REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Mohamed Khider – BISKRA

Faculté des Sciences Exactes, des Sciences de la Nature et de la Vie

Département d'informatique

N° d'ordre : GLSD 27/M2/2020

Mémoire

présenté pour obtenir le diplôme de master académique en

Informatique

Parcours : **génie logiciel, système distribue**

Classification des domaines protéiques par techniques d'apprentissage profond

---

Par :

Ben aissa Basma

Soutenu le 27 juillet 2020, devant le jury composé de :

		Président
Rahmani Salima	MAA	Rapporteur
		Examinateur

Dédicaces

Je dédie ce mémoire à l'esprit de mon père, à ma très chère mère,

à mes sœurs et frères et à tous mes amies.

Remerciements

En tout premier lieu je remercie de mon plus profond cœur ALLAH tout Puissant de m'avoir éclairé vers le bon chemin.

Je ne saurai suffisamment remercier la personne qui m'a aidé à réaliser ce travail l'encadreur Mme Rahmani Salima tout au long de l'année, je la remercie également pour sa disponibilité et sa compréhension.

Je tiens à remercier ma famille pour leur apport affectif et leurs sacrifices pendant toute la période universitaire pour leur soutien inconditionnel aussi je tiens à remercier infiniment mes amis et mes collègues pour leur aide et leur soutien sans conditions.

Résume

La classification de des domaine protéine est un champ de recherche qui sert de pointe de départ pour la prédiction de la fonction des gènes et des protéines et aux analyses expérimentales et les processus pour lesquels sont impliqués les protéines, elles jouent un rôle primordial dans la compréhension du comportement des protéines responsables des maladies, de développer les médicaments les plus efficaces et avoir une médecine préventive contre les fléaux éventuels. la prédiction est basé sur la comparaison de la protéine à fonction inconnue avec d'autres protéines de fonction connue. Le regroupement des protéines aux fonctions similaires et la caractérisation de ces groupes ont motivé l'émergence des classifications.

Depuis quelques années, le monde a connu un grand renouveau avec les techniques de l'apprentissage profond, inspirées des réseaux de neurones du cerveau. Plusieurs sciences utilisent l'apprentissage profond comme la bioinformatique, avec des outils et des objectifs souvent très différents. Avec l'intégration de l'apprentissage profond pour la classification supervisé dans le domaine protéique, ce dernier a assisté à une évolution de la méthode classique à la méthode moderne .en utilisant les méthodes de classification supervisée machine à Long Short Term Memory(en anglais) (Lstm) pour mettre en œuvre l'apprentissage, les taux de classification que nous avons trouvé montre que nos résultats sont compétitifs.

Mot clé : Bioinformatique, Protéine, classification des domaines protéine, Apprentissage profond, classification supervisée.

Table matières

Introduction générale

La bioinformatique est un champ de recherche multi-disciplinaire sur lequel sautèle de concerts avec des : biologistes, informaticiens, mathématiciens et physiciens. La bioinformatique a connu un essor extraordinaire de développement bien sûr lié à l'aboutissement de nombreux projets de séquençage, projets ayant conduit à l'arrivée d'énormes quantités de données dont il faut maintenant tirer le plus d'informations possibles et de développer les modèles, les méthodes et d'outils afin d'analyser les données biologiques (génomes, protéomes, etc.…). Le progrès de la bioinformatique a été rendu possible par les énormes progrès réalisés au niveau des capacités de calcul et de stockage des ordinateurs, sans ces progrès il n'est pas envisageable de construire des banques de données capables de manipuler l'intégralité des séquences biologiques publiées ou de développer des logiciels susceptibles d'effectuer des traitements sur de très large sous-ensembles de ces banques de données et produire de nouvelles connaissances pour mieux comprendre et résoudre des problèmes scientifiques posés par la biologie.

Pour solutionner le problème de la classification des protéines inconnu qui consiste à assigner à la protéine la classe famille correspondante afin de déterminer le rôle biologique et biochimique qu'elle joue dans l'organisme qui s'appelle également la protéomique in silico qui traite l'étude des séquences protéiques par simulation sur ordinateur. L'application de la fouille de données pour la résolution des problèmes biologiques, la fouille des données joue un rôle fondamental dans la compréhension des problèmes bioinformatiques émergeants tels que l'annotation des protéines. La tâche de la fouille de données la plus couramment utilisée est la classification incluant la découverte des règles de classification qui permettent de construire des modèles de prédiction et de classifications performantes pour l'annotation des séquences par les différentes méthodes qui sont développées et exploitées dans le but d'y apporter des solutions fiables, efficaces et peu coûteuses.

Ce présent mémoire englobe cinq chapitres :

- **Premier chapitre**: est consacré aux concepts et les principes de base de la :, bioinformatique et les protéines.
- **Deuxième chapitre :** exposition de la classification des domaines protéiques, la similarité structurale entre deux protéines et Identification de la super-famille

- **Troisième chapitre :** nous présentons un aperçu sur l'apprentissage profond, et les modèle d'apprentissage et leur fonctionnement.
- **Quatrième chapitre :** une vue conceptuelle de notre système est décrite dans ce chapitre, on va détailler les deux architectures (générale et détaillée) avec l'identification de la modélisation appropriée à atteindre avec un modèle d'apprentissage profond.
- **Cinquième chapitre :** ce chapitre présente les outils utilisés dans la phase de l'implémentation. Nous terminons notre mémoire par une conclusion générale.

Chapitre 01

Introduction à la Bioinformatique

1.1 Introduction

1.2 Bioinformatique

1.2.1 Définition Bioinformatique

La bioinformatique est une discipline récente qui propose et développe des modèles, des méthodes et des outils afin de gérer, manipuler et analyser l'information biologique. C'est à ce titre, la bioinformatique est une science interdisciplinaire en développement rapide, qui fait appel à des connaissances pointues en mathématique, en informatique et en biologie. L'objectif général de ce domaine est d'utiliser l'ordinateur et des modèles statistiques afin de cartographier et interpréter les données biologiques et utilisée dans l'analyse de génomes de génomes, de protéomes (séquences de protéines), de la modélisation tridimensionnelle de biomolécules et de systèmes biologiques etc...

D'après Claverie et al. **[1]** « la bioinformatique est la discipline de l'analyse de l'information biologique, en majorité sous la forme de séquences génétiques et de structures de protéines. C'est le décryptage de la bioinformation (Computational Biology" en anglais). La bioinformatique est donc une branche théorique de la Biologie. Son but, comme tout volet théorique d'une discipline, est d'effectuer la synthèse des données disponibles (à l'aide de modèles et de théories), d'énoncer des hypothèses généralisatrices (ex. : comment les protéines se replient ou comment les espèces évoluent), et de formuler des prédictions (ex. : localiser ou prédire la fonction d'un gène) ».

D'après Andrade et Sander **[2]** « Bioinformatics is a science of recent creation that uses biological data, completed by computational methods, to derive new biological knowledge ».

1.2.2 Objective de bioinformatique

Le bioinformatique à un rôle important de collecter et de traiter de données génomique et protéique :

- Organiser des grandes bases de données de biologie génomique de manière efficace accessibles en ligne, afin d'étudier des gènes et des protéines.
- Fournir des outils qui facilitent l'analyse ces données génomiques (la séquence, la structure, la fonction, les interactions, etc.…), des outils de comparaison de séquences protéiques et nucléotidiques, des outils de traductions dans le but de la prédiction physiologique, la prédiction expérimentale, la classification des domaines protéine et de découvrir des protéines qui sont de la même séquence, dans le cadre de connaitre leur familles et leur fonctionnement.

- Fournir la possibilité de comparaison de séquences protéiques et nucléotidiques en le comparant aux séquences d'une base de données pour interpréter les résultats de manière précise et significative et de découvrir des nouveaux séquence.
- Faciliter aux chercheurs des compagnies pharmaceutiques à élaborer des études détaillées des fonctions des protéines afin de faciliter la conception de médicaments.
- La bioinformatique peut être utilisée dans tout système où l'information peut être représentée numériquement, elle peut être appliquée à tout le spectre des organismes vivants, des cellules individuelles aux écosystèmes complexes.

1.3 La génomique

La génomique est la science des génomes qui étudie les séquences d'ADN des êtres vivants. Un génome est formé de l'ensemble des informations génétiques contenues dans la cellule Par exemple, dans une cellule humaine, le génome se compose des informations portées par les 23 paires de chromosomes du noyau ainsi que l'ADN présent dans les mitochondries.

La génomique analyse les génomes, leur structure, leur organisation et étudie leur fonctionnement, elle utilise la bioinformatique pour stocker et analyser les informations **[3].**

1.3.1 **Les acides nucléiques**

Les acides nucléiques sont des macromolécules, c'est-à-dire de grosses molécules relativement complexes. Les acides nucléiques sont constitués d'un enchaînement de nucléotides reliés par des liaisons phosphodiesters. Les nucléotides se composent toujours de trois éléments fondamentaux: un sucre, un groupe phosphate (acide phosphorique), une base azotée. On trouve des acides nucléiques (ADN et ARN) dans les cellules de presque chaque organisme. Il existe deux types d'acides nucléiques :

1.3.1.1 **Acide désoxyribonucléique (ADN)**

L'ADN ou l'acide désoxyribonucléique est une macromolécule biologique qui se trouve au cœur de chaque cellule vivante, qui contient le code génétique qui renferme toute l'information héréditaire d'un organisme.

Il est constitué de deux chaînes de nucléotides (l'unité de base de l'ADN) mono phosphates liés chacun par une liaison ester entre son carbone 3' (alcool secondaire) et le carbone 5' (alcool primaire) du nucléotide suivant. Ces deux chaînes de nucléotides sont unies entre elles par des liaisons hydrogènes pour former un hybride en forme de double hélice (c'est le modèle de Watson et Crick en 1953).

Chaque sortes nucléotides est formée de trois unités : une base azotée, un sucre et un groupe phosphate. Il existe quatre sortes de nucléotides formant l'ADN: l'adénine (A), la guanine (G), la thymine (T) et la cytosine (C). Les deux brins d'ADN sont reliés entre eux par les nucléotides qui forment des paires complémentaires et une structure hélicoïdale : l'adénine avec la thymine (A-T ou T-A) et la guanine avec la cytosine (G-C ou C-G) **[4].**

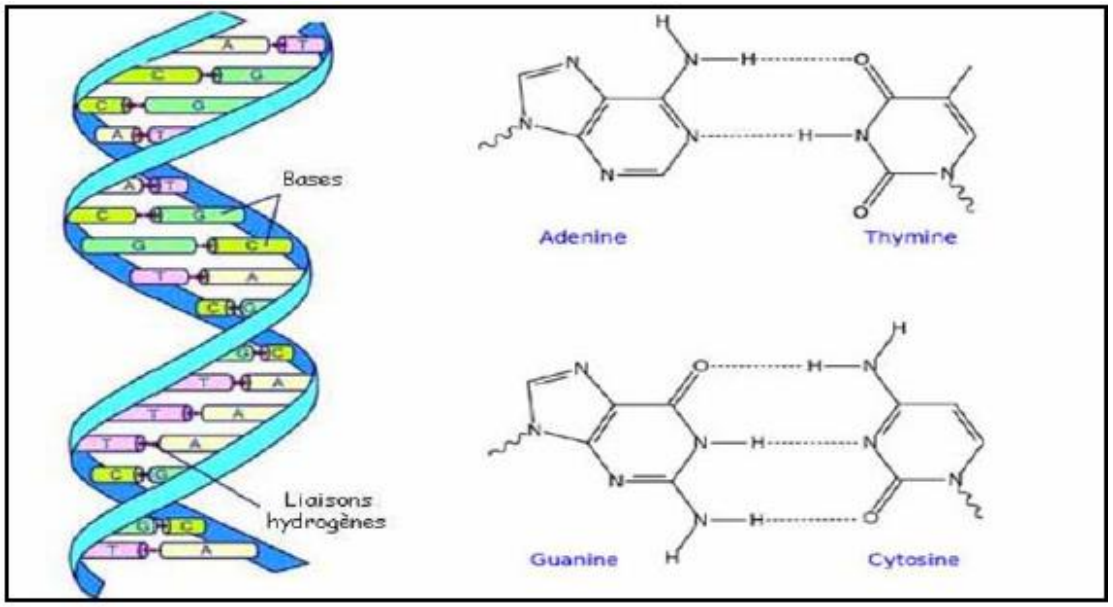

Figure1.1 : structure d'ADN.

1.3.1.2 **Acide Ribonucléique (ARN)**

L'ARN ou l'acide ribonucléique est une macromolécule similaire à l'ADN, constituée d'un enchaînement de nucléotides sur un seul brin. Mais il y a des différences telles que: Contrairement aux brins de l'ADN qui vont en couple, l'ARN est généralement simple brin. La Thymine (T) de l'ADN est remplacé par l'uracile (U) dans l'ARN **[6]**.

L'ARN joue plusieurs rôles: Il est exporté du noyau pour fournir l'information génétique de l'ADN et permettre la synthèse des protéines.

L'ARN dans la cellule peut être codant ou non codant :

- ARN codant: comme l'ARN messager qui traduits par la suite en protéine,
- ARN non codant: comme, les ARN ribosomiques et les ARN de transfert. Contrairement aux ARN messagers, ces ARN sont des molécules fonctionnelles non traduites en protéine **[5].**

1.3.2 Gènes

Un gène constitué d'une séquence particulière de nucléotides, il est l'unité physique et fonctionnelle de base de l'hérédité d'ADN, dans lequel on retrouve une information

génétique qui permet la fabrication d'une molécule particulière ou qui détermine un caractère bien précis. Certains gènes agissent comme des instructions pour fabriquer des molécules protéines. Cependant de nombreux gènes ne codent pas pour les protéines **[4].**

1.3.3 **Génome**

Les quatre types de nucléotides de l'ADN étant les mêmes pour toutes les espèces vivantes (adénine, cytosine, guanine et thymine), la diversité génétique des organismes repose sur la séquence des nucléotides dans leurs gènes. Chaque espèce vivante possède donc un ensemble de gènes qui lui est unique qui porte le nom de génome **[4].**

1.4 protéomique

Dans la pratique, la protéomique s'attache à identifier de manière globale les protéines extraites d'une culture cellulaire, d'un tissu ou d'un fluide biologique, leur localisation dans les compartiments cellulaires, leurs éventuelles modifications post-traductionnelles ainsi que leur quantité.

Elle permet de quantifier les variations de leur taux d'expression en fonction du temps, de leur environnement, de leur état de développement, de leur état physiologique et pathologique, de l'espèce d'origine. Elle étudie aussi les interactions que les protéines ont avec d'autres protéines, avec l'ADN ou l'ARN, ou d'autres substances.

La protéomique fonctionnelle étudie les fonctions de chaque protéine, elle étudie enfin la structure primaire, secondaire et tertiaire des protéines **[6].**

1.4.1 protéines

Les protéines représentent l'une des classes de molécules les plus importantes dans les organismes vivants. Leurs fonctions comprennent la catalyse des processus métaboliques sous forme d'enzymes; ils jouent un rôle important dans la transmission du signal, les mécanismes de défense et le transport des molécules; et ils sont utilisés comme matériau de construction d'organismes par exemple dans les cheveux (la protéine de la kératine).

Les protéines sont formées de chaînes d'acides aminés. Chaque acide aminé a une structure constante. Deux acides aminés peuvent se joindre, avec un " lien de peptide ", formant une chaîne: un " polypeptide ".

Les protéines sont généralement représentées sous forme de séquences, elles se replient en structures tridimensionnelles plus ou moins stables (voir Figure 1.2), les

Protéines ont des tailles de plusieurs centaines d'acides aminés plus spécifiquement les petites chaînes sont appelées peptides les protéines étant des polypeptides pouvant être réunies par des ponts disulfures.

Les protéines se répartissent en quatre classes (figure 1.2) générales sur la base de leur structure :

La structure primaire : décrit l'ordre unique dans lequel les acides aminés sont liés entre eux pour former une protéine .les sont enchaînés les uns à la suite des autres par des liaisons peptidiques (reliant l'extrémité C-terminale d'un acide aminé à l'extrémité N-terminale d'un autre). La séquence d'une protéine se lit de l'extrémité N-terminale vers l'extrémité C-terminale.

La structure secondaire : elle fait référence à l'enroulement ou au repliement d'une chaîne polypeptidique qui donne à la protéine sa forme 3D. Il existe deux types de structures secondaires observées dans les protéines :

Un type est la structure en hélice alpha (α). Cette structure ressemble à un ressort hélicoïdal et est fixée par une liaison hydrogène dans la chaîne polypeptidique. Le deuxième type de structure secondaire dans les protéines est la feuille plissée bêta (β). Cette structure semble être pliée ou p lissée et est maintenue ensemble par une liaison hydrogène entre des unités polypeptidiques de la chaîne repliée qui sont adjacentes les unes aux autres.

Structure tertiaire : la structure tridimensionnelle d'un polypeptide est principalement due aux interactions entre les groupes R des acides aminés qui composent la protéine. Les interactions du groupe R qui contribuent à la structure tertiaire comprennent les liaisons hydrogène, les liaisons ioniques, les interactions dipôle-dipôle et les forces de dispersion de London - essentiellement, toute la gamme de liaisons non covalentes. eaux de structure (ceux dont nous venons de parler).

Structure quaternaire : certaines protéines sont constituées de plusieurs chaînes polypeptidiques, également appelées sous-unités. Lorsque ces sous-unités se réunissent, elles donnent à la protéine sa structure quaternaire.

Nous avons déjà rencontré un exemple de protéine à structure quaternaire: l'hémoglobine. il transporte l'oxygène dans le sang et est composé de quatre sous-unités, deux de chacun des types α et β. Un autre exemple est l'ADN polymérase,

En général, les mêmes types d'interactions qui contribuent à la structure tertiaire, principalement des interactions faibles, telles que les liaisons hydrogène maintiennent également les sous-unités ensemble pour donner une structure quaternaire **[7].**

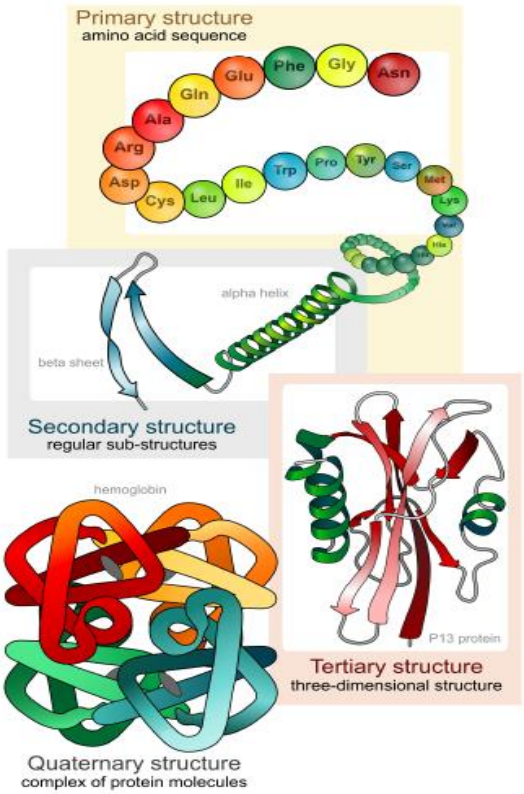

Figure 1.2 : les quatre type de structure protéine.

Le point auquel la structure des protéines a un choc sur leur fonctionnement est montré par l'effet des changements de la structure d'une protéine. N'importe quelle modification à une protéine à n'importe quel niveau structurel, y compris de légers changements du pliage et de la forme de la protéine, peut la rendre non fonctionnelle.

1.4.2 Code génétique

Le code génétique est l'ensemble des règles permettant de traduire les informations contenues dans le génome des cellules vivantes afin de synthétiser les protéines. Au sens large, il établit la correspondance entre le génotype et le phénotype d'un organisme. Ou L'assemblage des acides aminés pour former la chaîne polypeptidique se fait au sein du ribosome [P3]. L'ARN messager issu de la transcription du gène codant pour la chaîne est traduit en chaîne polypeptidique selon le code génétique *Figure 1.3.*

	U		C		A		G		
U	UUU	phénylalanine	UCU	sérine	UAU	tyrosine	UGU	cystéine	U
	UUC		UCC		UAC		UGC		C
	UUA	leucine	UCA		UAA	stop	UGA	stop	A
	UUG		UCG		UAG		UGG	tryptophane	G
C	CUU	leucine	CCU	proline	CAU	histidine	CGU	arginine	U
	CUC		CCC		CAC		CGC		C
	CUA		CCA		CAA	glutamine	CGA		A
	CUG		CCG		CAG		CGG		G
A	AUU	isoleucine	ACU	thréonine	AAU	asparagine	AGU	sérine	U
	AUC		ACC		AAC		AGC		C
	AUA		ACA		AAA	lysine	AGA	arginine	A
	AUG	méthionine	ACG		AAG		AGG		G
G	GUU	valine	GCU	alanine	GAU	acide aspartique	GGU	glycine	U
	GUC		GCC		GAC		GGC		C
	GUA		GCA		GAA	acide glutamique	GGA		A
	GUG		GCG		GAG		GGG		G

Figure 1.3 : Code génétique illustrant la correspondance codon/acide aminé.

1.4.3 Acides amines

Un acide aminé est une petite molécule élémentaire des protéines, se composent d'un atome de carbone central noté Cα, connecté à un groupe aminé NH2, à un groupe carboxyl COOH et une chaîne latérale R qui est spécifique à un acide aminé particulier appelé chaine latérale [7]. La formule générale d'acide amine est dans *la figure 1.4*:

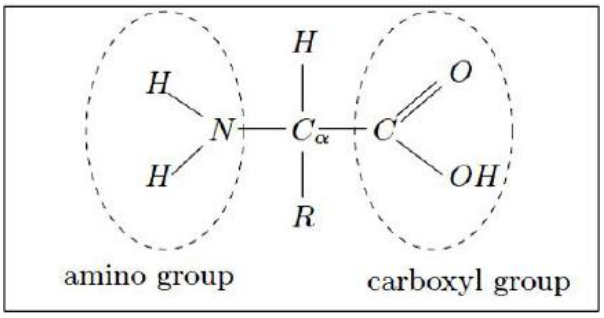

Figure 1.4 : donne la structure générale d'un acide aminé.

Dans la nature, il existe plus d'une centaine d'acides aminés, cependant, seuls 20 d'entre eux peuvent être intégrés dans les protéines synthétisées. Ces dernières se distinguent par leur dimension, leur forme, leur charge, leur capacité de contracter des liaisons hydrogènes et leur réactivité chimique. Une liste complète de structure de 20 acides amines est donnée dans *la figure 1.5*. Chacun d'eux est accompagne de son nom usuel et ses deux codes : l'un a 3 lettres et l'autre a une seule lettre **[8].**

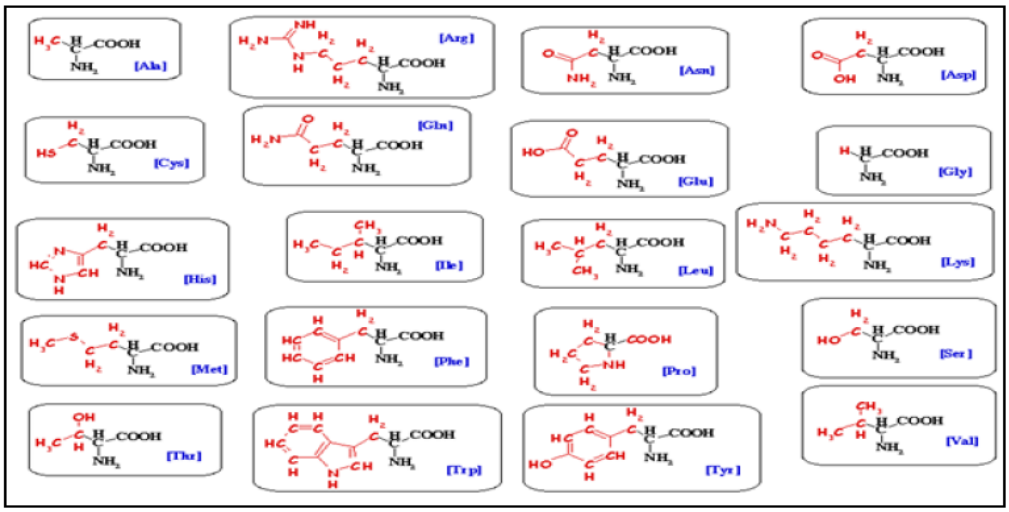

Figure 1.5 : Structure de 20 acides amines.

Les acides aminés de la chaîne protéique sont liés les uns aux autres selon une réaction typique où la racine «hydroxyle» est retirée de la partie (COOH) d'un acide aminé dans ce processus chimique, tandis qu'un hydrogène est éliminé de la molécule (NH2) de l'autre acide aminé. Ces deux racines se combinent pour former de l'eau.Et les deux sites restants sur les deux acides aminés se combinent pour générer une seule molécule, et ce processus est appelé «liaison peptidique» **[6].**

1.4.4 Synthèse des protéines

Comprendre le fonctionnement d'une cellule vivante suppose celle des mécanismes moléculaires complexes qui sous-entendent les diverses activités cellulaires, chaque type cellulaire d'un organisme exprimera un ensemble de protéines ou protéomes qui variera en fonction de l'environnement des cellules et qui sont synthétisées selon les deux étapes suivant :

- La transcription : est la première étape de la fabrication des protéines. Elle se déroule dans le noyau. Elle permet le passage de l'ADN à l'ARN: **(A->U, C->G, G->C, T->A)**
- La traduction : est la 2ème étape de la fabrication des protéines, elle a lieu dans le cytoplasme**.** Elle correspond au décodage de l'information porte par l'ARNm en polypeptides relis en protéines.

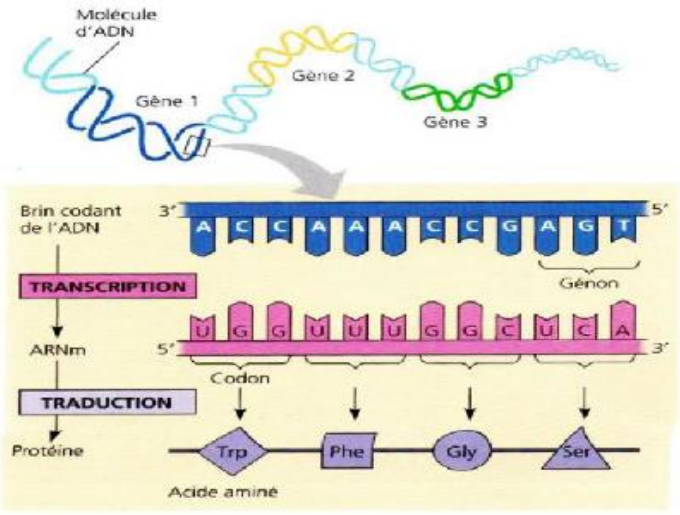

Figure 1.6 : Synthèse des protéines.

Codon: sur la molécule d'ARNm, un groupe de trois nucléotides successifs constitue un codon. Les codons s'enchaînent à la suite du codon d'initiation de la traduction AUG. A chaque codon correspond un acide aminé. La traduction s'arrête à un codon STOP spécifié par UAA, UGA ou UAG. La séquence comprise entre le codon d'initiation de la traduction et le codon stop s'appelle un cadre ouvert de lecture ou ORF (pour Open Reading Frame) **[12].**

1.5 Domaine protéique

Les protéines interagissent généralement ensemble par l'intermédiaire des domaines. Le terme *domaine* protéique associant une séquence compacte et stable protéique à une fonction, en général caractérisé par sa structure et par un certain nombre d'acides aminés [28]. Les domaines sont définis comme des zones compactes qui interagissent peu les unes avec les autres (la chaîne A de la protéine 1C9B, figure 1.7)

L'analyse des séquences et des structures de protéines révèle que beaucoup s'organisent en modules structuraux distincts. En effet, les protéines peuvent être vues comme composées d'une ou plusieurs unités fondamentales appelées domaines protéiques. Il existe plusieurs définitions du domaine protéique selon l'angle sous lequel on se place.

Du point de vue structuraliste, un domaine correspond à une sous séquence d'acides aminés capable de se replier indépendamment du reste de la protéine.

Pour le biochimiste, le domaine est utilisé pour décrire des régions protéiques pour les quelles une fonction propre a pu être caractérisée.

« Un domaine structural est une région compacte de la chaîne polypeptidique capable de se replier de manière stable et en conservant certaines ou toutes ses capacités lorsqu'extrait de la protéine entière. Une chaîne polypeptidique peut » ***[13].***

Un domaine protéique peut constituer à lui seul une protéine. On parle alors de protéine mono-domaine. Il peut aussi s'associer avec d'autres domaines au sein d'une protéine dite multidomaine. Ce domaine conservera sa fonction d'origine ou participera à une fonction différente en collaborant avec les autres domaines [21]

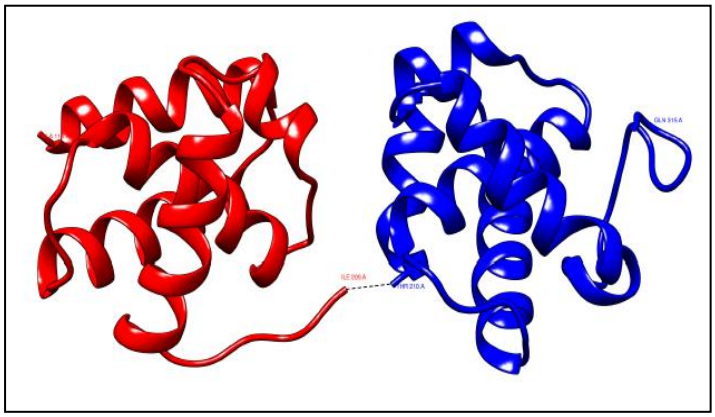

Figure 1.7 : Les deux domaines de la chaîne A de la protéine 1C9B tels que découpé par les protocoles de CATH [15], La jonction entre les deux domaines se fait entre le résidu 209 (isoleucine, au centre en rouge) et le résidu 210 (thréonine au centre en bleu) et est symbolisée par les pointillés noirs [12].

1.5.1 **Motif protéique**

Un motif est une séquence courte associée à des interactions bien précises : site actif. les motifs ont été définis comme des groupes d'acides aminés extrêmement bien conservés entre des séquences globalement différentes, les acides aminés caractérisées par des liaisons hydrogène entre certains de ces acides aminés et des valeurs spécifiques des angles de torsion de la liaison peptidique (angles Φ, Ψ et ω**)[13].** A' un même site, les acides aminés essentiels peuvent être assez proches dans la structure 3D .En définit le secteur protéique comme un groupe d'acides aminés corrélés quasi indépendants. Trois secteurs sont identifiés dans la famille d'enzyme S1A, dont les acides aminés se trouvent être connectés dans la structure 3D, ces secteurs exhibent des fonctions distinctes et une évolution indépendante au sein de cette famille, ce qui ouvre une réflexion intéressante sur l'émergence de la fonction au sein de certains domaines protéiques **[11] [10].**

1.5.2 Site actif d'un domaine protéique

Il existe un type de protéine catalysent des réactions qui sont les enzymes. La portion spécifique de l'enzyme qui effectue cette action se nomme le site actif ou site catalytique, ce dernier peut n'être composé que de quelques résidus et peut avoir de grosses répercussions sur la fonction de la protéine.

1.5.3 Interaction Domaines

L'identification des paires de domaines susceptibles d'interagir à partir de données d'interactions protéine-protéine à grande échelle (et des compositions en domaines de ces protéines) peut servir divers objectifs :

- prédire de nouvelles interactions protéine-protéine, en se basant sur la simple connaissance du contenu en domaines des protéines.
- nettoyer les données d'interactome issues d'expérimentations à grande échelle.
- identifier quels sont les domaines qui interagissent dans une interaction protéine-protéine donnée.

Différentes approches pour établir des paires de domaines qui interagissent et en évaluer la consistance. Certaines approches s'appuient sur la signature des séquences, sur des scores, des paires de profils des domaines interagissant , l'estimation du maximum de vraisemblance de leur modèle, des chaines de Markov Monte Carlo, l'analyse d'exclusion de paires de domaines (DPEA)...

Il existe Des bases de données répertorient les interactions domaines connues iPfam et 3DID. Ainsi que les interactions potentielles a partir la base DO- MINE.

1.5.4 Sites de liaisons

Les sites de liaisons sont des régions spécifiques en surface de la protéine pouvant former des liaisons chimiques avec d'autres protéines, ADN, ARN ou encore petites molécules (ligands), et donc permettre à la protéine d'interagir avec ces molécules **[25].** Les sites de liaisons de deux protéines interagissantes sont appelés interfaces protéine-protéine (encore interface protéine-ligand dans le cas d'interactions protéine-molécule). La compréhension du fonctionnement de ces interfaces, de ces interactions entre protéines est un domaine de recherche ouvert **[21].** L'augmentation du nombre de données structurales a permis l'expansion du nombre de méthodes de prédictions.

Il existe plusieurs approches de prédiction de sites de liaisons, l'une d'elles analyse des ensembles de sites connus (au niveau de complexes protéiques) pour les caractériser puis recherche ces sites à la surface d'une nouvelle protéine **[62].** Certaines méthodes de

prédiction sont basées sur l'observation que les résidus présents dans les sites de liaison sont plus conservés que le reste des résidus même si cette information de séquence n'est pas suffisante pour une prédiction complète d'une interface entre deux protéines **[21].**

1.6 Conclusion

La Bioinformatique est un domaine de recherche en plein essor beaucoup d'avancées sont réalisées du fait qu'elle combine diverses disciplines comme l'informatique, la biologie, les mathématiques, les statistiques, etc…

Dans ce chapitre nous avons présenté la discipline protéomique qui est le principe de notre étude qui consiste à prédire la fonction de protéines à partir de leurs structures ainsi qu'à partir des interactions protéine-protéine, il est donc important de résoudre ces problèmes pour mieux pouvoir identifier les protéines dans le suivant en parle sur la classification protéine

Chapitre 02

Classification des Domaine Protéique Protéine

Introduction

Le nombre de protéines est augmentation en raison de l'émergence de nouveaux gènes, ce qui a conduit à classifie dans des bases de données universel pour facilite leur utilisation dans la prédiction fonctionnel des domaines protéique.

Ce chapitre débute par la présentation de deux bases de données hiérarchisées de domaines structuraux. CATH et SCOP sont deux classifications hiérarchiques des domaines structuraux largement utilisées, elles concordent sur la classification d'une majorité de domaines ce qui tend à conforter leurs classifications respectives, Identification de la super-famille structurale et Le protocole de comparer de domaine structurel.

2.1. Classification structurale de protéines

les classifications structurales de protéines sont des classifications de domaines structuraux (une protéine pouvant être composée de plusieurs domaines et donc apparaître plusieurs fois dans la classification), qui consistent à regrouper ces domaines selon des critères purement structuraux puis des critères d'homologie afin de faire émerger des similitudes et ainsi aider à la compréhension de l'univers des protéines. classifications de domaines structuraux (une protéine pouvant être composée de plusieurs domaines et donc apparaître plusieurs fois dans la classification), qui consistent à regrouper ces domaines selon des critères purement structuraux puis des critères d'homologie afin de faire émerger des similitudes et ainsi aider à la compréhension de l'univers des protéines.

La classification des structures permet d'organiser les domaines structuraux et d'extraire des différents groupes des caractéristiques propres et d'évaluer entre autres la plasticité du groupe dans un de proposer une vue de l'évolution à travers les différentes hiérarchies menant aux familles protéiques et représenter les relations évolutives entre les protéines et aussi de comprendre le rôle fonctionnel des protéines.

Il existe de nombreuses ressources de domaine protéique qui fournissent des classifications de domaines protéiques en familles Celles- ci se basent sur des critères de similarité entre séquences primaires, secondaires ou tertiaires. On peut les opposer par l'expertise automatique ou semi-manuelle choisie pour la création des familles et des alignements, ou encore par les modèles. (par exemple Pfam) ou la structure [par ex. CATH (Sillitoe et al., 2015) et SCOP (Murzin et al., 1995) **[20].**

2.1.1. SCOP, Structural Classification Of Protéines

La classification SCOP est manuelle, basée sur l'inspection visuelle des domaines structuraux (avec l'aide d'outils de comparaison) ainsi que sur les informations relatives aux domaines issus de la littérature **[20].**

SCOP contient quatre principaux niveaux de classification :

– Class, les domaines sont regroupés selon leur composition en structures secondaires.

– Fold (repliement), deux domaines ont le même repliement s'ils appartiennent à la même classe et si l'arrangement spatial et la connectivité de leurs structures secondaires est la même.

– Superfamily, une superfamille contient des membres structurellement proches et partageant un ancêtre commun.

– Family, les membres d'une même famille ont soit des structures et des fonctions très proches, soit une similarité de séquence supérieure à 30%.

La version 1.75 de SCOP (2009) contient 38221 protéines et 110800 domaines structuraux.

Le prototype d'une nouvelle base de données, SCOP2 **[7],** est disponible depuis 2013. La structure hiérarchique n'est plus linéaire, un groupe pouvant appartenir à plusieurs groupes parents. Cela est dû à l'ajout d'informations comme les évènements évolutifs ou le type de protéines (qui n'apparaissent pas dans SCOP).

2.2.2. CATH

Développée par le groupe Orengo 1, elle est composée de quatre niveaux principaux dans lesquels elle classe les domaines protéiques **[15]** :

– Class, les domaines sont séparés en quatre classes suivant leur composition en structures secondaires (hélices _ et feuillet _).

– Architecture, à ce niveau, les groupes se caractérisent par leur arrangement spatial global, c'est-à-dire l'orientation des structures secondaires des domaines.

– Topology ou Fold family, classe les domaines selon leurs repliements en tenant compte de la connectivité des structures secondaires.

– Homologous superfamily, à ce dernier niveau, les domaines d'un même groupe sont estimés issus d'un même ancêtre commun. L'estimation est faite de manière manuelle, l'expert vérifie la présence de preuves d'une relation évolutive entre les membres du groupe.

Dans les niveaux suivants, SOLID, la classification n'est plus basée sur la similarité de structure mais sur la similarité de séquence des domaines (S : *similarité* _ 35%, O :

*similarit*é _ 60%, L : *similarit*é _ 95%, I : *similarit*é = 100%, D : domaines uniques).

L'identification de nouveaux domaines à partir de protéines et leur classification au sein de CATH sont schématisées dans la figure 2.1, extraite de l'article de Greene *et al.*

Ce protocole contient une série d'étapes automatiques mais également deux étapes manuelles lorsque le processus automatique n'obtient pas de résultats assez fiables. Grâce à ce protocole, la version 4 de CATH (mars 2013) contient 235 858 domaines structuraux répartis en 2738 superfamilles (niveau H) pour 69 058 protéines annotées (la PDB en contient 100 450 en avril 2015)[21].

2.3. Enrichir les bases de données hiérarchiques, problème d'identification des familles protéiques

La problématique est ici l'insertion d'un nouveau domaine structural au sein d'une classification hiérarchique. SCOP et CATH sont les deux principales classifications hiérarchiques de domaines structuraux (il existe également FSSP Familles of Structurally Similar Proteins qui a classé les protéines en groupes selon les résultats paire à paire de l'algorithme DALI). L'ensemble des domaines structuraux classés de la même manière est regroupé dans un jeu de données nommé SCOPCATH **[19]**. L'inconvénient majeur de ces classifications est la partie manuelle de l'insertion de nouveaux domaines car cela est coûteux en temps et nécessite des experts compétents. Cette insertion de domaines dans la hiérarchie est délicate principalement dans les niveaux superfamilles et se retrouve sous le nom de Family Identification Problème (**FIP**), le problème d'identification des familles et, un niveau au-dessus, le problème d'identification des superfamilles (SFIP). Ce problème est plus difficile que le précédent puisque les domaines structuraux d'une même superfamille peuvent avoir très peu de similarités au niveau de leur séquence. Par conséquent, l'un des grands challenges en biologie structurale est le développement de méthodes et algorithmes rapides, efficaces et automatiques d'assignation d'un domaine structural à une famille.

SCOP et CATH sont largement admises et utilisées par la communauté scientifique malgré leurs divergences mais le problème de classification des domaines structuraux reste ouvert. La principale difficulté dans la création d'une classification est la détection des similarités structurales entre domaines et l'identification de relations de parenté. Cela nécessite de trouver une mesure de similarité permettant de caractériser l'espace des domaines structuraux. Dernier souci et non des moindres, il est nécessaire de comparer chaque domaine à tous les autres domaines du jeu de données pour déterminer une classification

fiable. Le problème d'intégration de nouveaux domaines structuraux au sein des classifications existantes est ici nommé problème d'identification des familles protéiques.

2.4. Estimer la similarité structurale entre deux protéines

L'estimation de la similarité entre deux protéines trouve sa source dans la recherche de la quantification de la ressemblance d'une structure par rapport à l'autre, tant au niveau parenté (homologie, origine commune), qu'au niveau fonctionnel. Deux structures proches supposent ainsi une fonction proche de par le paradigme structure-fonction. De nombreuses Méthodes tentent de capturer cette ressemblance en mesurant la similarité ou la distance entre deux structures. Il en existe trois grands types :

– les mesures de distances entre matrices de distances inter-résidus,

– les mesures de recouvrement de cartes de contacts,

– la mesure de la déviation globale (RMSDc 2.4) des deux protéines après superposition optimale des deux structures.

2.5. Difficulté de la comparaison de deux structures

Tel qu'il est présenté ci-dessus, il suffit d'obtenir un score pour estimer la similarité de deux structures, mais de par nature, le problème de la comparaison de structures est complexe car une grande partie des versions du problème a été démontrée NP-difficile [18]. Cela implique que les algorithmes de résolution du problème sont soit conçus pour une version simplifiée du problème, renvoyant un résultat approché, ne pouvant être garanti comme optimal et de plus, comme Godzik [17] le fait remarquer, il n'y a pas qu'une seule solution au problème. Soit Ces algorithmes sont exacts et explorent un problème combinatoire donc peuvent nécessiter des temps non raisonnables pour retourner une solution.

2.6. Scores basés sur les mesures de distances inter-résidus

la présence de gaps dans l'alignement produit, la mesure de la différence de distances. I. Wohlers a dédié une partie thèse à l'étude et la comparaison de ces scores et en a conclu qu'ils étaient tous pertinents au regard de leur fonction objectif, cela les rend difficiles à comparer. Ces méthodes opèrent en premier lieu un passage de la 3D à la 2D, les matrices de distances étant une représentation en deux dimensions de la structure des protéines. Les matrices sont indépendantes de l'orientation des structures. Les structures similaires ont des distances inter-résidus internes similaires, les scores ici servent donc à mesurer la proportion de distances similaires. Parmi les scores issus de cette catégorie on peut noter le score de DALI, le RMSDd (qui, comme son homologue basé sur les coordonnées nécessite

d'être pondéré par la longueur de l'alignement correspondant), ou encore le score de DAST **[21].**

RMSDd Le RMSDd (Root Mean Square Deviation based on *d*istances) est une mesure de déviation globale basée sur les distances entre résidus d'une même protéine comparées aux distances associées dans l'autre structure. Le RMSDd se calcule sur la base de toutes les distances entre paires de résidus issus d'un alignement. Soit *Ali* = (*P*1,*i*1 , *P*2,*j*1 ; *P*1,*i*2 , *P*2,*j*2 ;. . . ; *P*1,*im*, *P*2,*jm*) un alignement quelconque de longueur *m* des protéines *P*1 et *P*2 (de longueurs respectives |*M*| et |*N*|). Soit *i*, *i*0, *j*, *j*0 quatre indices tels que *i* < *i*0, *j* < *j*0 et *i*(resp. *i*0) est aligné avec *j*(resp. *j*0). Le RMSDd se calcule la manière suivante :

$$RMSDd = \sqrt{\frac{1}{p}\sum_{i<i'j<j'}(d(P_{1,i},P_{1,i'}) - d(P_{2,j},P_{2,j'}))^2}$$

DALI Le principe de DALI est de représenter deux structures protéiques par leurs matrices de distances inter-résidus centrés en leurs *C*_ ou *C*_. Holm et Sander décrivent leur méthode comme le coulissement d'une matrice sur l'autre, les sous-structures similaires apparaissant

des patchs, des zones aux valeurs proches deux à deux, d'une matrice à l'autre. Un peu comme des cartes au trésor qu'il faut superposer à la lueur d'une bougie pour en extraire le chemin complet, le coulissement d'une matrice sur l'autre détermine l'alignement optimal, le meilleur appariement de résidus. L'algorithme de DALI a deux étapes : la première est une comparaison de toutes les sous-matrices de taille 6 (matrices contenant les distances entre les résidus (*i*, .., *i*+5) 2 P_1, (*j*, ..., *j* +5) 2 P_2 de deux protéines P_1, P_2), appelées motifs de contacts. Les sous-matrices similaires sont stockées et servent de graines à la seconde étape qui étend ces motifs pour maximiser le nombre de paires de résidus alignés. Pour chaque alignement DALI calcule le DALI-score, un score optimisé et retourne l'alignement avec le plus grand score. DALI est une heuristique, il en existe une version exacte, DALIX [**16].**

2.3.1 Scores de similarités basés sur la longueur d'un alignement de séquences

La majorité des outils de comparaison de structures protéiques retournent un alignement. A partir de celui-ci, on peut déduire une série de scores de similarité comparables d'un outil à l'autre. Soient *P*1 et *P*2 deux protéines, |*P*1| (resp. |*P*2|) est la longueur de la protéine *P*1 (resp. *P*2) et *Ne* la longueur de l'alignement de *P*2 avec *P*1 issu d'un outil de comparaison. A partir de ces valeurs, nous pouvons établir trois scores de similarité, présentés ci-dessous :

5 Proportion de résidus alignés sur le nombre moyen de résidus :

$$s_{sum} = \frac{2Ne}{|P_1| + |P_2|}$$

6 Proportion de résidus alignés sur le nombre minimal de résidus :

$$s_{min} = \frac{N_e}{min(|P_1|, |P_2|)}$$

7 Proportion de résidus alignés sur le nombre maximal de résidus :

$$s_{max} = \frac{N_e}{max(|P_1|, |P_2|)}$$

Ces scores pourraient servir de bases aux protocoles de classification présentés par la suite, néanmoins ils présentent plusieurs faiblesses : la première est l'absence de contrôle de la qualité de l'alignement fourni. La qualité du score va être totalement dépendante de l'outil de comparaison utilisé. C'est pourquoi un score basé sur un alignement local soumis à un seuil de RMSDc n'aura pas la même signification qu'un score basé sur un alignement global distance inter-résidus dépendant.

Ainsi, lorsqu'il est dit plus haut que les scores allaient être comparables d'un outil à l'autre, cela ne va être possible que dans le cadre d'une recherche globale telle une classification. C'est à dire, pour un même score de similarité, les deux outils vont-ils permettre d'aboutir à la même classification. Une comparaison plus directe de deux outils n'est pas envisageable. Une autre faiblesse réside dans le choix du score de similarité, chacun reflète un aspect de la similarité des protéines considérées et est donc un candidat à l'intégration dans le protocole de classification.

Similarité normalisée [21] Cette similarité est définie comme suit :

$$Snorm = 100 \times \frac{2N_e}{|P_1| + |P_2|}$$

2.7. Identification de la super-famille structurale

2.7.1. Méthode exhaustive ou one to all

Les scores utilisés dans ce chapitre sont commutatifs : *s*(*A*,*B*) = *s*(*B*,*A*). Soit *q* un domaine structural requête et *T* = {*t*1, *t*2, . . . , *tn*} un ensemble de domaines structuraux cibles issus d'une base de données hiérarchique et *S* : *q*×*T* ! *R*+ une fonction de similarité qui associe à toute instance (*q*, *ti*), *ti* 2 *T*, un score de similarité *s*(*q*, *ti*). La recherche exhaustive, nommée méthode one to all, du plus proche voisin de la requête *q* consiste à calculer pour toutes les instances (*q*, *ti*) le score de similarité associé puis de rechercher l'instance pour laquelle ce score est maximal. Cette méthode, décrite par l'algorithme 2.1, est utilisable avec n'importe quelle mesure de similarité s. Parmi tous les scores de similarité proposés, nous utilisons CMO **[17],** via l'outil Apurva qui dénombre le nombre maximal de contacts communs entre

deux domaines structuraux. A titre de comparaison nous avons utilisé le TMscore (via TMalign), qui est une mesure.

Algorithme 1 Méthode *one-to-all*, recherche du plus proche voisin (**NN**)

Require: *q*, T = {*t*1, ..., *tn*} domaine requête, ensemble de domaines cibles

For *ti* 2 T **do**

Compute *s* (*q*, *ti*)

End for

NN = *arg max*(*s*(*q*, *ti*)), *argmax* 2 [0, 1]

Largement répandue. Ces deux méthodes nous permettent également d'observer les différences de comportements entre une méthode exacte et une heuristique.

2.7.2. Identification de superfamilles protéiques par dominance par dominance directe et indirecte

Les travaux de cette section sont le fruit d'une collaboration avec Inken Wohlers, Gunnar Klau, Hristo Djidjev et Rumen Andonov. Les résultats sont extraits des publications produites. Nous remplaçons ici l'ancien score de similarité *s*(*A*,*B*) par la distanc D_{max}(*A*,*B*) prouvée métrique dans le chapitre 2 et nous nommons cette nouvelle mesure maxCMO. Cette caractéristique de la nouvelle mesure est fondamentale car elle permet d'appliquer l'inégalité triangulaire et ainsi évaluer la distance entre deux domaines à partir des distances.

liées à un troisième domaine. De plus, pour nos exemples, nous utiliserons une classification hiérarchique, à plusieurs niveaux donc, en nous plaçant au niveau des superfamilles de domaines.

A la différence de la similarité, plus une distance est faible, plus les protéines sont proches. Ainsi les dominances entre deux instances (définitions 3.1 et 3.2) deviennent :

Dominance exacte basée sur la distance Soient deux protéines t_i et t_j et une requête q. La protéine t_i domine la protéine t_j selon q si :

$$D_{max}(t_i, q) \leq D_{max}, t_j, q)$$

Dominance directe basée sur la distance Soient deux protéines ti et tj et une requête q. La protéine ti domine la protéine tj selon q si :

$$\overline{d}(q, t_i) \leq \underline{d}(q, t_j)$$

2.7.2.1. Caractérisation de la classification, domaines représentants des superfamilles

Le fil conducteur des sections de ce mémoire est de minimiser le nombre d'instances à résoudre, ce pour une simple raison : le nombre important de domaines inclus dans les classifications. Soit *F* une superfamille de domaines structuraux issue d'une classification *C* quelconque et *d* une mesure de distance métrique. *F* peut être caractérisé par un domaine représentatif *RF* et un rayon *rF* définis comme suit :

$$R_F = arg \min_{A \in F} \max_{B \in F} d(A, B)$$

$$r_F = \min_{A \in F} \max_{B \in F} d(A, B)$$

RF est le domaine de F le plus proche de tous les autres domaines de la famille et rf est la plus petite distance maximale entre les domaines, c'est-à-dire la distance entre RF et le domaine dont il est le plus éloigné.

2.7.3. Protocole *k plus proches voisins* (kNN)

La recherche des kNN (*k Nearest neaighbours*) est une extension de la recherche du plus proche voisin. Il s'agit ici de déterminer les k domaines de la classification qui sont les plus proches du domaine requête. Une fois ces kNN identifiés, la requête est assignée à la superfamille majoritairement présente.

Protocole

Soit *q* un domaine requête et T un ensemble de structures. Soient également F 2 C l'ensemble des superfamilles de la classification C. Chaque domaine *t* 2 T est associé à une unique famille F. On suppose que pour chaque superfamille *F* 2 F, les distances exactes inter-domaines ont été mesurées et que le domaine représentant *RF* ainsi que le rayon *rF* ont été déterminés.

L'algorithme 3 commence par estimer les bornes de chaque instance (*q*,*RF*). Puis les cibles *t* sont triées dans deux files de priorité **LB** et **UB** par ordre croissant de la borne correspondante (*d*O, *d*4) calculée selon un temps donné. L'étape suivante est un élagage des structures *t* dominées par *tUB k* . Si à la fin de cette étape, le nombre de structures restantes est égal à k alors l'algorithme retourne la superfamille majoritaire. Sinon le temps est incrémenté et le processus recommence au calcul des nouvelles bornes de l'instance (*q*,*RF*). Cela jusqu'à ce que toutes les bornes convergent ou que le nombre de

structures dans la file **UB** ne soit plus modifié. La seconde étape majeure est une dominance directe, les instances *q, t* sont calculées selon un paramètre temps croissant puis les instances dominées sont élaguées. A la fin du dernier tour (selon la dernière durée de comparaison utilisée -10 secondes ici-), les instances restantes, si la dominance n'est pas totale, sont triées selon LB et les kNN servent à l'assignation de la famille.

Les calculs de maxCMO pour les instances se font avec Apurva, à partir des bornes LB, UB qu'il retourne, les distances (exactes ou bornées) sont calculées. Nous avons ici limité le temps de calcul maximal à 10 secondes, cela implique que certaines instances peuvent ne pas être dominées et dans ce cas, la sélection des kNN est une heuristique puisque les kNN ne dominent pas l'ensemble des structures cibles T .

2.4 Expérimentations

Nous avons testé ce nouveau protocole à l'aide de deux jeux de données issus de SCOPCATH. SCOPCATH est le jeu de données consensuel des bases de données SCOP (1.75) [20] et CATH (3.2.0) [15]. Le jeu de données initial ne contient que des domaines avec un pourcentage d'identité de séquence inférieur à 50%. Cela équivaut à 6759 structures. La version étendue du jeu de données contient l'intégralité des structures consensuelles de SCOP et CATH, soit 67609 domaines structuraux. Ces domaines sont répartis dans 11 classes, 1348 superfamilles et 2480 familles (le tableau 2.1). Les onze classes sont des spécifications des quatre classes usuelles (_ principalement, _ principalement, _ et _ avec feuillets _ parallèles, _ et _ avec feuillets _ anti-parallèles) par des réarrangements des structures secondaires correspondant à des motifs caractéristiques dans les cartes de contacts.

Class	a	b	c	d	e	f	g	h	i	j	k
#str	1195 a	1593	1774	1591	30	103	342	72	11	38	10
#ext	10.796	19.215	17.491	15.679	349	1006	2892	520	43	81	25
#fam	524	516	548	632	6	59	121	32	5	29	8
#sup	303	266	191	191	6	52	82	31	5	29	8

Tableau 2.1 : Répartition des domaines dans les classes pour les jeux de données SCOPCATH (str) et SCOPCATH étendu (ext) ainsi que le nombre de superfamilles(sup) et familles(fam) associés.

Afin de constituer les ensembles de domaines requêtes, nous avons sélectionné aléatoirement un domaine dans chaque famille constituée d'au moins six domaines. Par

conséquent, l'ensemble de requêtes du petit jeu de données contient 236 domaines et celui du jeu de données étendu 1369 domaines.

2.8. Conclusion

Dans ce chapitre nous avons présenté deux bases de données hiérarchisées de domaines protéine CATH et SCOP, et quelques-uns des nombreux scores de similarité qu'il est
Possible de calculer pour entre deux structures et des notions de dominances entre instances, Le protocole standard d'assignation nécessite de comparer un domaine structural requête à tous les domaines de la classification dans le prochain chapitre nous allons aborder l'apprentissage proffond et par la suite exploiter son pouvoir sur la classification des domaines protéique.

Chapitre 03

Apprentissage Profond

3.1. Introduction

L'intelligence artificielle est une discipline scientifique recherchant des méthodes de solution de problèmes à forte complexité logique ou algorithmique, L'apprentissage profond champ d'étude de l'intelligence artificielle. Dans les dernières années Les recherches effectuée par des spécialistes sont en relation avec le l'apprentissage profond et apparaissent dans plusieurs science tel que la biologie par exemple le cas de la prédiction et la classification des domaines protéique.

Dans ce chapitre nous allons présenter tout d'abord les notions en relation avec l'apprentissage profond, et les différents types d'apprentissage automatique et détailler leur principe et les différents modèles pour l'apprentissage profond.

3.2. L'apprentissage profond

3.2.1. Définition

L'apprentissage profond (ou Deep Learning en anglais) est un sous-domaine de l'apprentissage automatique (ou Machine Learning en anglais) été introduit pour la première fois au ML par Dechter (1986**) [D1]**, ce dernier et une branche de l'intelligence artificielle **[D4].** La *figure (3.1)* illustre l'imbrication entre eux. Il est inspiré leurs algorithmes de la structure et le fonctionnement du cerveau humain.

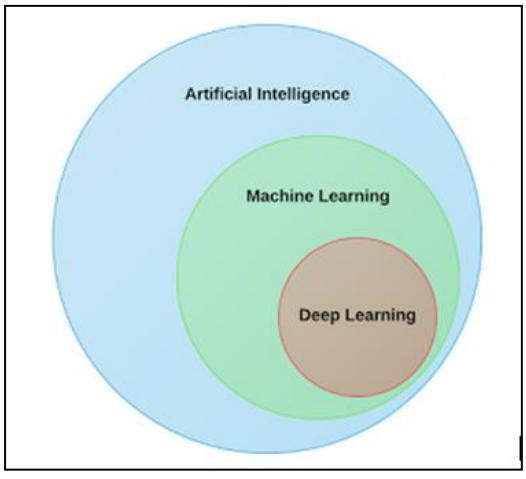

Figure 3.1 : La relation entre l'intelligence artificielle, le ML et le DL.

L'apprentissage profond basé sur les réseaux de neurones artificiels, il est capable d'apprendre et de gérer de larges quantités de données complexes et de résoudre de tache complexes, en ajoutant au réseau des multiples couches de traitement composé des multiples transformations linéaires et non linéaires et apprendre les caractéristiques de

données petit à petit à travers chaque couche avec une intervention humaine minime **[D2][D3].**elle a été introduit aux réseaux neuronaux artificiels par Aizenberg et al (2000) **[D5].**

« Le Deep learning permet à des modèle composés de plusieurs couches de traitement d'apprendre des représentations des données avec de multiples niveaux d'abstraction. » par : Deep learning –Yann LeCu, Yoshua Bengio and Geoffrey Hinton, Nature, 2015.

3.3. L'apprentissage profond et L'apprentissage automatique

3.3.1. L'apprentissage automatique

3.3.1.1. Définition

L'apprentissage automatique (ou apprentissage machine, Machine Learning en anglais) est un sous-domaine de l'intelligence artificielle(IA), qui a évolué a partir de l'étude de la reconnaissance des modèles et de la théorie de l'apprentissage computationnel en intelligence artificielle. Son but est d'entraîner un ensemble d'algorithme sur de grandes quantités de données afin de pouvoir classifier des données futur(la précision dépendant de la quantité et la qualité des données).Cette technique s'appuie sur la développent de programmes informatiques capables d'acquérir de nouvelle connaissances afin de s'améliorer et d'évoluer d'eux-mêmes des qu' ils sont exposer de nouvelles données .ils fonctionnent en construisant un modèle a partir d exemple d'entrées afin de faire des prédiction ou des choix basés sur les données plutôt que de suivre des instruction de programme statiques**[D6].**

L'apprentissage automatique est généralement divisé en :

- L'apprentissage automatique supervisé.
- L'apprentissage automatique non supervisé.
- Apprentissage par Renforcement.

3.3.1.2. Apprentissage supervisé

La forme la plus commune d'apprentissage automatique et l'apprentissage supervisé. L'apprentissage supervisé est une méthode permettant de transformer un jeu de données en un autre, le programme est formé sur un ensemble prédéfini d'exemples de formation, ce qui facilite en suit sa capacité à parvenir à une conclusion précise lorsque de nouvelles données sont fournies **[D6] [D7] [D8].**

Si nous prenons des données sous forme d'exemples avec des étiquettes, nous pouvons alimenter un algorithme d'apprentissage ces paires exemple-étiquette une par une,

permettant à l'algorithme de prédire l'étiquette pour chaque exemple, et de lui donner des informations pour savoir s'il a prédit la bonne réponse ou non. Au fil du temps, l'algorithme apprendra à se rapprocher de la nature exacte de la relation entre les exemples et leurs étiquettes. Lorsqu'il est entièrement formé, l'algorithme d'apprentissage supervisé sera en mesure d'observer un nouvel exemple jamais vu auparavant et de prédire une bonne étiquette pour celui-ci. *Figure 3.2.*

.

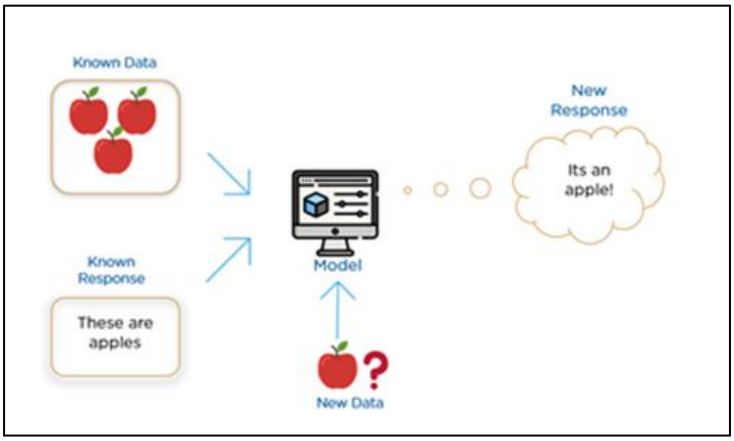

Figure 3.2 : Exemple d'apprentissage supervisé.

On retrouve ce type d'apprentissage dans nombreuses applications courantes suivantes:

Popularité de la publicité: La sélection des publicités qui fonctionnent bien est souvent une tâche d'apprentissage supervisé. De nombreuses publicités que vous voyez lorsque vous naviguez sur Internet y sont placées parce qu'un algorithme d'apprentissage a déclaré qu'elles étaient d'une popularité raisonnable. De plus, son placement associé sur un certain site ou à une certaine requête (si vous vous retrouvez à utiliser un moteur de recherche) est en grande partie dû à un algorithme appris disant que la correspondance entre l'annonce et le placement sera efficace.

Classification du spam: si vous utilisez un système de messagerie moderne, il est probable que vous ayez rencontré un filtre anti-spam. Ce filtre anti-spam est un système d'apprentissage supervisé. Nourris d'exemples et d'étiquettes d'e-mails (spam / non spam), ces systèmes apprennent à filtrer de manière préventive les e-mails malveillants afin que leur utilisateur ne soit pas harcelé par eux. Beaucoup d'entre eux se comportent également de manière à ce qu'un utilisateur puisse fournir de nouvelles étiquettes au système et qu'il puisse apprendre les préférences de l'utilisateur.

La reconnaissance faciale : aussi appelée reconnaissance de visage, consiste à identifier une ou plusieurs personnes automatiquement sur des photos ou dans des vidéos en analysant et en comparant des formes. Typiquement, les algorithmes de reconnaissance faciale extraient les caractéristiques faciales d'individus et les comparent à une base de données pour trouver la meilleure correspondance possible, ce dernier est un processus supervisé **[D11].**

3.3.1.3. Apprentissage non supervisé

L'apprentissage non supervisé, encore appelé apprentissage à partir d'observation, partage une propriété commune avec l'apprentissage supervisé, il transforme un jeu de données en un autre. Mais l'ensemble de données dans lequel il se transforme n'est pas connu ou compris auparavant.

Contrairement à l'apprentissage supervisé. Il ne comporte aucune étiquette. Au lieu de cela, notre algorithme recevrait beaucoup de données et disposerait des outils nécessaires pour comprendre les propriétés des données. À partir de là, il peut apprendre à regrouper, regrouper et / ou organiser les données de manière à ce qu'un humain (ou un autre algorithme intelligent) et créera lui-même les classes puisse entrer et donner un sens aux données nouvellement organisées **[D7] [D12].**

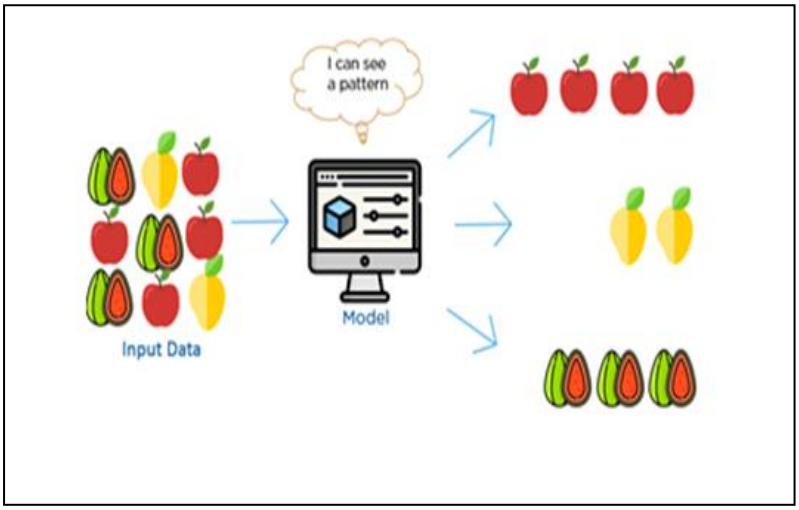

Figure3.3: Exemple Apprentissage non supervisé

3.3.1.4. Apprentissage par Renforcement

Les données en entrée sont les mêmes que pour l'apprentissage supervisé, cependant l'apprentissage est guidé par l'environnement sous la forme de récompenses ou de pénalités données en fonction de l'erreur commise lors de l'apprentissage **[D13]**.

3.3.2. Algorithmes d'apprentissage automatique

L'algorithme d'apprentissage automatique est une évolution de l'algorithme régulier. Il rend vos programmes «plus intelligents», en leur permettant d'apprendre automatiquement des données que vous fournissez. L'algorithme est principalement divisé en:

- Phase d'entraînement.
- Phase de test.

Phase d'entraînement : dans cette phase vous prenez une partie des données d'entrée (données d'entraînement), et faites un tableau de toutes les caractéristiques pour chaque un entrée par exemple la couleur, la taille, la forme la couleur, la longueur, le poids..etc.Vous transmettez ces données à l'algorithme d'apprentissage automatique (classification / régression), et trouver le modèle mathématique le plus adapté de corrélation entre les caractéristiques.

Phase de test : dans la phase de test (ou phase de **vérification**), vous utilisez le modèle qui a été calculé précédemment dans Phase d'entraînement pour la prédiction de chaque entrée de la seconde partie de données (données de test).Enfin l'optimisation se faire afin vise à amenuiser les erreurs**[D9]**.

Les algorithmes d'apprentissage automatique peuvent être classés a partir leur fonctionnement en deux types dans *la figure 3.4* suivante :

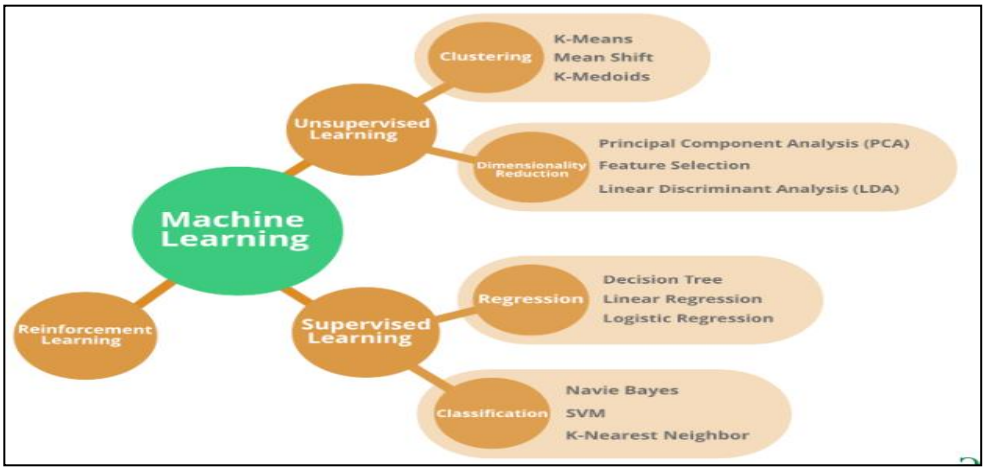

Figure 3.4 : type d'algorithme d'apprentissage automatique.

3.4. La classification supervisée

La classification supervisée cherche à prédire la classe des nouvelles instances en se basant sur des informations connues a priori. Elle est un processus à deux étapes : une étape d'apprentissage et une étape de classification.

Dans l'étape d'apprentissage, un modèle est construit en analysant un jeu de données dit "d'apprentissage" dans lequel la classe de chaque instance est supposée prédéfinie. Soit D = $\{(\boldsymbol{Xi}, \boldsymbol{Yi}) \ \boldsymbol{\epsilon} \{\mathbf{1}, \dots, \boldsymbol{N}\}\}$ un jeu de données d'apprentissage composé de N instances. Chaque instance (Xi = $\{(\boldsymbol{Xi} = \{\boldsymbol{Xi1}, \boldsymbol{Xi2}, \dots, \boldsymbol{Xid}\}, \boldsymbol{Yi} \boldsymbol{\epsilon} \{\mathbf{1}, \dots, \boldsymbol{J}\}$)est représentée par un vecteur de variables de dimension d et d'une variable cible *Yi* indiquant son appartenance à une des *J* classes. Soit *x* et *k* respectivement l'espaces des valeurs d'entré et de sortie. D'une manière plus formelle, l'étape d'apprentissage a pour but d'apprendre, à partir des données d'apprentissage, une fonction $(\boldsymbol{f} : \boldsymbol{x} \rightarrow \boldsymbol{k})$ de telle sorte que *f(X)* est un "bon" prédicateur de la valeur correspondante à *Y*.

Dans l'étape de classification, le modèle construit dans la première étape est utilisé pour classer les nouvelles instances.

Le modèle construit par un algorithme d'apprentissage doit en général remplir un certain nombre de critères. Citons à titre d'exemple :

- Le taux d'erreur doit être le plus bas possible. Ce point peut être mesuré en utilisant plusieurs critères d'évaluation. A titre d'exemple, la précision (ACC), l'air sous la courbe de ROC (AUC), l'indice ARI (Adjusted Rand Index),..., etc.
- Il doit être aussi peu sensible que possible aux fluctuations aléatoires des données d'apprentissage.
- les décisions de classification doivent autant que possible être explicites et compréhensibles.

3.4.1. Présentation de certaines d'algorithme de la classification d'apprentissage automatique

3.4.1.1. k-plus proches voisins

L'algorithme des k-plus proches voisins(KNN) est un des algorithmes de classification les plus simples. Le seul outil dont on a besoin est une distance entre les éléments que l'on veut classifier [D10]. Chaque observation de l'ensemble d'apprentissage est représentée par un point dans un espace à *n* dimensions ou n est le nombre de variables prédictives. Pour prédire la classe d'une observation, on cherche les *k* points les plus proches de cet exemple. La classe de la variable cible, est celle qui est la plus représentée parmi les *k* plus proches voisins. Il existe des variantes de l'algorithme ou on pondère les *k* observations en fonction de leur distance à l'exemple dont on veut classer [11N], les observations les plus éloignées de notre exemple seront considérées comme moins importantes.

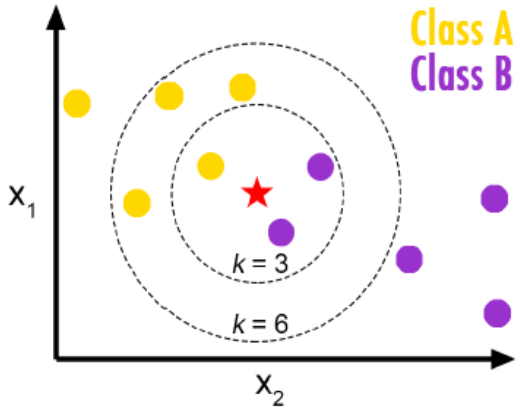

Figure 3.5 : Pour k = 3 la classe majoritaire du point central est la classe B, mais si on change la valeur du voisinage k = 6 la classe majoritaire devient la classe A.

3.4.1.2. Machines à vecteur support

Les machines à vecteur support se situent sur l'axe de développement de la recherche humaine des techniques d'apprentissage. Les SVMs sont des classes de techniques d'apprentissage introduite par Vladimir Vapnik au début des années 90, sont un ensemble de techniques d'apprentissage supervisé destinées à résoudre des problèmes de classification.les SVM sont une généralisation des classifiées linéaires, elles reposent sur une théorie mathématique.

SVM fonctionne par mappage des données à un espace d'attributs haute dimension pour que les points de données puissent être classés, même lorsque les données ne sont pas séparables sur un plan linéaire. Un séparateur entre les catégories est identifié *(figure 2.3)***.** Ensuite, les données sont transformées de sorte que le séparateur puisse être défini comme un hyperplan. Ensuite, les caractéristiques des nouvelles données peuvent être utilisées pour prédire le groupe auquel un nouvel enregistrement doit appartenir **[D14][D15].**

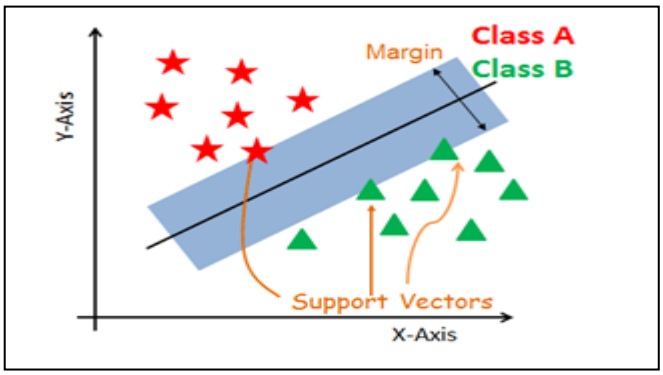

Figure 3.6: Support Vector Machine

3.4.1.3. Classification bayésienne

La classification bayésienne (**Naive Bayes)** est une technique de classification basée sur le théorème de Bayes avec une hypothèse d'indépendance entre les prédicteurs. Le principe de cette théorie est le suivant : Soit *X* un échantillon de données dont la classe est inconnue et qu'on veut la déterminer, et soit *H* une hypothèse (*X* appartient à la classe *C* par exemple). On cherche à déterminer *P(H/X)* la probabilité de vérification de *H* après l'observation de *X*. *P(H/X)* est la probabilité postérieure c'est-à-dire après la connaissance de *X* tandis que *P(H)* est la probabilité à priori représentant la probabilité de vérification de *H* pour n'importe quel exemple de données. Le théorème de Bayes propose une méthode de calcul de *P(H/X)* en utilisant les probabilités *P(H), P(X) et P(X/H)* :

$$P(H/X) = [P(X/H).P(H)] / P(X)$$

P(H/X) est donc la probabilité d'appartenance de *X* à la classe *C, P(H)* la probabilité d'apparition de la classe *C* dans la population et qui peut être calculée comme le rapport entre le nombre d'échantillons appartenant à la classe *C* et le nombre total d'échantillons **[D10].**

3.4.1.4. Réseaux de neurones

Les réseaux de neurones artificiels (RNA) sont inspirés de la méthode de travail du cerveau humain qui est totalement différente de celle d'un ordinateur. Le cerveau humai se base sur un système de traitement d'information parallèle et non linéaire, très compliqué, ce qui lui permet d'organiser ses composants pour traiter, d'une façon très performante et très rapide, des problèmes très compliqués tel que la reconnaissance des formes. Un réseau de neurones est une structure de réseau constituée d'un nombre de nœuds interconnectés par des liaisons directionnelles, Chaque nœud représente une unité de traitement et les liaisons représentent les relations causales entre les nouds. La figure suivante représente une schématisation d'un neurone **[D10].**

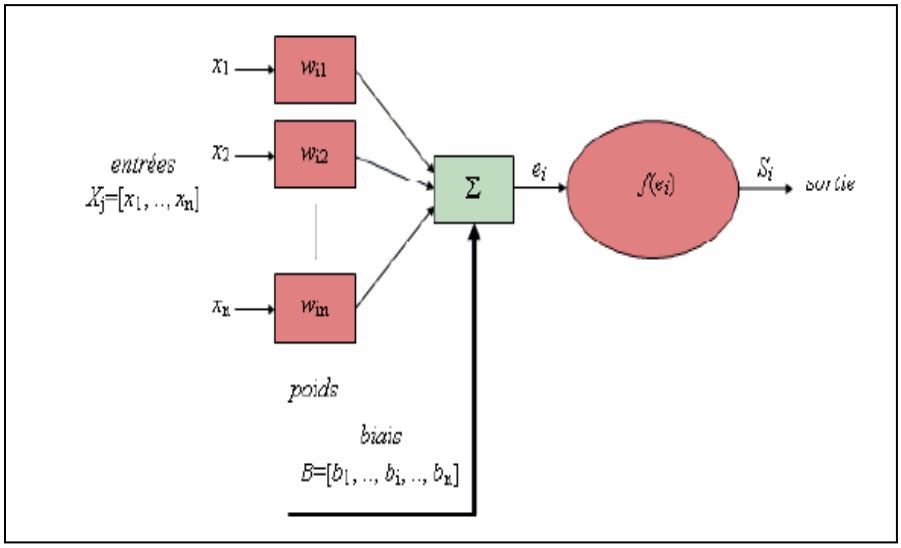

Figure 3.7: Modèle d'un neurone artificiel

La figure 2.11 montre qu'un neurone *k* se constitue de trois éléments basiques :

– Un ensemble de connexions avec les différentes entrées *xi*, pondérée chacune par un poids *wki*,

– Un additionneur permettant de calculer une combinaison linéaire des entrées *xi* pondérées par les coefficients *wki*,

– Un biais *bk* qui permet de contrôler l'entrée de la fonction d'activation,

– Une fonction d'activation f permettant de délimiter la sortie yi du neurone.

Mathématiquement, la sortie *yk* du neurone peut être exprimée par la fonction suivante :

$$yk = f(wk1x1 + wk2x2 + ... + wknxn + bk)$$

3.4.1.5. k-means

L'algorithme *k-means* est l'algorithme de regroupement le plus connu et le plus utilisé, du fait de sa simplicité de mise en œuvre. Il partitionne les données d'une image en K clusters. Contrairement à d'autres méthodes dites hiérarchiques, qui créent une structure en « arbre de clusters » pour décrire les groupements, *k-means* ne crée qu'un seul niveau de clusters.

L'algorithme renvoie une partition des données, dans laquelle les objets à l'intérieur de chaque cluster sont aussi proches que possible les uns des autres et aussi loin que possible des objets des autres clusters. Chaque cluster de la partition est défini par ses objets et son centroïde. Le *k-means* est un algorithme itératif qui minimise la somme des distances entre chaque objet et le centroïde de son cluster.

La position initiale des centroïdes conditionne le résultat final, de sorte que les centroïdes doivent être initialement placés le plus loin possible les uns des autres de façon à optimiser l'algorithme. *K-means* change les objets de cluster jusqu'à ce que la somme ne puisse plus diminuer. Le résultat est un ensemble de clusters compacts et clairement séparés, sous réserve qu'on ait choisi la bonne valeur K du nombre de clusters. Les principales étapes de l'algorithme *k-means* sont :

1. Choix aléatoire de la position initiale des K clusters.
2. (Réaffecter les objets à un cluster suivant un critère de minimisation des distances (généralement selon une mesure de distance euclidienne).
3. Une fois tous les objets placés, recalculer les *K* centroïdes.
4. Réitérer les étapes 2 et 3 jusqu'à ce que plus aucune réaffectation ne soit faite. **[EMC 15]**

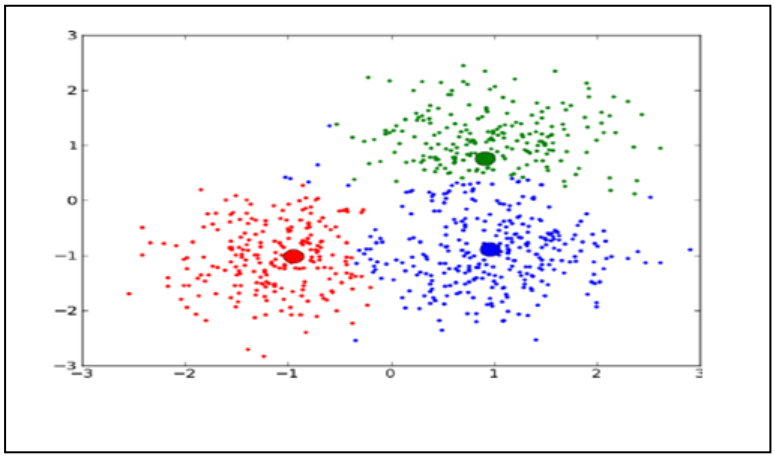

Figure 3.8 : L'algorithme k-means regroupe les données en k cluster, ici k = 3. Les centres de gravité sont représentés par de petit cercle.

3.5. Différence entre l'apprentissage profond et l'apprentissage automatique

Un processus d'apprentissage automatique commence par l'extraction manuelle de caractéristique pertinente à partir de données en s'appuyant sur ces caractéristiques, un modèle est créé, mais l'apprentissage profond ignore ces étapes manuelles. Par exemple pour classer des images avec la Machine Learning *(la figure 3.9)*, les choix des caractéristiques et de classificateur doivent être effectués manillèrent. Avec l'apprentissage profond, l'extraction de caractéristique et le processuel de modélisation sont automatique **[D17].**

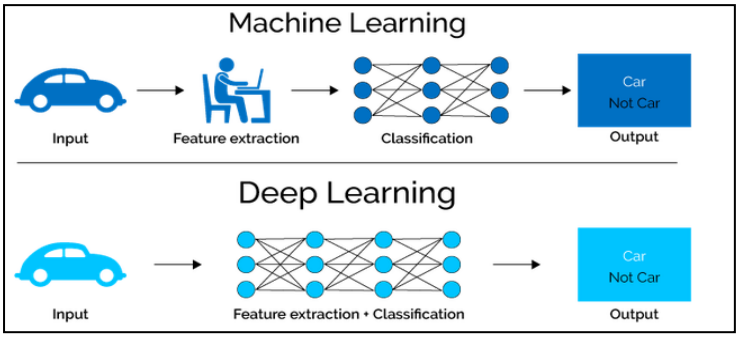

Figure 3.9 : la différence entre l'apprentissage profond et l'apprentissage automatique

L'entrainement des modèles s'effectue à laide de vastes ensembles de données labellisées et d'architectures de réseaux neurones qui apprennent des caractéristiques directement depuis les données, sans avoir à effectuer une extraction manuelle . Une autre différence majeure est le fait que l'algorithme de Deep Learning évolue avec les données.

Pour réussir une application de Deep Learning, vous avez besoin d'un volume de données très important(des milliers d'image par exemple, sauvent quelques millions et des entrée de très grande dimension) pour entrainer le modèle, en plus d un ou plusieurs GPU (processeur graphique) pour traiter les donnes rapidement. Si vous n'avez pas ces éléments, il est préférable d'utiliser l'apprentissage automatique plutôt que l'apprentissage profond **[D17].**

3.6. L'importance de l'apprentissage profond

L'apprentissage automatique n'est pas utile lorsque vos travaillez avec des données de grandes dimensions, c'est-a-dire que nous avons un grand nombre d'entrée et de sorties. Ne pas résoudre des problèmes cruciaux d'intelligence artificielle comme NLP, la reconnaissance d'image etc.

Extraction de caractéristiques est un des grands défis des modèles d'apprentissage machine traditionnels

Cette extraction automatisée de caractéristiques permet aux modèles de Deep learning d'atteindre un taux de précision particulièrement élevé pour la tache de vision par ordinateur *(la figure 3.10)* **[D17].**

Les modèles d'apprentissage profond son capables de se concentrer sur les fonctionnalités appropriées par eux-mêmes nécessitant peu de conseils de part du programmeur.

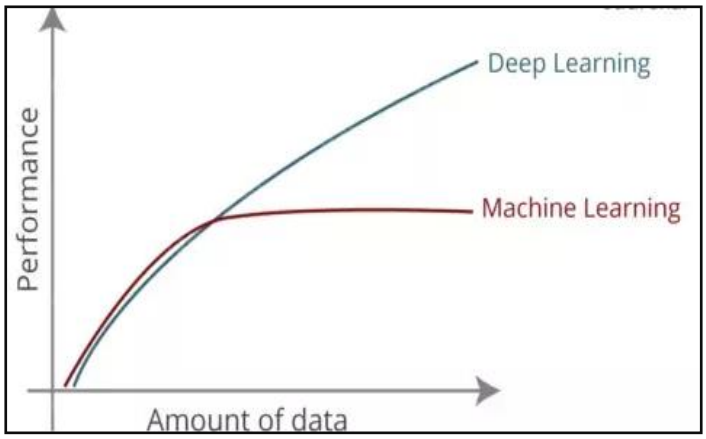

Figure 3.10 : illustration de performance l'apprentissage profond et l'apprentissage automatique

3.7. Fonctionnement d'apprentissage

Le modèle d'apprentissage base sur le réseau de neurones qui a trois composants principaux : couche d'entrée, couche cachée ou plusieurs couches de sortie. Le terme « profond » se rapporte généralement au nombre de couches cachées du réseau de neurones.les réseaux de neurones classique ne comportent que 2 à 3 couches cachées, tandis que les réseaux profonds peuvent en compter jusq'à150 [D18].

L'idée est utiliser la structure de couche réseau neuronal en empilant plusieurs couche les unes sur les autres, de manière a facilité le mécanisme décomposition. Par conséquence, chaque couche d'un réseau de neurones profond (DNN) fonctionnement comme une couche une seule transformation pour extraire davantage les données [D19].

Le réseau de neurones le plus simple anticipation. Il contient un calque d'entrée, un ou plusieurs calques masqués et un seul calque de sortie (figure 3.11). Dans chaque couche peut avoir un nombre différent de neurones et chaque couche connecte à la couche adjacent **[D4].**

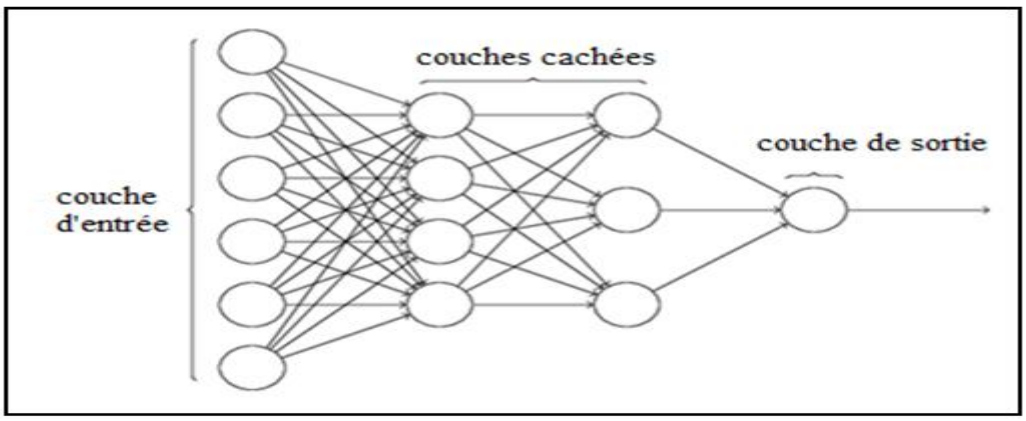

Figure 3.11 : Réseau de neurones avec une

Un réseau de neurones est défini comme un ensemble de nœuds (appelés nuerons) connectés vais des liaisons dirigées(flèche), chaque liaison représente un connexion entre la sortie de neurone et l'entrée du l'autre ,a un poids importance, chaque nœuds exécute une fonction de nœud sur son single entrant pour générer un seul sortie [D3].les valeur d'entrée, ou en d'autre termes, nos données sous-jacents, sont transmises via ce réseau de couches masquées jusqu'à ce qu'elle convergent vers la couche sortie. La couche en sortie correspond à notre prédiction : il peut s'agir d un nœud si le modèle ne génère qu'un nombre ou quelques nœuds il s'agite d un problème de classification multi-classe.la forme à l'intérieur des neurones dans les couches centrales représente une fonction d'activation ($\boldsymbol{f(X) = \frac{1}{(1+e-x)}}$) qui appliquée a la valeur du neurone avent de le transmettre à la sortie **[D20].**

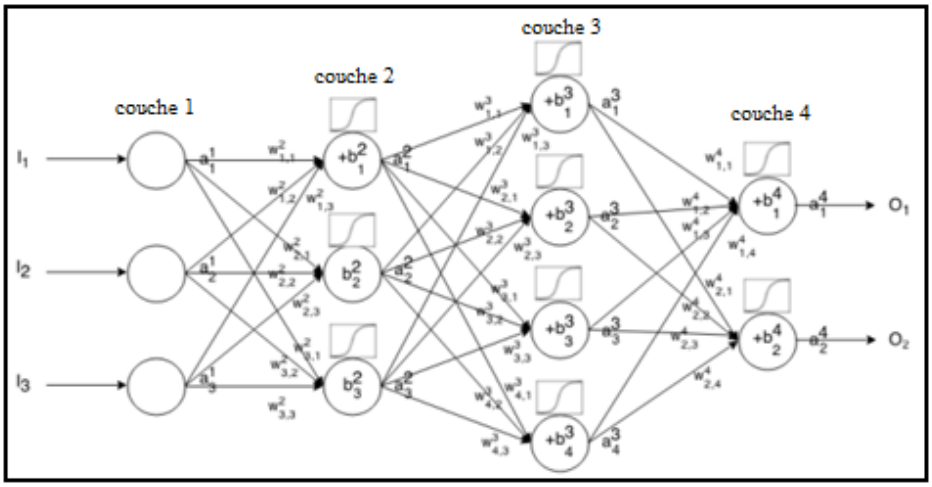

Figure 3.12 : Topologie de réseau de neurones profond.

Les couches cachées d'un réseau de neurones apportent des modifications aux données pour éventuellement déterminer quelle est sa relation avec la variable cible. Chaque nœud a un poids et multiplier sa valeur d'entrée par ce poids. Pour déterminer ce que devraient être ces petits poids, nous utilisons généralement un algorithme appelé Back Progration

- **Back Progration**

Back Progration est une forme abrégée de «propagation vers l'arrière des erreurs». C'est une méthode standard de formation des réseaux de neurones artificiels. Cette méthode permet de calculer le gradient d'une fonction de perte par rapport à tous les poids du réseau.

- **Fonction d'activation**

- **Fonction Sigmoïde :** a la forme mathématique $\sigma(x)=1 / (1+e-x)$, elle réduit la sortie entre 0 et 1. En particulier, les grands nombres négatifs deviennent 0 et les grands nombres positifs deviennent 1ce qui rend le problème de classification plus facile.
- **Fonction Tanh :** Tangente hyperbolique a la forme mathématique $tanh(x)=2\sigma(2x)-1$, *elle* comprise entre (-1 et 1). Comme le neurone sigmoïde mais sa sortie est centrée sur zéro.
- **RelU: (***Rectified Linear Unit)* elle calcule la fonction $f(x) = max(0, x)$.convertir tout ce qui inferieur à 0 en 0**[D16].**

- **Softmax** : étend cette idée à un monde à plusieurs classes. C'est-à-dire que Softmax attribue des probabilités décimales à chaque classe d'un problème à plusieurs classes. La somme de ces probabilités décimales doit être égale à 1. Cette contrainte supplémentaire permet de faire converger l'apprentissage plus rapidement qu'il ne le ferait autrement. Softmax est mis en œuvre via une couche de réseau de neurones juste avant la couche de sortie. La couche Softmax doit comporter le même nombre de nœuds que la couche de sortie **[D21].**

3.8. Modèles d'apprentissage profond

La plupart des réseaux de neurones dans les années 1980 ne formaient qu'une seul couche en raison d'un coût de calcul et de la disponibilité des données. Récemment, nous pouvons nous permettre d'avoir plus de couche cachées dans nos réseau de neurones, d'ou le surnom d'apprentissage profond.les déférent types de réseaux neurones disponibles à l'utilisation ont également proliféré, des modèles tel que les réseaux de neurones convolution (CNN), les réseaux de neurones récurrents (RNN) et LSTM.

3.8.1. Réseaux de neurones Convolutionnels (CNN)

Un réseau de neurones convolutionnels ou (convolution neural networks en engluais) un des types des réseaux de neurones profonds le plus utilise des couches à convolution 2D .cette architecteur sont donc parfaitement adaptées au traitement de données 2D tell que les images. Le réseau de neurones à convolution s'appuient sur plusieurs dizaines, voire plusieurs centaines de couches cachées pour appendre à identifier les caractéristiques d'une image, la complicité de la caractéristique apprise augmente avec le nombre de couches caché du réseau **[D22] [D17].**

CNN est une séquence de couches, et chaque couche transforme un volume d'activations en un autre par une fonction différentiable. Les trois principaux types de couches pour construire ce type de réseau sont : couche convolutive, couche de pooling et couche entièrement connectée.

-La couche convolutive : C'est la couche la plus importante et le cœur des éléments constitutifs du réseau convolutif, et c'est aussi elle qui effectue le plus de calculs lourd.

-La couche de pooling : Il est courant d'insérer périodiquement une couche Pooling dans ce type d'architecture. Sa fonction est de réduire progressivement la taille spatiale de la représentation pour réduire le nombre de paramètres et de calculs dans le réseau, et donc de contrôler également l'overfitting.

-La couche entièrement connectée : Comme nous l'avons mentionné précédemment, les neurones d'une couche entièrement connectée ont des connexions complètes à toutes les activations de la couche précédente **[D23].**

Les réseaux de neurones convolutionnels ont gagné en popularité pour leur succès dans les problèmes de vision par ordinateur, les CNN avérés efficaces pour la prédiction et la classification, sont applications incluent la compréhension vidéo, la reconnaissance vocale et la compréhension du traitement du langage naturel **[D22] [D17].**

3.8.2. Réseaux de neurones récurrents (RNN)

Est une class de réseaux de neurones artificiels dans lesquels les connexions entre les nœuds forment un graphe dirigé le long d'une séquence temporelle. Les réseaux de neurones récurrents, ont été cocus pur traiter les problèmes de prédiction de séquence **[D17].**

Les réseaux de neurones traditionnels ne peuvent pas le faire, et cela est un inconvénient majeur. Par exemple, imaginons qu'on souhaite classer quel genre d'événement se produit à chaque étape du film. On ne sait pas très bien comment un réseau de neurones traditionnels pourrait utiliser son raisonnement sur les événements précédents dans le film pour en informer les derniers.
Les réseaux de neurones récurrents résolvent ce problème. Ce sont des réseaux avec des boucles qui permettent à l'information de persister.

Dans la *figure 3.13* ci-dessus, un segment de réseau neuronal; "*A*" regarde une entrée *Xt* et fournit une valeur ht. Une boucle permet de passer des informations d'une étape du réseau à l'autre.

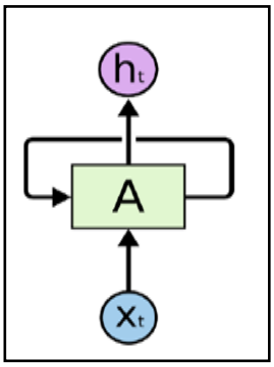

Figure 3.13 : les réseaux neurones ont des boucles.

Ces boucles rendent les réseaux neuronaux récurrents un peu mystérieux. Cependant, si vous pensez un peu plus, il s'avère qu'ils ne sont pas tous différents d'un réseau de neurones normal. Un réseau de neurones récurrent peut être considéré comme des copies multiples du même réseau, chacune transmettant un message à un successeur comme le montre *la figure 3.14*. Considérez ce qui se passe si nous déroulons la boucle:

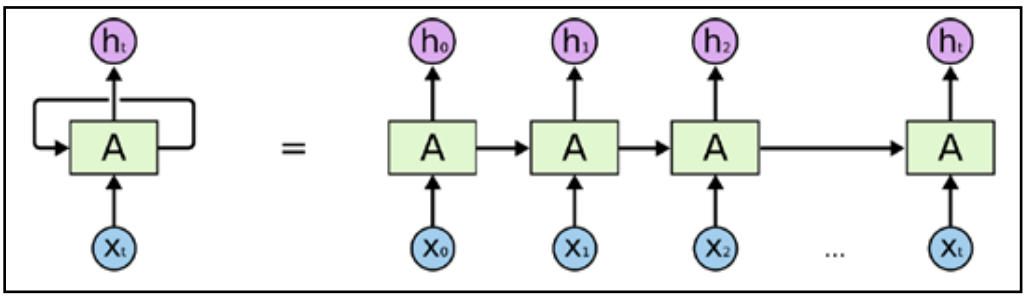

Figure 3.14 : Un réseau récurrent déroulé.

Cette nature en chaîne révèle que les réseaux neuronaux récurrents sont intimement liés aux séquences et aux listes. Ils sont l'architecture naturelle du réseau de neurones à utiliser pour de telles données.

Au cours des dernières années, il y a eu un succès incroyable en appliquant les RNN à une variété de problèmes: la reconnaissance de la parole, la modélisation du langage, la traduction, le sous-titrage des images ...etc. **[18N]**

3.8.3. Mémoire à long terme (LSTM)

Les réseaux de mémoire à long terme à court terme généralement appelés simplement (LSTM Long Short Term Memory) sont un type spécial de RNN. Ils ont été introduits par Hochreiter Schmidhuber (1997). Les Réseaux neuronaux récurrents présentés dans la section précédente sont capables d'apprendre des règles de mise à jour de séquence arbitraire en théorie. Dans la pratique, cependant, ces modèles oublient généralement rapidement le passé .C'est ce qu'on appelle le problème de la disparition de gradient et c'est pourquoi ils ont inventé le LSTM. La cellule LSTM est une adaptation de la couche récurrente qui permet aux signaux plus anciens des couches profondes de se déplacer vers la cellule du présent **[D23].**

3.9. Domaines d'applications de Deep Learning

Conduite automatisée : Les chercheurs du secteur automobile ont recours au Deep Learning pour détecter automatiquement des objets tels que les panneaux stop et les feux de circulation. Le Deep Learning est également utilisé pour détecter les piétons, évitant ainsi nombre d'accidents.

Aérospatiale et défense : Le Deep Learning sert à identifier des objets à partir de satellites utilisés pour localiser des zones d'intérêt et identifier quels secteurs sont sûrs ou dangereux pour les troupes au sol

Recherche médicale : À l'aide du Deep Learning, les chercheurs en cancérologie peuvent dépister automatiquement les cellules cancéreuses. Des équipes de l'Université de Californie à Los Angeles (UCLA) ont conçu un microscope qui génère un ensemble de données de grande dimension afin d'entraîner une application de Deep Learning à identifier avec précision des cellules cancéreuses.

Automatisation industrielle : Le Deep Learning participe à l'amélioration de la sécurité des employés travaillant à proximité d'équipements lourds, en détectant automatiquement les situations dans lesquelles la distance de sécurité qui sépare le personnel ou les objets des machines est insuffisante.

Électronique : Le Deep Learning est utilisé pour la reconnaissance audio et vocale. Par exemple, les appareils d'assistance à domicile qui répondent à votre voix et connaissent vos préférences fonctionnent grâce à des applications de Deep Learning.

3.10. Limites de l'apprentissage profond

Cela quelques limites de l'apprentissage profond :

- pour résoudre des problèmes de plus en plus complexes, il né est pas suffisant d'ajouter toujours plus de couches les deux grosses problématique de l'apprentissage profond;
- La difficulté d'apprentissage
- La complexité calculatoire croissant avec le nombre de couches.
- L'apprentissage profond exige une puissance de calcul considérable.
- l'entrainement couteux en ressources (calcul, mémoire, …).
- L'établissement de réseaux de neurones profond pose un défi : déterminer le nombre de couche cachée ainsi que le nombre de neurones par couches.

3.11. Conclusion

L'apprentissage profond est le domaine le plus émergeant de l'apprentissage automatique et a apporté une contribution importante dans divers domaines de recherche. Cela a permis de surmonter les inconvénients des méthodes traditionnelles en rendant les systèmes moins complexes et plus rapides. L'apprentissage profond a été utilisé avec le traitement automatique du langage dans plusieurs domaines de recherche, ce qui est très prometteur et constitue un succès. Dans ce chapitre nous avons exposé la technique de l'apprentissage profond, ainsi que ses avantages, et ses limites dans le prochain chapitre nous allons présenter notre système de classification des domaines protéique en utilisant une technique de l'apprentissage profond**.**

Chapitre 04

Conception

1.1. Introduction

Dans le cas de classifier un domaine protéique, nous devons généralement ; prétraiter un jeu de donnes de séquence protéine ; extraire des vecteurs de fonction numériques de la séquence prétraité ; former le modèle d'apprentissage et de valider le modèle. Dans ce chapitre on va expliquer l'architecture détaillée de notre système explication, quelques processus, la méthode utilisée pour la transformation de données séquentiels, en fin l'algorithme choisi pour l'implémentation.

1.2. Conception globale

Dans cette section, nous allons essayer de donner un vue général de notre système.

1.2.1. Architecture générale

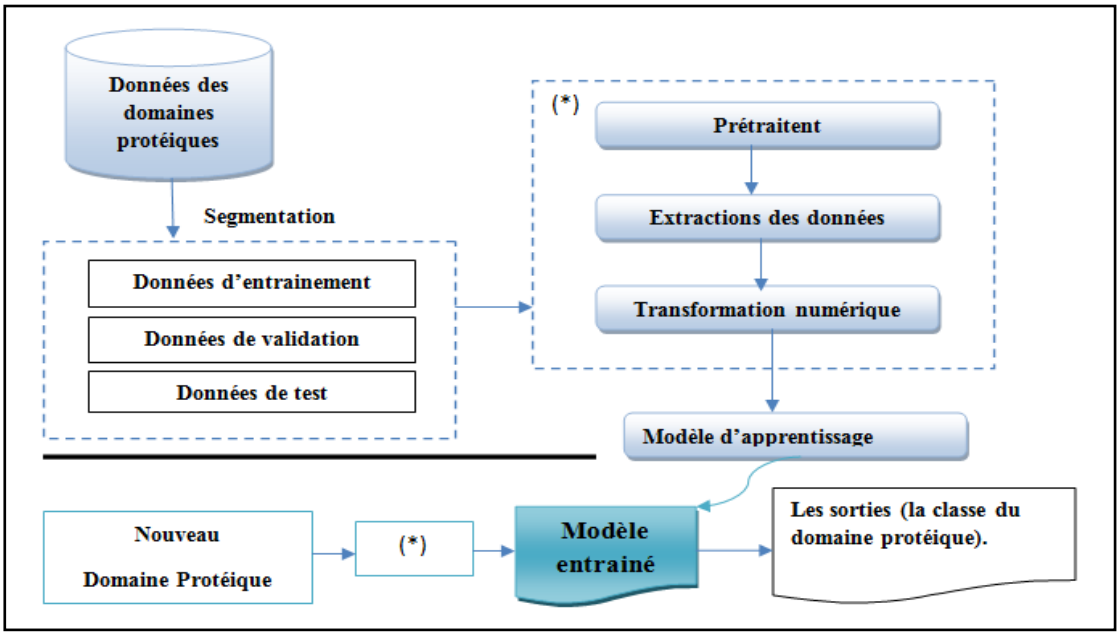

Figure 4.1 : Représentation de l'architecture générale du système.

Dans notre système, nous traiterons les données du domaine protéique qui seront divisées en trois parties (données d'entraînement, données de validation, données de test) et passeront par la phase de prétraitement (*), comme illustré à la figure 4.1. Ensuite, nous créons un modèle d'apprentissage qui prend les entrées de données traitées à l'étape (*) et s'entraîne sur une partie des données (données d'entraînement). Le système final consistera à insérer un nouveau domaine protéique pour classification.

1.3. Conception détaillée

Dans cette section, nous allons essayer d'expliquer l'architecture détaillée de notre système.

1.3.1. Architecture détaillée

L'architecture détaillé de notre système illustré à la *figure 4.2*, et on va l'expliquer dans la section suivante.

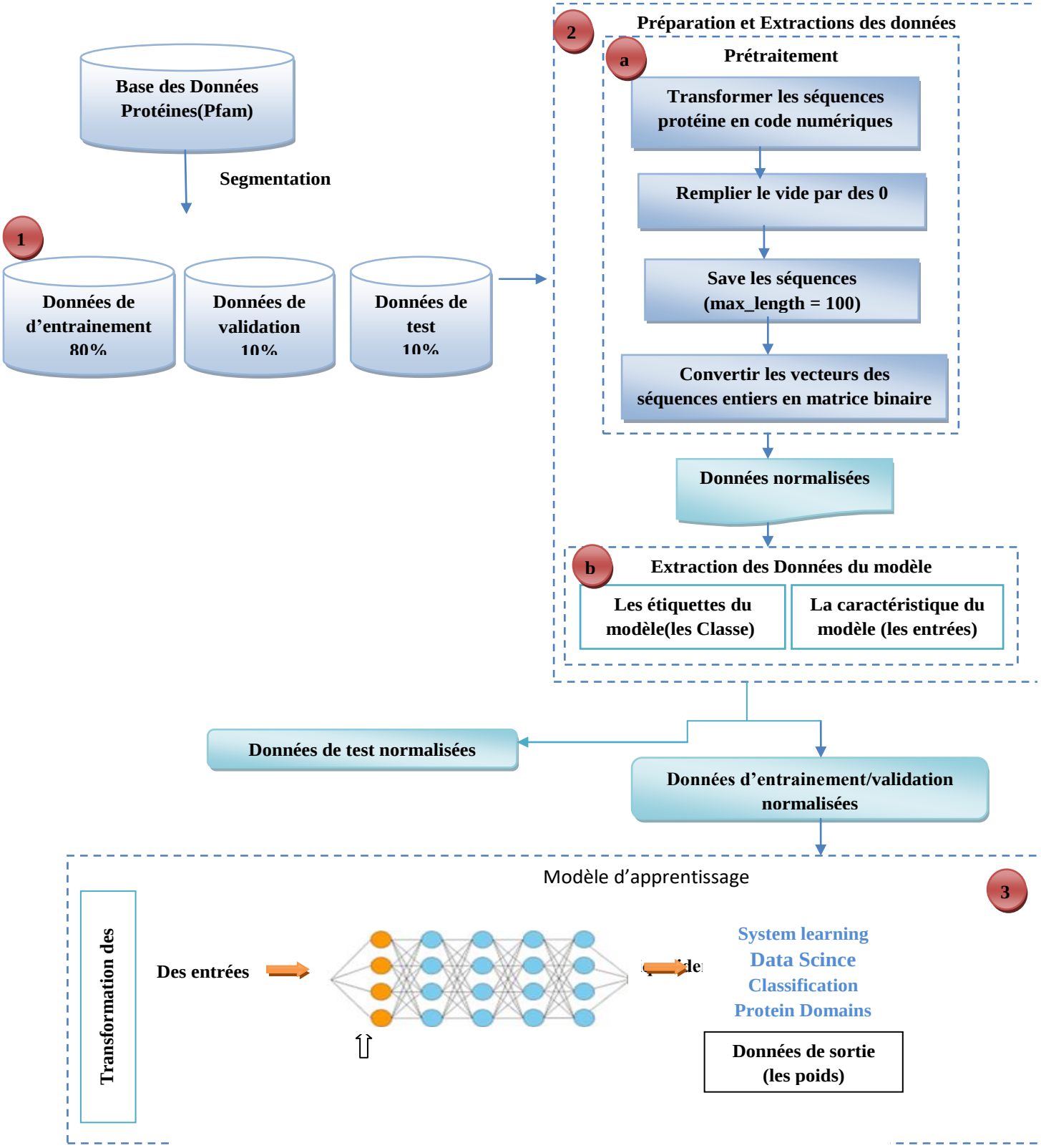

Figure 4.2 : Représentation de l'architecture détaillée de système.

1.3.2. Explication

Phase 1: Collection de données

La tâche de Classification des Domaines protéique est comme suivante:

Un domaine protéique est une la séquence d'acides aminés, la classification est de prédire à quelle classe il appartient.

Chaque séquence d'acides aminés de protéine à l'un des membres de la famille des protéines qui basée sur l'ensemble de données Pfam.

La base de données de : https://www.kaggle.com/googleai/pfam-seed-random-split/data.

La base de données a cinq fonctionnalités, elles sont les suivantes:

Family_id : le nom de famille protéine **.**

Sequence_name : c'est le nom qui signifie la séquence, sous la forme "uniprot_accession_id / start_index-end_index" dans pfm **.**

family_accession : c est le numéro d'accès sous la forme PFxxxxx.y où xxxxx dans (Pfam), est l'accession à la famille et y est le numéro de version. aligned_sequence : une seule séquence de l'alignement de séquences multiples avec le reste des membres de la famille dans la graine, avec des espaces conservés.

Sequence : est un Séquence d'acides aminés pour un domaine spécifique, Il existe 20 acides aminés très courants.et les et 4 acides aminés assez rares: X, U, B, O, Z.

Les family_accession Ce sont généralement les étiquettes du modèle et les Sequence Ce sont généralement les caractéristiques d'entrée du modèle.

family_id	sequence_name	family_accession	aligned_sequence	sequenc
MORN_2	Q8EI47_SHEON/428-449	PF07661.13	LHGEFRNQTSSGQLLELI.NFNH	LHGEFRNQTSSGQLLELINFN
exin_cytopl	H2TB23_TAKRU/1240-1793	PF08337.12	.MPFLDYKTYTDCNFFLPSKDGAND.....AMITRKLQIPE.....	MPFLDYKTYTDCNFFLPSKDGANDAMITRKLQIPEARRAIVAQALN
T_RNaseH	H3H8E9_PHYRM/405-501	PF17917.1	DYSRRFHVFADAS.GH.QIGGVIVQ.....	DYSRRFHVFADASGHQIGGVIVQGRRILACFSRSMTDTQKKYSTME
posase_20	Q981X5_RHILO/224-313	PF02371.16	VEAYQAMRGASFLVAVIFAAEI.GDV.RR.FDTPPQLMAFLGLVPG...	VEAYQAMRGASFLVAVIFAAEIGDVRRFDTPPQLMAFLGLVPGERS
act_memb	MMPS4_MYCLE/16-154	PF05423.13	LSRIWIPLVILVVLVVGGFVVYRVHSYFASEKRESYADSNLGSSKP...	LSRIWIPLVILVVLVVGGFVVYRVHSYFASEKRESYADSNLGSSKP

Figure 4.3 : Ensembles des domaines protéiques avec leur famille.

La base divisée aléatoirement en trois :

- Train : l'ensemble de données pour l'entrainement de modèle 80%.
- Dev : l'ensemble de données pour l'évaluation et validation de modèle au moment d'entrainement 10%
- Test : l'ensemble de données pour le test 10%.
  Chaque base contient un nombre de classe qui sont dans familly_accessoir :

- nombre de classe dans Train: 17929
- nombre de classe dans Train Val: 13071
- nombre de classe dans Train in Test: 13071

Phase 2: Préparation de données

Phase 2.a Prétraitement

Les séquences d'acides aminés sont représentées avec leur code à 1 lettre correspondant. Pour construire des modèles d'apprentissage en profondeur, nous devons transformer ces données textuelles en une forme numérique que les machines peuvent traiter. Apres la précision des séquences qui ont la même 1000 classe, nous utilisons une méthode d'encodage pour la même chose en considérant 20 acides aminés communs, car d'autres acides aminés rares sont moins nombreux.

La fonction (create_Dict()) crée un dictionnaire de 20 acides aminés considérés avec des valeurs entières dans l'ordre incrémentiel à utiliser ultérieurement pour le codage d'entiers. Pour chaque séquence d'acides aminés non alignée la fonction (integet_encoding (data)) remplace, chaque un lettre dans chaque séquence d'acides aminés non alignée par une valeur entière à l'aide du dictionnaire de codes créé. Si le code n'est pas présent dans le dictionnaire, la valeur est simplement remplacée par 0, ne considérant ainsi que 20 acides aminés courants. Exemple, séquence non alignée :

[PHPESRIRLSTRRDAHGMPIPRIESRLGPDAFARLRFMARTCRAILAAAGCAAPFEEFSSADAFSSTH VFGTCRMGHDPMRNVVDGWGRSHRWPNLFVADASLFPSSGGGESPGLTIQALALRT.]

La séquence convertira en données numériques comme celle-ci :

[13, 7, 13, 4, 16, 15, 8, 15, 10, 16, 17, 15, 15, 3, 1, 7, 6, 11, 13, 8, 13, 15, 8, 4, 16 , 15, 10, 6, 13, 3, 1, 5, 1, 15, 10, 15, 5, 11, 1, 15, 17, 2, 15, 1, 8, 10, 1, 1, 1, 6 , 2, 1, 1, 13, 5, 4, 4, 5, 16, 16, 1, 3, 1, 5, 16, 16, 17, 7, 18, 5, 6, 17, 2, 15, 11 , 6, 7, 3, 13, 11, 15, 12, 18, 18, 3, 6, 19, 6,

15, 16, 7, 15, 19, 13, 12, 10, 5, 18, 1, 3 , 1, 16, 10, 5, 13, 16, 16, 6, 6, 6, 4, 16, 13, 6, 10, 17, 8, 14, 1, 10, 1, 10, 15, 17]

Le remplissage de la séquence pour le modèle d'apprentissage ce fait selon de prendre les séquences qui sont la même famille de 1000 classes, les séquences sont effectués avec une longueur maximale (max_length = 100) et remplit avec 0 si la longueur totale de la séquence est inférieure à 100 sinon tronque la séquence jusqu'à une longueur maximale de 100, le résultat est un ensemble des entier. Enfin, chaque code dans les séquences est converti en un vecteur par la fonction to_categorical() ,qui convertit un vecteur de séquence entiers en matrice binaire ce dernier est les entrée de modèle pour l'entrainement .

Phase 2.b :

Les données traitent sont les classes de sortier ,les données l'entrainmment ,validation et le test qui sont convert en codage binaire .

```
y_train = to_categorical(y_train_le)
y_val = to_categorical(y_val_le)
y_test = to_categorical(y_test_le)
```

Classes : En prenez les 1000 classes premier à partir la base de donnée d'entrainement Train et les classe luis même qui sont dans la base de donnée validation et Test, pour assure que les 1000 classe dans les trois bases de données sont les même.

Phase 3 : Modèle d'apprentissage

La dernière étape du cadre d'apprentissage consiste à former un modèle à l'aide des fonctionnalités créées à l'étape précédente (étape de la préparation des données).

Le modèle d'apprentissage profond contient principalement trois types de couches:

- la couche d'entrée.
- plusieurs couches cachées.
- la couche de sortie

Phase 3.a: Transformation de données

Comme tous les modèles d'apprentissage nécessitant une transformation de données dans la couche d'entrée,

Comme nous le savons déjà, les machines ou les algorithmes ne peuvent pas comprendre les caractères / mots ou les phrases, ils ne peuvent prendre que des nombres en entrée qui incluent également des fichiers binaires. Mais la nature inhérente des données textuelles est non structurée et bruyante, ce qui rend impossible toute interaction avec les machines **[22].** Les performances et la précision des algorithmes d'apprentissage automatique et d'apprentissage approfondi dépendent fondamentalement du type de technique d'ingénierie de caractéristiques utilisée **[22]**.

Donc, pour le codage de données il doit appliquer le petit processus ci-dessous (la *figure 4.4)*.

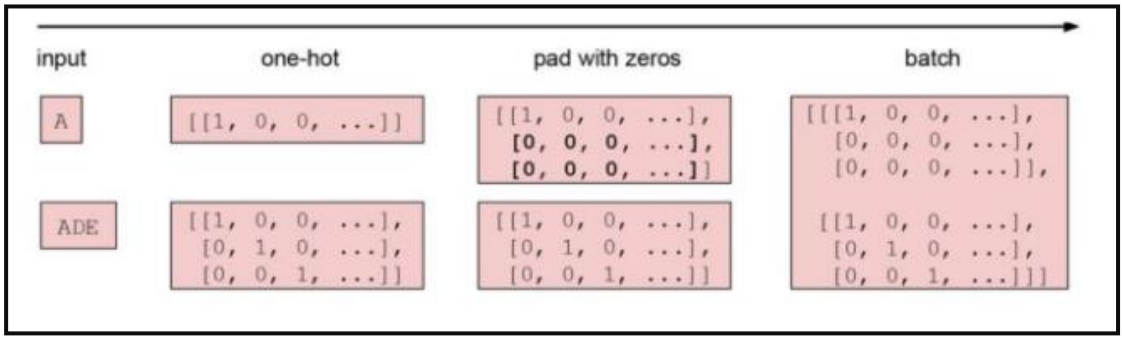

Figure 4.4 : Transformation de données

Phase 3.b : Choix d'algorithme

Il existe de nombreux choix de modèles d'apprentissage profond qui peuvent être utilisés pour former un modèle final. Nous allons implémenter l'algorithme LSTM (Long Short-Term Memory en anglais)

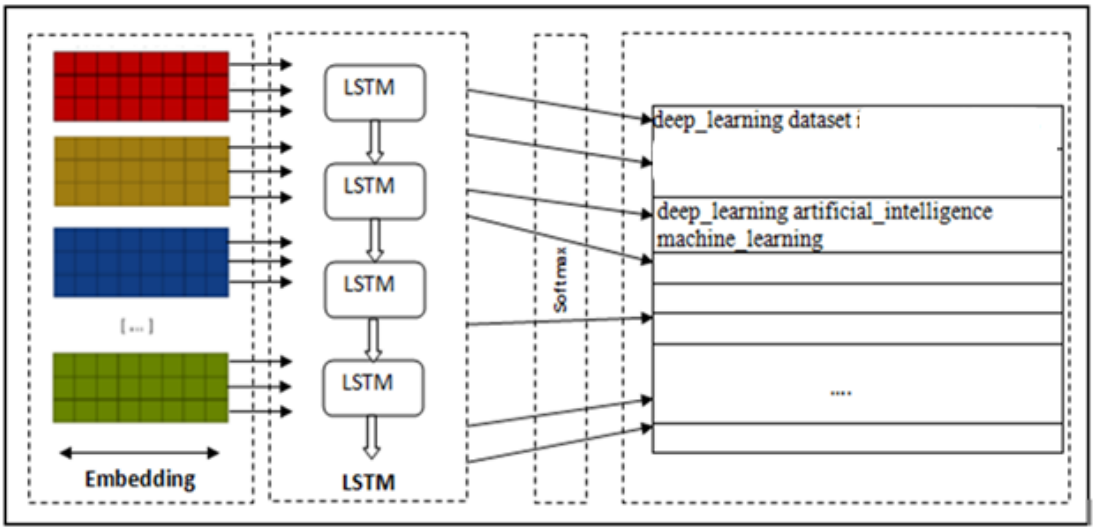

Figure 4.5 : Modèle d'apprentissage

Embedding : Avant de pouvoir modéliser un problème à travers un algorithme de Deep-Learning, il est souvent nécessaire d'effectuer un bon nombre de transformations sur les données. par le choix de l'algorithme utilisé. En générale la première couche apprend à transformer des séquences en vecteurs au cours du processus d'apprentissage, de sorte que chaque un est mappé sur un vecteur dense de valeurs réelles.

1.4. Conclusion

L'étape la plus importante pour atteindre les résultats souhaités en utilisant l'apprentissage profond est la bonne connaissance de la sélection des données appropriées et la meilleure façon de les représenter. Dans ce chapitre, nous avons exposé la conception de notre système avec l'explication de méthode de transformation de notre données ainsi que l'algorithme d'apprentissage choisi dans le chapitre suivant nous allons entamer l'implémentation de tel système.

Chapitre 05

Implémentation

2.1. Introduction

Il ne fait aucun doute que le plus important pour résoudre un problème dans le domaine d'informatique et un bon choix pour le langage de développement. Le développement du domaine d'informatique est devenu un problème multiple, et l'accès a la résolution des méthodes les plus simple est très important. Nous avons donc choisi le langage de programmation optimal qui nous a considérablement aidés à obtenir de bons résultats. Dans ce chapitre on va définir le langage choisi pour résoudre notre problème, l'environnement de développement et on va détailler sur l'application.

2.2. Chois de langage de programmation

Lorsque vous choisissez un langage de programmation qui se spécialise dans l'apprentissage profond, il doit considérer les compétences répertoriées dans les offres d'emplois actuels ainsi que les bibliothèques disponibles dans différent langage qui peuvent être utilisées pour les processus d'apprentissage profond. Python est le langage de programmation le plus recherché dans le domaine de l'apprentissage automatique et l'apprentissage profond. Python est suivi par java, puis par le R, puis c++ **[1].**

2.2.1. Langage de programmation (Python):

Python est un langage de programmation, interprété car, avant de pouvoir les exécuter, un logiciel spécialisé se charge de transformer le code du programme en langage machine, multi-paradigme et multiplatformes, et placé sous une licence libre qui vous permet de travailler rapidement et d'intégrer les systèmes plus efficacement. Python put utilise pour gérer des données volumineuses et effectuer des calculs complexes .il existe ce qu'on appelle des bibliothèques qui aident le développeur a travaille sur des projets particuliers .plusieurs bibliothèques peuvent ainsi être installées pour, par exemple les interfaces graphique en python .

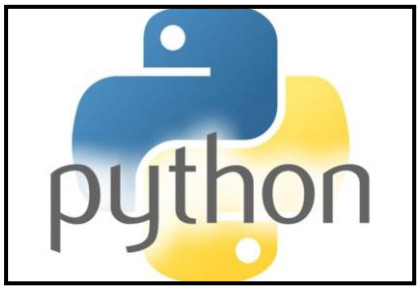

Figure 5.1 : logo en langage python

Ce choix a été motivé par les raisons suivant :

- L'une des principaux langages parmi le langage approprié pour le programmation de problème d'apprentissage profond.
- Il dispose un grand nombre de bibliothèque pour le traitement des séquences pour faire la classification telle que keras.utils, preprocessing.sequence, … .
- Un langage simple, productif et utilisable dans presque tous les domaines et systèmes.

2.3. Environnement de développement

PyCharm

Pour l'environnement de développement, nous avons utilisé l'environnement du PyCharm (JetBrains PyCharm Community Edition 2019.2 x64).

Figure 5.2 : L'environnement de PyCharm.

Anaconda

Installe Anaconda pour l Configurer un environnement Python pour l'apprentissage profond.

Figure 5.3 : logo Anaconda.

2.4. Affichage des Statistiques de base

Pour la lecteur base de données train utilise la fonction (read_data()), puis en peut voir le résumé avec (data_train.info()),qui donne la liste de toutes les colonnes avec leurs types de données et le nombre de valeurs non nulles dans chaque colonne. nous avons également la valeur de range index fournie pour l'axe d'index.

Nombre de séquences

Apres comptons le nombre de codes (acides aminés) dans chaque séquence non alignée, en trouve La plupart des séquences d'acides aminés non alignées ont un nombre de caractères compris entre 50 et 250.

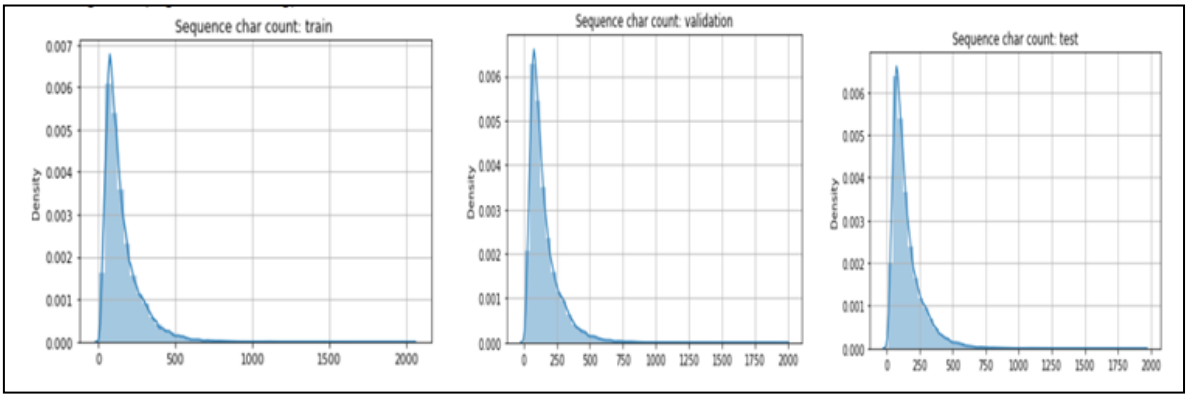

Figure 5.4 :illustraction de nombre de séquences.

Fréquence du code de séquence

En trouvons également la fréquence de chaque lettre de code acide aminé dans chaque séquence non alignée par la fonction (get__freq_code(data)) ,Puis Nous dessinons le diagramme ci-dessous:

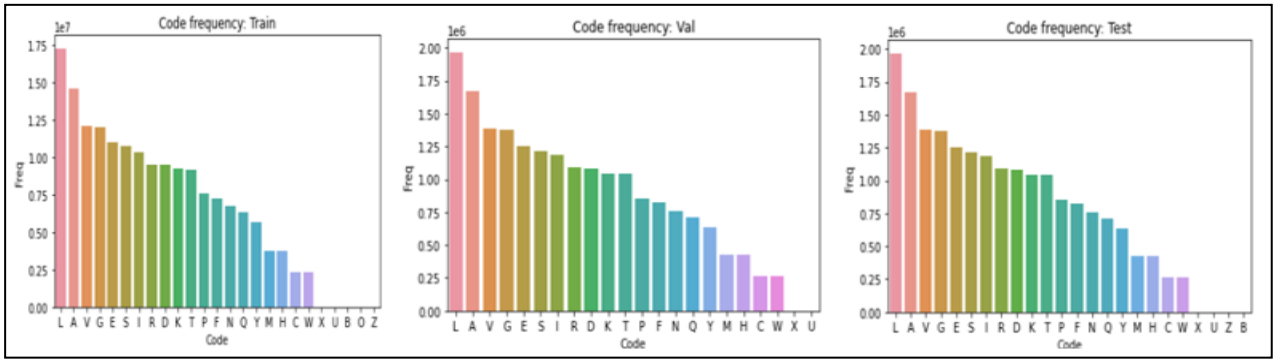

Figure 5.5 :illustraction de nombre de séquences.

Le code d'acides aminés le plus fréquent est la leucine (L) suivie de l'alanine (A), de la valine (V) et de la glycine (G).

Comme nous pouvons le voir, les acides aminés rares (c'est-à-dire X, U, B, O, Z) sont présents en très moins grande quantité. Par conséquent, nous ne pouvons considérer que 20 acides aminés naturels communs pour le codage de séquence à l'étape de prétraitement.

2.5. Processus générale de la création du modèle d'apprentissage

2.5.1. Les étapes de la création de model

Le processus de création d'un modèle d'apprentissage doit suivre les étapes suivantes (figure), en commencent par le chargement des données jusqu'à la validation du modèle.

```
--------------- prosessus de code ---------------

    =====> import libraries
    =====> Dataset preparation
              -looding dataset
              -pre_processing
              -split the dataset(training/validation)
    =====> Converting sequence to features
    =====> Builde Model training
    =====> Training Model
    =====> Save Model
```

Figure5.6 : Processus de la création du modèle d'apprentissage.

2.5.2. Importer les bibliothèques

Avant la création de modèle, nous devons importer un ensemble de bibliothèque disponibles dans les package Keras en python .tel que :

```
from keras.preprocessing.sequence import pad_sequences
from keras.callbacks import EarlyStopping
from keras.layers import Input, Dense, Dropout, Flatten, Activation
from keras.layers import Embedding, Bidirectional, LSTM
```

2.5.3. Prétraitement des domaines protéique

La figure (ci-dessous) représente le processus pour faire la classification des domaines protéiques avec quelque fonction prédéfinie en python.

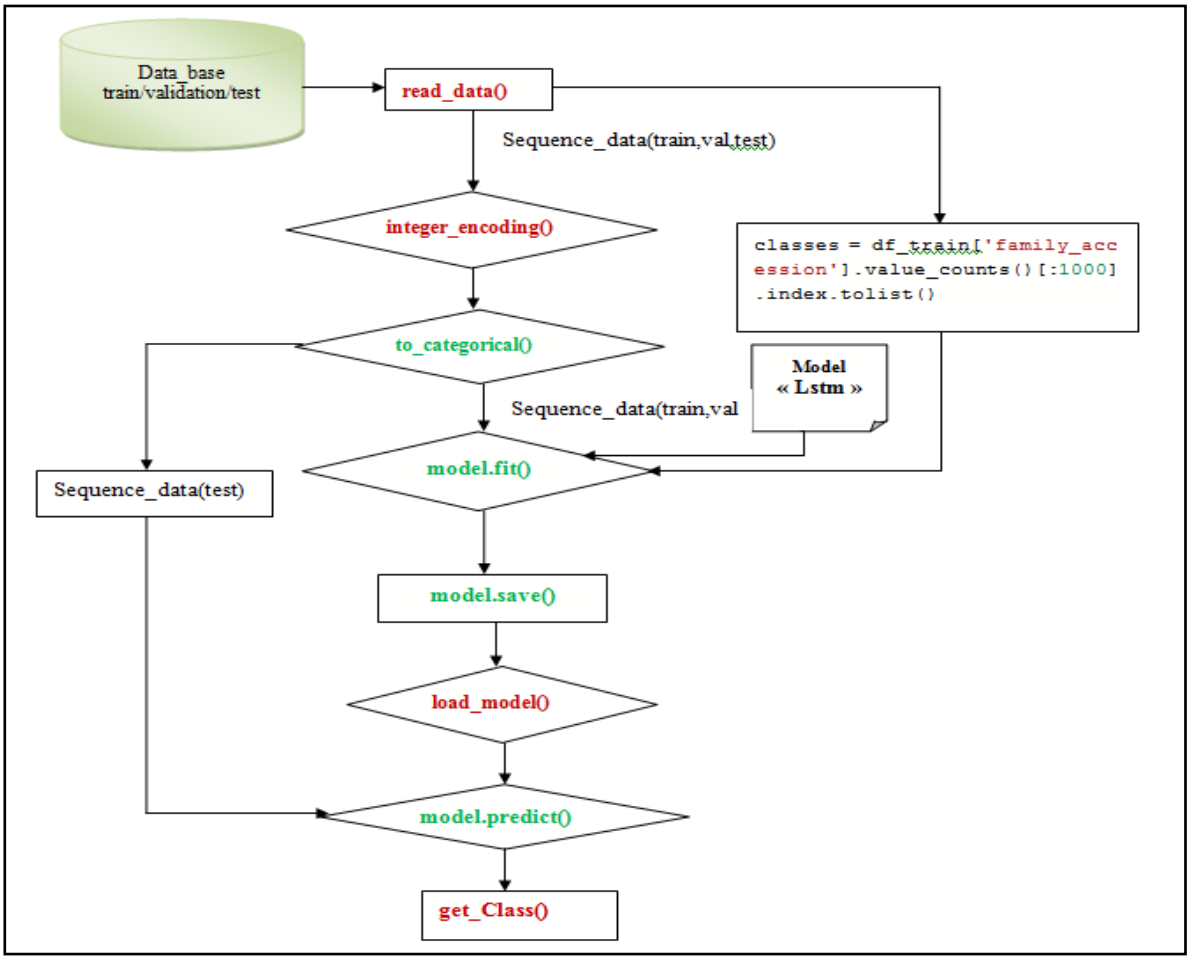

Figure 5.7: processus de l'exécution.

2.5.4. création de modèle

L'étape la plus importante de notre projet est l'étape de la création du modèle, la *figure* 4.4 ci-dessous représente des quelques instructions de la création du modèle.

Pour obtenir un taux de reconnaissance du modèle plus élevé il doit changer quelques paramètres à chaque itération d'entrainement et vérifier le taux, tel que le nombre de couches cachés …

```
x_input = Input(shape=(100,21))
bi_rnn = Bidirectional(LSTM(64, kernel_regularizer=l2(0.01),
                            recurrent_regularizer=l2(0.01) ,
                           bias_regularizer=l2(0.01)))(x_input)
x = Dropout(0.3)(bi_rnn)

# softmax classifier

x_output = Dense(1000, activation='softmax')(x)

model1 = Model(inputs=x_input, outputs=x_output)
model1.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])

model1.summary()
```

Figure 5.8 : illustration des quelque instruction de la création du modèle.

```
history1 = model1.fit(
    train_ohe, y_train,
    epochs=50, batch_size=256,
    validation_data=(val_ohe, y_val),
    callbacks=[es])
```

D'après le training du model Lstm :

Résultat d'entrainement de modèle :

Epoch 1/50

1739/1739 [==============================] - 33s 19ms/step - loss: 6.2013 - accuracy: 0.0367 - val_loss: 5.2869 - val_accuracy: 0.0747

Epoch 2/50

1739/1739 [==============================] - 32s 18ms/step - loss: 5.1040 - accuracy: 0.0946 - val_loss: 4.7809 - val_accuracy: 0.1499

.....

Epoch 46/50

1739/1739 [==============================] - 32s 19ms/step - loss: 1.1468 - accuracy: 0.8641 - val_loss: 0.8946 - val_accuracy: 0.9323

Epoch 47/50

1739/1739 [==============================] - 32s 19ms/step - loss: 1.1406 - accuracy: 0.8647 - val_loss: 0.8686 - val_accuracy: 0.9397

Epoch 48/50

1739/1739 [==============================] - 32s 19ms/step - loss: 1.1383 - accuracy: 0.8652 - val_loss: 0.8657 - val_accuracy: 0.9424

Epoch 49/50

1739/1739 [==============================] - 33s 19ms/step - loss: 1.1348 - accuracy: 0.8652 - val_loss: 0.8629 - val_accuracy: 0.9397

Epoch 50/50

1739/1739 [==============================] - 32s 19ms/step - loss: 1.1322 - accuracy: 0.8657 - val_loss: 0.8667 - val_accuracy: 0.9378

Le modèle est formé avec 33 époques obtenir une perte de (0,386) avec une précision de (95,8%) pour les données de test.

4.5.5 Prédiction d'un exemple

Dans la prédiction en test un exemple à partir la base de données test.

```
model1= keras.models.load_model('drive/My Drive/Lstm_model1.h5')
pred = model1.predict(np.expand_dims(test_ohe[5], axis=0))
out = np.argmax(pred, 1)
```

2.6. Presentation de lntrface graphique

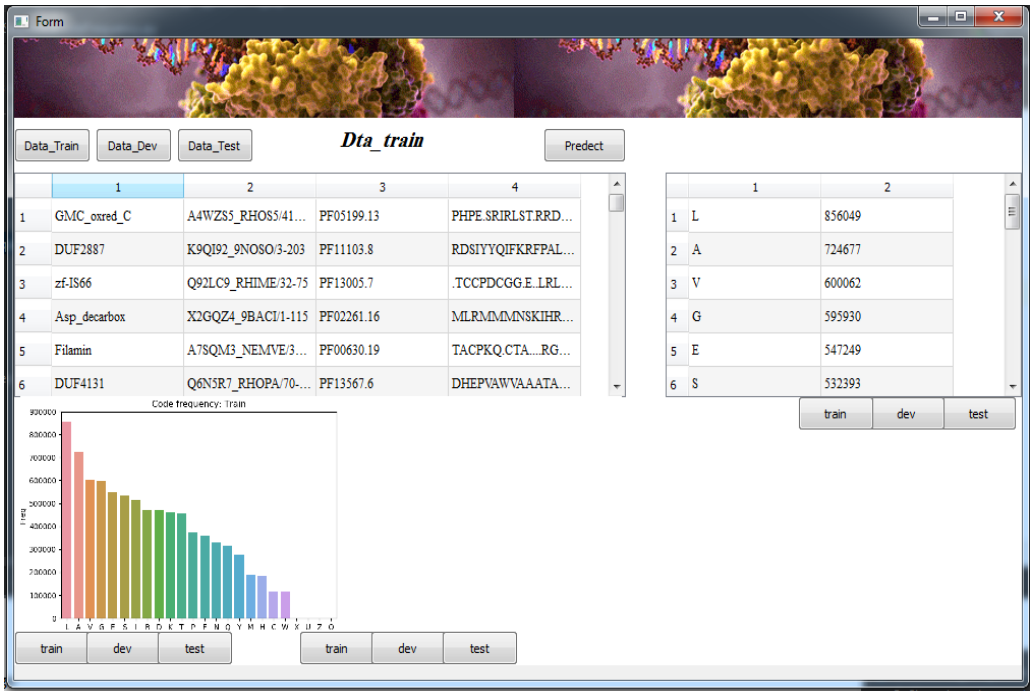

Figure 5.9 : interface graphique de l'application.

3. Conclusion

Les modèles accules d'apprentissage profond reposent sur un bon traitement des données et sur le choix de l'algorithme approprié, mais les spécialistes ne sont pas encore parvenus à définir des paramètres statique ni un modèle général permettant de résoudre tous les problèmes similaires, mais ils ont atteint le soi-disant apprentissage par transfert. Ce chapitre, nous avons expliqué comment nous traitons les domaines protéiques dans le but de classification et sélectionnons un modèle qui convient a nos données.

Conclusion générale

L'apprentissage profond a révolutionné la plus récente ces dernières années, particulièrement son entrée dans plusieurs domaines, a réalisé le progrès significatif dans tous les domaines et, particulièrement avec le traitement de langage naturel, nous ne parlons pas de l'analyse de sentiment et l'extraction d'informations très précisément …, sans mentionner profondément l'apprentissage et comment effectif il est pour des développeurs pour atteindre ce niveau de développement scientifique.

Les techniques d'apprentissage profonds ont été appliqués avec succès à de nombreuses tâches bioinformatique , conduisant à des systèmes efficaces, mais parfois à la taille du réseau (nombre accru de couches et de neurones) et au temps de formation sont interdits pour une utilisation efficace. Malgré cet inconvénient, il y a encore peu de travaux de recherche sur les moyens de trouver des moyens plus efficaces de former un réseaux de neurones profonds ou de trouver une structure optimale (sans former des centaines de réseaux différents) [8].

Ce mémoire a passé en revue d'utilisation de l'apprentissage profond avec la classification supervisée des domaines protéiques, notamment les problèmes de la difficulté de les analyser et l'extraction du contenu le plus important des domaines.

Parce que l'apprentissage profond est un domaine en constante innovation, il est important de garder à l'esprit que les algorithmes, les méthodes et les approches continueront de changer.

Bibliographie

[1] Claverie, J.-M., Audic, S., et Abergel, C. (1999). La Bioinformatique: une discipline stratégique pour l'analyse et la valorisation des génomes. *Conference Proceeding:Rencontres de Lu miny.*

[2]Andrade et Sander. (1997). From genome data to biological knowledge, Current Opinion in Biotechnology. Journal of Bioinformatics.

[3]https://www.futura-sciences.com/sante/definitions/adn-mitochondrial-genomique-156/

[4]Housset,C. et Raisonnier, A.(2009). Biologie Moléculaire. Université Pierre et Marie Curie.

[5] Arnaud,F.(2009). Classification d'ARN codants et d'ARN non-codants. Thèse de doctorat,Université des Sciences et Technologies de Lille.

[6] https://fr.wikipedia.org/wiki/Protomique.

[7] Hans-Joachim, B. (2007). Algorithms Aspects of bioinformatics. Ouvrage, Natural

Computing Series, Springer.

[8] Modelisation et Prediction de la Structure des Proteines Transmembranaires

[9]Kafri M, Metzl-Raz E, Jona G, Barkai N. 2016. The Cost of Protein Production.

[10] Lesk, A. (1988). *Computational Molecular Biology : Sources and Methods for Sequence Analysis*. Oxford University Press.

[11]Halabi.N, Rivoire.O, Leibler.S et Ranganathan.R, Proteinsectors : evolutionary units of three-dimensional structure, 2009.

[12] C A Orengo, A D Michie, S Jones, D T Jones, M B Swindells, and J M Thornton. CATH–a hierarchic classification of protein domain structures. *Structure (London, England : 1993)*, 5 :1093–1108, 1997.

[13]http://biochimej.univangers.fr/Page2/COURS/7RelStructFonction/3Structure/3ProteinStructure/1ProteinStructure.htm

[14]https://www.ebi.ac.uk/training/online/course/introduction-protein-classification-ebi/protein-classification/what-are-protein-domains

[15] C A Orengo, A D Michie, S Jones, D T Jones, M B Swindells, and J M Thornton. CATH–a hierarchic classification of protein domain structures. *Structure (London,England : 1993)*, 5 :1093–1108, 1997.

[16] Inken Wohlers, Noël Malod-Dognin, Rumen Andonov, and Gunnar W Klau. CSA : comprehensive comparison of pairwise protein structure alignments. *Nucleic acids research*, 40(Web Server issue) :W303–9, July 2012.

[17] a Godzik. The structural alignment between two proteins : is there a unique answer *Protein science : a publication of the Protein Society*, 5(7) :1325–38, July 1996.

40[18] D. Goldman, S. Istrail, and C.H. Papadimitriou. Algorithmic aspects of protein structure similarity. *40th Annual Symposium on Foundations of Computer Science (Cat. No.99CB37039)*, 1999.

31[19] Gergely Csaba, Fabian Birzele, and Ralf Zimmer. Systematic comparison of SCOP and CATH : a new gold standard for protein structure analysis. *BMC structural biology*,9 :23, January 2009.

[20] Alexey G. Murzin, Steven E. Brenner, Tim Hubbard, and Cyrus Chothia. SCOP : A structural classification of proteins database for the investigation of sequences and structures. *Journal of Molecular Biology*, 247(4) :536–540, April 1995.

[21] Mathilde .l Similarités et divergences, globales et locales, entre structures protéiques, *de l'Université Européenne de Bretagne,2015*

[22] Akshay K.,Adarsha S., *Natural Language Processing Recipes Unlocking Text Data with Machine Learning and Deep Learning using Python*

[D4] Josh Patterson & Adam Gibson ,2017 .*Deep learning A Practitioner's Approach*, 1ére(Ed), O'Reilly Media, inc., 1005 Gravenstien Highway North, Sebastopol, CA 95472 , Mike loukides & Tim McGovern, 532p, pp(. 28)
[D2] J. Schmidhuber, "Deep learning.," Scholarpedia, vol. 10, no. 11, p. 32832, 2015.
[D3]http://link.spinger.com/article/.

[D1] R. Dechter and J. Pearl, The cycle-cutset method for improving search performance in AI applications. University of California, Computer Science Department, 1986.

[D5] I. Aizenberg, N. N. Aizenberg, and J. P. Vandewalle, Multi-Valued and Universal Binary Neurons: Theory, Learning and Applications. Springer Science & Business Media, 2013.

[D6] Annina S., Mhima S,S.VenKatesan3, D.R. Ramesh Babu, An Overview of Machine Learning and its applications. International journal of Electrical Sciences & Engineering (IJESE)

[D7] Andrew W.Trast,2019.grokking Deep Learning.

[D8]yann LeCun, Yoshua Bengio &Geoffery Hinton., my 2005 Review Deep Learning

[D9] https://www. https://www.edureka.co/blog/machine-learning-tutorial/

[D10] Dr. Abdelhamid DJEFFAL, Cours Fouille de données avancée.2014.

[1dd] **BOUCHER, ALAIN.** *OUTIL D'AIDE A L'ANNOTATION*. 22 janvier 2007.

[D11] https://fr.mathworks.com/discovery/face-recognition.html

[D12] Vincent Bouchet, 2017 Mémoire de master Machine learning.

[D13] Al-albad and Y. Ouli, Initiation à l'apprentissage automatique.

[D14] BOUCHER, ALAIN, *Outil d'aide A L'annotation.* janvier 2007.

[D17]https://www.saagier.com/fr/blog/qu-est-ce-que-le -deep-learning/.

[D15]https://www.ibm.com/support/knowledgecenter/fr/SS3RA7_sub/modeler_mainhelp_client_ddita/clementine/svm_howwork.html

[D16]https://cs231n.github.io/neural-networks-1/

[D18] https://fr.mathworks.com/discovery/deep-learning.html .

[D19]Chirtoper. W,Gerasimos S,Gerhard W., Felexible Deep Learning on natural language Processing. Department of Knowledge Engineering, Masstricht University, The Netherlands.

[D20] Yoav Goldberg, 2015. A Primer on Neural Network Models for natural Language Processing.

[D21]https://developers.google.com/machine-learning/crash-course/multi-class-neural-networks/softmax?hl=fr.

[D22] Yoon Kim, 2014. Convolution Neural Networks for sentence Classification New York University.