الجمهورية الجزائرية الديمقراطية الشعبية
وزارة التعليم العالي والبحث العلمي

UNIVERSITÉ BADJI MOKHTAR - ANNABA
BADJI MOKHTAR – ANNABA UNIVERSITY

جامعة باجي مختار – عنابـــــــــــة

Faculté : Sciences de l'Ingénieur
Département : Informatique
Domaine : Mathématique-Informatique
Filière : Informatique
Spécialité : Gestion et Analyse des Données Massive

Mémoire

Présenté en vue de l'obtention du Diplôme de Master

Thème :

Apprentissage Profond basé CNN pour la Reconnaissance des Molécules Actives/Inactives à partir d'Images 2D

Présenté par : *Talaa Ihab*

Encadrant : *Mohamed Ben Ali Yamina* *Pr* *UBMA*

Jury de Soutenance :

Azizi Nabiha	Pr	UBMA	Président
Mohamed Ben Ali Yamina	Pr	UBMA	Encadrant
Zenakhra Djamel	Dr	UBMA	Examinateur

Année Universitaire : 2020/2021

REMERCIEMENTS

Je remercie « ALLAH » le tout puissant, qui ma donnée la foi, la force et la patience pour aller jusqu'au bout de ce travail

Ma remerciements et reconnaissances va à tous ceux qui m'ont aidé au cours de ce travail à son terme : tout particulièrement, j'aimerais remercier vivement, mon encadreur de mémoire Mme Mohamed Ben Ali Yamina, pour ses conseils, son encadrement, sa présence, son soutien, sa disponibilité, et pour le courage qu'elle m'a donné afin d'accomplir ce projet

Je ne peux achever ce projet sans exprimer mes gratitudes à tous mes enseignants pour leur dévouement et leur assistance tout au cours de mon parcours universitaire

Mes sincères remerciements vont également aux membres du jury pour l'intérêt qu'ils ont accepté d'examiner mon travail et de l'enrichir par leurs propositions

Finalement, je veux remercier toutes les personnes qui m'ont aidé directement ou indirectement à la réalisation de ce travail

DEDICACES

À mes chers parents, Pour qui nous devons ce que nous sommes aujourd'hui. Grâce à vos prières, soutiens et sacrifices tout au long de notre cursus. Que dieu, le tout puissant vous préserve et vous procure une santé et une longue vie

À mes chers frères et sœurs, Aucune dédicace ne serait suffisante pour vous exprimer ce que nous ressentons envers vous. Nous vous dirons juste merci pour vos conseils et vos encouragements, et que nous vous souhaitons une vie pleine de succès et de prospérité

À mes amis(Hamza Zarzouni, Ahmed Zadoud et Akrem Yousfi) , Pour leur affection et leur soutien

TABLE DES MATIERES

Table des matières

Remerciements I
Dédicaces II
Table des matiers III
Table d'illustration VI
Résume VIII
Introduction Générale 1
Problématique 1
Objectifs 2
Contenu du mémoire 2
Chapitre 1 Apprentissage automatique et apprentissage profond 3
I. Apprentissage automatique 3
1- Définition 3
2- Principes 4
3- Types d'apprentissage 5
3.1- Apprentissage supervisé 5
3.2- Apprentissage non supervisé 5
3.3- Apprentissage semi-supervisée 6
3.4- Apprentissage partiellement 6
3.5- Apprentissage par renforcement 6
3.6- Apprentissage par transfert 6
4- Étapes d'apprentissage automatique 6
4.1- Définir le problème à résoudre 6
4.2- Analyser et explorer les données 7
4.3- Ingénierie ou extraction de caractéristiques 7
4.4- Choisir ou construire un modèle d'apprentissage 7
4.5- Entrainer, évaluer et optimiser 7
4.6- Test 7
5- Domain d'applications 7
II. Apprentissage profond 8
1- Définition 8
2- Les types d'apprentissage profond 9

2.1- Réseau de neurones artificiels (ANN) 9
2.2- Réseau de neurones récurrents (RNN) 9
2.3- Réseau neuronal convolutionnel (CNN) 10
2.3.1- Définition 10
2.3.2- Le fonctionnement du CNN 10
2.3.3- Blocs de construction 12
3- Domaines d'application 14
3.1- La reconnaissance visuelle 14
3.2- La robotique 14
3.3- La bio-informatique 14
3.4- La Chemoinformatique 14
Conclusion 14
Chapitre 2 Chemoinformatqiue 15
1. Définition 16
2. Molécule chimique 17
2.1- Définition 17
2.2- La structure 18
2.3- Collage (bonding) 20
3. Propriété physico-chimique de molécule (protéines) 22
4. Représentation des composés chimique 23
5. Les caractéristiques 23
5.1- Ordonnancement 23
5.2- Stabilité 24
5.3- Les macromolécules et polymères 24
6. Données en chimie 25
6.1- Les sources de données et les bases de données 25
6.2- Méthodes d'analyse de données 25
7. Les formats des données chimique 25
8. Exemple d'application 27
Conclusion 27
Chapitre 3 Conception et implémentation 28
1.Introduction 28
2. Conception 28
2.1- Architecture fonctionnelle de l'application 28
2.2- Source de la base 29
2.3- Modélisation des données 29
3. Implémentation 30

3.1- Les outils utilisés 30
3.1.1- Python 30
3.1.2- Anaconda3 30
3.1.3- Jupyter Notebook 31
3.2- Les étapes de l'implémentation 31
3.2.1- Importation des bibliothèques (package) 32
3.2.2- Importation du dataset 34
3.2.3- Prétraitement des données 34
3.2.3.1- Importation et lecture des données 35
3.2.3.2- Eliminer les éléments indésirables 35
3.2.3.2.1- Nettoyage de la base 35
3.2.3.2.2- l'affichage toute les base qui sont active et non active 36
3.2.3.3- Convertir les formats smiles en format image 2D 38
3.2.3.3.1- L'affichage de tous les formats smiles 38
3.2.3.3.2- Crée un dossier (répertoire) 38
3.2.3.3.3- Convertir les smiles et enregistrer les images des molécules 39
3.2.3.4- Séparation des données (Split_folder) 40
3.2.4- Phase d'apprentissage du model 41
3.2.5- la phase test (lancement de l'apprentissage) 43
3.2.6- L'évaluation de model 45
Conclusion 47
Conclusion et Perspectives 48
Références 49

TABLE DES ILLUSTRATIONS

Figure 1 : Apprentissage Automatique 4
Figure 2 : Le classement d'apprentissage profond dans IA 8
Figure 3 : Fonctionnement apprentissage profond 8
Figure 4 : Réseau de neurones convotionnel 10
Figure 5 : le CNN il doit déterminer si les 2 photos représentent X ou O 11
Figure 6 : le CNN compare les 2 images fragment par fragment 11
Figure 7 : la comparaison des caractéristiques extraites 12
Figure 8 : exemple de modele CNN 13
Figure 9 : Atomes liés par des liaisons chimiques 16
Figure 10 : Schéma de la molécule d'eau 17
Figure 11 : Tableau périodique des éléments chimiques 18
Figure 12 : Tableau nombre d'électron et leur groupe géométrique 19
Figure 13 : Structure 2D de molécule chimiques 19
Figure 14 : Structure 3D de molécule chimiques 20
Figure 15 : Montre la liaison covalente 21
Figure 16 : Montre la liaison ionique 21
Figure 17 : La dénaturation des molécules 22
Figure 18 : Architecture de l'application 28
Figure 19 : Source de la base 29
Figure 20 : Modélisation de neurones 29
Figure 21 : lego de langage python 30
Figure 22 : Plateforme d'Anaconda3 30
Figure 23 : Plateforme de jupyter notebook 31
Figure 24 : La base de données formats xlsx 34
Figure 25 : le pseudo code pour lecture de la base 35
Figure 26 : Les étapes de nettoyage 36
Figure 27 : Les molécules qui sont déjà non active 36
Figure 28 : Les molécules qui sont active (IC50 <= 1000) 37
Figure 29 : Les molécules qui sont non active (IC50 > 1000) 37
Figure 30 : les formats smiles des molécules qui sont active 38
Figure 31 : Pseudo code pour la création de dossier 38
Figure 32 : Convertir les smiles et enregistrer les images de molécules actives 39
Figure 33 : Convertir et enregistrer les images de molécules qui sont déjà inactives 39

Figure 34 : Convertir les smiles et enregistrer les images de molécules inactives 40
Figure 35 : pseudo code pour la division de la base.....40
Figure 36 : les différentes couches utilisées pour notre modèle CNN 42
Figure 37 : pseudo code de la méthode compile et ces paramètres 43
Figure 38 : les données d'apprentissage et de validation 43
Figure 39 : l'entrainement du modèle en appelant la fonction fit() 44
Figure 40 : Pseudo code pour l'évaluation final de données train et données validation 45
Figure 41 : La nouvelle base pour la prédiction.....45
Figure 42 : l'évaluation de model sur les données qu'il a jamais vues 46
Figure 43 : Matrice de confusion 47

RESUME

La classification des molécules est l'un des domaines de recherche en chemoinformatique. Le défi principal consiste à développer une méthode performante pour prédire si la molécule est active ou non à partir d'un grand ensemble de données qui peut contenir des données non pertinentes et redondantes. Pour ce faire, la technique l'apprentissage supervisé (classification) est utilisées. Dans ce travail, nous proposons une approche qui contient un classifieur pour Classifier et prédire l'activité ou l'inactivité d'une molécule, l'approche proposée (CNN) est composée de 3 étapes : l'étape d'extraction des caractéristiques d'un molécule, l'étape de classification (d'apprentissage et validation) et en fin l'étape de prédiction La connaissance d'une molécule (dite active ou non active). Les expérimentations ont montré que notre approche est efficace conservant des taux d'erreur de classification très faible est une stabilité très satisfaisante.

Mots clés : [Apprentissage automatique, apprentissage profond, CNN, Chemoinformatique, molécule, format smiles, IC50, RDkit, Active ou non …]

ملخص

يعتبر تصنيف الجزيئات أحد مجالات البحث في المعلوماتية الكيميائية. يتمثل التحدي الرئيسي في تطوير طريقة قوية للتنبؤ بما إذا كان الجزيء نشطًا أم لا من مجموعة بيانات كبيرة قد تحتوي على بيانات غير ذات صلة وزائدة عن الحاجة. للقيام بذلك ، يتم استخدام أسلوب التعلم الخاضع للإشراف (التصنيف). في هذا العمل ، نقترح نهجًا يحتوي على مصنف لتصنيف وتوقع نشاط الجزيء أو عدم نشاطه ، يتكون النهج المقترح من 3 خطوات: خطوة استخراج خصائص الجزيء ، وخطوة التصنيف (التعلم والتحقق من الصحة) وفي النهاية خطوة التنبؤ معرفة الجزيء (يقال أنه نشط أو غير نشط). أظهرت التجارب أن نهجنا فعال ، مع الحفاظ على معدلات خطأ تصنيف منخفضة للغاية واستقرار مرض.

الكلمات الرئيسية: [التعلم الآلي ، التعلم العميق ، المعلوماتية الكيميائية ، الجزيء ، تنسيق الابتسامات ، نشط أم لا ...]

INTRODUCTION GENERALE

Depuis quelques années, la récolte des données en chimie a connu une explosion quantitative grâce notamment au développement de nouveaux moyens techniques servant à comprendre les composants chimiques. Pour analyser ces données, plus nombreuses et plus complexes aussi, les scientifiques se sont tournés vers les nouvelles technologies de l'information. L'immense capacité de stockage et d'analyse des données qu'offre l'informatique leur a permis de gagner en puissance pour leurs recherches. Et la rencontre entre la chimique et l'informatique, c'est ce qu'on appelle la chemoinformatique.

La chemoinformatique sert donc à stocker, traiter et analyser des grandes quantités des données de chimie. Le but est de mieux comprendre et mieux connaître les phénomènes et processus chimiques. Grâce à ces nouvelles connaissances ainsi acquises, les chercheurs ont la possibilité de faire de nouvelles découvertes scientifiques. Des découvertes qui peuvent par exemple améliorer la qualité de vie de personnes malades grâce à la mise en place de nouveaux traitements médicaux plus efficaces.

La chemoinformatique nous aide à visualiser les structures invisibles tels que les molécules et d'en apprendre davantage sur leur travail et leur fonction. Cela conduit à comprendre les questions essentielles de la vie : Comment les organismes fonctionnent-ils ? Comment la vie s'est-elle développée ?

Problématique

Plusieurs méthodes basées sur les images moléculaires ont été récemment introduites pour la classification des petites molécules chimiques. La plupart des méthodes disponibles sur les molécules sont basés sur des représentations 2D obtenues à partir de structures chimiques. Nous introduisons des nouvelles idées pour construire un programme sur les molécules qui peuvent utiliser et combiner efficacement les informations 2D pour faire la classification, la prédiction sur l'inactivité ou l'activité des molécules.

Objectifs

L'objectif principal de notre travail est comment prédire l'activité ou l'inactivité des molécules chimiques à partir des images 2D en utilisant le réseau de neurone convolutif(CNN). La partie des données qui est en format smiles est d'abord transformée en image 2D en utilisant la technique RDkit. Ce travail est basé sur le succès actuel des modèles d'apprentissage profond en traitement d'image.

Contenu du mémoire

Outre la partie introductive et la conclusion générale, le travail est organisé en 3 chapitres :

- Le premier chapitre est consacré à l'apprentissage automatique et profond (machine learning et deep learning) qui sont des champs d'étude de l'intelligence artificielle qui se fonde sur des approche mathématiques et statistiques pour donner aux ordinateurs la capacité d'« apprendre » à partir de données,

- Dans le deuxième chapitre, on passera à une tentative de définir la chemoinformatique nous commençons par la définition pour terminer avec les domaines majeurs d'application.

- Dans Le dernier chapitre nous présentons la conception et l'implémentation en détails, nous validons l'approche mentionnée dans le 1er chapitre et nous montrons l'efficacité de notre travail avec des captures d'écrans et des fragments de code source, et de l'application elle-même en cours d'exécution, en expliquant chaque point.

CHAPITRE 1
APPRENTISSAGE AUTOMATIQUE ET APPRENTISSAGE PROFOND

Plusieurs chercheurs ont étudié les techniques d'apprentissage pour prédire l'activité chimique des molécules. Toutes ces techniques doivent résoudre efficacement deux sous-problèmes distincts : l'un lié à la représentation et l'autre axé sur les problèmes d'apprentissage. Le premier revient essentiellement à concevoir un ensemble approprié qui saisit efficacement les caractéristiques représentatives des molécules idéalement. La seconde consiste à trouver une fonction lisse et stable soit à un nombre réel exprimant l'activité de la molécule (régression), soit à une décision booléenne telle qu'actif/inactif (classification).

Dans ce chapitre nous proposons donc des nouvelles fonctions, qui extrait les caractéristiques et préserver autant d'informations que possible sur la structure moléculaire et les intègre dans un espace euclidien. Ainsi dans la deuxième section nous présentons Les algorithmes d'apprentissage automatique pour la classification binaire sur des données moléculaires selon leur dimension.

I. Apprentissage automatique

1- Définition :

L'apprentissage automatique (en anglais : *machine learning)*, est un champ d'étude de l'intelligence artificielle qui se fonde sur des approches mathématiques et statistiques pour donner aux ordinateurs la capacité d'« apprendre » à partir de données, c'est-à-dire d'améliorer leurs performances à résoudre des tâches sans être explicitement programmés pour chacune. Plus largement, il concerne la conception, l'analyse, l'optimisation, le développement et l'implémentation de telles méthodes.

L'apprentissage automatique comporte généralement deux phases. La première consiste à estimer un modèle à partir de données, appelées observations, qui sont disponibles et en nombre fini, lors de la phase de conception du système. L'estimation du modèle consiste à résoudre une

tâche pratique, telle que traduire un discours, estimer une densité de probabilité, reconnaître la présence d'un chat dans une photographie ou participer à la conduite d'un véhicule autonome. Cette phase dite « d'apprentissage » ou « d'entraînement » est généralement réalisée préalablement à l'utilisation pratique du modèle. La seconde phase correspond à la mise en production : le modèle étant déterminé, de nouvelles données peuvent alors être soumises afin d'obtenir le résultat correspondant à la tâche souhaitée.

En pratique, certains systèmes peuvent poursuivre leur apprentissage une fois en production, pour peu qu'ils aient un moyen d'obtenir un retour sur la qualité des résultats produits. [Web 1]

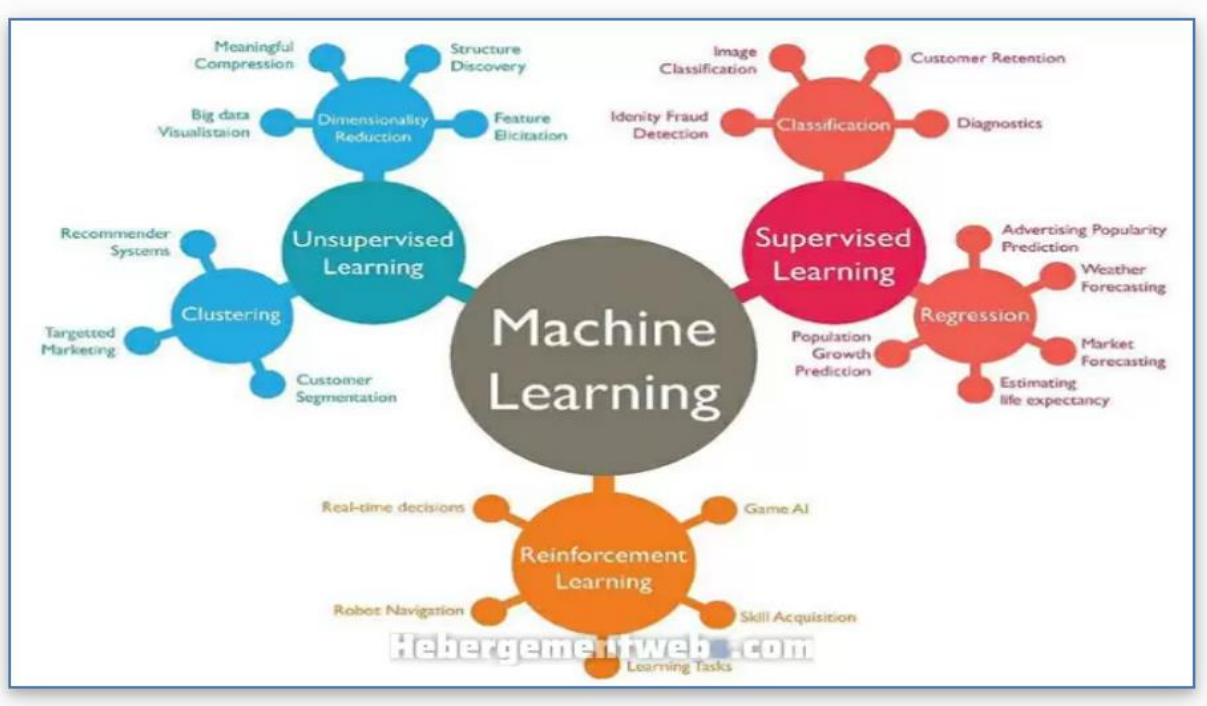


Figure 1 : Apprentissage Automatique. **[Web 2]**

2- Principes :

L'apprentissage automatique (AA) permet à un système piloté ou assisté par ordinateur comme un programme, une IA ou un robot, d'adapter ses réponses ou comportements aux situations rencontrées, en se fondant sur l'analyse de données empiriques passées issues de bases de données, de capteurs, ou du web.

L'AA permet de surmonter la difficulté qui réside dans le fait que l'ensemble de tous les comportements possibles compte tenu de toutes les entrées possibles devient rapidement trop complexe à décrire et programmer de manière classique (on parle d'explosion combinatoire). On confie donc à des programmes d'AA le soin d'ajuster un modèle pour simplifier cette complexité et de l'utiliser de manière opérationnelle. Idéalement, l'apprentissage visera à

être non supervisé, c'est-à-dire que les réponses aux données d'entraînement ne sont pas fournies au modèle.

Ces programmes, selon leur degré de perfectionnement, intègrent éventuellement des capacités de traitement probabiliste des données, d'analyse de données issues de capteurs, de reconnaissance (reconnaissance vocale, de forme, d'écriture…), de fouille de données, d'informatique théorique… [Web 1]

3- Types d'apprentissage :

Les algorithmes d'apprentissage peuvent se catégoriser selon le mode d'apprentissage qu'ils emploient.

3.1- Apprentissage supervisé :

Si les classes sont prédéterminées et les exemples connus, le système apprend à classer selon un modèle de classification ou de classement ; on parle alors d'apprentissage supervisé (ou d'analyse discriminante). Un expert (ou oracle) doit préalablement étiqueter des exemples. Le processus se passe en deux phases. Lors de la première phase (hors ligne, dite d'apprentissage), il s'agit de déterminer un modèle à partir des données étiquetées. La seconde phase (en ligne, dite de test) consiste à prédire l'étiquette d'une nouvelle donnée, connaissant le modèle préalablement appris. Parfois il est préférable d'associer une donnée non pas à une classe unique, mais une probabilité d'appartenance à chacune des classes prédéterminées (on parle alors d'apprentissage supervisé probabiliste) (ex des algorithmes utiliser : SVM, Régression linaire, Régression logistique, Les arbres de décision, Réseau de neurones artificiels). [1]

3.2- Apprentissage non supervisé :

Quand le système ou l'opérateur ne dispose que d'exemples, mais non d'étiquette, et que le nombre de classes et leur nature n'ont pas été prédéterminées, on parle d'apprentissage non supervisé ou clustering en anglais. Aucun expert n'est requis. L'algorithme doit découvrir par lui-même la structure plus ou moins cachée des données. Le partitionnement de données, data clustering en anglais, est un algorithme d'apprentissage non supervisé.

Le système doit ici — dans l'espace de description (l'ensemble des données) — cibler les données selon leurs attributs disponibles, pour les classer en groupes homogènes d'exemples. La similarité est généralement calculée selon une fonction de distance entre paires d'exemples. C'est ensuite à l'opérateur d'associer ou déduire du sens pour chaque groupe et pour les motifs (patterns en anglais) d'apparition de groupes, ou de groupes de groupes, dans leur « espace ».

Divers outils mathématiques et logiciels peuvent l'aider (ex des algorithmes utiliser : K-means, clastring Réseaux de neurones/Deep learning, algorithme génétique). [1]

3.3- Apprentissage semi-supervisé :

Effectué de manière probabiliste ou non, il vise à faire apparaître la distribution sous-jacente des exemples dans leur espace de description. Il est mis en œuvre quand des données (ou « étiquettes ») manquent… Le modèle doit utiliser des exemples non étiquetés pouvant néanmoins renseigner. [1]

3.4- Apprentissage partiellement supervisé :

Probabiliste ou non, quand l'étiquetage des données est partiel20. C'est le cas quand un modèle énonce qu'une donnée n'appartient pas à une classe A, mais peut-être à une classe B ou C (A, B et C étant trois maladies par exemple évoquées dans le cadre d'un diagnostic différentiel). [1]

3.5- Apprentissage par renforcement :

L'algorithme apprend un comportement étant donné une observation. L'action de l'algorithme sur l'environnement produit une valeur de retour qui guide l'algorithme d'apprentissage. ex. : L'algorithme de Q-learning est un exemple classique. [1]

3.6- Apprentissage par transfert :

L'apprentissage par transfert (transfer learning, en anglais) consiste à appliquer des connaissances obtenues en effectuant une tâche afin de résoudre un problème différent, mais qui présente des similitudes.

L'apprentissage par transfert vise à utiliser les connaissances d'un jeu de tâches sources pour non seulement influencer l'apprentissage mais aussi améliorer les performances sur une autre tâche cible. Il consiste en quelque sorte à utiliser les connaissances acquises pour les ré-appliquer dans un autre environnement. Pour être efficace l'environnement cible ne doit pas être trop différent de celui des tâches sources, sinon des transferts négatifs seront réalisés menant au contraire du résultat recherché. [Web 2]

4- Étapes d'apprentissage automatique :

L'apprentissage automatique ne se résume pas à un ensemble d'algorithmes, mais suit une succession d'étapes.

4.1- Définir le problème à résoudre :

Acquérir des données : l'algorithme se nourrissant des données en entrée, c'est une étape importante. Il en va de la réussite du projet, de récolter des données pertinentes et en quantité et qualité suffisantes, et en évitant tout biais dans leur représentativité**.** [Web 1]

4.2- Analyser et explorer les données :

Préparer et nettoyer les données : les données recueillies doivent être retouchées avant utilisation. En effet, certains attributs sont inutiles, d'autre doivent être modifiés afin d'être compris par l'algorithme, et certains éléments sont inutilisables car leurs données sont incomplètes. Plusieurs techniques telles que la visualisation de données, la transformation de données (en) ou encore la normalisation sont alors employées. [Web 1]

4.3- Ingénierie ou extraction de caractéristiques :

Les attributs peuvent être combinés entre eux pour en créer de nouveaux plus pertinents et efficaces pour l'entrainement du modèle. [Web 1]

4.4- Choisir ou construire un modèle d'apprentissage :

Un large choix d'algorithmes existe, et il faut en choisir un adapté au problème et aux données**.**

4.5- Entrainer, évaluer et optimiser :

L'algorithme d'apprentissage automatique est entraîné et validé sur un premier jeu de données pour optimiser ses hyper-paramètres. [Web 1]

4.6- Test :

Puis il est évalué sur un deuxième ensemble de données de test afin de vérifier qu'il est efficace avec un jeu de donnée indépendant des données d'entrainement, et pour vérifier qu'il ne fasse pas de sur apprentissage**.** [Web 1]

5- Domain d'applications :

L'apprentissage automatique est utilisé dans un large spectre d'applications pour doter des ordinateurs ou des machines de capacité d'analyser des données d'entrée comme : perception de leur environnement (vision, Reconnaissance de formes tels des visages, schémas, segmentation d'image, langages naturels, caractères dactylographiés ou manuscrits ; moteurs de recherche, analyse et indexation d'images et de vidéo, en particulier pour la recherche d'image par le contenu ; aide aux diagnostics, médical notamment, bio-informatique, chemoinformatique ; interfaces cerveau-machine ; détection de fraudes à la carte de crédit, cybersécurité, analyse financière, dont analyse du marché boursier ; classification des séquences d'ADN ; jeu ; génie logiciel ; adaptation de sites Web ; robotique (locomotion de robots, etc.) ; analyse prédictive dans de nombreux domaines (financière, médicale, juridique, judiciaire). [Web 1]

II. Apprentissage profond

L'apprentissage profond (ou *Deep Learning*) est un sous-domaine particulièrement puissant du Machine Learning. L'apprentissage profond ou apprentissage en profondeur (en anglais : deep learning) est un ensemble de méthodes d'apprentissage automatique tentant de modéliser avec un haut niveau d'abstraction des données grâce à des architectures articulées de différentes transformations non linéaires. Ces techniques ont permis des progrès importants et rapides dans les domaines de l'analyse du signal sonore ou visuel et notamment de la reconnaissance faciale, de la reconnaissance vocale, de la vision par ordinateur, du traitement automatisé du langage.

[Web 3]

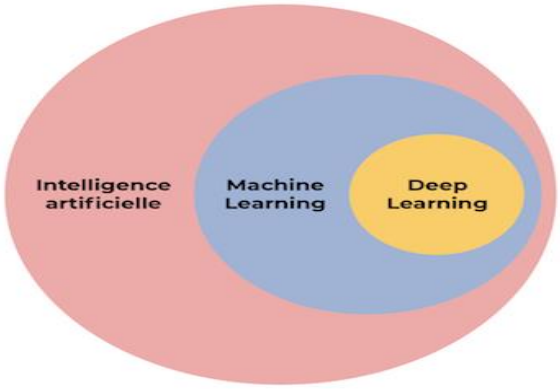


Figure 2 : Le classement d'apprentissage profond dans IA. [Web 4]

1- Définition :

Le deep learning ou apprentissage profond est un sous-domaine de l'intelligence artificielle (IA). Ce terme désigne l'ensemble des techniques d'apprentissage automatique (machine learning), autrement dit une forme d'apprentissage fondée sur des approches mathématiques, utilisées pour modéliser des données. [Web 5]

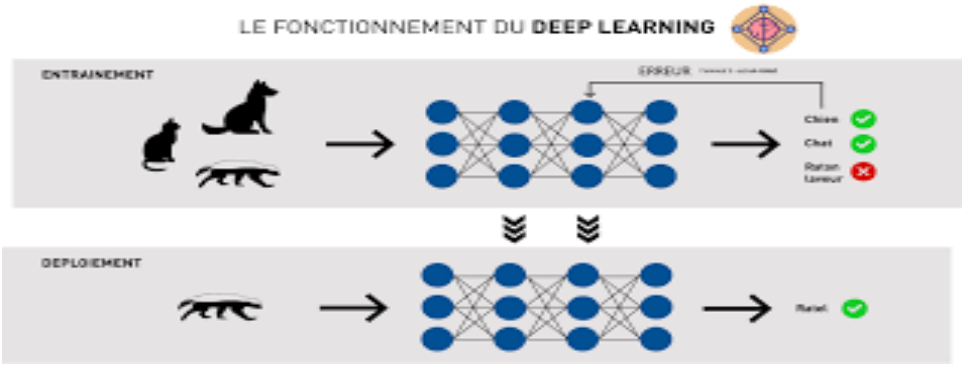


Figure 3 : Fonctionnement apprentissage profond. [Web 6]

Les techniques d'apprentissage profond constituent une classe d'algorithmes d'apprentissage automatique qui :

- Utilisent différentes couches d'unité de traitement non linéaire pour l'extraction et la transformation des caractéristiques ; chaque couche prend en entrée la sortie de la précédente ; les algorithmes peuvent être supervisés ou non supervisés, et leurs applications comprennent la reconnaissance de modèles et la classification statistique ;
- Fonctionnent avec un apprentissage à plusieurs niveaux de détail ou de représentation des données ; à travers les différentes couches, on passe de paramètres de bas niveau à des paramètres de plus haut niveau, où les différents niveaux correspondent à différents niveaux d'abstraction des données.

  Ces architectures permettent aujourd'hui de conférer du « sens » à des données en leur donnant la forme d'images, de sons ou de textes. [Web 5]

2- Les types d'apprentissage profond :

2.1- Réseau de neurones artificiels (ANN) :

Un réseau de neurones artificiels, ou réseau neuronal artificiel, est un système dont la conception est à l'origine schématiquement inspirée du fonctionnement des neurones biologiques, et qui par la suite s'est rapproché des méthodes statistiques.

Les réseaux de neurones sont généralement optimisés par des méthodes d'apprentissage de type probabiliste, en particulier bayésien. Ils sont placés d'une part dans la famille des applications statistiques, qu'ils enrichissent avec un ensemble de paradigmes4 permettant de créer des classifications rapides (réseaux de Kohonen en particulier), et d'autre part dans la famille des méthodes de l'intelligence artificielle auxquelles ils fournissent un mécanisme perceptif indépendant des idées propres de l'implémenter, et des informations d'entrée au raisonnement logique formel (voir Deep Learning). [Web 7]

2.2- Réseau de neurones récurrents(RNN) :

Un réseau de neurones récurrents est un réseau de neurones artificiels présentant des connexions récurrentes. Un réseau de neurones récurrents est constitué d'unités (neurones) interconnectées interagissant non-linéairement et pour lequel il existe au moins un cycle dans la structure. Les unités sont reliées par des arcs (synapses) qui possèdent un poids. La sortie d'un neurone est une combinaison non linéaire de ses entrées.

Les réseaux de neurones récurrents sont adaptés pour des données d'entrée de taille variable. Ils conviennent en particulier pour l'analyse de séries temporelles. Ils sont utilisés en reconnaissance automatique de la parole ou de l'écriture manuscrite - plus en général en reconnaissance de formes - ou encore en traduction automatique. [Web 8]

2.3- Réseau neuronal convolutionnel (CNN) :

2.3.1- Définition :

Dans l'apprentissage profond, un réseau de neurones convolutifs (CNN / ConvNet) est une classe de réseaux de neurones profonds, le plus couramment appliqués pour analyser l'imagerie visuelle. Maintenant, quand on pense à un réseau de neurones, on pense aux multiplications matricielles mais ce n'est pas le cas avec ConvNet. Il utilise une technique spéciale appelée Convolution. Or, en mathématiques, la convolution est une opération mathématique sur deux fonctions qui produit une troisième fonction qui exprime comment la forme de l'une est modifiée par l'autre. [Web 9]

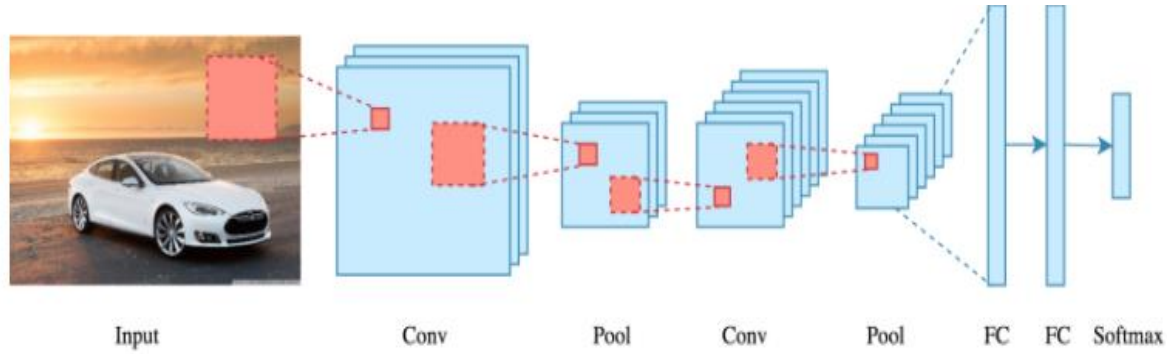


Figure 4 : Réseau de neurones convotionnel. [Web 9]

En fin de compte, le rôle du ConvNet est de réduire les images sous une forme plus facile à traiter, sans perdre les fonctionnalités essentielles pour obtenir une bonne prédiction.

2.3.2- Le fonctionnement du CNN :

Pour nous aider à comprendre le fonctionnement des réseaux de neurones convolutifs, nous utiliserons un exemple simplifié et tenterons de déterminer si l'image représente X ou O. Dans cette partie le CNN n'a qu'une tâche : chaque fois que nous lui montrons une photo, il doit déterminer si la photo représente X ou O. Il pense que dans chaque situation, il ne peut y en avoir qu'une. [3]

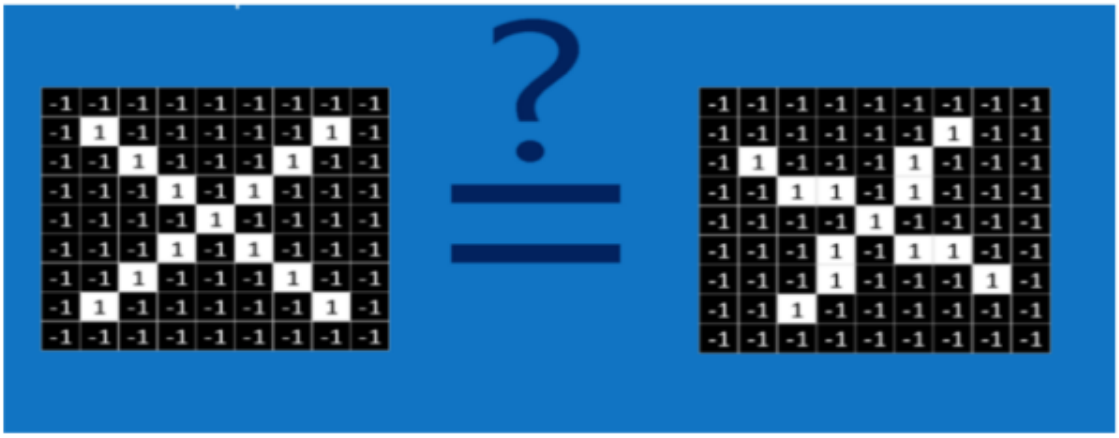


Figure 5 : le CNN il doit déterminer si les 2 photos représentent X ou O. [3]

La manière naïve de résoudre de tels problèmes est de sauvegarder l'image représentant X et l'image représentant O, et de comparer chaque nouvelle image avec ces deux images pour voir l'image la plus proche c'est-à-dire l'image la plus similaire. Pour un ordinateur, une image n'est rien de plus qu'un tableau bidimensionnel de pixels ce qui rend cette tâche compliquée, chaque case du tableau contient un nombre spécifique : dans cet exemple, un pixel avec une valeur de 1 est un pixel blanc, et avec une valeur - 1 est un pixel noir. Lors de la comparaison directe de deux images, si même un pixel contient des valeurs différentes, l'ordinateur pensera que ces images sont différentes. [3]

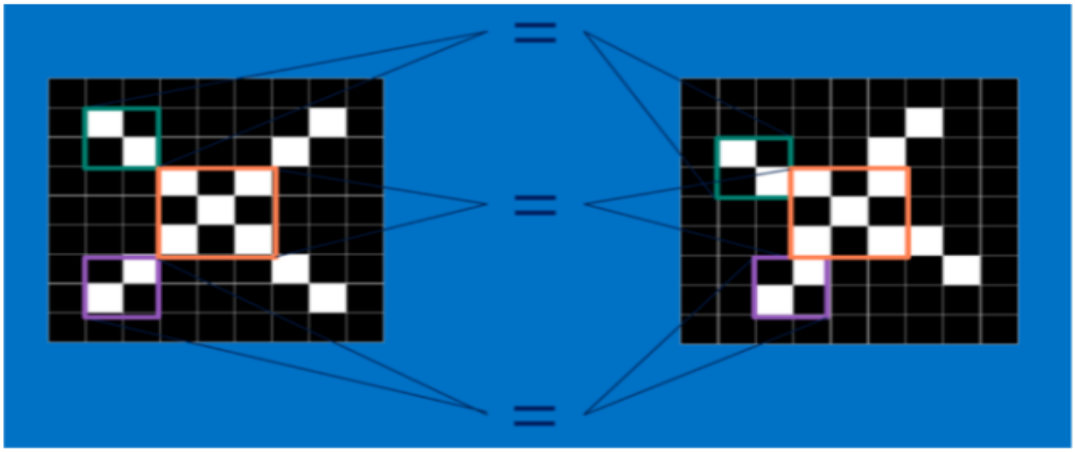


Figure 6 : le CNN compare les 2 images fragment par fragment. [3]

Le CNN commence de comparer les images fragment par fragment. Les fragments recherchés sont appelés des caractéristiques. En trouvant des caractéristiques qui sont similaire à peu près dans les 2 images différentes

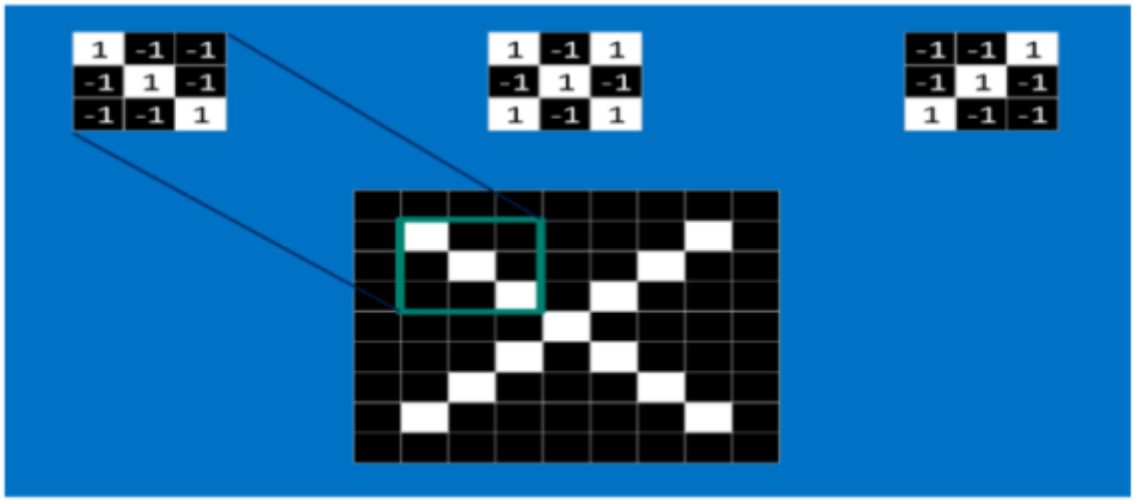


Figure 7 : la comparaison des caractéristiques extraites. [3]

Chaque caractéristique est considérée comme un petit tableau bidimensionnel. Les caractéristiques affichent les attributs les plus communs des images. Dans le cas de l'image qui représente un X, les caractéristiques : les deux diagonales et l'entrecroisement de ces dernières (correspondent aux bras et au centre d'un X) représentent les trais les plus courants de X. [3]

2.3.3- Blocs de construction

Une architecture de réseau de neurones convolutifs est formée par un empilement de couches de traitement :

- La couche de convolution (CONV) qui traite les données d'un champ récepteur ;
- La couche de pooling (POOL), qui permet de compresser l'information en réduisant la taille de l'image intermédiaire (souvent par sous-échantillonnage) ;
- la couche de correction (ReLU), souvent appelée par abus « ReLU » en référence à la fonction d'activation (Unité de rectification linéaire) ;
- La couche « entièrement connectée » (FC), qui est une couche de type perceptron ;
- La couche de perte (LOSS). [Web 10]

➢ **Couche de Convolutifs :**

Dans cette couche pour que puisse détecter des caractéristiques de l'image il s'agit d'appliquer un filtre de convolution à l'image. L'image passe par une série de filtres ou de noyaux de convolution, pour créer une image appelée carte de convolution. Enfin, les cartes de convolution sont concaténées en un vecteur de caractéristiques. [Web 10]

- **Couche de pooling (pool) :**

  Cela comprend la réduction progressive de la taille de l'image en ne conservant que les informations les plus importantes (par exemple, pour chaque groupe de 4 pixels, ce pixel a la valeur maximale) donc on prend la valeur maximal de ce groupe (Max Pooling, le plus populaire). [Web 10]

- **Couche de correction ReLU :**

  Consiste à passer les cartes de convolutions à travers une couche d'activation non linéaire comme Rectified Linear Unit (ReLu), son rôle est de remplacer les nombres négatifs des images filtrées par des zéros. [Web 10]

- **Couche entièrement connectée (Fully connected) :**

  Il reçoit un vecteur d'entrée qui contient les pixels aplatis de toutes les images qui ont été filtrées, corrigées et réduites par combinaison. [Web 10]

- **Couche de perte (LOSS) :**

  La couche de perte spécifie comment l'entrainement du réseau pénalise l'écart entre le signal prévu et réel. Elle est normalement la dernière couche dans le réseau. La perte « Softmax »est utilisée pour prédire une seule classe parmi K classes mutuellement exclusives. La perte par entropie croisée sigmoïde est utilisée pour prédire K valeurs de probabilité indépendante dans $[0,1]$. La perte euclidienne est utilisée pour régresser vers des valeurs réelles dans $[-\infty,\infty]$. [Web 10]

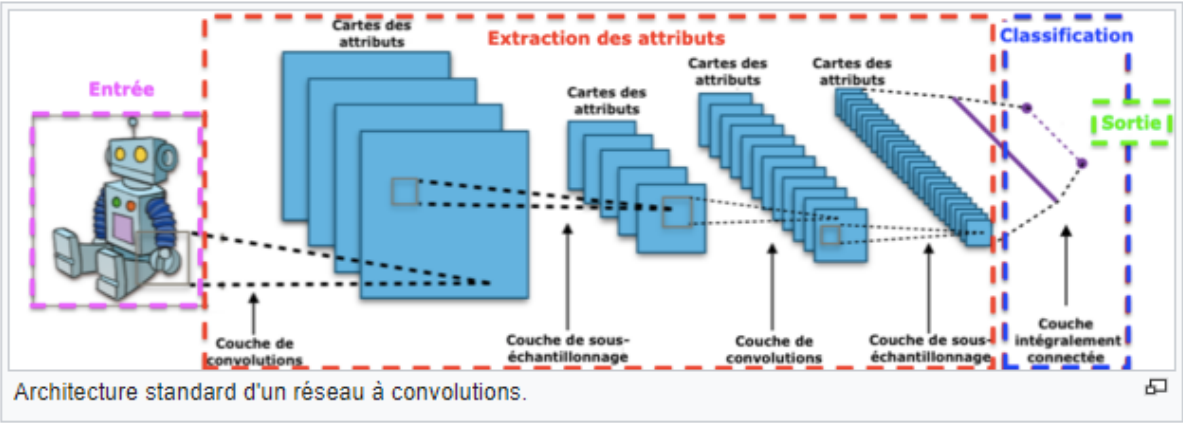


Figure 8 : exemple de modele CNN. [Web 11]

3- Domaines d'application :

L'apprentissage profond s'applique à divers secteurs, notamment **:**

3.1- La reconnaissance visuelle :

La reconnaissance faciale est une technologie de plus en plus répandue, basée sur l'intelligence artificielle, permettant d'identifier une personne sur une photo ou une vidéo en comparant son visage avec ceux sauvegardés dans une base de données. [Web 12]

3.2- La robotique :

La robotique est l'ensemble des techniques permettant la conception et la réalisation de machines automatiques ou de robots. [Web 13]

3.3- La bio-informatique :

La bio-informatique est une science à l'interface des disciplines numériques (l'informatique et les mathématiques) et des sciences de la vie (biochimie, biologie, microbiologie, écologie, épidémiologie). Étant donné que les scientifiques de la vie génèrent une quantité croissante de nouvelles données portant sur les génomes, les biomolécules, les organismes, leurs interactions et leur évolution, il y a un besoin croissant d'approches informatiques pour la manipulation, le stockage, la visualisation et l'analyse de ces données souvent très complexes. [Web 14]

3.4- La Chemoinformatique :

La chemoinformatique (anglicisme) ou chimio-informatique, est le domaine de la science qui consiste en l'application de l'informatique aux problèmes relatifs à la chimie. Elle a pour objectif de fournir des outils et des méthodes pour l'analyse et le traitement des données issues des différents domaines de la chimie.

Elle est notamment utilisée en pharmacologie pour la découverte de nouvelles molécules actives et la prédiction de propriétés à partir de structures moléculaires. [Web 15]

Conclusion

Dans ce chapitre, nous avons passé en revue quelques méthodes et concepts liés au traitement du donnée (image) et plus précisément la classification. Ainsi nous avons listé de manière non-exhaustive les différentes sortes de classification. Ceci dit il existe plusieurs méthodes différentes de classification qui permettent de traiter et de bien comprendre les images.

CHAPITRE 2
CHEMOINFORMATIQUE

Chemoinformatique est une discipline scientifique qui a évolué dans les 10 dernières années à l'interface entre la chimie et l'informatique. Il a été constaté que, dans de nombreux domaines de la chimie, l'énorme quantité de données et d'informations produites par la recherche en chimie ne peut être traitée et analysée que par les méthodes assistées par ordinateur.

Ainsi, les méthodes ont été développées pour la construction des bases de données sur les composés chimiques et leurs réactions, pour la prédiction des propriétés physiques, chimiques et biologiques des composés et des matériaux à l'échelle de l'atome et de la molécule, dans tous les secteurs de l'activité humaine, la conception des médicaments, pour élucider la structure, pour la prévision des réactions chimiques et de la conception de synthèse organique.

La recherche et le développement sont essentiels en chemoinformatique d'une part pour accroître notre compréhension des phénomènes chimiques et d'autre part pour que l'industrie reste compétitive dans une économie mondiale.

C'est une branche de la chimie et/ou de la physico-chimie qui utilise les lois de la chimie théorique exploitées dans des codes informatiques spécifiques afin de calculer structures et propriétés d'objets chimiques (molécules, solides, clusters, surfaces ou autres), en appliquant autant que possible ces programmes à des problèmes chimiques réels.

Dans ce chapitre nous présenterons la Chemoinformatique en détaillait sa définition, la molécule chimique, sa structure …etc.

1. Définition

La chemoinformatique (anglicisme) ou chimio-informatique, est le domaine de la science qui consiste en l'application de l'informatique aux problèmes relatifs à la chimie. Elle a pour objectif de fournir des outils et des méthodes pour l'analyse et le traitement des données issues des différents domaines de la chimie.

Elle est notamment utilisée en pharmacologie pour la découverte de nouvelles molécules actives et la prédiction de propriétés à partir de structures moléculaires.

Certaines applications de la chemoinformatique reposent sur les équations de la physique quantique. Elles permettent ainsi de modéliser les conformations des molécules.

Cette très belle image qui représente des atomes liés par des liaisons chimiques n'est qu'une des innombrables possibilités de la chemoinformatique [2]

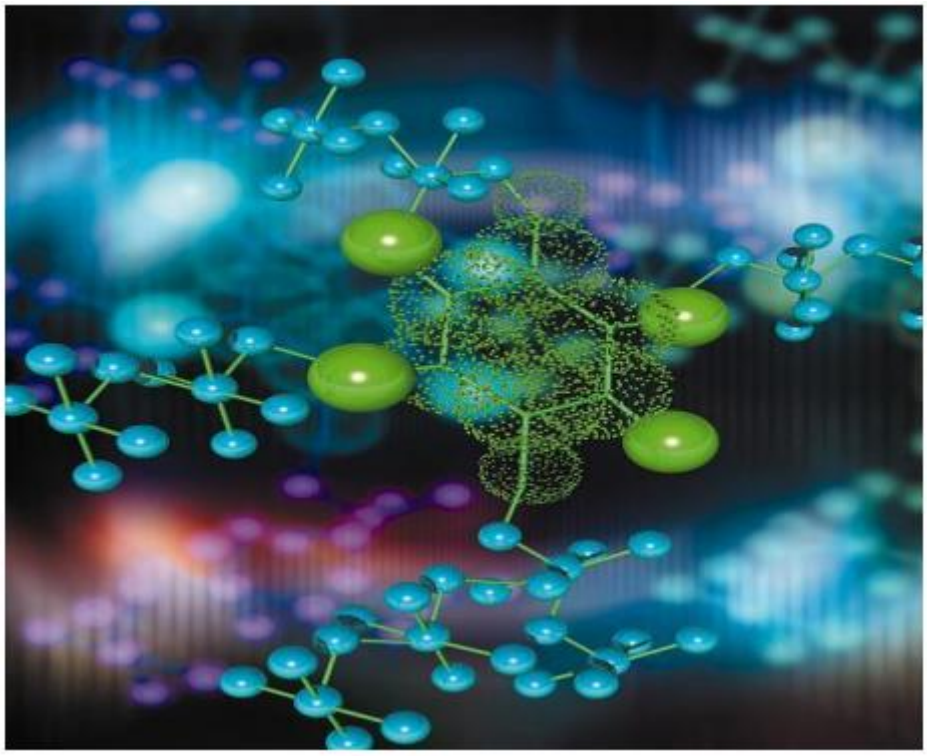


Figure 9 : Atomes liés par des liaisons chimiques. [2]

La recherche et le développement en chemoinformatique sont essentiels pour :

- L'amélioration de la compréhension des phénomènes chimiques.
- Permettre à l'industrie chimique de rester compétitive dans le marché mondial.

2. Molécule chimique

2.1- Définition

Une molécule est un groupe électriquement neutre de deux atomes ou plus maintenus ensemble par des liaisons chimiques. Les molécules se distinguent des ions par leur manque de charge électrique.

Une molécule peut être homonucléaire, c'est-à-dire qu'elle se compose d'atomes d'un élément chimique, comme avec deux atomes dans la molécule d'oxygène.

Une molécule est une structure de base de la matière appartenant à la famille des composés covalents.

Exemple : L'eau : trois atomes, deux éléments, deux liaisons, une molécule. Un atome d'oxygène (ici en rouge), se lie à deux atomes d'hydrogène (ici en blanc). [2]

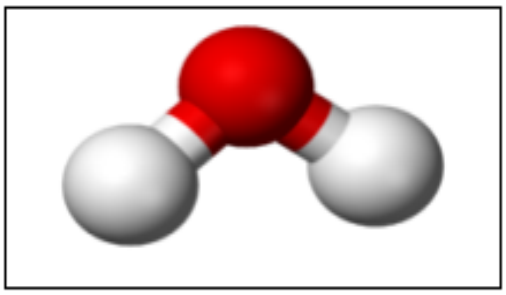


Figure 10 : Schéma de la molécule d'eau. [2]

L'atome : Un **atome** (du grec ατομος, atomos, " que l'on ne peut diviser ") est la plus petite partie d'un corps simple pouvant se combiner chimiquement avec une autre. En chimie, les atomes sont les éléments de base. Ils constituent la matière et forment les molécules en partageant des électrons. Les atomes restent grosso modo indivisibles au cours d'une réaction chimique (en acceptant les légères exceptions que constituent les échanges des électrons périphériques). [Web 16]

L'ion : L'ion est une espèce chimique chargée électriquement, un atome ou une molécule ayant gagné ou perdu un ou plusieurs électrons. [Web 17]

L'élément chimique : On appelle élément chimique l'ensemble des entités - atomes ou ions - qui présentent le même nombre Z de protons dans leur noyau. Nous connaissons aujourd'hui 118 éléments chimiques différents dont 94 existent à l'état naturel sur Terre.

Ainsi, par exemple, le cuivre (Cu) et l'ion cuivre II (Cu^{2+}) sont deux entités de l'élément cuivre. Elles partagent les mêmes propriétés chimiques. [Web 18]

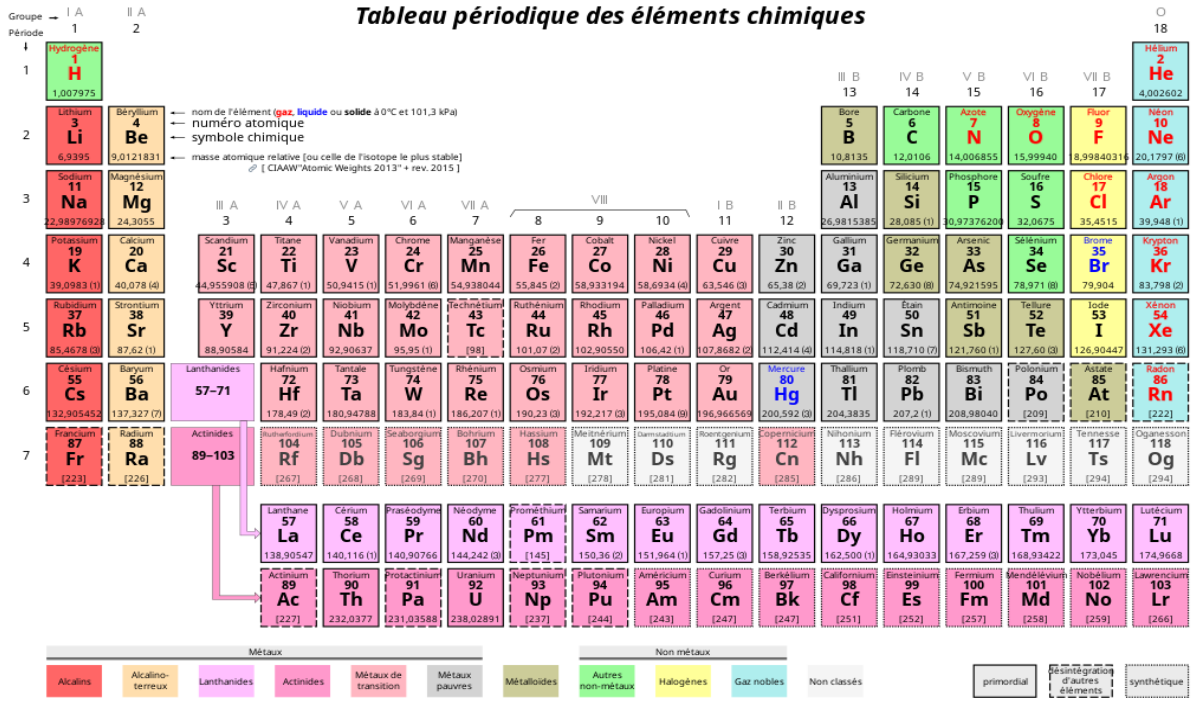


Figure 11 : Tableau périodique des éléments chimiques. [Web 19]

La liaison chimique : est le phénomène qui lie les atomes entre eux en échangeant ou partageant un ou plusieurs électrons ou par des forces électrostatiques.

2.2- La structure

La formule structurelle d'un composé chimique est une représentation graphique de la structure moléculaire (déterminée par des méthodes de chimie structurale), montrant comment les atomes sont éventuellement disposés dans l'espace tridimensionnel réel. La liaison chimique au sein de la molécule est également indiquée, soit explicitement, soit implicitement.

La forme d'une molécule est déterminée par l'emplacement des noyaux et de ses électrons. Les électrons et les noyaux s'installent dans des positions qui minimisent la répulsion et maximisent l'attraction.

Ainsi, la forme de la molécule reflète son état d'équilibre dans lequel elle a l'énergie la plus faible possible dans le système. Bien que la théorie **valence-shell electron-pair repulsion** (VSEPR) prédit la distribution des électrons, nous devons prendre en considération le déterminant réel de la forme moléculaire. Nous séparons cela en deux catégories, la géométrie du groupe d'électrons et la géométrie moléculaire. [Web 20]

La géométrie des groupes d'électrons est déterminée par le nombre de groupes d'électrons.

Number of electron groups	Name of electron group geometry
2	linear
3	trigonal-planar
4	tetrahedral
5	trigonal-bipyramidal
6	octahedral

Figure 12 : Tableau nombre d'électron et leur groupe géométrique. [Web 20]

Différents termes sont utilisés pour désigner les représentations graphiques de molécules :

On parle de représentation de Cram ou de projection de Fischer.

- **Les projections de Fischer** de molécules ne les représentent pas *directement* : les molécules sont projetées et aplaties sur **deux dimensions** (une feuille) de différentes manières selon la projection employée. Elles permettent de représenter indirectement des parties de molécules telles qu'elles existent dans l'espace en appliquant des règles strictes de projection. [Web 21]

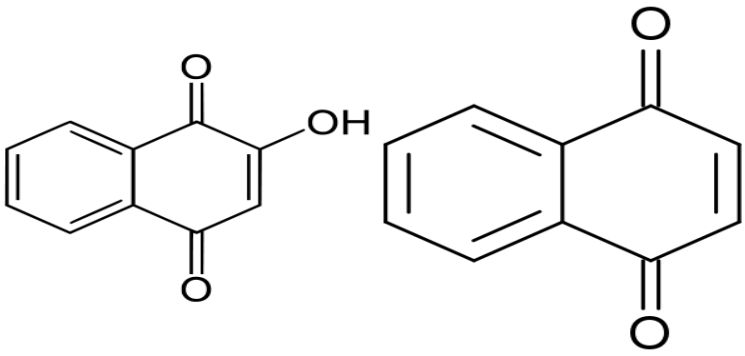


Figure 13 : Structure 2D de molécule chimiques. [Web 22]

- **La représentation de Cram** permet de décrire directement la structure **tridimensionnelle** d'une molécule, par un schéma qui permet de visualiser la molécule telle qu'elle existe dans l'espace. [Web 21]

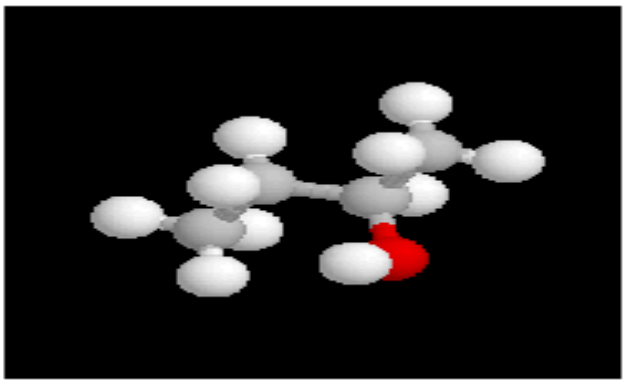


Figure 14 : Structure 3D de molécule chimiques. [Web 23]

➢ **Types de formules chimiques**

La formule chimique d'une molécule utilise une ligne de symboles d'éléments chimiques, de nombres et parfois également d'autres symboles, tels que des parenthèses, des tirets, des crochets et des signes plus (+) et moins (-). Ceux-ci sont limités à une ligne typographique de symboles, qui peut inclure des indices et des exposants. [Web 24]

- **La formule reflète** le nombre exact d'atomes qui composent la molécule et caractérise ainsi différentes molécules. Cependant, différents isomères peuvent avoir la même composition atomique tout en étant des molécules différentes. [Web 24]
- **La formule empirique** est souvent la même que la formule moléculaire mais pas toujours. Par exemple, la molécule d'acétylène a la formule moléculaire C2H2, mais le rapport entier le plus simple des éléments est CH. [Web 24]

2.3- Collage (bonding)

Les molécules sont maintenues ensemble par une liaison covalente ou une liaison ionique. Plusieurs types d'éléments non métalliques n'existent que sous forme de molécules dans l'environnement. Par exemple, l'hydrogène n'existe que sous forme de molécule d'hydrogène. Une molécule d'un composé est constituée d'au moins deux éléments. Une molécule homonucléaire est composée d'au moins deux atomes d'un seul élément. [Web 24]

- **Covalent**

Une liaison covalente formant H2 (à droite) où deux atomes d'hydrogène partagent les deux électrons.

Une liaison covalente est une liaison chimique qui implique le partage de paires d'électrons entre les atomes. Ces paires d'électrons sont appelées paires partagées ou paires de liaisons, et l'équilibre stable des forces attractives et répulsives entre les atomes, lorsqu'ils partagent des électrons, est appelé liaison covalente. [Web 24]

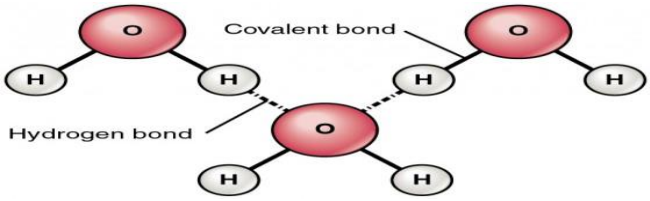


Figure 15 : Montre la liaison covalente. [Web 25]

- **Ionique**

Le sodium et le fluor subissent une réaction redox pour former du fluorure de sodium. Le sodium perd son électron externe pour lui donner une configuration électronique stable, et cet électron pénètre dans l'atome de fluor de manière exothermique.

La liaison ionique est un type de liaison chimique qui implique l'attraction électrostatique entre des ions de charge opposée, et est la principale interaction se produisant dans les composés ioniques. Les ions sont des atomes qui ont perdu un ou plusieurs électrons (appelés cations) et des atomes qui ont gagné un ou plusieurs électrons (appelés anions). Ce transfert d'électrons est appelé électrovalence par opposition à la covalence. Dans le cas le plus simple, le cation est un atome métallique et l'anion est un atome non métallique, mais ces ions peuvent être de nature plus compliquée, par ex. ions moléculaires comme NH4 + ou SO42−. [Web 24]

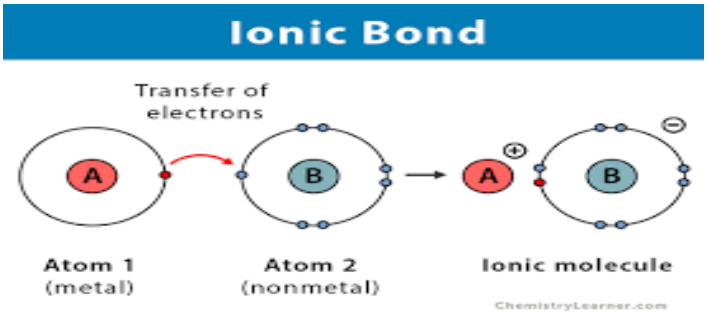


Figure 16 : Montre la liaison ionique. [Web 26]

3. Propriété physico-chimique de molécule (protéines)

Dénaturation

Une protéine est dénaturée lorsque sa conformation tridimensionnelle spécifique est changée par rupture de certaines liaisons sans atteinte de sa structure primaire. Il peut s'agir, par exemple, de la désorganisation de zones en hélice α. La dénaturation peut être réversible ou irréversible. Elle entraîne une perte totale ou partielle de l'activité biologique. Elle produit très souvent un changement de solubilité de la protéine. [2]

Les agents de dénaturation sont nombreux :

- **Agents physiques :** chaleur, radiations, pH ;
- **Agents chimiques** : solution d'urée qui forme de nouvelles liaisons hydrogène dans la protéine, solvants organiques, détergents... [2]

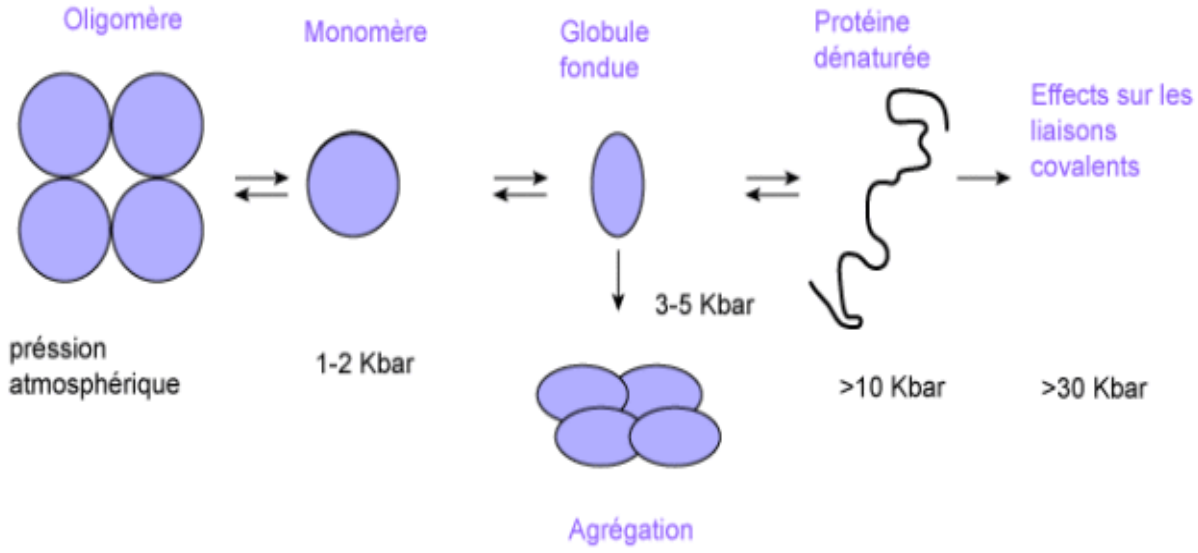


Figure 17 : La dénaturation des molécules. [Web 27]

4. Représentation des composés chimique

Une gamme entière des méthodes pour la représentation sur l'ordinateur des composés et des structures de produit chimique a été développée comprenant des codes linéaires, des tables de raccordement, et des matrices. Des méthodes spéciales ont dû être conçues pour représenter uniquement une structure chimique, pour percevoir des dispositifs tels que l'aromaticité et pour traiter la stéréochimie, les structures 3D, ou les surfaces moléculaires. [2]

5. Les caractéristiques

5.1- Ordonnancement

Les molécules d'un corps sont en agitation permanente (sauf au zéro absolu). Cette agitation, appelée mouvement brownien, a été décrite la première fois par Robert Brown en 1821 dans les liquides (mais expliquée presque 100 ans plus tard).

À l'état gazeux, les molécules sont très espacées, très agitées, avec des mouvements désordonnés provoqués par les chocs entre elles ou avec les parois (les corps solides avec lesquels elles sont en contact).

À l'état liquide, l'espace entre les molécules est beaucoup plus restreint, et l'agitation beaucoup plus lente.

À l'état solide, les molécules sont rangées selon un empilement, régulier ou non, et vibrent autour d'une position moyenne.

La température d'un corps donne une indication du degré d'agitation des molécules.

Les forces d'interaction de très faible intensité qui s'exercent à distance entre les molécules, appelées forces de van der Waals, conditionnent ces arrangements et par conséquent les propriétés physiques des composés moléculaires.

Ainsi, par exemple, les propriétés physiques exceptionnelles de l'eau sont dues pour beaucoup aux liaisons hydrogène. [Web 28]

5.2- Stabilité

Les molécules sont des ensembles a priori électriquement neutres, dans lesquels les atomes sont liés entre eux majoritairement par des liaisons covalentes (il existe de nombreux exemples d'assemblages supra-moléculaires par liaisons de type van der Waals, hydrogène ou ionique), où apparaissent parfois des dissymétries électroniques pouvant aller jusqu'à donner des ions par solvatation (solvants polaires).

Dès lors, on doit conclure que le dihydrogène (H2), le dichlore, le difluor et tant d'autres gaz diatomiques, sont électriquement neutres. Ce qui laisse entendre que lorsqu'ils sont isolés, ils sont zérovalents, pour respecter l'équivalence qu'il doit y avoir dans toute équation équilibrée en charges et globalement neutre comme : 2 H2 + O2 = 2H2O. Ici, dans la partie des réactifs, le dihydrogène et le dioxygène sont des molécules isolées et donc n'ont pas de charge propre, comme H2O (bien que molécule polaire). L'équation chimique vérifie donc la neutralité de la charge globale.

La forme et la taille d'une molécule (ou de l'une de ses parties) peut jouer un rôle dans son aptitude à réagir. La présence de certains atomes ou groupes d'atomes à l'intérieur d'une molécule joue un rôle majeur dans sa capacité à se rompre ou à fixer d'autres atomes issus d'autres corps, c'est-à-dire à se transformer pour donner naissance à d'autres molécules.

Les différents modes de représentation des molécules sont destinés à expliciter les différents sites réactifs ; certains enchaînements d'atomes, appelés groupes fonctionnels, produisent ainsi des similitudes de propriétés, tout particulièrement dans les composés organiques. [Web 28]

5.3- Les macromolécules et polymères

Les molécules possédant au moins plusieurs dizaines d'atomes sont appelées macromolécules ou polymères. [Web 28]

6. Données en chimie

Beaucoup de connaissances chimiques ont été dérivées des données. La chimie doit offrir une gamme riche des données sur les propriétés physiques, chimiques et biologiques, par exemple, données binaires pour la classification, vraies données pour la modélisation, et données spectrales ayant une densité élevée de l'information. Ces données doivent être introduites dans une forme favorable à l'échange d'information facile et analyse de données. [2]

6.1- Les sources de données et les bases de données

L'énorme quantité de données en chimie a mené au développement des bases de données pour stocker et disséminer les données en forme électronique. Par exemple, des bases de données ont été développées pour la littérature chimique, composés chimiques, structures 3D, réactions, et spectres. L'Internet est de plus en plus employé pour distribuer des données et l'information en chimie. [2]

6.2- Méthodes d'analyse de données

Une variété de méthodes pour apprendre des données par des méthodes d'étude inductives sont employées dans la chimie, par exemple, statistiques, méthodes d'identification de modèle, réseaux neurones artificiels, et algorithmes génétiques. Ces méthodes peuvent être classifiées dans des méthodes d'apprentissage supervisé et non supervisé et sont employés pour la classification ou modélisation quantitatif.

7. Les formats des données chimique

Il existe deux techniques principales pour représenter les structures chimiques dans les bases numériques : [Web 29]

- Tables de connexions / matrices d'adjacences / listes avec des informations supplémentaires sur la liaison chimique (arêtes) et données atomiques (nœuds) comme :

MDL Molfile :

Un MDL Molfile est un format de fichier contenant des informations sur les atomes, les liaisons, la connectivité et les coordonnées d'une molécule. Il se compose de quelques informations d'en-tête, la table de connexion (CT) contenant des informations sur les atomes, puis les connexions et les types de liaison, suivis de sections pour des informations plus complexes. Il est suffisamment courant pour que la plupart, sinon la totalité, des systèmes/applications logiciels de chemoinformatique soient capables de lire le format, mais pas toujours au même degré. [Web 30]

PDB :

La banque de données sur les protéines ou BDP du Research Collaboratory for Structural Bioinformatics, plus communément appelée Protein Data Bank ou PDB est une collection mondiale de données sur la structure tridimensionnelle (ou structure 3D) de macromolécules biologiques : protéines, essentiellement, et acides nucléiques. [Web 31]

CML :

Le Chemical Markup Language (CML) est un format pour les données chimiques. Ce format est basé sur du XML. Il s'agit de la première implémentation spécifique à un domaine (chimie) basée strictement sur XML. [Web 32]

- notation linéaire basée sur un parcours en largeur ou un parcours en profondeur :

SLN :

Un **signal de localisation nucléaire** (SLN) ou NLS (de l'anglais Nuclear localization sequence) est une petite séquence d'acides aminés (8 à 10 acides aminés) qui cible les protéines vers le noyau de la cellule. Les protéines portant un signal de localisation nucléaire sont reconnues par la protéine importune dans le cytosol et sont guidées vers les pores nucléaires. [Web 33]

WLN :

WLN a été inventé en 1949, par William J. Wiswesser, comme l'une des premières tentatives de codification de la structure chimique en tant que notation linéaire, permettant la collation sur des cartes perforées à l'aide de machines de tabulation automatique et des premiers ordinateurs électroniques. WLN était un précurseur de la notation SMILES utilisée dans les systèmes informatiques modernes, qui tentait de simplifier les règles complexes utilisées dans l'encodage WLN. [Web 34]

InChI :

L'International Chemical Identifier ou InChI (en français : Identifiant chimique international) est un identifiant textuel pour les substances chimiques, conçu pour être un standard d'encodage des informations moléculaires accessible humainement et pour faciliter la recherche de telles informations dans les bases de données ou sur le web. [Web 35]

SMILES/SMARTS :

La spécification d'entrée de ligne d'entrée moléculaire simplifiée ou SMILES est une spécification permettant de décrire sans ambiguïté la structure des molécules chimiques à l'aide de courtes chaînes ASCII. Les chaînes SMILES peuvent être importées par la plupart des éditeurs de molécules pour être reconverties en dessins bidimensionnels ou en modèles tridimensionnels des molécules. [Web 36]

L'avantage principale d'une représentation informatique est la possibilité d'un stockage croissant et d'une recherche rapide et flexible.

8. Exemple d'application

Les applications de la chemoinformatique sont nombreuses :

- Gestion de bases de données de molécules et de réactions.
- Prédiction de propriétés physiques, chimiques ou biologiques.
- Aide pour la conception de nouveaux médicaments.
- Résolution de structures moléculaires.
- Prédiction de réactions chimiques. [Web 15]

Conclusion

Nous avons analysé le concept de chemoinformatique sous l'angle de sa modélisation. Il nous a été donné de constater que dans ce domaine, d'énorme volume d'information produite par la recherche en chimie ne peut être traitée et analysée que par les moyens informatiques. Dans ce chapitre nous avons présenté la définition de la chemoinformatique en tant que domaine de la science qui consiste en l'application de l'informatique aux problèmes relatifs à la chimie.

CHAPITRE 3
CONCEPTION ET IMPLEMENTATION

1. Introduction

Dans ce chapitre je montre mes démarches de développement et des méthodes informatiques que j'ai utilisé pour analyser de données chimio-moléculaires pour mon système d'apprentissage. Ce chapitre vous permet de voir le fonctionnement, programmation et l'exécution de notre application, ainsi que les différentes technologies et les outils utilisées pour y parvenir.

2. Conception

2.1- Architecture fonctionnelle de l'application :

Nous allons appliquer une approche visant à transformer des données brutes qui sont en format smiles en image 2D pour les rendre accessibles au traitement après on divise les données en trois parties (train, validation et test). Nous pouvons effectuer l'apprentissage pour construire un modèle de prédiction de leurs classes active ou non active en utilisant CNN.

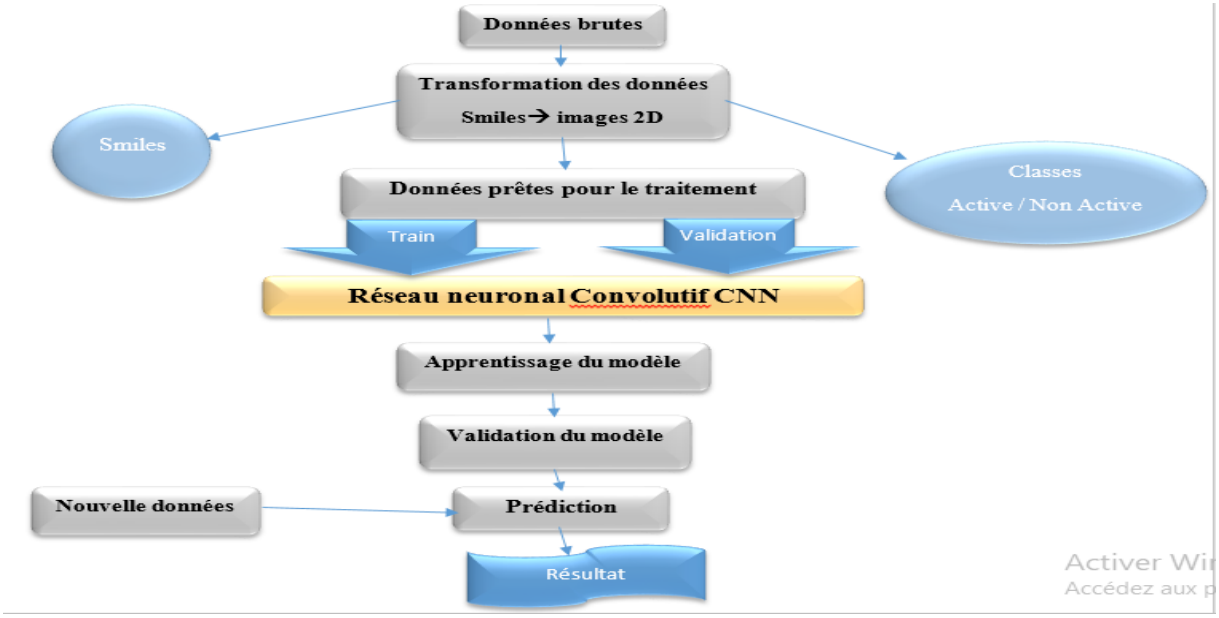


Figure 18 : Architecture de l'application.

2.2- Source de la base :

J'ai choisi 1156 molécules pour la classification. Nous allons ici passer en revue une approche dépend sur des images pour prédire l'activité ou l'inactivité. Nous avons effectué l'apprentissage sur des ensembles de données (format smile) qui sont transformés en image 2D.

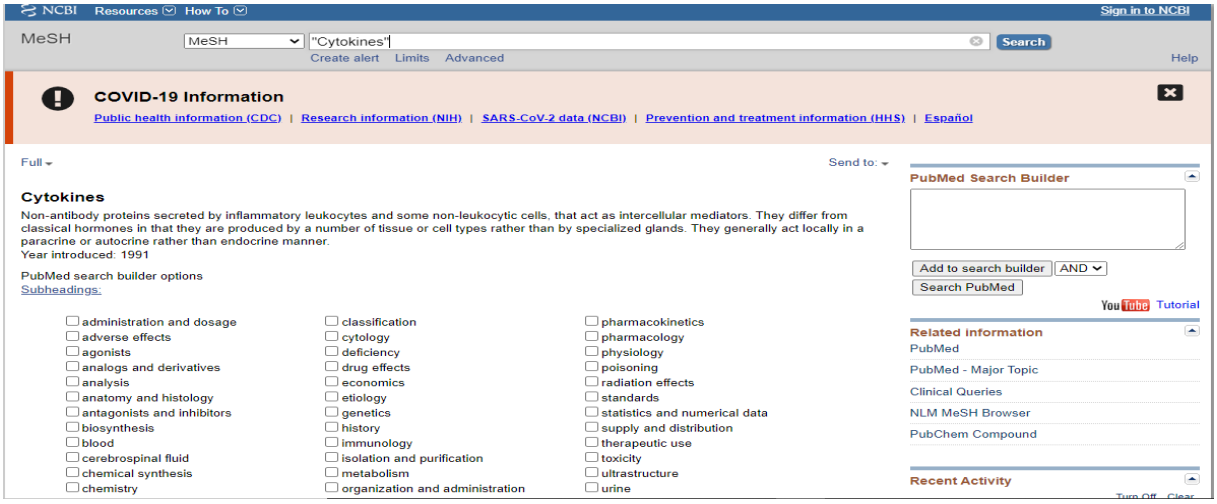


Figure 19 : Source de la base.

2.3- Modélisation des données

On peut modéliser nos données en 4 couches qui sont :

- **La couche d'entrée :** que de descripteurs pour l'ensemble de données. Chaque neurone est connecté aux neurones de la couche cachée.
- **La couche convolution :** permet d'apprendre les motifs les plus importants.
- **La couche cachée** : elle est constituée d'un nombre variable de neurones. Pour chaque neurone, le réseau effectue une opération de somme pondérée avec les différents poids de chaque neurone d'entrée.
- **La couche de sortie** : le nombre de neurones est égale au nombre de propriétés modélisées.

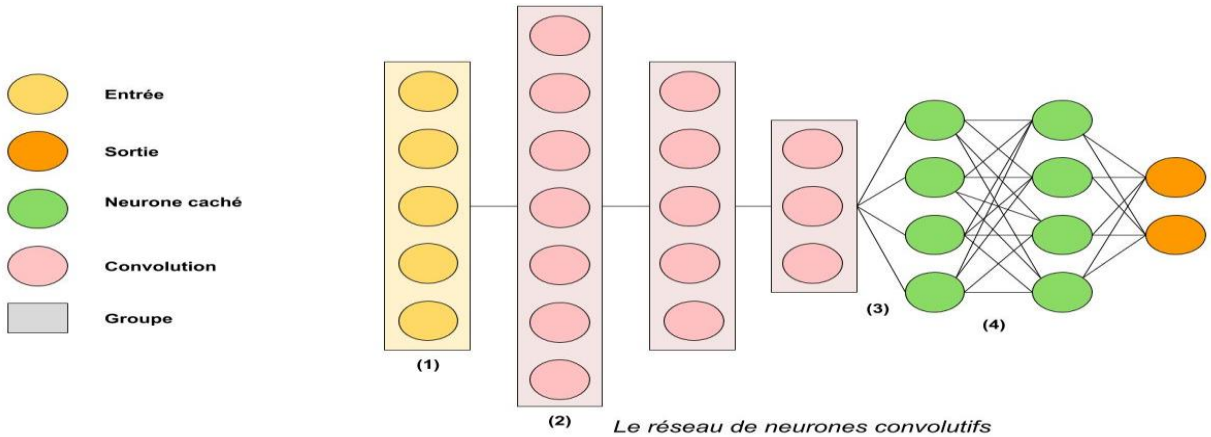


Figure 20 : Modélisation de neurones.

3. Implémentation

3.1- Les outils utilisés

3.1.1- Python

Python est l'un des langages de programmation les plus populaires et est connu pour sa syntaxe simple et sa vaste collection de bibliothèques. Il aide les développeurs à créer des applications écrivant moins de lignes de codes et à les rendre plus productifs.

Enfin, concernant le domaine de l'apprentissage automatique Python se distingue tout particulièrement en offrant une pléthore de librairies de très grande qualité, couvrant tous les types d'apprentissages disponibles sur le marché. [Web 37]

Figure 21 : lego de langage python. [Web 38]

3.1.2- Anaconda3

Anaconda est une distribution libre et open-source des langages de programmation Python et R appliqué au développement d'applications dédiées à la science des données et à l'apprentissage automatique (traitement de données à grande échelle, analyse prédictive, calcul scientifique), qui vise à simplifier la gestion des paquets et de déploiement. [Web 39]

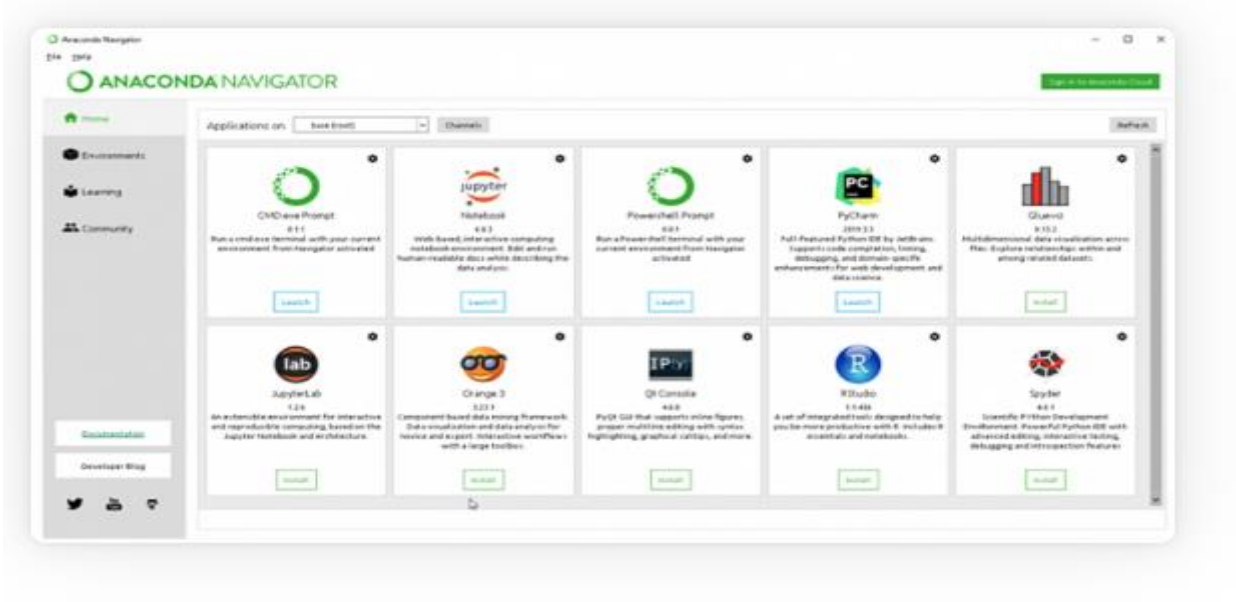


Figure 22 : Plateforme d'Anaconda3. [Web 39]

3.1.3- Jupyter Notebook

Jupyter Notebook est une application Web Open Source permet de créer et de partager des documents contenant du code (exécutable directement dans le document), peut rassembler du texte, des images, des formules mathématiques et du code informatique exécutable. Ils sont manipulables interactivement dans un navigateur web. Il est possible de faire du traitement de données, de la modélisation statistique, de la visualisation de données et apprentissage automatique. [Web 40]

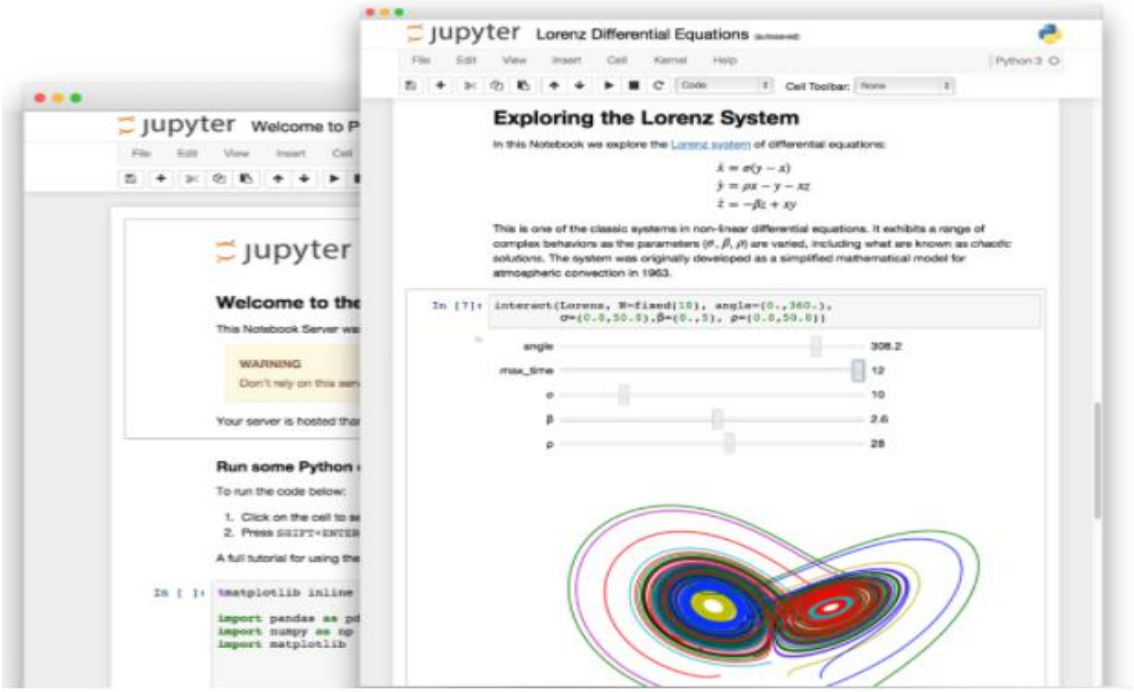


Figure 23 : Plateforme de jupyter notebook. [Web 40]

3.2- Les étapes de l'implémentation

Cette phase se compose de différentes étapes :

- L'importation des bibliothèques.
- L'importation du dataset.
- Prétraitement des données.
- Phase d'apprentissage.
- Phase de test.
- L'évaluation.

3.2.1- Importation des bibliothèques (package) :

La première chose à faire avant de débuter chaque programme et d'importer les bibliothèques importantes et nécessaires pour bien mener les différentes tâches.

- **Pandas :**

  Pandas est une bibliothèque open-source qui offre s une large gamme d'outils pour la manipulation et l'analyse des données. Avec cette bibliothèque, vous pouvez lire des données à partir d'un large éventail de sources telles que CSV, bases de données SQL, fichiers JSON et Excel. [Web 41]

- **RDkit :**

  Il présente une solution statistique du problème de la difficulté du calcul direct des propriétés physiques et biologiques à partir de la structure. L'intérêt d'un modèle de RDKIT est de tirer des informations à partir de l'ensemble des descripteurs numériques caractérisant la structure moléculaire et prédire ainsi La dimension de nouvelles structures. C'est une boîte à outils Open Source pour Cheminformatics, La majorité des fonctionnalités moléculaires de base se trouvent dans cet outil, comme dessiner les molécules, générer un bloc mol pour une molécule qui n'a pas de coordonnées entraînera, par défaut, automatiquement la génération de coordonnées. [4]

- **Scikit-learn :**

  Scikit-learn est une bibliothèque en Python qui fournit de nombreux algorithmes d'apprentissage non supervisés et supervisés. Il s'appuie sur certaines des technologies que vous connaissez peut-être déjà, comme NumPy, les pandas et Matplotlib. [Web 42]

- **PIL :**

  Python Imaging Library (ou PIL) est une bibliothèque de traitement d'images pour le langage de programmation Python. Elle permet d'ouvrir, de manipuler, et de sauvegarder différents formats de fichiers graphiques. [Web 43]

- **Tensorflow :**

  TensorFlow est une plate-forme Open Source de bout en bout dédiée au machine learning. Il est l'un des outils les plus utilisés en IA dans le domaine de l'apprentissage machin. Elle propose un écosystème complet et flexible d'outils, de bibliothèques et de ressources communautaires permettant aux chercheurs d'avancer dans le domaine du machine learning, et aux développeurs de créer et de déployer facilement des applications qui exploitent cette technologie. [Web 44]

- **Keras :**

  Keras est une bibliothèque Python open source de premier plan écrite pour la construction de réseaux de neurones et de projets d'apprentissage automatique. Il peut fonctionner sur Deeplearning4j, MXNet, Microsoft Cognitive Toolkit (CNTK), Theano ou TensorFlow. [Web 41]

- **Numpy :**

  Numpy est une bibliothèque open source associée au langage Python. Elle est très utile pour effectuer des opérations mathématiques et statistiques en Python. Elle fonctionne à merveille pour la multiplication de matrices ou de tableaux multidimensionnels. [Web 45]

- **Split_folder :**

  Split_folder est une bibliothèque open source associée au langage Python, permet à l'utilisateur de diviser le dossier qui porter base de données (image).

- **Matplotib :**

  Matplotlib est une bibliothèque Python capable de générer des graphiques de qualité. Il peut être utilisé dans les scripts Python, Python et IPython Shell, les notebooks Jupyter et les serveurs d'applications Web. Elle essaie et de rendre les choses complexes possibles. Vous pouvez utiliser quelques lignes de code pour générer des graphiques, des histogrammes, des graphiques à barres, des graphiques en nuage de points, etc. [Web 46]

- **OS :**

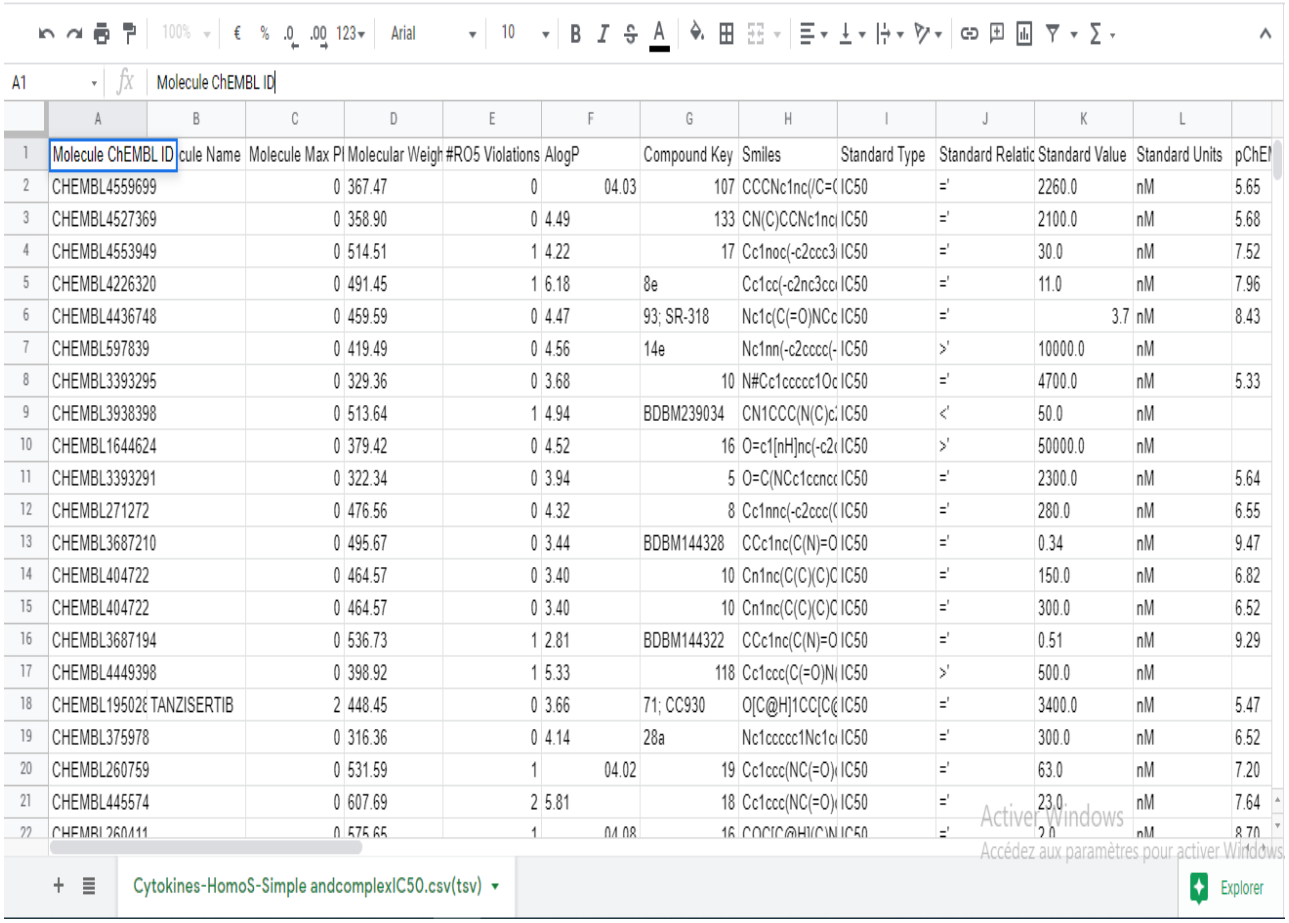
  Le module OS de Python fournit des fonctions pour interagir avec le système d'exploitation. Le système d'exploitation fait partie des modules utilitaires standard de Python. Ce module fournit un moyen portable d'utiliser les fonctionnalités dépendantes du système d'exploitation. Les modules *os* et *os.path* incluent de nombreuses fonctions pour interagir avec le système de fichiers. [Web 47]

- **CV2 (OpenCV) :**

  Le CV2 (OpenCV) est une bibliothèque graphique libre, initialement développée par Intel, spécialisée dans le traitement d'images en temps réel. En effet, la structure de base est la matrice. [Web 48]

3.2.2- Importation du dataset :

Dans ce mémoire nous allons utiliser la base de données chemogenomics, Il s'agit d'un ensemble de données sur les molécules chimiques.



	Molecule ChEMBL ID	cule Name	Molecule Max Ph	Molecular Weigh	#RO5 Violations	AlogP	Compound Key	Smiles	Standard Type	Standard Relatic	Standard Value	Standard Units	pChEM
2	CHEMBL4559699		0	367.47	0	04.03	107	CCCNc1nc(/C=(	IC50	=	2260.0	nM	5.65
3	CHEMBL4527369		0	358.90	0	4.49	133	CN(C)CCNc1nc	IC50	=	2100.0	nM	5.68
4	CHEMBL4553949		0	514.51	1	4.22	17	Cc1noc(-c2ccc3	IC50	=	30.0	nM	7.52
5	CHEMBL4226320		0	491.45	1	6.18	8e	Cc1cc(-c2nc3cc	IC50	=	11.0	nM	7.96
6	CHEMBL4436748		0	459.59	0	4.47	93; SR-318	Nc1c(C(=O)NCc	IC50	=	3.7	nM	8.43
7	CHEMBL597839		0	419.49	0	4.56	14e	Nc1nn(-c2cccc(-	IC50	>	10000.0	nM	
8	CHEMBL3393295		0	329.36	0	3.68	10	N#Cc1ccccc1Oc	IC50	=	4700.0	nM	5.33
9	CHEMBL3938398		0	513.64	1	4.94	BDBM239034	CN1CCC(N(C)c:	IC50	<	50.0	nM	
10	CHEMBL1644624		0	379.42	0	4.52	16	O=c1[nH]nc(-c2(	IC50	>	50000.0	nM	
11	CHEMBL3393291		0	322.34	0	3.94	5	O=C(NCc1ccncc	IC50	=	2300.0	nM	5.64
12	CHEMBL271272		0	476.56	0	4.32	8	Cc1nnc(-c2ccc((	IC50	=	280.0	nM	6.55
13	CHEMBL3687210		0	495.67	0	3.44	BDBM144328	CCc1nc(C(N)=O	IC50	=	0.34	nM	9.47
14	CHEMBL404722		0	464.57	0	3.40	10	Cn1nc(C(C)(C)C	IC50	=	150.0	nM	6.82
15	CHEMBL404722		0	464.57	0	3.40	10	Cn1nc(C(C)(C)C	IC50	=	300.0	nM	6.52
16	CHEMBL3687194		0	536.73	1	2.81	BDBM144322	CCc1nc(C(N)=O	IC50	=	0.51	nM	9.29
17	CHEMBL4449398		0	398.92	1	5.33	118	Cc1ccc(C(=O)N(	IC50	>	500.0	nM	
18	CHEMBL19502	TANZISERTIB	2	448.45	0	3.66	71; CC930	O[C@H]1CC[C(	IC50	=	3400.0	nM	5.47
19	CHEMBL375978		0	316.36	0	4.14	28a	Nc1ccccc1Nc1c	IC50	=	300.0	nM	6.52
20	CHEMBL260759		0	531.59	1	04.02	19	Cc1ccc(NC(=O)(	IC50	=	63.0	nM	7.20
21	CHEMBL445574		0	607.69	2	5.81	18	Cc1ccc(NC(=O)(	IC50	=	23.0	nM	7.64
22	CHEMBL260411		0	575.65	1	04.08	16	COC[C@H](C)N	IC50	=	2.0	nM	8.70

Cytokines-HomoS-Simple andcomplexIC50.csv(tsv)

Explorer

Figure 24 : La base de données formats xlsx.

3.2.3- Prétraitement des données :

Le prétraitement des données est une étape très importante qui consiste à transformer les formats smiles des molécules en format image 2D pour pouvoir les traiter.

3.2.3.1- Importation et lecture des données

La figure suivante contient un pseudo code de la lecture des bases et l'affichage de ses tailles :

```
Entrée [2]: pd.read_excel("C:/Users/CHULUS Info/Desktop/Mon cursus informatique 2016-2021/Master GADM/Mémoire/dataset/Cytokines.xlsx", engin
ead()
```

Out[2]:

	Molecule ChEMBL ID	Molecule Name	Molecule Max Phase	Molecular Weight	#RO5 Violations	AlogP	Compound Key	Smiles	Standard Type	Standard Relation	...
0	CHEMBL4559699	NaN	0	367.47	0	2021-03-04 00:00:00	107	CCCNc1nc(/C=C/c2ccc(S(C)(=O)=O)cc2)nc2ccccc12	IC50	=	...
1	CHEMBL4527369	NaN	0	358.90	0	4.49	133	CN(C)CCNc1nc(/C=C/c2cccs2)nc2ccc(Cl)cc12	IC50	=	...
2	CHEMBL4553949	NaN	0	514.51	1	4.22	17	Cc1noc(-c2ccc3nc(Nc4cc([C@@H](C)N5CCN(C(=O)CC(...	IC50	=	...
3	CHEMBL4226320	NaN	0	491.45	1	6.18	8e	Cc1cc(-c2nc3ccccnn3c2-c2ccnc(NC(=O)c3cccc(C(F)(...	IC50	=	...
4	CHEMBL4436748	NaN	0	459.59	0	4.47	93; SR-318	Nc1c(C(=O)NCc2ccc(C(=O)NCCCC3CCCCC3)cc2)onn1-c...	IC50	=	...

5 rows × 45 columns

Figure 25 : le pseudo code pour lecture de la base.

3.2.3.2- Eliminer les éléments indésirables :

3.2.3.2.1- Nettoyage de la base

Dans ce cas, on a une base brute donc il faut la nettoyer les étapes de nettoyage sont :

- Éliminer Les molécules qui sont non déterminer (qui ont valeur not determined dans la colonne comment).
- Remplace les valeurs non nM en nM.
- Remplace les valeurs not (Not Active inihibition ...) en Not Active pour extraire une base des molécules non active.
- Remplace les valeurs null dans la colonnes Smiles avec des zéros pour les éliminer.
- Affiche la base sans les valeurs null de la colonne on considère comme une nouvelle base qui on va les traiter avec des molécules qui ont tout leur format smiles.
- Mettre les molécules qui sont déjà mentionner comme des molécules non actives.
- Enfin, extraire une base sans les molécules qui sont déjà non actives et on va la traiter.

Entrée [3]:
```
## eliminer les lignes qui ont la valeur not determinet (dans comment)
dt = data[data.Comment != 'Not Determined']

## remplace les valeurs not nM en nM
dt['Standard Units'] = dt['Standard Units'].replace(['ug.mL-1'],'nM')

## remplace les valeurs not Not Active (inihibition ....) en Not Active
dt['Comment'] = dt['Comment'].replace(['Not Active (inhibition < 50% @ 10 uM and thus dose-reponse curve not measured)'],'Not Act

## remplace les valeurs null de column Smiles with 0
dt['Smiles']=dt['Smiles'].fillna(0)

## affiche dataset sans qui ont la valeur de smiles 0
dt = dt[dt.Smiles != (0)]

## met les molécules qui sont deja mentionnée non active dans un base ..... (1)
dt_nonactiv = dt[dt.Comment == ('Not Active')]

## nouvelle base sans molécule déja non active
dt1 = dt[dt.Comment != 'Not Active']
```

Figure 26 : Les étapes de nettoyage.

3.2.3.2.2- l'affichage toute les base qui sont active et non active :

- L'affichage des molécules (la base) qui sont déjà mentionnée non active.

Entrée [4]:
```
## la base pour les molécules qui sont déja mentionnée non_active
dt_nonactiv
```

Out[4]:

	Molecule ChEMBL ID	Molecule Name	Molecule Max Phase	Molecular Weight	#RO5 Violations	AlogP	Compound Key	Smiles	Star
55	CHEMBL1670	MITOTANE	4	320.05	1	5.93	MITOTANE	Clc1ccc(C(c2ccccc2Cl)C(Cl)Cl)cc1	
85	CHEMBL388978	STAUROSPORINE	0	466.54	0	4.35	Staurosporine	CN[C@@H]1C[C@H]2O[C@@](C)([C@@H]1OC)n1c3ccccc3...	
151	CHEMBL26260	BETA-NAPHTHOFLAVONE	0	272.30	0	4.61	BETA-NAPHTHOFLAVONE	O=c1cc(-c2ccccc2)oc2ccc3ccccc3c12	
152	CHEMBL1542	AZATHIOPRINE	4	277.27	0	1.15	AZATHIOPRINE	Cn1cnc([N+](=O)[O-])c1Sc1ncnc2nc[nH]c12	
153	CHEMBL661	ALPRAZOLAM	4	308.77	0	3.58	ALPRAZOLAM	Cc1nnc2n1-c1ccc(Cl)cc1C(c1ccccc1)=NC2	
...	...	...	...	...	...	...	...	...	
11165	CHEMBL4576555	NaN	0	593.82	2	6.90	None	CN1CCN(C(=O)C(C)(C)c2ccc(C(=O)Nc3cn4cc(-c5cc(C...	

Figure 27 : Les molécules qui sont déjà non active.

- Divise la base en deux parties(Split), des molécules active et les autres non active, Si on trouve l'IC50 des molécules <= 1000 on considere ces molécules comme active les autres inactives.

Entrée [5]:
```
## split la base en deux partie
## active (IC50 (standard value) <= 1000 nM )
s = 1000

df1 = dt1[dt1['Standard Value'] <= s]
df1
```

Out[5]:

	Molecule ChEMBL ID	Molecule Name	Molecule Max Phase	Molecular Weight	#RO5 Violations	AlogP	Compound Key	Smiles	Standard Type	Standa Relati
2	CHEMBL4553949	NaN	0	514.51	1	4.22	17	Cc1noc(-c2ccc3nc(Nc4cc([C@@H](C)N5CCN(C(=O)CC(...	IC50	
3	CHEMBL4226320	NaN	0	491.45	1	6.18	8e	Cc1cc(-c2nc3ccccnn3c2-c2ccnc(NC(=O)c3cccc(C(F)(...	IC50	
7	CHEMBL3938398	NaN	0	513.64	1	4.94	BDBM239034	CN1CCC(N(C)c2ccc(C(=O)Nc3n[nH]c4cc(OCCOCc5cccc...	IC50	
10	CHEMBL271272	NaN	0	476.56	0	4.32	8	Cc1nnc(-c2ccc(C)c(-c3ccc(C(=O)NCc4ccc(NS(C)(=O...	IC50	
11	CHEMBL3687210	NaN	0	495.67	0	3.44	BDBM144328	CCc1nc(C(N)=O)c(Nc2ccc(N3CCN(C(C)C)CC3)c(C)c2)...	IC50	
...	...	...	...	...	...	...	...	...	...	...

Figure 28 : Les molécules qui sont active (IC50 <= 1000).

Entrée [6]:
```
##Non_active (standard value > 1000 nM)
df2 = dt1[dt1['Standard Value'] > s]
df2
```

Out[6]:

	Molecule ChEMBL ID	Molecule Name	Molecule Max Phase	Molecular Weight	#RO5 Violations	AlogP	Compound Key	Smiles	Standard Type	Standa Relati
0	CHEMBL4559699	NaN	0	367.47	0	2021-03-04 00:00:00	107	CCCNc1nc(/C=C/c2ccc(S(C)(=O)=O)cc2)nc2ccccc12	IC50	
1	CHEMBL4527369	NaN	0	358.90	0	4.49	133	CN(C)CCNc1nc(/C=C/c2cccs2)nc2ccc(Cl)cc12	IC50	
4	CHEMBL4436748	NaN	0	459.59	0	4.47	93; SR-318	Nc1c(C(=O)NCc2ccc(C(=O)NCCCC3CCCCC3)cc2)onn1-c...	IC50	
5	CHEMBL597839	NaN	0	419.49	0	4.56	14e	Nc1nn(-c2cccc(-c3ccc4cc[nH]c4c3)c2)cc1-c1ccc2c...	IC50	
6	CHEMBL3393295	NaN	0	329.36	0	3.68	10	N#Cc1ccccc1Oc1ccc(C(=O)NCc2ccncc2)cc1	IC50	
...	...	...	...	...	...	...	...	...	...	...
11278	CHEMBL4227523	NaN	0	438.47	0	4.86	8o	Cc1ccc(C(=O)Nc2cc(-c3c(-c4ccc(F)c(C)c4)nc4cccn...	IC50	

Figure 29 : Les molécules qui sont non active (IC50 > 1000)

3.2.3.3- Convertir les formats smiles en format image 2D :

3.2.3.3.1- L'affichage de tous les formats smiles :

La figure suivante contient des pseudo code de l'affichage des formats smiles.

```
Entrée [7]: ### les formats smiles de toutes la base active et non_active
            active = df1['Smiles']
            nonactiv1 = dt_nonactiv['Smiles'] ## déja mentionnée
            nonactiv2 = df2['Smiles']
            active

Out[7]: 2        Cc1noc(-c2ccc3nc(Nc4cc([C@@H](C)N5CCN(C(=O)CC(...
        3        Cc1cc(-c2nc3cccnn3c2-c2ccnc(NC(=O)c3cccc(C(F)(...
        7        CN1CCC(N(C)c2ccc(C(=O)Nc3n[nH]c4cc(OCCOCc5cccc...
        10       Cc1nnc(-c2ccc(C)c(-c3ccc(C(=O)NCc4ccc(NS(C)(=O...
        11       CCc1nc(C(N)=O)c(Nc2ccc(N3CCN(C(C)C)CC3)c(C)c2)...
                                    ...
        11282    Nc1nccc2c(S(=O)(=O)NCCNC/C=C/c3ccc(-c4cccnc4)c...
        11284    COc1ccc(-c2cc3c(NN4C(=O)C=C(C)C4=O)nc(-c4cccs4...
        11285     CC(C)c1ccc(/C=C/c2nc(NCCN(C)C)c3cc(Cl)ccc3n2)cc1
        11287    Nc1c(C(=O)c2ccc(F)cc2F)ccc(=O)n1-c1c(F)cc(OCCC...
        11288    CCCCN(C)C1CCN(c2ccc(Nc3ncc(C)c(-c4cnn(C(C)C)c4...
        Name: Smiles, Length: 5974, dtype: object
```

Figure 30 : les formats smiles des molécules qui sont active.

3.2.3.3.2- Crée un dossier (répertoire) :

On a créé un répertoire pour l'enregistrement des images des molécules, voici la figure suivante :

```
Entrée [11]: ## pour crée dossier pour enregistrer les images
             import os
             !mkdir data\dataset\Non_Active
             !mkdir data\dataset\Active
```

Figure 31 : Pseudo code pour la création de dossier.

3.2.3.3.3- Convertir les smiles et enregistrer les images des molécules :

La figure suivante contient un pseudo code de convertir les formats smiles en format image 2D après on va les mettre dans un répertoire :

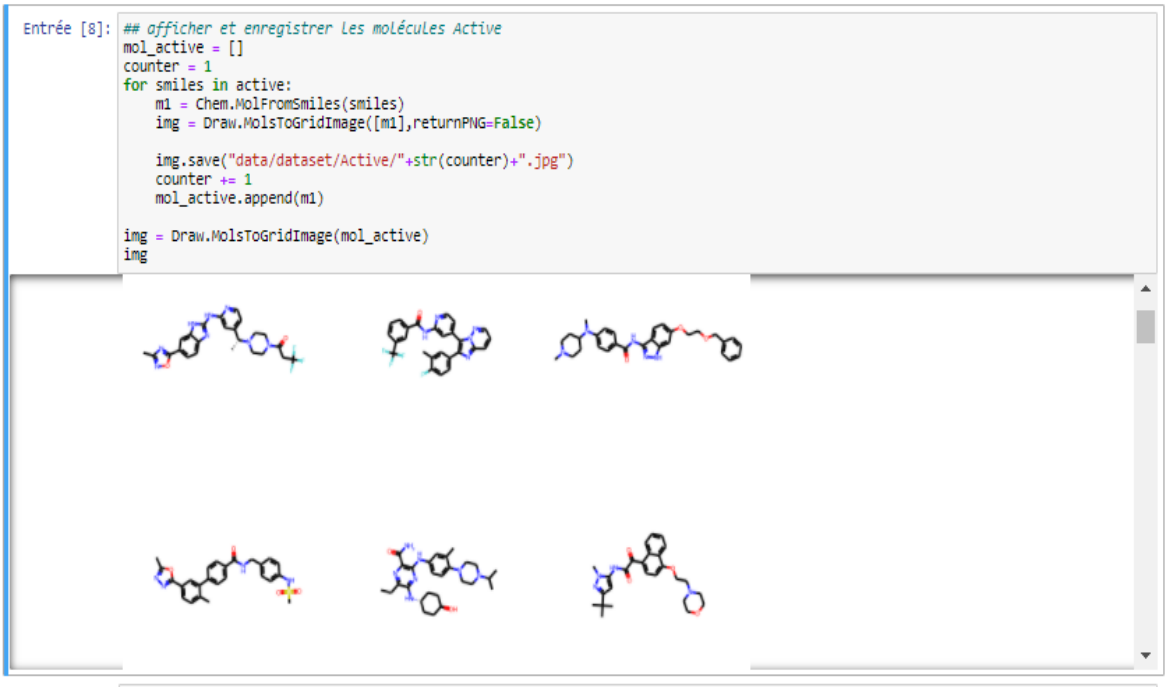


Figure 32 : Convertir les smiles et enregistrer les images de molécules actives.

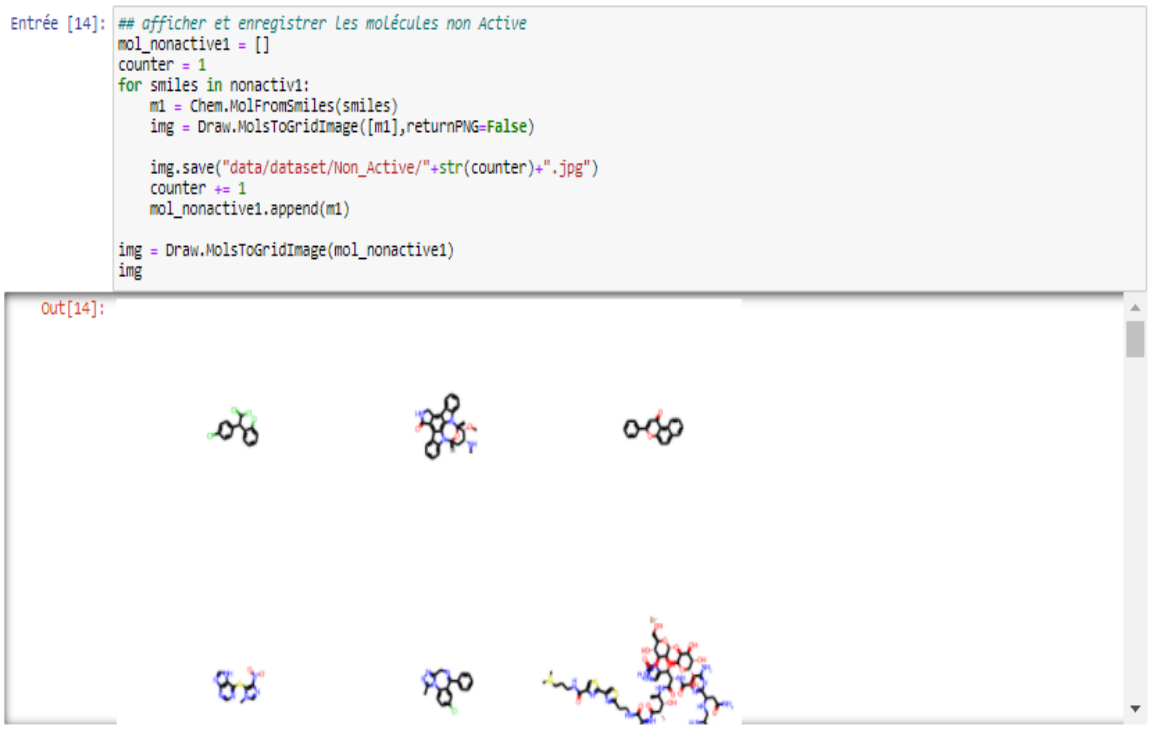


Figure 33 : Convertir et enregistrer les images de molécules qui sont déjà inactives.

```
Entrée [15]: ## afficher et enregistrer les molécules non Active
mol_nonactive2 = []
counter = 854
for smiles in nonactiv2:
    m1 = Chem.MolFromSmiles(smiles)
    img = Draw.MolsToGridImage([m1],returnPNG=False)

    img.save("data/dataset/Non_Active/"+str(counter)+".jpg")
    counter += 1
    mol_nonactive2.append(m1)

img = Draw.MolsToGridImage(mol_nonactive2)
img
```

Out[15]:

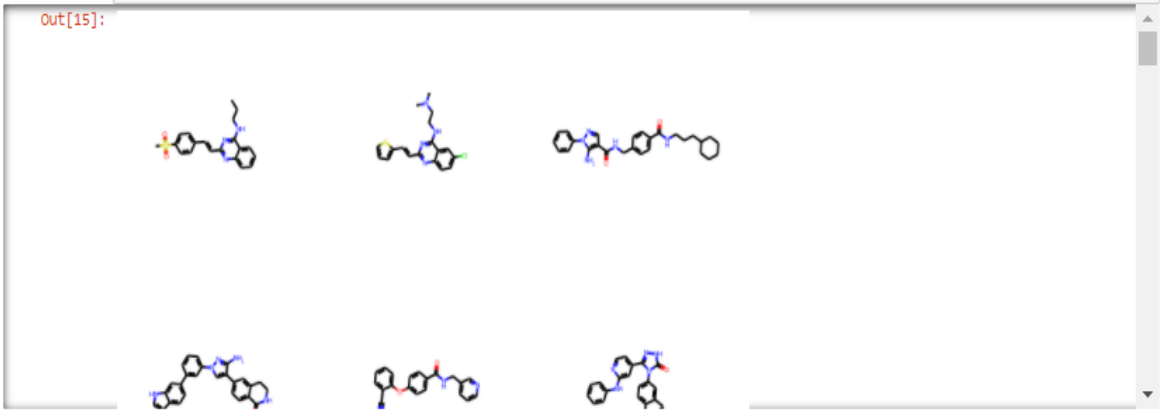


Figure 34 : Convertir les smiles et enregistrer les images de molécules inactives.

3.2.3.4- Séparation des données (Split_folder) :

La figure suivante représente la division du répertoire principale qu'on avait fait dans un nouveau répertoire qui contient trois dossier train, val et test compte tenu du pourcentage du ratio train = 70%, validation=20% et test= 10%.

```
Entrée [4]: ##split_folders (split data on new folder (train, val, test))
dataset = ('C:/Users/CHULUS Info/Desktop/dataset/')
datasets = ('C:/Users/CHULUS Info/Desktop/datasets/')

split_folder.ratio(dataset, datasets , seed = 1337, ratio = (0.7,0.2,0.1))

Copying (4181) of .. train/Active

100% 4181/4181 [00:59<00:00, 85.31it/s]

Copying (1194) of .. val/Active

100% 1194/1194 [00:16<00:00, 132.36it/s]

Copying (599) of .. test/Active

100% 599/599 [00:07<00:00, 92.37it/s]

Copying (3557) of .. train/Non_Active

100% 3557/3557 [00:44<00:00, 46.28it/s]

Copying (1016) of .. val/Non_Active

100% 1016/1016 [00:10<00:00, 129.83it/s]

Copying (509) of .. test/Non_Active

100% 509/509 [00:07<00:00, 44.71it/s]
```


Figure 35 : pseudo code pour la division de la base.

3.2.4- Phase d'apprentissage du model :

En Deep Learning, l'algorithme se construit une "représentation interne" afin de pouvoir effectuer les tâches requises (prédiction, reconnaissance, etc.), vous devez d'abord entrer un ensemble d'exemples de données afin qu'il puisse s'entraîner et s'améliorer, ce jeu de données s'appelle le training set.

La construction du modèle CNN en python se compose de plusieurs couche d'entrées, cachées et de sorties, Ksera propose :

A- Couche Conv2D :

Les CNN peuvent également apprendre la hiérarchie spatiale de ces motifs. Une première couche de convolution apprendra les petits motifs, une deuxième couche de convolution apprendra des motifs plus importants constitués des caractéristiques des premières couches, etc. Cela permet aux convnets d'apprendre efficacement des concepts de plus en plus complexes et abstraits.

Filter : Définit le nombre de filtres (le nombre de neurones) de sortie utilisés dans l'opération de convolution. Qui est 16 filtres.

La taille du motif : le nombre de pixel q'il contiendra aussi appeler kernel – 3×3 pixels.

Activation : (RelU) Rectified Linear Unit est une fonction qui transforme tous les données de valeur négative en 0, dans le but de ne pas activer tous les neurones en même temps. [Web 19]

Input_shape : C'est le tenseur de départ que vous envoyez à la première couche cachée. Ce tenseur doit avoir la même forme que vos données d'entraînement. Exemple : input_shape (largeur, hauteur,profondeur)→ input_shape(200,200,3).

Largeur : Largeur de l'image en pixels.

La hauteur : La hauteur de l'image en pixels.

Profondeur : Le nombre de canaux pour l'image.

B- Couche MaxPool2D :

Le MaxPooling est aussi une couche de convolution. Il extrait des motifs, des tendances d'une donnée. Mais là où le Conv2D extrait des caractéristiques d'une image pour créer des feature-maps, le MaxPooling2D, lui, extrait la valeur la plus importante de chaque motif des feature-maps. [Web 19]

Kernel : un kernel de taille (2,2).

C- Couche Flatten :

La couche Flatten permet d'aplatir le tenseur, de réduire sa dimension. Elle prend en entrée un 3D-tensor et retourne un 1D-tensor. Elle permet d'établir une connexion entre les couches de convolution et les couches de base du Deep Learning. Elle permet de diffuser la donnée à travers les couches en réduisant sa dimension.

D- Couche Dense Cachée :

Définir le nombre de nœuds qu'on veut, on a choisi 512 neurones dans la première couche cachée.

E- Couche Dense Sortie :

On considère cette couche comme une couche finale de le model.

```
Entrée [8]: model = tf.keras.models.Sequential([ tf.keras.layers.Conv2D(16,(3,3),activation = 'relu',input_shape = (200,600,3)),
                                                  tf.keras.layers.MaxPool2D(2,2),
                                                #
                                                  tf.keras.layers.Conv2D(32,(3,3),activation = 'relu'),
                                                  tf.keras.layers.MaxPool2D(2,2),
                                                #
                                                  tf.keras.layers.Conv2D(64,(3,3),activation = 'relu'),
                                                  tf.keras.layers.MaxPool2D(2,2),
                                                ##
                                                   tf.keras.layers.Flatten(),
                                                ##
                                                   tf.keras.layers.Dense(512,activation= 'relu'),
                                                ##
                                                  tf.keras.layers.Dense(1,activation= 'sigmoid')
                                                ])

Entrée [9]: model.summary()

Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 198, 598, 16)      448
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 99, 299, 16)       0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 97, 297, 32)       4640
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 48, 148, 32)       0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 46, 146, 64)       18496
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 23, 73, 64)        0
_________________________________________________________________
flatten (Flatten)            (None, 107456)            0
_________________________________________________________________
dense (Dense)                (None, 512)               55017984
_________________________________________________________________
dense_1 (Dense)              (None, 1)                 513
=================================================================
Total params: 55,042,081
Trainable params: 55,042,081
Non-trainable params: 0
_________________________________________________________________
```


Figure 36 : les différentes couches utilisées pour notre modèle CNN.

Maintenant que le modèle est défini, nous pouvons le compiler avant d'entraîner le modèle, il faut configurer le processus d'apprentissage en appelant la méthode compile qui accepte trois arguments :

Loss : il s'agit de la fonction de coût que le modèle va utiliser pour minimiser les erreurs. Elle peut être définie par son appellation (exemple : loss='binary_crossentropy').

Optimizer : a pour but de réduire l'erreur actuelle qui est définit dans loss (modifier les poids de neurones).

Metrics : les fonctions métriques sont similaires aux fonctions de perte, à quelle point notre réseau de neurone a juste sur les prédictions qui la faites.

```
Entrée [10]: ##Compile model
             model.compile(loss='binary_crossentropy',
                           optimizer = RMSprop(lr=0.001),
                           metrics =['accuracy'])
```

Figure 37 : pseudo code de la méthode compile et ces paramètres.

3.2.5- la phase test (lancement de l'apprentissage)

Une fois le modèle construit et sauvegardé nous passons à l'étape suivante pour ce faire nous allons additionner une partie pour l'apprentissage et une partie pour le test que nous avions préparé et ensuite nous faisons l'apprentissage avec le total de ces données, train_dataset et validation_dataset les données d'apprentissage et de validation des images.

```
Entrée [5]: train = ImageDataGenerator(rescale= 1/255)
            validation = ImageDataGenerator(rescale= 1/255)
            test = ImageDataGenerator(rescale= 1/255)
```

```
Entrée [6]: train_dataset = train.flow_from_directory('C:/Users/CHULUS Info/Desktop/datasets/train/',
                                                       target_size= (200,600),
                                                       batch_size = 3,
                                                       class_mode='binary')

            validation_dataset = validation.flow_from_directory('C:/Users/CHULUS Info/Desktop/datasets/val/',
                                                       target_size= (200,600),
                                                       batch_size = 3,
                                                       class_mode='binary')

            Found 7738 images belonging to 2 classes.
            Found 2210 images belonging to 2 classes.
```

Figure 38 : les données d'apprentissage et de validation.

Nous avons défini notre modèle et l'avons compilé pour qu'il soit prêt pour un calcul efficace, il est maintenant temps d'exécuter le modèle sur quelques données nous pouvons entraîner ou ajuster notre modèle sur nos données chargées en appelant la fonction fit() sur le modèle, l'entraînement se fait sur :

- **Epoch :**

  On passe par toutes les lignes du jeu de données d'entraînement (les itérations) et chaque Epoch est composée d'un ou plusieurs Batch, en fonction de la taille du Batch choisi et le modèle est adapté à de nombreuses époques.

- **Steps_per_epoch:** Entier.

  Nombre total d'étapes (lots d'échantillons) à produire du générateur avant de déclarer une époque terminée et de commencer l'époque suivante. Il doit généralement être égal à ciel (num_samples / batch_size). Facultatif pour la séquence : si non spécifié, utilisera le len (générateur) comme nombre d'étapes.

  Nous voulons entraîner le modèle suffisamment pour qu'il apprenne à faire une bonne (ou assez bonne) correspondance entre les données d'entrée et la classification de sortie. Le modèle comportera toujours une certaine erreur et précision, mais la quantité d'erreur se stabilisera après un certain point pour une configuration donnée du modèle. C'est ce qu'on appelle la convergence des modèles.

```
Entrée [34]: ##fit le model
model_fit = model.fit(train_dataset,
                      steps_per_epoch = 3,
                      epochs= 20,
                      batch_size = 900,
                      validation_data=validation_dataset)
```

```
Epoch 1/20
3/3 [==============================] - 29s 10s/step - loss: 0.4054 - accuracy: 0.7778 - val_loss: 0.7433 - val_accuracy: 0.48
82
Epoch 2/20
3/3 [==============================] - 28s 9s/step - loss: 0.7273 - accuracy: 0.6667 - val_loss: 0.7360 - val_accuracy: 0.459
7
Epoch 3/20
3/3 [==============================] - 28s 9s/step - loss: 0.7345 - accuracy: 0.4444 - val_loss: 0.6613 - val_accuracy: 0.475
1
Epoch 4/20
3/3 [==============================] - 28s 9s/step - loss: 0.6132 - accuracy: 0.6667 - val_loss: 0.6586 - val_accuracy: 0.532
1
Epoch 5/20
3/3 [==============================] - 28s 9s/step - loss: 0.5408 - accuracy: 0.6667 - val_loss: 0.6490 - val_accuracy: 0.567
4
Epoch 6/20
3/3 [==============================] - 28s 9s/step - loss: 0.9622 - accuracy: 0.6667 - val_loss: 0.6238 - val_accuracy: 0.634
8
Epoch 7/20
```

Figure 39 : l'entrainement du modèle en appelant la fonction fit ().

La fonction fit () retournera une liste avec quatre valeurs. La première sera la perte du modèle sur l'ensemble de données de train, le deuxième sera la précision du modèle sur l'ensemble de données train (train_dataset), La troisième sera la perte du modèle sur l'ensemble de données de validation et quatrième sera la précision du modèle sur l'ensemble de données validation (validation_dataset), on voit un message pour chacune des 150 Epochs imprimant la perte et la précision, suivi de l'évaluation finale du modèle formé sur l'ensemble de données de formation.

```
Entrée [31]: model.evaluate(train_dataset)

             2580/2580 [==============================] - 149s 58ms/step - loss: 0.6003 - accuracy: 0.6581

Out[31]: [0.6003321409225464, 0.6580511927604675]

Entrée [32]: model.evaluate(validation_dataset)

             737/737 [==============================] - 27s 36ms/step - loss: 0.5997 - accuracy: 0.66200s - loss: 0.6003 - accuracy

Out[32]: [0.5997092723846436, 0.6619909405708313]
```

Figure 40 : Pseudo code pour l'évaluation final de données train et données validation.

Idéalement, nous aimerions que la perte soit nulle et la précision de 1.0 (par exemple, 100%). Cela n'est possible que pour les problèmes du Deep Learning les plus insignifiants. Au lieu de cela, nous aurons toujours une certaine erreur dans notre modèle. L'objectif est de choisir une configuration de modèle et une configuration d'apprentissage qui permettent d'obtenir la perte la plus faible et la précision la plus élevée possible pour un ensemble de données donné, nous ne voulons que rapporter la précision(**Accuracy**), donc nous ignorerons la valeur de la perte(**loss**).

3.2.6- L'évaluation de model

Une fois qu'on a formé le modèle, comment puisse-nous l'utiliser pour faire des prévisions sur de nouvelles données ? Il s'agit d'appliquer un nouvel ensemble de données **pred_dataset** que le modèle n'a jamais vu auparavant, faire des prédictions est aussi simple que d'appeler la fonction ***predict*** **()** sur le modèle. Le but de l'évaluation est d'estimer au mieux les performances d'un modèle sur de nouvelles données.

```
Entrée [*]: pred = 'C:/Users/CHULUS Info/Desktop/datasets/test/'
            pred_dataset = train.flow_from_directory(pred,
                                                     target_size= (200,200),
                                                     batch_size = 3,
                                                     class_mode='binary')
```

Figure 41 : La nouvelle base pour la prédiction.

- **L'évaluation :**

```
Entrée [38]: model.evaluate(pred_dataset)
             370/370 [==============================] - 15s 40ms/step - loss: 0.6107 - accuracy: 0.6390
Out[38]: [0.6106906533241272, 0.6389891505241394]
```

Figure 42 : l'évaluation de model sur les données qu'il a jamais vues.

On observe au-dessus (Voir la Figure 43) que la précision **train_dataset** de notre modèle sur les données d'entrainement est **65%,** le plus important ce n'est pas cette valeur c'est la capacité du modèle de bien prédire les données qu'il a jamais vu.

On donne à notre modèle les données de l'évaluation **test_dataset** et lui essaye de nous prédire les classes Active ou non. Pour qu'on puisse savoir si le modèle est bon ou non, il va nous afficher **Loss** et l'**Accuracy**.

Les résultats obtenus sur les données tests représentées dans la figure ci-dessus nous ont donné un taux de **0.6389**, une précision environ **64%,** donc on peut dire que notre modèle est bon pour classifier.

- **Matrice de Confusion :**

  Une **matrice de confusion** est une matrice qui permet de mesurer la qualité d'un modèle de classification.

  Sur **les lignes** de la matrice, on trouve les classes réelles.

  Sur **les colonnes**, on retrouve les prévisions calculées par le modèle.

Dans notre cas, Il y a **1108 images** .

599 appartiennent à la classe 0 (Active), et 509 à la classe 1 (Non_Active).

- Le modèle en a détecté 699 images dans la classe 0 (colonne Active), et 409 dans la classe 1 (colonne Non_Active).
- Le modèle a détecté **correctement** 426 images de la classe 0 (Active), et 236 images de la classe 1 (Non_Active).
- On peut lire les données correctement prédites en regardant **la diagonale** de la matrice.

- Le modèle a prédit 173 images dans la classe 1 (Non_Active) alors que ce sont des données de la classe 0 (Active). On parle de **faux positifs**.
- Le modèle a prédit 273 échantillons dans la classe 0 (Active) alors qu'il s'agissait de données de la classe 1 (Non_Active). On parle de **faux négatifs**. Voici la figure suivante

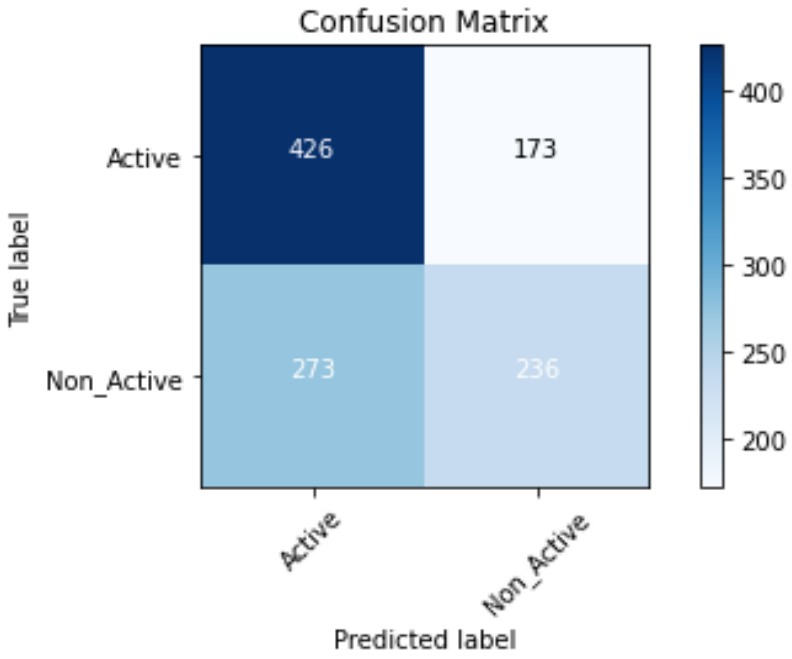


Figure 43 : Matrice de confusion.

Conclusion

Après avoir réalisé l'implémentation de mon modèle, qui se traduit en fait par la présentation de la modélisation des données, ainsi le processus général de classification et prédiction et ses différentes étapes, les outils, les technologies utilisées, plus la présentation des différentes étapes de la réalisation de ce système d'apprentissage profond, et par conséquence nous avons appris à utiliser les caractéristiques des images pour faire classifier les molécules. Cela nous mène à fermer cette mémoire avec une conclusion et perspectives

CONCLUSION ET PERSPECTIVES

Au cours de ce mémoire, nous avons détaillé notre démarche et notre état d'avancement final. Nous proposons différentes perspectives que nous aurions mises en œuvre afin de créer un modèle pour classifier et prédire l'inactivité ou l'activité d'une molécule. Nous nous concentrons sur le code Python.

Dans le deuxième chapitre, j'ai introduit les notions de base de la chemoinformatique. Après j'ai donné une petite définition de la chemoinformatique, la chemoinformation et les descripteurs moléculaires pour bien définir le domaine de travail. Ensuite j'ai focalisé sur les méthodes de la classification et la prédiction des molécules. Enfin, j'ai validé l'approche mentionnée ou j'ai montré l'efficacité de mon travail avec des captures d'écrans et des fragments de code source, en expliquant chaque point.

Pour finir, avant de passer aux perspectives, ce travail nous a permis de mettre en pratique nos connaissances sur les réseaux de neurones et d'en acquérir d'autres et le temps passé à lire des articles nous a servi d'une bonne initiation à la recherche.

Perspectives :

Comme perspectives de recherches futures, nous envisageons de :

1. De tester sur notre modèle d'autre base de données.
2. D'augmenter de nouvelle couche afin de voir ce que ça va donner (pour obtenir une performance élevé).

REFERENCES

[Web 1] Apprentissage automatique. (s.d.). Récupéré sur wikipedia: https://fr.wikipedia.org/wiki/Apprentissage_automatique

[Web 2] apprentissage par transfert. (s.d.). Récupéré sur Tech target: https://whatis.techtarget.com/fr/definition/Apprentissage-par-transfert

[Web 3] Apprentissage profond. (s.d.). Récupéré sur wikipedia: https://fr.wikipedia.org/wiki/Apprentissage_profond

[Web 4] Appréhendez le Deep Learning ou l'apprentissage profond. (s.d.). Récupéré sur openclassrooms: https://openclassrooms.com/fr/courses/6417031-objectif-ia-initiez-vous-a-lintelligence-artificielle/6823506-apprehendez-le-deep-learning-ou-lapprentissage-profond

[Web 5] intelligence artificielle deep-learning. (s.d.). Récupéré sur futura-sciences: https://www.futura-sciences.com/tech/definitions/intelligence-artificielle-deep-learning-17262/

[Web 6] intelligence artificielle le pari de la reconnaissance visuelle. (s.d.). Récupéré sur carrementbechtle: http://carrementbechtle.com/index.php/2019/02/05/intelligence-artificielle-le-pari-de-la-reconnaissance-visuelle/

[Web 7] Réseau de neurones artificiels. (s.d.). Récupéré sur wikipedia: https://fr.wikipedia.org/wiki/R%C3%A9seau_de_neurones_artificiels#:~:text=Un%20r%C3%A9seau%20de%20neurones%20artificiels,est%20rapproch%C3%A9%20des%20m%C3%A9thodes%20statistiques.

[Web 8] Réseau de neurones récurrents. (s.d.). Récupéré sur wikipedia: https://fr.wikipedia.org/wiki/R%C3%A9seau_de_neurones_r%C3%A9currents

[Web 9] Réseaux de neurones convolutifs (CNN). (s.d.). Récupéré sur ichi.pro: https://ichi.pro/fr/reseaux-de-neurones-convolutifs-cnn-238106537306603

[Web 10] Classification des images médicales : comprendre le réseau de neurones convolutifs (CNN). (s.d.). Récupéré sur imaios: https://www.imaios.com/fr/Societe/blog/Classification-des-images-medicales-comprendre-le-reseau-de-neurones-convolutifs-CNN

[Web 11] reseau de neurones artificiels definition. (s.d.). Récupéré sur lebigdata: https://www.lebigdata.fr/reseau-de-neurones-artificiels-definition

[Web 12] reconnaissance faciale. (s.d.). Récupéré sur lebigdata: https://www.lebigdata.fr/reconnaissance-faciale-tout-savoir#:~:text=La%20reconnaissance%20faciale%20est%20une,dans%20une%20base%20de%20donn%C3%A9es.

[Web 13] Robotique. (s.d.). Récupéré sur wikipedia: https://fr.wikipedia.org/wiki/Robotique#:~:text=La%20robotique%20est%20l'ensemble,machines%20automatiques%20ou%20de%20robots.

[Web 14] bio-informatique. (s.d.). Récupéré sur biochimie.umontreal.ca: https://biochimie.umontreal.ca/etudes/bio-informatique/quest-ce-que-la-bio-informatique/

[Web 15] Chemoinformatique. (s.d.). Récupéré sur wikimonde: https://wikimonde.com/article/Ch%C3%A9moinformatique

[Web 16] Atome. (s.d.). Récupéré sur techno-science: https://www.techno-science.net/definition/3472.html

[Web 17] Ion. (s.d.). Récupéré sur futura-sciences: https://www.futura-sciences.com/sciences/definitions/chimie-ion-861/

[Web 18] Élément chimique. (s.d.). Récupéré sur futura-sciences: https://www.futura-sciences.com/sciences/definitions/chimie-element-chimique-15852/

[Web 19] Tableau périodique des éléments. (s.d.). Récupéré sur wikipedia: https://fr.wikipedia.org/wiki/Tableau_p%C3%A9riodique_des_%C3%A9l%C3%A9ments

[Web 20] Geometry of Molecules. (s.d.). Récupéré sur chem.libretexts.org: https://chem.libretexts.org/Bookshelves/Physical_and_Theoretical_Chemistry_Textbook_Maps/Supplemental_Modules_(Physical_and_Theoretical_Chemistry)/Chemical_Bonding/Lewis_Theory_of_Bonding/Geometry_of_Molecules?fbclid=IwAR123dzcHcHMccnlNBNNOSUqs9k_nrfDcFgT

[Web 21] Représentation des molécules. (s.d.). Récupéré sur techno-science.net: https://www.techno-science.net/glossaire-definition/Representation-des-molecules.html

[Web 22] Exercice : Lawsone et naphtoquinone. (s.d.). Récupéré sur wikiversity: https://fr.wikiversity.org/wiki/Mol%C3%A9cules_organiques_de_la_mati%C3%A8re_color%C3%A9e/Exercices/Lawsone_et_naphtoquinone

[Web 23] Représentation des molécules. (s.d.). Récupéré sur faidherbe.org: https://www.faidherbe.org/site/cours/dupuis/regcip.htm

[Web 24] Molecule. (s.d.). Récupéré sur en.wikipedia.org: https://en.wikipedia.org/wiki/Molecule

[Web 25] Découverte d'un nouveau type de liaison chimique ultra-forte. (s.d.). Récupéré sur futura-sciences: https://www.futura-sciences.com/sciences/actualites/chimie-decouverte-nouveau-type-liaison-chimique-ultra-forte-85110/

[Web 26] ionic-bond. (s.d.). Récupéré sur chemistrylearner: https://www.chemistrylearner.com/chemical-bonds/ionic-bond

[Web 27] Dénaturation des protéines. (s.d.). Récupéré sur biochim-agro.univ-lille: https://biochim-agro.univ-lille.fr/proteines/co/ch1_II_a.html

[Web 28] Molecule. (s.d.). Récupéré sur techno-science.net: https://www.techno-science.net/glossaire-definition/Molecule.html

[Web 29] Base de données chimiques. (s.d.). Récupéré sur wikipedia.org: https://fr.wikipedia.org/wiki/Base_de_donn%C3%A9es_chimiques

[Web 30] Chemical table file. (s.d.). Récupéré sur vvikipedla: https://bi.vvikipedla.com/wiki/Chemical_table_file

[Web 31] Protein Data Bank. (s.d.). Récupéré sur wikipedia.org: https://fr.wikipedia.org/wiki/Protein_Data_Bank

[Web 32] Chemical Markup Language. (s.d.). Récupéré sur wikipedia.org: https://fr.wikipedia.org/wiki/Chemical_Markup_Language

[Web 33] signal de localisation nucléaire. (s.d.). Récupéré sur owlapps.net: http://www.owlapps.net/owlapps_apps/articles?id=3226029

[Web 34] Wiswesser Line Notation. (s.d.). Récupéré sur openbabel.readthedocs.io: https://openbabel.readthedocs.io/en/latest/FileFormats/Wiswesser_Line_Notation.htm

[Web 35] international Chemical Identifier. (s.d.). Récupéré sur wikimonde: http://wikimonde.com/article/International_Chemical_Identifier

[Web 36] Simplified molecular input line entry specification. (s.d.). Récupéré sur chemeurope: https://www.chemeurope.com/en/encyclopedia/Simplified_molecular_input_line_entry_specification.html

[Web 37] bibliotheques python apprentissage automatique. (s.d.). Récupéré sur astuces-informatique: https://astuces-informatique.com/bibliotheques-python-apprentissage-automatique/

[Web 38] alpha edition. (s.d.). Récupéré sur docs.lattepanda: http://docs.lattepanda.com/content/alpha_edition/ide/

[Web 39] Your data science toolkit. (s.d.). Récupéré sur anaconda: https://www.anaconda.com/products/individual-b

[Web 40] jupyter. (s.d.). Récupéré sur jupyter.org: https://jupyter.org/

[Web 41] bibliothèques Python. (s.d.). Récupéré sur hebergementwebs: https://www.hebergementwebs.com/nouvelles/8-meilleures-bibliotheques-python-pour-l-apprentissage-automatique-et-l-intelligence-artificielle

[Web 42] scikit learn. (s.d.). Récupéré sur codecademy: https://www.codecademy.com/articles/scikit-learn

[Web 43] Python Imaging Library. (s.d.). Récupéré sur wikipedia.org: https://fr.wikipedia.org/wiki/Python_Imaging_Library

[Web 44] tensorflow. (s.d.). Récupéré sur tensorflow.org: https://www.tensorflow.org/?hl=fr

[Web 45] numpy. (s.d.). Récupéré sur datascientest: https://datascientest.com/numpy

[Web 46] matplotlib. (s.d.). Récupéré sur he-arc.github.io: https://he-arc.github.io/livre-python/matplotlib/index.html

[Web 47] os. (s.d.). Récupéré sur geeksforgeeks.org: https://www.geeksforgeeks.org/os-module-python-examples/

[Web 48] OpenCV. (s.d.). Récupéré sur cours-gratuit : https://www.cours-gratuit.com/cours-framework-java/cours-sur-le-traitement-d-images-avec-opencv

[1] Marref Nadia.apprentissage Incrémental & Machines à Vecteurs Supports.Université HADJ LAKHDAR – BATNA Faculté des sciences Département d'Informatique

[2] Zohra, D. F. (Année 2009/2010). Un modèle chimio-informatique pour une synthèse virtuelle. UNIVERSITE BADJI MOKHTAR-ANNABA.

[3] Amani, M. (2019/2020). La prédiction des familles de protéines en utilisant le réseau. UNIVERSITE BADJI MOKHTAR ANNABA.

[4] Roufaida, H. (2019/2020). *Classification de structures 2D/3D.* UNIVERSITÉ BADJI MOKHTAR ANNABA.