proposé dans [29] pour extraire des caractéristiques multi-échelles avant de les fusionner.

II.5.4.2. **Modèles basés sur les RNN**

Les signaux de communication sans fil présentent des caractéristiques de corrélation temporelle, que les réseaux de neurones récurrents (RNN) peuvent exploiter pour la reconnaissance de modulation. Plusieurs nouvelles architectures basées sur des RNN ont été proposées atteignant des performances de pointe en AMR. Par exemple, une nouvelle méthode AMR basée sur des RNN est présentée dans [30], où des unités de récurrence contrôlées (GRU) sont utilisées pour atteindre une précision de reconnaissance supérieure à celle de quelques modèles CNN. Les modèles utilisant uniquement deux couches de RNN démontrent des performances remarquables en AMR, confirmant la nature distincte des caractéristiques temporelles des signaux de communication ainsi que la forte capacitée d'extraction de caractéristiques des RNN. D'autres modèle basé sur LSTM intégrant des couches entièrement connectées ainsi qu'un mécanisme d'attention temporelle, s'avère robuste face à des conditions de bruit incertaines.

II.5.4.3. **Modèles hybrides**

Étant donné que les modèles purement CNN ou RNN se concentrent uniquement sur les caractéristiques spatiales ou temporelles des signaux AMR, l'utilisation d'un seul de ces deux types de modèles peut ne pas atteindre des performances optimales. C'est pourquoi des chercheurs ont proposé de combiner les caractéristiques des deux types de couches de réseaux de neurones afin de concevoir des modèles hybrides pour l'AMR. Un modèle de réseaux de neurones profonds convolutionnels et à long terme (CLDNN) a été proposé dans [31], constitué d'un LSTM et de trois couches CNN. Ce modèle possède une connexion de dérivation avant le LSTM, contournant ainsi deux des couches CNN, dans le but de fournir un contexte temporel plus long pour les caractéristiques extraites. La sortie de la première couche de convolution est concaténée à celle de la troisième couche avant d'être transmise à la couche LSTM, permettant à celle-ci d'extraire des caractéristiques temporelles plus efficaces, ce qui conduit à une précision de reconnaissance améliorée ainsi qu'à une descente de gradient plus stable que d'autres modèles.

Pour réduire la complexité du calcul [32], propose un réseau de neurones récurrents quasi-récurentiel (SQRNN) utilisant des couches convolutionnelles à faible complexité pour reproduire le fonctionnement des couches récurrentes.

De plus, une structure à deux flux basés sur CNN-LSTM ont été proposée, où l'amplitude/phse (A/P) ainsi que les données I/Q servent d'entrées pour les deux flux du modèle.

Les approches AMR basées sur l'apprentissage profond reposent fortement sur la qualité des jeux de données. Des bases de données fiables sont essentielles pour l'entraînement, le test et l'évaluation des modèles DL-AMR. Le jeu de données RML2016.10a est devenu une référence, intégrant des effets aléatoires tels que le décalage de fréquence, le bruit gaussien, l'évanouissement multi-trajets ainsi que la dérive de l'horloge d'échantillonnage. RML2016.10b est plus volumineux tandis que RML2018.01a propose 24 schémas de modulation ainsi que des séquences plus longues (1024 points), offrant davantage de détails pour concevoir des modèles avancés malgré un coût de calcul plus élevé. Le jeu de données HisarMod2019.1 fournit 26 schémas de modulation simulés à l'aide de Matlab, couvrant des environnements variés (Rayleigh, Rician, Nakagami-m) mais plus favorables à la reconnaissance. Pour atteindre des performances opérationnelles robustes, un

modèle AMR nécessite un apprentissage affiné à partir de jeux de données captés en conditions de propagation réelle.

Tableau II:1: Tableau comparatif des bases de données AMR.

Jeu de données	Année	Source	Schémas de modulation	Longueur du signal	Conditions de canal
RML2016.10a	2016	O'Shea et al.	11	128	AWGN, décalage de fréquence, fading
RML2016.10b	2016	O'Shea et al.	11	128	AWGN, fading multi-trajets, offset
RML2018.01a	2018	O'Shea et al.	24	1024	AWGN, offset, fading multi-trajets
HisarMod2019.1	2019	Simulation Matlab	26	Variable	Idéal, statique, Rayleigh, Rician, Nakagami-m

II.6. **Conclusion**

Ce chapitre a été consacré à l'étude des méthodes basées sur l'intelligence artificielle appliquées à la classification des signaux de communication. Dans la première partie, nous avons vu les notions générales de l'intelligence artificielle ainsi que le fonctionnement des réseaux de neurones artificiels, qui constituent la base de nombreuses méthodes de traitement du signal moderne. Ensuite, nous avons abordé le Deep Learning, en détaillant ses principes ainsi que sa capacité à extraire automatiquement des caractéristiques complexes à partir de grands volumes de données. Puis, nous nous sommes intéressés aux réseaux de neurones convolutionnels (CNN), adaptés à l'identification de motifs spécifiques grâce à leur structure de convolution, ce qui les rend particulièrement efficaces pour la classification de signaux représentés visuellement. Et finalement, nous avons présenté la classification automatique de la modulation par deep learning, en mettant en évidence des modèles CNN, RNN et hybrides ainsi que les base de données disponibles et utilisées pour l'entrainement et l'évaluation des modèles de classification AMR à base du Deep Learning.

CHAPITRE III : RESULTATS DE SIMULATION : PRESENTATION ET INTERPRETATION

III.1. **Introduction :**

Ce chapitre est dédié à la présentation et à l'analyse des résultats expérimentaux obtenus lors de la classification des types de modulation à l'aide de réseaux de neurones convolutifs (CNN). Le travail repose sur l'exploitation de la base de données RadioML 2016.10a, qui constitue une référence dans le domaine de la reconnaissance automatique de modulations (AMC). Deux représentations du signal ont été utilisées : la forme brute en composantes I/Q (In-phase et Quadrature) et la forme image à travers des représentations temps-fréquence, notamment les spectrogrammes et la transformée de Wigner-Ville (WVD).

Les simulations ont été réalisées dans l'environnement MATLAB, choisi pour ses outils performants en traitement du signal et en apprentissage profond, ainsi que pour sa souplesse d'intégration des réseaux pré-entraînés. Deux approches ont été comparées : un réseau CNN que nous avons proposé et conçu spécifiquement pour ce travail, et un modèle issu du transfert learning basé sur le réseau léger et optimisé SqueezeNet. Les performances ont été évaluées sur différents niveaux de rapport signal sur bruit (SNR) pour refléter des conditions réalistes de communication.

L'objectif est de comparer l'efficacité des différentes approches en fonction du format d'entrée (I/Q ou image), du type de réseau, et du niveau de bruit. Les résultats sont analysés à l'aide des deux métriques classiques qui sont l'exactitude (accuaracy) et la matrice de confusion.

Les simulations ont été effectuées sous MATLAB R2024b, installé sur un ordinateur portable équipé d'un processeur Intel Core i7 de 11^{e} génération cadencé à 3 GHz. La machine dispose de 32 Go de mémoire RAM, permettant un traitement efficace des volumes de données élevés.

III.2. **Présentation de la base de données RML2016a**

La base de données **RadioML 2016.10a,** abrégée RML2016a, est une base de référence utilisée pour l'évaluation des algorithmes de classification de modulation automatique (AMC). Elle a été développée pour simuler un environnement de communication réaliste, incluant des effets de bruit, de décalage de fréquence, et de gigue temporelle. Elle contient des signaux I/Q complexes générés pour **11 types de modulations** (analogiques et numériques) suivantes :

1. **BPSK (Binary Phase Shift Keying)** : Modulation de phase à deux états représentant les bits 0 et 1 par un changement de phase de 180°.

2. **QPSK (Quadrature Phase Shift Keying)** : Utilise quatre phases différentes (0°, 90°, 180°, 270°) pour coder deux bits par symbole.
3. **8PSK (8-Phase Shift Keying)** : Modulation de phase à huit états, codant 3 bits par symbole pour une meilleure efficacité spectrale.
4. **QAM16 (16-Quadrature Amplitude Modulation)** : modulation sur deux porteuses en quadrature codant 4 bits par symbole.
5. **QAM64 (64-Quadrature Amplitude Modulation)** : Extension du QAM16 avec 64 symboles, permettant de coder 6 bits par symbole.
6. **GFSK (Gaussian Frequency Shift Keying)** : Modulation de fréquence lissée par un filtre gaussien.
7. **CPFSK (Continuous Phase Frequency Shift Keying)** : Variante de FSK avec une phase continue, réduisant la largeur de bande occupée.
8. **AM-DSB (Amplitude Modulation - Double Side Band)** : Modulation d'amplitude classique avec deux bandes latérales autour de la porteuse.
9. **AM-SSB (Amplitude Modulation - Single Side Band)** : Variante de l'AM-DSB où une seule bande latérale est transmise pour économiser de la bande.
10. **FM (Frequency Modulation)** : Modulation où l'information est transmise par la variation de la fréquence de la porteuse.
11. **WB-FM (Wideband Frequency Modulation)** : Version à large bande de la FM, comme utilisée dans la radio FM, offrant une meilleure qualité sonore au prix d'une bande passante plus large.

Pour chaque type de modulation le signal reçu est évalué à vingt valeurs du rapport signal sur bruit **(SNR)** allant de -20 dB à +18 dB avec un pas de 2.

Pour chaque type de modulation et pour chaque valeur du rapport signal sur bruit la base offre 1000 exemples de signal. Chaque exemple de signal est représenté par une séquence complexe de **128 échantillons** I/Q, accompagnée de son label de modulation et de la valeur SNR correspondante. Cette base est particulièrement bien adaptée aux techniques d'apprentissage profond, car elle offre un grand volume de données annotées. Elle permet de tester la robustesse des modèles face à des conditions de canal variées, simulant un environnement réaliste pour les systèmes de radio cognitive et les applications de guerre électronique.

III.2.1. **RML2016a en format image en utilisant le spectrogramme :**

Le spectrogramme est une version glissante de la transformée de Fourier. Il montre comment le contenu fréquentiel du signal évolue dans le temps. La fonction *spectrogram* de MATLAB permet de calculer et d'afficher le spectrogramme d'un signal, c'est-à-dire sa représentation temps-fréquence. Elle est très utilisée en traitement du signal pour analyser les signaux non stationnaires, comme les signaux modulés.

Pour transformer la base RML2016a en images de spectrogrammes nous avons suivi les étapes suivantes :

1. Chargement de la base de données : Charger le fichier RML2016.10a.mat, qui contient une structure associant les signaux I/Q aux types de modulation et aux valeurs SNR.
2. Extraction des parties I/Q : Extraire les parties réelle (I) et imaginaire (Q) de chaque échantillon du signal.
3. Génération du spectrogramme : Utiliser la fonction MATLAB spectrogram() pour générer le spectrogramme de chaque exemple de signal.
4. Sauvegarde de l'image (option couleur) : Sauvegarder chaque image avec le nom structuré selon la modulation, le SNR et l'index. La taille des images est choisie 128×128×3.

Pour faciliter la tâche de classification, La base est organisée en dossiers hiérarchiques, selon le **SNR** et le type de modulation selon la structure de la figure III.1.

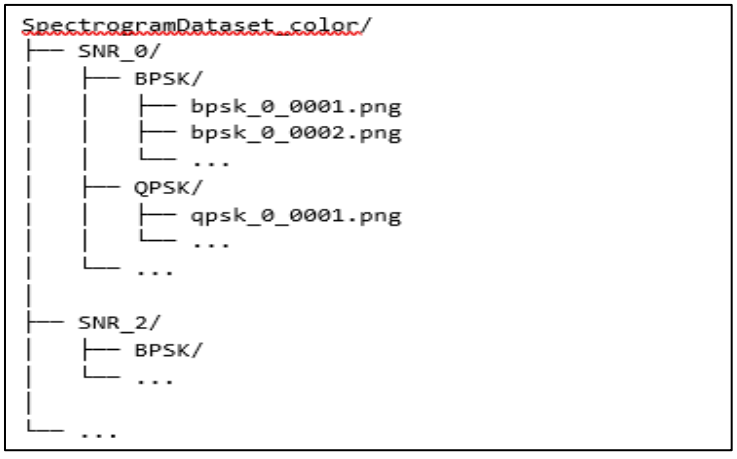


Figure III:1: Organisation hiérarchique des images de spectrogramme de la base RML2016a

La figure III.2 présente quatre exemples de spectrogrammes de signaux, pour les deux modulations BPSK et 64-QAM et pour deux valeurs du rapport SNR 10 dB et 18 dB.

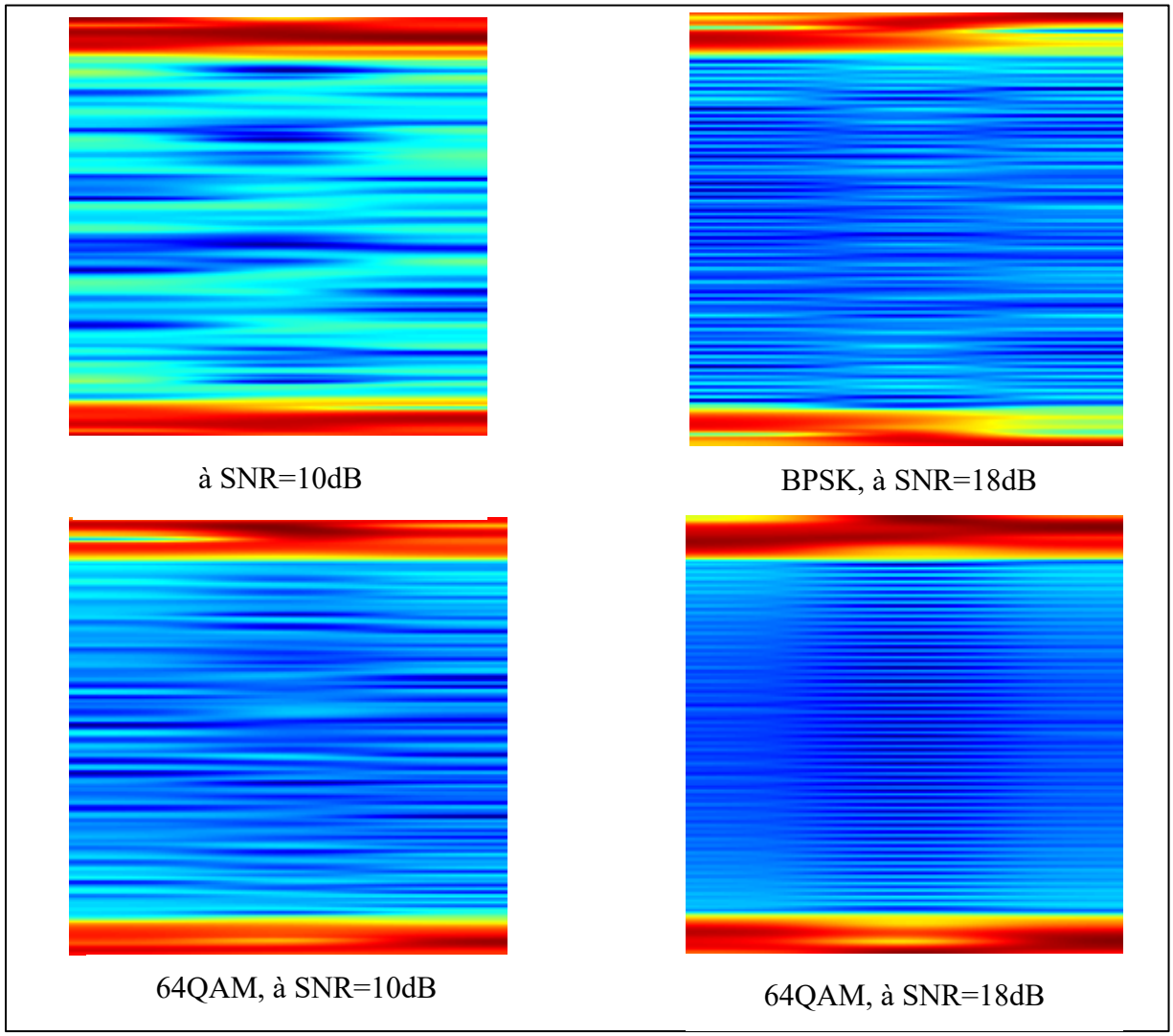


Figure III:2: Exemples de spectrogrammes des signaux de la base RML2016a

III.2.2. **RML2016a en format image en utilisant la transformée de Wigner-Ville (WVD).**

La transformée de Wigner-Ville permet de transformer chaque signal complexe I/Q de 128 échantillons de la base RML2016a en représentation temps fréquence plus précise que le spectrogramme, notamment pour les signaux modulés. Elle est définie pour un signal $x(t)$ par :

$$W_x(t,f) = \int x\left(t+\frac{\tau}{2}\right) x^*\left(t-\frac{\tau}{2}\right) e^{-j2\pi f\tau}\, d\tau$$

Les étapes de conversion de la base RML2016a.mat vers une base en images WVD sont similaires à ceux du paragraphe 2.1 à l'exception de l'utilisation de la fonction wvd de Matlab au lieu de la fonction spectrogram. La hiérarchie des dossiers de la base obtenue est aussi similaire à celle du paragraphe 2.1. La figure III.3 quatre exemples de représentation WVD de signaux , pour les deux modulations BPSK et 64-QAM et pour deux valeurs du rapport SNR 10 dB et 18 dB.

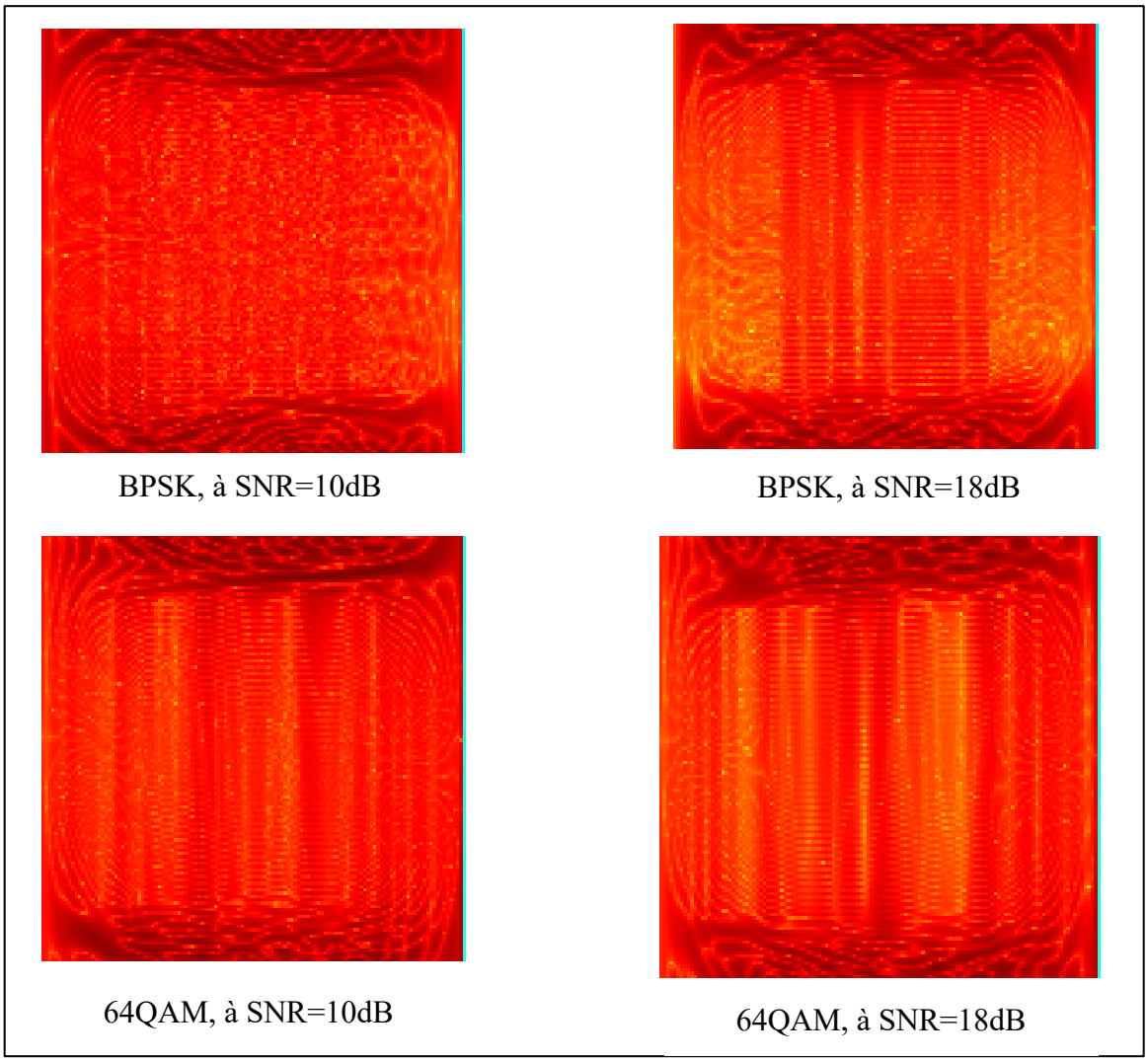


Figure III:3: Exemples des images de transformée WVD des signaux de la base RML2016a

Modèle de réseaux utilisant la représentation fréquentielle :

III.2.3. **Classification par un modèle CNN proposé :**

Après plusieurs essais réalisés sur différentes architectures de réseaux de convolution, nous avons réussi à concevoir un modèle à complexité réduite tout en offrant des performances de

classification satisfaisantes. Ce modèle repose sur trois blocs convolutionnels successifs suivis d'une couche de dropout, d'une couche entièrement connectée ainsi que des couches de classification classiques (softmax et classification). Ce choix permet de trouver un équilibre entre la profondeur du modèle, la capacité de généralisation ainsi que la rapidité de calcul, ce qui est particulièrement adapté à la reconnaissance de la modulation à partir d'images de taille 128×128×3.

Le logiciel MATLAB 2024b a été utilisé pour implémenter ce modèle grâce à son environnement intégré, idéal pour le prototypage rapide et la création de modèles de deep learning spécifiques aux télécommunications. Il fournit des outils adaptés à la préparation des signaux, à la construction du modèle ainsi qu'au suivi de l'entraînement. De plus, les nombreuses fonctions de visualisation simplifient la validation du modèle.

Le programme Matlab contient les étapes suivantes :

1. **Génération du Datastore et répartition :** Nous avons commencé par la génération d'un datastore des différentes images de la base de données pour chaque valeur du SNR allant de 0dB à 18dB. Un **datastore d'images** est une structure de MATLAB permettant de gérer facilement de grandes quantités d'images sans les charger entièrement en mémoire. Il organise les chemins, les labels et fournit des mécanismes de lecture par lot adaptés à l'entraînement de modèles. On l'utilise pour simplifier le traitement des données visuelles tout en optimisant la mémoire et les performances. Pour chaque SNR nous avons séparé les images en **Train** (70%), **Validation** (10%) et **Test** (20%)
2. **Construction du modèle CNN :** Le tableau III.1 donne les caractéristiques de chaque couche du réseau pour avoir une idée détaillée sur la complexité du modèle.

Tableau III:1: Architecture du premier réseau proposé pour la classification automatique de la modulation

Couche	Entrée	Sortie	Nombre de paramètres
imageInputLayer	128×128×3	128×128×3	0
convolution2dLayer 1	128×128×3	128×128×32	(5×5×3)×32 + 32 = 2 432
batchNormalization 1	128×128×32	128×128×32	128
relu 1	128×128×32	128×128×32	0
maxPooling 1	128×128×32	64×64×32	0
convolution2dLayer 2	64×64×32	64×64×64	(3×3×32)×64 + 64 = 18 496
batchNormalization 2	64×64×64	64×64×64	256

Couche	Entrée	Sortie	Nombre de paramètres
relu 2	64×64×64	64×64×64	0
maxPooling 2	64×64×64	32×32×64	0
convolution2dLayer 3	32×32×64	32×32×128	(3×3×64)×128 + 128 = 73 856
batchNormalization 3	32×32×128	32×32×128	512
relu 3	32×32×128	32×32×128	0
maxPooling 3	32×32×128	16×16×128	0
dropout	16×16×128	16×16×128	0
fullyConnected 1	16×16×128	256	(16×16×128)×256 + 256 = 8 388 864
relu 4	256	256	0
fullyConnected out	256	11	(256×11) + 11 = 2 827
softmax	11	11	0
classification	11	11	0

3. **Choix des options d'apprentissage** : le tableau III.2 résumes les options choisis pour l'apprentissage pour assurer un temps d'apprentissage raisonnable sans perdre la justesse de la classification.

Tableau III:2: Les options d'apprentissage du réseau.

Paramètre	Valeur
Optimiseur	Adam
Taux d'apprentissage	1e-4
Époques	10
Taille du mini-batch	64
Fréquence de validation	Toutes les 120 itérations
Affichage du progrès	training-progress
Verbose	false

4. **Entraînement du modèle :** Le modèle CNN est entraîné à partir des données d'entraînement définies.
5. **Evaluation** Pour chaque SNR, on applique le modèle entraîné aux images de test, on calcule la précision pour chaque niveau de bruit (SNR) et on génère une matrice de confusion pour analyser visuellement la performance.

III.2.3.1. **Résultats de classification du premier modèle proposé appliqué à des image WVD**

La figure III.4 illustre l'évolution du processus d'entraînement du modèle en termes de justesse (accuracy) ainsi que de la fonction de perte. Elle met en évidence qu'au terme de l'apprentissage, la justesse globale de validation, calculée sur toutes les valeurs de SNR ainsi que pour les 11 classes de modulation de la base RML2016a, atteint 81,99 %. Ce résultat est particulièrement satisfaisant compte tenu de la complexité du jeu de données utilisé. Par ailleurs, l'entraînement du modèle s'avère extrêmement coûteux en termes de ressources de calcul, nécessitant environ 655 minutes (près de 11 heures) pour atteindre la convergence. Le graphique montre également que les courbes de justesse pour l'entraînement et la validation sont très proches tout au long du processus, suggérant ainsi une généralisation maîtrisée du modèle. Ce faible écart traduit l'absence manifeste de sur-apprentissage (overfitting). Le graphe de la fonction perte (loss) atteint une valeur légèrement inférieure à 0,5, cela signifie que l'erreur de classification est maîtrisée, indiquant ainsi que le modèle est capable de généraliser efficacement sans sur-apprentissage majeur.

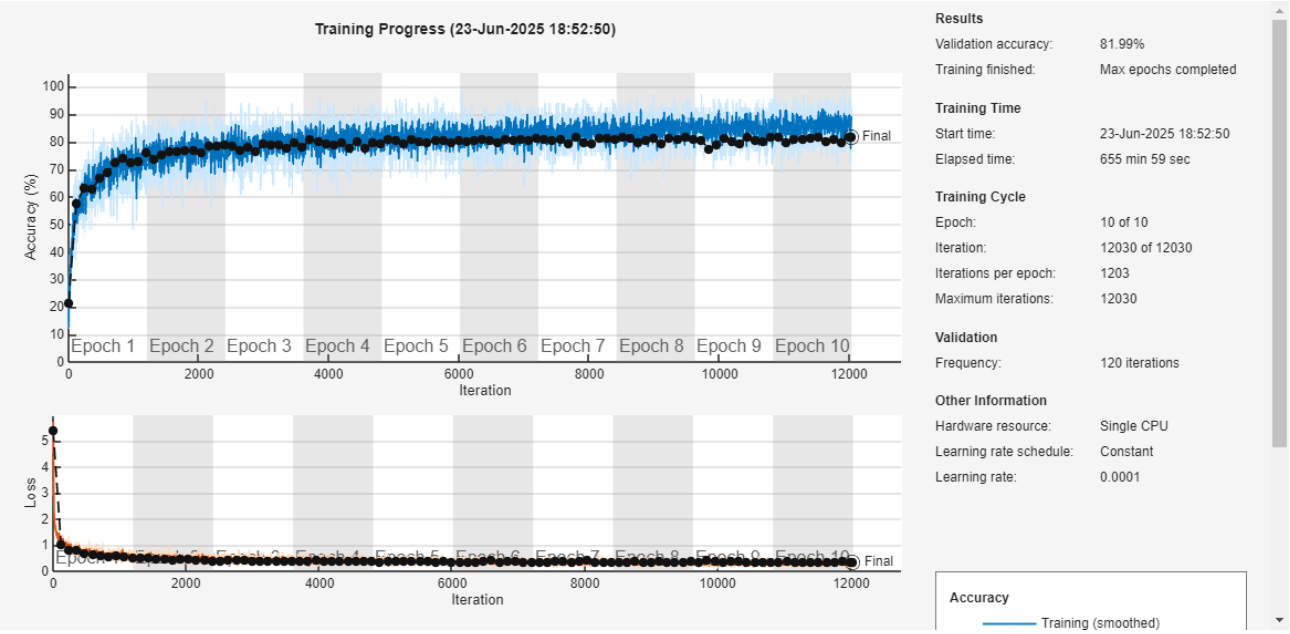


Figure III:4: Evolution de l'entrainement du premier modèle proposé appliqué à des image WVD de la base RML2016a

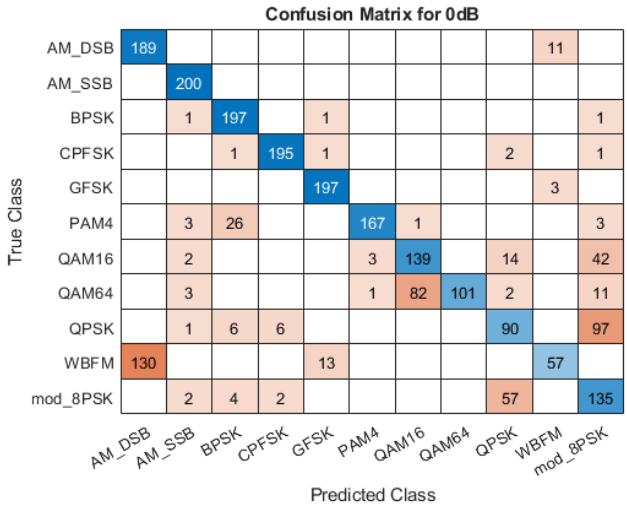
Après la phase d'apprentissage du modèle, une évaluation détaillée est effectuée en calculant la justesse (accuracy) pour chaque valeur de SNR afin de mesurer la robustesse du classificateur selon le niveau de bruit.

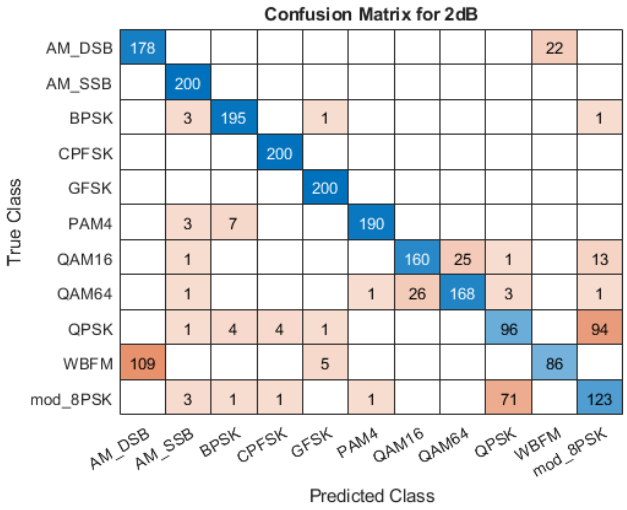
Le tableau III.3 présente la justesse de classification en fonction des différentes valeurs de SNR. L'analyse des résultats permet de constater que, pour des niveaux de SNR supérieurs ou égaux à 10 dB, la précision du modèle atteint environ 84 %, illustrant ainsi une bonne capacité de reconnaissance des différentes classes de modulation. Pour des valeurs de SNR inférieures à 10 dB, la précision reste globalement satisfaisante, oscillant entre 75 % et 82 %. Ce constat souligne la robustesse du modèle proposé, capable de maintenir des performances élevées malgré des conditions de bruit plus défavorables.

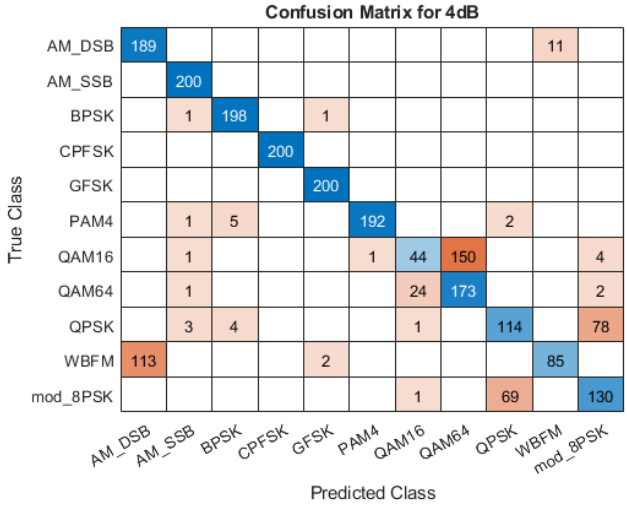
Tableau III:3: Justesse de test du premier modèle proposé appliqué aux images de spectrogramme pour les valeurs positifs du SNR

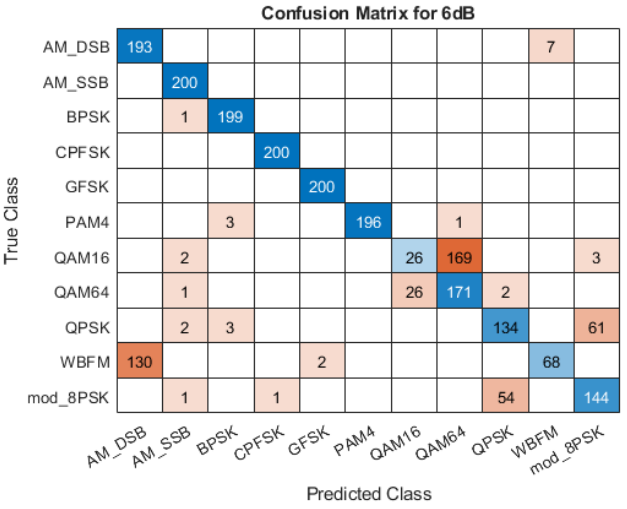
SNR	Accuracy
{'0dB' }	0.75773
{'10dB'}	0.84182
{'12dB'}	0.84818
{'14dB'}	0.84091
{'16dB'}	0.84091
{'18dB'}	0.84045
{'2dB' }	0.81636
{'4dB' }	0.78409
{'6dB' }	0.78682
{'8dB' }	0.82

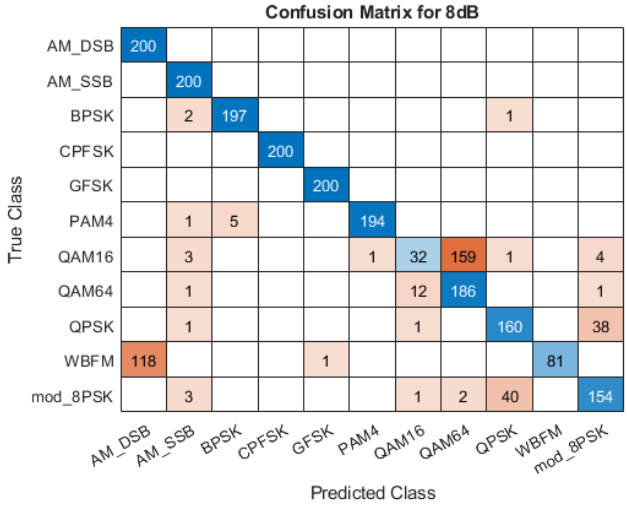
Pour affiner l'analyse des performances, des matrices de confusion spécifiques à chaque SNR ont été générées, offrant ainsi une visualisation précise des taux de reconnaissance pour toutes les classes de modulation. Ces matrices permettent d'identifier facilement les schémas de modulation qui sont correctement classés ainsi que ceux qui posent des difficultés, révélant ainsi les confusions fréquentes entre certaines classes. Ce niveau de détail est crucial pour comprendre les faiblesses du modèle et envisager des améliorations spécifiques. La figure III.5 donne les matrices de confusion obtenues pour les 10 valeurs de SNR (allant de 0dB à 18dB avec pas de 2)

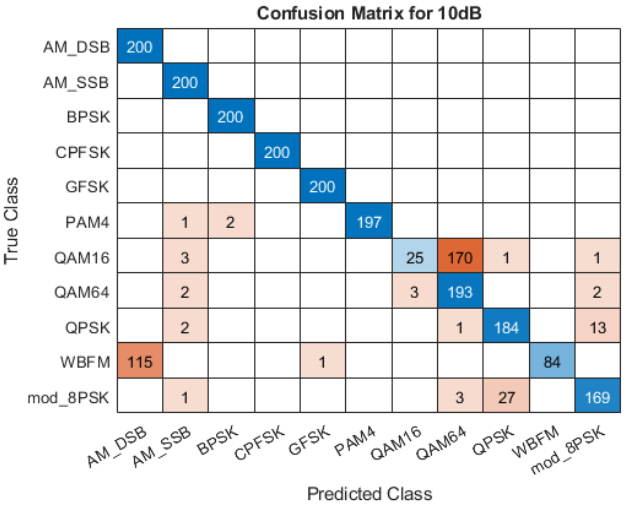

Confusion Matrix for 0dB
AM_DSB 189 11
AM_SSB 200
BPSK 1 197 1 1
CPFSK 1 195 1 2 1
GFSK 197 3
PAM4 3 26 167 1 3
QAM16 2 3 139 14 42
QAM64 3 1 82 101 2 11
QPSK 1 6 6 90 97
WBFM 130 13 57
mod_8PSK 2 4 2 57 135
AM_DSB AM_SSB BPSK CPFSK GFSK PAM4 QAM16 QAM64 QPSK WBFM mod_8PSK
True Class
Predicted Class


Confusion Matrix for 2dB
AM_DSB 178 22
AM_SSB 200
BPSK 3 195 1 1
CPFSK 200
GFSK 200
PAM4 3 7 190
QAM16 1 160 25 1 13
QAM64 1 1 26 168 3 1
QPSK 1 4 4 1 96 94
WBFM 109 5 86
mod_8PSK 3 1 1 1 71 123
AM_DSB AM_SSB BPSK CPFSK GFSK PAM4 QAM16 QAM64 QPSK WBFM mod_8PSK
True Class
Predicted Class


Confusion Matrix for 4dB
AM_DSB 189 11
AM_SSB 200
BPSK 1 198 1
CPFSK 200
GFSK 200
PAM4 1 5 192 2
QAM16 1 1 44 150 4
QAM64 1 24 173 2
QPSK 3 4 1 114 78
WBFM 113 2 85
mod_8PSK 1 69 130
AM_DSB AM_SSB BPSK CPFSK GFSK PAM4 QAM16 QAM64 QPSK WBFM mod_8PSK
True Class
Predicted Class


Confusion Matrix for 6dB
AM_DSB 193 7
AM_SSB 200
BPSK 1 199
CPFSK 200
GFSK 200
PAM4 3 196 1
QAM16 2 26 169 3
QAM64 1 26 171 2
QPSK 2 3 134 61
WBFM 130 2 68
mod_8PSK 1 1 54 144
AM_DSB AM_SSB BPSK CPFSK GFSK PAM4 QAM16 QAM64 QPSK WBFM mod_8PSK
True Class
Predicted Class


Confusion Matrix for 8dB
AM_DSB 200
AM_SSB 200
BPSK 2 197 1
CPFSK 200
GFSK 200
PAM4 1 5 194
QAM16 3 1 32 159 1 4
QAM64 1 12 186 1
QPSK 1 1 160 38
WBFM 118 1 81
mod_8PSK 3 1 2 40 154
AM_DSB AM_SSB BPSK CPFSK GFSK PAM4 QAM16 QAM64 QPSK WBFM mod_8PSK
True Class
Predicted Class


Confusion Matrix for 10dB
AM_DSB 200
AM_SSB 200
BPSK 200
CPFSK 200
GFSK 200
PAM4 1 2 197
QAM16 3 25 170 1 1
QAM64 2 3 193 2
QPSK 2 1 184 13
WBFM 115 1 84
mod_8PSK 1 3 27 169
AM_DSB AM_SSB BPSK CPFSK GFSK PAM4 QAM16 QAM64 QPSK WBFM mod_8PSK
True Class
Predicted Class

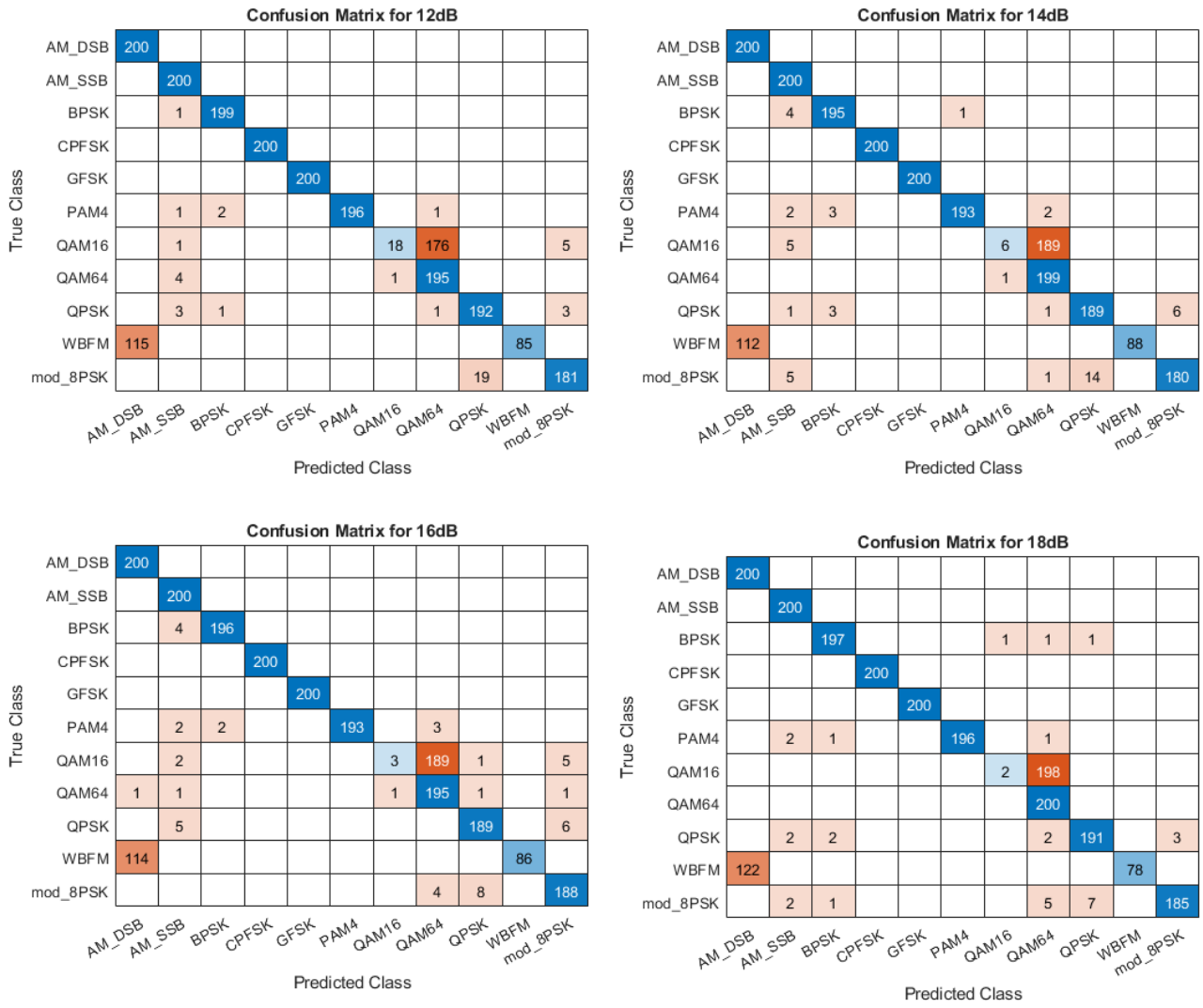


Figure III:5: Matrices de confusion pour les différentes valeurs du SNR obtenu par le premier modèle proposé appliqué à des images WVD.

À partir des matrices de confusion de la Figure III.5, on peut dégager les observations suivantes :

- Les modulations **AM_SSB**, **BPSK**, **CPFSK** et **GFSK** présentent des performances de classification remarquables, avec des taux de justesse situés entre **97,5 % et 100 %** pour toutes les valeurs positives du **SNR**.
- La modulation **AM_DSB** bénéficie d'une précision globalement élevée, variant de **89 % à 100 %**, avec des taux de classification excellents à partir de **4 dB**.

- Le schéma **PAM4** fournit des résultats très satisfaisants, atteignant des taux de justesse compris entre **95 % et 100 %** à des valeurs de **SNR** supérieures à **0 dB**.
- Pour la modulation **QAM16**, le taux de justesse est particulièrement faible à bas **SNR**, et reste limité même à des valeurs plus élevées, illustrant sa sensibilité au bruit ainsi que la difficulté du modèle à la distinguer correctement.
- La modulation **QAM64** connaît des taux de précision faibles à **0 dB** et **2 dB**, mais elle bénéficie d'une amélioration notable à partir de **4 dB**, atteignant d'excellentes performances à partir de **10 dB**.
- Le schéma **QPSK** est affecté par des taux de précision limités à bas **SNR**, mais s'améliore nettement à mesure que le **SNR** devient supérieur à **10 dB**, où elle atteint des taux de reconnaissance élevés.
- Le schéma **WBFM** est plus difficile à identifier correctement, avec des taux de justesse faibles, généralement compris entre **28,5 %** et **44 %**, quelle que soit la valeur du **SNR**.
- Enfin, la modulation **8PSK** fournit des taux de précision modestes à bas **SNR**, mais bénéficie d'une forte amélioration à partir de **12 dB**, atteignant des taux supérieurs à **90 %**.

Ces résultats illustrent la manière dont la robustesse du modèle dépend du schéma de modulation ainsi que du niveau de bruit du canal. Les schémas à faible complexité spectrale (AM, BPSK, CPFSK, GFSK) demeurent facilement reconnaissables à tout niveau de **SNR**, tandis que les schémas à forte densité de constellation (QAM16, QAM64) ainsi que le schéma **WBFM** présentent des taux de reconnaissance plus faibles, requérant des **SNR** plus élevés pour atteindre des performances acceptables.

III.2.3.2. **Résultats de classification du premier modèle proposé appliqué à des images de spectrogramme**

La figure III.6 illustre l'évolution du processus d'entraînement du modèle en termes de justesse (accuracy) ainsi que de la fonction de perte. On remarque que la justesse globale de validation, atteint 68,86 %. Ce qui représente une dégradation de plus de 10% par rapport à la justesse obtenue pour des images de la distribution avec un temps d'apprentissage plus élevé (1070 mn ou environ 18 heures). Le graphique d'apprentissage montre un décalage remarquable de la

courbe de justesse d'apprentissage par rapport à la courbe de justesse d'évaluation à partir de l'époch 9 ce qui indique la présence de sur-apprentissage (overfitting).

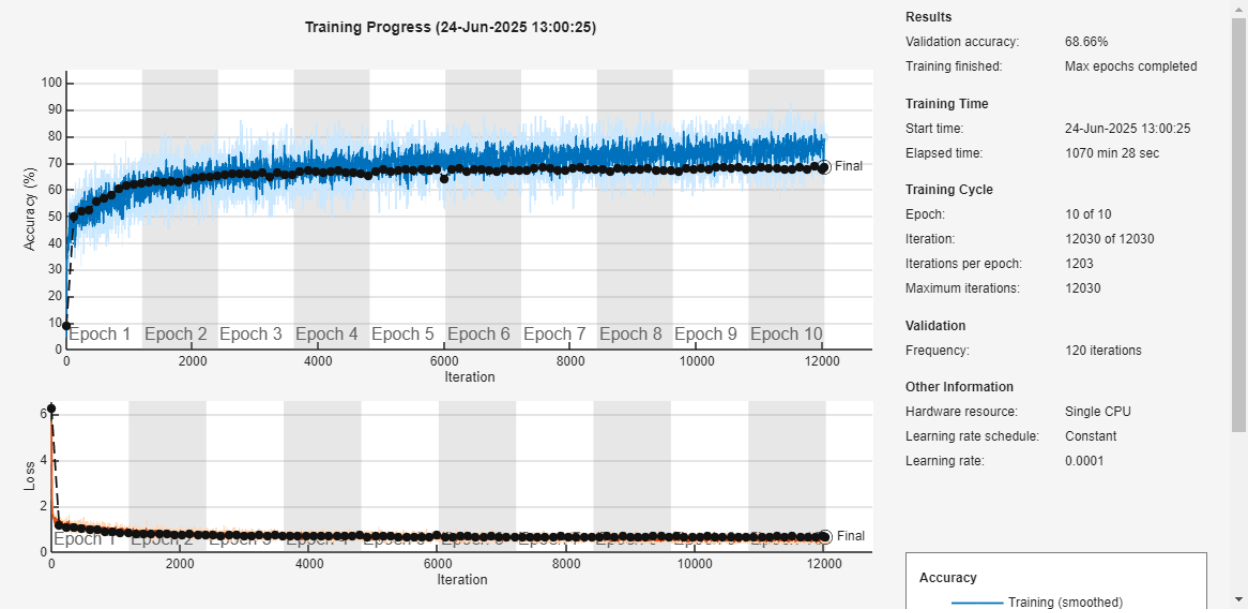


Figure III:6: Evolution de l'entrainement du premier modèle proposé appliqué à des images de spectrogramme de la base RML2016a.

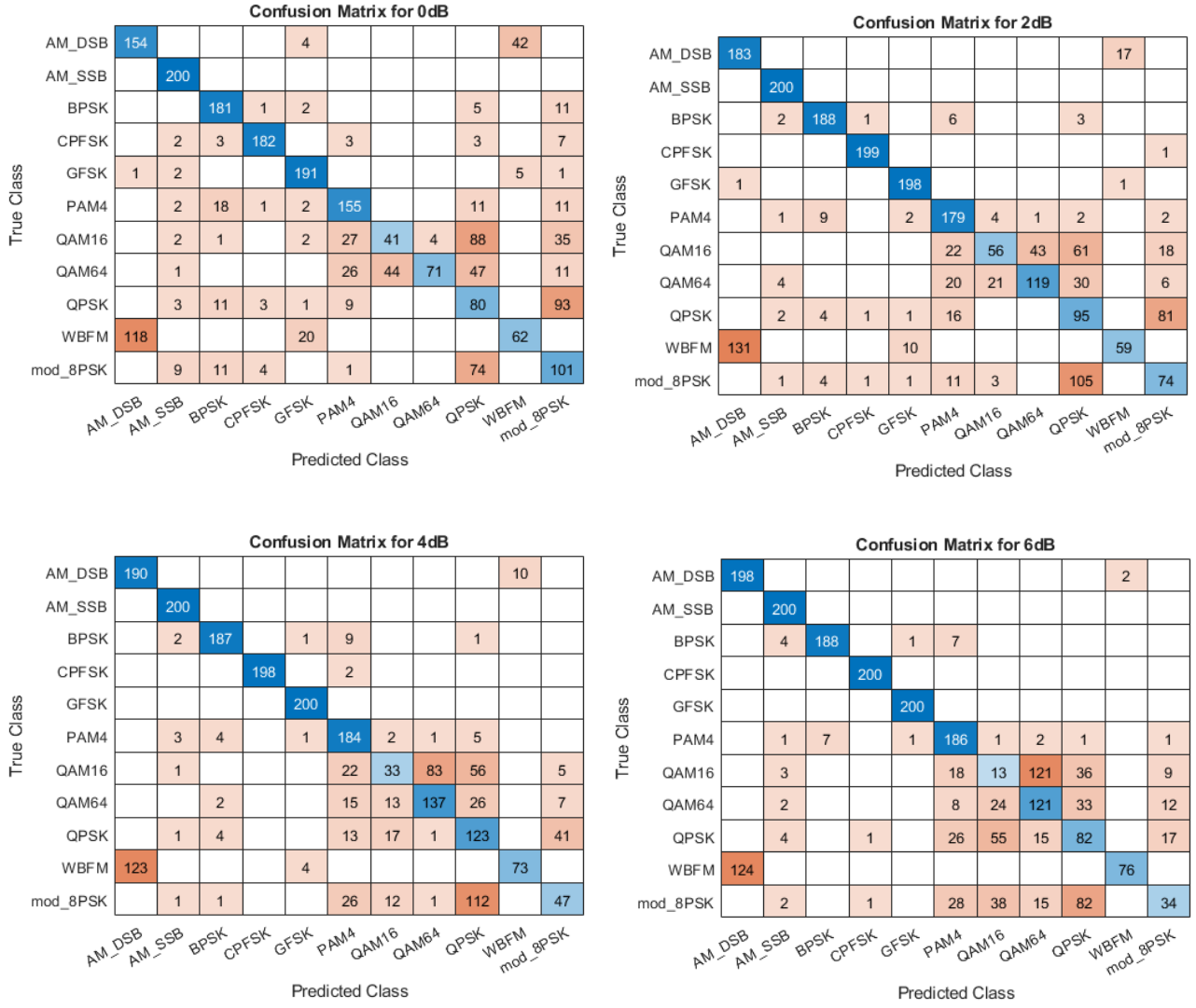
Le tableau III.4 présente la justesse de classification en fonction des différentes valeurs de SNR pour le premier modèle proposé appliqué aux images de spectrogramme.

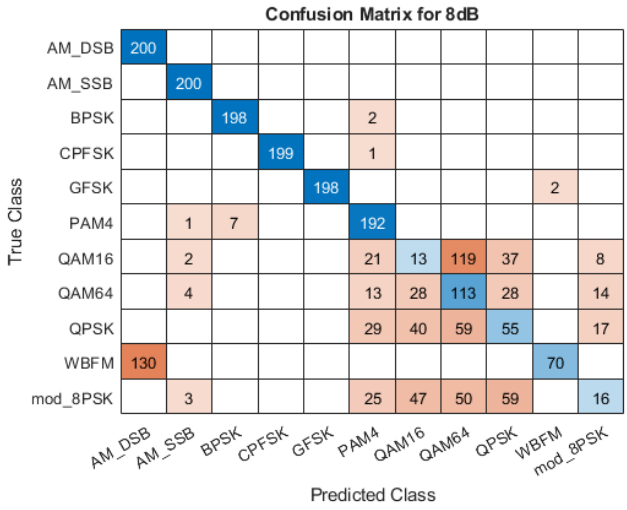
Tableau III:4: justesse de test du premier modèle proposé appliqué aux images de spectrogramme pour les valeurs positifs du SNR

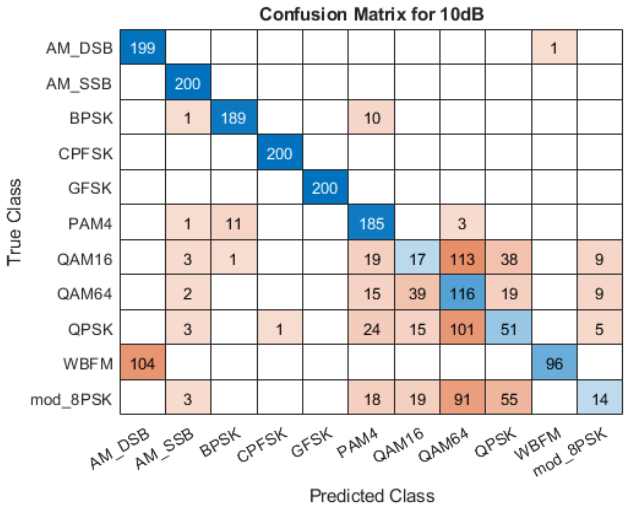
SNR	Accuracy
{'0dB' }	0.64455
{'10dB'}	0.66682
{'12dB'}	0.68182
{'14dB'}	0.68955
{'16dB'}	0.74
{'18dB'}	0.725
{'2dB' }	0.70455
{'4dB' }	0.71455
{'6dB' }	0.68091
{'8dB' }	0.66091

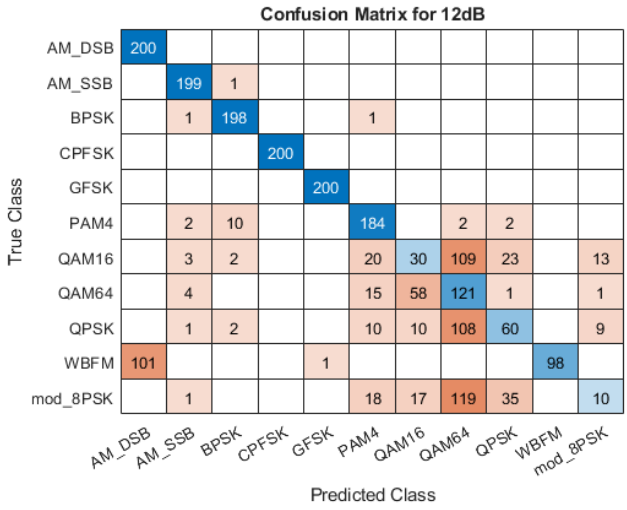
Le tableau montre que la précision du test varie de 64 % pour un SNR de 0 dB à 74 % pour un SNR de 16 dB. Ce résultat est inférieur d'environ 10 % à la précision obtenue avec des images WVD. Cette comparaison suggère que le premier modèle proposé est plus performant sur des images WVD que sur des spectrogrammes.

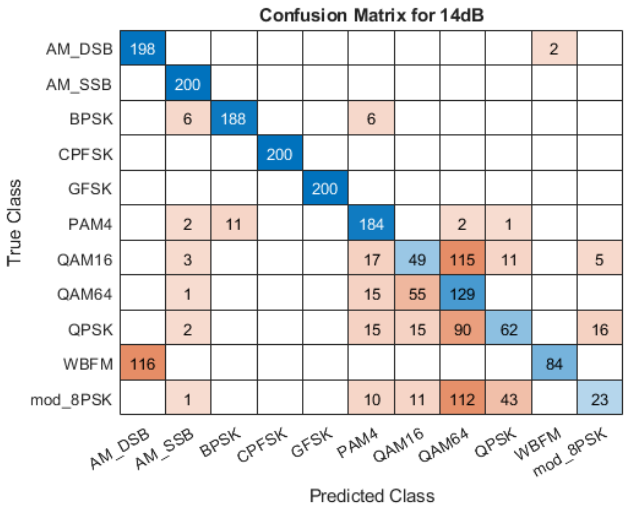
La figure III.7 présente les matrices de confusion obtenues pour les 10 niveaux de SNR considérés, allant de 0 dB à 18 dB par pas de 2 dB.




Confusion Matrix for 8dB
True Class
Predicted Class
AM_DSB
AM_SSB
BPSK
CPFSK
GFSK
PAM4
QAM16
QAM64
QPSK
WBFM
mod_8PSK


Confusion Matrix for 10dB
True Class
Predicted Class
AM_DSB
AM_SSB
BPSK
CPFSK
GFSK
PAM4
QAM16
QAM64
QPSK
WBFM
mod_8PSK


Confusion Matrix for 12dB
True Class
Predicted Class
AM_DSB
AM_SSB
BPSK
CPFSK
GFSK
PAM4
QAM16
QAM64
QPSK
WBFM
mod_8PSK


Confusion Matrix for 14dB
True Class
Predicted Class
AM_DSB
AM_SSB
BPSK
CPFSK
GFSK
PAM4
QAM16
QAM64
QPSK
WBFM
mod_8PSK

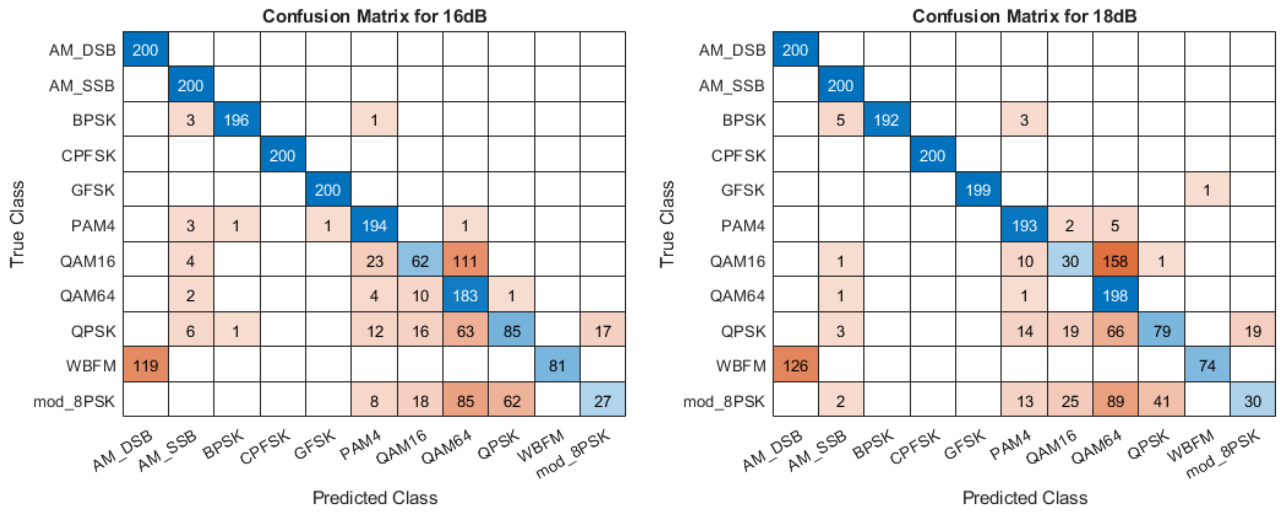


Figure III:7: Matrices de confusion pour les différentes valeurs du SNR obtenu par le premier modèle proposé appliqué à des images de spectrogramme.

À partir des matrices de confusion de la Figure III.7, on peut dégager les observations suivantes :

- **Les modulations AM_SSB** et **GFSK** obtiennent des taux de classification extrêmement élevés (de 95.5 % à 100 %) pour toutes les valeurs de SNR considérées, illustrant la robustesse du modèle vis-à-vis du bruit.
- **Les modulations CPFSK et BPSK** donnent un excellent taux de justesse à partir de SNR 2dB.
- **AM_DSB** atteint des taux de classification élevés, passant de **77 % à 0 dB** à presque **100 %** à partir de 8 dB.
- **PAM4** s'améliore de manière significative, de **155 (77,5 %) à 0 dB** à environ **95–97 % à haut SNR**.
- **PSK** est globalement bien classée (plus de 90 %) à tous les niveaux de SNR, avec des taux variant de **90,5 % à 0 dB** à **95–99 % à haut SNR**.
- **QAM64** bénéficie de la hausse forte du SNR : elle donne un taux supérieur à 90% pour 16dB et 18 dB uniquement.
- Le taux de justesse **de QAM16** reste extrêmement faible pour toutes les valeurs du SNR: elle débute à **20,5 %) à 0 dB**, tombe à des taux extrêmement bas à SNR intermédiaires

de 13 à 30%, puis remonte à environ **30–31 % à haut SNR**, ce qui suggère des difficultés à isoler visuellement ce type de modulation à l'aide du spectrogramme.

- **QPSK** bénéficie peu de la hausse du SNR, avec des taux de classification faibles et irréguliers : de **40 % à 0 dB** à **30–40 % à haut SNR**.
- **WBFM** reste limité, avec des taux variant de **31 % à 0 dB** à environ **37–48 % à haut SNR**.
- **8PSK** est extrêmement affecté par le bruit, avec des taux tombant à **7–8 %** à moyen SNR, puis légèrement améliorés à haut SNR 13.5%–15%, sans jamais atteindre des niveaux acceptables.

III.2.4. **Classification par l'utilisation du "transfert learning"**

Le « transfert learning » est une technique d'apprentissage automatique consistant à réutiliser un modèle pré-entraîné sur une grande base de données pour une nouvelle tâche similaire. Le principe repose sur l'idée que les premières couches du réseau capturent des caractéristiques générales (bords, textures…) utiles à plusieurs problèmes. Cette méthode permet de réduire le temps d'entraînement et les besoins en données annotées.

Dans notre travail, nous avons exploré l'intérêt de cette technique pour la classification automatique des modulations numériques. Nous avons utilisé le réseau *SqueezeNet* comme modèle de base. Ce choix s'explique par sa légèreté architecturale par rapport à d'autre réseau ce qui permet un entraînement rapide avec de bonnes performances de classification, même sur des machines avec des ressources limitées.

Le tableau III.5 résume l'architecture de SqueezeNet avec, pour chaque couche, son nom, les dimensions d'entrée et de sortie, ainsi que le nombre de paramètres entrainables [33].

Tableau III:5: caractéristiques les différentes couches du réseaux SqueezeNet.

N°	Nom de la couche	Dimension d'entrée	Dimension de sortie	Nombre de paramètres
1	`conv1`	224×224×3	111×111×96	14208
2	`Relu1`	111×111×96	111×111×96	0
3	`maxpool1`	111×111×96	55×55×96	0
4	`fire2`	55×55×96	55×55×128	11 920
5	`fire3`	55×55×128	55×55×128	12 432

N°	Nom de la couche	Dimension d'entrée	Dimension de sortie	Nombre de paramètres
7	fire4	55×55×128	55×55×256	45.433
	Maxpool4	55×55×256	27×27×256	0
8	fire5	27×27×256	27×27×256	49 440
10	fire6	27×27×256	27×27×384	104 880
11	fire7	27×27×384	27×27×384	111 024
12	fire8	27×27×384	27×27×512	188 992
	Maxpool8	27×27×512	13×12×512	0
13	fire9	13×13×512	13×13×512	197 184
14	conv10	13×13×512	13×13×1000	513.000
15	avgpool10	13×13×1000	1×1×1000	0
16	softmax	1×1×1000	1×1×1000	0

III.2.4.1. **Résultats de classification du SqueezeNet appliqué à des images WVD**

Les étapes de programmation du réseau SqueezeNet sont similaire à ceux expliqués au début du paragraphe 3.1 sauf dans l'étape de détermination de l'architecture du réseau où nous avons utilisé le transfert Learning pour construire un modèle de classification comme suivant :

- Chargement du modèle SqueezeNet pré-entraîné sur ImageNet.
- Suppression des couches de classification finales (de la couche conv10 à la couche ClassificationLayer_predictions).
- Ajout de nouvelles couches adaptées à notre application qui consiste à classer les 11 types de modulation.
- Comme le montre la figure III.8, les résultats obtenus lors de la phase d'entraînement avec SqueezeNet sont très similaires à ceux du premier réseau proposé (figure III.4). Toutefois, SqueezeNet nécessite un temps d'apprentissage nettement plus long, avec 1575 minutes contre 655 minutes pour le premier modèle, en raison d'une architecture plus complexe et plus profonde

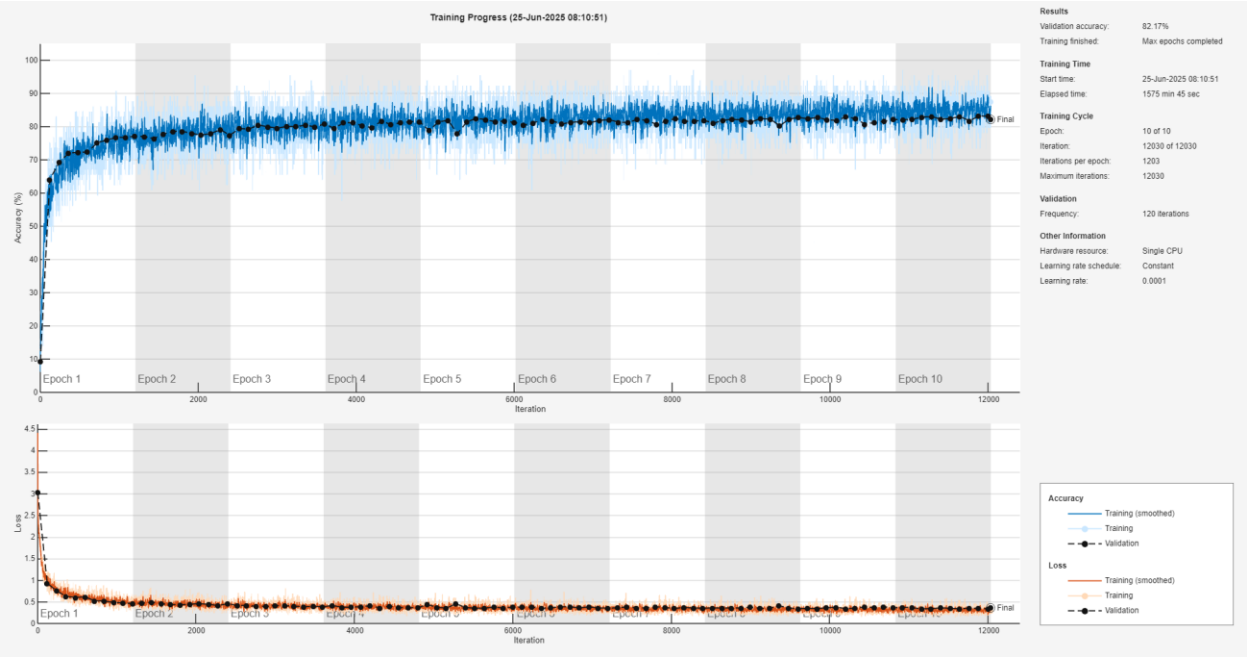


Figure III:8: Evolution de l'entrainement du réseau SqueezeNet appliqué à des image WVD de la base RML2016a.

Le tableau III.6 présente la précision de classification en fonction des différentes valeurs de SNR. L'analyse des résultats montre que, pour des niveaux de SNR supérieurs ou égaux à 12 dB, le modèle atteint une précision d'environ 85 %. Entre 2 dB et 10 dB, la précision oscille entre 80 % et 84 %. En revanche, pour un SNR de 0 dB, le modèle affiche une performance relativement plus faible, avec une précision d'environ 73 %.

Tableau III:6: Justesse de test du réseau SqueezeNet appliqué aux images WVD pour les valeurs positives du SNR

Final Results across all SNRs:

SNR	Accuracy
{'0dB' }	0.73136
{'10dB'}	0.84091
{'12dB'}	0.85227
{'14dB'}	0.85864
{'16dB'}	0.85364
{'18dB'}	0.84909
{'2dB' }	0.81773
{'4dB' }	0.8
{'6dB' }	0.80136
{'8dB' }	0.81773

Les matrices de confusion obtenues avec le modèle SqueezeNet pour les différentes valeurs de SNR présentent une forte similarité avec celles du modèle proposé, illustrées dans la figure III.5. Afin d'éviter des redondances, nous avons choisi de ne pas reproduire l'ensemble des matrices de confusion. À la place, le tableau III.7 présente, pour chaque valeur de SNR, le nombre de classifications correctes obtenues par les deux modèles, ce qui permet une comparaison plus directe et synthétique.

Dans ce tableau, les colonnes numérotées 1 indiquent les performances du modèle proposé, tandis que les colonnes numérotées 2 correspondent aux résultats du réseau SqueezeNet.

Tableau III:7: Comparaison entre le réseau proposé et le réseau SqueezeNet en termes de précision de classification par classe de modulation et pour chaque valeur de SNR.

	0dB		*2dB*		*4dB*		*6dB*		*8dB*	
	1	2	1	2	1	2	1	2	1	2
AM_DSB	189	188	178	198	189	200	193	200	200	199
AM_SSB	200	200	200	200	200	200	200	200	200	200
BPSK	197	197	195	199	198	199	199	199	197	198
CPFSK	195	192	200	200	200	200	200	200	200	200
GFSK	197	197	200	200	200	200	200	200	200	200
PAM4	167	180	190	189	192	195	196	198	194	200
QAM16	139	109	160	143	44	70	26	43	32	37
QAM64	101	72	168	144	173	170	171	166	186	170
QPSK	90	63	96	109	114	126	134	149	160	160
WBFM	57	60	86	74	85	66	68	70	81	84
8 PSK	135	151	123	143	130	134	144	138	154	151

	10dB		*12dB*		*14dB*		*16dB*		*18dB*	
	1	2	1	2	1	2	1	2	1	2
AM_DSB	200	196	200	195	200	190	200	196	200	199
AM_SSB	200	200	200	200	200	200	200	200	200	200
BPSK	200	197	199	198	195	196	196	194	197	194
CPFSK	200	200	200	200	200	200	200	200	200	200
GFSK	200	200	200	200	200	200	200	200	200	200
PAM4	197	199	196	199	193	195	193	195	196	194
QAM16	25	43	18	52	6	51	3	41	2	41

QAM64	193	165	195	169	199	183	195	188	200	185
QPSK	184	176	192	176	189	190	189	193	191	185
WBFM	84	90	85	99	88	96	86	79	78	82
8 PSK	169	184	181	187	180	188	188	192	183	188

Basé sur le tableau III.7 on peut mentionner les remarques suivantes :

- **Les modulation AM_SSB, CPFSK, GFSK, BPSK** : présentent une justesse très élevée (près de 100 %) pour toutes les valeurs de SNR, sans différence significative entre les deux modèles.
- **La modulation AM_DSB** : présente une légère supériorité du modèle SqueezeNet aux SNR faible contre une légère supériorité du modèle proposé à partir de SNR ≥ 12 dB.
- **Pour la modulation PAM4** : Les performances de SqueezeNet sont légèrement supérieures pour la majorité des SNR.
- **Pour 8PSK** : le réseau SqueezeNet dépasse légèrement le modèle proposé sur l'ensemble des SNR.
- **Pour QAM16** : Le modèle proposé montre de meilleures performances à bas SNR (sauf exceptions ponctuelles), bien que les deux réseaux présentent une précision faible globalement.
- **QAM64** : Le modèle proposé surpasse généralement SqueezeNet à **SNR ≥ 10 dB**.
- Pour QPSK les deux réseaux donne sont presque équivalent.
- La modulation WBFM montre des performances faibles sur les deux modèles avec une légère supériorité de SqueezeNet à partir de 6 dB.

III.3. **Modèle de réseaux utilisant la représentation I.Q**

Pour les données au format I/Q, le réseau proposé est constitué de cinq couches de convolution. Ce choix résulte d'une étude comparative de plusieurs configurations comportant 3, 4 et 6 couches, dont les performances sont présentées dans le tableau III.8. Le modèle à 5 couches s'est avéré offrir le meilleur compromis entre complexité et précision.

Tableau III:8: Justesse d'apprentissage des réseaux essayés avec la Base RML2016.10a.

Nombre de couche du réseau	Justesse d'apprentissage
3 couches de convolution	79.92%
4 couches de convolution	79.75%

5 couches de convolution	**83.85%**
6 couches de convolution	82.87%
SqueezeNet	63.82%

Le tableau III.9 donne les caractéristiques de chacune des couches du réseau proposé pour des donné en format IQ.

Tableau III:9: Caractéristiques les différentes couches du réseau proposé pour les données IQ.

N°	Couche	Dimension d'entrée	Dimension de sortie	Nombre de paramètres
1	imageInputLayer([128,2,1])	128×2×1	128×2×1	0
2	conv2d([5,2],16)	128×2×1	124×1×16	(5×2×1)×16 = 160
3	batchNormalizationLayer	124×1×16	124×1×16	2×16 = 32
4	reluLayer	124×1×16	124×1×16	0
5	maxPooling([2,1],Stride=[2,1])	124×1×16	62×1×16	0
6	conv2d([5,1],32)	62×1×16	58×1×32	(5×1×16)×32 = 2,560
7	batchNormalizationLayer	58×1×32	58×1×32	2×32 = 64
8	reluLayer	58×1×32	58×1×32	0
9	maxPooling([2,1],Stride=[2,1])	58×1×32	29×1×32	0
10	conv2d([3,1],64)	29×1×32	27×1×64	(3×1×32)×64 = 6,144
11	batchNormalizationLayer	27×1×64	27×1×64	2×64 = 128
12	reluLayer	27×1×64	27×1×64	0
13	maxPooling([2,1],Stride=[2,1])	27×1×64	13×1×64	0
14	conv2d([3,1],128)	13×1×64	11×1×128	(3×1×64)×128 = 24,576
15	batchNormalizationLayer	11×1×128	11×1×128	2×128 = 256
16	reluLayer	11×1×128	11×1×128	0
17	maxPooling([2,1],Stride=[2,1])	11×1×128	5×1×128	0
18	conv2d([3,1],256)	5×1×128	3×1×256	(3×1×128)×256 = 98,304
19	batchNormalizationLayer	3×1×256	3×1×256	2×256 = 512
20	reluLayer	3×1×256	3×1×256	0
21	maxPooling([2,1],Stride=[2,1])	3×1×256	1×1×256	0
22	fullyConnectedLayer(512)	256	512	256×512 + 512 = 131,584
23	batchNormalizationLayer	512	512	2×512 = 1,024
24	reluLayer	512	512	0
25	dropout(0.5)	512	512	0
26	fullyConnectedLayer(256)	512	256	512×256 + 256 = 131,328
27	batchNormalizationLayer	256	256	2×256 = 512

N°	Couche	Dimension d'entrée	Dimension de sortie	Nombre de paramètres
28	reluLayer	256	256	0
29	dropout(0.3)	256	256	0
30	fullyConnectedLayer(11)	256	11	256×11 + 11 = 2,827
31	softmaxLayer	11	11	0
32	classificationLayer	11	11	0

La figure III.9 illustre l'évolution des courbes d'apprentissage du réseau proposé pour la classification des 11 modulations en utilisant la base RML2016.a en format IQ.

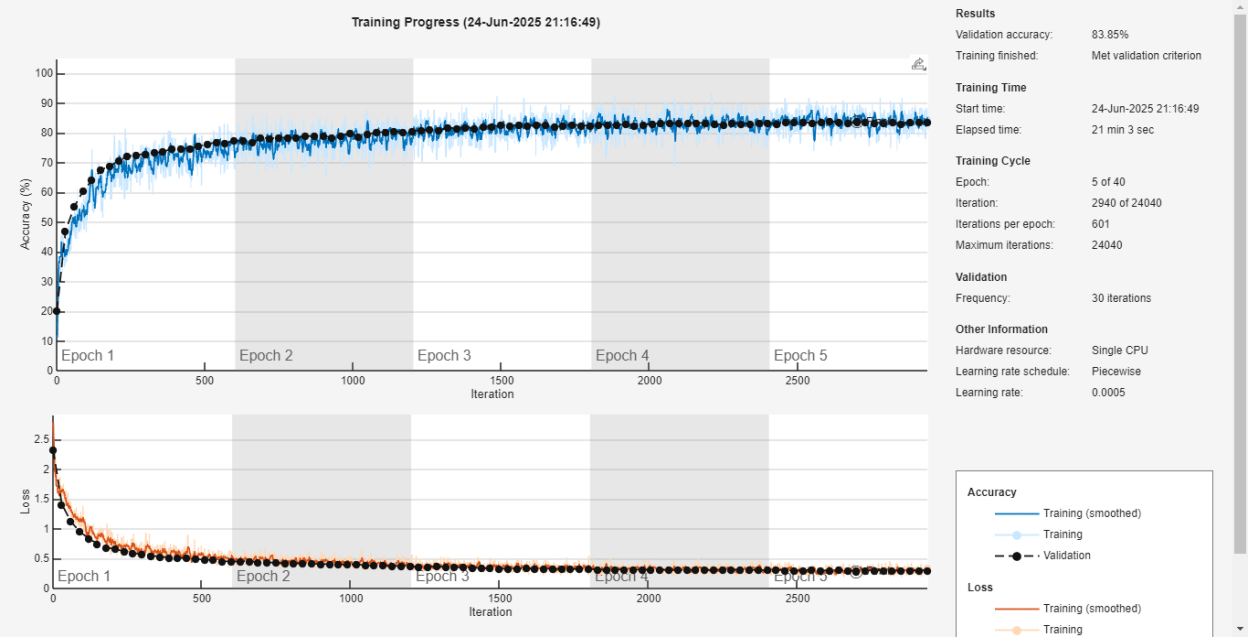


Figure III:9 Evolution de l'entrainement du réseau proposé appliqué à des données en format IQ de la base RML2016a.

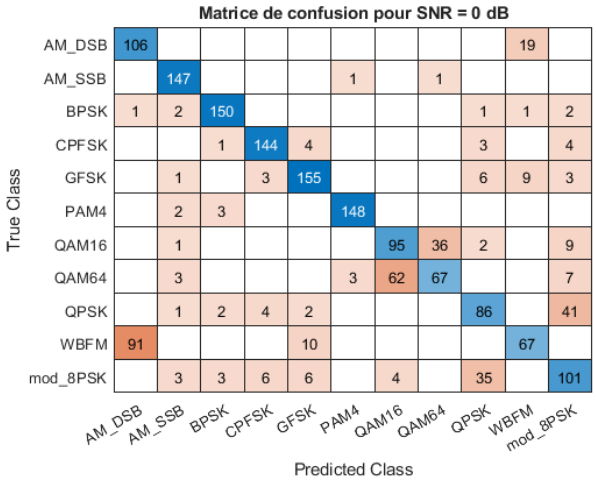
La justesse (accuracy) augmente rapidement pour atteindre environ 80% après deux époques, puis continue à s'améliorer plus lentement pour se stabiliser autour de 83%. Comparé aux résultats précédents utilisant les représentations WVD et spectrogrammes, ce modèle utilisant directement les données IQ montre une performance similaire à ceux du premier modèle proposé et du réseau SqueezeNet utilisant les données WVD tout en évitant le coût computationnel lié à la génération des représentations temps-fréquence, ce qui représente un avantage significatif pour les applications en temps réel.

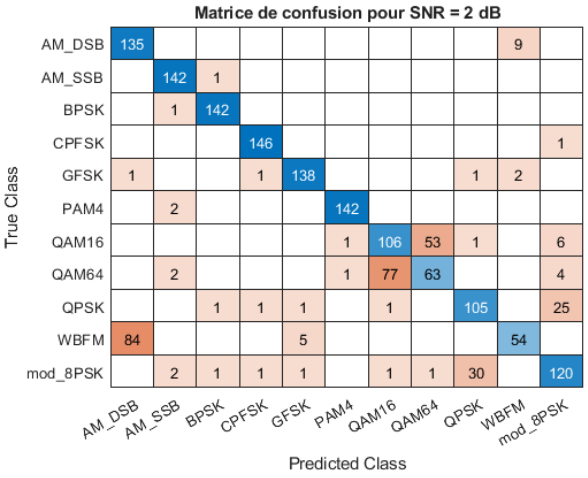
Les valeurs de justesse de test correspondant aux différentes valeurs du SNR sont données dans le tableau III.10.

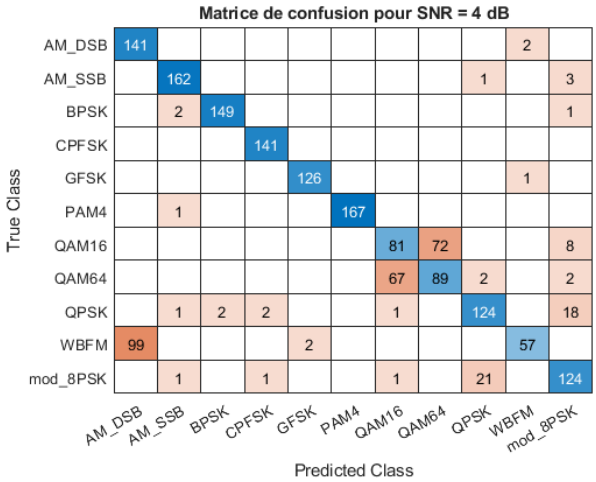
Tableau III:10: justesses de test du réseau proposé appliqué à des données en format IQ de la base RML2016a.

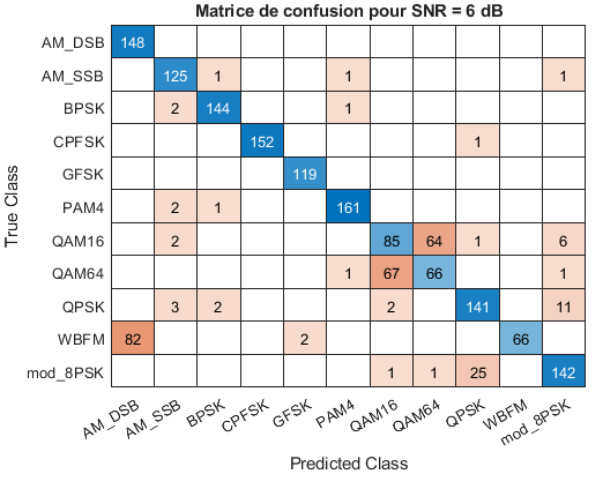
```
Analyse des performances par SNR...
0 dB: 83.61%
2 dB: 84.02%
4 dB: 83.09%
6 dB: 85.17%
8 dB: 84.97%
10 dB: 84.53%
12 dB: 83.70%
14 dB: 84.76%
16 dB: 82.61%
18 dB: 83.87%
```

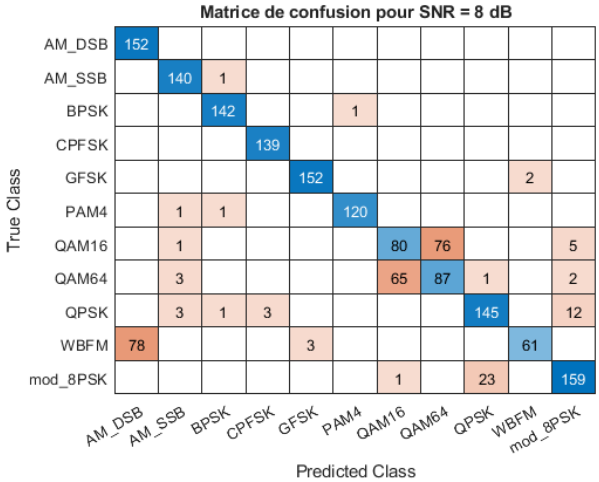
Le tableau montre une performance remarquablement stable du modèle à travers les différentes valeurs de SNR, avec une précision oscillante entre 82.61% (minimum à 16 dB) et 85.17% (maximum à 6 dB), et une moyenne d'environ 84%. Ces résultats sont significativement supérieurs à ceux obtenus avec les spectrogrammes (qui variaient de 64% à 74%) et légèrement meilleurs que ceux des images WVD (environ 81.99%). La stabilité exceptionnelle des performances (variation de seulement ~2.5%), combinée à une excellente robustesse même à faible SNR (83.61% à 0 dB) et l'absence de dégradation significative à haut SNR, suggère que le traitement direct des signaux I/Q par le réseau neuronal est plus efficace que l'utilisation de représentations temps-fréquence intermédiaires, probablement parce que le réseau peut apprendre directement les caractéristiques pertinentes des signaux sans perte d'information due à une transformation intermédiaire.

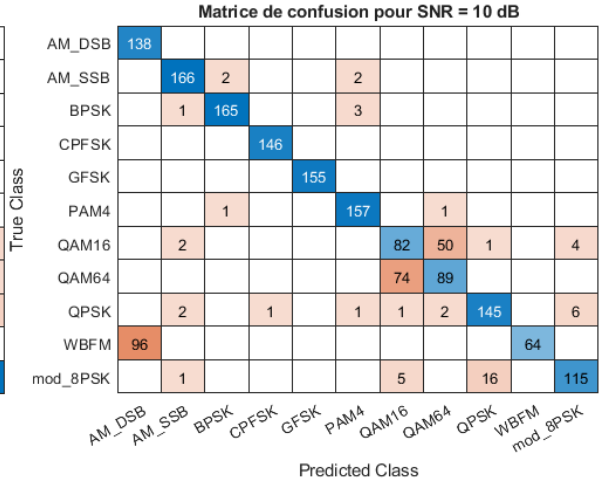

Matrice de confusion pour SNR = 0 dB
True Class
Predicted Class
AM_DSB AM_SSB BPSK CPFSK GFSK PAM4 QAM16 QAM64 QPSK WBFM mod_8PSK
106 19
147 1 1
1 2 150 1 1 2
1 144 4 3 4
1 155 6 9 3
2 3 148
1 95 36 2 9
3 3 62 67 7
1 2 4 2 86 41
91 10 67
3 3 6 6 4 35 101


Matrice de confusion pour SNR = 2 dB
True Class
Predicted Class
AM_DSB AM_SSB BPSK CPFSK GFSK PAM4 QAM16 QAM64 QPSK WBFM mod_8PSK
135 9
142 1
1 142
146 1
1 1 138 1 2
2 142
1 106 53 1 6
2 1 77 63 4
1 1 1 1 105 25
84 5 54
2 1 1 1 1 1 30 120


Matrice de confusion pour SNR = 4 dB
True Class
Predicted Class
AM_DSB AM_SSB BPSK CPFSK GFSK PAM4 QAM16 QAM64 QPSK WBFM mod_8PSK
141 2
162 1 3
2 149 1
141
126 1
1 167
81 72 8
67 89 2 2
1 2 2 1 124 18
99 2 57
1 1 1 21 124


Matrice de confusion pour SNR = 6 dB
True Class
Predicted Class
AM_DSB AM_SSB BPSK CPFSK GFSK PAM4 QAM16 QAM64 QPSK WBFM mod_8PSK
148
125 1 1 1
2 144 1
152 1
119
2 1 161
2 85 64 1 6
1 67 66 1
3 2 2 141 11
82 2 66
1 1 25 142


Matrice de confusion pour SNR = 8 dB
True Class
Predicted Class
AM_DSB AM_SSB BPSK CPFSK GFSK PAM4 QAM16 QAM64 QPSK WBFM mod_8PSK
152
140 1
142 1
139
152 2
1 1 120
1 80 76 5
3 65 87 1 2
3 1 3 145 12
78 3 61
1 23 159


Matrice de confusion pour SNR = 10 dB
True Class
Predicted Class
AM_DSB AM_SSB BPSK CPFSK GFSK PAM4 QAM16 QAM64 QPSK WBFM mod_8PSK
138
166 2 2
1 165 3
146
155
1 157 1
2 82 50 1 4
74 89
2 1 1 2 145 6
96 64
1 5 16 115

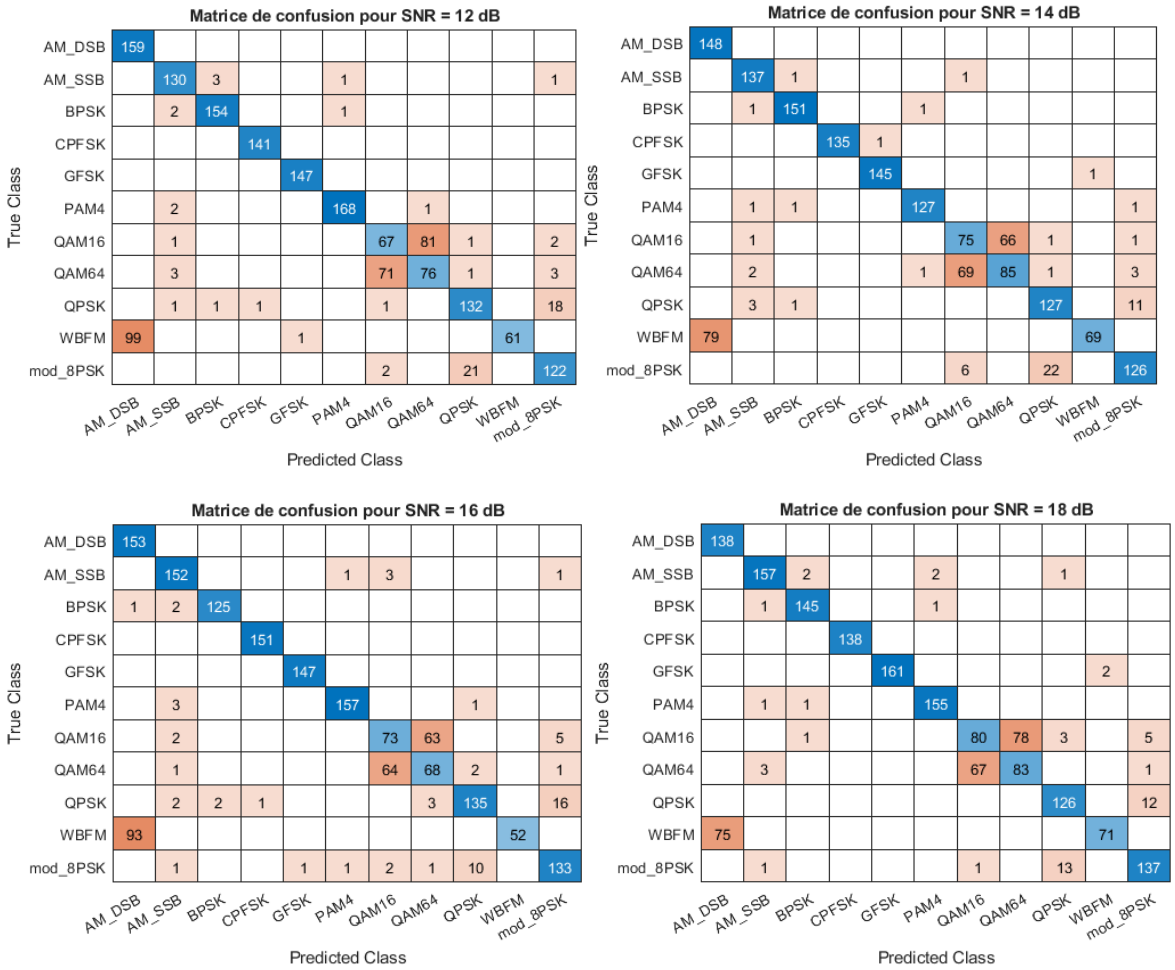


Figure III:10: Matrices de confusion pour les différentes valeurs du SNR obtenu par le premier réseau proposé appliqué à des données en format IQ de la base RML2016a

À partir des matrices de confusion présentées, on peut dégager les observations suivantes :

- **AM_SSB, CPFSK et GFSK** montrent une excellente performance de classification pour toutes les valeurs de SNR, démontrant une robustesse remarquable face au bruit.
- **BPSK** présente des taux de classification très élevés, commençant à environ 97% à 0 dB et maintenant une performance stable autour de 98-99% pour les SNR plus élevés.
- **AM_DSB** montre une bonne performance globale, avec un taux de classification de 94.5% à 0 dB, qui s'améliore progressivement pour atteindre près de 100% à partir de 10 dB.

- **PAM4** affiche une performance solide, débutant à 83.5% à 0 dB et s'améliorant graduellement pour atteindre et maintenir environ 95-98% à partir de 8 dB.
- **QAM64** montre une amélioration significative avec l'augmentation du SNR, Performance modérée (50.5%) à 0 dB, Amélioration progressive jusqu'à atteindre plus de 90% à partir de 10 dB, Excellente performance (près de 100%) à 18 dB
- **QAM16** présente des difficultés majeures de classification, Performance très faible (69.5%) à 0 dB, Dégradation significative pour les SNR plus élevés, Taux extrêmement bas (1-3%) à partir de 14 dB, Confusion fréquente avec QAM64
- **QPSK** montre une amélioration progressive, Performance modeste (45%) à 0 dB, Amélioration graduelle avec le SNR, Atteint environ 95% à partir de 14 dB
- **WBFM** présente des performances limitées, Taux de classification faible (28.5%) à 0 dB, Légère amélioration avec le SNR, Plafonne autour de 40-44% même à haut SNR
- **8PSK** montre une amélioration notable avec le SNR, Performance modérée (67.5%) à 0 dB, Amélioration progressive, Atteint environ 90-94% à haut SNR

III.4. **Conclusion :**

Ce chapitre a démontré l'efficacité de différentes approches de classification automatique des modulations, révélant des résultats significatifs et complémentaires. Le CNN avec données I/Q directes a montré une remarquable stabilité avec une précision moyenne de 84% sur toutes les valeurs de SNR, tandis que la représentation WVD a atteint des performances similaires, les deux approches surpassant significativement l'utilisation des spectrogrammes classiques. Les modulations AM_SSB, CPFSK, GFSK et BPSK ont atteint des taux de classification exceptionnels (95-100%) avec les deux approches, bien que certaines limitations persistent, notamment pour la classification de QAM16 , QAM64 et WBFM. L'approche I/Q directe s'est distinguée par sa stabilité et sa complexité computationnelle réduite, tandis que la représentation WVD a démontré une meilleure capacité de discrimination pour certaines modulations complexes à haut SNR. Ces résultats suggèrent qu'une approche hybride, combinant les avantages des différentes méthodes selon le contexte d'application, pourrait constituer une piste prometteuse pour les développements futurs, permettant d'optimiser la robustesse et l'adaptabilité des systèmes de classification.

CONCLUSION GENERALE

Ce mémoire a exploré l'application des réseaux de neurones profonds à la classification automatique des modulations numériques, en progressant des fondamentaux jusqu'aux implémentations avancées à travers trois chapitres complémentaires.

Le premier chapitre a établi les bases théoriques essentielles des modulations numériques et leurs méthodes de classification traditionnelles. Nous avons exploré les différents types de modulations (ASK, PSK, FSK, QAM), leurs caractéristiques distinctives, et les approches classiques de classification basées sur la vraisemblance (LB) et les caractéristiques (FB). Cette fondation a mis en évidence les limitations des méthodes conventionnelles, particulièrement leur sensibilité au bruit et leur complexité computationnelle.

Le deuxième chapitre a introduit les concepts du Deep Learning et leur application à la classification des modulations. L'accent a été mis sur les réseaux de neurones convolutifs (CNN), leur architecture, et leur capacité à apprendre automatiquement les caractéristiques discriminantes des signaux. Cette partie a démontré l'avantage théorique des approches Deep Learning sur les méthodes traditionnelles, notamment leur capacité d'adaptation et leur robustesse au bruit.

Le troisième chapitre a présenté nos contributions expérimentales avec trois approches innovantes comprenant un CNN personnalisé avec représentation WVD atteignant une stabilité remarquable, un traitement direct des signaux I/Q montrant une précision moyenne de 84% et l'adaptation de l'architecture SqueezeNet aux signaux de télécommunication. Les résultats expérimentaux ont démontré une excellente classification atteignant 95-100% pour AM_SSB, CPFSK, GFSK et BPSK, avec une stabilité remarquable des performances à travers différents SNR et une complémentarité intéressante des approches WVD et I/Q directe.

Nous avons identifié certaines limitations incluant des difficultés persistantes avec certaines modulations complexes comme QAM16, QAM64 et WBFM, des confusions entre modulations d'ordre supérieur et un compromis nécessaire entre complexité computationnelle et performance

Comme perspectives de ce travail on peut citer à titre d’exemples :

CONCLUSION GENERALE

Les améliorations techniques visent le développement d'architectures hybrides combinant les avantages des approches WVD et I/Q, l'optimisation des prétraitements pour les modulations problématiques et l'implémentation de solutions temps réel.

Les extensions fonctionnelles comprennent l'élargissement à d'autres types de modulations, l'intégration dans des systèmes de radio cognitive et le développement de solutions adaptatives.

REFERENCES BIBLIOGRAPHIQUES

[1] M. Bouguerra, "Modulation Numérique," Université Badji Mokhtar Annaba, Département d'Électronique, Support de cours. [En ligne]. Disponible sur: https://elearning-facsci.univ-annaba.dz/pluginfile.php/93325/mod_resource/content/1/MODULATION%20NUMERIQUE.pdf [Consulté en Mai 2025]

[2] "Modulation numérique," dans Ingénierie des Télécommunications, StudySmarter, [En ligne]. Disponible sur: https://www.studysmarter.fr/resumes/ingenierie/ingenierie-des-telecommunications/modulation-numerique/ [Consulté en Mai 2025]

[3] Département Sciences de l'Ingénieur, "Les modulations numériques," Lycée Blaise Pascal, Support pédagogique, [En ligne]. Disponible sur: https://si.blaisepascal.fr/1t-les-modulations-numeriques/ [Consulté en Mai 2025]

[4] J. Mitola III, "Software Radio Architecture: Object-Oriented Approaches to Wireless Systems Engineering," Wiley, 2000.

[5] T. Ulversoy, "Software Defined Radio: Challenges and Opportunities," IEEE Communications Surveys & Tutorials, vol. 12, no. 4, pp. 531-550, 2010.

[6] S. Huang, Y. Yao, Z. Wei, Z. Feng et P. Zhang, "Automatic Modulation Classification Method for Multiple Antenna System Based on Convolutional Neural Network," IEEE Access, vol. 8, pp. 42938-42950, 2020.

[7] B. Sklar, "Digital Communications: Fundamentals and Applications," 2nd ed., Prentice Hall, 2001.

[8] M. Coupechoux, "Radio Cognitive - Introduction," Institut Mines-Télécom, IMT Atlantique, Support de cours, 2018. [En ligne]. Disponible sur: https://marceaucoupechoux.wp.imt.fr/files/2018/02/Radio-Cog-intro.ppt.pdf [Consulté en juin 2025]

[9] O. A. Dobre et R. Inkol, "Automatic Modulation Classification: Theory and Applications," dans Wireless Communications Systems, Springer, 2021.

[10] MathWorks, "Communications Toolbox Documentation," dans MATLAB Documentation, [En ligne]. Disponible sur: https://fr.mathworks.com/help/comm/index.html [Consulté en juin 2025]

REFERENCES BIBLIOGRAPHIQUES

[11] Turing, A. M., “Computing Machinery and Intelligence,” *Mind*, vol. 59, no. 236, pp. 433–460, 1950.

[12] Minsky, M., “Steps Toward Artificial Intelligence,” *Proceedings of the IRE*, vol. 49, no. 1, pp. 8–30, 1961.

[13] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, **“**Gradient-based learning applied to document recognition,” *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[14] W. S. McCulloch and W. Pitts, “A logical calculus of the ideas immanent in nervous activity,” *The Bulletin of Mathematical Biophysics*, vol. 5, no. 4, pp. 115–133, 1943.

[15] Goodfellow, I., Bengio, Y., & Courville, A."Deep Learning."MIT Press, 2016.

[16] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R.,“Dropout: A simple way to prevent neural networks from overfitting,”*Journal of Machine Learning Research (JMLR)*, vol. 15, pp. 1929–1958, 2014..

[17] T. J. O'Shea, T. Roy, and T. C. Clancy, _Over the air deep learning based radio signal classification,” IEEE Journal of Selected Topics in Signal Processing, vol. 12, no. 1, pp. 168_179, 2018.

[18] D. Zhang, W. Ding, B. Zhang, F. Lin, and R. Tang, “Deep learning for radio signal modulation recognition : A survey,” IEEE Access, vol. 8, pp. 132 922_132 956, 2020.

[19] F. Meng, P. Chen, L. Wu, and X. Wang, “Automatic modulation classification : A deep learning enabled approach,” IEEE Transactions on Vehicular Technology, vol. 67, no. 11, pp. 10 760_10 772, 2018.

[20] T. J. O'Shea, J. Corgan, and T. C. Clancy, “Convolutional radio modulation recognition networks,” International conference on engineering applications of neural networks, pp. 213_226, 2016.

[21] J. Hong, W. Wang, Y. Ban, Y. Luo, and J. Ge, “Automatic modulation classification using deep learning,” Journal of Communications and Information Networks, vol. 2, no. 4, pp. 47_56, 2017.

[22] K. Sankhe, M. Belgiovine, F. Zhou, L. Angioloni, F. Restuccia, S. D'Oro, T. Melodia, S. Ioannidis, and K. Chowdhury, “No radio left behind: Radio fingerprinting through deep learning of physical-layer hardware impairments,” IEEE Transactions on Cognitive

Communications and Networking, vol. 6, no. 1, pp. 165_178, 2020.

[23] A. Ali, Y. Fan et al., "Automatic modulation classification using deep learning based on sparse autoencoders with nonnegativity constraints," IEEE Signal Processing Letters, vol. 24, no. 11, pp. 1626_1630, 2017.

[24] M. Sadeghi and E. G. Larsson, "Adversarial training for deep learning-based automatic modulation classification," IEEE Communications Letters, vol. 23, no. 5, pp. 879_882, 2018.

[25] Z. Ahmad and S. K. Jayaweera, "Automatic modulation classification using deep learning architectures," 2019 53rd Asilomar Conference on Signals, Systems, and Computers, pp. 1851_1855, 2019.

[26] J. Xu, C. Luo, G. Parr, and Y. Luo, "A deep learning approach to automatic modulation classification based on time-frequency analysis," Sensors, vol. 20, no. 23, p. 6912, 2020.

[27] Zeng, Y., Zhang, M., Han, F., Gong, Y., Zhang, J., 2019. Spectrum analysis and convolutional neural network for automatic modulation recognition. IEEE Wireless Commun. Lett. 8, 929–932

[28] Hiremath, S.M., Behura, S., Kedia, S., Deshmukh, S., Patra, S.K., 2019. Deep learning-based modulation classification using time and stockwell domain channeling, in: Proc. Nat. Conf. Commun., pp. 1–6.

[29] Wu, H., Li, Y., Zhou, L., Meng, J., 2019. Convolutional neural network and multi-feature fusion for automatic modulation classification. Electron. Lett. 55, 895–897.

[30] Hong, D., Zhang, Z., Xu, X., 2017. Automatic modulation classification using recurrent neural networks, in: Proc. IEEE Int. Conf. Comput. Commun., pp. 695–700.

[31] West, N.E., O'Shea, T., 2017. Deep architectures for modulation recognition, in: Proc. IEEE Int. Symp. Dyn. Spectr. Access Netw.,, pp. 1–6

[32] Ghasemzadeh, P., Hempel, M., Sharif, H., 2022. GS-QRNN: A high-efficiency automatic modulation classifier for cognitive radio IoT. IEEE Internet Things J., 1–1

[33] Forrest N. Iandola, Song Han, Matthew W. Moskewicz, Khalid Ashraf, William J. Dally, and Kurt Keutzer, "**SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size**",*arXiv preprint arXiv:1602.07360*, 2016.