

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITÉ MOHAMED BOUDIAF - M'SILA

FACULTÉ : TECHNOLOGIE
DEPARTEMENT : ELECTRONIQUE
N° : 2019/ ESEM /



DOMAINE: Sciences et Technologies
FILIERE : Électronique
OPTION : Electronique des Systèmes
Embarqués

Mémoire présenté pour l'obtention du diplôme de Master Académique

Par : ALOUACHE Rayhana

CHIA Radhia

Intitulé

**ÉVALUATION DE LA PERFORMANCE
D'UN CAPTEUR LOGICIEL EN
UTILISANT L'APPRENTISSAGE EN
PROFONDEUR**

Soutenu publiquement le 26 Juin 2019 devant un jury composé de :

Mr. LADJAL Mohamed	Université de M'sila	Président
Mr. DJERIOUI Mohamed	Université de M'sila	Encadreur
Mr. BRIK Youcef	Université de M'sila	Examineur

Année universitaire : 2018 /2019

REMERCIEMENTS

Tout d'abord nous remercions notre encadreur

Dr. DJERIOUI Mohamed qui a accepté de nous encadrer pour réaliser ce mémoire. Nous lui sommes reconnaissants de nous avoir fait bénéficier tout au long de ce travail de sa grande compétence, de sa rigueur intellectuelle, de son efficacité et de ses précieux conseils.

Nous tenons à remercier également **Dr. BRIK Youcef** d'avoir réservé une partie de son temps pour suivre notre travail.

Nous tenons également à remercier les membres de jury qui nous font l'honneur de juger ce mémoire et venir proclamer sa valeur scientifique. Nous ne manquerons pas de remercier également tous ceux qui ont contribué de près ou de loin à la réalisation de ce mémoire.

Dédicace

Je dédie ce modeste travail à mon cher père pour ses sacrifices en témoignage de tout mon affection au long de mes études.

A ma chère mère qui n'a pas cessé de m'encourager depuis ma première année scolaire.

A mon cher mari qui m'a encouragé et m'a apporté un soutien particulier.

A mon frère Abderrraouf.

A mes sœurs Hadjer, Khaoula, Maria

Et

A toute la famille

Rayhana

Dédicace

*Après de longues années d'études et de travail, sachant l'importance de
l'aide des êtres qui m'aiment*

Je dédie ce travail :

*Aux deux personnes, les plus chers pour moi dans ce monde, Qui ne
m'a pas quitté pour toujours, Qui ont veillé à ce que je sois ce que je
suis devenu maintenant.*

*Mon cher père qui a su m'imprégner de son savoir et de sa sagesse.
ET qui ma transmis toute sa grande patience,*

*A Ma chère mère qui s'est sacrifiée pour moi et témoignée de toute
mon affection tout le long de mon parcours scolaire et académique.
ET qui m'a transmit toute sa volonté, sa persévérance.*

Que dieu les préserve et les bénisse.

*A mon chère collègue d'étude Djoudi Boubaya qui m'a aidé et m'a
apporté un soutien particulier et m'a conseillé à tout moment. Que
dieu le préserve.*

A ma belle sœur ACHOIK, c'est tout ce que j'ai dans cette vie

A mon binôme Ryhana

A Toutes ma famille

*A toutes mes amies surtout : Fahima, Turkia, Sihem, Safa , Nadjet,
Maysa, Hafida Hassiba et Wafa.*

Radhia

Table des matières

LISTE DES FIGURES	v
LISTE DES TABLEAUX	vi
ABREVIATION ET NOTATION.....	vii
INTRODUCTION GENERALE	1

CHAPITRE I

Les Capteurs Utilisés Dans L'industrie

INTRODUCTION	3
I.1. DEFINITION D'UN CAPTEUR	3
I.2. PRINCIPE DE FONCTIONNEMENT	4
I.3. CONSTITUTION D'UN CAPTEUR	4
I.3.1. Le corps d'épreuve	5
I.3.2. Transducteur	5
I.3.3. Le boîtier	5
I.3.4. L'électronique de conditionnement	5
I.4. TYPES DE CAPTEURS	5
I.4.1. Capteurs actifs	5
I.4.2. Capteurs passifs	6
I.5. PERFORMANCES D'UN CAPTEUR	6
I.5.1. Etendue de la mesure	6
I.5.1.1. Zone nominale d'emploi	6
I.5.1.2. Zone de non-détérioration	6
I.5.1.3. Zone de non-destruction	6
I.5.2. Résolution.....	6
I.5.3. Caractéristique d'entrée-sortie d'un capteur	6
I.5.4. Sensibilité	6
I.5.5. Finesse	7
I.5.6. Linéarité	7
I.5.7. Caractéristiques statistiques d'un capteur	7
I.5.7.1. Fidélité	7

I.5.7.2. Justesse	7
I.5.7.3. Précision	7
I.5.8. Rapidité	8
I.6. L'EMPLACEMENT D'UN CAPTEUR	8
I.7. CAPTEUR INTELLIGENT	10
I.7.1. Définition d'un capteur intelligent	10
I.7.2. Constitution des capteurs intelligents	10
I.7.3. Domaines d'applications des capteurs intelligents	12
I.7.4. Avantages de l'intelligence d'un capteur	12
I.8. LES CAPTEURS LOGICIELS	13
I.8.1. Définition d'un capteur logiciel (soft sensor).....	13
I.8.2. Propriétés des capteurs logiciels	13
I.8.3. L'utilisation des capteurs logiciels	14
I.8.4. Les types des capteurs logiciels	15
I.8.4.1. Capteurs logiciels pilotés par les données (Data-driven Soft Sensors) ..	15
I.8.4.2. Capteurs logiciels entraînés par le modèle (Model Driven Soft Sensors).....	15
I.8.5. Développement du capteur logiciel	16
I.8.5.1. Acquisition des données	17
I.8.5.2. Sélection des variables d'entrées	17
I.8.5.3. Choix du modèle et apprentissage	18
I.8.5.4. Validation du modèle	19
I.8.5.5. Maintenance du modèle	19
I.8.6. Applications des capteurs logiciels	20
I.8.6.1. Prédiction en ligne	20
I.8.6.2. Surveillance de processus	21
I.8.6.3. Détection et reconstruction de défauts de capteurs	21
CONCLUSION	22

CHAPITRE II

Les techniques d'apprentissages statistiques

INTRODUCTION	23
II.1. RESEAUX DE NEURONES	23
II.2. INTRODUCTION SUR DEEP LEARNING	27
II.3. DEFINITION DE L'APPRENTISSAGE EN PROFONDEUR	27
II.4. HISTOIRE DU DEEP LEARNING	28
II.5. DOMAINES D'APPLICATION DE DEEP LEARNING	28
II.6. DIFFERENTS TYPES DE MODELES DE DEEP LEARNING.....	28
II.6.1. Réseaux de neurones convolutionnels.....	28
II.6.2. Deep Belief Network(DBN).....	31
II.6.3. Auto-encodeur (Auto-encoders).....	32
II.6.3.1. Architecture	32
II.6.3.2. Entraînement d'auto-encodeurs	33
II.6.3.4. SAE-DNN (Stacked autoencoder - Deep Neural Network)	35
CONCLUSION	36

CHAPITRE III

Résultats et Discussion

INTRODUCTION	37
III.1. DESCRIPTION DE LA BASE DE DONNÉES	37
III.2. PRÉTRAITEMENT	38

III.2.1 Normalisation des données	38
III.3. CRITÈRE D'ÉVALUATION	39
III.4. DÉVELOPPEMENT D'UN CAPTEUR LOGICIEL	40
III.4.1 Choix du modèle	40
III.4.1.1. Modèle Neuronal-MLP	40
III.4.1.2. Modèle Neuronal-RBF	41
III.4.1.3 Modèle Deep Learning	42
III.4.2. Évaluation des performances	42
III.5. APPLICATION	43
CONCLUSION	45
CONCLUSION GENERALE.....	46
BIBLIOGRAPHIE.....	48

LISTE DES FIGURES

Fig. I.1. Schéma synoptique d'un capteur	3
Fig. I.2. Principe d'un capteur	4
Fig. I.3. Structure d'un capteur	5
Fig. I.4. Caractéristique statistique d'un capteur	8
Fig. I.5. Constitution d'une chaîne de mesure classique	9
Fig. I.6. Architecture matérielle d'un capteur intelligent	11
Fig. I.7. Capteur logiciel connecté à l'usine	15
Fig. I. 8. Principales étapes de développement d'un capteur logiciel	16
Fig. II.1.(a) un RN organisé en 4 couches, (b) le mécanisme d'activation d'un neurone	23
Fig. II.2. La relation entre l'intelligence artificielle, le ML et le deep Learning	27
Fig. II.3. Fonctionnement d'une convolution sur une image	30
Fig. II.4. Deep Belief Network qui identifie des images de digits	31
Fig. II.5. Architecture des Auto-encodeurs	33
Fig. II.6. Un apprentissage non supervisé d'un auto-encodeur	34
Fig. II.7. Un apprentissage non supervisé multicouches d'un auto-encodeur	34
Fig. II.8. Un apprentissage supervisé d'un auto-encodeur	35
Fig. II.9. Architecture de SAE-DNN	36
Fig. III.1. classifieur et capteur logiciel OD à base du modelé DL	44

LISTE DES TABLEAUX

Tableau. II.1. Les fonctions d'activations	24
Tableau. II.2. Les étapes majeures du Deep Learning	28
Tableau. III.1. Résultats d'apprentissage et de test (Modèle neural-MLP) pour le capteur d'OD	40
Tableau. III.2. Résultats d'apprentissage et de Test (Modèle neural-MLP) pour le capteur logiciel de CI	40
Tableau. III.3. Résultats d'apprentissage et de Test (Modèle neural-RBF) pour le capteur logiciel d'OD	41
Tableau. III.4. Résultats d'apprentissage et de Test (Modèle neural-RBF) pour le capteur logiciel de CI	41
Tableau. III.5. Résultats d'apprentissage et de Test (Modèle Deep Learning) pour le capteur logiciel d'OD	42
Tableau. III.6. Résultats d'apprentissage et de Test (Modèle Deep Learning) pour le capteur logiciel de CI	42
Tableau. III.7. Évaluation des performances d'OD pour des différentes techniques ..	43
Tableau. III.8. Évaluation des performances de CI pour des différentes techniques ...	43
Tableau. III.9. Résultats obtenus pour les 3 scénarios/cas	44

ABREVIATIONS ET NOTATIONS

AE : Auto-encodeur

C : Capacité

Convertisseur A/N : Convertisseur Analogique/Numérique

CNN: Convolutional neural network

C: Conductivité

Cl^- : Chlore

Ca^{2+} : Calcium

Cov: La covariance

DL: Deep Learning

DBN: DeepBelief Network

DNN: Deep Neural Network

DDP: Différence De Potentiel

EAM: Erreur absolue moyenne

EQMG: Erreur quadratique moyenne de généralisation

fcts : Fonctions d'activations.

HCO_3^- : Bicarbonate

L : L'inductance

m: La grandeur à mesurer (mesurande)

ML: Machine Learning

MLP: Multi-Layer-Perceptron

***min*(X)** : La valeur minimale de *X*

***max*(X)** : La valeur maximale de *X*

***Mg*²⁺** : Magnésium

NN : Neural Network

OBD: Demande Biochimique d'Oxygène

OD: Oxygène dissous

PIC: Contrôleur d'Interface Périphérique

PROM: Programmable Read Only Memory

PH: Potentiel Hydrogène

R: Résistance

RAM: Random Access Memory

RF: Radio Fréquence

RBF: Radial Basis Function

RN : Réseaux de Neurones

***R*²**: Le coefficient de corrélation

REQM : Racine de l'erreur quadratique moyenne

SAE-DNN: Stacked autoencoder - Deep Neural Network

SM : Matières en Suspension

T° : Température

***t*_{app}**: Le temps d'apprentissage

***t*_{test}**: Le temps de test

***W*_{ij}** : Poids synaptiques

$\mathbf{X} = [x_1, x_2, \dots, x_i]$: Le vecteur à normaliser

x_{iN} : La valeur normalisée

x_{\min} : la valeur minimale

x_{\max} : la valeur maximale

y_r : Sortie réelle

y_c : Sortie calculée

φ : Fonction d'activation

σ : l'écart type

INTRODUCTION GENERALE

INTRODUCTION GENERALE

L'industrie regroupe l'ensemble des activités économiques qui produisent en série des biens matériels, par la transformation de matières premières ou de matières ayant déjà subi une ou plusieurs transformations, et l'exploitation de sources d'énergie.

Le contrôle de processus industriel permet de surveiller et réguler les machines, les installations et les processus dans un grand nombre de secteurs industriels : chimie, pharmacie, biotechnologie, énergie, eau/eaux usées, pétrole, gaz, matières plastiques, papier, alimentation et boisson. Aujourd'hui, les processus industriels de traitement, de transformation et de fabrication sont fortement automatisés afin de gérer les matières premières et l'énergie en toute sécurité d'une manière efficace et sans gaspillage des ressources. Pour cet effet, il est indispensable d'utiliser des capteurs physiques qui assurent des mesures précises dans des environnements difficiles, humides ou poussiéreux, ainsi que dans des liquides et gaz agressifs, corrosifs, explosifs. Bien que certains de ces paramètres puissent être mesurés en continu à l'aide de capteurs à faible coût, il existe d'autres paramètres nécessitant des analyses de laboratoire spécifiques et coûteuses en raison de l'absence de capteurs dédiés. En fait, et pour plusieurs raisons, telles que le coût élevé de certains capteurs, leur nombre, le temps consacré à leur contrôle, leur nettoyage, les routines de calibrage et de remplacement, rendent souvent leur bon fonctionnement difficile.

Suite aux problèmes susmentionnés, les capteurs logiciels (Soft Sensors) conçus à base d'algorithmes inspirés des techniques de l'intelligence artificielle, sont devenus un moyen efficace et très attractif pour pallier au manque de capteurs physiques et/ou spécifiques. Ils permettent la réalisation de réseaux de surveillance plus complets, et peuvent facilement être implantés dans des micro-informatiques et réajustés lorsqu'il y a un changement dans les paramètres du système. Ils permettent aussi d'estimer en temps réel des données et de diminuer les retards introduits par des capteurs physiques lents. Par ses interventions, ils permettent d'améliorer la performance des stratégies de contrôle [1]. Plusieurs recherches ont porté sur le développement de ce type de capteurs, en particulier dans le secteur de l'eau. Dans [2], les auteurs ont développé un capteur logiciel en utilisant un algorithme génétique basé sur le système neuro-flou dans le

traitement des eaux usées pour estimer la dynamique des nutriments de l'opération anoxique / oxiqne en utilisant des mesures en ligne plus fiables de perméat-concentration d'hydrogène où le potentiel de réduction et l'oxygène dissous (OD) sont utilisés comme données d'entrée. Lamrini et al [3] ont présenté l'élaboration et la validation d'un capteur logiciel en utilisant un Perceptron Multi- Couche (PMC) pour la prédiction en ligne du dosage optimal de coagulant à partir de mesures de qualité d'eau brute dans une usine de traitement d'eau potable.

Le travail réalisé s'articule sur trois chapitres qui sont présentés comme suit :

Le premier chapitre est une description générale sur les capteurs, leurs structures, et leurs différents types notamment les capteurs physiques, les capteurs intelligents, et également capteur logiciel.

Dans le deuxième chapitre, nous détaillons les fondements théoriques des méthodes d'apprentissage statistiques déjà évoquées. Nous rappelons la notion des Réseaux des Neurones (RN) artificiels et leur fonctionnement. Outre, nous abordons une nouvelle technique dérivée de RN qui s'appelle l'apprentissage en profondeur (En Anglais : Deep Learning DL) et nous discutons leurs différents types.

Le troisième et le dernier chapitre est consacré à la simulation et l'interprétation des expériences menées sur des bases de données afin de valider les modèles proposés. Pour la validation du système proposé, nous avons évalué trois techniques différentes à savoir le Multi-Layer-Perceptron (MLP), le Radial Basis Function (RBF) et le Deep Learning pour créer un capteur logiciel qui répond efficacement aux besoins de l'application traitée. En effet, le but d'un capteur logiciel est de remplacer un capteur physique pour des raisons multiples comme la défaillance, le coût élevé ou l'indisponibilité de ce genre de capteur.

Afin de mener une étude comparative permettant un choix décisif de la méthode la mieux adaptée à l'application proposée, nous avons récapitulé les meilleurs performances de chaque méthode testé dans des tableaux lisibles facilement qui permettent de sortir quelle méthode plus adaptée à l'application ciblée.

Finalement, notre mémoire sera clôturé par une conclusion générale qui retrace les différentes étapes réalisées et souligne les futures perspectives envisagées.

CHAPITRE I

Les Capteurs Utilisés Dans L'industrie

CHAPITRE I

Les Capteurs Utilisés Dans L'industrie

INTRODUCTION

Les capteurs jouent des rôles de plus en plus important car ce sont eux qui permettent de mesurer les effets des phénomènes de toutes natures qui agissent sur l'environnement de l'homme, avec l'évolution de la technologie, l'électronique en particulier, leur importance s'accroît car il permette d'assurer la liaison homme, machine et environnement. En instrumentation industrielle, particulièrement au niveau des chaînes de mesures, Le capteur convertit toute quantité physique ou chimique mesurée en un signal électrique.

Dans ce chapitre, nous allons parler des différents types de capteurs utilisés dans l'industrie. Et on va donner également des généralités sur ces capteurs et leurs différents types. Il existe trois types de capteurs : capteur physique, capteur intelligent et capteur logiciel.

I.1. DÉFINITION D'UN CAPTEUR

Un capteur est un composant électronique qui transforme une grandeur physique observée (Température, pression, humidité, etc.) en grandeur utilisable (intensité électrique, tension ou charge) pour cela, il possède au moins un transducteur dont le rôle est de convertir une grandeur physique en une autre (la figure I.1 présente le principe d'un capteur) [4].

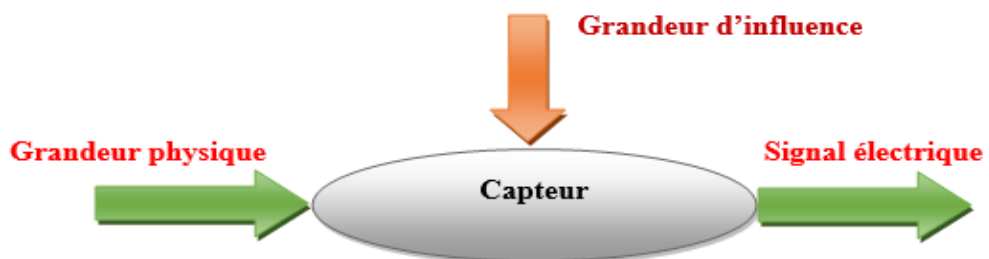


Fig. I.1. Schéma synoptique d'un capteur.

I.2. PRINCIPE DE FONCTIONNEMENT

La grandeur à mesurer est appelée mesurande « m ». Le rôle du capteur est donc de convertir « m » en une grandeur électrique qu'on appellera la mesure « S » [5], et qui peut être une impédance (on parle dans ce cas de capteurs passifs) ou une charge, un courant ou une différence de potentiel (DDP) (on parle ici de capteurs actifs).

On parle aussi de transducteur. Il permet de transformer la grandeur physique d'entrée (le mesurande) en une autre grandeur physique de sortie ou en un signal électrique.

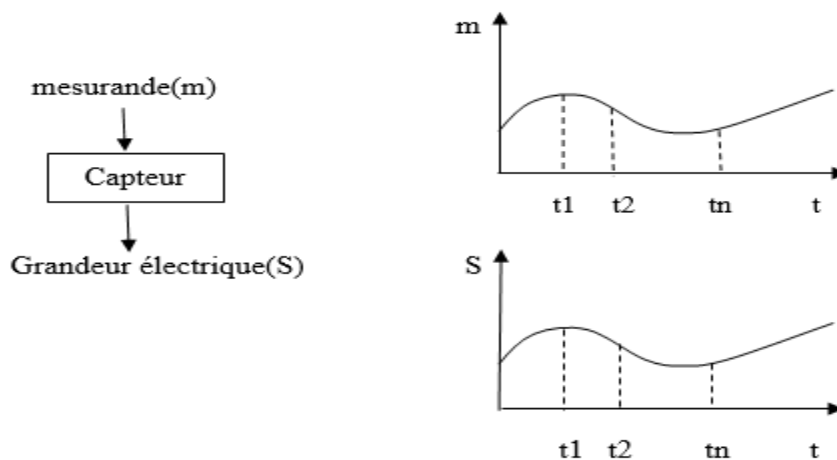


Fig. I.2. Principe d'un capteur.

Généralement, on obtient une grandeur de sortie du type électrique. Elle peut être soit :

- Une charge.
- Une tension.
- Un courant.
- Une impédance (R, L, C).

En général, le signal délivré par le capteur n'est pas directement utilisable et a besoin d'être amplifié et adapté. L'ensemble des circuits et appareils qui assure ces opérations est appelé chaîne de mesure et du traitement du signal [5].

I.3. CONSTITUTION D'UN CAPTEUR

Les différentes parties pouvant constituer un capteur en général sont décrites dans la figure I.3

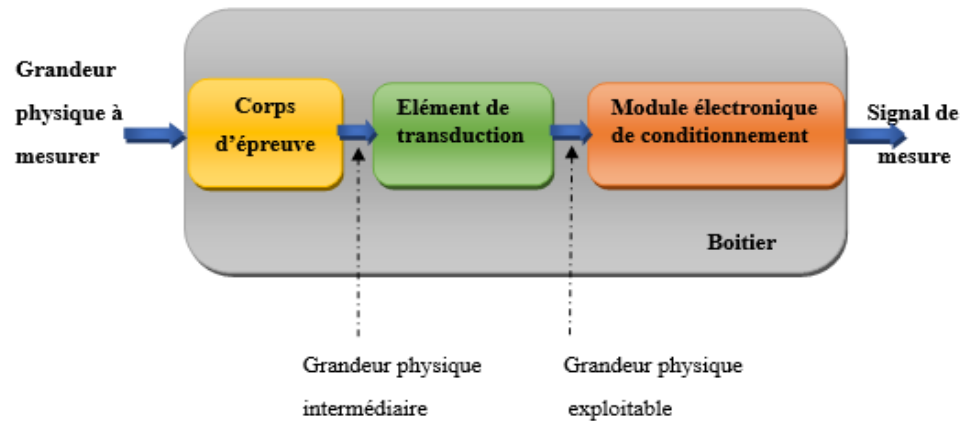


Fig. I.3. Structure d'un capteur

I.3.1. Le corps d'épreuve

Le corps d'épreuve est un élément sensible qui réagit à la grandeur à mesurer. Il a pour rôle de transformer la grandeur à mesurer en une autre grandeur physique intermédiaire dite mesurable.

I.3.2. Transducteur

Le transducteur est un élément sensible lié au corps d'épreuve. Il traduit les réactions du corps d'épreuve en une grandeur électrique constituant le signal de sortie.

I.3.3. Le boîtier

Le boîtier est un élément mécanique de protection, de maintien et de fixation du capteur.

I.3.4. L'électronique de conditionnement

C'est un dispositif qui convertit le signal de sortie du capteur en un signal de mesure standard [6].

I.4. TYPES DE CAPTEURS

Suivant l'origine du signal électrique de sortie, on peut classer les capteurs en deux types :

I.4.1. Capteurs actifs

Source qui produit un signal électrique traduisant le mesurande aussi fidèlement que possible. La sortie "S" est une : charge, tension ou courant.

I.4.2. Capteurs passifs

Impédance dont la variation traduit le mesurande et qui est mesurable que par un circuit approprié (conditionneur) alimenté par une source extérieure. La sortie "S" est une : résistance, inductance ou capacité [7].

I.5. PERFORMANCES D'UN CAPTEUR

De manière à classer les capteurs en fonction de leurs performances, on est amené à définir des paramètres qui permettent de les sélectionner en fonction de l'application.

I.5.1. Étendue de la mesure

Elle définit la zone dans laquelle les caractéristiques du capteur sont assurées par rapport à des spécifications données. On peut classer cette zone en trois familles :

I.5.1.1. Zone nominale d'emploi

Zone dans laquelle le mesurande peut évoluer sans modification des caractéristiques du capteur.

I.5.1.2. Zone de non-détérioration

Valeurs limites des grandeurs influençant le capteur (mesurande, température environnante, etc...) sans que les caractéristiques du capteur ne soient modifiées après annulation de surcharges éventuelles.

I.5.1.3. Zone de non-destruction

Elle définit les limites garantissant la non-destruction du capteur mais dans laquelle il peut y avoir des modifications permanentes des caractéristiques du capteur.

I.5.2. Résolution

Elle correspond à la plus petite variation du mesurande que le capteur est susceptible de déceler.

I.5.3. Caractéristique d'entrée-sortie d'un capteur

Elle donne la relation d'évolution de la grandeur de sortie en fonction de la grandeur d'entrée. Elle est donnée classiquement par une courbe en régime permanent. Elle ne donne pas d'informations sur les caractéristiques transitoires du capteur.

I.5.4. Sensibilité

Elle détermine l'évolution de la grandeur de sortie en fonction de la grandeur d'entrée en un point donné. C'est la pente de la tangente à la courbe issue de la caractéristique du capteur. Dans le cas d'un capteur linéaire, la sensibilité du capteur est constante.

$$\text{Sensibilité} = \frac{\Delta(\text{Grandeur de sortie})}{\Delta(\text{mesurande})} \quad (\text{I.1})$$

Il faut noter que la sensibilité d'un capteur peut être fonction du conditionneur auquel il est associé.

I.5.5. Finesse

C'est la qualité d'un capteur à ne pas venir modifier par sa présence la grandeur à mesurer. Cela permet d'évaluer l'influence du capteur sur la mesure. On la définit non seulement vis à vis du capteur mais aussi vis à vis de l'environnement d'utilisation du capteur. Par exemple, dans le cas d'une mesure thermique, on cherchera un capteur à faible capacité calorifique vis à vis des grandeurs l'entourant. Finesse et sensibilité sont en général antagonistes. Il peut y avoir un compromis à faire. Pour un capteur d'induction B, un capteur à forte perméabilité sera très sensible, par contre sa présence aura tendance à perturber les lignes de champ et la mesure de l'induction ne sera pas celle sans capteur, d'où une mauvaise finesse. Mais cette erreur peut être évaluée en vue d'une correction post-mesure et ainsi faire abstraction de la présence du capteur.

I.5.6. Linéarité

Zone dans laquelle la sensibilité du capteur est indépendante de la valeur du mesurande.

I.5.7. Caractéristiques statistiques d'un capteur

Ces paramètres permettent de prendre en compte la notion d'erreurs accidentelles qui peuvent survenir sur un capteur.

I.5.7.1. Fidélité

Elle définit la qualité d'un capteur à délivrer une mesure répétitive sans erreurs. L'erreur de fidélité correspond à l'écart type obtenu sur une série de mesures correspondant à un mesurande constant.

I.5.7.2. Justesse

C'est l'aptitude d'un capteur à délivrer une réponse proche de la valeur vraie et ceci indépendamment de la notion de fidélité. Elle est liée à la valeur moyenne obtenue sur un grand nombre de mesures par rapport à la valeur réelle.

I.5.7.3. Précision

Elle définit l'écart en % que l'on peut obtenir entre la valeur réelle et la valeur obtenue en sortie du capteur. Ainsi un capteur précis aura à la fois une bonne fidélité et une bonne justesse.

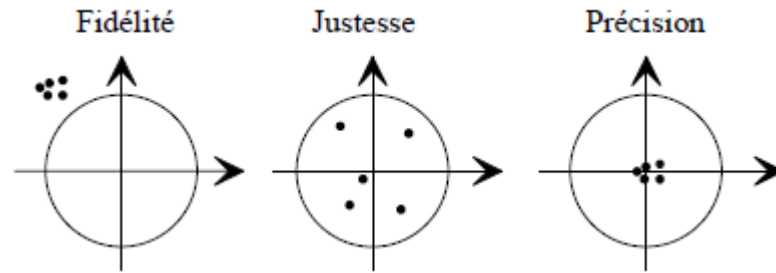


Fig. I.4. Caractéristique statistique d'un capteur [8].

I.5.8. Rapidité

C'est la qualité d'un capteur à suivre les variations du mesurande. On peut la chiffrer de plusieurs manières [5] :

- bande passante du capteur.
- Fréquence de résonance du capteur.
- Temps de réponse (à x%) à un échelon du mesurande.

I.6. EMPLACEMENT D'UN CAPTEUR

Pour obtenir une image d'une grandeur physique, on fait appel à une chaîne de mesure qui peut faire intervenir plusieurs phénomènes différents. La mesure d'un débit par exemple peut se faire en plusieurs étapes :

- transformation du débit en une pression différentielle ;
- transformation de la pression différentielle en la déformation mécanique d'une membrane ;
- transformation de la déformation mécanique en une grandeur électrique (à l'aide d'un piézo-électrique) via un circuit électronique associé. L'ensemble de ces étapes constitue la chaîne de mesure (fig. I.5).

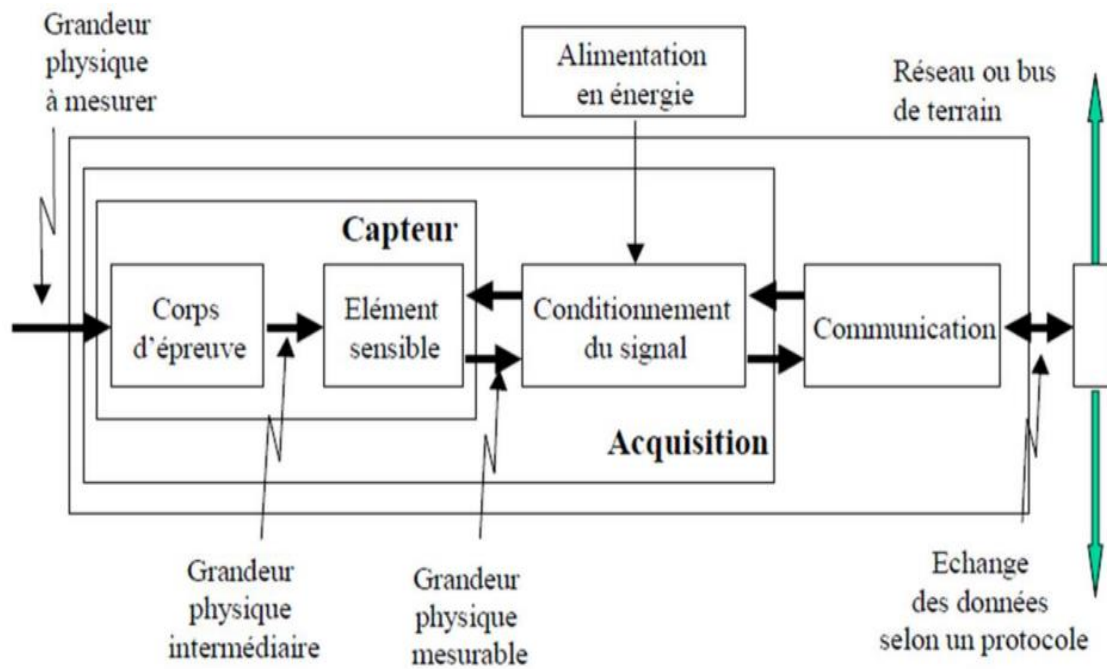


Fig. I.5. Constitution d'une chaîne de mesure classique [1].

Le corps d'épreuve est l'élément influencé par la grandeur physique à mesurer. Il convertit cette grandeur en une autre grandeur physique intermédiaire, très souvent un déplacement.

L'élément sensible convertit cette grandeur en une grandeur mesurable, le plus souvent une grandeur électrique.

Le circuit qui assure le conditionnement du signal, sert à traiter la grandeur mesurable pour délivrer un signal de sortie ayant des caractéristiques spécifiées.

Les fonctions assurées peuvent être multiples de la simple mise en forme et adaptation du signal à un traitement élaboré de corrections des grandeurs d'influence.

Dans la plupart des cas, la sortie d'une chaîne de mesure est du type électrique. Si la chaîne de mesure fait intervenir plusieurs transducteurs, on appelle corps d'épreuve celui en contact direct avec le mesurande. Le dernier transducteur est associé à un conditionneur qui fournit la grandeur électrique de sortie de manière exploitable. Le choix de ce conditionneur est une étape importante dans le cadre de la chaîne de mesure car, associé au capteur, il détermine la nature finale du signal électrique et va influencer les performances de la mesure [1].

I.7. CAPTEUR INTELLIGENT

Les possibilités technologiques de la micro-électronique ont contribué à l'apparition du concept de capteur intelligent. En dotant le transmetteur associé aux transducteurs d'un organe de calcul interne (microprocesseur ou autre), les performances métrologiques et fonctionnelles d'un capteur sont considérablement accrues. Les caractéristiques nouvelles sont principalement un gain important sur la précision et la crédibilité des mesures, le traitement de signaux numériques et la possibilité de communication avec un réseau local ou à un ordinateur. L'avènement prochain de ce type d'instrumentation constitue un enjeu technologique et économique important, dans le domaine du procédé, de la productique et de la robotique particulièrement [9].

I.7.1. Définition d'un capteur intelligent

Deux définitions de capteur intelligent semblent alors répandues. La première fait référence à la présence d'un microprocesseur embarqué. Celui-ci peut avoir comme objectif de modifier le comportement interne du capteur afin d'optimiser sa capacité à collecter les données, ou simplement pour effectuer localement des traitements et des calculs. La seconde définition se focalise sur la capacité de communication bidirectionnelle du capteur, avec des systèmes extérieurs et des opérateurs humains. Le capteur reçoit et traite des commandes extérieures, et envoie des mesures et des informations de statut [10].

I.7.2. Constitution des capteurs intelligents

Un capteur intelligent se compose alors de quatre unités montrées dans la figure I.6.

- 1) **Un capteur principal** spécifique au mesurande avec ses dispositifs d'acquisition et de numérisation du signal : transducteur, conditionneur qui adapte le signal électrique en vue de sa transmission, multiplexeur, amplificateur, échantillonneur bloqueur, convertisseur analogique/numérique.
- 2) **Un organe de calcul numérique** (microcontrôleur, microprocesseur, des PIC) servant au calcul et à la gestion de l'acquisition, la correction des effets des grandeurs d'influence au moyen de paramètres stockés en mémoire PROM, la linéarisation, le diagnostic des capteurs.

- 3) **Une interface de communication** assurant la liaison du capteur à un calculateur central et permettant un dialogue bidirectionnel de données numériques avec le système d'automatisation.
- 4) **Une alimentation** assurant une stabilisation des tensions est nécessaire à l'électronique de l'instrument. Une batterie peut être envisagée pour maintenir certaines activités en l'absence de source d'énergie extérieure (horloge, mémoire,...).

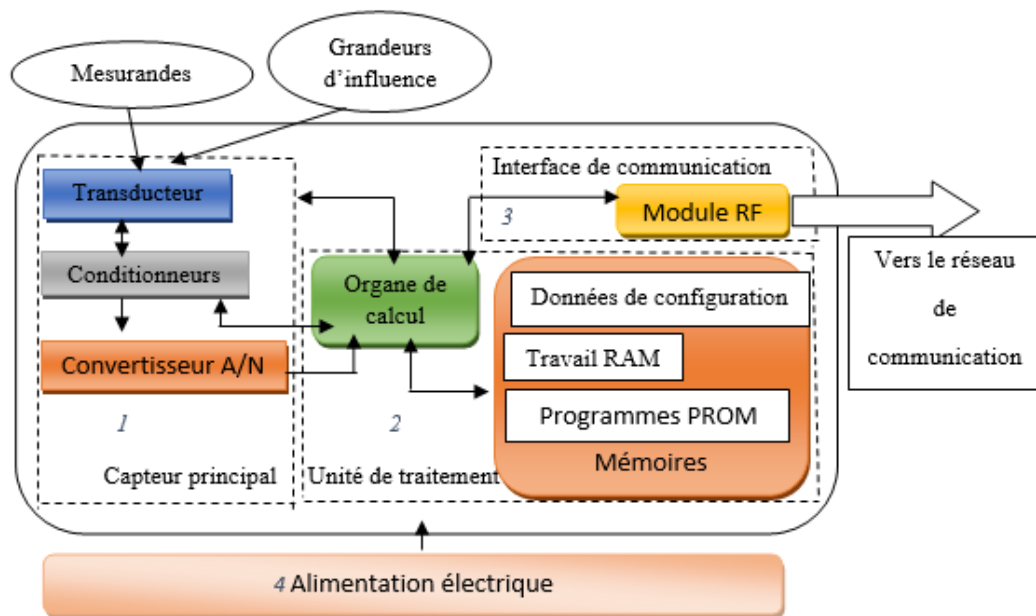


Fig. I.6. Architecture matérielle d'un capteur intelligent.

Le transducteur permet de détecter toute variation de la grandeur physique en entrée du capteur. Sa conception est étroitement liée au domaine d'application pour lequel le capteur sera utilisé.

L'interface de communication permet également au capteur intelligent de recevoir les informations du système nécessaires à l'élaboration de sa mesure et à sa validation. Elle peut également être utilisée dans les phases de calibration et de mise en service de l'équipement dans son environnement du travail.

Un capteur intelligent peut donc être considéré comme un véritable système embarqué autonome, qui devra posséder son propre système d'exploitation lui permettant de coopérer au sein d'une organisation [11].

I.7.3. Domaines d'applications des capteurs intelligents

On dénombre quatre grands domaines d'application dans lesquels les capteurs intelligents sont plus particulièrement usités :

- l'industrie manufacturière, caractérisée par des contraintes de temps-réel fortes et des contraintes environnementales variables suivant le type d'application ;
- l'industrie des « processus continus » caractérisée par des contraintes de temps-réel plutôt faibles mais qui doit satisfaire des contraintes environnementales fortes (processus chimiques, thermiques, etc.) ;
- les systèmes embarqués, qui doivent également satisfaire des contraintes temporelles très variables suivant les applications, mais des contraintes relatives à l'environnement très fortes ;
- enfin, le tertiaire, qui est le plus souvent le moins contraignant tant au niveau temps réel qu'environnemental [12].

I.7.4. Avantages de l'intelligence d'un capteur

L'intelligence du capteur intelligent réside dans sa capacité de vérification du bon déroulement d'un algorithme de métrologie.

Cette intelligence est liée à l'amélioration de performances de capteur (exactitude, temps de réponse,...) par l'accroissement de la crédibilité de la mesure. Un capteur intelligent offre des avantages spécifiques tels que [11] :

- la possibilité de configurer le capteur à distance ;
- la crédibilité accrue des mesures ;
- la coopération via un système de communication dédié en temps réel ;
- l'aide à la maintenance et à la prise de décision grâce aux informations d'état fournies ;
- la participation à la commande du système en intégrant des fonctions de commande-régulation ;
- la participation à la sécurité du système en offrant des possibilités d'alarme
- la télésurveillance.

Les fonctions d'un système aux capteurs intelligents peuvent être décrites en termes de :

- Compensation.
- Validation.
- Traitement de l'information.
- Communications.
- Intégration.

I.8. CAPTEURS LOGICIELS

Les capteurs logiciels sont un outil précieux dans de nombreux domaines industriels. Ils sont utilisés pour résoudre un certain nombre de problèmes tels que les systèmes, l'analyse d'hypothèses, la prévision en temps réel pour le contrôle de l'usine et les stratégies de diagnostic de défauts.

Dans cette partie de travail on s'intéresse à la conception d'un capteur logiciel qui permettant de fournir une information sur un paramètre à partir d'autres paramètres physicochimiques issus de capteurs physiques. L'intérêt aussi bien économique que technique, permet d'assurer un fonctionnement permanent et stable de l'analyse et la surveillance de l'état de l'eau. Le problème posé est vu comme un problème de régression [1].

I.8.1. Définition d'un capteur logiciel (soft sensor)

Le terme (capteur logiciel) combine les mots "logiciel", parce que les modèles d'évaluation des signaux de capteurs sont généralement mis en œuvre dans les programmes informatiques, et les "capteurs", parce que ces modèles fournissent des informations similaires aux capteurs matériels " [13].

Un capteur logiciel est un modèle de régression qui utilise des variables faciles à mesurer pour prédire une variable difficile à mesurer. Il fait l'objet de recherches dans de nombreux domaines. À l'origine, les capteurs logiciels ont été étudiés dans le cadre de la chimométrie, ce qui signifie des méthodes statistiques pour extraire des informations à partir d'ensembles de données qui consistent souvent en de nombreuses variables mesurées [14].

I.8.2. Propriétés des capteurs logiciels

Les capteurs logiciels offrent un certain nombre de propriétés intéressantes :

- Ils représentent une alternative à faible coût aux capteurs physiques. En effet, la disponibilité des analyseurs en ligne et des systèmes numériques qui sont à la fois utilisés pour la surveillance et le contrôle, donnent aux concepteurs et aux opérateurs les outils nécessaires à la conception et la mise en œuvre des capteurs logiciels, sans augmenter les coûts initiaux ;
 - Ils permettent la réalisation de réseaux de surveillance plus complets ;
 - Ils peuvent travailler en parallèle avec des capteurs physiques, donnant des informations utiles pour les tâches de détection de défaut, permettant ainsi la réalisation de processus plus fiables ;
 - Ils peuvent facilement être implantés dans des micro-informatiques et réajustés lorsqu'il y a un changement dans les paramètres du système ;
 - Ils permettent d'estimer en temps réel des données et de diminuer les retards introduits par des capteurs physiques lents, améliorant ainsi la performance des stratégies de contrôle [1].

I.8.3. L'utilisation des capteurs logiciels

Les capteurs logiciels peuvent être utilisés en tant que :

- Back-up des appareils de mesure : un capteur logiciel est dans ce cas spécialement conçu pour être momentanément substitué à un équipement de mesure en panne et permet ainsi d'éviter la dégradation des performances de l'usine.
- Réduction des besoins matériels de mesure : Supervision, automate programmable, base de données, les systèmes informatisés sont souvent présents dans la chaîne de production. Ces systèmes peuvent héberger les capteurs logiciels. Les experts peuvent donc être encouragés à concevoir des modèles d'inférence qui sont destinés à remplacer définitivement les capteurs physiques.
- Validation du capteur, détection et diagnostic de défauts : Les stratégies de Détection et de diagnostic exploitent toujours une certaine forme de redondance. C'est la capacité d'avoir deux ou plusieurs façons de déterminer quelques propriétés caractéristiques (variables, paramètres, symptômes) du processus. L'idée principale, à partir de laquelle découlent toutes les stratégies de détection de défaut est de comparer les informations recueillies à l'information correspondante d'une source redondante.
- Analyse d'hypothèses Le modèle, qui a permis d'élaborer le capteur logiciel, est utilisé dans ce cas pour effectuer une simulation de la dynamique du système corres-

pendant aux tendances d'entrée, dans le but d'obtenir une meilleure compréhension du comportement système. Cette utilisation particulière des modèles de processus pour effectuer une simulation est appelée analyse de scénarios [13].

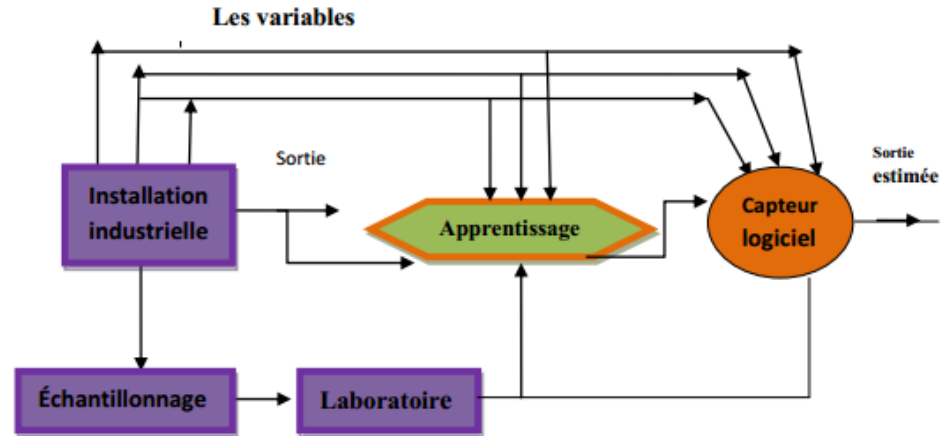


Fig. I.7. Capteur logiciel connecté à l'usine [13].

I.8.4. Les types des capteurs logiciels

Les capteurs logiciels peuvent être classés en fonction de différents critères. Sur la base de leur méthodologie de modélisation, les capteurs logiciels sont fondamentalement classés en deux catégories principales [14].

I.8.4.1. Capteurs logiciels pilotés par les données (Data-driven Soft Sensors)

Ce sont des modèles basés sur des données de la boîte noire, qui décrivent les conditions réelles d'une plante, induite par des données de mesure d'entrée-sortie observées.

Contrairement aux capteurs mous pilotés par les modèles, qui seront décrits ci-après, ils ressemblent traiter les états de façon plus réaliste et décrire les relations entre les entrées et les Sorties plus précisément.

Comme leur nom l'indique, ils sont basés sur des données mesurées, qui peut être soit prélevé sur l'installation pour laquelle le capteur souple est développé, soit à partir d'expériences spécialement conçues Les données.

I.8.4.2. Capteurs logiciels entraînés par le modèle (Model Driven Soft Sensors)

Ce sont les premiers modèles qui se concentrent sur l'état du processus, décrivant leurs origines physiques et chimiques. Ce type de capteur est principalement

basé sur la dérivation d'équations qui peut décrire les caractéristiques du processus, et à cause de cela, ils sont aussi dénommés modèles basés sur la connaissance ou modèles de boîte blanche [13].

I.8.5. Développement du capteur logiciel

Les capteurs logiciels pilotés par les données sont conçus pour constituer une alternative intéressante, non seulement aux capteurs matériels, mais également aux capteurs logiciels pilotés par les modèles. Ceci est principalement dû au coût relativement élevé pour le premier et au coût et à la complexité de ce dernier. De plus, les techniques de mesure modernes utilisées dans les installations industrielles permettent de collecter, de stocker et d'analyser des données, ce qui suggère fortement l'utilisation de capteurs logiciels basés sur les données en manipulant l'utilisation des mesures disponibles dans les bases de données d'installations. Par conséquent, la qualité d'un capteur logiciel dépend de la qualité des données utilisées pour son développement. De plus, la conception d'un capteur logiciel dépend en grande partie de l'objectif pour lequel il est conçu et des tâches à accomplir. Cela signifie que les procédures de conception de capteurs souples à des fins de sauvegarde diffèrent des celles conçues pour les prévisions en ligne. Cependant, ils partagent tous un cadre pratiquement unifié qui peut être divisé en étapes, comme indiqué dans le diagramme de la figure I.8

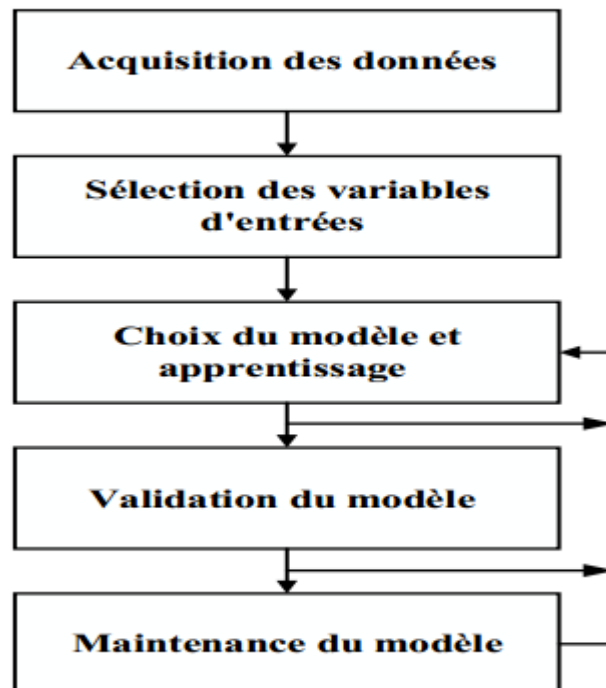


Fig. I. 8. Principales étapes de développement d'un capteur logiciel.

La conception d'un capteur logiciel piloté par les données repose principalement sur des données historiques extraites des bases de données d'installations industrielles. Dans ces usines, la production constitue l'objectif principal du processus, tandis que les mesures variables, l'enregistrement des échantillons, la récupération des données, la réconciliation et la synchronisation sont des objectifs subsidiaires. Cela pose un certain nombre de problèmes aux développeurs de capteurs logiciels, tels que l'évaluation de la qualité des données (par exemple, la fiabilité, la précision, l'exhaustivité et la représentativité), la sélection des variables et des mesures indépendantes pertinentes. En conséquence, les étapes de prétraitement des données qui traitent de ces problèmes sont essentielles à la qualité du capteur logiciel, comme indiqué dans les sections suivantes.

I.8.5.1. Acquisition des données

Les industries sont généralement tenues de stocker leurs données dans les processus. Ceci est la base de l'utilisation ultérieure de ces données pour l'optimisation du système ou d'autres méthodes liées aux données. La collecte des données et l'inspection ultérieure des données sont les premières étapes du développement d'un capteur logiciel. L'inspection des données est une étape préliminaire dans la conception de capteur logiciel, car elle aide à choisir les variables indépendantes les plus influentes, à identifier les profils de données, à classer les profils de processus et à extraire les informations pertinentes contenues dans les données. Elle est généralement effectuée en tant qu'étape initiale, car elle permet au concepteur d'obtenir une vue d'ensemble de la structure de données.

I.8.5.2. Sélection des variables d'entrées

Les caractéristiques remarquables des données acquises dans les installations de systèmes sont redondantes et peuvent être insignifiantes. Des perturbations qui corrompent les mesures sont également parfois présentes. Souvent, la quantité et la qualité des données ainsi que leur haute dimensionnalité peuvent être un facteur limitant pour le développement du capteur logiciel. Par conséquent, il est nécessaire de préparer les données avant qu'elles ne soient traitées par le modèle du capteur logiciel. La compréhension du processus et la connaissance à priori sont requises dans cette phase.

Le choix des variables d'entrée est une étape cruciale. La sélection des variables consiste à choisir les variables secondaires qui sont les plus informatives pour le processus

en cours de modélisation, ainsi que celles qui fournissent le plus généralisable. Cette étape est fondamentale, car les modèles sont construits à partir d'un nombre fini d'observations et un modèle avec trop d'entrées peut conduire à un ajustement excessif et engendrer une charge de calcul importante. En outre, décrire un processus en fonction de quelques variables sélectionnées permet de conserver l'interopérabilité. Un nombre réduit de variables présente plusieurs avantages, tels que la réduction du temps de développement du modèle, la possibilité d'agrégation de l'information sur l'interprétation physique du processus ou l'amélioration de la performance du modèle. De plus, une réduction du nombre de variables implique un nombre plus faible de capteurs réels requis, une diminution des coûts et une augmentation de la faisabilité des applications.

I.8.5.3. Choix du modèle et apprentissage

Le choix du modèle est une étape critique dans le développement du capteur logiciel. En particulier, la structure du modèle définit la tâche d'application spécifique et la sélection des paramètres du modèle détermine la capacité de généralisation du capteur logiciel. Cependant, une approche cohérente de la tâche n'existe pas.

Il existe deux approches de modèle distinctes appliquées pour le développement de capteurs logiciels. La première est basée sur des modèles en boîte blanche, obtenus par une connaissance physique du processus et la seconde est basée sur des modèles basés sur des boîtes noires ou des données, basées exclusivement sur la construction d'un modèle à partir de données empiriques du processus. La modélisation par l'approche de la boîte blanche nécessite de solides connaissances sur le processus et demande un long travail de modélisation pour construire les modèles. Pour les systèmes complexes, l'approche de modélisation boîte blanche peut être pratiquement impossible. Les modèles à boîte noire sont basés sur des observations empiriques du processus (les méthodes elles-mêmes sont des méthodes prédictives empiriques). La modélisation en boîte noire est capable de décrire les conditions réelles du processus et nécessite peu de connaissances sur le système à modéliser.

Toutefois, cela nécessite un travail intensif sur les données de processus. Certaines difficultés avec ces types d'approches sont liées à la difficulté de choisir le type et la structure du modèle, les fonctions à utiliser et la quantité de termes de fonction nécessaires au développement. Dans la modélisation en boîte noire, le premier aspect à décider est de quel type de modèle s'agira-t-il ? Les modèles choisis pour étude dans le

cadre de ce travail, sont les réseaux de neurones artificiels de types Multi-Layer-Perceptron (MLP), Radial Basis Function (RBF) et Deep Learning. Dans la majorité des applications de capteurs logiciels, un seul modèle est réglé en utilisant tous les échantillons d'apprentissage disponibles, sans distinguer les modes de fonctionnement du processus pendant la phase d'apprentissage. Cependant, l'existence de plusieurs modes de fonctionnement dans un processus est une caractéristique inhérente à la plupart des applications industrielles.

I.8.5.4. Validation du modèle

La validation du modèle est une étape très importante dans le développement du capteur logiciel, dans laquelle le concepteur estime à quel point le modèle va fonctionner sur les données invisibles. Idéalement, si suffisamment de données étaient disponibles, le développeur du capteur logiciel mettrait de côté un ensemble de validation et l'utiliserait pour évaluer un modèle dont les paramètres de base sont étalonnés sur un ensemble d'apprentissage, pour les différentes valeurs de ses méta-paramètres. Après avoir trouvé l'ensemble optimal de méta-paramètres, le développeur calibrerait alors le modèle pour définir ses paramètres de base, en utilisant toutes les données d'entraînement disponibles. Le modèle résultant est finalement évalué sur un ensemble de données de test indépendant. L'objectif de l'étape de validation du modèle est d'évaluer la capacité du modèle formé à effectuer une généralisation à de nouveaux échantillons. La précision de la généralisation peut également être utilisée comme estimateur pour le classement d'un modèle dans une approche de sélection de variables.

I.8.5.5. Maintenance du modèle

Lors de la conception d'un capteur logiciel, les données historiques du processus sont utilisées pour apprendre le modèle de capteur logiciel. Cependant, les données historiques contiennent des informations limitées, correspondant à une période de temps limitée, et pouvant également se concentrer sur un ensemble limité de zones d'opérations de l'espace d'état. Lorsqu'il s'agit de nouveaux événements, non décrits dans les données historiques, le capteur logiciel tend à diminuer ses performances. Dans ce contexte, et pour surmonter une telle détérioration des performances, l'objectif de la maintenance du capteur logiciel est de maintenir une bonne réponse du capteur logiciel même en présence de variations de processus,

ou de certaines modifications de données. Généralement, cela se fait en mettant à jour le modèle de capteur logiciel en ligne / récursivement, en mode batch ou échantillon, en utilisant les échantillons entrants du processus (dans ce contexte, les capteurs logiciels sont appelés "capteurs logiciels adaptatifs") [14].

I.8.6. Applications des capteurs logiciels

Les capteurs logiciels ont une large gamme d'applications et peuvent remplir une grande variété de tâches. Cela est principalement dû à leur coût relativement faible par rapport à leurs homologues matériels ; de plus, les capteurs matériels impliquent des coûts d'acquisition, d'installation, de fonctionnement et de maintenance. Les capteurs logiciels deviennent des outils courants, non seulement en tant que source d'information pour les opérateurs de salle de contrôle, mais également pour le contrôle industriel en boucle fermée. A cet effet, il a été rapporté dans qu'il y a dix ans, 80% des capteurs logiciels étaient déployés dans des applications en boucle ouverte, alors que 20% étaient utilisés dans des applications en boucle fermée. Actuellement, 80% de ces capteurs sont plutôt utilisés dans des applications de contrôle en boucle fermée.

Le domaine d'application dominant des capteurs logiciels est la prédiction en ligne de variables de processus difficiles à mesurer. Cependant, d'autres domaines d'application incluent la surveillance de processus et la détection de défauts.

I.8.6.1. Prédiction en ligne

L'application la plus courante des capteurs logiciels est la prédiction de valeurs ne pouvant pas être mesurées en ligne à l'aide de mesures automatisées. Cela peut être dû à des raisons technologiques (par exemple, aucun équipement n'est disponible pour la mesure requise), ou à des raisons économiques (par exemple, l'équipement nécessaire est trop coûteux), etc. D'ailleurs, la prévision en temps réel des variables cibles de processus est un domaine d'application qui a fait augmenter l'usage des capteurs logiciels comme outils d'optimisation essentiels dans les processus industriels. Contrairement à cette approche, des échantillons sont envoyés aux laboratoires pour une analyse hors ligne, ce qui crée des retards importants, ou des variables sont mesurées en ligne par un analyseur, ce qui crée des retards et des problèmes de fiabilité.

Les capteurs logiciels développés pour les prédictions en ligne sont basés sur des approches d'apprentissage supervisées ; les données d'apprentissage comprennent à la fois des données d'entrée et des mesures de variable cible correspondante.

I.8.6.2. Surveillance de processus

La surveillance des processus est un autre domaine d'application des capteurs logiciels. La surveillance du processus peut être une tâche d'apprentissage non supervisée ou de classification binaire. Les systèmes peuvent être formés pour décrire/analyser l'état de fonctionnement normal ou pour reconnaître d'éventuels défauts de processus.

La qualité des produits et la sécurité des processus sont deux critères principaux pris en compte lors de la conception des processus industriels. Il est souvent impossible d'avoir un processus entièrement surveillé, car cela nécessite la construction d'usines fortement instrumentées et à coût élevé. Par conséquent, les capteurs logiciels sont considérés comme une alternative permettant de surveiller la sécurité des processus et d'améliorer la qualité du produit.

I.8.6.3. Détection et reconstruction de défauts de capteurs

Comme déjà mentionné, les usines chimiques modernes sont équipées d'un grand nombre de capteurs. Parmi ces nombreux paramètres de processus, certains sont des paramètres clés sans lesquels les opérateurs de processus auront du mal à maintenir la qualité et le débit du produit. Étant donné que ces paramètres de processus sont mesurés par des capteurs physiques, il est toujours possible que ces capteurs échouent. Une fois qu'un capteur physique défectueux est détecté et identifié, il peut être reconstruit ou le capteur peut être remplacé par un capteur logiciel, formé pour agir en tant qu'élément de secours du dispositif de mesure câblé. Si le capteur logiciel s'avère efficace en remplacement du capteur physique, il fournira alors à l'opérateur une indication en attente des variables de processus. Dans certains cas, les capteurs souples robustes peuvent remplacer l'appareil de mesure même s'il est en état de fonctionnement. L'outil logiciel peut donc facilement être mis à jour, il ne présente pas de défaillance mécanique, est facile à mettre en œuvre dans le système de contrôle et peut donc fournir un avantage financier substantiel [14].

CONCLUSION

Le domaine du développement des capteurs logiciels a récemment gagné en importance considérable avec la création des outils de calcul efficaces et facilement utilisables. Les capteurs logiciels (Soft Sensors) conçus à base d'algorithmes inspirés des techniques de l'intelligence artificielle, sont devenus un moyen efficace et très attractif. Ils sont considérés comme alternative des capteurs physiques. Le chapitre suivant va discuter les techniques d'apprentissages statistiques les plus connues.

CHAPITRE II

Les Techniques d'apprentissages statistiques

CHAPITRE II

Les Techniques d'apprentissages statistiques

INTRODUCTION

Le Deep Learning (DL) est une nouvelle technique de recherche du Machine Learning (ML) qui a été introduite dans le but de rapprocher le ML de son objectif principal qui est l'automatisation des procédés par des modèles plus similaires au concepts et raisonnement de l'être humain. Il concerne les algorithmes inspirés par la structure et le fonctionnement du cerveau humain. Ils peuvent apprendre plusieurs niveaux de représentation dans le but de modéliser des relations complexes entre les données [15].

II.1. RESEAUX DE NEURONES

L'idée de construire des réseaux de neurones (RN) artificiels n'est pas neuve, elle remonte à la fin des années 50. Très schématiquement l'idée consiste à s'inspirer du fonctionnement du cortex visuel des animaux. Dans une version élémentaire, chaque neurone i d'un tel réseau possède un niveau d'activation x_i compris entre 0 et 1. Le schéma d'interconnexion entre neurones définit l'architecture du réseau. Une architecture classique consiste à organiser les neurones en couches successives avec des interconnexions limitées aux couches adjacentes comme le montre la figure II.1(a).

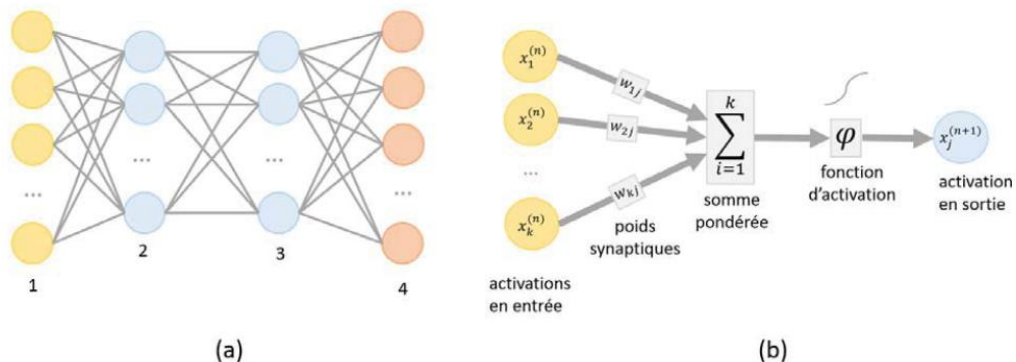


Fig. II.1. (a) un RN organisé en 4 couches, (b) le mécanisme d'activation d'un neurone [16].

Dans le cadre d'un apprentissage supervisé, un exemple classique d'utilisation d'un RN est celui d'un système chargé de classer des images de chiffres manuscrits. Dans cet exemple les niveaux d'activations x_i des neurones $i = 1, \dots, k$ de la couche d'entrée correspondent aux niveaux de gris des pixels de l'image, k étant le nombre de pixels des

images. La couche de sortie est en l'occurrence constituée de neurones $y_j, j = 0, 1, \dots, 9$ qui correspondent aux dix chiffres que l'on peut attribuer à chaque image d'un ensemble d'entraînement. Les niveaux d'activation des neurones sont déterminés récursivement, couche par couche. Ceux de la couche $n + 1$ sont calculés à l'aide d'une fonction d'activation φ en fonction des niveaux d'activation des neurones de la couche n pondérés par certains poids synaptiques W_{ij} :

$$x_j^{(n+1)} = \varphi\left(\sum_{i=1}^k W_{ij}x_i^{(n)}\right) \quad (\text{II.1})$$

La somme porte sur tous les neurones i de la couche n connectés au neurone j de la couche $n + 1$, voir la figure II.1(b). La fonction d'activation φ est typiquement une fonction parmi les fonctions représentées dans le tableau II.1

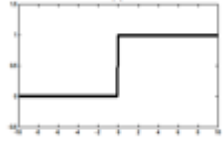
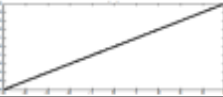
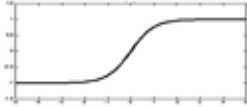
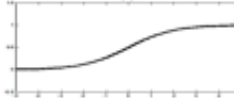
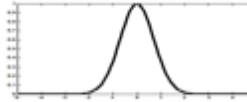
Fonction d'activation	Equation	1D graphe
Echelon(Seuil)	$\varphi(x) = \begin{cases} 1 & \text{si } x \geq 0 \\ 0 & \text{sinon} \end{cases}$	
Linéaire	$\varphi(x) = x$	
Tangente hyperbolique	$\varphi(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$	
Sigmoïde	$\varphi(x) = 1/(1 + e^{-x})$	
Radiale gaussienne	$\varphi(x) = \sqrt{1/2\pi} \exp(-x^2/2)$	

Tableau. II.1. Les fonctions d'activations.

L'entraînement du réseau consiste à trouver des poids synaptiques W_{ij} tels que la couche de sortie permette de classer avec précision les images d'un ensemble

d'entraînement. On espère naturellement que le RN présentera des capacités de généralisation sur des exemples qu'il n'a jamais rencontrés.

Une question se pose d'emblée : « De tes poids W_{ij} existent-ils toujours quel que soit l'objectif assigné au RN ? ». Par chance, un résultat mathématique connu sous le nom de théorème d'approximation universel, garantit que la chose est effectivement possible, même pour un réseau ne comportant qu'une seule couche cachée, à condition toutefois que φ soit non-linéaire et qu'un nombre suffisant de neurones soient mis en jeu en fonction de la marge d'erreur tolérée. En revanche si l'on se limite à des fonctions d'activation φ linéaire, le réseau fonctionnera comme une régression linéaire ordinaire et ne sera par conséquent d'aucune utilité pour prédire des phénomènes non-linéaires complexes.

Reste à construire un algorithme qui fournit une bonne approximation des poids W_{ij} en un temps acceptable.

Jusqu'à il y a peu l'algorithme phare pour l'entraînement des systèmes de neurones multicouches était l'algorithme dit de rétro-propagation. Pour fixer les idées, restons sur l'exemple des digits. Définir une fonction de coût C qui mesure l'écart entre les prédictions $y_j(x)$ des neurones de sorties et les valeurs cibles t_j spécifiées dans l'ensemble d'entraînement :

$$C(X; W) = \sum_{j=0}^9 (y_j(X) - t_j)^2 \quad (\text{II. 2})$$

Naturellement $y_j(x)$ est une fonction extrêmement compliquée de la configuration x en entrée et des poids w , spécifiée par l'itération de (1). Ce que l'on cherche à minimiser en principe c'est la somme ou la moyenne $C_{MSE}(w)$ de ces erreurs sur toutes les configurations x de l'ensemble d'entraînement ε :

$$C_{MSE}(w) = \sum_{X \in \varepsilon} C(X; W) \quad (\text{II. 3})$$

Une descente de gradient consiste à calculer la direction dans l'espace des poids W dans laquelle la décroissance de $C_{MSE}(w)$ est maximale. Cette direction est naturellement donnée par l'opposé du gradient $\nabla C_{MSE}(w)$. Si tout se passe bien, c.à.d. qu'il n'y a pas de minima locaux, on s'approchera du minimum de $C_{MSE}(w)$ par itérations successives de petites corrections apportées à w :

$$W^{(new)} = W^{(old)} - \alpha \nabla C_{MSE}(W^{(old)}) \quad (\text{II. 4})$$

Le coefficient α permet d'ajuster la vitesse d'apprentissage, le cas échéant dynamiquement. Dans la pratique cependant, le calcul de la somme sur toute les configurations $X \in \varepsilon$ est impossible car beaucoup trop couteux en temps de calcul. Pour remédier à cette situation on utilise une descente de gradient stochastique (SGD) qui revient à approximer à chaque étape le gradient $\nabla C_{MSE}(W)$ par le gradient $\nabla C(X; W)$ d'un seul échantillon X tiré au hasard dans l'ensemble d'entraînement ε . Cette stratégie qui peut paraître audacieuse a priori fonctionne car, intuitivement, les erreurs occasionnées par cette approximation se compensent sur le long terme. L'algorithme de SGD a fait ses preuves dans de nombreux problèmes d'optimisation et des résultats théoriques viennent étayer cette intuition dans certains cas particuliers.

Reste à calculer toutes les composantes $\partial C(X; W)/\partial W_{ij}$ du gradient $\nabla C(X; W)$. Le calcul est simple dans son principe puisqu'il ne s'agit ni plus ni moins que de calculer la dérivée d'une fonction composée un peu compliquée définie récursivement par (1) et (2). C'est là qu'intervient l'algorithme de rétro-propagation, celui-ci permet d'organiser efficacement ce calcul de dérivée. Il s'avère que les dérivées par rapport aux W_{ij} associés à la couche de sortie sont élémentaires. Le reste du calcul procède par introduction car on montre en effet que les dérivées par rapport aux W_{ij} de la couche $n - 1$ sont calculables dès lors celles de la couche n ont déjà été calculées. Ce calcul à reculons est à l'origine du nom de l'algorithme.

Tout astucieux qu'il soit l'algorithme de rétro-propagation souffre cependant de deux inconvénients majeurs :

1. L'expérience montre que le temps d'entraînement d'un RN croit rapidement lorsque le nombre de couches augmente. C'est d'ailleurs l'une des raisons pour lesquelles à partir des années 1990 les RN ont été remisés au profit d'autres algorithmes-non- linéaires moins gourmand en ressources.
2. Les RN n'échappent pas au problème central du Machine Learning : le surapprentissage.

Ces résultats empiriques sont en partie corroborés par des résultats en théorie de la complexité qui démontrent que l'entraînement d'un RN est un problème complexe dans un sens précis.

Pour progresser, de nouvelles idées sont nécessaires. Après plusieurs décennies de stagnation c'est G.E Hinton et son équipe qui, en 2006, ont fait la principale percée dans ce domaine [16].

II.2. INTRODUCTION SUR DEEP LEARNING

Depuis 2006, le domaine a fait de grandes avancées avec l'apprentissage de réseaux profonds (deep learning). Cette approche permet, à partir de données d'entrée, d'en extraire une représentation plus riche. Les réseaux profonds sont composés de plusieurs couches de neurones. Chaque couche est une étape qui représente les données de façon un peu plus complexe (abstraite) en se basant sur ce qui a été appris dans la couche précédente. Cette approche est à l'image de l'apprentissage humain qui commence par apprendre des concepts simples, comme l'addition et la soustraction en mathématiques, pour ensuite se baser sur ces concepts afin d'en apprendre des plus complexes, comme la multiplication et le principe de fonction [17]. (La Figure II.2)

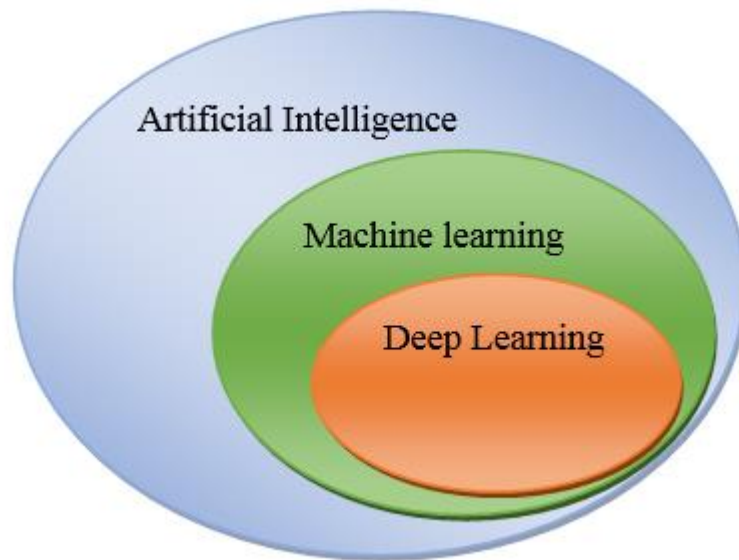


Fig. II.2. La relation entre l'intelligence artificielle, le ML et le deep Learning.

II.3. DEFINITION DE L'APPRENTISSAGE EN PROFONDEUR

L'apprentissage en profondeur (Deep learning : DL) est une branche de l'apprentissage automatique qui enseigne aux ordinateurs à faire ce qui vient naturellement aux humains : apprendre de l'expérience. Les algorithmes d'apprentissage automatique utilisent des méthodes de calcul pour "apprendre" des informations directement à partir de données sans s'appuyer sur une équation prédéterminée comme modèle.

DL est une classe de techniques d'apprentissage automatique qui exploitent de nombreuses couches de traitement d'informations non linéaires pour l'extraction et la transformation de fonctions supervisées ou non supervisées, et pour l'analyse et la classification de modèles. Les caractéristiques et les concepts de niveau supérieur sont donc

définis en termes de niveaux inférieurs, et une telle hiérarchie de caractéristiques est appelée architecture profonde. La plupart de ces modèles sont basés sur un apprentissage non supervisé des représentations [18].

II.4. HISTOIRE DU DEEP LEARNING

Année	Contributeur	Contribution
2006	Geoffrey Hinton	Introduction des deep Belief network
2009	Salakhutdinov and Hinton	Introduction des deep Boltzmann machines
2012	Alex Krizhevsky	Introduction de AlexNet qui remporta le challenge ImageNet

Tableau. II.2. Les étapes majeures du Deep Learning [15].

II.5. DOMAINES D'APPLICATION DE DEEP LEARNING

L'apprentissage en profondeur investit progressivement notre quotidien :

- La reconnaissance vocale
- Le tagging automatique de morceaux de musique
- La synthèse vocale avancée
- L'étiquetage automatique d'image
- La conception de nouvelles molécules pharmaceutiques

Toutes ces applications mettent aujourd'hui en œuvre des techniques de Deep Learning [16].

II.6. DIFFERENTS TYPES DE MODELES DE DEEP LEARNING

Il existe un grand nombre de variantes d'architectures profondes. La plupart d'entre elles sont dérivées de certaines architectures parentales originales. Il n'est pas toujours possible de comparer les performances de toutes les architectures, car elles ne sont pas toutes évaluées sur les mêmes ensembles de données. Le Deep Learning est un domaine à croissance rapide, et de nouvelles architectures, variantes ou algorithmes apparaissent toutes les semaines [15].

II.6.1. Réseaux de neurones convolutionnels

Les réseaux de neurones convolutionnels (CNN) sont un type de réseau de neurones artificiel [19]. Les premiers CNN ayant eu du succès ont été inventés en 1989 par LeCun dans et appliqués à la reconnaissance d'écriture manuscrite [20].

Les réseaux de neurones convolutionnels sont des structures de réseaux de neurones particulières puisque l'opération élémentaire n'est plus une multiplication matricielle mais une convolution. Ces réseaux ont été inventés afin de tirer parti de données possédant une structure (spatiale, temporelle, ...) telles que des images, des vidéos, ou encore des signaux temporels. Pour des images, il est courant d'avoir une entrée de taille $256 \times 256 \times 3$ qui correspond à une image classique sur Internet. Historiquement, des grandeurs ou attributs étaient calculés sur cette image pour servir d'entrées au réseau de neurones. Mais si l'entrée est directement une image soit une liste de presque 200000 entiers, l'utilisation de couches complètement connectées (décrite précédemment) est à proscrire et serait une perte de temps de calcul et de poids à apprendre. Les couches de convolutions permettent de mieux rendre compte de la distribution spatiale grâce à une sensibilité locale tout en maintenant un nombre de poids à estimer raisonnable.

Leur fonctionnement est simple et basé sur une convolution. La sortie est obtenue par convolution de l'image d'entrée et d'un nombre fixé de filtres. La sortie d'un filtre est obtenue en multipliant le contenu d'une fenêtre glissante appliquée à l'image par le filtre (Figure II.3).

Une convolution prend en entrée une image I de taille $W \times H \times B$ et un noyau de convolution G de taille $F \times F \times B$ et renvoie le résultat $F = I * G$ (où $*$ symbolise la convolution) défini par :

$$F(i, j) = \sum_{k=1}^B \sum_{m=1}^F \sum_{n=1}^F I(i+m, j+n, k) G(i, j, k) \quad (\text{II.5})$$

De manière générale, les résultats de plusieurs filtres sont concaténés pour former ce que l'on appelle couramment une carte de features. Dans ce cas-là, le nombre de paramètres à estimer peut se calculer de la manière suivante pour K filtres :

Paramètres par filtres : $F \times F \times D$, pour un total $F \times F \times D \times K$ poids et K biais.

Des couches de pooling ou de sous échantillonnage sont également ajoutées aux opérations élémentaires.

Elles permettent de décroître la taille de la représentation avec un sous échantillonnage spatial : de cette manière, elles permettent de contrôler le sur-apprentissage. Le sous échantillonnage est appliqué de manière indépendante sur chaque carte d'activation (sortie correspondant à un filtre). Par exemple, dans une configuration classique, l'utilisation d'un sous échantillonnage par maximum (Max Pooling) dans un voi-

sinage de 2×2 avec un pas de 2 permet de réduire la taille de l'entrée par 4 (par 2 dans chacune des dimensions).

Supposons que l'entrée est une matrice I de taille $H \times W$, le voisinage du pooling et le pas salors le résultat du max pooling F est défini comme suit :

$$F(i, j) = \max(\{I(i * s + m, j * s + n)\}_{m=1...F, n=1...F}) \tag{II. 6}$$

L'opération consiste simplement à calculer le maximum d'une fenêtre glissante de pass . Le max pooling est appliqué de manière indépendante sur chaque bande de l'image si l'image possède plusieurs bandes et les résultats sont concaténés pour former la sortie.

L'enchaînement de couches convolutionnelles, de couches de pooling et enfin de couches connectées est souvent désigné par le terme d'architecture de type ConvNet (LeCun Y., et al. 1998). Une architecture typique suit le schéma suivant :

$$[\text{INPUT}, (\text{CONV}, \text{POOL})^k, (\text{FC})^l]$$

$k = 2, l = 2, [\text{INPUT}, \text{CONV}, \text{POOL}, \text{CONV}, \text{POOL}, \text{FC}, \text{FC}]$ (LeCun Y., et al. 1998)

De la même manière que pour les réseaux de neurones, la dernière couche connectée a autant de sorties que de classes à prédire et est utilisée pour faire la classification [21].

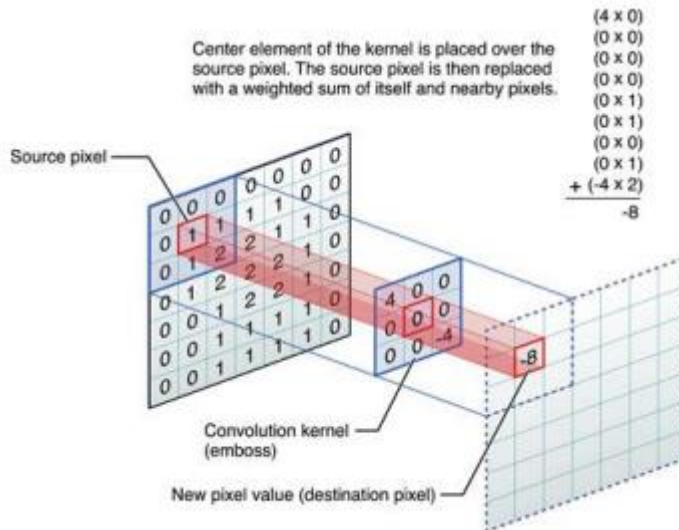


Fig. II.3. Fonctionnement d'une convolution sur une image [21].

II.6.2. Deep Belief Network (DBN)

Est une architecture de RN utilisée pour réaliser le modèle génératif de Hinton [22]. Elle contient deux parties représentées dans la figure II.4 (ici, l'exemple de la reconnaissance des chiffres a été pris juste pour expliquer le fonctionnement de ce type de réseau).

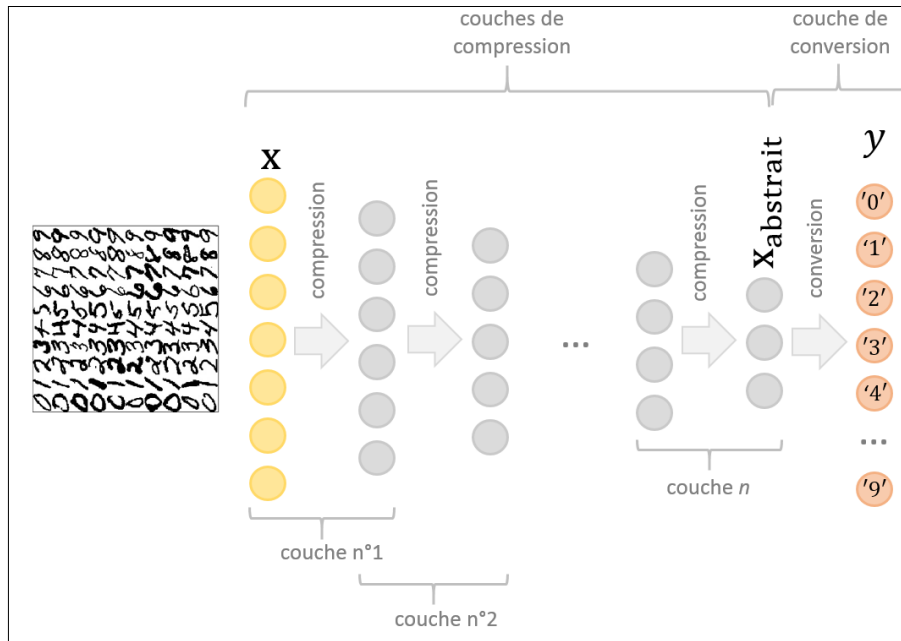


Fig. II.4. Un Deep Belief Network qui identifie des images de digits [22].

La première partie du DBN est constituée d'un ensemble de couches de compression qui convertissent les données entrées X en une représentation abstraite $X_{abstrait}$. La seconde partie convertit cette représentation en labels de classification y .

La première partie a pour objectif d'apprendre la distribution des données X présentées en entrée sans tenir compte des labels y . Elle est constituée d'une succession de couches dont chacune contiendra une représentation plus abstraite (ou compressée) que la précédente. Hinton et al. ont par ailleurs découvert un algorithme rapide pour entraîner. Une fois entraînée, cette première section du RN contiendra une représentation hiérarchique des données en entrée, la dernière couche encodant la représentation $X_{abstrait}$ la plus abstraite et aussi, c'est l'idée, la plus utile. Cette première phase peut être conçue comme une initialisation efficace du DBN, on l'appelle aussi pré-entraînement.

Le rôle de la seconde partie du DBN est de convertir la représentation abstraite et obscure $X_{abstrait}$ en labels y utilisables par exemple dans le cadre d'un apprentissage su-

pervisé. Dans l'exemple des digits, cette représentation sera constituée d'une couche de sortie de dix neurones, un neurone par digit.

L'entraînement du DBN est considéré comme achevé lorsque la performance du DBN évaluée sur un ensemble de validation distinct de l'ensemble d'entraînement ne progresse plus significativement. Cette seconde étape est appelée le fine-tuning, elle est généralement beaucoup plus lente que l'initialisation.

II.6.3. Auto-encodeurs (Auto-encoders)

L'auto-encodeur est un modèle permettant de faire une compression de l'entrée (encodeur) et une décompression de celle-ci (décodeur) [23].

- Objectif : compresser tout en gardant l'erreur $\|X - \hat{X}\|^2$ faible.
- Poids du décodeur liés aux poids de l'encodeur (habituellement, transposé).

II.6.3.1. Architecture

L'Auto-encodeur est un type spécial du DNN (Deep Neural Networks) sans classe étiquetée, dont les vecteurs de sortie ont la même dimensionnalité que les vecteurs d'entrées.

Il est souvent utilisé dans l'encodage de données. Un auto-encodeur a typiquement une couche d'entrée (couche $L1$) qui représente les vecteurs de données ou de caractéristique, une ou plusieurs couches cachées qui représentent la caractéristique transformée (couche $L2$) et une couche de sortie qui correspond à la couche d'entrée (couche $L3$). Lorsque le nombre de couches cachées est supérieur à un, l'auto-encoder est considéré comme profond. La dimension des couches peut être soit plus petite (lorsque l'objectif est la compression) ou grande (lorsque l'objectif est d'augmenter la dimension d'espace) (voir La Figure II.5) [24].

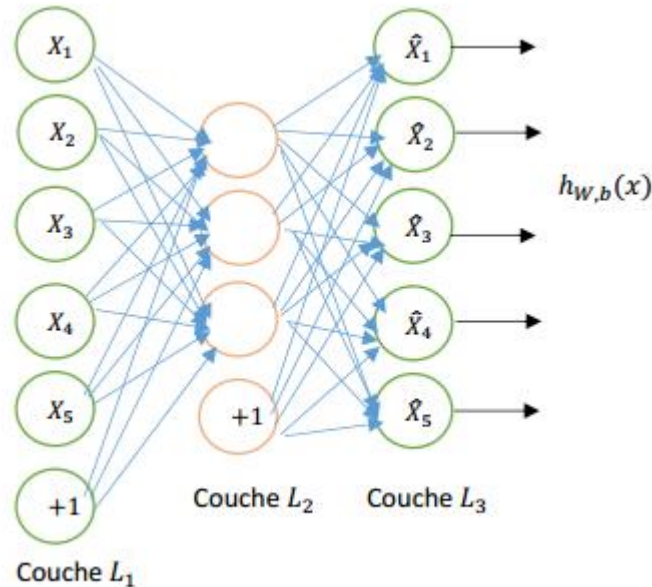


Fig. II.5. L'architecture des Auto-encodeurs [24].

II.6.3.2. Entraînement d'auto-encodeurs

- Auto-encodeur entraîné de façon non supervisée, pour apprendre représentation.
 - ✓ Encodeur utilisé pour extraire une représentation compacte.
- Entraînement vorace, une couche à la fois.
 - ✓ Entraînement de la couche la plus externe.
 - ✓ Ajout d'une nouvelle couche, qui est entraînée individuellement, couche externe étant fixée, et ainsi de suite.
- Fonction de transfert non linéaire entre les couches.
 - ✓ Nécessaire, sinon plusieurs couches non linéaires pourraient se simplifier en une seule couche.
 - ✓ Apprentissage des poids par descente du gradient (mais utilisation d'une ACP serait possible).
- Couche de sortie ajoutée à l'encodeur, avec entraînement supervisé.
 - ✓ Entraînement complet de la couche de sortie par rétropropagation.
 - ✓ Ajustement des poids de l'encodeur par rétropropagation (fine-tuning).

Exemples d'entraînement d'un auto-encodeur :

Exemple 1 :

La figure II.6 présente un apprentissage non supervisé d'un auto-encodeur.

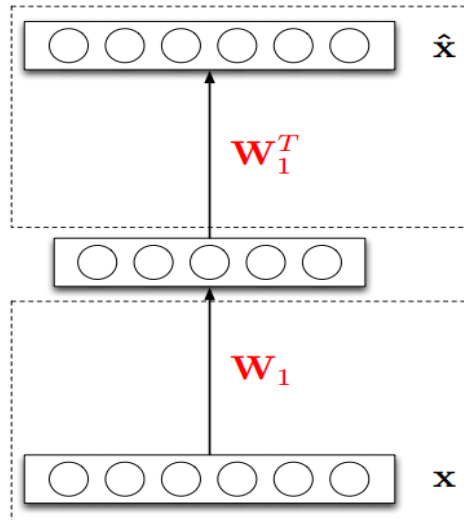


Fig. II.6. Un apprentissage non supervisé d'un auto-encodeur [23].

- entraînement non supervisé du Poids W_1 , poids W_1^T lié ;
- minimise erreur $\|X - \hat{X}\|^2$;
- représentation intermédiaire dans valeurs centrales.

Exemple 2 :

La figure II.7 présente un apprentissage non supervisé multicouches d'un auto-encodeur.

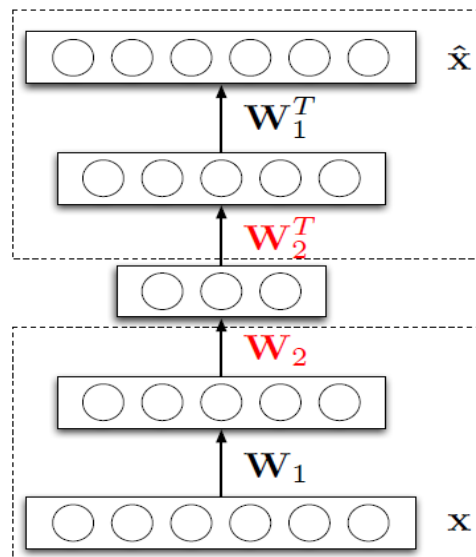


Fig. II.7. Un apprentissage non supervisé multicouches d'un auto-encodeur [23].

- ajout de deux nouvelles couches (une dans encodeur et une dans décodeur) ;
- entraînement non supervisé du poids W_2 , poids W_1 fixés ;

- minimise toujours erreur $\|X - \hat{X}\|^2$;
- nouvelle représentation intermédiaire ;
- peut-être répète ainsi sur plusieurs couches.

Exemple 3 :

La figure II.8 présente un apprentissage supervisé d'un auto-encodeur [23].

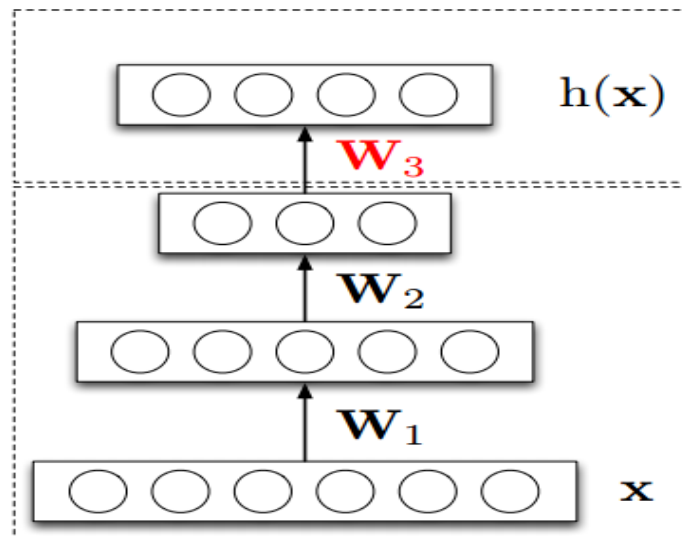


Fig. II.8. Un apprentissage supervisé d'un auto-encodeur [23].

- Retrait de la partie décodeur du réseau ;
- Ajout d'une couche de sortie, avec autant de sorties que de classe ;
- Entraînement supervisé de W_3 par rétropropagation ;
- Poids W_1 et W_2 souvent également ajustés finement par rétro-propagation (fine-tuning).

II.6.3.4. SAE-DNN (Stacked autoencoder - Deep Neural Network)

Le SAE-DNN consiste à deux types de réseau de neurones profond à savoir le SAE et le DNN. Le SAE est constitué de plusieurs couches d'auto-encodeurs dans lesquels les sorties de chaque couche sont reliées aux entrées de la couche suivante. Ces auto-encodeurs entraînent les couches cachées de DNN l'un après l'autre. L'apprentissage de DNN comprend deux étapes (figure II.9). À la première étape, l'« auto-encodeur 1 » subit un apprentissage non supervisé. Ensuite, la « couche h1 » de DNN est initialisée par les poids d'« auto-encodeur 1 » après son apprentissage. Par la suite, les poids de «

couche h_1 » de premier auto-encodeur deviennent les entrées de la seconde (couche h_2) et ainsi de suite. À la deuxième étape, l'apprentissage supervisé de DNN s'amorce [24].

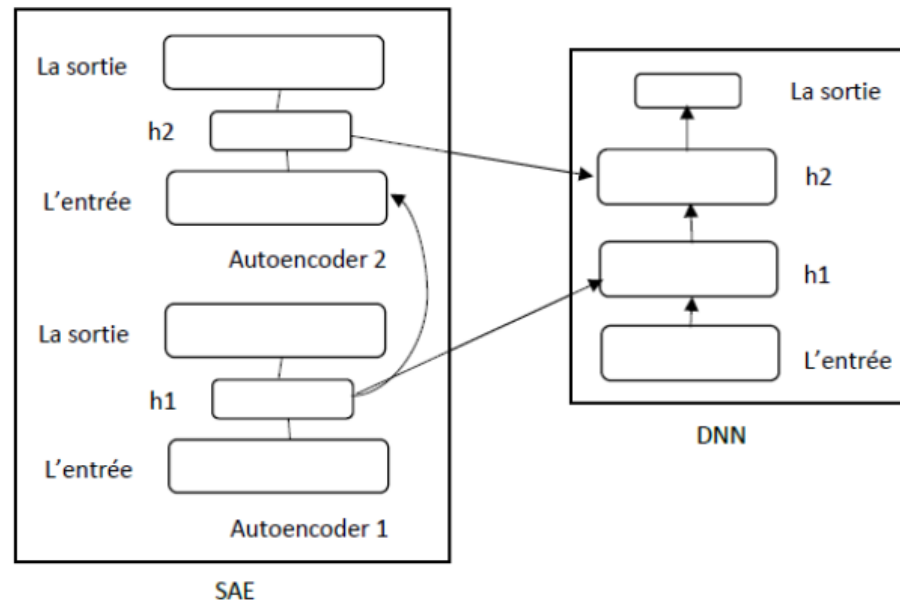


Fig. II.9. L'architecture de SAE-DNN [24].

CONCLUSION

Dans ce chapitre nous avons présenté les notions de base de RN notamment son principe de fonctionnement, ses composantes principales et également ses limitations. Ensuite, nous avons décrit une nouvelle variante de RN qui s'appelle Deep Learning (DL). Cette technique se caractérise par son habilité de résoudre le problème de la complexité de l'entraînement de (RN) ainsi que son pouvoir de représenter les formes (les entrées) d'une manière puissante, automatiques et discriminante.

Finalement, nous avons discuté les différents modèles de Deep Learning à savoir le CNN, le DBN, et l'Auto-encodeur. Ce dernier a été bien détaillé car il va faire l'objet de plusieurs expériences dans le chapitre suivant.

CHAPITRE III

Résultats Et Discussion

CHAPITRE III

Résultats et Discussion

INTRODUCTION

Ce dernier chapitre est consacré à une étude en simulation et vise principalement l'évaluation des performances des techniques proposées comme étant une solution dans la conception de capteurs logiciels utilisés dans le domaine de surveillance de la qualité des eaux propres. Donc il est de valider et d'évaluer les performances de chacune des méthodes présentées (MLP, RBF et DL) en termes de temps d'apprentissage et d'autres paramètres de régression et de classification est alors présentée les exigences principales d'efficacité sont formulées sur deux points essentiels à savoir, les tests de spécification qui vérifient que le programme réalise bien la tâche pour laquelle il a été conçu, et les tests de performances qui vont servir à mesurer l'efficacité avec laquelle cette tâche est remplie. Afin de mener une étude comparative permettant un choix décisif de la méthode la mieux adaptée à l'application indiquée. L'objectif est de mettre en place un système multi capteur à base de capteurs logiciels afin de prendre une décision adaptée au contrôle et au suivi des problèmes de la qualité de l'eau. Un exemple d'application est prévu à cet effet ; il s'agit de développer un capteur logiciel d'oxygène vu comme un problème de régression, permettant de contribuer au contrôle de la potabilité de l'eau, vu cette fois comme un problème de classification.

III.1. DESCRIPTION DE LA BASE DE DONNÉES

Nous cherchons à décider sur la qualité de l'eau à travers ses paramètres descripteurs. Nous n'avons en fait aucune connaissance a priori sur un type de modèle représentant parfaitement ce procédé, par contre nous pouvons porter notre jugement sur la qualité de cette eau à partir de quelques données descriptives. Il y a dix paramètres physico-chimiques qui sont souvent utilisés dans plusieurs travaux, et qui renseignent sur les dangers majeurs qu'il faut surveiller. Ces paramètres sont résumés comme suit : le pH, Température (T°), Magnésium (Mg^{2+}), Conductivité (C), chlore (Cl^-), calcium (Ca^{2+}), Bicarbonate (HCO_3^-), Matières en suspension (SM), Demande biochimique d'oxygène (DBO) en plus d'Oxygène dissous (OD) L'objectif qui se

trouve derrière la collecte des données relatives à ces paramètres est de Trouver un modèle de classification permettant de distinguer deux états bien distincts de l'eau (état potable et état non potable) dans un premier temps, et trois états différents par la suite après sélection de la technique adoptée. La qualité de cette eau reflétée par sa potabilité repose en fait sur une corrélation qui ne peut être identifiée que statistiquement.

III.2. PRÉTRAITEMENT

III.2.1. Normalisation des données

La normalisation des données permet de s'affranchir des différences de « normes » des variables. En effet, des variables avec des grandes valeurs peuvent avoir une plus grande Influence que des variables avec des petites valeurs, sans pour autant être plus significatives. La normalisation est un outil de prétraitement de données utilisé dans le système d'exploration de données Un attribut d'un ensemble de données est normalisé en mettant ses valeurs à l'échelle afin qu'elles se situent dans une plage spécifiée de petite taille, telle que 0 à 1. La normalisation est particulièrement utile pour les algorithmes de classification et de régression. Dans le but d'optimiser les modèles de prédiction, nous introduisons la normalisation des données par la méthode min-max. Cette technique permet de fournir une transformation linéaire sur la plage de données originale. Le rééchelonnement est souvent accompli en utilisant une formule d'interprétation linéaire, tels que :

$$xi_N = \frac{xi - \min(X)}{\max(X) - \min(X)} \quad (\text{III.1})$$

Avec :

xi_N : La valeur normalisée ;

$X = [x_1, x_2, \dots, x_i]$: Le vecteur à normalisé ;

$\min(X)$: La valeur minimale de X ;

$\max(X)$: La valeur maximale de X .

Lorsque la normalisation min-max est appliquée, chaque entité se trouvant dans la nouvelle plage de valeurs restera la même. La normalisation min-max a l'avantage de préserver exactement toutes les relations dans les données.

III.3. CRITÈRE D'ÉVALUATION

Pour une évaluation des performances de modèle d'apprentissage définissons quelques mesures statistiques

Soient :

- Racine de l'erreur quadratique moyenne (REQM)

$$REQM = \left(\frac{1}{n} \sum_{i=1}^n e^2(i) \right)^{\frac{1}{2}} \quad (\text{III.2})$$

- Erreur absolue moyenne (EAM)

$$EAM = |e(i)| \quad (\text{III.3})$$

- Erreur quadratique moyenne de généralisation (EQMG)

$$EQMG = \frac{1}{n} \sum_{i=1}^n e^2(i) \quad (\text{III.4})$$

Où : $e(i) = y_r(i) - y_c(i)$

tel que :

y_r : Sortie réelle

y_c : Sortie calculée

- Coefficient de corrélation (R) :

$$R = \frac{Cov(X,Y)}{\sigma_X \sigma_Y} \quad (\text{III.5})$$

Où :

$Cov(X, Y)$: La covariance des variables X et Y.

σ_X et σ_Y : écarts types.

III.4. DÉVELOPPEMENT D'UN CAPTEUR LOGICIEL

III.4.1. Choix du modèle

En utilisant des différents techniques avec les paramètres T° , C, pH, SM, OBD, Ca^{2+} , Mg^{2+} et HCO_3^- qui sont considérés comme des variables d'entrée pour le développement des capteurs logiciels (OD et Cl), cependant la sortie de celui-ci est prise comme la concentration de l'OD et le Cl. Afin de comparer l'efficacité des modèles : MLP, RBF, et DL.

III.4.1.1. Modèle Neuronal-MLP

Dans un premier temps on commence par la conception du modèle neuronal de type MLP Pour être appliqué dans le développement du capteur logiciel. Nos réseaux utilisés organisé en plusieurs couches, ils comprennent Une couche d'entrée, une ou plusieurs couches cachées intermédiaires, et une couche de sortie qui doit restituer l'ensemble des sorties désirées après apprentissage. Les résultats sont présentés dans les tableaux III.1 et III.2.

Capteur Logiciel	Technique			Apprentissage				Test			
		Couches	Itérations	EAM	REQM	R^2	tapp	EAM	REQM	R^2	ttest
OD	MLP	8_4_1	1000	0.44	0.621	0.92	41.55	0.52	0.708	0.90	0.03
		12_8_4_1		0.25	0.39	0.97	111.4	0.33	0.48	0.96	0.05
		12_4_1	3000	0.330	0.435	0.961	176.28	0.35	0.459	0.96	0.031
		12_8_1		0.130	0.221	0.99	258.88	0.18	0.282	0.985	0.031
		12_8_1		0.125	0.204	0.99	273.72	0.18	0.336	0.979	0.031

Tableau. III.1. Résultats d'apprentissage et de test (Modèle neural-MLP) pour le capteur d'OD.

Capteur Logiciel	Technique			Apprentissage				Test			
		Couches	Itérations	EAM	REQM	R^2	tapp	EAM	REQM	R^2	ttest
CL	MLP	8_4_1	1000	2.78	3.97	0.77	12.3	3.25	4.55	0.56	0.05
		12_8_1		2.27	3.22	0.85	108.6	2.9	3.92	0.70	0
		12_4_1	3000	2.35	3.52	0.82	372.5	2.96	4.24	0.63	0.016
		12_8_1		0.93	1.40	0.97	482.9	0.91	1.4	0.97	0.02

Tableau. III.2. Résultats d'apprentissage et de Test (Modèle neural-MLP) pour le capteur logiciel de Cl.

Pour avoir une meilleure configuration on doit changer les paramètres à savoir :

- Les nombres des couches.
- Les nombres des itérations.

- Les fonctions d'activations.

Après plusieurs essais des paramètres sus indiqués on a obtenu un résultat favorable : lorsqu'on augmente les nombres des couches le temps d'exécution augmenté.

III.4.1.2. Modèle Neuronal-RBF

Le réseau de neural-RBF comporte une seule couche cachée dont la fonction d'activation est appelée fonction-noyau ou gaussienne et une couche de sortie avec une fonction d'activation linéaire. Différentes architectures sont testées pour déterminer le nombre adéquat de neurones dans la couche cachée du modèle Neuronal-RBF. Les tableaux III.3 et III.4 montrent pour plusieurs architectures de réseaux testés, les résultats correspondants aux différents paramètres d'entraînement.

Capteur Logiciel	Technique		Apprentissage				Test			
			nombres des neurones	EAM	REQM	R^2	tapp	EAM	REQM	R^2
OD	RBF	200	0.36	0.68	0.90	27.47	0.63	0.93	0.82	0.03
		300	0.1	0.28	0.98	38.45	0.19	0.49	0.95	0.03
		400	4.11E-15	1.83E-14	1	59.70	0.017	0.28	0.98	0.078
		600	4.65E-15	2.07E-14	1	91.41	0.017	0.28	0.98	0.1

Tableau. III.3. Résultats d'apprentissage et de Test (Modèle neural-RBF) pour le capteur logiciel d'OD.

Capteur Logiciel	Technique		Apprentissage				Test			
			nombres des neurones	EAM	REQM	R^2	tapp	EAM	REQM	R^2
CI	RBF	300	0.6	1.46	0.97	38.05	0.58	1.47	0.97	0.05
		600	2.75E-14	5.5E-14	1	132.3	0.02	0.35	0.998	0.0625
		900	2.75E-14	5.5E-14	1	135.5	0.02	0.35	0.998	0.0625
		6000	2.75E-14	5.5E-14	1	137	0.02	0.35	0.998	0.0625

Tableau. III.4. Résultats d'apprentissage et de Test (Modèle neural-RBF) pour le capteur logiciel de CI.

En remarque qui en changeant le nombre des neurones de la couche cachée qu'on augmente les nombres de neurones les performances deviennent meilleurs et dès qu'on atteint un certain seuil malgré l'augmentation des nombres de neurones les performances restent les mêmes. La structure du réseau composé de 600 neurones de la couche cachée est la mieux placée.

III.4.1.3. Modèle Deep Learning

Les résultats d'entraînement et de test obtenus pour le modèle DL sont présentés dans les tableaux III.5 et III.6. Les paramètres, tels que : R^2 , EAM et REQM pour les deux phases sont indiqués pour différentes architectures d'auto-encodeurs.

Capteur Logiciel	Technique					Apprentissage				Test			
		EA1	EA2	MLP	fcts	EAM	REQM	R^2	tapp	EAM	REQM	R^2	ttest
OD	DL	4	22	8_4_1	tan_tan_pu	0.02	0.02	0.98	167	0.017	0.025	0.97	0.1
		4	22	12_12_1	tan_tan_pu	0.01	0.01	0.99	417	0.014	0.024	0.98	0.08
		4	22	22_22_1	rad_tan_pu	0.01	0.01	0.99	2029	0.007	0.015	0.99	0.13

Tableau. III.5. Résultats d'apprentissage et de Test (Modèle Deep Learning) pour le capteur logiciel d'OD.

Capteur Logiciel	Technique					Apprentissage				Test			
		EA1	EA2	EAM	REQM	R^2	tapp	EAM	REQM	R^2	ttest		
CI	DL	4	4	0.61	0.87	0.99	806.9	0.80	1.07	0.98	0.09		
		4	19	0.53	0.78	0.99	2498.5	0.63	0.92	0.99	0.06		
		4	15	0.39	0.60	0.995	2672.2	0.48	0.79	0.989	0.03		
		4	18	0.05	0.12	0.9998	2905.3	0.066	0.16	0.9995	0.05		

Tableau. III.6. Résultats d'apprentissage et de Test (Modèle Deep Learning) pour le capteur logiciel de CI.

Dans cette partie, nous avons présenté les résultats obtenus avec différentes architectures de DL. Après plusieurs exécutions, en optimisant les différents paramètres des capteurs (nombre d'auto encodeur, nombre de neurones dans les réseaux de neurones artificiels et les fonctions d'activations). Nous avons donc conclu que l'architectures apprises étaient seules responsables des résultats obtenus et par la suite comparer ces caractéristiques avec d'autres techniques.

III.4.2. Évaluation des performances

A travers les résultats de simulation obtenus dans les tests de validation des trois techniques neuronales MLP, neuronal-RBF et DL, on déduit les résultats suivants :

Techniques	Apprentissage				Test			
	EAM	REQM	R^2	tapp	EAM	REQM	R^2	ttest
MLP	0.130	0.221	0.99	258.88	0.18	0.282	0.985	0.031
RBF	4.65E-15	2.07E-14	1	91.41	0.017	0.28	0.98	0.1
DL	0.01	0.01	0.99	2028.7	0.007	0.015	0.9908	0.13

Tableau. III.7. Évaluation des performances d'OD pour des différentes techniques.

Techniques	Apprentissage				Test			
	EAM	REQM	R^2	tapp	EAM	REQM	R^2	ttest
MLP	0.93	1.40	0.97	482.9	0.91	1.4	0.97	0.02
RBF	2.75E-14	5.5E-14	1	137	0.02	0.35	0.998	0.0625
DL	0.05	0.12	0.9998	2905.3	0.066	0.16	0.999	0.05

Tableau. III.8. Évaluation des performances de Cl pour des différentes techniques.

À partir des résultats obtenus des tableaux III.7 et III.8, on peut noter ce qui suit.

- La valeur calculée R^2 indique un degré de corrélation élevé entre les données réelles et les données prédites en utilisant les techniques MLP, RBF et DL. Le coefficient R^2 est compris entre 0,97 et 0,99 pour le capteur logiciel d'OD et le capteur logiciel de Cl.
- Généralement, la valeur de la REQM obtenue avec DL à la valeur la plus faible par rapport à la valeur obtenue avec les autres techniques dans la phase de test.
- Le temps d'apprentissage obtenu par DL est plus grand par rapport aux autres techniques.

III.5. APPLICATION

Afin de montrer l'importance et l'intérêt du capteur logiciel. Une application de surveillance de la qualité de l'eau basée sur un classifieur DL est présentée dans la figure III.1. Dans cet exemple, une simulation à base de ce classifieur est effectuée. Trois scénarios sont proposés à cet effet : le premier utilise dix capteurs, le second neuf avec un capteur défaillant, et le troisième, neuf capteurs avec un capteur logiciel OD à base du modèle DL.

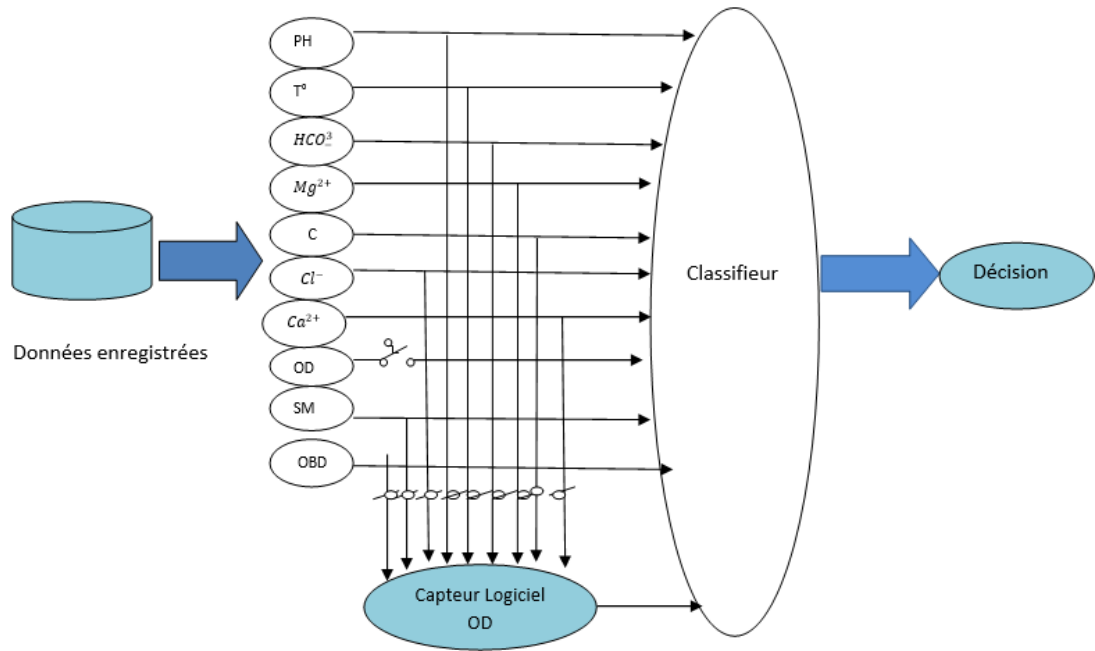


Fig. III.1. Classifieur et capteur logiciel OD à base du modelé DL.

Cas	Test	
	Taux (%)	Temps(s)
Cas 1	98.9051	0.0566
Cas 2	43	0.047
Cas 3	93.07	0.0556

Tableau. III.9. Résultats obtenus pour les 3 scénarios/cas.

Les résultats obtenus, montrant le taux de classification quant à la décision de potabilité pour ces trois cas, sont montrés dans le tableau III.9. On peut remarquer à partir de ce tableau, une détérioration importante du taux de classification dans le cas du deuxième scénario (cas de capteur en panne), alors que dans le troisième scénario, un taux de classification élevé est obtenu, soit 93,07% (cas de remplacement du capteur défaillant par le capteur logiciel OD).

CONCLUSION

Ce dernier chapitre a fait l'objet d'une étude en simulation concernant la mise en œuvre de trois techniques d'apprentissage statistique neuronal-MLP, neuronal-RBF et Deep Learning appliquées dans le domaine du contrôle et de surveillance des eaux potables. Cette étude a permis la validation et l'évaluation des performances de chacune de ces méthodes présentées. Une étude comparative dans le but d'un choix décisif de la méthode la mieux adaptée à l'application a été effectuée pour la régression et la classification. La technique Deep Learning retenue pour ses qualités et avantages adaptés au problème posé. Cette technique a fourni de très bons résultats de simulation au développement du capteur logiciel de OD qui présente une solution satisfaisante pour notre problème de régression. Le contrôle et surveillance de potabilité de l'eau présentée pour un problème de classification multi-classe, la technique a fourni de très bons résultats de simulation. Il a présenté en fait de nombreux avantages, que ce soit en termes de classification et / ou de régression.

CONCLUSION GENERALE

CONCLUSION GENERALE

L'étude présentée découle des progrès technologiques importants qui ont été enregistrés ces dernières années, où l'intérêt d'une surveillance moderne et efficace de la qualité des eaux propres s'est vu imposé. Dans ce domaine, les capteurs logiciels sont devenus un moyen très attrayant pour faire face au manque de capteurs spécifiques dans les chaînes de contrôle et de surveillance. Notre travail peut être considéré comme une contribution aux solutions proposées, pour résoudre des problèmes d'intérêt stratégique à préoccupation nationale et internationale, utilisant des outils modernes à base de techniques avancées.

Le travail de recherche présenté dans ce mémoire a été consacré à la mise en œuvre de trois techniques d'apprentissages statistiques appliquées (MLP, RBF et Deep Learning) à la régression et la classification dans le domaine du contrôle et de surveillance des eaux potables. Dans l'objectif de découvrir laquelle des techniques est la plus efficace, dans ce domaine à partir d'une étude comparative.

Cette étude a été structurée autour de trois chapitres essentiels. Le premier consacré aux capteurs utilisés dans l'industrie, il a permis de présenter des généralités sur les capteurs ainsi leurs différents types et on s'est basé sur les capteurs logiciels qui sont considérés comme alternative des capteurs physiques.

Dans le second chapitre, nous avons parlé des techniques d'apprentissages statistiques et particulièrement de l'apprentissage en profondeur (Deep Learning), Nous avons traité les différents types de modèles (CNN, DBN, L'auto-encodeur). Enfin le troisième et dernier chapitre, a fait l'objet d'une étude en simulation concernant la mise en œuvre les modèles d'apprentissages statistiques appliqués (MLP, RBF et Deep Learning) dans le domaine du contrôle et de surveillance des eaux potables. Cette étude a permis la validation et l'évaluation des performances de chacune des méthodes susmentionnées. Une étude comparative dans le but d'une sélection de la meilleure méthode adaptée à l'application effectuée. La discussion des résultats obtenus, nous a permis d'opter pour la technique Deep Learning qui représente la meilleure solution.

Le capteur logiciel est l'une des solutions en cas de défaillance du capteur physique qui donne l'estimation de la concentration de l'oxygène dessous (OD) permettant au classificateur de prendre la décision de la potabilité de l'eau.

Enfin, Comme perspectives, nous proposons d'appliquer des techniques de prétraitement des données pour la réduction de dimension des variables d'entrée afin d'améliorer ces performances.

BIBLIOGRAPHIE

BIBLIOGRAPHIE

- [1] A. Hadj Kaddour, A.Djedid, "Évaluation des performances des techniques RNA et ELM utilisées dans le développement des capteurs logiciels pour la surveillance de la qualité de l'eau", Mémoire de master, Université de M'sila, Algérie, 2017.
- [2] M. Huang, Y. Mab, J. Wan, "A sensor-software based on a genetic algorithm-based neural fuzzy system for modeling and simulating a wastewater treatment process", *Applied Soft Computing*, Vol.27, pp. 1-10, 2015.
- [3] B. Lamrini, et al, "A neural software sensor for on-line prediction of coagulant dosage in a drinking water treatment plant", *Transactions of the Institute of Measurements and Control*, Vol. 27, pp. 195-213, 2005.
- [4] A. Bougharouat, "Modification de la sensibilité d'un QCM (Quartz Cristal Microbalance) par le dépôt dessus de films minces élaborés par plasma à partir de vapeurs de TEOS", Université mentouri-constantine, Algérie, 2009.
- [5] A. Benichou, "Conception et modélisation d'un accéléromètre microélectronique triaxial". Magister. Université Abou bakr Belkaïd-Tlemcen, Algérie, 2012.
- [6] H. Imen, "Etudes physico-chimique de capteurs à base de nanomatériaux pour des applications biomédicales", Thèse de doctorat, INSA, Lyon, France, 2009.
- [7] G. Asch, "Les capteurs en instrumentation industrielle", Dunod, Edition 4, p. 816, 1993.
- [8] M. Ladjal, "Contribution au développement de systèmes de surveillance innovants dédiés au contrôle de la qualité des eaux potables", Thèse de doctorat, Université de M'sila, Algérie, 2013.
- [9] Elhiri, Mohammed, Étude de développement d'un capteur intelligent pour la détection des vapeurs de styrène, Diss, Université du Québec à Trois-Rivières, 1991.
- [10] Brissaud, Florent, et al, "Capteurs intelligents : Nouvelles technologies et nouvelles problématiques pour la sûreté de fonctionnement." *Maîtrise des Risques et de Sûreté de Fonctionnement*, Lambda-Mu 16. Institut pour la Maîtrise des Risques, 2008.
- [11] S. Chakkor, "E-diagnostic de processus physiques à base des méthodes de haute résolution Application : machines éoliennes", Thèse de doctorat, Université Abdelmalek Essaâdi, Maroc, 2015.

- [12] M. Zayed, "Véhicules Intelligents : Etude et développement d'un capteur intelligent de vision pour l'attelage virtuel", Thèse de Doctorat, Université des Sciences et Technologie de Lille, France, 2005.
- [13] S. Djerarda, K.Hedli, "Traitement automatique des eaux basées sur les techniques d'apprentissages statistiques", Mémoire de master, Université de M'sila, Algérie, 2018.
- [14] M. Djerioui, "Contribution au Développement de Systèmes Multicapteurs Intelligents Dédiés à la Surveillance et au Contrôle de la Qualité des Eaux Propres", Thèse de doctorat, Université de M'sila, Algérie, 2019.
- [15] Y. Moualek Djaloul, "Deep Learning pour la classification des images", Mémoire de master, Université de Tlemcen, Algérie, 2017.
- [16] A. Manuel, L. Pirmin, "Le deep learning pas à pas", PARTIE I : Concepts, Des labos de R&D à la vie quotidienne. SQLI Digital experience, 2019.
- [17] L. Stanislas, "Exploration des réseaux de neurones à base d'autoencodeur dans le cadre de la Modélisation des données textuelles", thèse de doctorat, université de Sherbrooke, 2016.
- [18] H. Tolgui, " Deep learning par reconnaissances du visage ", mémoire de master, université du Biskra, Algérie, 2018.
- [19] Tsai, M. J., Tao, Y. H., et Yuadi, I., "Deep learning for printed document source identification", Signal Processing : Image Communication, vol. 70, pp. 184-198, 2019.
- [20] B. Valentin, "Approches Deep Learning Appliquées aux signaux audio : parole et musique", Rapport de Projet de Fin d'Étude, Institut de Recherche en Informatique de Toulouse (IRIT), 2015.
- [21] M. Matthieu le goff, "Techniques d'analyse de contenu appliquées à l'imagerie spatiale", Thèse de Doctorat, Université de Toulouse ,2017.
- [22] Hinton, G. E., Osindero, S., & Teh, Y. W. A fast learning algorithm for deep belief nets. Neural computation, 18(7), 1527-1554,2006.
- [23] G. Christian, "Apprentissage profond ", Cours en ligne, Université de LAVAL, 29 novembre 2017.
- [24] A. Labiad, " Sélection des mots clés basée sur la classification et l'extraction des règles d'association ", Mémoire de master, Université du Québec à Trois-Rivières, 2017.

ملخص :

في الصناعة، تعتبر المستشعرات الفيزيائية أدوات قياس لجمع المعلومات اللازمة. ومع ذلك، غالبًا ما تكون هذه المستشعرات غير متوفرة أو تكون تكلفتها عالية جدًا. يمكن للنظام القائم على أجهزة استشعار البرمجيات توفير وسيلة فعالة واقتصادية لحل هذه المشاكل.

لهذا الغرض، يركز عملنا على تصميم وتطوير مستشعرات البرامج استنادًا إلى تقنيات التعلم الآلي. يمكن دمج هذه المستشعرات في أجهزة استشعار ذكية واستخدامها في التطبيقات الصناعية. من أجل تقييم الأساليب المختارة، أجريت دراسة مقارنة لتحديد مزايا كل منها.

الكلمات المفتاحية: مستشعر ذكي، ذكاء اصطناعي، مستشعرات برامج، انحدار، تصنيف، تعلم عميق.

Résumé :

Dans l'industrie, les capteurs physiques représentent des outils de mesures pour collecter les informations nécessaires. Cependant, ces capteurs sont souvent indisponibles ou leur coût est très élevé. Néanmoins, le système à base des capteurs logiciels peut fournir un moyen efficace et économique pour résoudre ces problèmes.

Pour cet objectif, notre travail s'articule sur la conception et le développement des capteurs logiciels à base des techniques de l'apprentissage des machines. Ces capteurs peuvent être intégrés au niveau de capteurs intelligents et utilisés dans des applications industrielles. Afin d'évaluer les méthodes optées, une étude comparative a été réalisée pour dégager les avantages de chacune.

Mots-clés : Capteur intelligent, Intelligence artificielle, Capteurs logiciels, Régression, Classification, Apprentissage en profondeur.

Abstract :

In the industry, physical sensors are measurement tools for collecting the necessary information. However, these sensors are often unavailable or their cost is very high. Nevertheless, the system based on soft-sensors can provide an effective and economical solution to solve these problems.

For this purpose, our work focuses on the design and development of soft-sensors based on machine learning techniques. These sensors can be integrated in intelligent sensors and used in industrial applications. In order to evaluate the proposed methods, a comparative study was conducted to identify the advantages of each.

Keywords : Intelligent sensor, Artificial intelligence, Soft-sensors, Regression, Classification, Deep learning.