

UNIVERSITÉ DU QUÉBEC À MONTRÉAL

L'APPRENTISSAGE PONCTUEL EN PROFONDEUR APPLIQUÉ À LA DÉCOUVERTE DE  
MÉDICAMENTS

MÉMOIRE PRÉSENTÉ  
COMME EXIGENCE PARTIELLE  
DE LA MAÎTRISE EN GÉNIE ÉLECTRIQUE

PAR  
FATHI SOUAKER

OCTOBRE 2021

UNIVERSITÉ DU QUÉBEC À MONTRÉAL  
Service des bibliothèques

Avertissement

La diffusion de ce mémoire se fait dans le respect des droits de son auteur, qui a signé le formulaire *Autorisation de reproduire et de diffuser un travail de recherche de cycles supérieurs* (SDU-522 – Rév.04-2020). Cette autorisation stipule que «conformément à l'article 11 du Règlement no 8 des études de cycles supérieurs, [l'auteur] concède à l'Université du Québec à Montréal une licence non exclusive d'utilisation et de publication de la totalité ou d'une partie importante de [son] travail de recherche pour des fins pédagogiques et non commerciales. Plus précisément, [l'auteur] autorise l'Université du Québec à Montréal à reproduire, diffuser, prêter, distribuer ou vendre des copies de [son] travail de recherche à des fins non commerciales sur quelque support que ce soit, y compris l'Internet. Cette licence et cette autorisation n'entraînent pas une renonciation de [la] part [de l'auteur] à [ses] droits moraux ni à [ses] droits de propriété intellectuelle. Sauf entente contraire, [l'auteur] conserve la liberté de diffuser et de commercialiser ou non ce travail dont [il] possède un exemplaire.»



## REMERCIEMENTS

Je remercie chaleureusement mon directeur de thèse, monsieur Mounir Boukadoum qui m'a permis de choisir cet axe de recherche et de travailler sur le domaine de l'application de l'intelligence artificielle à la découverte des médicaments. Ses efforts louables, pour me soutenir, me conseiller et sa confiance ainsi que sa disponibilité m'ont permis d'accomplir le travail dans de bonnes conditions et d'aspirer à le valoriser parmi les meilleurs travaux dans ce domaine.

Je remercie particulièrement ma famille et surtout ma femme pour leur soutien et leur encouragement.

Je dédie ce travail à tous ceux qui adorent la science et offrent leur fruit de la recherche au bien-être de l'humanité.



## TABLE DES MATIÈRES

LISTE DES TABLEAUX .....	IX
TABLE DES FIGURES .....	X
RÉSUMÉ .....	XIII
ABSTRACT .....	XV
CHAPITRE I	
INTRODUCTION .....	1
1.1 Mise en contexte .....	1
1.2 Motivation .....	5
1.3 Problématique .....	5
1.4 Méthodologie proposée .....	6
1.5 Contribution .....	7
1.6 Organisation de l'ouvrage .....	8
CHAPITRE II	
Les réseaux de neurones artificiels .....	9
2.1 Introduction .....	9
2.2 Topologies des réseaux de neurones .....	10
2.2.1 Réseaux à propagation directe .....	10
2.2.2 Réseaux récurrents .....	11
2.3 Algorithmes d'apprentissage .....	12
2.3.1 Apprentissage supervisé .....	13
2.3.1.1 Les données .....	13
2.3.1.2 Modèle .....	14
2.3.1.3 Optimisation .....	15
2.3.4 Apprentissage en profondeur .....	15
2.4 Fonctions de sorties .....	16
2.5 Réseaux de neurones convolutifs .....	17
2.5.1 Architecture à convolutions notable .....	18
CHAPITRE III	
État de l'art sur l'apprentissage profond avec un nombre de données limitées. ....	20
3.1 Zero-shot learning .....	20
3.2 Apprentissage ponctuel et en $N$ coups .....	22
3.2.1 Réseaux correspondants (Matching Networks) .....	22
3.2.2 Réseaux Siamois .....	24
3.3 Conclusion .....	25

---

## CHAPITRE IV

L'intelligence artificielle et la découverte des médicaments .....	26
4.1 Choix des molécules .....	26
4.2 La Pénicilline .....	27
4.2.1 Les types de pénicilline et leurs structures moléculaires .....	28
A. Pénicilline de type V .....	28
B. Pénicilline de type G .....	29
C. Pénicilline de type A .....	29
4.3 L'acide mycophénolique .....	30
4.3.1 Les types d'acide mycophénolique .....	31
A. Le mycophénolate mofétil .....	31
B. Le mycophénolate de sodium .....	31
4.4 La Sulconazole .....	32
4.4.1 Les types de Sulconazole .....	33
A. La nitrate de sulconazole .....	33
C. Butoconazole .....	33

## CHAPITRE V

Architecture neuronale .....	35
5.1 Méthodologie .....	35
5.1.1 Apprentissage ponctuel.....	35
5.2 Inception.....	36
5.3 Incorporation de l'Inception.....	38
5.4 Architecture neuronale.....	39
5.4.1 Représentation des données.....	40
5.4.2 Entraînement du réseau .....	44
A. Fonction de perte Contrastive.....	44
B. Fonction d'entropie croisée binaire .....	47
5.4.3 Choix de la fonction perte .....	48
5.4.4 Ajustement du réseau de neurones .....	48

## CHAPITRE VI

Résultats et discussions .....	50
6.1 Validation des Résultats.....	50
6.2 Résultats.....	52
6.3 Évolution de l'entraînement .....	53
6.4 Comparaison avec un modèle CNN .....	58

VII	
6.5 Discussion des résultats.....	61

CHAPITRE VII	
CONCLUSION GÉNÉRALE.....	65
BIBLIOGRAPHIE.....	66

---

LISTE DES TABLEAUX

Tableau	Page
Résultat des simulations . . . . .	53
Résultats de la simulation avec un réseau siamois utilisant les CNN. . . . .	59
Comparaison avec d'autres travaux utilisant des réseaux siamois (meilleurs résultats). . . . .	62

## TABLE DES FIGURES

Figure	Page
2.1 Réseau de neurones multicouche . . . . .	10
2.2 Le perceptron unicouche. . . . .	10
2.3 Le perceptron multicouche. . . . .	11
2.4 Réseaux récurrents . . . . .	12
2.5 Apprentissage supervisé . . . . .	16
2.6 Fonctions d'activation . . . . .	17
2.7 Architecture du CNN . . . . .	18
3.1 Zero-shot learning . . . . .	22
3.2 Réseau correspondant . . . . .	24
3.3 One shot learning avec reseau Siamois . . . . .	25
3.4 Méthodes d'apprentissage utilisées en cas de manque d'informatio. . . . .	26
4.1 Structure générale de la Pénicilline . . . . .	29
4.2 Pénicilline de type V. . . . .	30
4.3 Pénicilline de type G . . . . .	30
4.4 Pénicilline de type A . . . . .	31
4.5 Structure générale de l'acide mycophénolique. . . . .	31
4.6 Structure générale de la Mycophénolate Mofetil . . . . .	32
4.7 Mycophénolate de sodium . . . . .	33
4.8 Structure générale du Sulconazole . . . . .	34
4.9 Nitrate de sulconazole . . . . .	34
4.10 Butoconazole . . . . .	35
5.1 Schéma de principe de l'Inception. . . . .	39
5.2 Architecture neuronale . . . . .	41
5.3 Structure moléculaire de la pénicilline . . . . .	42
5.4 Exemple de paire d'entraînement positive, utilisant la molécule de sulconazole et la molécule de nitrate de sulconazole . . . . .	43
5.5 Exemple de paire d'entraînement positive, utilisant la molécule de mycophénolate et la molécule de mycophénolate mofétil . . . . .	43
5.6 Exemple d'une paire d'entraînement négative, utilisant la molécule de mycophénolate de sodium) et la molécule de pénicilline G . . . . .	44
5.7 Exemple d'une paire d'entraînement négative, utilisant la molécule de mycophénolate mofétil et la molécule de butoconazole . . . . .	44
5.8 Exemple d'une paire d'entraînement négative, utilisant la molécule d'acide mycophénolate et la molécule de sulconazole . . . . .	45
5.9 Perte contrastive. . . . .	47

5.10	Algorithme d'entraînement de la fonction Perte Contrastive . . . . .	48
6.1	Évolution de la perte et de la précision du modèle avec entraînement de 25 paires pour la pénicilline . . . . .	54
6.2	Évolution de la perte et de la précision du modèle avec entraînement de 50 paires pour la pénicilline. . . . .	54
6.3	Évolution de la perte et de la précision du modèle avec entraînement de 25 paires pour la mycophénolate . . . . .	54
6.4	Évolution de la perte et de la précision du modèle avec entraînement de 50 paires pour la mycophénolate . . . . .	55
6.5	Évolution de la perte et de la précision du modèle avec entraînement de 25 paires pour la Sulconazole. . . . .	55
6.6	Évolution de la perte et de la précision du modèle avec entraînement de 50 paires pour la Sulconazole. . . . .	55
6.7	Évolution de la perte et de la précision du modèle avec entraînement de 5 paires pour la Pénicilline. . . . .	56
6.8	Évolution de la perte et de la précision du modèle avec entraînement de 10 paires pour la Pénicilline. . . . .	56
6.9	Évolution de la perte et de la précision du modèle avec entraînement de 5 paires pour la mycophénolate . . . . .	56
6.10	Évolution de la perte et de la précision du modèle avec entraînement de 10 paires pour la mycophénolate. . . . .	57
6.11	Évolution de la perte et de la précision du modèle avec entraînement de 5 paires pour la Sulconazole. . . . .	57
6.12	Évolution de la perte et de la précision du modèle avec entraînement de 10 paires pour la Sulconazole. . . . .	57
6.13	Architecture de réseau neuronal siamois utilisant des CNN. . . . .	58
6.14	Structure CNN pour le réseau neuronal siamois basé sur CNN. . . . .	58
6.15	Courbes de perte et de précision du modèle basé sur le CNN avec entraînement de 50 paires pour la pénicilline. . . . .	59
6.16	Courbes de perte et de précision du modèle basé sur le CNN avec entraînement de 50 paires pour la mycophénolate. . . . .	59
6.17	Courbes de perte et de précision du modèle basé sur le CNN avec entraînement de 50 paires pour la Sulconazole. . . . .	60
6.18	Courbes de perte et de précision du modèle basé sur le CNN avec entraînement de 25 paires pour la pénicilline. . . . .	60
6.19	Courbes de perte et de précision du modèle basé sur le CNN avec entraînement de 25 paires pour la mycophénolate. . . . .	60

XI

6.20 Courbes de perte et de précision du modèle basé sur le CNN avec entraînement de 25 paires pour la Sulconazole. . . . .	61
---	----

## RÉSUMÉ

Nous présentons une architecture neuronale qui exploite l'apprentissage ponctuel pour la découverte des médicaments. Elle peut prédire si une molécule candidate se qualifie pour une investigation plus approfondie en laboratoire pour devenir un médicament. Ceci est accompli en évaluant la similitude entre sa structure moléculaire et celle d'une molécule éprouvée qui sert de mémoire de référence, sans avoir besoin de nouvel entraînement pour de nouvelles classes. Le modèle utilise deux modules de traitement identiques configurés en réseau siamois pour créer des représentations internes des deux molécules, et une distance métrique évalue la proximité des sorties des deux modules avant la classification, conduisant à une architecture d'apprentissage légère et avec peu de paramètres à régler. Les expériences de validation en utilisant trois molécules de référence aux propriétés thérapeutiques différentes montrent une précision moyenne de 95,6% pour prédire si une molécule candidate est similaire à la référence. L'entraînement pour chacune des trois molécules a été accompli avec deux ensembles de 25 paires de molécules composés respectivement d'échantillons positifs et négatifs tirés au hasard de la base de données ChemPub. Chaque simulation a pris 43 minutes à compléter sur une station PC de performances moyenne, en l'absence d'une carte de GPU pour accélérer les calculs.

Mots clés— découverte des médicaments, apprentissage ponctuel, Réseau Siamois, Inception, CNN, distance Euclidienne, perte contrastive, fonction de perte croisée.

## ABSTRACT

We present a neural architecture that exploits one-shot learning for drug discovery. It can predict whether a candidate molecule qualifies for further laboratory investigations to become a drug. This is accomplished by assessing the similarity between its molecular structure and that of a proven molecule that serves as a memorized reference, without the need of retraining for new classes. The model uses two identical embedding modules configured in a Siamese network to create internal representations of the two molecules, and a distance metric assesses the proximity of the outputs of the two modules before classification, leading to a light learning architecture with few parameters to adjust. The validation experiments using three reference molecules with different therapeutic properties show an average accuracy of 95.6% in predicting whether a candidate molecule is similar to the reference. The training for each of the three molecules was carried out with two sets of 25 pairs of molecules composed respectively of positive and negative samples taken at random from the ChemPub database. Each simulation took 43 minutes to complete on an average performance PC station, in the absence of a GPU card to speed up the calculations.

Index Terms— drug discovery, one-shot learning, Siamese networks, Inception, CNN, Euclidian distance, contrastive loss, cross-entropy loss.

---



## CHAPITRE I

### INTRODUCTION

#### 1.1 Mise en contexte

Découvrir de nouveaux traitements pour les maladies humaines est un défi extrêmement compliqué, car tout médicament potentiel doit attaquer la source d'une maladie tout en satisfaisant des contraintes métaboliques et de toxicité restrictives. Traditionnellement, le processus de découverte prend des années à compléter, avec un taux d'échec élevé le long du chemin. Une tâche importante de ce processus est la recherche de candidats prometteurs parmi les millions de composés possibles. L'apprentissage automatique peut contribuer à accélérer considérablement la durée de cette quête.

La compagnie Google est l'un des intervenants qui travaillent à accélérer la découverte de nouveaux médicaments. En collaboration avec le laboratoire Pande de l'Université de Stanford, les chercheurs de Google ont présenté un document intitulé « massivement multitask Networks for Drug Discovery » qui examine comment l'utilisation de données provenant de sources différentes peut mieux déterminer quels composés chimiques serviront de traitements efficaces et deviendront des médicaments pour plusieurs maladies [1]. Bien que la publication en elle-même n'offre pas de percées médicales majeures, les auteurs ont pu montrer l'efficacité de l'apprentissage dit profond à utiliser de grandes quantités d'informations et accélérer la découverte de médicaments, en utilisant des réseaux de neurones artificiels capables d'apprendre à partir de bases de données auxquelles s'ajoutent de nouveaux éléments. Une conclusion encourageante de ce travail est que les modèles de l'apprentissage profond sont en mesure d'utiliser des données de nombreuses expériences différentes pour augmenter la précision de prédiction pour de nombreuses maladies. Malgré l'étendue de l'échelle expérimentale, les chercheurs ont pu tester la sensibilité des modèles utilisés à une variété de changements dans leur structure et dans les données d'entrée, ce qui a révélé des résultats intéressants quant à la performance, la sensibilité et la stabilité des modèles.

Aujourd'hui, la communauté scientifique oriente de plus en plus ses efforts de recherche vers l'intelligence artificielle (IA) afin d'accélérer la découverte de nouveaux traitements et de médicaments. Ainsi, Les membres de l'équipe Google Brain ont

conçu un système de vision informatique qui identifie la cristallisation des protéines avec une précision autour de 94% [2]. La cristallisation des protéines détermine la forme des cellules et peut jouer un rôle dans la découverte de médicaments pour traiter diverses maladies. Des centaines d'expériences sont généralement exécutées pour chaque protéine. La configuration du système informatique et l'imagerie sont souvent automatisées, mais la tâche de trouver des cristaux protéiques individuels est largement réalisée par l'inspection visuelle qui est lente et sujette à l'erreur humaine. Le problème peut être résolu par l'utilisation efficace de l'intelligence artificielle comme l'a montré l'équipe Google Brain.

L'Université de Toronto (Canada) et la startup américaine Insilico Medicine ont publié un autre résultat spectaculaire dans la revue Nature Biotechnology [3]. Grâce à une approche qui intègre deux techniques de l'IA, à savoir l'apprentissage par renforcement [4], [5] et les réseaux antagonistes génératifs (GAN [6], [7], [8], [9]), un nouveau médicament a été développé en 46 jours. De ce temps, 21 jours ont permis de générer 30 000 modèles de molécules qui ont donné naissance à six molécules dont deux ont été sélectionnés pour être testées sur des souris de laboratoire.

La prédiction computationnelle des interactions des médicament-cible [10] est devenue maintenant un outil important dans le domaine, grâce à l'automatisation du processus de recherche des molécules candidates à la validation en laboratoire, par opposition aux techniques classiques à caractère manuel utilisées dans les laboratoires de chimie humides. En adoptant cette approche, divers modèles ont été développés pour prédire les combinaisons potentielles des médicament-cible, en intégrant les algorithmes d'apprentissage automatique [11]. Par exemple, les réseaux de neurones profonds (DNN) [12] [13] ont ouvert la porte à de grandes applications pour le criblage de composés virtuels [14], avec le potentiel de devenir une approche standard dans la conception de médicaments [15].

En 2012, l'équipe gagnante du concours Kaggle sponsorisé par la compagnie Merck pour améliorer la précision de l'évaluation des propriétés moléculaires a utilisé un DNN multi-tâches pour progresser la précision relative de la prédiction des propriétés de 15% par rapport à l'algorithme de base de l'entreprise Merck fondé sur la technique de l'arbre décisionnel [16]. Depuis, plusieurs architectures DNN ont été proposées pour caractériser les structures moléculaires pour la découverte de médicaments. On peut citer Ma et al. [17] qui ont montré que les DNN peuvent faire de meilleures prédictions que d'autres algorithmes d'apprentissage automatique dans les modèles de relation quantitative structure-activité (quantitative structure– activity relationship : QSAR), mais à un coût de calcul plus élevé. Dans [1], Ramsundar et al. ont utilisé un réseau de neurones multi-tâches pour synthétiser les informations

provenant de nombreuses sources biologiques et créer un cadre de découverte de médicaments qui repose sur près de 40 millions de mesures sur plus de 200 cibles biologiques. Dans [18], Lin et al. démontrent que la précision prédictive de l'architecture multitâche est nettement meilleure que celle de l'approche à tâche unique, avec des performances croissantes à mesure que davantage de tâches sont ajoutées. Cependant, la transférabilité des connaissances vers de nouvelles tâches est limitée.

Dans une approche différente basée sur des graphes convolutifs, Duvenaud et al. [19] et Kearnes et al. [20] abordent les lacunes de la pratique des empreintes digitales pour souligner certains aspects de la structure moléculaire tout en ignorant les autres. Leur approche permet au modèle de prendre des décisions sur la base de toutes les données moléculaires, stimulant ainsi l'utilisation de la chimie informatique et de l'apprentissage automatique dans les applications de découverte de médicaments.

D'autres travaux utilisent l'architecture neuronale Inception [21] [22], où un réseau neuronal convolutif (CNN) [23] utilise des filtres de différentes tailles pour identifier des modèles dans une image d'entrée, et où une structure modulaire permet de construire de grands réseaux avec moins de paramètres à régler et une réponse plus rapide qu'avec d'autres architectures neuronales. En se basant sur ce principe, Goh et al. [24] proposent le réseau Chemception pour la prédiction de propriétés chimiques telles que la toxicité. À partir d'images de dessins 2D de molécules, ils ont pu atteindre une précision proche des modèles basés sur la physique, en utilisant des données d'apprentissage allant de 600 à 40 000 échantillons.

Plus récemment, Smith et Smith [25] décrivent un modèle de découverte de médicaments en relation avec la pandémie de Covid-19. Il repose sur le T-REMD restreint (dynamique moléculaire d'échange de répliques de température restreinte), avec un criblage virtuel à haut débit pour exploiter la similitude entre les virus SARSCoV-2 et les virus SARS-CoV. Leurs simulations partent de médicaments éprouvés contre le SRAS-CoV pour synthétiser des molécules qui pourraient être efficaces contre le nouveau coronavirus. Huit mille médicaments, métabolites, produits naturels et leurs isomères ont été utilisés pour établir une liste de sept composés les mieux classés qui étaient soit réglementés et approuvés en tant que médicaments, soit soumis à des études antérieures indiquant une utilisation thérapeutique.

Nonobstant l'avancée extraordinaire des méthodes de l'apprentissage automatique, le besoin demeure de trouver des modèles de classificateurs dont la performance rivalise avec celle des DNN classiques, qui demandent de grands ensembles de données pour l'entraînement, mais qui sont adaptés à des domaines pauvres en données comme

l'ingénierie des médicaments. Plusieurs approches ont été proposées à cet effet, dont les suivantes :

Santoro et al. [26] ont montré qu'en utilisant une architecture neuronale dont la capacité de mémoire est augmentée par des modules externes, comme le Neural Turing Machine (NTM) [27], on peut encoder et récupérer de nouvelles informations sans avoir besoin de réentraînement du réseau, et des prédictions précises peuvent être faites à partir de quelques exemples d'échantillons.

Zakharov et al. [28] ont effectué un méta-apprentissage sur un grand ensemble de données de type vidéo, ce qui leur a permis de réaliser un apprentissage sur un ou quelques exemples de modèles de tête parlante pour identifier de nouvelles personnes. En utilisant une architecture de réseau antagoniste génératif (GAN) [9], leur approche atteint une précision prédictive de 62%.

Gidas et Komodakis [29] utilisent un système d'apprentissage visuel pour apprendre de nouvelles catégories à partir de quelques données d'entraînement sans oublier les catégories initiales sur lesquelles il a été entraîné. Les auteurs proposent un générateur de classification basé sur le principe l'attention et redessinent ainsi le classificateur CNN pour obtenir une précision sur la reconnaissance des images de 56,20% et 73,00% sur l'ensemble de données Mini-ImageNet pour des entraînements sur un et cinq échantillons.

Koch et al. [30] développent le réseau neuronal siamois pour classer la similitude entre les paires d'entrée en fonction de la distance. Ils généralisent le pouvoir prédictif du modèle à la fois aux nouvelles données et aux nouvelles classes de distributions inconnues, conduisant à une précision de 92% sur le jeu de données Omniglot qui contient 1653 images de caractères dessinés à la main de 50 alphabets différents [31].

Puzanov et Cohen [32] proposent un algorithme d'apprentissage ponctuel en renforcement profond pour la classification. Il a été testé sur la base de données Omniglot et sur un ensemble de données réel pour la détection des intrusions, donnant une précision de 94% sur Omniglot et de 99% pour les intrusions malveillantes.

Une nouvelle architecture, le LSTM résiduel est combinée aux graphes convolutifs par Altae-Tran et al. pour améliorer la capacité à apprendre à partir de petites molécules avec des distances métriques [33]. En utilisant la distance cosinus, ils montrent que l'apprentissage ponctuel offre une forte amélioration de la précision par rapport aux méthodes standard telles que les forêts aléatoires, notamment dans le cas où le

nombre de données est réduit. Par exemple, ils atteignent une précision de 62% avec une seule paire de molécules utilisées pour l'entraînement.

Les méthodes d'apprentissage non supervisées [34] [35] ont également été utilisées dans la classification d'objets. Par exemple, l'approche bayésienne est appliquée à l'apprentissage ponctuel non supervisé des catégories d'objets dans Fei-Fei et al. [36]. L'efficacité de cette méthode est démontrée sur quatre catégories d'objets différents et le modèle a pu apprendre en utilisant 1 à 5 exemples d'entraînement avec une vitesse d'apprentissage inférieure à une minute et un taux d'erreur compris entre 8 et 22%.

Cependant, la plupart des travaux précédents offrent une précision insuffisante pour la découverte de médicaments où une performance de prédiction presque parfaite est requise, comme celle que peuvent atteindre les DNN avec de grands ensembles d'entraînement. Malheureusement, travailler avec de petits ensembles d'entraînement impacte considérablement la précision. Par exemple dans [36], elle diminue avec la réduction du nombre d'échantillons d'apprentissage et dans [33], elle n'atteint que 43,7% lors de l'utilisation d'une paire de molécules pour l'entraînement avec la méthode de convolution graphique appliquée aux troubles du système nerveux.

## 1.2 Motivation

L'intelligence artificielle offre des outils judicieux pour orienter les efforts de recherche et développement vers les molécules ayant un fort potentiel thérapeutique, grâce à son pouvoir prédictif et la puissance de l'apprentissage à partir d'expériences et de l'historique du processus de la découverte des médicaments. Ainsi, les essais virtuels menés par l'IA permettent d'économiser temps et l'argent en favorisant la préqualification des molécules avant de mener des expériences plus coûteuses in vivo en termes de toxicité et de solubilité. Néanmoins, la plupart de outils offerts demandent beaucoup de données pour l'entraînement des modèles. Il reste donc à développer de nouvelles approches de l'apprentissage automatique, ou d'améliorer une technique existante, pour la découverte de nouveaux médicaments, d'où la motivation du travail décrit dans ce mémoire.

## 1.3 Problématique :

La recherche de nouveaux médicaments à partir de petites molécules suit un long processus fait d'étapes d'identification et de filtrage. Il comprend les étapes suivantes :

1. Étude physiopathologique de la maladie qui sert à reconnaître la cible thérapeutique qui peut être un virus ou une hormone.
2. Inventaire des molécules capables d'avoir une activité suffisante sur la cible, appelé *hit génération* en anglais. Ensuite, les molécules sont sélectionnées selon des critères biophysiques, chimiques et industriels par le processus de l'identification des prospects pour isoler celles qui rencontrent le mieux les critères établis.
3. Optimisation des molécules sélectionnées. Les biochimistes créent des modèles selon les structures chimiques afin de concevoir les dérivées ou les isomères qui répondent au maximum de critères et ainsi éliminer les molécules sous-optimales. Cette étape prend 40% des coûts alloués à la recherche et développement.
4. Test des molécules in vivo sur des animaux en stade préclinique pour les qualifier de point de vue toxicité et pharmacologie. Enfin, les molécules considérées comme candidat médicament entre en phase d'essais cliniques sur les systèmes humains.

Cette procédure comprend deux inconvénients majeurs :

- La lenteur du processus, car il faut compter jusqu'à une quinzaine d'années avant qu'une molécule intéressante devienne un médicament [37],
- Le coût moyen de la recherche et développement pour mettre un médicament sur le marché est estimé présentement à environ 2,6 milliards de dollars [38].

Il est donc utile d'accélérer le processus en identifiant automatiquement au début des molécules ayant un potentiel thérapeutique pour la maladie d'intérêt.

#### 1.4 Méthodologie proposée

Notre approche consiste à chercher les similarités entre des molécules qui sont déjà des médicaments et des molécules candidates afin de favoriser les efforts de recherche et développement sur les molécules ayant le plus de chance de répondre aux exigences pharmaceutiques. Ce tri de candidats est susceptible de raccourcir considérablement le temps pour qualifier et certifier une molécule en tant que médicament, menant ainsi à une économie substantielle des budgets alloués au processus de conception.

Pour ce faire, nous utilisons l'apprentissage en profondeur qui permet à un ordinateur d'apprendre une tâche à accomplir à partir d'exemples et de généraliser ce qu'il a

appris à de nouveaux cas. Cependant, l'apprentissage en profondeur de base nécessite une grande quantité de données pour l'entraînement, et ces données sont rares dans le domaine de la découverte des médicaments. Il y a donc le risque fréquent du manque d'information sur les molécules d'intérêt et leurs propriétés.

Nous adaptons dans ce travail le paradigme d'apprentissage ponctuel [39], qui permet l'apprentissage en profondeur dans un contexte de carence de données d'apprentissage, au problème de découverte de médicaments. Nous proposons une architecture neuronale simple d'apprentissage en profondeur qui offre l'avantage de minimiser le nombre de paramètres impliqué dans la composition neuronale, réduisant ainsi la complexité de l'apprentissage et minimisant l'effort computationnel. Par ailleurs, l'apprentissage ponctuel permet d'ajouter de nouvelles molécules sans besoin de réentraînement comme cela est le cas avec d'autres architectures neuronales profondes.

L'architecture que nous proposons consiste à coder en entrée les images en deux dimensions des structures chimiques d'une molécule de référence et une molécule candidate par la méthode de l'Inception de Google, décrite plus en détail dans la suite de ce document, et calculer la distance euclidienne entre les deux images codées. Finalement la pertinence de la molécule candidate pour des analyses plus poussées en laboratoire humide est prédite sur la base du résultat. Le système développé est différentiable de bout en bout et, de ce fait, peut être entraîné par la méthode standard de descente du gradient [40].

### 1.5 Contribution

Le travail que nous proposons dans ce mémoire se veut un enrichissement dans le domaine de la découverte des médicaments. Cet axe de recherche devient de plus en plus intéressant en appariement avec l'intelligence artificielle. Face au problème de manque de données qui caractérise cette discipline, nous proposons une architecture neuronale qui permet l'utilisation de l'apprentissage en profondeur avec peu de données d'entraînement, en contra distinction des architectures profondes connues.

Notre contribution réside dans le jumelage de l'algorithme de l'Inception avec le paradigme de l'apprentissage ponctuel pour être appliqués aux petites molécules. Il sera montré que l'architecture neuronale obtenue, basée sur le concept de réseau siamois [30],[41] et [42], est capable d'apprendre la similitude entre deux entrées indépendantes avec aussi peu que 25 paires d'exemples de données utilisées pour l'entraînement.

## 1.6 Organisation de l'ouvrage

Le chapitre 1 introduit la découverte des médicaments et l'utilisation de l'intelligence artificielle dans ce domaine, ainsi que la problématique de ce projet de recherche. Le chapitre 2 donne une description générale des réseaux de neurones et leurs applications. Le chapitre 3 présente l'état de l'art des techniques utilisées en apprentissage automatique dans les cas d'une carence de données. Le chapitre 4 fait le point sur l'application de l'apprentissage profond dans le domaine de la découverte des médicaments et présente les molécules choisies pour tester l'efficacité de l'architecture neuronale développée. Le chapitre 5 présente l'approche d'apprentissage ponctuel développée. Le chapitre 5 décrit les expériences de simulation menées et les résultats obtenus, suivis d'une discussion. Finalement, le chapitre 6 conclut le mémoire en situant notre travail par rapport à des travaux similaires et donne des perspectives sur l'utilisation de l'intelligence artificielle dans la découverte des médicaments.

## CHAPITRE II

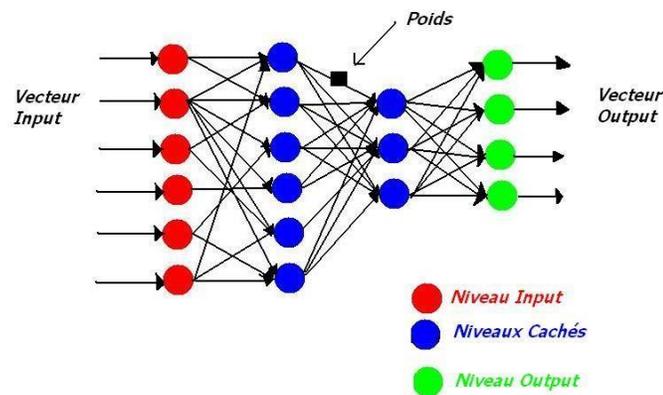
## LES RÉSEAUX DE NEURONES ARTIFICIELS

## 2.1 Introduction

Un réseau de neurones artificiel est un modèle de l'apprentissage automatique qui s'inspire schématiquement des neurones biologiques du cerveau humain. Il permet à une machine numérique, grâce à un ensemble de paradigmes représentés par des algorithmes, d'apprendre une tâche à partir de données fournies en exemples.

Dans sa configuration la plus populaire, le réseau de neurones apprend à partir d'exemples étiquetés. La reconnaissance d'objets est un bon exemple, où le réseau de neurones est entraîné à partir d'un grand nombre d'exemples de même type, tels les images d'un objet. Le réseau détecte les motifs récurrents dans les images qui lui sont présentées et apprend ainsi à classer de nouvelles entrées qu'il n'a pas vues auparavant.

Plusieurs architectures de réseaux de neurones artificiels existent, dépendant de la topologie, des fonctions de sorties des neurones et de l'algorithme d'apprentissage pour les entraîner. Le modèle le plus populaire est composé de couches de neurones successives, incluant une couche d'entrée, une couche de sortie et plusieurs couches intermédiaires dites couches cachées, qui sont connectées par des liens pondérés appelés liaisons synaptiques. Le réseau est dit profond lorsque le nombre couches cachées est élevé.



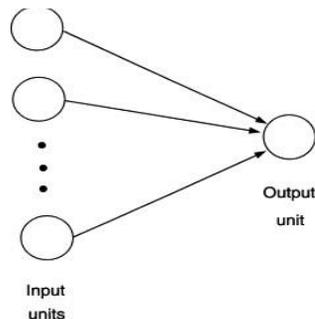
**Figure 2.1** Réseau de neurones multicouche

## 2.2 Topologies des réseaux de neurones

On peut classer les réseaux de neurones en réseaux à propagation directe et réseaux récurrents.

### 2.2.1 Réseaux à propagation directe

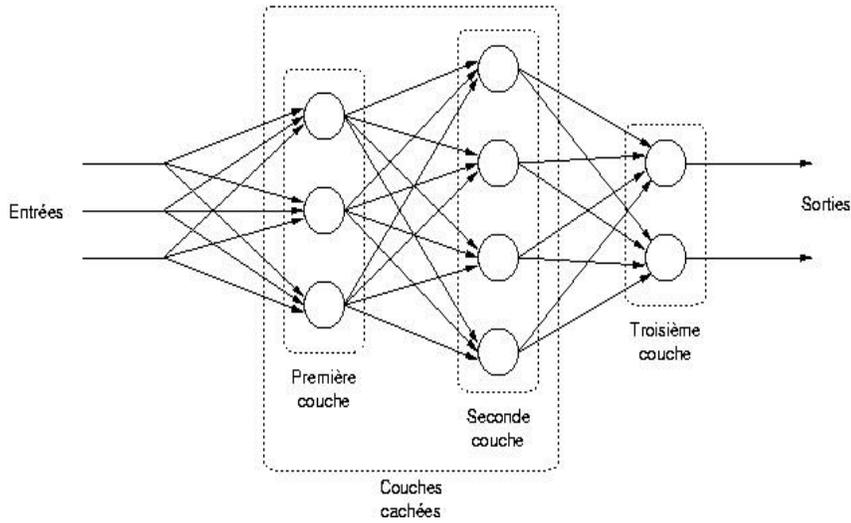
Dans les réseaux à propagation directe les signaux se propagent dans une seule direction de la couche d'entrée vers la couche de sortie, en traversant une ou plusieurs couches cachées. Ils sont quelquefois appelés perceptrons multicouches en référence au perceptron dont la structure est illustrée à la figure 2.2. Ce modèle d'origine ne possédait pas de couche cachée et était limité dans ses capacités [43].



**Figure 2.2** Le perceptron unicouche

Les perceptrons multicouches sont une version améliorée du perceptron avec la capacité d'apprendre des motifs non-linéairement séparables. Leurs entrées sont constituées des éléments à apprendre qui circulent à travers le réseau de la couche d'entrée vers la couche de sortie en passant par les couches cachées [44]. Dans l'apprentissage dit supervisé, l'entraînement du réseau est réalisé en repassant les mêmes entrées, qui utilisent le même chemin et activent les mêmes neurones jusqu'à l'atteinte de la couche de sortie où plusieurs neurones sont activés. La comparaison entre la sortie du réseau et celle souhaitée est enclenchée en respectant un processus d'optimisation. Les coefficients synaptiques qui lient les neurones sont modifiés afin d'avoir la meilleure sortie qui répond aux exigences établies.

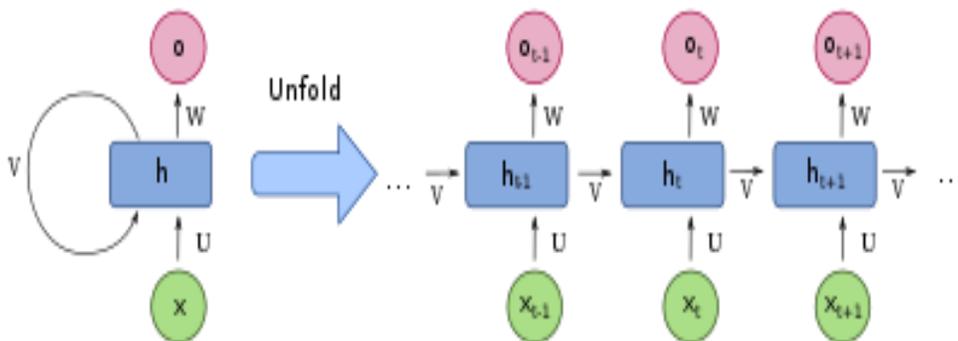
Le perceptron multicouche peut être utilisé pour une multitude d'applications comme la reconnaissance d'images.



**Figure 2.3** Le perceptron multicouche

### 2.2.2 Réseaux récurrents

Les réseaux récurrents (RNN) [40], [43] utilisent des boucles de rétroaction pour créer une mémoire de contexte comme qu'illustré dans la figure 2.4. Ils sont généralement dépliés à l'usage pour devenir des réseaux à propagation directe dont les couches cachées sont identiques.



**Figure 2.4** Réseaux récurrents

Ces réseaux présentent une faiblesse qui les rend inefficaces lorsqu'une mémoire à long terme est requise. Cela est dû au fait que chaque boucle fait intervenir une multiplication avec la sortie précédente, ce qui peut mener à l'implosion ou explosion du résultat à la longue. Dans le modèle d'apprentissage le plus populaire, la rétropropagation d'erreur par descente de gradient [45], cela se traduit par la dissipation progressive du gradient de l'erreur. Dans cette méthode, le réseau apprend à ajuster les poids des liens entre neurones en ajoutant à chaque valeur un terme qui dépend du gradient de l'erreur par rapport au poids considéré. Ce produit diminue à fur et à mesure qu'on se déplace de la couche de sortie vers la couche de l'entrée et l'ajustement des poids des premières couches devient défectueux à long terme.

### 2.3 Algorithmes d'apprentissage

Les réseaux de neurones apprennent par l'usage d'une structure et d'un algorithme et le changement de l'un ou de l'autre influence la qualité de l'apprentissage. En fait, le nombre de couche à travers lesquelles les données cheminent déterminent la profondeur du réseau et impactent sa précision.

Il existe plusieurs façons d'entraîner un réseau de neurones à réaliser la relation d'entrée-sortie désirée. On peut les classer les algorithmes d'apprentissage en trois grandes familles. Le premier type est l'apprentissage où le réseau dispose d'exemples d'entrées qui sont étiquetés avec les sorties souhaitées, ce qui permet de déterminer facilement l'erreur entre la sortie produite par le réseau et la sortie souhaitée et simplifie ainsi le processus de mise à jour des poids en cherchant le minimum du gradient de l'erreur.

Le second type est l'apprentissage non-supervisé, car le réseau arrive à catégoriser les données sans connaître leurs étiquettes au préalable. Dans l'apprentissage non supervisé, l'algorithme découvre des associations, des corrélations et des tendances sur des données non étiquetées, tel que les algorithmes de groupement (K-plus proches voisins, etc.)

Le troisième type est l'apprentissage par renforcement qui consiste à apprendre les actions à prendre en fonction des expériences et de la situation actuelle de l'environnement de façon à optimiser un enjeu quantitatif au cours du temps.

### 2.3.1 Apprentissage supervisé :

L'algorithme apprend à l'aide d'exemples d'entrées qui sont étiquetées avec les sorties souhaitées. Cette tâche d'apprentissage automatique vise à apprendre une fonction qui lie une entrée à une sortie en fonction des échantillons organisés en couple entrée-sortie, où l'entrée est généralement un vecteur et la sortie souhaitée est aussi quelquefois nommée signal de supervision. Un algorithme d'apprentissage supervisé utilise les données fournies et produit une fonction inférée qui peut être utilisée avec de nouvelles données exemples [46].

Les modèles sont ajustés par les données d'entraînement pour établir des prévisions sur des ensembles de test où seules les entrées sont fournies et les sorties sont comparées aux cibles fixées et l'écart est utilisé pour estimer la précision du modèle. L'apprentissage est une forme de recherche de l'hypothèse apte à donner les résultats escomptés, même à partir de nouveaux exemples complètement différents des données utilisées en apprentissage.

L'apprentissage supervisé utilise trois étapes : 1. préparation des données d'entraînement, 2. développement d'un modèle de réseau, 3. entraînement avec une fonction d'optimisation qui évalue la performance du réseau pour les données d'entraînement et ajuste les paramètres du réseau neuronal en conséquence.

#### 2.3.1.1 Les données :

Les ensembles d'apprentissage sont typiquement récoltés à partir d'une base de données et mis sous forme uniformisée dans un tableau unique  $D$  pour faciliter leur utilisation. On a :

$$D = \{(x_1, y_1), (x_2, y_2), \dots (x_n, y_n)\} \quad \text{Tel}$$

que :

$D$  est le tableau de données,  $n$

est la taille du tableau  $D$ ,

$x_i$  est le vecteur des caractéristiques de l'échantillon d'ordre  $i$ .

$y_i$  est l'étiquette de la variable de sortie associée à l'échantillon d'ordre  $i$ .

Les données sont prétraitées [47] pour simplifier leur exploitation en faisant un nettoyage qui élimine ou remplace les valeurs nulles ou manquantes et évite de dupliquer les mêmes informations. La mise à l'échelle est souvent utilisée pour les données numériques afin de faire face aux grandes variations de valeurs, et éviter un biais basé sur l'amplitude. Cela peut être fait par la normalisation par exemple. On a aussi recours à la transformation des données en leur donnant un format numérique approprié tel que la définition du nombre de pixel et le nombre de bits par pixel dans le traitement d'une image.

Pour exploiter convenablement les données, on les divise en échantillons d'entraînement qui servent à entraîner l'algorithme et échantillons de test qui servent à examiner si la performance voulue est atteinte. Généralement, on divise les données disponibles en 70-80% utilisées pour l'apprentissage et 20-30% utilisées pour tester la performance. Les éléments de chacun de ces ensembles sont sélectionnés par échantillonnage aléatoire. De plus, afin de s'assurer que les résultats ne sont pas le fruit du partage, on peut utiliser la validation croisée à  $n$  tours qui consiste à diviser le lot de données en  $n-1$  partitions destinées à l'apprentissage et la partition restante servant à la validation. Le processus est répété  $n$  fois avec, à chaque tour, une rotation pour renouveler les portions destinées à l'apprentissage et celle à la validation. Deux valeurs fréquentes  $n$  sont 3 et 10.

Une alternative à la validation croisée à  $n$  tours et le brassage aléatoire. Ce dernier consiste à choisir au hasard avec remise les échantillons pour les ensembles d'apprentissage et de validation. Ce processus est aussi répété  $n$  fois.

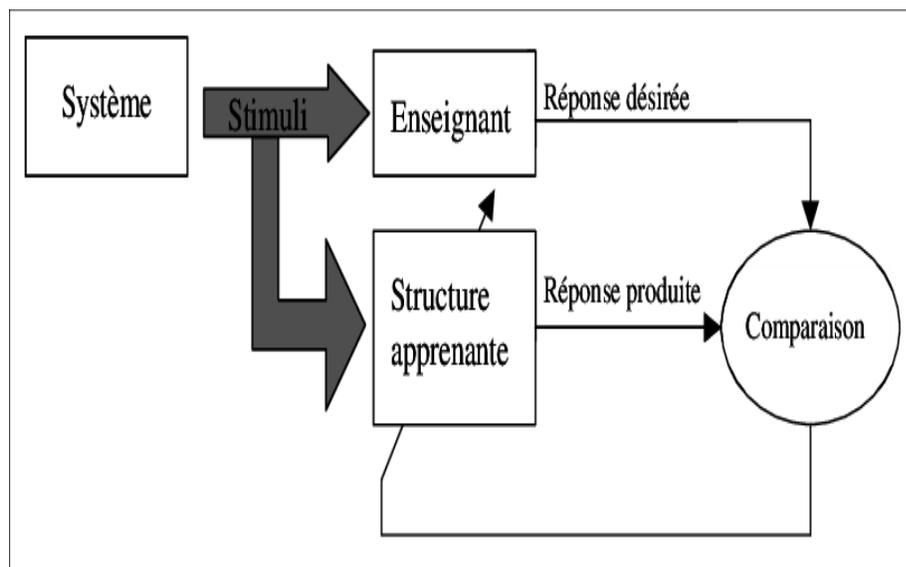
Dans le cas où le nombre d'échantillons est réduit, les données peuvent être traitées par la méthode de validation à une donnée exclue (leave-one-out en anglais). Elle consiste à valider en utilisant un seul échantillon et d'utiliser le reste pour l'entraînement. Le processus est répété pour tout l'ensemble de données.

#### 2.3.1.2 Modèle :

Une fois les données préparées, le système les utilise pour réaliser une relation de la forme  $y = f(x)$  tel que  $y$  est la sortie de prédiction à la suite des modifications subies par les échantillons à l'entrée. Le modèle peut être à propagation directe ou récurrent. Il utilise un algorithme pour régler les coefficients synaptiques entre les neurones du réseau en minimisant un critère d'optimisation afin de réaliser les tâches de regroupement, classification et la régression [40].

### 2.3.1.3 Optimisation :

À chaque fois que le modèle génère une sortie, elle est comparée à la sortie désirée afin de déterminer l'erreur entre les deux. Le rôle de l'optimisation est de réduire cette erreur et l'annuler dans un cas idéal. On minimise communément l'erreur quadratique pour éviter d'avoir une moyenne nulle en utilisant l'algorithme de descente de gradients [40].



**Figure 2.5** Apprentissage supervisé

### 2.3.4 Apprentissage en profondeur

L'apprentissage en profondeur est une catégorie de l'apprentissage automatique qui utilise l'un ou l'autre des types d'apprentissage précédents et permet de réaliser des architectures neuronales complexes qui contiennent un nombre important de couches aux fonctions diverses. Cela inclut les perceptrons multicouches avec couches convolutives, dits réseaux de neurones convolutifs (CNN), et les réseaux récurrents à mémoire à court et long-terme comme les modèles LSTM et GRU [40], [43].

L'apprentissage profond consiste donc à créer des réseaux multicouches avec des fonctions connexes ajoutées à l'architecture standard. Le but de l'ajout est souvent de déterminer automatiquement une hiérarchie des traits qui caractériseraient les

données, allant du bas niveau comme des segments élémentaires dans une image au haut niveau comme la détection des motifs complexes. L'apprentissage se fait à plusieurs niveaux d'abstraction et permet d'apprendre des fonctions complexes en utilisant les données seulement sans faire appel à l'être humain [48].

L'apprentissage en profondeur nécessite d'une façon générale de grandes quantités de données pour l'entraînement et une puissance de calcul importante pour le traitement. Il permet de réaliser différentes tâches tel que le regroupement, la classification et la régression. Ainsi, l'apprentissage profond permet de trier les données en fonction de leurs similitudes, de les classer en différentes catégories selon l'étiquetage des données et de faire de la prédiction.

#### 2.4 Fonctions de sorties

La fonction de sortie du neurone joue un rôle essentiel dans un réseau, en transformant les signaux qui se propagent entre les couches de manière non linéaire. Elle est connue aussi sous le terme "fonction d'activation" qui prend son origine de son analogue biologique le "potentiel d'activation", qui actionne la réponse du neurone lorsqu'un seuil est atteint [43].

Il existe plusieurs types de fonctions d'activation générant différentes valeurs de sortie pour représenter la réponse du réseau de neurones (voir la figure 2.6 pour des exemples). Le choix de la bonne fonction d'activation est important pour la performance d'un réseau. Dans les réseaux profonds, la fonction ReLU est très populaire.

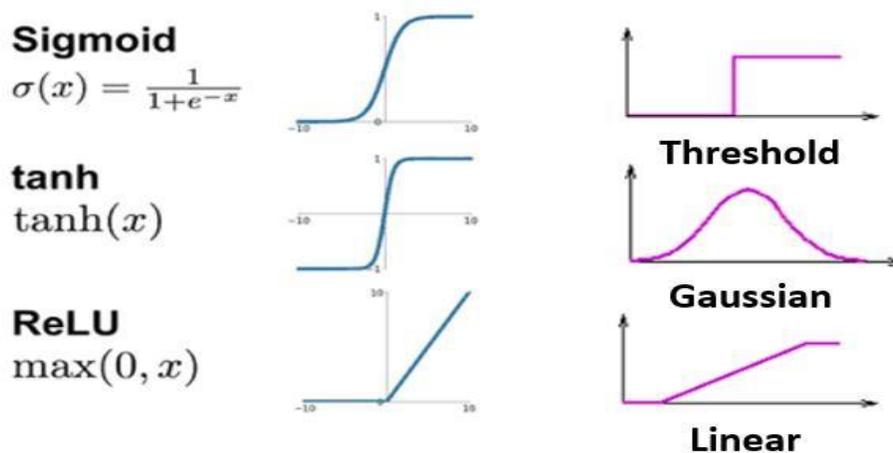


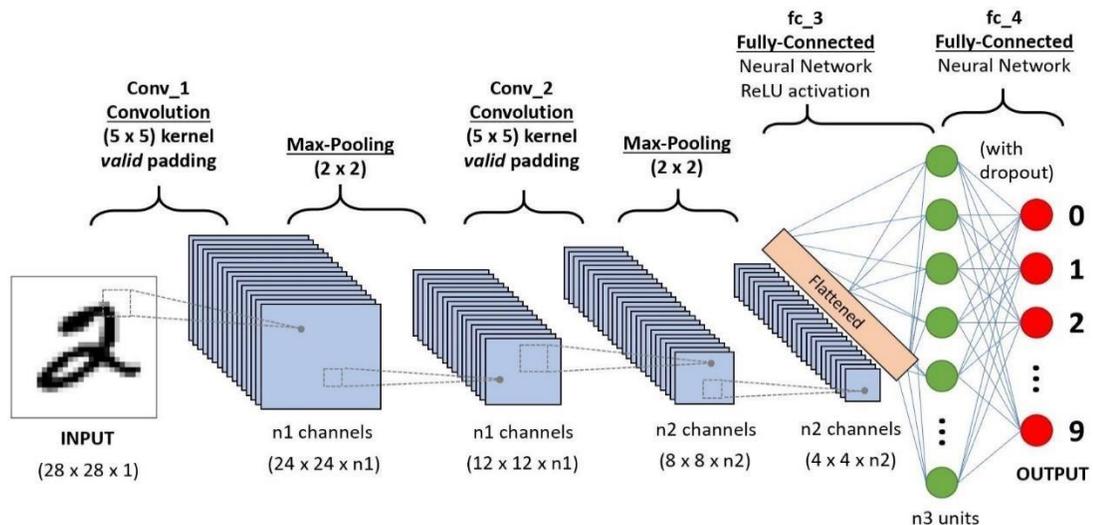
Figure 2.6 Fonctions d'activation

## 2.5 Réseaux de neurones convolutifs

Un CNN est construit d'une succession de couches qui incluent des couches de convolution, des couches de mise en commun et des couches complètement connectées. Il est l'exemple typique du réseau de neurones utilisé pour l'apprentissage en profondeur.

Le rôle de la convolution est d'extraire les caractéristiques du signal d'entrée par au moyen de filtres glissants qui donnent le produit de de convolution avec des éléments successifs du signal d'entrée. Le résultat obtenu est compressé pour réduire sa taille en préservant les caractéristiques principales par des méthode de mise en commun ou sous-échantillonnage. Le Max-pooling est l'une des méthodes communément utilisées pour réduire la taille du signal sans modifier son maximum. Le signal à la sortie des couches de convolution et de mise en commun est aplati sous la forme d'un vecteur qui alimente la couche complètement connectée. Cette dernière finalise l'architecture du CNN et elle est entièrement connectée aux neurones de sortie et utilisée pour optimiser des critères tel que le score de classification.

Le CNN convertit le signal d'entrée sous une forme exploitable pour à l'apprentissage profond en réduisant le coût du calcul [49].



**Figure 2.7** Architecture du réseau CNN

### 2.5.1 Architecture à convolutions notable

Les réseaux convolutifs (CNN) ont généralement une structure standard représentée par un empilement de couches convolutives suivi par une opération de mise en commun connue dans la littérature par le maxpooling [49]. Le réseau est organisé en plusieurs couches entièrement connectées. Les CNN ont connu une avancée importante grâce à la synergie entre l'architecture profonde et de la vision informatique classique, comme illustré par l'algorithme R-CNN de Girshick et al [50]. L'approche consiste à décomposer le problème global en deux sous-problèmes pour alléger les calculs. Il s'agit de déterminer les zones sensibles qui sont admissibles à contenir des objets et passer par la suite au réseau CNN pour identifier les catégories d'objets particulièrement dans les emplacements identifiés dans le sous problème de l'étape précédente. Une telle approche en deux étapes exploite la précision de la segmentation de des signaux, ainsi que le puissant pouvoir de classification des CNN à la pointe de la technologie.

L'influence de l'informatique mobile et des applications embarquées a encouragé la conception d'architecture profonde et de dépasser le cadre de la recherche d'une précision notable. Les nouvelles structures sont faciles à mettre en œuvre et utilisables à des fins réelles à un coût raisonnable. Notamment, l'architecture baptisée Inception qui permet un le traitement profond de l'image.

Dans ce concept les auteurs Christian Szegedy et all. ont introduit un nouveau niveau qui est la profondeur du réseau neuronal [21][51]. Les avantages de cette structure ont été vérifiées dans le défi de classification et de détection de l'ILSVRC (ImageNet Large Scale Visual Recognition Competition) organisé en 2014 où l'algorithme de l'Inception a dépassé de manière significative l'état de l'art [52].

Une nouvelle conception inspirée d'un modèle neuroscientifique du cortex visuel principal a été introduite par Serre Thomas en 2007 (IEEE Trans. Pattern Anal. Mach. Intell., 29(3):411–426, 2007) [53] où il a utilisé des séries de filtres de différentes tailles afin de gérer plusieurs échelles, à l'instar du modèle de l'Inception. De plus, les couches de l'Inception sont répétées plusieurs fois, ce qui conduit à un modèle profond pouvant traiter un volume important de données.

Les réseaux imbriqués (Network-in-Network) est une approche proposée par Lin en 2013 (Min Lin, Qiang Chen, and Shuicheng Yan. Network in network. CoRR, abs/1312.4400, 2013) [54] pour augmenter la puissance de représentation des réseaux neuronaux. Cette méthode peut être considérée comme l'ajout une convolution supplémentaire  $1 \times 1$  suivie de manière typique par la fonction d'activation linéaire ce qui lui permet d'être facilement intégré dans les réseaux de

type CNN. Cette approche est très utilisée dans l'architecture de l'Inception où les convolutions  $1 \times 1$  ont un double objectif, dans un premier temps, elles sont principalement utilisées comme modules de réduction des dimensions pour éliminer les goulots d'étranglement informatiques qui pourraient limiter la taille des réseaux, et en second lieu, cette façon de faire favorise non seulement l'augmentation de la profondeur, mais également la largeur des réseaux neuronaux sans affectation considérable des performances [55]. Nous allons exploiter la spécificité de l'Inception à réduire la taille du réseau en gardant l'indice de performance élevé pour concevoir notre architecture neuronale et le schéma détaillé sera exposé au chapitre 5.

L'apprentissage en profondeur est un outil prometteur pour résoudre les problèmes les plus complexes. Cependant, ses algorithmes classiques sont gourmands en données d'apprentissage et celles-ci sont rares dans plusieurs domaines, dont ceux de la découverte de médicaments, des diagnostics de maladies rares et de l'imagerie médicale en général. Ce problème est adressé au chapitre suivant qui résume les efforts à mettre au point des modèles d'apprentissage profond dans un contexte de manque de données.

## CHAPITRE III

### ÉTAT DE L'ART SUR L'APPRENTISSAGE PROFOND AVEC UN NOMBRE LIMITÉ DE DONNÉES

La classification d'objets avec peu d'exemples d'entraînement s'avère difficile, car le réseau neuronal peut les confondre et les considérer de la même classe comme dans le cas d'une molécule et ses isomères. Heureusement, les avancées récentes de l'intelligence artificielle créent la possibilité d'utiliser l'apprentissage en profondeur même avec des données limitées, et cela est aujourd'hui un des domaines de recherche les plus prometteurs [56] [57].

Par exemple, dans un article de l'équipe de Richard Berk de l'université de la Pennsylvanie intitulé « Modern Neural Networks Generalize on Small Data Sets » [58], ils considèrent que les réseaux de neurones profonds peuvent être décomposés. Plus précisément, les sous couches intermédiaires entre l'entrée et la sortie forment un système hiérarchisé et la couche de sortie finale offre un mécanisme pour générer le signal final du système global. Les ensembles de sous-réseaux sont à faible biais et relativement non corrélés. Les sous réseaux neuronaux profonds sont capables d'atteindre une précision de classification supérieure sans surajustement.

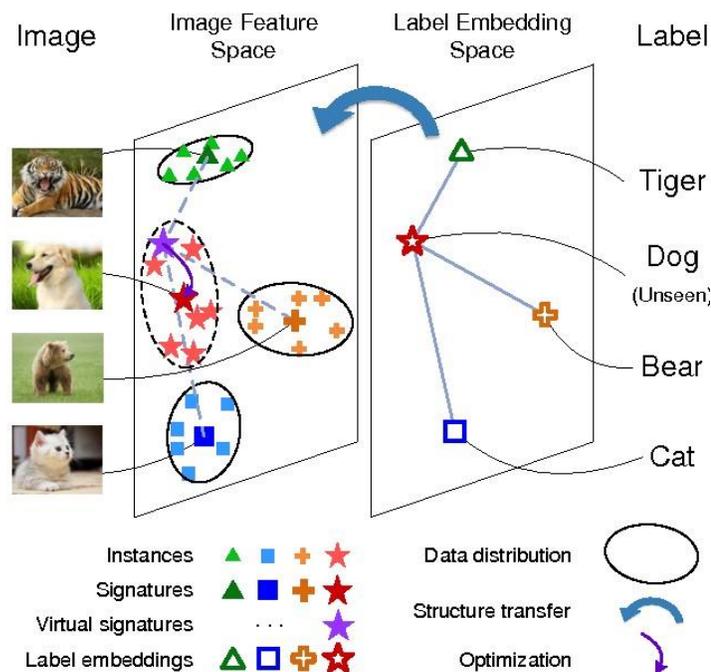
Les paradigmes utilisés en apprentissage profond dans un contexte de données limitées sont regroupés sous l'appellation d'apprentissage à données réduites. Ils utilisent les techniques d'apprentissage zéro coup, ponctuel et N coups.

#### 3.1 Zero-shot learning

La méthode d'apprentissage Zero-Shot réalise la tâche d'apprentissage sans besoin d'exemples explicites [56][57]. En fait, cela nous permet simplement de reconnaître des objets que nous n'avons pas vus auparavant.

Aksoy et al. ont produit un article intitulé Fine-Grained Object Recognition and ZeroShot Learning (ZSL) in Remote Sensing Imagery [59] qui est l'une des études pratiques intéressantes sur le sujet. Dans cet article, la reconnaissance et la classification des espèces d'arbres sont faites à l'aide de leurs images aériennes prises à partir d'un satellite. Une fois les données collectées, la phase d'apprentissage est

accomplie dans un contexte où on ne dispose d'aucun exemple d'entraînement ; à défaut d'apprendre à classer à partir d'échantillons d'images, la méthode ZSL le fait à partir de descriptions des classes. Deux représentations complémentaires des données sont utilisées par ZSL, la première étant l'image considérée comme caractéristique principale et la seconde étant les attributs de classes considérés comme représentation auxiliaire. Les étiquettes des classes et les images sont codées sous forme de plongements lexicaux (*vector embeddings*) qui permettent d'établir des liens sémantiques entre objets à travers la proximité de leurs vecteurs représentatifs. De cette façon, le réseau apprendra fondamentalement comment associer une image d'entrée à un vecteur contenant la description de la classe d'objet. Une fois la phase d'apprentissage terminée, lorsque le réseau recevra une nouvelle image d'un objet appartenant aux classes zéro-shot, il pourra générer un vecteur en sortie. Ensuite, en utilisant la distance entre ce dernier et tous les vecteurs décrivant les classes d'objets dont nous disposons et sur lesquels le réseau est déjà entraîné nous pourrons effectuer une classification.



**Figure 3.1** Zero-shot learning

### 3.2 Apprentissage ponctuel et en $N$ coups:

L'apprentissage ponctuel, connu sous l'appellation *one-shot learning* en anglais, est une autre approche d'apprentissage à classer des objets dans un contexte de manque de données d'apprentissage. Il se veut un outil d'apprentissage à partir d'un ou de quelques échantillons [39] [60]. De plus, de nouvelles classes peuvent être ajoutées sans avoir à requalifier le modèle [42] [61].

L'apprentissage ponctuel fait partie d'une classe d'algorithmes qui visent à apprendre à classer des objets en utilisant le moins d'exemples par catégorie que possible, idéalement un seul [36] [62]. Formellement, étant donné une requête  $x$ , le but est d'apprendre sa classe  $y$  à partir d'un ensemble de support fourni aussi petit que possible. Une façon d'y parvenir avec les réseaux de neurones est de réduire le nombre de paramètres libres [63].

Typiquement, étant donné  $K$  classes, chacune représentée par  $N_k$  exemples

$\{(\mathbf{x}_{ki}, \mathbf{y}_{ki})\}_{i=1, \dots, N_k}$  où  $\mathbf{y}_{ki}$  est l'étiquette de  $\mathbf{x}_{ki}$ , l'objectif est d'utiliser le support  $S$  obtenu de la cardinalité  $N = \sum_k N_k$  pour apprendre à déterminer quand une nouvelle entrée  $x$  est similaire à l'un des  $N$  exemples étiquetés stockés en mémoire. L'approche est appelée apprentissage en  $N$  coups et  $K$  façons ( *$N$ -shot- $K$ -way learning*), ou apprentissage en quelques coups (*few-shot learning*), pour des valeurs arbitraires de  $N$  et  $K$ , et simplement apprentissage ponctuel (*1-shot learning*) lorsqu'une seule classe et un exemple de référence sont considérés.

L'objectif de l'apprentissage en  $N$  coups est de classer les éléments d'un ensemble de requêtes  $Q$  étant donné le ou les exemples disponibles de chacune des classes  $K$ . Pour ce faire, les chercheurs ont développé plusieurs architectures de réseaux, notamment les réseaux correspondants et les réseaux siamois qui sont décrits à la section suivante.

#### 3.2.1 Réseaux correspondants (Matching Networks)

Cette architecture neuronale est l'une des premières à être entraînée et testée sur des tâches de type  $N$ -coups [64]. L'attrait de cette solution est simple, elle consiste à utiliser deux réseaux de neurones en parallèle, dont le premier code l'entrée  $x$  venant d'un ensemble de requête  $Q$  et le second les éléments  $\mathbf{x}_i$  d'un support  $S$ . Ensuite, une prospection est effectuée pour chercher la correspondance entre  $x$  et chacun des  $\mathbf{x}_i$ .

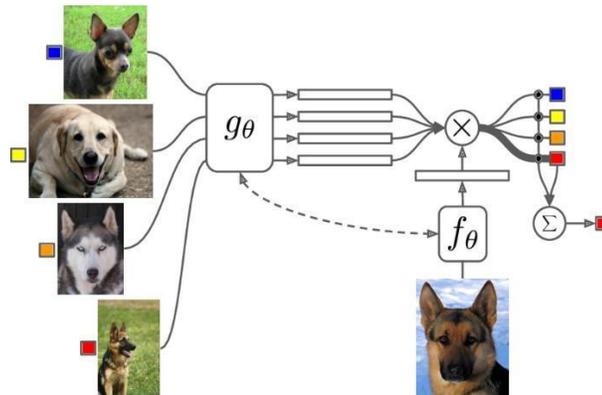
Ceci est possible grâce à une fonction d'attention  $a(\mathbf{x}, \mathbf{x}_i)$  qui détermine la proximité de la requête avec chaque élément de  $S$ . Finalement, un score moyen est établi pour  $\mathbf{x}$  par rapport à l'ensemble des éléments de  $S$ :

$$\alpha_s(\mathbf{x}) = \sum_{i=1}^N a(\mathbf{x}, \mathbf{x}_i) \cdot y_i$$

Et  $\mathbf{x}$  est attribué à la classe  $y_i$  qui est la plus proche du score. La fonction  $a(\mathbf{x}, \mathbf{x}_i)$  est définie telle que  $0 \leq a(\mathbf{x}, \mathbf{x}_i) \leq 1$  et  $\sum_{i=1}^m a(\mathbf{x}, \mathbf{x}_i) = 1$ , ce qui génère une distribution de probabilité par rapport aux différentes classes. Par conséquent, elle fournit une mesure relative de la façon dont la requête se rapporte à chaque élément de l'ensemble de support [64] [65].

Les réseaux correspondants [66] attribuent un score d'attention pour une entrée  $\mathbf{x}$  par rapport à chaque classe  $y_i$  par l'usage de la fonction Softmax. Cette dernière est appliquée à la distance cosinus entre  $\mathbf{x}$  et  $\mathbf{x}_i$ , et le résultat fournit une mesure relative de la façon dont la requête se rapporte à chaque classe, permettant ainsi une classification basée sur le rang. De ce fait, l'entrée  $\mathbf{x}$  est attribuée à la classe  $y_i$  dont la valeur est la plus proche du score obtenu après le calcul de  $\alpha_s(\mathbf{x})$ .

La figure ci-dessous résume le principe de fonctionnement du réseau correspondant.



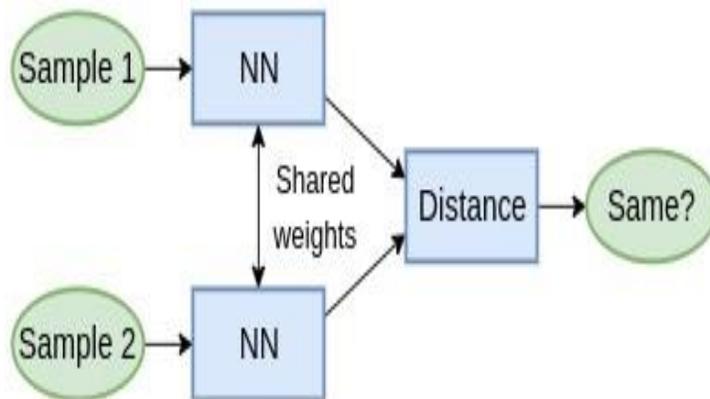
**Figure 3.2** Réseau correspondant

Les réseaux correspondants apprennent à détecter la similarité entre la requête et les éléments de l'ensemble de support par un mécanisme d'attention qui demande peu de données pour l'apprentissage.

### 3.2.2 Réseaux Siamois

Les réseaux siamois présentent un moyen de détecter la relation de similitude entre deux entrées indépendantes [67]. Ils ont été introduits pour la première fois par Bromley et al. [42] en 1993 pour vérifier l'authenticité d'une signature. Ces réseaux sont aussi utilisés pour réaliser l'apprentissage n-coups.

Le réseau siamois est composé de deux réseaux neuronaux identiques appelés réseaux jumeaux. Les deux opèrent en parallèle pour générer des représentations vectorielles de leurs entrées respectives. Ils comparent à la fin leurs sorties grâce à la distance euclidienne ou la distance cosinus [33]. Les réseaux siamois utilisent une vérification par paire pour effectuer un apprentissage métrique entre les sorties des deux réseaux jumeaux. Par la suite, dans une phase distincte, ils utilisent l'espace métrique appris pour effectuer la classification.

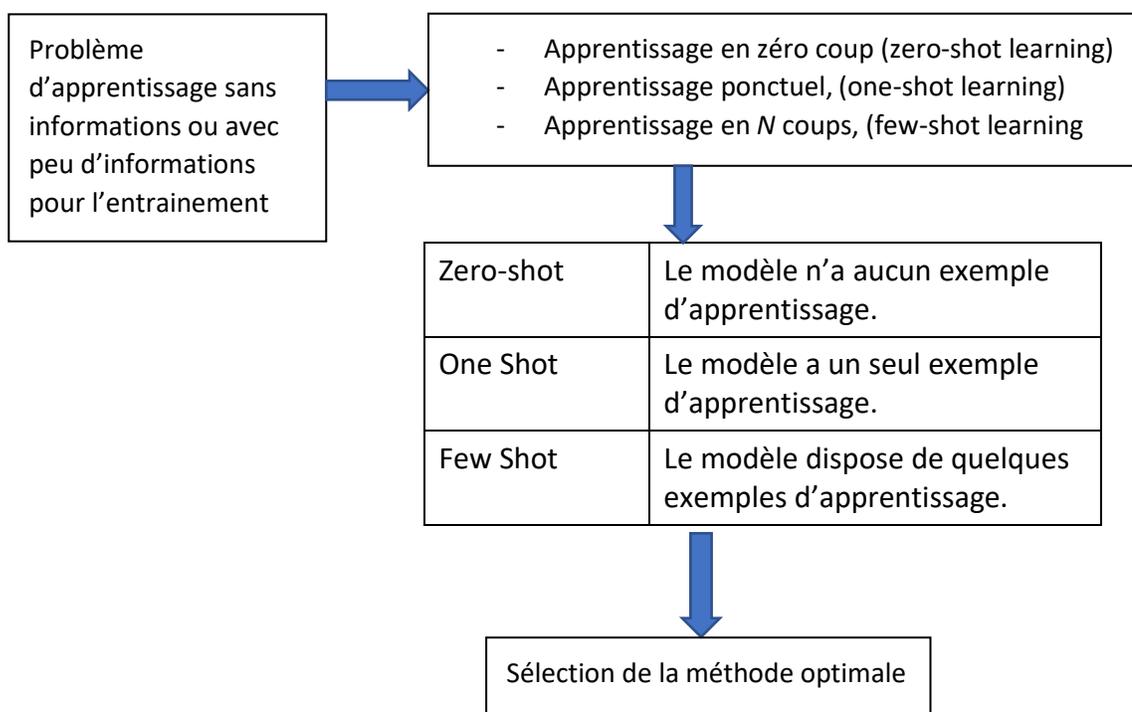


**Figure 3.3** One shot learning avec réseau siamois

Les réseaux siamois détectent la similarité entre la requête et l'élément de l'ensemble de support en utilisant une distance métrique qui demande peu de données lors de la phase d'apprentissage. Ainsi, au lieu d'attribuer des données aux classes comme le fait un DNN normal, l'apprentissage en un coup examine plutôt la similitude avec une référence (ou  $N$  références dans l'extension en  $N$  coups).

### 3.3 Conclusion

Pour faire face au problème de la rareté de données, les chercheurs en I.A. ont développé plusieurs approches, dont celles présentées dans ce chapitre et résumées à la figure 3.4. L'idée derrière tous ces paradigmes est d'entraîner les modèles pour distinguer entre les entrées similaires et dissimilaires et de généraliser l'apprentissage pour évaluer de nouvelles catégories. Le progrès consenti à développer ces approches a permis à la communauté scientifique d'avoir des niveaux de performance d'apprentissage.



**Figure 3.4** Méthodes d'apprentissage utilisées en cas de manque d'informations

Étant donné le nombre et la qualité des échantillons, et le fait de les introduire en paires dans l'étape de traitement, cela nous a conduit dans ce travail à choisir la méthode de l'apprentissage ponctuel, en développant une structure neuronale adaptée au problème de la découverte des médicaments. Elle sera présentée en détail au chapitre V du mémoire et validée avec les molécules décrites dans ce qui suit.

## CHAPITRE IV

### L'INTELLIGENCE ARTIFICIELLE ET LA DÉCOUVERTE DES MÉDICAMENTS

L'ingénierie des médicaments cherche activement à éviter le scénario catastrophe de découvrir les effets indésirables d'une substance après y avoir investi des années de travail. Les statistiques de la recherche et développement dans le domaine pharmaceutique indiquent que 40 % des candidats-médicaments atteignent le stade des tests sur les humains pour échouer ensuite à cause de problèmes de toxicité [68]. C'est dire l'importance d'identifier les bonnes molécules candidates pour la réussite du processus d'invention de nouveaux médicaments.

#### 4.1 Choix des molécules

Pour ce travail de recherche, on a choisi des petites molécules ayant des structures chimiques différentes et des domaines d'application distincts. Notre choix est guidé par le désir de développer une architecture neuronale capable de reconnaître les similitudes entre les molécules malgré leur différence et d'élargir son champ d'application à plusieurs utilisations pharmaceutiques. Il faut éviter d'entraîner notre réseau de neurones sur une structure chimique unique ce qui risque d'ajuster ses paramètres sur la reconnaissance d'une seule forme et d'ignorer par conséquent les autres compositions chimiques qui peuvent aussi être utiles.

Les molécules que nous avons visées pour tester notre approche sont la pénicilline, l'acide mycophénolique et le sulconazole. Leur structure chimique sont différentes et chacune d'elles a plusieurs molécules semblables qui diffèrent par un ou plusieurs éléments chimiques qui ont été rattachés à la molécule originale pour aboutir sur un nouveau médicament.

Les molécules choisies sont utilisées pour différents objectifs médicaux. La Pénicilline est un antibiotique et ses molécules semblables ont le même effet, mais sont administrés de différentes façons pour faire face à diverses maladies [69].

De même, la molécule de l'acide mycophénolique ainsi que les molécules semblables sont utilisées pour la prévention du rejet de la greffe d'un organe transplanté [69].

L'utilisation d'une molécule ou l'autre dépend de l'organe dont on veut lui éviter le rejet.

En dernier lieu, le sulconazole et ses molécules dérivées sont des antifongiques utilisés pour le traitement de la peau et le choix d'une structure chimique ou l'autre dépend de la nature de l'affection cutanée [69].

Ainsi, notre choix couvre différentes applications médicales et pharmaceutiques allant de l'antibiotique dont l'usage peut être externe sur la peau ou interne, à l'antifongique à usage externe en finissant par un médicament à usage purement interne utilisé pour éviter le phénomène de rejet après à une transplantation d'organes.

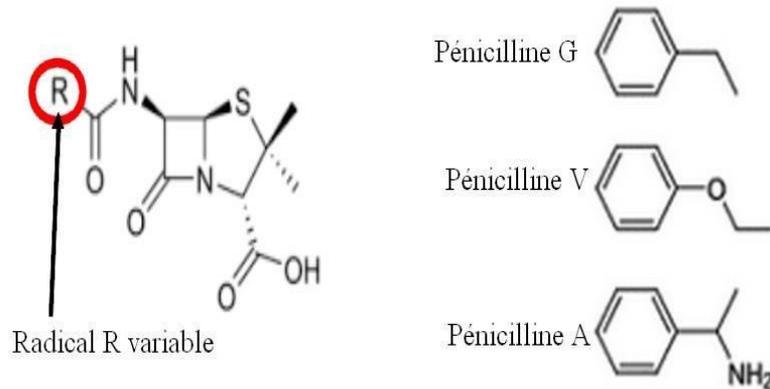
Pour chaque molécule, on présentera la structure chimique originale ainsi que deux molécules semblables pour donner une idée sur la variation des éléments chimiques entre les molécules similaires [70].

#### 4.2 La pénicilline

Ce mot générique réfère à une famille de molécules ayant des propriétés antibiotiques. La molécule de base se lie à certaines protéines et des enzymes catalysent la synthèse du peptidoglycane, qui est un composant essentiel de la paroi cellulaire bactérienne. Cela conduit à l'interruption de la synthèse de la paroi cellulaire, entraînant par conséquent une inhibition de la croissance cellulaire bactérienne et une lyse cellulaire [69].

La structure chimique générale d'une molécule de pénicilline est  $C_{10}H_{13}N_2O_4S-R$ . Elle comprend un cœur composé de dix atomes de carbone, treize atomes d'hydrogènes, deux atomes d'azotes, quatre atomes d'oxygène, 1 atome de soufre, plus un radical R qui représente les différentes protéines qui peuvent être combinées avec le cœur pour passer d'une variante à l'autre. On greffe la protéines adéquate pour diversifier le spectre d'action afin de neutraliser les bactéries nuisibles.

**Radical R variant selon le type de pénicilline :**



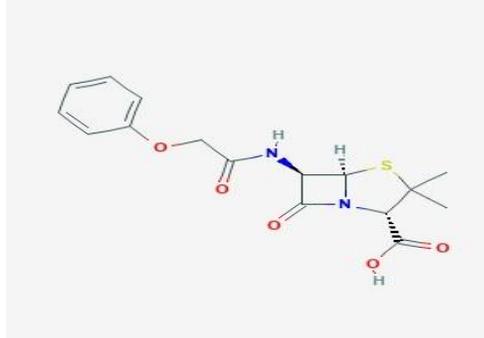
**Figure 4.1** Structure générale de la Pénicilline

#### 4.2.1 Les types de pénicilline et leurs structures moléculaires

La structure d'un antibiotique n'est pas unique et peut être ramifiée pour donner naissance à plusieurs familles en fonction du spectre d'utilisation, comme par exemple les bêtalactamines dans le cas de la Pénicilline. Dans ces familles, il existe aussi des sous-familles, comme les Pénicilline de type G, de type A et V [69]. Ainsi, les pénicillines de type G ont un champ d'activité moins étendu que les pénicillines de type A qui sont parfois associées à d'autres éléments chimiques tels que l'acide clavulanique qui leur permet de mieux résister face à certaines bactéries [69].

##### A. Pénicilline de type V

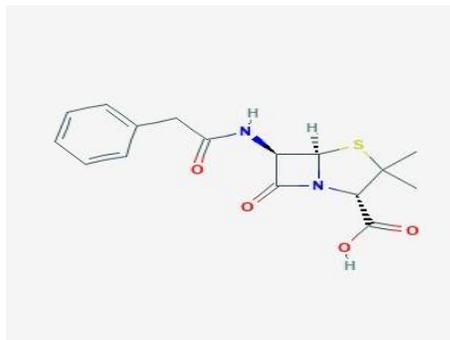
La pénicilline V est la forme orale de la pénicilline. C'est un antibiotique à spectre étroit utilisé pour traiter les infections légères à modérées causées par des bactéries sensibles. Cet antibiotique naturel peut également être utilisé dans certains cas comme prophylaxie contre les organismes sensibles. Les pénicillines naturelles sont considérées comme les médicaments de choix pour plusieurs infections causées par des organismes aérobies à Gram positif [69]. Son élimination est urinaire [69].



**Figure 4.2** Pénicilline de type V

#### B. Pénicilline de type G

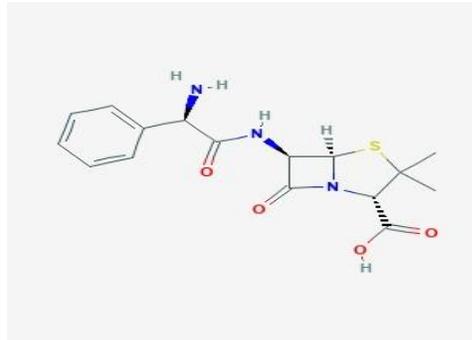
La pénicilline G est la forme parentérale, intraveineuse ou intramusculaire, de la pénicilline. Elle est utilisée pour des infections sévères où l'on ne peut se servir de la forme orale de la pénicilline. Sa molécule diffère de celle de la pénicilline V par le remplacement du groupe phénoxy méthyle par un groupe benzyle. Elle possède le même spectre d'action que la pénicilline V [69].



**Figure 4.3** Pénicilline de type G

#### C. Pénicilline de type A

Les pénicillines A sont des aminopénicillines utilisées dans les infections à spectre large. L'ampicilline et, surtout, l'amoxicilline dont l'effet est plus rapide sont ses deux principaux représentants selon [69].

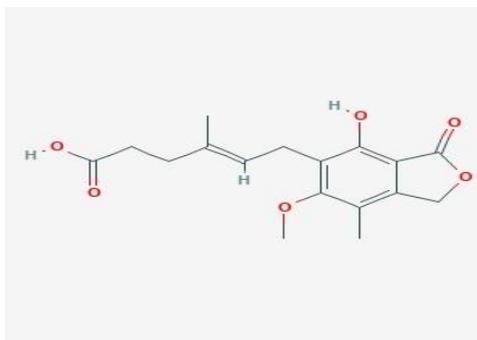


**Figure 4.4** Pénicilline de type A

Nous remarquons que dans la famille des pénicillines, les structures chimiques des différents éléments diffèrent les unes des autres tout en gardant un cœur commun.

#### 4.3 L'acide mycophénolique

L'acide mycophénolique est un agent immunosuppresseur, indiqué en transplantation contre le rejet d'organe. Il agit comme le métabolite actif, par hydrolyse *in vivo*. Il se présente essentiellement sous la forme de mycophénolate de sodium et de mycophénolate mofétil. La structure chimique générale de la molécule de l'acide mycophénolique est  $C_{17}H_{20}O_6$ .

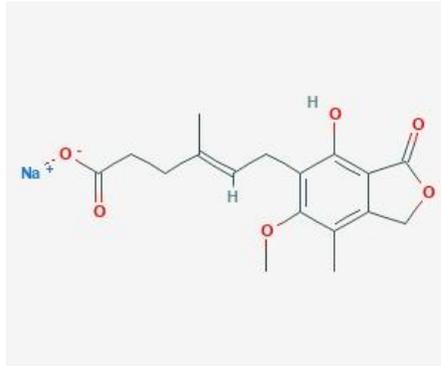


**Figure 4.5** Structure générale de l'acide mycophénolique.



des tissus suite à des greffes d'organes et pour le traitement de certaines maladies auto-immunes.

Formule chimique :  $C_{17}H_{19}NaO_6$



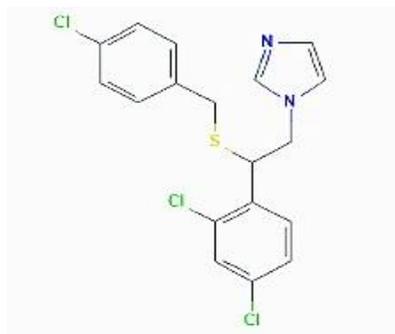
**Figure 4.7** Mycophénolate de sodium

Nous remarquons que dans la famille de l'acide mycophénolique, les structures chimiques des différents éléments diffèrent les unes des autres tout en gardant une grande ressemblance au niveau des liaisons carbone hydrogène et oxygène.

#### 4.4 La Sulconazole

La sulconazole est un médicament antifongique à large spectre de la classe des imidazoles. Il est destiné à traiter les infections cutanées telles que le pied d'athlète, la teigne, l'eczéma. La molécule de sulconazole est dérivée du cycle imidazole ( $C_3N_2H_4$ ), en remplaçant deux atomes d'hydrogène par des groupes dichlorophényle et sulfanediyle [69].

Formule chimique :  $C_{18}H_{15}Cl_3N_2S$



**Figure 4.8** Structure générale du Sulconazole

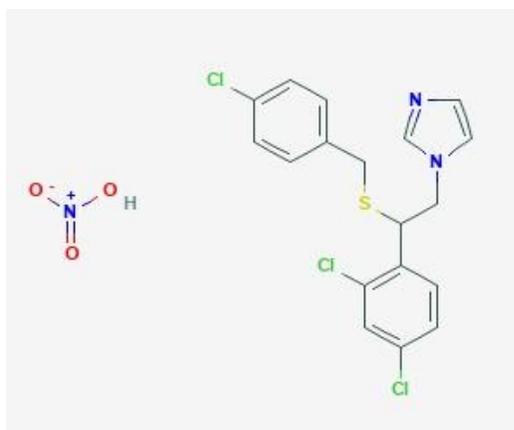
#### 4.4.1 Les types de Sulconazole

Il y a plusieurs médicaments à base de sulconazole dont la structure chimique diffère selon leur utilisation pharmaceutique. On remarque que les molécules sont composées de liaisons entre carbone-hydrogène-carbone en plus d'un élément chimique additionnelle ce qui confère aux molécules un certain degré de ressemblance.

##### A. La nitrate de sulconazole

La nitrate de sulconazole possède un large spectre d'activité in vitro contre les dermatophytes et certaines bactéries [69].

Formule chimique :  $C_{18}H_{16}Cl_3N_3O_3S$



**Figure 4.9** Nitrate de sulconazole

##### B. Butoconazole

Le nitrate de Butoconazole est un dérivé imidazolé doué d'une activité antifongique et antibactérienne. Cette activité a été démontrée in vitro et s'exerce sur les agents responsables de mycoses cutanéomuqueuses [69]. Cette substance dérivée de la sulconazole a pour formule chimique :  $C_{19}H_{17}Cl_3N_2S$



**Figure 4.10** Butoconazole

Nous remarquons que les structures chimiques des isomères de la Sulconazole, diffèrent les unes des autres tout en gardant une grande similarité dans les liaisons carbone hydrogène et chlore.

Les structures moléculaires qu'on a répertoriées seront traitées par l'architecture neuronale qui sera présentée au prochain chapitre.

## CHAPITRE V

### ARCHITECTURE NEURONALE

Pour rappel, nous nous intéressons dans ce travail à une architecture neuronale qui permet au réseau développé d'apprendre à partir de quelques exemples tout en préservant les objectifs de la conception de nouveaux médicaments.

Nous nous inspirons pour cela des modèles neuronaux Inception [21] [22] et siamois [30] [42] pour créer une architecture légère et avec peu de paramètres à régler, et nous prenons trois molécules différentes comme exemples pour montrer l'efficacité de l'architecture de réseau proposée à prédire la qualification d'une molécule candidate pour devenir potentiellement un médicament, en utilisant seulement deux ensembles de 25 données étiquetées pour l'entraînement.

Les sections suivantes fournissent un résumé des travaux connexes effectués dans les domaines de l'apprentissage ponctuel et de la découverte de médicaments, le contexte méthodologique sur les techniques et enfin l'architecture neuronale proposée pour déterminer les similitudes entre les molécules candidates et les médicaments.

#### 5.1 Méthodologie :

Cette section débute par le choix de la méthode pour réaliser l'apprentissage ponctuel. Ensuite, elle décrit le module inspiré du réseau Inception qui réalise le traitement des représentations moléculaires en 2D des entrées et finit par la présentation des différentes parties de l'architecture neuronale utilisée pour la classification des paires des molécules.

##### 5.1.1 Apprentissage ponctuel :

Pour réaliser l'apprentissage ponctuel sur les structures des molécules choisies dans le chapitre précédent, on a opté pour les réseaux siamois, car ils nous offrent la possibilité de jumeler la molécule cible à celle qui lui ressemble le plus grâce à un critère simple de distance Euclidienne. Ainsi deux réseaux de neurones jumeaux opèrent ensemble sur les paires  $(x, x_i)$  de manière que le premier réseau produit la

représentation  $f(\mathbf{x})$  pour la requête  $\mathbf{x}$  et le deuxième réseau neuronal produit la représentation  $g(\mathbf{x}_i)$  pour chaque élément  $\mathbf{x}_i$  de l'ensemble de support. La distance entre les deux sorties est alors définie comme :

$$D = \|f(\mathbf{x}) - g(\mathbf{x}_i)\|^2 \quad (5.1)$$

L'équation 5.1 exprime une manière simple d'effectuer une mesure de similarité, la valeur de  $D$  diminuant lorsque  $f(\mathbf{x})$  et  $g(\mathbf{x}_i)$  se rapprochent et augmentant lorsqu'ils deviennent plus éloignés. Si  $f(\mathbf{x})$  et  $g(\mathbf{x}_i)$  sont des normes unitaires, l'équation 5.1 se réduit essentiellement à l'approche de distance cosinus utilisée dans les réseaux d'appariement, avec moins d'effort de calcul. À cet égard, la distance métrique peut être considérée comme une forme d'attention élémentaire.

Nous avons conçu un système d'apprentissage ponctuel composé de deux modules Inception identiques (décrits ci-dessous) dont la distance euclidienne entre les sorties est calculée et utilisée pour générer un résultat de similarité binaire.

Cette approche pallie au besoin d'augmenter la taille du réseau en profondeur et en largeur afin d'améliorer sa performance, ce qui augmente le nombre de paramètres à régler au détriment des petits ensembles d'apprentissage, et augmente la complexité computationnelle suivant une loi de puissance. En effet, les modules Inception sont suffisamment simples pour contourner les deux exigences précédentes et suffisamment complexes pour permettre un entraînement adéquat du réseau si comme nos résultats de simulations le montrent.

## 5.2 Inception

L'idée principale de l'architecture Inception est de construire un réseau par optimisations locales successives qui suivent le même modèle. La structure d'un réseau de neurones Inception est donc modulaire et il suffit de trouver la construction locale optimale et de la répéter spatialement pour construire le réseau. Chaque module comprend des blocs de convolutions à résolution variable auxquels s'ajoute un chemin de mise en commun (maxpooling) pour réduire la taille de l'information reçue de l'étage précédent.

L'avantage principal de cette architecture est qu'elle permet d'augmenter le nombre d'unités à chaque étape de manière significative tout en maintenant le contrôle sur la capacité computationnelle et éviter les divergences causées par la complexité du calcul grâce à sa structure décentralisée. Un autre aspect pratique de cette conception est qu'elle s'aligne sur l'intuition que les informations visuelles doivent être traitées à

différentes échelles, puis agrégées de manière que la prochaine étape peut abstraire des caractéristiques de différentes échelles simultanément [21].

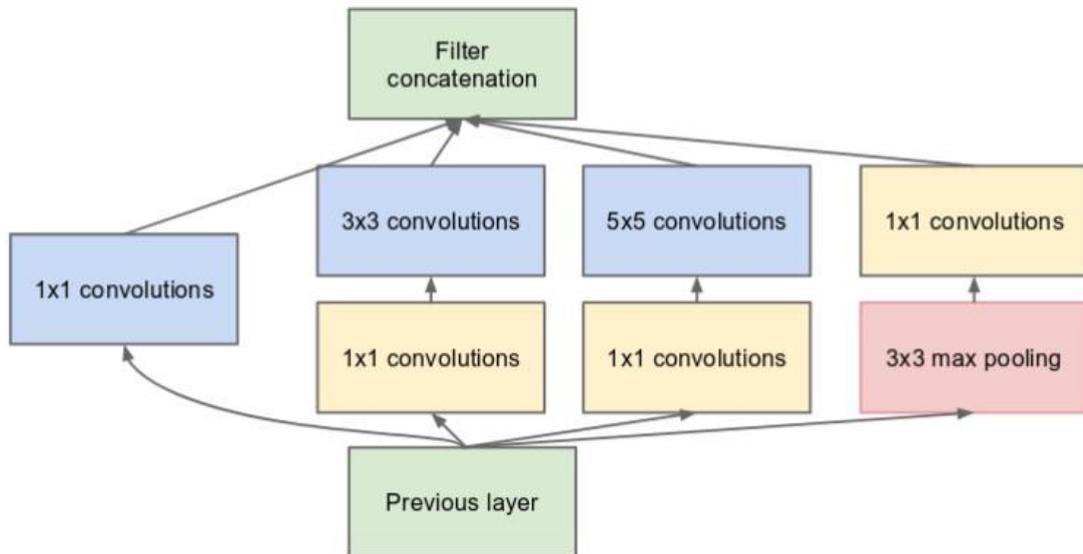
La construction d'un réseau Inception procède couche par couche après l'analyse des statistiques de corrélation de la dernière couche pour regrouper les unités à forte corrélation. Ces groupes forment les éléments de la couche suivante et sont connectés aux unités de la couche précédente. Cette architecture suppose que chaque unité de la couche précédente correspond à une région de l'image d'entrée et que toutes les unités sont regroupées dans des filtres.

Le module Inception typique comprend trois filtres de convolution de taille de  $1 \times 1$ ,  $3 \times 3$  et  $5 \times 5$  comme indiqué à la figure 5.1, mais cette décision se base davantage sur la commodité que sur la nécessité [21]. Pour des raisons purement techniques telles que l'utilisation efficace de la mémoire durant l'entraînement, il peut être aussi avantageux de commencer à utiliser les modules de l'Inception uniquement aux couches supérieures tout en utilisant la convolution traditionnelle pour les couches inférieures.

La compagnie Google a utilisé son réseau GoogleNet pour gagner le défi ImageNet 2014, grâce aux performances de ce réseau Inception qui a donné des améliorations notables par rapport au réseau neuronal convolutif classique (CNN). L'approche modulaire de GoogleNet suit le modèle de la figure 5.1, dont les propriétés suivent :

- 1- Utilisation d'un filtre de convolution  $1 \times 1$  pour la réduction des dimensions. Ce filtre permet de garder la taille d'entrée intacte mais en réduisant le nombre de canaux. L'exemple le plus frappant est la conversion d'une image RGB  $256 \times 256 \times 3$  en image composite  $256 \times 256 \times 1$ .
- 2- Usage de deux filtres de convolution de granularité  $3 \times 3$  et  $5 \times 5$  afin de capturer des détails de tailles différentes
- 3- Sorties des différents filtres concaténées pour alimenter l'étape suivante.

La connexion de plusieurs modules d'Inception dos à dos permet de bâtir un réseau très profond avec une structure décentralisée qui permet d'optimiser le nombre de paramètres et le temps de traitement.



**Figure 5.1** Schéma de principe de l'Inception

On peut aussi réduire le nombre de blocks Inception de façon à créer des versions légères avec un moindre coût de calcul. Cette variante peut donner des réseaux dont la rapidité est supérieure au réseau similaire sans architecture de l'Inception.

### 5.3 Incorporation de l'Inception :

Nous avons incorporé le paradigme de l'Inception dans la conception de notre architecture siamoise en utilisant le module de la figure 5.1 pour les réseaux jumeaux. La figure montre quatre chemins de traitement parallèles pour l'entrée qui effectuent tous une réduction de la taille des données. Ensuite, un bloc de convolution 1x1 additionnel est appliqué à la sortie du bloc de mise en commun et les deux blocs restants appliquent des masques de convolution appris pour extraire des caractéristiques spatiales utiles de l'entrée. Le bloc de convolution 3x3 se concentre sur les fonctionnalités de niveau inférieur par rapport au bloc de convolution 5x5 et la fonction d'activation ReLU est utilisée aux sorties. Les tailles des masques de convolution et de la fenêtre de pooling, ainsi que le saut de cette dernière sont des hyperparamètres.

L'entrée du premier module du réseau Inception est une image couleur de taille  $n \times n$  qui est codée en trois matrices de même taille pour les composants rouge, vert et

bleu. Les trois matrices sont traitées via les quatre chemins parallèles de la figure 5.1, et les résultats sont concaténés avant de produire la sortie du module.

Le bloc de contournement 1x1 à gauche de la figure a été éliminé dans ce travail, car il n'a eu aucun impact sur les résultats lors de nos expériences préliminaires. Cela a conduit à un module encore plus simple

#### 5.4 Architecture neuronale

Le schéma bloc de notre réseau de neurones siamois est illustré à la figure 5.2. Il combine deux modules Inception avec une unité de distance euclidienne et un classificateur de sortie pour construire un réseau qui prédit si une molécule en phase de recherche peut être qualifiée pour devenir un médicament. Il est composé de deux branches en parallèle qui utilisent des filtres 64 x64 pour réaliser les opérations de convolution et le passage entre les blocs de la même branche est réalisée par la fonction Relu. Les sorties des deux branches sont concaténées avant d'être aplaties sous la forme vectorielle dite : «Flatten». L'entrée de chaque module est une image couleur bidimensionnelle d'une structure moléculaire tirée de la base de données PubChem [70]. Chaque image a une taille de 32x32 pixels, ce qui conduit à une représentation matricielle 32x32 pour chaque plan de couleur. Cette taille est suffisamment petite pour un traitement efficace et elle est la même que dans le jeu d'images populaire CIFAR [71]. Les pixels des dessins 2D sont normalisés avant traitement.

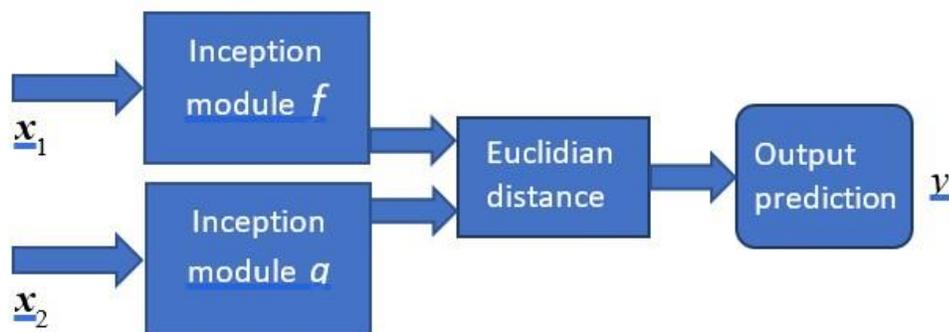
Notre conception globale suit [24] avec des modifications apportées pour l'apprentissage ponctuel. Mais au lieu d'utiliser des CNN comme dans [28] [30], notre modèle proposé repose sur le module Inception de la figure 5.1. Nous montrons dans la section de validation que ce modèle plus léger simplifie suffisamment la complexité du réseau pour en apprendre davantage sur les molécules étudiées avec quelques échantillons.

Le modèle fonctionne en trois étapes:

1. Présentation aux deux modules Inception identiques ( $f=g$ ) de la figure 5.2 de deux structures moléculaires en 2D,  $\mathbf{x}_1$  et  $\mathbf{x}_2$ , qui représentent une molécule candidate et une molécule de référence ;
2. Calcul de la distance euclidienne entre les deux vecteurs de représentation  $f(\mathbf{x}_1)$  et  $g(\mathbf{x}_2)$  produits;

3. Prédiction binaire du réseau à l'aide une fonction sigmoïde appliquée à la distance obtenue : 1 si les deux images sont similaires et 0 sinon.

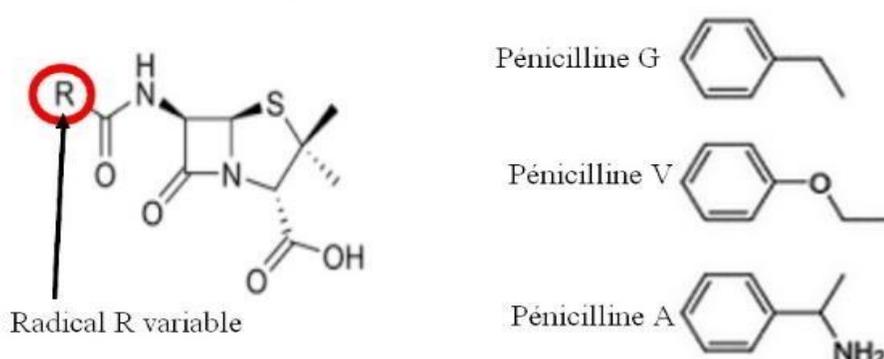
Globalement, un chemin différentiable est créé entre les entrées et la sortie du réseau, et ses paramètres peuvent ainsi être réglés par la méthode de descente du gradient [40].



**Figure 5.2** Architecture neuronale

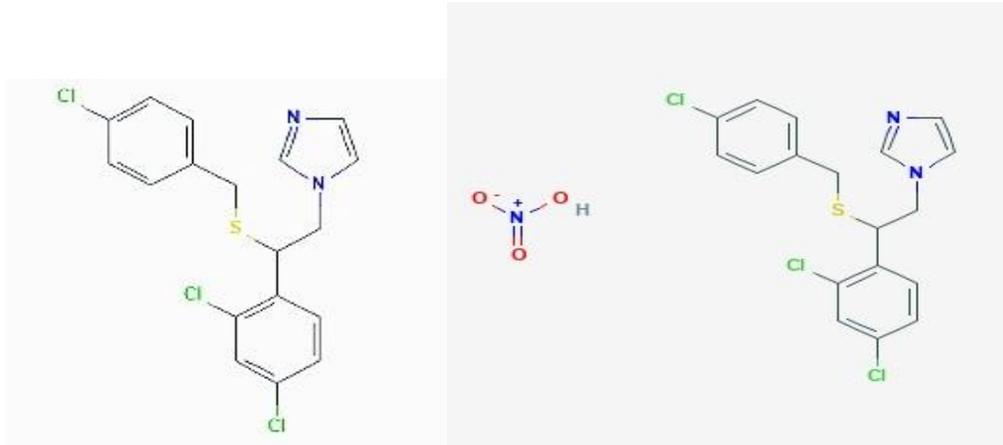
#### 5.4.1 Représentation des données

Le modèle est entraîné à la fois sur des paires de molécules positives (c'est-à-dire ayant des similitudes structurelles) et des paires de molécules négatives (donc, ayant peu ou pas de similitudes structurelles). Pour illustrer la procédure, considérons l'exemple de la pénicilline. La figure 5.3 montre que cette molécule a un noyau composé de dix atomes de carbone, treize atomes d'hydrogène, deux atomes d'azote, quatre atomes d'oxygène et un atome de soufre, plus un radical libre R. Ce radical représente différentes protéines qui peuvent se lier au noyau pour fabriquer des variantes de la pénicilline, avec pour effet d'élargir le spectre d'action de la pénicilline contre les bactéries nocives ou de modifier son mode d'absorption par le corps humain. Les variantes G, V et A représentées dans la figure 5.3, ainsi que d'autres molécules apparentées, partagent toutes la structure centrale montrée et peuvent ainsi former des paires positives pour l'entraînement.

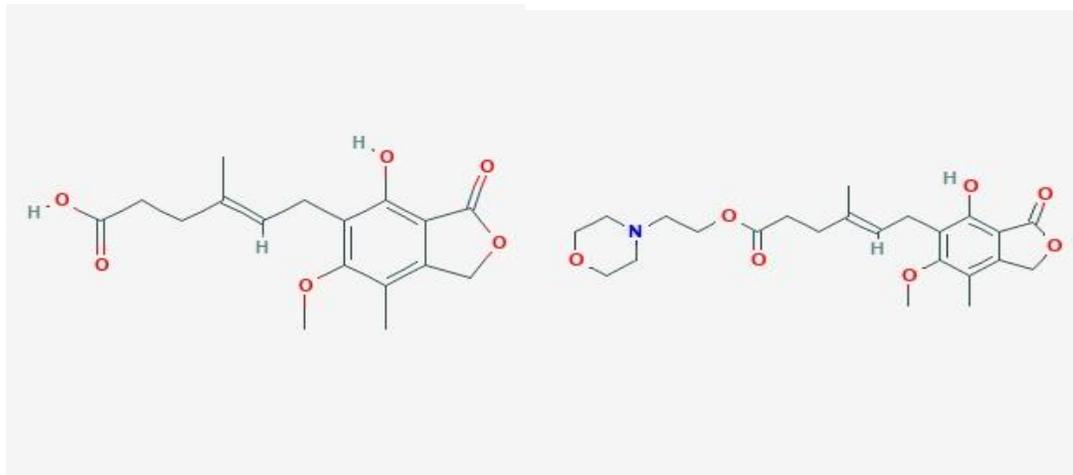


**Figure 5.3** Structure moléculaire de la pénicilline

La deuxième molécule considérée dans ce travail est le sulconazole (C<sub>18</sub>H<sub>15</sub>Cl<sub>3</sub>N<sub>2</sub>S). Ses dérivés sont obtenus en remplaçant certains atomes dans sa constitution de base par d'autres atomes ou groupes. À partir de la molécule de sulconazole, de nombreuses paires positives d'entraînement peuvent être réalisées en utilisant des variantes comme le nitrate de sulconazole (C<sub>18</sub>H<sub>15</sub>Cl<sub>3</sub>N<sub>3</sub>O<sub>3</sub>S), le butoconazole (C<sub>19</sub>H<sub>17</sub>Cl<sub>3</sub>N<sub>2</sub>S), le méthylsulfate de sulconazole (C<sub>17</sub>H<sub>15</sub>Cl<sub>3</sub>N<sub>2</sub>S) et d'autres molécules compatibles. La même chose peut être faite avec la troisième molécule considérée dans ce travail, l'acide mycophénolique (C<sub>17</sub>H<sub>20</sub>O<sub>6</sub>). Par exemple, il peut former des paires positives d'entraînement avec le mycophénolate de sodium (C<sub>17</sub>H<sub>19</sub>NaO<sub>6</sub>), le mycophénolate de méthyle (C<sub>18</sub>H<sub>22</sub>O<sub>6</sub>) et le mycophénolate mofétil (C<sub>23</sub>H<sub>31</sub>NO<sub>7</sub>). La figure 5.4 illustre la similitude structurale entre la molécule de sulconazole et son dérivé de nitrate de sulconazole, et la figure 5.5 fait de même pour la molécule de mycophénolate et son dérivé de mycophénolate mofétil.

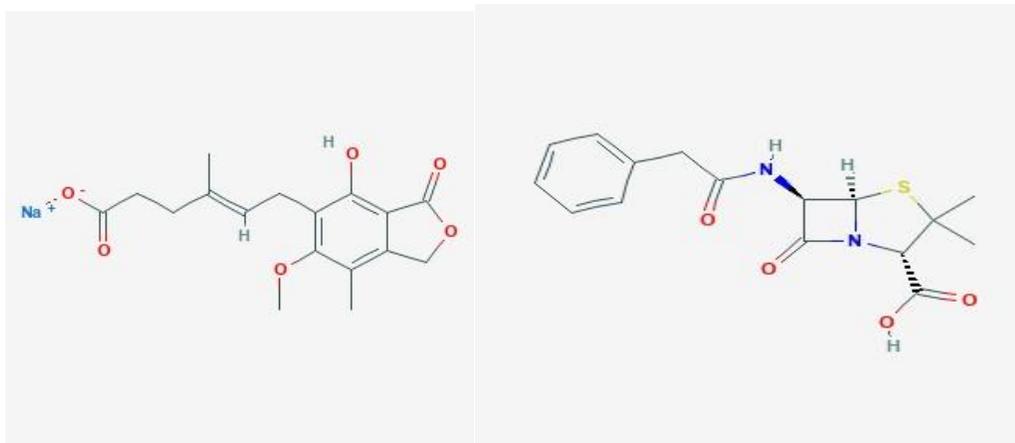


**Figure 5.4** Exemple d'une paire d'entraînement positive, utilisant la molécule de sulconazole (à gauche) et la molécule de nitrate de sulconazole (à droite)



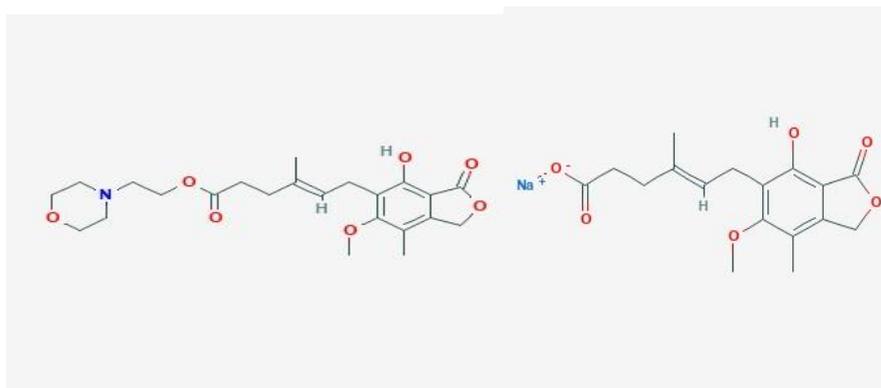
**Figure 5.5** Exemple d'une paire d'entraînement positive, utilisant la molécule d'acide mycophénolique (à gauche) et la molécule de mycophénolate mofétil (à droite)

D'autre part, des paires d'entraînement négatives sont obtenues avec des structures moléculaires sensiblement différentes. C'est le cas, par exemple, des molécules de pénicilline G et de mycophénolate de sodium, qui présentent plusieurs différences fondamentales comme le montre la figure 5.6.

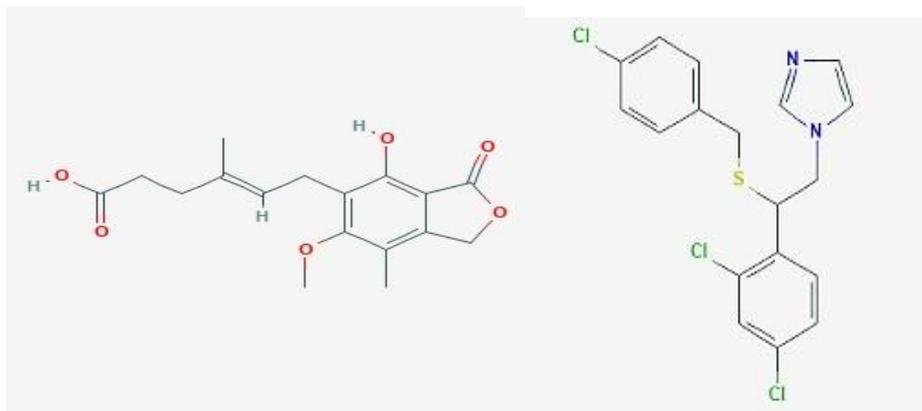


**Figure 5.6** Exemple d'une paire d'entraînement négative, utilisant la molécule de mycophénolate de sodium (à droite) et la molécule de pénicilline G (à gauche)

D'une façon similaire, les appariements négatifs peuvent être réalisés avec les molécules de mycophénolate mofétil (C<sub>23</sub>H<sub>31</sub>NO) et de butoconazole (C<sub>18</sub>H<sub>15</sub>Cl<sub>3</sub>N<sub>2</sub>S) comme le montre la figure 5.7, ou les molécules d'acide mycophénolate (C<sub>17</sub>H<sub>20</sub>O<sub>6</sub>) et de sulconazole comme le montre la figure 5.8.



**Figure 5.7** Exemple d'une paire d'entraînement négative, utilisant la molécule de mycophénolate mofétil (à gauche) et la molécule de butoconazole (à droite).



**Figure 5.8** Exemple d'une paire d'entraînement négative, utilisant la molécule d'acide mycophénolate (à gauche) et la molécule de sulconazole (à droite)

#### 5.4.2 Entraînement du réseau

Comme nous voulons entraîner le réseau de neurones sur des paires de molécules positives et négatives, nous présentons dans ce qui suit deux fonctions de perte qui conviennent à cet apprentissage binaire, à savoir la perte contrastive et la perte binaire. La perte contrastive en triplet [72] est une troisième fonction de perte possible, mais elle n'a pas été utilisée dans ce travail.

##### A. Fonction de perte contrastive

La fonction de perte contrastive a été utilisée par plusieurs chercheurs, notamment Yann LeCun et al. [73] dans l'article "Dimensionality Reduction by Learning an Invariant Mapping" qui traite de l'apprentissage d'une représentation vectorielle d'une image. L'objectif de la réduction de dimensionnalité est d'apprendre une nouvelle représentation dimensionnelle inférieure qui préserve la structure de l'entrée de sorte que des objets similaires soient représentés par des points proches et que les distances entre les vecteurs de sortie captent de manière significative les différences invariantes entre les entrées. Le problème consiste à trouver une fonction qui associe des modèles d'entrée de grande dimension sur des sorties de dimension inférieure, en fonction des relations de voisinage entre les échantillons dans l'espace d'entrée.

Contrairement aux systèmes d'apprentissage conventionnels où la fonction de perte est souvent définie par une somme d'erreurs, la fonction de perte contrastive est

appliquée sur des paires d'échantillons avec pour but d'encourager la génération de vecteurs de représentation proches pour les éléments de paires positives et éloignés pour ceux de paires négatives. Soit  $x_1$  et  $x_2$  une paire de vecteurs d'entrée présentée au système, et soit  $y$  une étiquette binaire affectée à cette paire avec valeur  $y = 1$  si  $x_1$  et  $x_2$  sont jugés similaires, et  $y = 0$  s'ils sont jugés différents. Comme notre architecture neuronale apprend à partir de la distance entre les sorties des deux réseaux inception de la figure 5.2, on peut définir la fonction de l'écart entre la sortie prédite de de la figure 5.2 et la valeur réelle attribuée à l'étiquette de chaque couple  $x_1$  et  $x_2$  soit la variable  $E$ :

$$E(x_1, x_2) = \|\text{Sortie prédite} - \text{étiquette du } (x_1, x_2)\|^2 \quad (5.3)$$

On définit la fonction perte  $L(W)$  en fonction des de l'ensemble  $W$  des poids synaptiques  $w$  du réseau. Elle peut être exprimée selon [73] comme la somme de toutes les pertes sur chaque couple  $(x_1, x_2)$  appartenant à l'ensemble d'entrée. La fonction de perte totale dans sa forme la plus générale est [73]:

$$L(W) = \sum_{i=1}^p (L(W, (y, x_1, x_2)))_i \quad (5.4)$$

où  $(y, x_1, x_2)$  est la paire d'entrées associée à l'étiquette  $y$ . On peut diviser la fonction de perte totale en deux composantes :  $L_s$  la perte pour une paire de points similaires et  $L_D$  la perte partielle pour une paire de points différents.

La perte combinée de l'échantillon d'ordre  $i$  peut être exprimée en fonction des pertes partielles selon :

$$L(W, y, x_1, x_2)_i = (1-y)L_s(E_i) + yL_D(E_i) \quad (5.5)$$

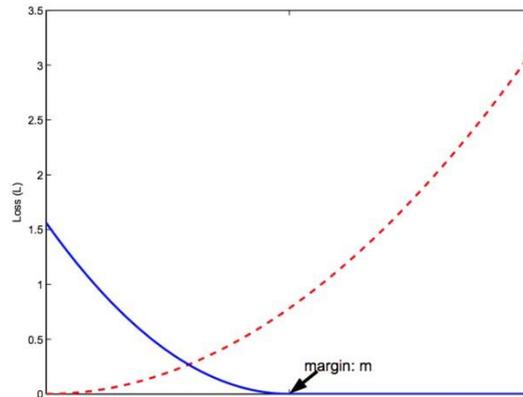
Elles sont définies comme suit dans [73] :

$$L(W, y, x_1, x_2) = (1-y) \times 1/2 \times (E)^2 + (y) \times 1/2 \{ \max(0, m - (E)) \}^2 \quad (5.6)$$

Où le paramètre  $m$  est supérieur à zéro ( $m > 0$ ) est une marge. La marge définit un rayon autour de la zone contenant les paires positives. Les paires différentes ne contribuent à la fonction de perte que si leur distance se situe dans ce rayon (voir figure 5.10). Le terme contrastif impliquant des paires différentes d'où l'importance de la fonction  $L_D$  [73].

La minimisation de la perte  $L$  est réalisée lorsque l'erreur de sortie  $E$  atteint des petites valeurs pour les paires similaires et de grandes valeurs pour les paires dissimilaires tel qu'indiqué dans la figure ci-dessous qui montre la contribution des paires positives

représentées par des pointillés et négatives représentés par un trait continu dans le calcul de la perte de contraste.



**Figure 5.9** Perte Contrastive [73]

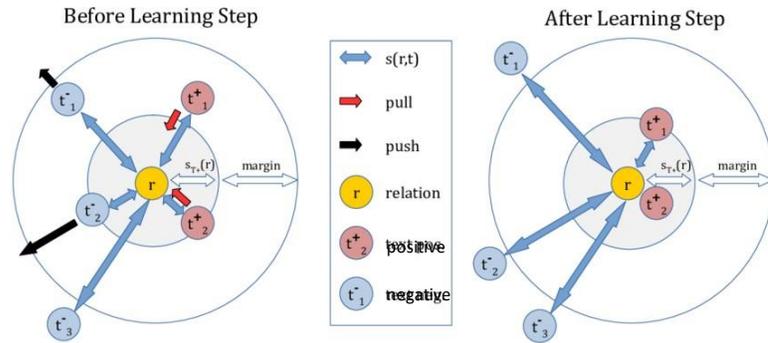
La fonction perte contrastive est appliquée à chaque lot d'exemple de l'ensemble d'apprentissage, et l'algorithme d'optimisation l'utilise pour mettre à jour les poids durant l'entraînement du réseau en respectant les étapes suivantes :

- 1- Pour chaque échantillon d'entrée  $x_i$  :
  - A- Utiliser la connaissance disponible sur les  $p$  échantillons d'entrée pour chercher l'ensemble  $S$  des échantillons tel que :
 
$$S = \{x_j\}_{j=1}^p, \text{ tel que } x_j \text{ est réputé être similaire à } x_i.$$
  - B- Étiqueter les paires  $(x_i, x_j)$  de sorte que l'étiquette est :
 
$$y_{ij} = 1 \text{ si } x_j, \text{ est similaire à } x_i, \text{ et } y_{ij} = 0 \text{ ailleurs.}$$
  - C- Combiner l'ensemble d'apprentissage en paires similaires et dissimilaires.
- 2- Pour chaque paire  $(x_i, x_j)$ , de l'ensemble d'apprentissage : Selon la valeur de  $y_{ij}$  : ( $y_{ij} = (1 \text{ ou } 0)$ )
 

Mettre à jour les paramètres en calculant  $L(W)$  selon l'équation 5.6.

La minimisation la fonction contrastive entraîne une augmentation et diminution des distances euclidiennes dans l'espace de sortie.

La figure ci-dessous illustre le comportement des paires d'échantillons avant et après l'apprentissage.



**Figure 5.10** Algorithme d'entraînement de la fonction Perte Contrastive [74]

La fonction de perte contrastive est définie par:

$$LS(D, y) = \frac{1}{y^2} D^2 + (1 - y) \frac{1}{2} \{ \max(0, (m - D)) \}^2 \quad (5.7)$$

Où  $y$  est l'étiquette binaire,  $D$  la distance euclidienne entre le codage neuronal des images d'entrée, et  $m$  un paramètre positif utilisé pour délimiter la zone qui nous intéresse. L'objectif est de favoriser les distances qui se situent dans la marge comme ce serait le cas pour les paires d'entraînement positives et de pénaliser les distances qui dépassent comme ce serait le cas pour les paires d'entraînement négatives. Dans l'équation 5.7, cela est pris en charge par le deuxième terme à gauche, qui s'applique uniquement aux distances qui sont au-delà de la marge. Dans ce travail, la valeur de la marge a été fixée à 1 par essais et erreurs.

## B. Fonction d'entropie croisée binaire

La classification binaire résout des problèmes de modélisation prédictive où les exemples se voient attribuer une valeur parmi deux étiquettes. Le problème est souvent défini comme prédisant une valeur de 0 ou 1 pour la première ou la deuxième classe et est souvent implémenté comme prédisant la probabilité que l'exemple appartient à la valeur de classe 1 [75].

La fonction d'entropie croisée binaire ou de perte logarithmique est définie comme suit :

$$LC(y) = -\frac{1}{N} \sum_{i=1}^N y_i \log(P(y_i)) + (1 - y_i) \log(1 - P(y_i)) \quad (5.8)$$

Où  $y_i$  est l'étiquette binaire d'un élément de l'ensemble de supports et  $P(y_i)$  est la probabilité de prédire qu'il est égal à 1. Les termes log dans cette fonction de perte

pénalisent les mauvaises prédictions près des limites de marge plus que la fonction de perte contrastive. En effet, lorsque la probabilité associée à la vraie classe est faible, le terme log correspondant est très important; à l'inverse, lorsque la probabilité est élevée, le terme est proche de 0. Par conséquent, minimiser cette fonction de perte maximise la probabilité de vraies prédictions.

#### 5.4.3 Choix de la fonction perte

Comme notre problème de classification est de type binaire, les exemples à traiter seront regroupées en paires positives s'ils sont de la même classe ou en paires négatives s'ils sont de classes différentes, et Le modèle génère un score de similarité pour chaque paire d'entrée compris entre 0 et 1. Il détermine aussi à quel groupe de parmi les trois molécules de l'ensemble d'apprentissage appartiennent les paires positives.

La fonction de perte contrastive est la mieux appropriée pour traiter notre problème relatif à l'apprentissage ponctuel pour la simplicité de son implémentation.

Cependant, on lui ajoute la perte binaire pour une meilleure précision d'apprentissage, aboutissant ainsi à une somme pondérée entre les deux fonctions. Cette combinaison permet de mieux contrôler les paires positives et négatives qui sont au bord de la limite et peuvent la franchir dans un sens ou dans l'autre.

#### 5.4.4 Ajustement du réseau de neurones

Le réseau neuronal siamois proposé est entraîné par une somme pondérée des fonctions de perte de contraste [73] et de la perte d'entropie croisée binaire [75] comme cela a été fait dans Wen et al. [76]. À l'origine, seule la fonction de perte de contraste était considérée, car elle est généralement adaptée à l'apprentissage binaire utilisant des données regroupées par paires [73]. Cependant, cela a conduit à une précision de prédiction moyenne de seulement 75% lors des expériences de validation, probablement en raison du chevauchement entre les paires d'apprentissage positives et négatives dans les régions frontalières de l'espace du classificateur. Nous avons donc ajouté la fonction de perte binaire pour améliorer la discrimination.

La somme pondérée des deux fonctions de perte est :

$$L = LS + \lambda LC \quad (5.9)$$

Où  $LS$  est la perte de contraste,  $LC$  est la perte binaire et  $\lambda$  est un paramètre de réglage.

Au chapitre suivant, l'architecture neuronale de la figure 5.2 sera soumise à un ensemble de tests en utilisant différents lots d'échantillons avec un maximum de 50 paires de molécules pour évaluer l'efficacité de l'apprentissage ponctuel dans un contexte de manque d'informations. Chaque molécule sera représentée par une image 32x32 pixels et voici un a



**Figure 5.11** Image 2D de la molécule de Pénicilline et sa représentation 32x32 pixels

**NB :** Le flou de la représentation 32x32 pixels est dû à l'agrandissement de l'image.

## CHAPITRE VI

### RÉSULTATS ET DISCUSSION

#### 6.1 Validation des Résultats

Comme mentionné précédemment, nous avons considéré trois petites molécules de structures chimiques différentes et de propriétés thérapeutiques distinctes dans ce travail. Le choix de ces molécules pour tester notre modèle a été guidé par la volonté de développer une architecture avec un large champ d'application sans surapprentissage. Les trois molécules ont été sélectionnées dans la banque de données PubChem [70], qui contient plus de 16 millions de structures moléculaires de base et leurs dérivés. Pour rappel, les trois molécules sont:

- Pénicilline qui est un antibiotique lent pour combattre les bactéries dans le corps humain,
- Mycophénolate qui empêche le rejet d'organe transplanté,
- Sulconazole qui est un antifongique pour les infections cutanées.

Les trois molécules ont des structures chimiques différentes comme le montrent les figures 5.5 à 5.7, et chacune peut conduire à de nouvelles molécules qui diffèrent par un ou plusieurs atomes ou radicaux attachés à la structure centrale, avec potentiellement de nouvelles propriétés thérapeutiques. Par exemple, les variantes de pénicilline illustrées sur la figure 5.3 sont toutes des antibiotiques, mais elles ont des spectres d'application différents. De même, la molécule de mycophénolate et ses molécules similaires sont utilisées pour empêcher le rejet d'organes transplantés, et l'utilisation d'une molécule ou d'une autre dépend de l'organe. Enfin, le sulconazole et ses molécules dérivées sont des antifongiques pour la peau et le choix d'une molécule par rapport à l'autre dépend de la nature de l'état de la peau. Ainsi, notre choix couvre différentes applications médicales et pharmaceutiques, allant d'un antibiotique à usage externe ou interne, à un antifongique à usage externe, en passant par un médicament à usage purement interne.

Dans nos simulations, nous avons apparié chacune des trois molécules de référence avec des molécules similaires et dissimilaires pour former des paires positives et négatives. Nous avons pris chaque médicament et ses dérivés pour former les paires positives et un mélange du médicament avec les deux autres et leurs dérivés pour former les paires négatives, par exemple, en combinant une molécule de pénicilline

avec une molécule de mycophénolate ou de sulconazole. L'ensemble d'entraînement est construit de trois classes qui représentent les trois molécules ci-dessus présentées. Nous avons utilisé des lots de 5, 10, 25 et 50 paires pour évaluer l'impact de la taille de l'ensemble d'entraînement sur le processus d'apprentissage unique, et les données ont été divisées à 80% pour l'entraînement et 20% pour les tests. Les lots ont été mélangés au hasard trois fois.

Pour chaque classe, on a utilisé un maximum 76 molécules dont 52 sont exploitées pour former les 26 paires positives du lot 50 paires et les 24 autres servent à constituer la moitié des paires négatives. Nous utilisons l'approche N coups tel que la valeur de N est inférieure ou égale à 50. On se fixe une précision supérieure à 90% pour entraîner le réseau à séparer les paires positives des paires négatives. Lorsqu'une nouvelle molécule est présentée au réseau, elle est associée avec une molécule de même structure ou similaire puis elle est alliée à une molécule dissimilaire et l'ensemble formé des trois classes sera testé sans refaire l'entraînement. La précision représente le rapport entre les prévisions correctes et le total des prévisions à réaliser sur l'ensemble de test. Si on veut tester un ensemble de 10 paires de molécules par exemple et qu'on arrive à faire 9 prédictions correctes, notre précision est de 90%.

L'implémentation logicielle du modèle a été réalisée en python, à l'aide de l'API Keras [77] et les dessins couleur 32x32 pixels des molécules étudiées ont été traités comme des images 2D. On a préféré cette représentation à la représentation structurée comme les graphes orientés pour simplifier le processus de classification dans un premier temps. Dans un deuxième temps, on veut éviter les problèmes liés à la représentation graphique qui résultent dans le choix de la forme de la représentation structurée et l'absence d'un formalisme mathématique qui lie la représentation à la structure chimique de la molécule. Après la création des structures moléculaires synthétiques, chaque plan de couleur 32x32 des images originales et synthétiques est converti en matrices à virgule flottante normalisées en divisant les valeurs de pixels 8 bits par 255. Ensuite, les matrices obtenues sont manipulées par les modules Inception pour fournir les arguments du module de distance euclidienne, avant la prédiction finale comme illustré dans la figure 5.1.

Au cours de la phase d'apprentissage, le modèle a été entraîné par la descente de gradient en utilisant la fonction combinée de perte contrastive et binaire définie dans l'équation 5. Le vecteur d'initialisation du poids a été réglé sur de très petites valeurs ( $\sim 10^{-4}$ ) et le paramètre lambda était égal à 0,1. L'apprentissage a été arrêté après 1000 itérations car la sortie s'est stabilisée avant cela. Toutes ces valeurs ont été obtenues par essais et erreurs.

La plate-forme de simulation consistait en une station PC avec un processeur Intel i5 cadencé à 2,5 GHz, 8 Go de RAM et un système d'exploitation Windows 64 bits. Aucun GPU n'a été utilisé.

## 6.2 Résultats :

Le tableau 6.1 résume les résultats obtenus. Le temps d'apprentissage variait de 9 à 86 minutes en fonction de la taille du lot, avec une convergence vers une perte inférieure ou égale à 0,03, et une précision de prédiction moyenne de l'ensemble de test dépassant 97% pour la taille de lot de 25 paires. Le tableau 6.1 montre également que les résultats des ensembles d'entraînement à 5 et 10 paires ont fourni une précision instable. La fonction de perte pour les deux cas était comprise entre 0,025 et 0,040, mais la précision variait de 0,50 à 0,98, probablement en raison d'un surajustement. Ces résultats ne sont pas compatibles avec la découverte de médicaments qui nécessite des résultats précis et stables. En revanche, les résultats des lots de 25 et 50 paires répondaient à ces exigences. Il est intéressant de noter également que doubler la taille du lot de 25 à 50 paires a réduit la perte de l'entraînement au détriment de la généralisation, ce qui indique généralement un surajustement. En effet, l'ensemble de 50 échantillons a conduit à une perte d'entraînement inférieure à l'ensemble de 25 paires, mais la précision moyenne obtenue avec l'ensemble de 25 échantillons était meilleure.

**Tableau 6.1** : Résultats des simulations :

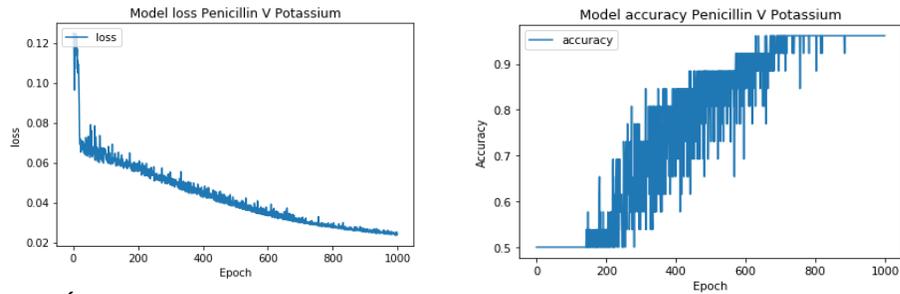
Taille de l'ensemble d'entraînement (paires + & -)	Temps d'entraînement (min)	Molécule	Précision (%)
50	86	Pénicilline	97
		Mycophénolate	96.0
		Sulconazole	92.0
25	43	Pénicilline	97
		Mycophénolate	96.15
		Sulconazole	98.0
10	17	Pénicilline	50.0-98.0*
		Mycophénolate	50.0-98.0*
		Sulconazole	70.0-88.0*
5	9	Pénicilline	70.0-98.0*
		Mycophénolate	50.0-97.0*
		Sulconazole	65.0-98.0*

\*Oscille entre les valeurs indiquées sans convergence.

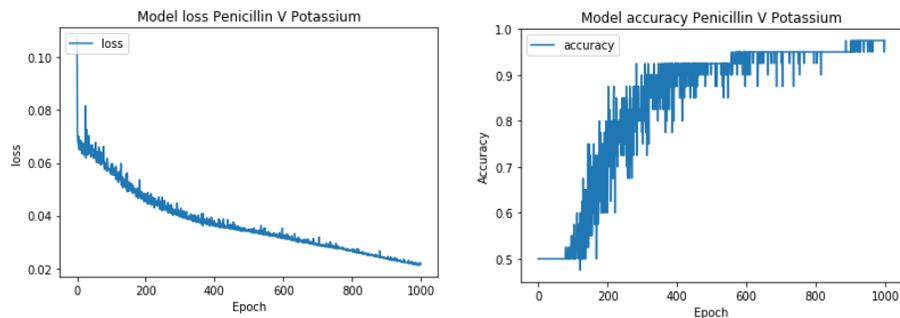
### 6.3 Évolution de l'entraînement :

Nous illustrons dans cette section l'évolution des courbes de perte et de précision de chaque simulation pour les expériences utilisant un ensemble d'entraînement de 25 et de 50 paires dont les paires positives sont formées respectivement avec chaque classe de molécule.

Pénicilline : La manière de former des paires positives et négatives était celle décrite dans la section 5.4.1 et la nouvelle molécule introduite sur l'ensemble d'entraînement formé des trois classes déjà énoncées et la molécule candidate est la Pénicilline V Potassium. Les figures 6.1 et 6.2 montrent la diminution progressive de la fonction de perte et l'augmentation progressive de la précision vers un plateau à fur à mesure que le nombre d'époques d'apprentissage augmente, en utilisant la fonction de perte définie par l'équation (5) avec  $\lambda = 0,1$ .

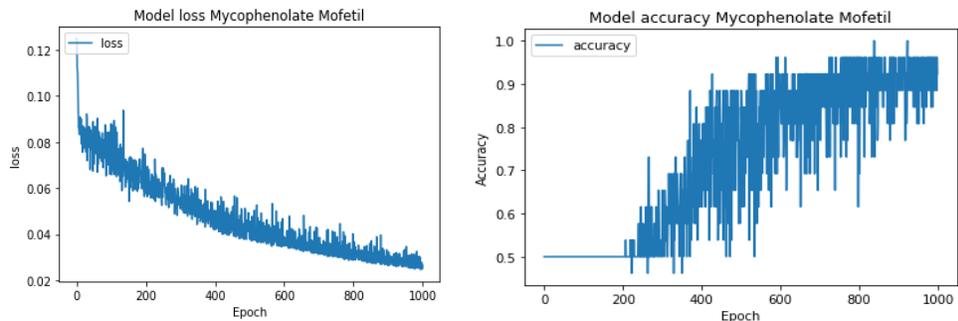


**Figure 6.1** Évolution de la perte et de la précision du modèle avec entraînement de 25 paires pour la pénicilline

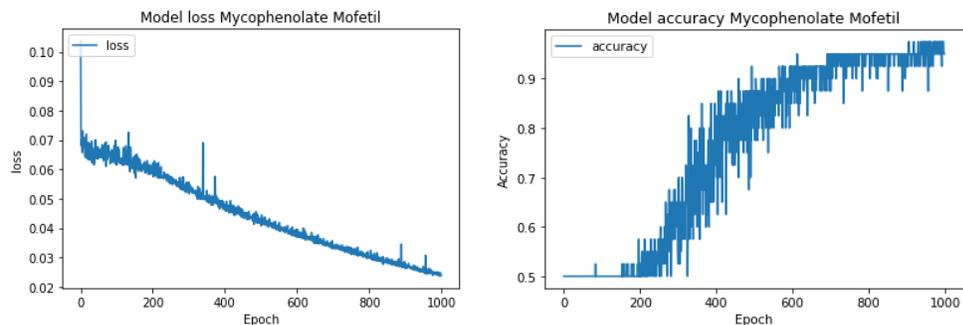


**Figure 6.2** Évolution de la perte et de la précision du modèle avec entraînement de 50 paires pour la pénicilline

**Mycophénolate:** Le même protocole expérimental utilisé avec la pénicilline a été reconduit et la molécule candidate est la Mycophénolate mofétile. Les courbes de perte et de précision qui en résultent sont illustrées aux figures 6.3 et 6.4.

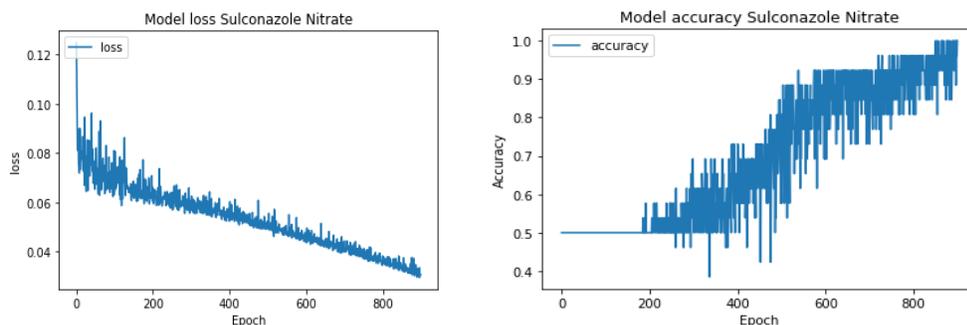


**Figure 6.3** Évolution de la perte et de la précision du modèle avec entraînement de 25 paires pour la mycophénolate mofétile.

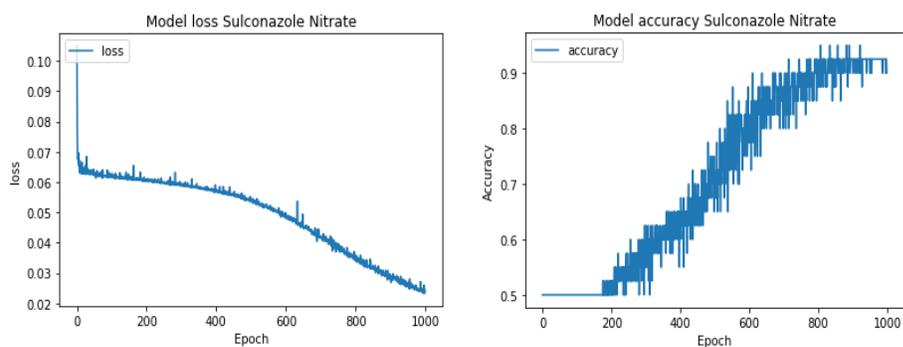


**Figure 6.4** Évolution de la perte et de la précision du modèle avec entrainement de 50 paires pour la mycophénolate mofetil

**Sulconazole:** Les figures 6.5 et 6.6 montrent des performances similaires par rapport aux deux autres molécules et la molécule candidate est la sulconazole nitrate.

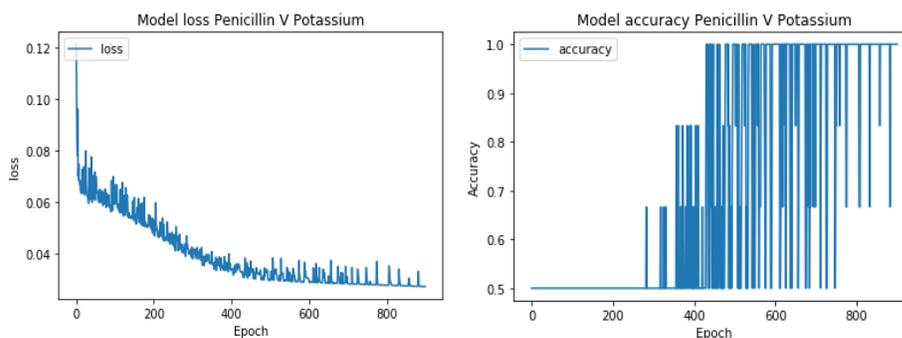


**Figure 6.5** Évolution de la perte et de la précision du modèle avec entrainement de 25 paires pour la Sulconazole Nitrate.

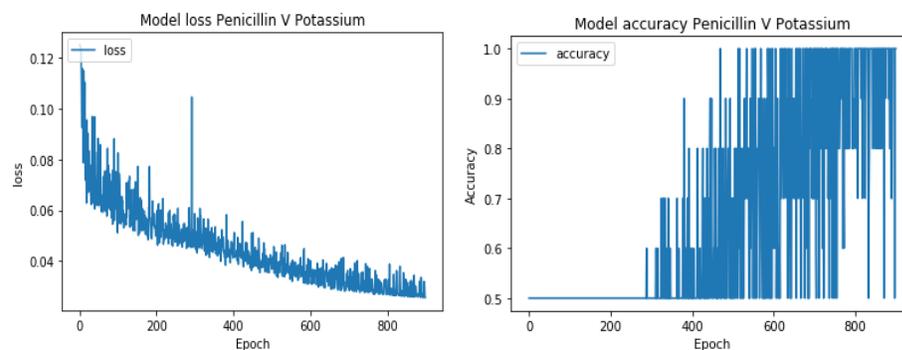


**Figure 6.6** Évolution de la perte et de la précision du modèle avec entrainement de 50 paires pour la Sulconazole Nitrate.

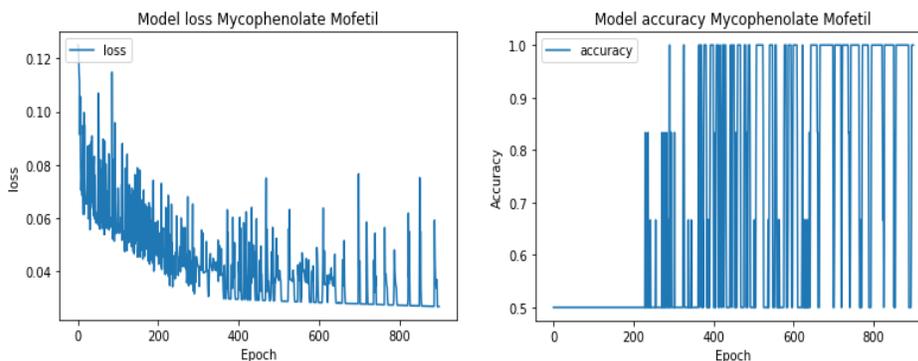
Les figures 6.7 à 6.12 montrent des performances de la fonction perte et de la précision pour un ensemble d'entraînement de 5 paires pour les trois molécules utilisées pour la simulation avec les mêmes candidates.



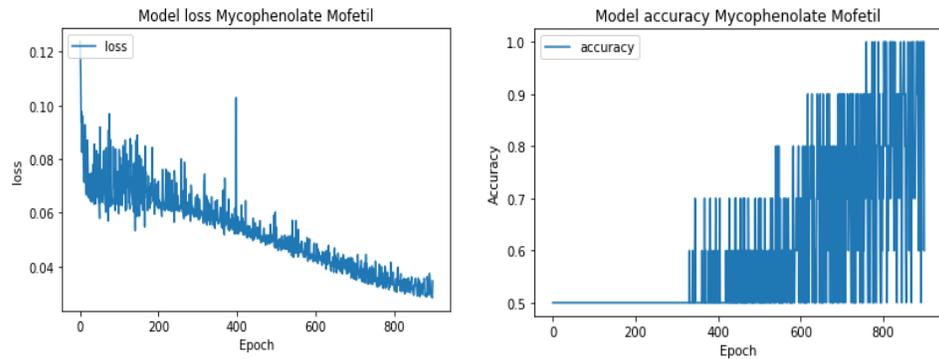
**Figure 6.7** Évolution de la perte et de la précision du modèle avec entraînement de 5 paires pour la Pénicilline V Potassium.



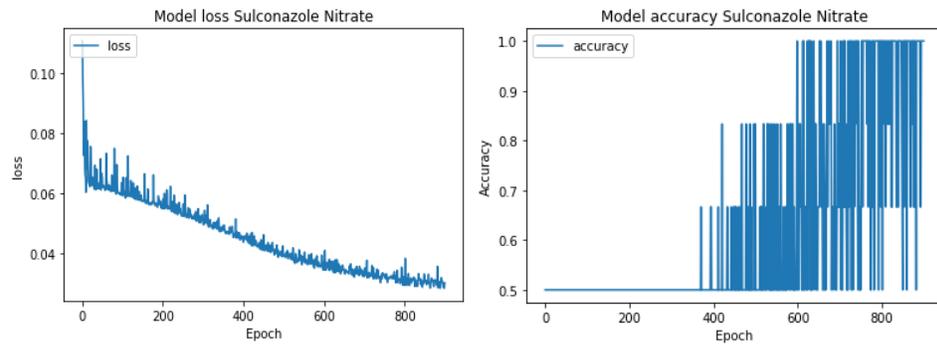
**Figure 6.8** Évolution de la perte et de la précision du modèle avec entraînement de 10 paires pour la Pénicilline V Potassium



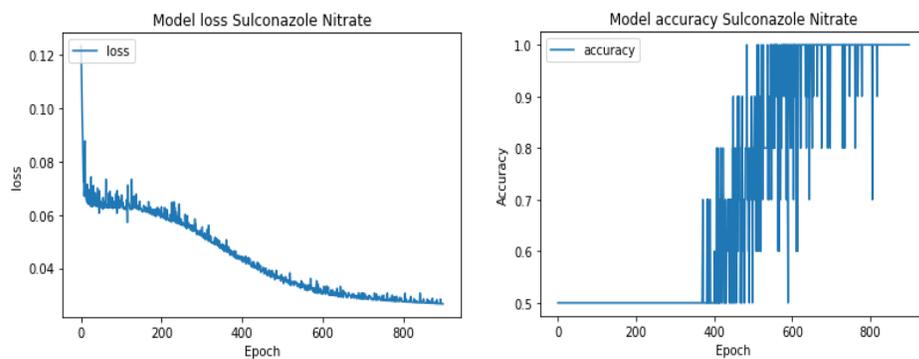
**Figure 6.9** Évolution de la perte et de la précision du modèle avec entraînement de 5 paires pour la mycophénolate mofetil



**Figure 6.10** Évolution de la perte et de la précision du modèle avec entraînement de 10 paires pour la mycophénolate mofetil



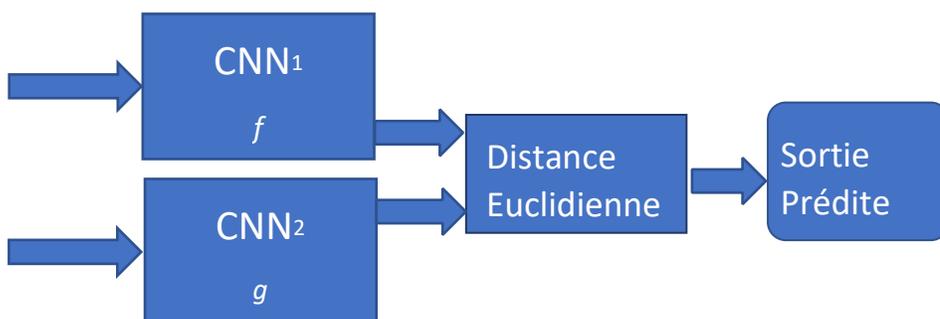
**Figure 6.11** Évolution de la perte et de la précision du modèle avec entraînement de 5 paires pour la Sulconazole Nitrate



**Figure 6.12** Évolution de la perte et de la précision du modèle avec entraînement de 10 paires pour la Sulconazole Nitrate

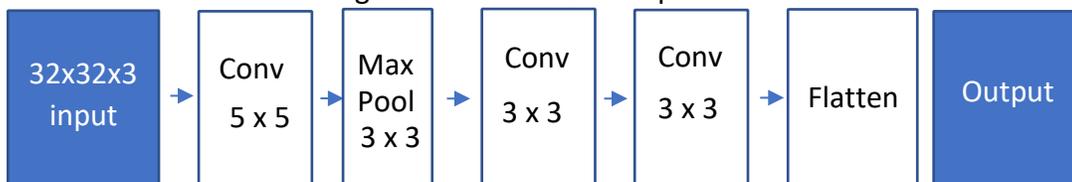
#### 6.4 Comparaison avec un modèle CNN :

Étant donné que la plupart des réseaux siamois sont basés sur des CNN et que le module Inception utilisé dans ce travail utilise également des convolutions, nous avons évalué l'utilisation d'un CNN régulier avec différentes tailles de masque à des fins de comparaison. Cela a conduit au modèle de réseau de neurones siamois illustré à la figure 6.13



**Figure 6.13** Architecture de réseau neuronal siamois utilisant des CNN

Afin de définir les composants du CNN, nous avons utilisé le modèle VGG-16 [78] qui minimise l'utilisation des opérations maxpooling pour éviter la perte de l'information. Nous avons étudié plusieurs configurations en modifiant le nombre et le type de couches convolutives. La figure 6.14 montre celle qui a conduit aux meilleurs résultats.



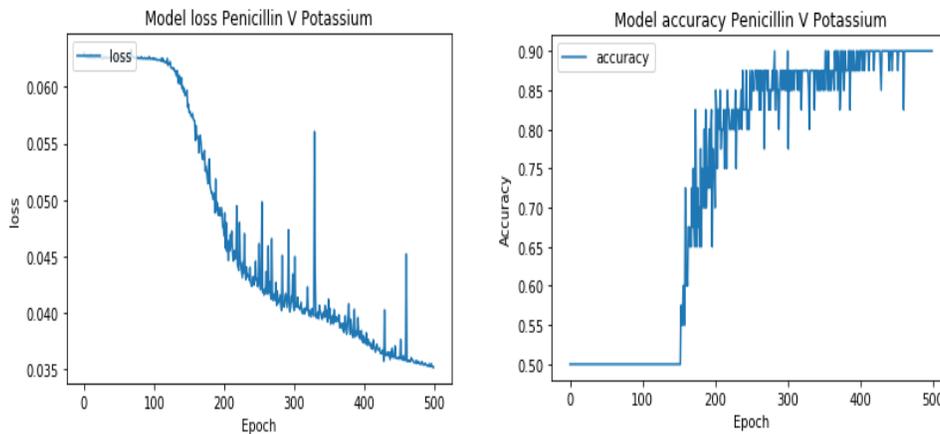
**Figure 6.14** Structure CNN pour le réseau neuronal siamois basé sur CNN

Pour accomplir les expériences avec le CNN, on utilise le même ensemble d'entraînement et les mêmes molécules candidates. Le tableau 6.2 montre que l'architecture basée sur le CNN nécessitait 30% de temps de traitement de plus que l'alternative basée sur l'Inception, avec une différence moyenne de précision de classification de + 7.66% pour l'ensemble d'entraînement à 50 paires et -3.71% pour l'entraînement à 25 paires. De plus, les figures 6.15 à 6.17 et 6.18 à 6.20 montrent que

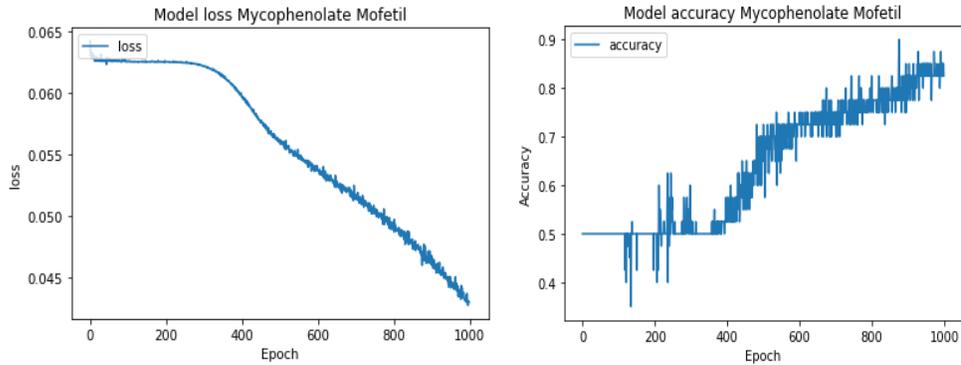
la convergence du modèle basé sur le CNN était très lente pour l'ensemble d'entraînement à 25 paires par rapport à l'ensemble d'entraînement à 50 paires. Ces résultats indiquent que l'architecture siamoise basée sur l'Inception est la plus efficace, car elle est plus rapide, plus simple à mettre en œuvre et, surtout pour la découverte de médicaments, plus précise pour les petits ensembles d'entraînement de taille réduite.

**Tableau 6.2** Résultats de la simulation avec un réseau siamois utilisant les CNN

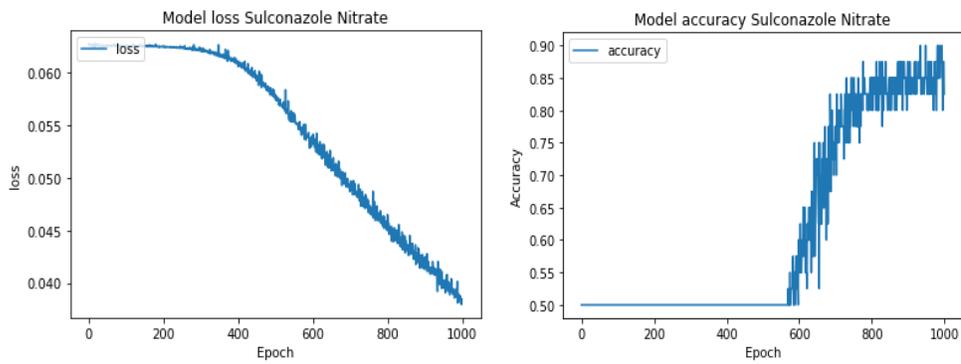
Taille de l'ensemble d'entraînement	Temps d'entraînement (min)	Molécule	Précision (%)
50	112	Pénicilline	90.0
		Mycophénolate	85.0
		Sulconazole	87.0
25	56	Pénicilline	92.0
		Mycophénolate	95.0
		Sulconazole	93.0



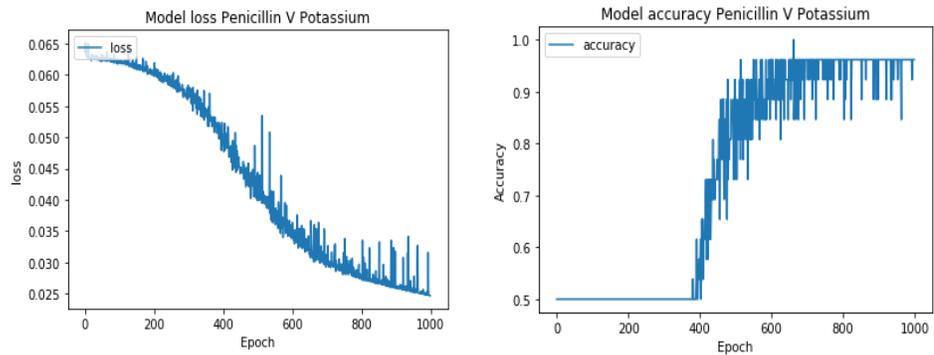
**Figure 6.15** Courbes de perte et de précision du modèle basé sur le CNN avec entraînement de 50 paires pour la pénicilline V Patassium



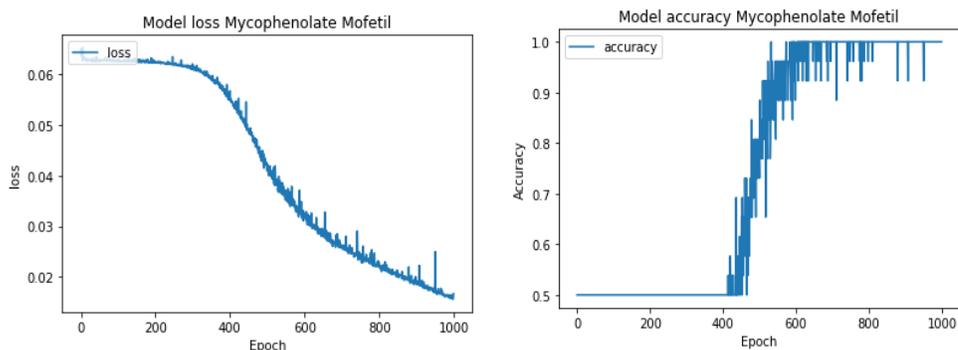
**Figure 6.16** Courbes de perte et de précision du modèle basé sur le CNN avec entraînement de 50 paires pour la mycophénolate mofetil.



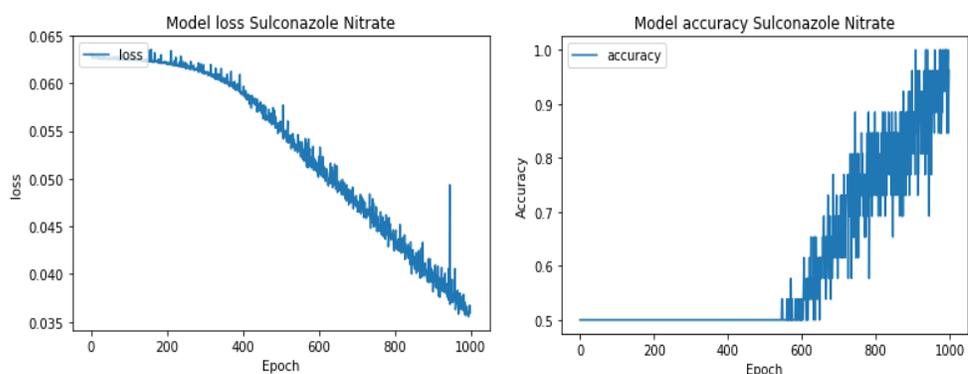
**Figure 6.17** Courbes de perte et de précision du modèle basé sur le CNN avec entraînement de 50 paires pour la Sulconazole Nitrate.



**Figure 6.18** Courbes de perte et de précision du modèle basé sur le CNN avec entraînement de 25 paires pour la pénicilline V Potassium



**Figure 6.19** Courbes de perte et de précision du modèle basé sur le CNN avec entraînement de 25 paires pour la mycophénolate mofetil.



**Figure 6.20** Courbes de perte et de précision du modèle basé sur le CNN avec entraînement de 25 paires pour la Sulconazole Nitrate.

## 6.5 Discussion des résultats

Nos résultats expérimentaux montrent que, pour les trois molécules considérées, la méthode d'apprentissage ponctuel proposée était toujours efficace pour prédire la similitude d'une molécule candidate avec la molécule de référence, avec une précision moyenne supérieure à 97% obtenue pour une taille de lot de 25 paires positifs et paires d'entraînement négatives.

La méthode proposée peut également traiter de nouvelles données de catégorie sans nécessiter un nouvel entraînement et fournir une bonne précision de prédiction.

Le temps consacré à l'entraînement est de 43 minutes sur une station PC qui permet de faire fonctionner le système dans un cadre quasi-temps réel, en particulier si un

accélérateur GPU est utilisé. L'architecture proposée peut également être implémentée pour former un système embarqué en raison de sa relative simplicité.

Nous n'avons trouvé dans la littérature qu'un seul autre travail qui permet une comparaison directe avec le nôtre [33]. D'autre part, les réseaux siamois ont été utilisés dans plusieurs domaines d'application avec des préoccupations similaires aux nôtres. Le tableau 6.3 présente une comparaison générale de ce travail avec les résultats de l'article [33] et avec les autres techniques d'apprentissage ponctuelles basées sur le réseau siamois. Le tableau montre des résultats nettement meilleurs lors de l'utilisation de notre modèle en termes de taille combinée de l'ensemble d'apprentissage et de l'erreur de prédiction.

Comme d'autres algorithmes fondés sur les réseaux de neurones, le modèle proposé fait des prédictions basées sur les similitudes de données. Par conséquent, il n'a aucune connaissance de ce que les données représentent et de leurs propriétés biochimiques comme le ferait un biochimiste, basant sa production uniquement sur les similitudes entre les représentations moléculaires 2D. Par conséquent, le modèle ne peut suggérer que des molécules candidates et une intervention humaine est nécessaire pour la validation avant les étapes suivantes

**Tableau 6.3** Comparaison avec d'autres travaux utilisant des réseaux siamois (meilleurs résultats)

Travail	Taille de l'ensemble d'entraînement	Temps d'apprentissage (itération)	Application	Erreur (%)
[33]	2x10	18000	Découverte de médicaments	0.173
[79]	25000	30	Prédiction des cibles de médicament	20
[80]	1296	200	Chromosomes	4.4
[81]	2054	10 000	Spectres de résonance nucléaire magnétique 2D	2-50
[82]	67360	50	Interaction médicament	6.1
[83]	2x15	N/A	Liaison ligandrécepteur	4.9
[84]	1258	1000	Similitude des séquences peptidiques	6.7
Ce travail	2x25	1000	Découverte de médicaments	0.2

On a également remarqué que tous les réseaux siamois que nous avons développés fonctionnaient anormalement lorsque les images des molécules de l'ensemble de test étaient très différentes des références, ce qui entraînait une baisse importante de la précision des prédictions. Des travaux supplémentaires sont nécessaires pour déterminer les limites du modèle à cet égard et de rechercher les moyens judicieux pour étendre son utilisation.

Enfin, les deux modules de l'Inception de la figure 5.1 peuvent subir une simplification supplémentaire. Nous avons utilisé les images des structures moléculaires telles

qu'elles apparaissent en couleur dans la base de données PubChem [70]. S'ils pouvaient être transformés en images en noir et blanc sans pénalité sur la précision de prédiction du modèle, les blocs de convolution 1x1 de la figure 5.2 peuvent être éliminés pour un traitement plus rapide.

## CHAPITRE VII

### CONCLUSION GÉNÉRALE

En conclusion, nous avons présenté une solution d'apprentissage ponctuel appliqué à la découverte de médicaments sous la contrainte du manque de données. Le réseau neuronal siamois proposé peut faire des prédictions sur la similitude entre les molécules candidates et celles de référence avec une précision moyenne de plus de 97% après entraînement avec des ensembles de 25 paires d'exemples positifs et négatifs. En tant que tel, il permet de gagner du temps et des efforts dans les projets de la découverte de médicaments en orientant dès le début la recherche vers les molécules ayant le potentiel le plus élevé. Le modèle peut également être utilisé dans d'autres domaines où un manque de données d'entraînement empêche l'utilisation de réseaux d'apprentissage en profondeur réguliers, ou lorsque la manipulation de nouvelles classes sans réentraînement est souhaitée.

Nos résultats peuvent être facilement trouvés en utilisant l'architecture de la figure 5.2 avec un bon choix de paires positives et négatives des images d'entrée. D'autres versions du modèle de l'Inception peuvent être choisies.

Notre approche enrichit la littérature d'apprentissage ponctuelle appliquée à la découverte de médicaments avec peu d'informations et donne des résultats intéressants en termes de précision et de temps de réponse. La distance métrique qui est la base de l'apprentissage du réseau peut être revue à son tour. Avec une meilleure disponibilité des données sur les raisons de l'échec des molécules candidates à devenir des médicaments tel que la toxicité, la solubilité et le degré d'activité, on pourra introduire la distance cosinus pour limiter l'espace où se trouve les coordonnées des représentations moléculaires en 2D et prédire ainsi les molécules ayant la structure chimique qui leur permet d'éviter les échecs ci-haut cités. Par ailleurs, ce travail peut servir de base pour développer la recherche dans le domaine de la découverte des médicaments et donne l'opportunité d'utiliser d'autres approches pour mieux intégrer l'intelligence artificielle dans le domaine pharmaceutique.

Comme perspective future, l'utilisation des réseaux antagonistes génératifs GAN peut être envisagée, car ce type de réseau a un grand pouvoir de générer de nouvelles structures moléculaires proches de celles des molécules cibles avec un degré de réalisme appréciable.

## BIBLIOGRAPHIE

- [1] B. Ramsundar et al., "Massively multitask networks for drug discovery", arXiv:1502.02072 [stat.ML], 6 Feb 2015.
  - [2] V.Vanhoucke, "Automating Drug Discoveries Using Computer Vision", Google AI Blog, 2018, Disponible sur : <https://ai.googleblog.com/2018/07/automating-drugdiscoveries-using.html>
  - [3] C.Jee, "An AI system identified a potential new drug in just 46 days", MIT Technology Review, 2019, Disponible sur : <https://www.technologyreview.com/2019/09/03/133175/an-ai-system-identified-a-potential-new-drug-in-just-46-days/>
  - [4] "Apprentissage par renforcement", Data Analyst Post, Disponible sur : <https://dataanalyticspost.com/Lexique/apprentissage-par-renforcement/>
  - [5] L.Bastien, "Reinforcement Learning : qu'est-ce que l'apprentissage par renforcement ?", Le Big Data, 2019, Disponible sur : <https://www.lebigdata.fr/reinforcement-learning-definition>
  - [6] D.Nelson, "What is a Generative Adversarial Network(GAN) ?", Unite.AI, 2020, Disponible sur : <https://www.unite.ai/what-is-a-generative-adversarial-networkgan/>
  - [7] Google Developers/ Generative Adversarial Networks/ The Discriminator, Disponible sur : <https://developers.google.com/machine-learning/gan/discriminator>
  - [8] J.Brownlee, "A Gentle Introduction to Generative Adversarial Networks (GANs)", Machine Learning Mastery, 2019, Disponible sur : <https://machinelearningmastery.com/what-are-generative-adversarial-networksgans/>
  - [9] S. Tsang, 'CGAN – Conditional GAN (GAN)', Medium, 2020, Disponible sur : <https://medium.com/ai-in-plain-english/review-cgan-conditional-gan-gan78dd42eee41>
  - [10] X. Chen et al., "Drug-target interaction prediction: Databases, web servers and computational models", Briefings in Bioinformatics, vol. 17, no. 4, pp. 696–712, 2016.
-

- [11] A. Mayr et al., “A Large-scale comparison of machine learning methods for drug target prediction on ChEMBL”, *Chemical Science*, 6 June 2018
- [12] I. Goodfellow, Y. Bengio and A. Courville, *Deep Learning*, MIT Press, 2016, Available at <http://www.deeplearningbook.org>.
- [13] S. Thomas et al., “Robust Object Recognition with Cortex-Like Mechanism”, *IEEE Trans. Pattern Anal. Mach. Intel.*, vol. 29, no. 3, pp. 411–426, March 2007.
- [14] T. Unterthiner et al., “Deep Learning as an Opportunity in Virtual Screening”, *NIPS DL Workshop*, vol. 27, 2014.
- [15] J. You, R.D. McLeod, P. Hu, “Predicting drug-target interaction network using deep learning model”, *Comput. Biol. Chem.*, vol. 80, pp. 90–101, March 2007.
- [16] E.D. George et al., “Multi-task Neural Networks for QSAR Predictions”, arXiv:1406.1231v1 [stat.ML], 4 June 2014.
- [17] J. Ma, R.P. Sheridan, A. Liaw, G.E. Dahl and V. Svetnik, “Deep neural nets as a method for quantitative structure-activity relationships”, *J. Chem. Inf. Model*, vol. 55, pp. 263–274, 2015.
- [18] M. Lin et al., “Network in network”, arXiv:1312.4400v3 [cs.NE], 4 Mar 2014.
- [19] D.K. Duvenaud et al., “Convolutional networks on graphs for learning molecular fingerprints”, *NIPS*, vol. 28, pp. 2224–2232, 2015.
- [20] S. Kearnes et al., “Molecular Graph Convolutions: Moving Beyond Fingerprints”, *J. Comput.-Aided Mol. Des.*, vol. 30, pp. 595–608, 2016.
- [21] C. Szegedy et al., “Going Deeper with Convolutions”, C. Szegedy et al., "Going deeper with convolutions," *CVPR 2015*, Boston, MA, pp. 1-9.
- [22] C. Szegedy et al., “Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning”, arXiv:1602.07261 [cs.CV], 23 Aug 2016.
- [23] A. Krizhevsky, I. Sutskever and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” *Advances in Neural Information Processing Systems 25*, pp. 1097–1105, 2012.
- [24] G.B. Goh et al., “Chemception: A Deep Neural Network with Minimal Chemistry Knowledge Matches the Performance of Expert-developed QSAR/QSPR Models”, arXiv:1706.06689 [stat.ML], 20 June 2017.
- [25] M. Smith and J.C. Smith, “Repurposing Therapeutics for COVID-19:

Supercomputer-Based Docking to the SARS-CoV-2 Viral Spike Protein and Viral Spike Protein-Human ACE2 Interface”, ChemRxiv. 11871402.v3, Feb. 2020.

- [26] S. Adam et al., “One-shot Learning with Memory-Augmented Neural Networks”, arXiv:1605.06065v1 [cs.LG], 19 May 2016.
- [27] G. Alex, W. Greg and D. Ivo, “Neural Turing Machines”, arXiv 1410.5401 [cs.NE], 10 Dec 2014.
- [28] E. Zakharov et al., “Few-Shot Adversarial Learning of Realistic Neural Talking Head Models”, arXiv:1905.08233v2 [cs.CV], 25 Sep. 2019.
- [29] S. Gidaris and K. Nikos, “Dynamic Few-Shot Visual Learning without Forgetting”, arXiv:1804.09458v1 [cs.CV], 25 Apr. 2018.
- [30] G. Koch, R. Zemel and R. Salakhutdinov., “Siamese Neural Networks for Oneshot Image Recognition,” ICML deep learning workshop, Lille, France, vol. 2, 2015.
- [31] B.M. Lake et al., “One shot learning of simple visual concepts”, Proc. 33rd Conf. of the Cognitive Science Society, vol. 172, 2011.
- [32] A. Puzanov and K. Cohen, “Deep Reinforcement One-Shot Learning for Artificially Intelligent Classification Systems”, arXiv:1808.01527v1 [cs.LG], 4 Aug. 2018.
- [33] H. Altae-Tran et al. “Low Data Drug Discovery with One-Shot Learning”, ACS Cent. Sci. 2017, 3, 4, 283–293, 3 April 2017.
- [34] M. Weber, M. Welling and P. Perona, “Unsupervised learning of models for recognition”, Proc. 6th ECCV, vol. 2, pp. 101-108, 2000.
- [35] R. Fergus, P. Perona and A. Zisserman, “Object class recognition by unsupervised scale-invariant learning”, Proc. CVPR, vol. 2, pp. 264-271, 2003.
- [36] L. Fei-Fei, R. Fergus and P. Perona, “A Bayesian approach to unsupervised oneshot learning of object categories”, Proc. 9th IEEE International Conference on Computer Vision, vol. 2, pp. 1134-1141, 2003.
- [37] L.E. Lansdowne, “Exploring the Drug Development Process”, Technology Networks, 2020, Disponible sur : <https://www.technologynetworks.com/drugdiscovery/articles/exploring-the-drug-development-process-331894>

- [38] "Biopharmaceutical Research & Development : The Process Behind Medicines", PhRMA, Disponible sur : [http://phrmadocs.phrma.org/sites/default/files/pdf/rd\\_brochure\\_022307.pdf](http://phrmadocs.phrma.org/sites/default/files/pdf/rd_brochure_022307.pdf)
- [39] H.Sankesara, "N-Shot Learning : Learning More with Less Data", Floydhub, 2019, Disponible sur : <https://blog.floydhub.com/n-shot-learning/>
- [40] U. Kamath, J. Liu et J. Whitaker, "Deep Learning for NLP and Speech Recognition", Springer, 2019.
- [41] I. Melekhov, J. Kannala and E. Rahtu, "Siamese network features for image matching", *2016 23rd International Conference on Pattern Recognition (ICPR)*, pp. 378-383, Dec 2016.
- [42] J. Bromley et al., "Signature verification using a siamese time delay neural network", *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 7, no. 04, pp. 669–688, 1993.
- [43] L.Deng et Y.Liu, "Deep learning in natural language processing", Springer, 2018.
- [44] R. Lambert, "Focus : Le Réseau de Neurones Artificiels ou Perceptron Multicouche", *Pensée Artificielle*, 2018, Disponible sur : <https://penseeartificielle.fr/focus-reseau-neurones-artificiels-perceptronmulticouche/>
- [45] X. Jiang et al., "Deep Learning in Object Detection and Recognition", Springer, 2019.
- [46] "Différence entre apprentissage supervisé et apprentissage non supervisé", *ActuIA*, 2018, Disponible sur : <https://www.actuia.com/vulgarisation/differenceentre-apprentissage-supervise-apprentissage-non-supervise/>
- [47] Jason Brownlee , "Framework for Data Preparation Techniques in Machine Learning", July 13, 2020 in *Data Preparation*, Disponible sur: <https://machinelearningmastery.com/framework-for-data-preparation-for-machinelearning/>
- [48] ] J.Brownlee, "What is Deep Learning ?", *Machine Learning Mastery*, 2019, Disponible sur : <https://machinelearningmastery.com/what-is-deep-learning/>
- [49] C.Kone, "Introduction aux réseaux de neurones de convolution", *Medium*, 2019, Disponible sur :

[https://medium.com/@penanklihicyrille\\_16953/introductionaux-r%C3%A9seaux-de-neurones-de-convolution-b5da87a5321f](https://medium.com/@penanklihicyrille_16953/introductionaux-r%C3%A9seaux-de-neurones-de-convolution-b5da87a5321f)

- [50] R.Girshick, J.Donahue, T.Darell, J.Malik, "Rich feature Hierarchies for Accurate Object Detection and Semantic Segmentation", IEEE Conference on Computer Vision and Pattern Recognition, p. 580-587, 2014.
- [51] B.Raj, "A Simple Guide to the Versions of the Inception Network", Towards Data Science, 2019, Disponible sur : <https://towardsdatascience.com/a-simpleguide-to-the-versions-of-the-inception-network-7fc52b863202>
- [52] O. Russakovsky, J. Deng, H. Su et al., ImageNet Large Scale Visual Recognition Challenge. Int J Comput Vis, vol. 115, p. 211–252, 2015.
- [53] T. Serre, L. Wolf, S. Bileschi, M. Riesenhuber, T. Poggio, "Robust object recognition with cortex-like mechanisms". IEEE Trans Pattern Anal Mach Intell., vol. 3, 2007.
- [54] M.Lin, Q.Chen, S.Yan, "Network In Network", arXiv:1312.4400v3, 2013.
- [55] J.Brownlee, "A Gentle Introduction to 1x1 Convolutions to Manage Model Complexity", Deep Learning for Computer Vision, 2019, Disponible sur : <https://machinelearningmastery.com/introduction-to-1x1-convolutions-to-reducethe-complexity-of-convolutional-neural-networks/>
- [56] A.Gonfalonieri, "Applications of Zero-Shot Learning", Towards Data Science, 2019, Disponible sur : <https://towardsdatascience.com/applications-of-zero-shotlearning-f65bb232963f>
- [57] S.Chandhok, "Zero-Shot Learning : An Introduction", Learn OpenCV, 2020, Disponible sur : <https://www.learnopencv.com/zero-shot-learning-an-introduction/>
- [58] M.Olson, A.Wyner et R.Berk, "Modern Neural Networks Generalize on Small Data Sets", NeurIPS, 2018.
- [59] G.Sumbul, R.G.Cinbis, S.Aksoy, " Fine-Grained Object Recognition and Zero-Shot Learning in Remote Sensing Imagery", IEEE Transactions on Geoscience and Remote Sensing, vol. 56, 2018.
- [60] F.G.Mohammadi, M.H.Amini et H.R.Arabnia, " An Introduction to Advanced Machine Learning : Meta Learning Algorithms, Applications and Promises", arXiv:1908.09788v1 [cs.LG] 26 Aug 2019.

- [61] S. Hsiao, D. Kao, Z. Liu et R.Tso, “Malware Image Classification Using One-Shot Learning with Siamese Networks”, ScienceDirect, vol. 159, pp. 1863–1871, 2019, Disponible sur : <https://www.sciencedirect.com/science/article/pii/S1877050919315595>
- [62] H. Jung et al., “Utilizing Natural Language for One-Shot Task Learning”, Journal of Logic and Computation, vol. 18, no. 3, pp. 475–493, June 2008.
- [63] S. Maitra, R.K. Ojha and K. Ghosh, “Impact of Convolutional Neural Network Input Parameters on Classification Performance”, 2018 4th Int. Conf. Convergence in Technology (I2CT), pp. 1-5, 2018.
- [64] J. Wayim, “Paper Review: Matching Networks for One Shot Learning”, Medium, 2020, Disponible sur : <https://medium.com/@jonhwayim/paper-review-matchingnetworks-for-one-shot-learning-f7300c09e180>
- [65] Q.Cai et al., “Memory Matching Networks for One-Shot Image Recognition”, arXiv:1804.08281v1, 2018.
- [66] O. Vinyals et al., “Matching Networks for One Shot Learning”, arXiv:1606.04080 [cs.LG], 29 Dec. 2017.
- [67] A.Gresse, R.Dufour, V.Labatut, M.Rouvier et J.Bonastre, “Mesure de similarité fondée sur des réseaux de neurones siamois pour le doublage de voix”, XXXIIèmes Journées d’Études sur la Parole (JEP), 2018.
- [68] P.Mercure, “Développement de médicaments: l'intelligence artificielle peut-elle prévenir les échecs? ”, LaPresse, 2018, Disponible sur : <https://www.lapresse.ca/sciences/medecine/201811/12/01-5203820developpement-de-medicaments-lintelligence-artificielle-peut-elle-prevenir-lesechecs.php>
- [69] Revue Médicale Suisse, disponible sur : <https://www.revmed.ch/>
- [70] National Library of Medicine, National Center for Biotechnology Information, Disponible sur : <https://pubchem.ncbi.nlm.nih.gov/>
- [71] The CIFAR-10 dataset, Disponible sur : <https://www.cs.toronto.edu/~kriz/cifar.html>.
- [72] B. Williams, “Contrastive Loss Explained”, Towards Data Science, 2020, Disponible sur : <https://towardsdatascience.com/contrastive-loss-explained159f2d4a87ec>

- [73] R. Hadsell, S. Chopra, Y. Lecun, "Dimensionality Reduction by Learning an Invariant Mapping", IEEE Computer Society Conference, vol. 2, 2006.
- [74] T. Kiliyas et al., "IDEL : In-Database Entity Linking with Neural Embeddings", arXiv:1803.04884v1 [cs.DB], 13 Mar 2018.
- [75] Keras API reference / Losses, Disponible sur: <https://keras.io/api/losses/>.
- [76] Y. Wen et al., "A Discriminative Feature Learning Approach for Deep Face Recognition," ECCV 2016, LNCS 9911, Springer, Cham.
- [77] Keras Developer guides / The functional API, Disponible sur: [https://keras.io/guides/functional\\_api/](https://keras.io/guides/functional_api/).
- [78] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks For LargeScale Image Recognition", arXiv:1409.1556v6, 10 Apr. 2015.
- [79] K. Y. Gao et al., "Interpretable Drug Target Prediction Using Deep Neural Representation," IJCAI'18, Stockholm, SW, pp. 3371–3377, 2018.
- [80] Swati et al., "Siamese Networks for Chromosome Classification", IEEE Int. Conf. Comp. Vision Workshops (ICCVW), Venice, 2017, pp. 72-81.
- [81] C. Zhang et al., "Small Molecule Accurate Recognition Technology (SMART) to Enhance Natural Products Research," Sci Rep. 2017 Oct 27;7(1): 14243, 2017.
- [82] D. S. Dhama et al., "Beyond Textual Data: Predicting Drug-Drug Interactions from Molecular Structure Images using Siamese Neural Networks," arXiv:1911.06356v2 [cs.LG] 29 Jun 2020.
- [83] A. O. D. Santos and D. D. A. Ruiz, "Virtual Screening Assisted by Siamese Neural Networks," 13th Int. Florida Artificial Intelligence Research Society Conference (FLAIRS), 2017.
- [84] N. Savioli, "One-shot screening of potential peptide ligands on HR1 domain in COVID-19 glycosylated spike (S) protein with deep siamese network," arXiv:2004.02136v3 [q-bio.QM] 11 Apr 2020.