الجمهورية الجزائرية الديمقراطية الشعبية

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE YAHIA FARES DE MEDEA

FACULTE DES SCIENCES

Département Mathématique et Informatique



Projet de fin d'études

Présenté pour l'obtention du Diplôme de

Master en Informatique

Option : Ingénierie des Systèmes et Technologie du Web

THEME

Séléction et extraction des attributs à l'aide de l'apprentissage profond:

application sur des données médicales réelles

Proposé et dirigé par :

Mr. Sahmadi Brahim

Préparé et présenté par :

Melle. Boukerroucha Ouafa
Mr. Harbi abdenacer

Année Universitaire 2017-2018

Remerciements

Nous remercions d'abord le bon Dieu qui nous a aidé et qui nous a donné le courage et la patience pour réaliser ce modeste travail.

Notre profonde gratitude et sincère remerciement vont à notre promoteur Mr.Sahmadi Brahim de nous avoir encadrer, pour son suivi et son orientation.
Nous tenons également, à lui notre profonde reconnaissance pour le temps précieux qu'il nous a consacré, ses remarques pertinentes et précieuses, ses conseils et ses encouragements.

Nous tenons à remercier nos enseignants qui nous ont transmis le savoir et le bien surtout Mr.Redha Mazari pour son aide et sa présence.

Nos remerciements vont aussi aux membres de jury pour avoir accepté de juger ce travail.

Nous tenons à remercier également tous ceux qui nous ont aidé et contribué de prés ou de loin à la réalisation de notre projet plus précisement nos amis et collègues : Melle.Amraoui Yasmine, Melle.Mesbah Souad et Mr.Hayi Mohammed Yassine.

Dédicace

À ma confidente et meilleure amie :Maman je te dédie ce modeste travail afin de te remercier pour tes efforts, sacrifices, prières et soutien pour que je concrétise mes rêves ; merci pour te ce que tu as fait pour moi depuis le jour de ma naissance j'espère que je serais à la hauteur de la confiance et l'espoir que tu as en moi.

À l'homme de ma vie : Papa tous ce que j'ai pu réaliser jusqu'à aujourd'hui c'est surement grâce à toi et à tes sacrifice pour notre bien êtres, ton soutien pour continuer et aller de l'avant, voici un fruit de tes efforts et ton éducation.

À mon grand frère, mon protecteur : Mustapha je te remercie d'être toujours là pour moi, je te souhaite tous le bonheur du monde.

À ma petite sœur la prunelle de mes yeux :Fairouz je te remercie pour tes encouragements et ta présence à mes côtés, je te souhaite la réussite dans tes études.

À ma tendre grand-mère : Mani que dieu te garde pour nous merci pour tes encouragement et prières, merci d'être toujours à nos côtés.

À ceux qui m'ont quitté beaucoup trop Baba sidou, Yaya, et mani j'espère que la ou vous vous trouvez vous etes fière de votre petite fille qui vous ais tant aimé.

À tous les membres de ma famille, ainsi que mes amis qui m'ont encouragé et soutenu durant mon parcours.

Boukerroucha Ouafa

Dédicace

C'est avec profonde gratitude et sincères mots,que je dédie ce modeste travail de fin d'étude à mes chers parents ; qui ont sacrifié leur vie pour ma réussite et m'ont éclairé le chemin par leurs conseils judicieux.

J'espère qu'un jour je pourrais leurs rendre un peu de ce qu'ils ont fait pour moi, que dieu leur prête bonheur et longuevie.

Je dédie ce travail mes frères et à ma sœur, ainsi qu'à tous les membres de ma famille.

A toute ma promotion, mes amis et tous ceuxqui ont été toujours là pour moi.

Harbi abdenacer

Résumé

Les bases de données volumineuses causent beaucoup de problème dans divers domaines, nous nous intéressons dans ce travail au domaine de la classification.
Quatre bases de données à classes binaire ont été choisies pour appliquer successivement un filtrage de donnée aussi surnommé de sélection d'attributs ensuite une extraction de ces derniers ; ceci est la première partie du travail.
La deuxième partie consiste à prédire les classes des nouvelles bases de données, elle a été effectuée via deux méthodes, par réseaux de neurones profond et par l'algorithme SVM, tous les détails a propos de ces méthodes seront disponible au cours de ce rapport.
Les technique et méthodes appliquées ont donné des résultats satisfaisants.

Mot clés : apprentissage profond, autoencodeur empilé, selection d'attributs, extraction d'attributs, séparateurs à vaste marge, validation croisée.

ملخص

قواعد البيانات الكبيرة تسبب الكثير من المشاكل في مختلف المجالات ، ونحن مهتمون بهذا العمل
في مجال التصنيف.
تم اختيار أربعة قواعد بيانات من الدرجة الثنائية لتطبيق ترشيح البيانات بشكل متتابع وكذلك اختيار السمة المستخلصة ثم استخلاص هذه الأخيرة ؛ هذا هو الجزء الأول من العمل
الجزء الثاني يتمثل في التنبؤ بفئات قواعد البيانات الجديدة ، وتم تنفيذها عبر طريقتين ، من خلال الشبكات العصبية العميقة ومن خلال الخوارزميات "فواصل النطاق العريض"، وستتوفر جميع التفاصيل حول هذه الطرق في خلال هذا التقرير
.أعطت التقنيات والأساليب المطبقة نتائج مرضية

كلمات مفتاحية: التعلم العميق ،التشفير التلقائي المكدس اختيار السمة ،استخراج السمة ،الفواصل ذات الهامش العريض ، التصديق المتبادل

Absract

Large databases cause a lot of problems in various fields, we are interested in this work in the field of classification.
Four binary-class databases have been chosen to successively apply data filtering also nicknamed feature selection and then extraction of those latter; this is the first part of the job.
The second part consists in predicting the classes of the new databases, it was carried out via two methods, by deep neural networks and by the SVM algorithm, all the details about these methods will be available during this report.
The techniques and methods applied have given satisfactory results.

Key words : deep learning, stacked auto encoder, feauture selection, attribute extraction, support vector machine, cross validation.

Tables des matières

Liste des figures
Liste des tableaux
Liste des acronymes
Résumé
Introduction Générale……01
chapitre I : Réduction de la dimensionalité pour la classification
1. Introduction……03
2. Classification……..03
2.1. Classification supervisée……03
2.1.1. Approche générative……..…04
2.1.2. Approche Discriminante……..…04
2.2. Classification non supervisée……..…04
2.2.1. La notion de similarité……..05
2.2.2. Les approches classique pour la classification non-supervisée…..……..…05
2.2.2.1. Méthode hierarchique……..……..…05
2.2.2.2. Méthode de partitionnement……..…06
2.2.2.3. Modèle de mélange……..……..…06
2.2.2.4. Approche neuromémitique……..……...…06
2.3. Classification semi-supervisée……...……..…07
3. Classification des données de grande dimension……...……07
3.1. Fléau de dimsionnalité……...……07
3.1.1. Limite entre une grande et une petite dimension de l'espace..……..…08
3.1.2. Nombre de points nécessaires pour l'apprentissage dans un espace de grande dimension……...08
3.2. Fléau de dimensionnalité en classification……..08
4. Réduction de dimension……..……09
4.1. Sélection d'attributs……..10
4.1.1. Définition……..10
4.1.2. Techniques de sélection dattributs……11
4.1.2.1 Méthodes filtres……..……11
4.1.2.2 Méthodes enveloppes……...…12
4.1.2.3. Méthodes embarquées……..……..…12
4.1.3. Les étapes de procédure de sélection……....13
4.1.3.1. Procédure de génération……..…13
4.1.3.2. Fonction d'évaluation……14
4.1.3.3. Condition d'arrêt……14
4.2. Extraction des attributs……15
4.2.1. Méthodes linaires……..…15
4.2.1.1. Analyse en composantes principales (ACP)……...…16
4.2.1.2. Analyse discriminante linéaire……...…16
4.2.2. Méthodes non linéaires……...……..…16
4.2.2.1. Méthodes globales……16
4.2.2.2. Méthodes locales……17
5. Conclusion……..17
Chapitre 2 : L'APPRENTISSAGE PROFOND
1. Introduction……18
2. Apprentissage automatique……18
2.1. défintion……..…..18
2.2.les réseaux de neurones……18

2.3. limitation des algorithmes classique……...18
3. Apprentissage profond……19
3.1. Groupe d'architectures connues……..19
3.1.1. réseaux de neurone profond……...…...19
3.1.1.1. perceptron multicouche……..20
3.1.1.2. réseaux de croyance profond……..21
3.1.1.3. Autoencodeur empilé……22
3.1.2. réseaux de neurones convolutionnels……...23
3.1.3. réseaux de neurone récurrent……24
3.1.3.1. LSTM……25
3.1.3.2. GRU……...…25
3.2. Nouvelles architecures……...…..27
3.2.1. Réseaux de neurone temporel spatial-profond…….....…..27
3.2.2. Réseaux de neurones récurent multidimensionnel…….…..…….27
3.2.3. Les Auto-encodeur convolutionnels……...…..…….27
4. Conclusion…….…….28
chapitre III : Méthode de Conception
1. introduction……29
2. méthode proposée pour la réduction de dimensionalité……29
2.1. La sélection d'attributs avec une méthode de filtrage……30
2.1.1. Gain d'information……...31
2.1.2. Inforation mutuelle……...31
2.1.3. Le X^2 statistique……...31
2.2. Réduction de dimension via l'extraction d'attributs……..32
2.2.1. L'encodage……..33
2.2.2. Le décogae……..34
2.2.3. La reconstruction……35
3. Classification des données……35
3.1. Méthode de réseaux de neurones profond……...35
3.2. Les séparateurs à vaste marges……36
3.2.1. Notion de base : Hyperplan, marge et support vecteur…….37
3.2.2. Linéarité et non-linéarité……...38
3.3. Procédure de validation……...39
4. Conclusion……..40
chapitre IV : Implémentation et Résultats
1. introduction……41
2. Description des bases de données utilisées……...41
3. Selection d'attributs……..42
3.1. Etapes de filtrage de donnée avec weka……..42
3.2. Résultat de filtrage……...45
4. Extraction des attributs et classification……46
4.1. Resultat d'extraction……...46
4.2. Analyse des résultats d'extraction……..49
5. prédiction des classes par SVM et tests de performance avec la validation croisée……49
5.1. Resultats de prédiction par SVM……...49
5.2. Analyses des résultats de SVM……..……..51
6. Conclusion……..53
Conclusion Générale……..54
Bibliographie

Listes des figures

figure I.1. Principe de la sélection d'attribut dans un système de classification……11
figure I.2. Méthode filtre ……11
figure I.3. Approche enveloppe ……12
figure I.4. Procédure de sélection de variables ……..13
figure I.5. Principe de l'extraction d'attributs ……..15
figure II.1. Représentation d'un réseau de neurones profond……20
figure II.2. Illustration d'un perceptron multi-couches……21
figure II.3. Un schéma représentant la forme d'un réseau de croyance profond……...22
figure II.4. Mode de fonctionnement d'un auto-encodeur empilé……23
figure II.5. Réseau de neurones convolutionnel……24
figure II.6. Graphe de réseau de neurones récurrent……...25
figure II.7. LSTM bidirectionnel……25
figure II.8. Gated Reccurent Unit……...26
figure II.9. Autoencodeur covolutionnel appliqué sur une image RGB……28
figure III.1. étape de la méthode de conception……30
figure III.2. Autoencodeur avec plusieurs couches cachées……...32
figure III.3. Illustration d'encodage avec trois couches cachées……...33
figure III.4. Illustration de décodage du réseau précédent……34
figure III.5. Construction de DNN pour la classification……...36
Figure III.6. un hyperplan……...37
Figure III.7. Linéarité et non-linéarité……38
Figure III.8. Schéma de validation croisée avec 10-fold……40
figure IV.1. lancer Weka……43
figure IV.2. Ouverture d'un fichier……43
figure IV.3. Choix de l'algorithme voulu……...44
figure IV.4. Choix de l'algorithme……44
figure IV.5. Méthode d'évaluation……45
figure IV.6. Les taux de precisions des deux méthodes de classification ……..54

Liste Des Tableaux

Tableau II.1. Différence entre un GRU et un LSTM……..24
Tableau IV.1. Caractéristiques des bases de test……...41
Tableau IV.2. Résultats de la sélection sur les bases de test……..44
Tableau IV.3. Extraction et classification pour la base de cancer de prostate……...46
Tableau IV.4. Extraction et classification pour la base de cancer de colon……..46
Tableau IV.5. Extraction et classification pour la base de leukemia……..……...47
Tableau IV.6. Extraction et classification pour la base de CNS…….……..47
Tableau IV.7. Résulats de prédiction par SVM pour la base de cancer de prostate……49
Tableau IV.8. Résulats de prédiction par SVM pour la base de leukemia…….50
Tableau IV.9. Résulats de prédiction par SVM pour la base de cancer de colon……..51
Tableau IV.10. Résulats de prédiction par SVM pour la base de CNS……..……51
Tableau IV.11. Résultats par SVM sur les bases originales……52

Liste Des Acronymes

AE : Auto-Encoded

CAE : Convolutional Auto-Encoded

CNN : Convolutional Neural Network

DBN : Deep Belief Network

DNN : Deep Neural Network

DST-NN : Deep Spatio-Temporal Neural Network

GRU : Gated Recurrent Unit

K-PCA : Kernal Principal Component Analysis

LSTM : Long Short-Term Memory

MD-RNN : MultiDimensional Recurrent Neural Network

MLP : MultiLayer Perceptron

PCA : Principal Component Analysis

RBM : Restricted Boltzmann Machine

RNN : Recurrent Neural Network

SAE : Stacked Auto-Encoded

SVM : Support Vector Machine

Introduction Générale

Le progrès scientifique cause l'augmentation de taille des bases de données dans les applications réelles dans tous les domaines comme la médecine, le Web, l'industrie...etc. Ces bases de données deviennent de plus en plus chargées et volumineuses, ce qui engendre une difficulté à les traiter par un être humain, et même le traitement par les machines devient couteux en terme de temps et d'espace mémoire. Les bases de données sont vraiment nécessaire pour tous types de recherche, surtout dans le domaine de classification, afin d'avoir une meilleure classification il faut un très bon système de prédiction, la prédiction de classe peut s'avérer très compliquée si une base est volumineuse, c'est ce qui nous attirer l'intention pour chercher sur les méthodes de réduction de dimension.

Après avoir cherché nous avons trouvé plein de problèmes liés aux bases de données volumineuses, et le problème majeur s'agit du fléau de dimensionnalité qui est un obstacle dans l'apprentissage automatique. Le fléau de dimensionnalité est une problématique majeure qui apparait lorsqu'on se retrouve avec un grand nombre d'attributs relativement au nombre d'instances, on peut aussi dire qu'il y a une explosion dans la taille du jeu de données nécessaire à l'apprentissage du modèle lorsqu'on augmente le nombre d'attributs en entrée.

Si on ne traite pas le problème de fléau de dimensionnalité, un problème de sur-apprentissage se déclenchera, le sur-apprentissage s'agit tout simplement d'un excès de données qui perturbe l'identification ou dans notre cas la classification. La grande majorité de recherches précédentes s'intéresse à la réduction de dimension pour la reconnaissance d'image ou de voix ainsi que la catégorisation des termes, rare sont les recherches qui se sont intéressées à la réduction des attributs dans les bases de données génétiques.

Dans le but de réduire la dimension de plusieurs données génétique, on a appliqué dans notre projet de fin d'étude divers techniques de réduction de dimension, on cite la sélection et l'extraction des attributs qui sont avérer très satisfaisant car ils ont réussi à donner des classifications presque parfaite, cela dit que le système de réduction était très efficace et fiable.

Tant dis que la deuxième phase de notre projet consiste à faire la prédiction ou bien la classification des données réduites dans la première phase de travail, deux technique également ont étés utilisés, la première est la classification en utilisant un réseau de neurones profond qui se base sur une architecture d'un Autoencodeur empilé, la deuxième technique de classification est l'utilisation de l'algorithme RBF-SVM très connu pour la classification binaire.

En dernier une méthode de jugement était nécessaire pour savoir la fiabilité du système et nous avons utilisé la Cross-validation avec k-fold.

Le plan de notre travail est comme suite, tout d'abord dans le premier chapitre nous avons défini les types de classification et leurs relations avec les données de grande dimension pour parler dans la suite du même chapitre des techniques de réduction de dimension.

Le deuxième chapitre détaille les architectures de l'apprentissage automatique profond qui sont les réseaux de neurones profond bien évidemment, car ils seront combinés avec les techniques du premier chapitre pour une meilleure réduction de dimension.

Le troisième chapitre explique la méthode proposée pour la réduction de la dimensionnalité et la classification des données à grande dimension, tant que le quatrième et dernier chapitre contient les détails sur les outils utilisés, les bases de test ainsi que les résultats obtenues.

Chapitre I
Réduction de la dimensionalité pour la classification

1. Introduction

Le progrès scientifique permet l'augmentation de la taille des bases de données qui sont devenue difficile à traiter par un humain dans un espace de temps raisonnable ce qui a rendu la tâche de prise de décision beaucoup plus complexe pour un opérateur humain. Les bases de données volumineuses ont causé plusieurs problèmes comme le fléau de dimensionnalité qui nécessite une réduction de dimension afin d'éviter le sur-apprentissage, des techniques de réduction de dimension sont conçu pour régler ce problème ainsi que pour faciliter la classification des données.

Ce chapitre va présenter les différents types de classification, on va spécifier la classification des données de grande dimension ainsi que les techniques de réduction de dimension.

2. Classification

La classification consiste à organiser un ensemble d'objets en plusieurs classes selon des critères prédéfinis, chaque classe est définie comme un sous ensemble d'objets à classer, ce dernier est composé de plusieurs éléments que chacun d'eux est définit par p variables descriptives.

Le but de la classification est d'assigner un vecteur d'élément X à une des K classes préfinies tout en se basant sur les attributs qui les décrits, et en définissant des labels $Y \in \{1,\ldots, K\}$ pour chaque élément X tel que : $Y = i$ si X appartient à la $i^{ème}$ classe.

Le classificateur génère des règles de décisions α en associant à chaque vecteur $X \in R^p$ un label $y \in \{1,\ldots, K\}$:

$$\alpha : \quad R^p \rightarrow \{1,\ldots,K\} \qquad X \rightarrow y$$

[El ferchichi, 2013]

Selon la nature des données d'apprentissage on distingue trois types d'apprentissage :

- Apprentissage supervisé.
- Apprentissage non-supervisé.
- Apprentissage semi-supervisé.

2.1. Classification supervisée

Les données d'apprentissage de la classification supervisée sont disponibles pour apprendre les règles de décisions afin de distinguer les K classes de données, les observations d'apprentissage supervisent la construction du classificateur.

La classification supervisée se base sur deux principaux aspects : aspect descriptif et aspect décisionnel.
Le premier aspect tente de trouver une représentation pour interpréter des groupes grâce aux variables explicatives mais la complexité de l'espace de la représentation peut rendre la recherche de la représentation difficile surtout quand les données qui dépassent la dimension 3. Quant au deuxième aspect, il vise la définition de la meilleure affectation d'un nouvel individu, dont on ne connait que les valeurs des variables explicatives qui le décrivent, l'aspect décisionnel est particulièrement intéressant dans les domaines ou la notion de diagnostic est essentielle.
Dans la classification supervisée on possède un ensemble d'apprentissage qui contient les données sous forme de couples (Observations, Labels) ou l'Observation est le vecteur de la donnée et le Label est la classe à laquelle appartient notre donnée.**[El ferchichi, 2013]**
La classification supervisée a pour mission de prédire l'appartenance des observations décrites par des variables explicatives à une classe définie à priori tout en utilisant le jeu de données.
Afin d'aborder la classification supervisée il existe deux approches principales :

2.1.1. Approche générative

Cette approche modélise la densité de chaque classe, l'estimation de la densité peut être soit paramétrique et est basée sur un modèle de mélange gaussien ou bien non-paramétrique qui est basée sur la méthode de noyau par exemple.
L'approche générative possède deux méthodes d'analyses : l'analyse discriminante quadratique et l'analyse discriminante linéaire.**[El ferchichi, 2013]**

2.1.2. Approche Discriminante

Cette approche a pour but d'estimer directement la probabilité à posteriori en minimisant le cout de classification, l'approche discriminante est efficace sur le plan pratique et le meilleur exemple pour cela est bien les séparateurs à vaste marge (SVM) mais aussi les K plus proches voisins et les arbres de décisions.**[El ferchichi, 2013]**

2.2. Classification non supervisée

Le jeu de données de la classification des données non supervisée ne contient aucune information sur la distribution des données d'apprentissage, ni le nombre de classe ni leurs labels donc le classificateur se base sur la notion de similarité entre les attributs pour générer ses règles de décisions en identifiant des groupes d'objets similaires surnommés **clusters** à partir d'un ensemble de données sans en connaitre au préalable sa structure. C‘est à dire que la classification non supervisée essaye de prédire les labels à partir des données existantes qui ne

sont d'autres que les variables explicatives mais la plus grandes difficulté reste l'ignorance de nombre groupes ou classes.**[El ferchichi, 2013]**

2.2.1. La notion de similarité

Cette notion consiste à évaluer les ressemblances ou les relations entre les ensembles de données, afin d'ordonner ses ensembles, les hiérarchiser ou encore en extraire des invariants. La notion de similarité était l'objet de plusieurs recherches dans différent domaines comme l'analyse des données, l'apprentissage symbolique ou même la reconnaissance, donc les fonctions de similarité sont conçues sous multiples formes pour différents buts poursuivis, et ceci afin de structurer des ensembles de données selon leurs ressemblance sous forme d'un ensemble de classes à la fois homogène et contrasté tout en maximisant la mesure de similarité intra-classe et/ou en minimisant la mesure de similarité inter-classe.

La classification des fonctions de similarités se base sur le langage utilisé pour la représentation des similarités, les ressemblances sont quantifiées sous forme de valeur dans les similarités numérique, tant dis que les similarités symboliques les caractérisent.

Les similarités sont classifiées selon le langage de description des données utilisées qui sont représentées soit par un type proportionnel ou relationnel. On peut aussi classifier les similarités en deux, la première s'appelle similarité non-informée qui n'utilise que les données et la deuxième connue sous le nom de similarité informée qui utilise les connaissances à priori.**[El ferchichi, 2013]**

Avec les connaissances des buts poursuivis, les informations sur le contexte de travail qui peuvent être statistiques ou symboliques permettent de sélectionner la meilleure fonction de classification pour le cas courant ou d'en concevoir une spécifique.

2.2.2. Les approches classique pour la classification non-supervisée

2.2.2.1. Méthode hiérarchique

La classification hiérarchique est une suite de partitions emboitées ordonnée, cette méthode possède deux approches qui sont :

- Méthodes ascendantes

Cette méthode consiste à fusionner entre les points qui sont considérer comme des classes afin de concevoir de nouvelles classes.**[El ferchichi, 2013]**

- Méthodes descendantes

Cette méthode considère que tous les points sont une seule classe, et pour avoir des classes raffinées on divise successivement les classes du début.**[El ferchichi, 2013]**

2.2.2.2. Méthode de partitionnement

Au lieu de produire une structure de classe, la méthode de partitionnement produit une seule partition de données et crée les classes en optimisant une fonction objective qui est définie d'une façon locale ou globale.
Chaque partition est représenté par un centre ou centroïde, et chaque objet est affecté au groupe dont il est le plus proche. **[El ferchichi, 2013]**

- K-medoids : est un algorithme de clustering qui tente de minimiser la somme des différences entre les points étiquetés comme étant dans un cluster et un point désigné comme le centre de ce cluster.[2]
- K-means : est une méthode de partitionnement de données et un problème d'optimisation combinatoire, le problème dévise les points en K clusters afin de minimiser l'erreur quadratique totale. [1]

2.2.2.3. Modèle de mélange

En supposant que l'on dispose d'un ensemble d'objets qui a été obtenu en fusionnant plusieurs sous populations qui suivent chacune une loi de probabilité propre.**[El ferchichi, 2013]**

2.2.2.4. Approche neuromémitique : les cartes auto-organisées de Kohenen

Cette approche consiste à transformer des signaux de départ de plusieurs dimensions en signaux à une ou deux dimensions, c'est-à-dire projeté les données initiales sur un espace discret et réguliers de faible dimension.**[El ferchichi, 2013]**
Les relations de voisinage entre formes d'entrées sont inconnues dans l'apprentissage non-supervisé, c'est l'observation des voisinages produits par la carte qui vont interpréter les données initiales tout en définissant la notion de formes proches dans l'espace initial.**[El ferchichi, 2013]**

2.3. Classification semi-supervisée

Les données non labelisées sont faciles à obtenir, tant dis les données labelisées demande un effort humain couteux pour leur procuration, pour cela la classification semi-supervisé a attiré l'intention ces dernières années.
La classification semi-supervisée se situe entre l'apprentissage supervisé et l'apprentissage non-supervisée, l'ensemble des données peut être devisé en deux parties dans la première on trouve des points pour lesquels les labels sont fournies, et dans la deuxième parties on trouve des points qui n'ont pas de labels connus.
On peut avoir des contraintes qui informent que certains points possèdent (ou pas) le même label, de ce fait l'apprentissage semi-supervisé peut être considéré comme un problème non supervisé guidé par des contraintes.

Etant donné que l'apprentissage supervisé n'utilise que des données labelisées, l'apprentissage semi-supervisée espère obtenir plus de précision à partir de données non labellisées.
Les données labellisées sont rares ce qui fait que les différentes méthodes qui entament la classification semi-supervisée se basent sur des principes strictes et on doit choisir la méthode appropriée à notre problème.**[El ferchichi, 2013]**

3. Classification des données de grande dimension

Au fil du temps, le nombre de données ainsi que leur dimension ne cesse d'augmenter à cause du développement de la technologie surtout pour les données des clients dans les entreprises, les données génétiques, les données textuelles et pleines d'autre… Quand la dimension augmente les propriétés de l'espace euclidien 2D et 3D deviennent sans grande importance ce qui crée des difficultés pour la manipulation et ceci est appelé le phénomène de fléau de dimensionnalité.**[El ferchichi, 2013]**

3.1. Fléau de dimensionnalité

Le terme fléau de la dimensionnalité a été évoqué la première fois dans le cadre d'optimisation d'une fonction par une recherche exhaustive dans un espace discrétisé par Bellman**[Verleysen, 2003]**.
En cas d'absence d'hypothèse simplificatrice et de présence d'un espace de données et d'un taux d'erreur, le nombre de données qu'il faut pour l'estimation d'une fonction multi-variables est autant plus grand que la dimension elle-même et tous ceci est référer par le fléau de dimensionnalité.
Puisque que plusieurs données sont indispensables pour l'optimisation et l'approximation des fonctions, il faut savoir les différences entre une petite et une grande dimension dans l'espace, ainsi que le nombre de points nécessaires pour effectuer un apprentissage dans un espace de grande dimension.**[El ferchichi, 2013]**

3.1.1. Limite entre une grande et une petite dimension de l'espace

Contrairement aux méthodes d'apprentissage, les performances de ces dernières n'ont pas de problèmes avec les espaces de grande dimension, ce qui est utile car les problèmes réels sont généralement de grande dimension.
La redondance des variables n'est pas caractériser car les variables de l'espace d'entrée sont liées, ce qui cause un problème dans l'espace de dimension, appelé aussi le phénomène de l'espace vide, il est considéré comme un effet de la grande dimension qui n'aide pas à la contribution de la représentation habituelle, donc c'est une caractéristique de fléau de dimensionnalité.

Intuitivement, l'intégrité d'une fonction gaussienne est dans un tour limité au tour du centre, mais quand la dimension augmente son volume devient entre les marges et non pas un centre, ces propriétés locales empêche la fonction gaussienne d'être bien présenté dans un espace de grande dimension.**[El ferchichi, 2013]**

3.1.2. Nombre de points nécessaires pour l'apprentissage dans un espace de grande dimension

L'apprentissage dans des espaces de grande dimension a besoin d'un grand nombre de données qui croit exponentiellement avec la dimension de l'espace.
Les problèmes réels ne souffrent pas vraiment du problème de fléau de dimensionnalité, et puisque les données sont dans une dimension plus petite il est nécessaire de réduire la dimension dans l'apprentissage en grande dimension.**[El ferchichi, 2013]**

3.2. Fléau de dimensionnalité en classification

Le fléau de dimensionnalité affecte l'apprentissage de données de grande dimension qui a besoin d'un grand nombre d'observation lié à la dimension pour être efficace.
Si un classificateur de classification supervisée est trop complexe on peut avoir un problème de sur-apprentissage, donc la capacité de généralisation de classificateur sera moins fiable.
Lorsque le jeu d'apprentissage ne représente pas le processus qui le génère, les règles de décisions ne traiteront pas correctement les nouvelles données; et si le modèle est trop complexe l'erreur de prédiction sur le jeu de validation croit encore plus.
Le problème de sous-apprentissage se présente en cas de classificateur simple et son efficacité sera réduite sur les données d'apprentissage et ceux du test. De ce fait, la détermination du bon degré de complexité du classificateur est importante pour garantir son efficacité.
Les méthodes de classification basée sur les fonctions noyaux ne sont pas robustes contre le fléau de dimensionnalité, car un noyau ne fait que mesurer la similarité entre deux observations.
La dimension de données est augmentée artificiellement par un processus d'acquisition, alors que les données sont concentrées dans une plus petite dimension. Les données fournies en grandes dimension sont modélisées par un nombre d'observation beaucoup plus faible que la dimension.
Il faut minimiser les variables d'espace d'entrée à une dimension intrinsèque plus faible que la dimension initiale afin d'éviter les problèmes liés au fléau de dimensionnalité.
La grande dimension de l'espace des attributs peut contribuer à la classification des données.
Les méthodes de discriminations SVM, augmentent artificiellement la dimension des données pour faciliter leurs discriminations en exploitant le phénomène précédent.**[El ferchichi, 2013]**

4. Réduction de dimension

Les nouvelles méthodes d'apprentissage donnent la possibilité d'en déduire les informations depuis les bases des données énormes et permettent de les exploiter, donc la classification aide à établir une forte relation entre les variables (attributs) avec leurs étiquettes (classes) et ceci à partir d'un échantillon d'apprentissage.

Il faut mentionner que les variables qui caractérises les données affectent pleinement le travail du classificateur pour générer les règles de décision, on prend comme exemple les variables qui correspondent à du bruit ou les variables redondantes qui n'ont pas vraiment de la valeur mais qui peuvent aussi nuire ou causer des problèmes lors de la classification ; ceci ne peut pas être pris comme règle stricte car ces données peuvent être très importantes si on les combinent avec d'autres données.

Plusieurs problèmes sont rencontrés lors de l'analyse et l'exploration des données de grande dimension qui peuvent dégrader la performance de classificateur, car les données non pertinentes affectent la classification supervisée ou non-supervisée.

Pour cela, il faut traiter les données au préalable pour une meilleure compréhension des informations afin de faciliter la classification, car l'identification des variables pertinentes est nécessaire pour le succès de la classification ; le traitement le plus logique est bien la réduction de dimension car elle vise :

- Identification des attributs pertinents
- Amélioration de la classification
- Facilitation de la visualisation et de la compréhension des données
- Réduction de l'espace de stockage nécessaire
- Réduction du temps d'apprentissage et d'utilisation

La réduction de dimension est la solution optimale pour résoudre les problèmes de l'apprentissage automatique par :

- La sélection d'attributs pertinents
- L'extraction des attributs.

[El ferchichi, 2013]

4.1. Sélection d'attributs

Le domaine d'apprentissage automatique et les systèmes de fouilles de données ont exigé une réduction de dimension de leurs grandes bases de données avant l'étape de classification afin d'éviter les problèmes de fléau de dimensionnalité.**[El ferchichi, 2013]**

4.1.1. Définition

La sélection d'attributs est un sujet de recherche très actif depuis une dizaine d'année dans différents domaines, cette méthode a pour but de filtrer des vecteurs d'attributs de manière à en extraire des informations discriminantes et pertinentes afin d'améliorer la qualité d'un système.

Elle joue un grand rôle dans la classification, car lorsqu'on a un grand nombre de variables, certains peuvent êtres peu significative ou non-pertinents au regard de l'application considérée, La sélection d'attributs consiste à sélectionner un sous-ensemble de variables pour que la règle de classement augmente et elle facilite l'étape d'apprentissage et réduit la complexité des algorithmes ainsi que les temps de calcul. Voire la figure I.1.**[Samb, Camara et Ndiaye, 2014]**

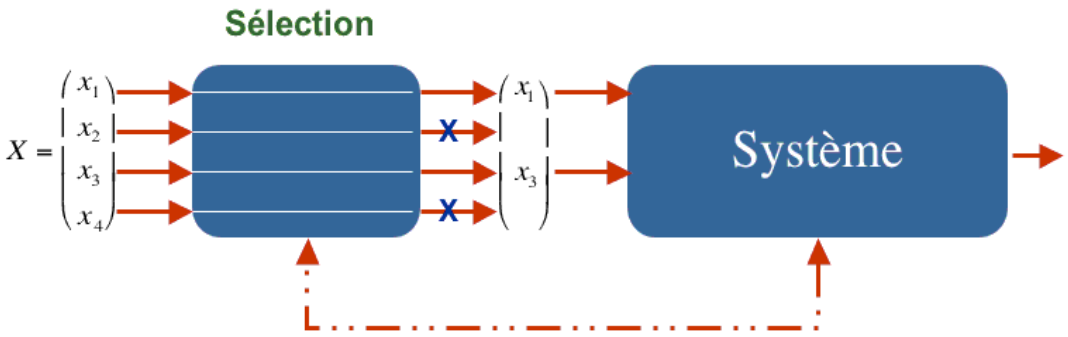


Figure I.1. *Principe de la sélection d'attribut dans un système de classification*

la sélection d'attributs ne veut pas forcément dire qu'on va prendre que les variables pertinents mais il peut aussi y avoir des éléments non pertinents mais qui avec d'autres éléments auront plus de valeurs et contribuent à l'évolution du système, cette méthode de réduction tente d'avoir le sous-ensemble de l'ensemble original le plus petit tout en faisant attention que la précision de la classification ne dégrade pas ainsi la distribution des classes soit le plus proche possible de la distribution originale.**[El ferchichi, 2013]**

4.1.2. Techniques de sélection d'attributs

Les techniques de sélection d'attributs sont selon leur manière d'interaction avec le classificateur divisées en trois catégories :

4.1.2.1 Méthodes filtres

L'interaction est directement avec le jeu de données afin de fournir une pondération, un classement ou un sous-ensemble de variables. Cette méthode est rapide et indépendante de la méthode de classification.**[Samb, Camara et Ndiaye, 2014]**

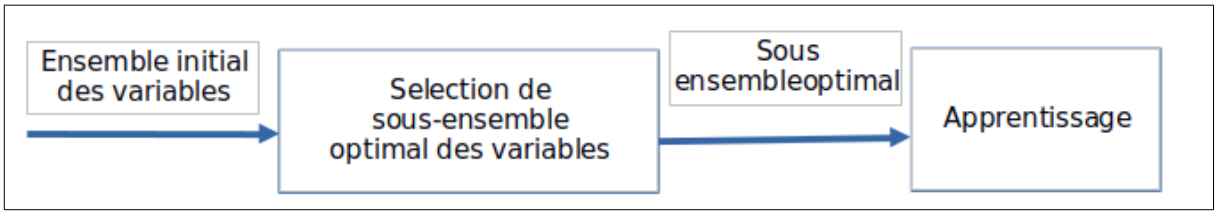


Figure I.2. *Méthode filtre*

4.1.2.2 Méthodes enveloppes

Cette méthode effectue une recherche dans un espace de sous-ensemble de variable, guidée par le résultat du modèle ce qui fait qu'elle n'ignore pas l'influence des attributs sélectionnés sur la classification, cette méthode donne souvent des résultats meilleures que ceux de la méthode filtre mais elle prend plus de temps que la méthode précédente.**[Article]**, **[El ferchichi, 2013]**

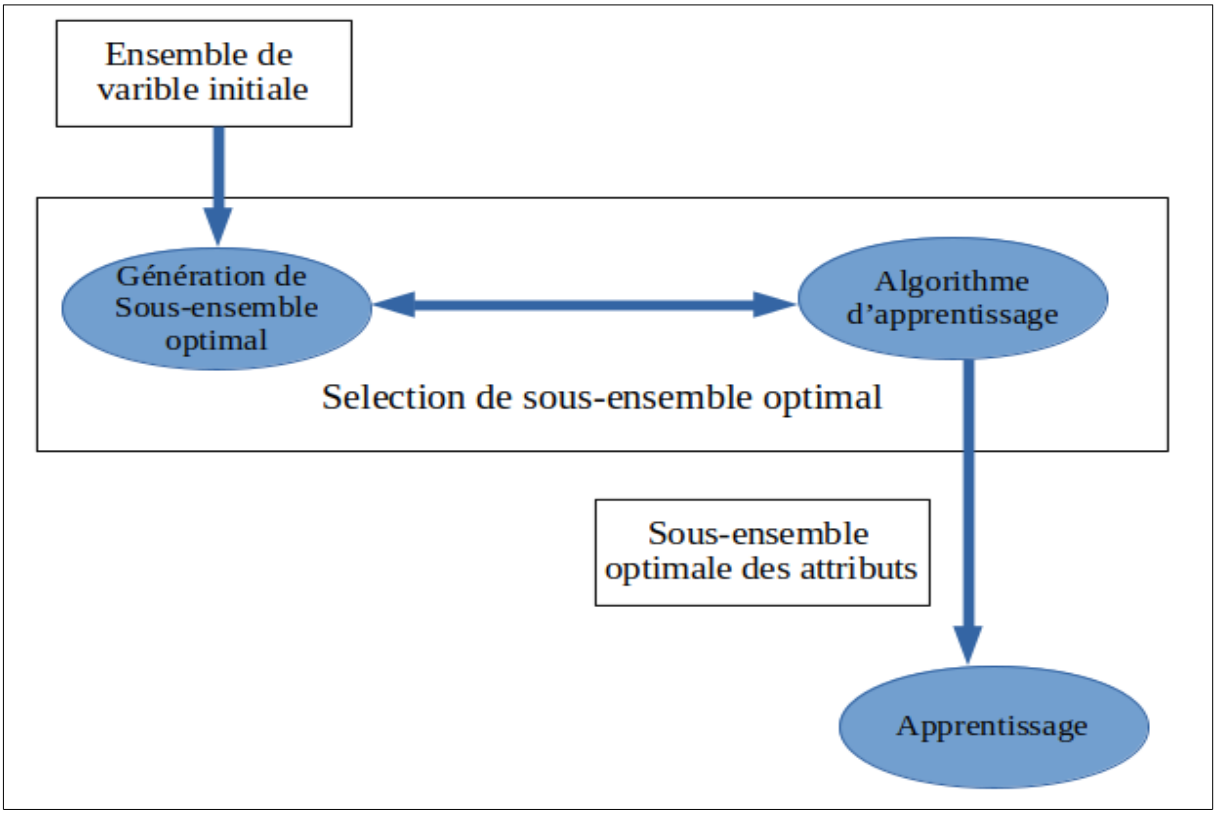


Figure I.3. Approche enveloppe

4.1.2.3. Méthodes embarquées

Cette méthode combine le processus d'exploration avec un algorithme d'apprentissage sans l'étape de validation afin de minimiser le nombre d'attributs au même temps maximiser la qualité de l'ajustement, tout en utilisant l'information interne du modèle de classification, elle est proche de la méthode enveloppe, mais la différence est que le classificateur en plus d'évaluer les sous-ensemble de candidats il sert aussi à guider le mécanisme de sélection.

La méthode embarquée est plus rapide en calcul par rapport à la méthode enveloppe et elle est plus robuste avec les problèmes de sur-ajustement.**[Samb, Camara et Ndiaye, 2014**

4.1.3. Les étapes de procédure de sélection

La procédure de sélection d'attributs se fait en quatre étapes illustrées dans la figure suivante :

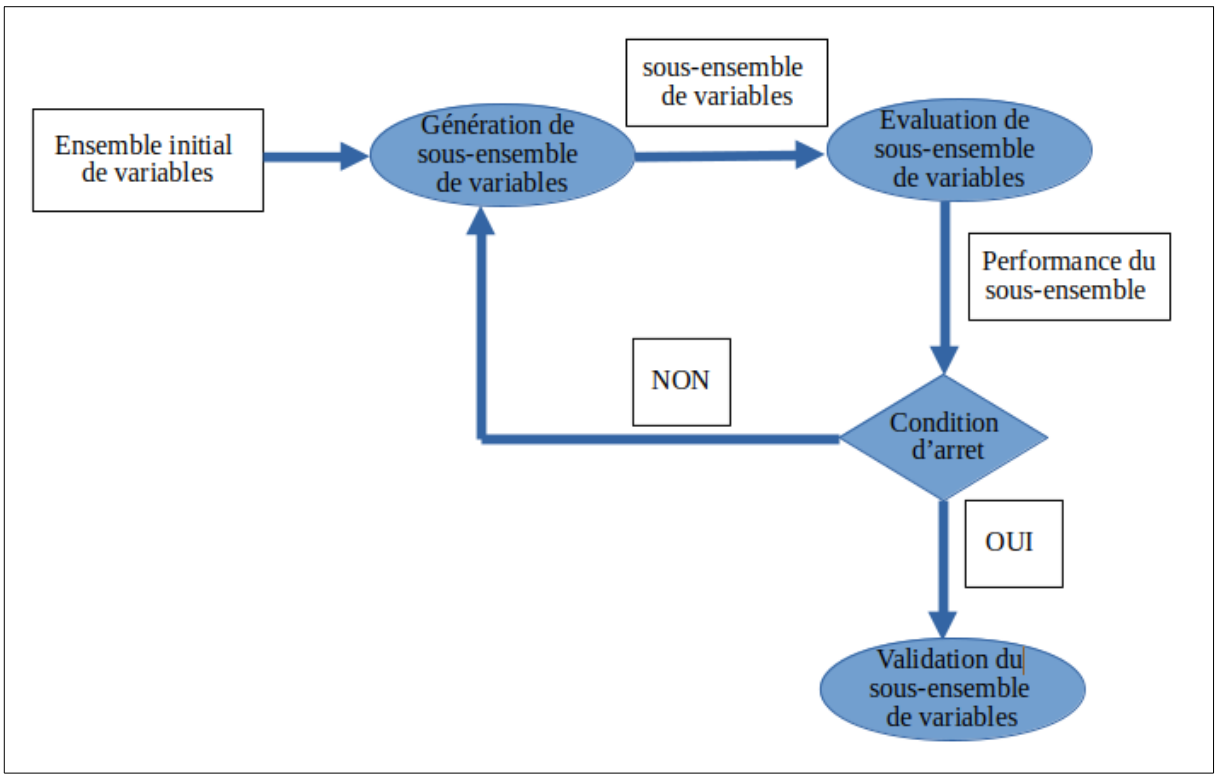


Figure I.4. *Procédure de sélection de variables*

4.1.3.1. Procédure de génération

La procédure de génération est une procédure de recherche pour avoir des sous-ensemble d'attributs qui vont être juger tout en se basant sur des critères strictes ; la méthode de sélection peut être initialisée soit par un ensemble d'attributs vide ou bien un sous-ensemble d'attributs aléatoire et elle peut même prendre tous l'ensemble original d'attributs.
A fur et à mesure de cette procédure chaque variable ou ensemble de variables qui améliore le critère d'évaluation sera pris et rajouté dans les variables sélectionnés, tant dis que celui qui dégrade l'évaluation sera logiquement éliminé.
On suppose qu'on a un ensemble initiale avec *n* variable, donc on aura 2^{n-1} combinaisons possibles; le 2 représente les décisions possible pour un attribut : sélectionner l'attribut ou ne pas le sélectionner, ce qui fait qu'on aura un nombre inconnu de variable sélectionné de cela la procédure de génération peut être classé en trois approches :

- Génération complète : consiste à faire une recherche sur tout l'ensemble de variables afin de trouver le sous-ensemble optimal parmi les solutions possible, cette approche et de l'ordre $O(2^N)$
- Génération séquentielle : après chaque itération de cette approche on considère que tout l'ensemble d'attributs restant est l'ensemble à sélectionner, la sélection est donc incrémentale dans un espace de recherche d'ordre $O(N^2)$.
- Génération heuristique : cette approche se base sur les algorithmes d'optimisation comme l'algorithme génétique ou recherche taboue, elle s'effectuent sur un espace de recherche d'ordre $O(2^N)$ mais elle n'évalue pas toutes les solutions possibles de cet espace ce qui pousse à imposer un nombre d'itération maximal dans le but de limiter le temps de calcul. **[El ferchichi, 2013]**

4.1.3.2. Fonction d'évaluation

Une fonction d'évaluation est vraiment nécessaire pour juger la pertinence des variables et leur pouvoir discriminant afin d'améliorer les performances d'un système d'apprentissage ce qui rend la fonction d'évaluation une mesure de pertinence.

Le choix de sous-ensemble optimal résultant de la procédure de recherche dépend de la fonction d'évaluation utilisée cette dernière se base généralement sur les matrices de dispersion inter et intra classes ainsi que les distances probabilistes et les entropies ces deux dernières sont basées sur l'information mutuelle entre les attributs et les classes.

Les fonctions d'évaluation se devisent en cinq catégories de mesure :

- Mesure de distance : c'est une mesure de discrimination comme la distance euclidienne en cas de classification binaire
- Mesure d'information : cette catégorie se base sur l'information apporté par un attribut sur sa classe, on prend comme exemple l'entropie.
- Mesure de dépendance : on prend en considération la corrélation entre un attribut et une classe. L'attribut qui possède la corrélation la plus élevé sera préféré par rapport aux autres attributs.
- Mesure de consistance : un sous-ensemble avec un taux de consistance élevé veut dire que ses variables permettent une bonne prédiction de classes.
- Mesure d'erreur de classification : ce type de fonctions d'évaluation est utilisé par les méthodes enveloppes, elle consiste à prendre que les attributs qui minimisent l'erreur de classification afin de garantir une meilleure précision de classification mais cela au prix d'un taux de calcul couteux et un grand espace mémoire.**[El ferchichi, 2013]**

4.1.3.3. Condition d'arrêt

le critère d'arrêt doit être lié avec la procédure de recherche et la fonction d'évaluation afin d'avoir le meilleur sous-ensemble sélectionné, le critère le plus connu est bien celui du

seuil de nombre d'itération qui peut tous de même nuire a notre sélection qui peut être arrêter soit trop ou bien trop tard, on peut aussi utiliser le critère de limiter le nombre de variables à sélectionner, sinon le critère d'arrêt se base sur la fonction d'évaluation cette dernière sera contrôlée entre chaque deux itérations consécutive.**[El ferchichi, 2013]**

4.2. Extraction des attributs

L'extraction des variables est la deuxième technique pour réduire la dimension de grandes bases de données, cette technique combine entre l'ensemble de variables initiales et fait en sorte d'en produire d'autres meilleurs avec un nombre d'attributs réduit mais qui représente parfaitement l'ensemble original, la figure suivante explique ce principe .

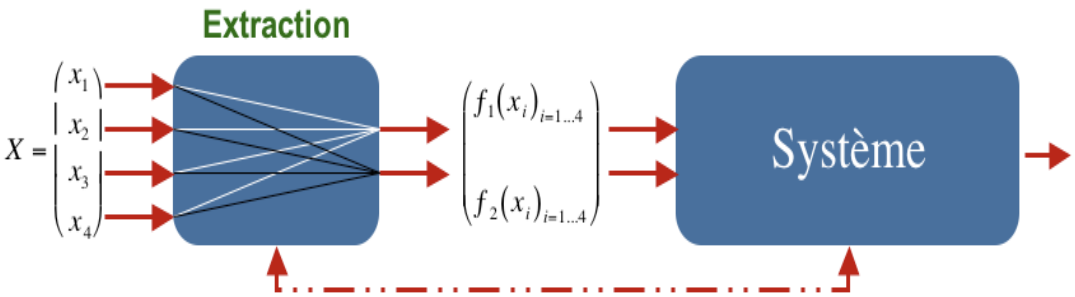


Figure I.5. *Principe de l'extraction d'attributs*

Les méthodes les plus connu de l'extraction des variables peuvent être regroupés en deux grandes catégories : les méthodes linéaires et les méthodes d'extraction non linéaire, dans la suite de ce chapitre on va présenter deux exemples pour chaque catégorie. [Extraction.pdf]

4.2.1. Méthodes linaires

Il existe cinq méthodes connues pour l'extraction linéaire :

- Analyse en composantes principales
- Factorisation de matrice non négative
- Analyse en composants indépendants
- Analyse discriminante linéaire
- Modèle de Fisher amélioré

Nous allons décrire seulement deux méthodes, la première sera l'analyse de composantes principales tant dis que la deuxième méthode choisie est l'analyse discriminante linéaire. [Extraction.pdf]

4.2.1.1. Analyse en composantes principales (ACP)

Cette méthode est utilisée en statistique, elle est aussi connu sou le nom de «transformée de karhunen-loève», elle consiste à réduire la dimension en tenant compte de garder le maximum d'information via une projection qui maximise la variance et minimise l'erreur quadratique moyenne de la reconstruction.

Cette méthode devient difficile à appliquer, car il faut chercher les éléments propres de la matrice que covariance mais cette dernière dépend des données initiales qui sont généralement volumineux. [Extraction.pdf]

4.2.1.2. Analyse discriminante linéaire

Cette méthode est plus connue sous le nom de «analyse discriminante de Fisher», elle tente de maximiser la distance entre les classes et minimiser la variance intra-classe dans le but d'assurer une classification meilleure ou plus efficace.

Cette méthode et la méthode citée précédemment sont liées car elles essayent de trouver une meilleure représentation des données tout en faisant des combinaisons linéaires. L'inconvénient de cette méthode est qu'elle exige un grand nombre de données pour une bonne généralisation sinon on va avoir un problème de sur-ajustement qui va nuire au nouvelles données, la solution pour ce problème est bien le modèle de Fisher amélioré.

4.2.2. Méthodes non linéaires

Les méthodes de réduction non linéaire possèdent trois catégories selon l'objectif visé :

- préservation des propriétés globales des données de faible dimension.
- préservation des propriétés locales des données.
- Réalisation d'un alignement globale de modèle linéaire.

4.2.2.1. Méthodes globales

Ce genre de méthode essaye de réservé l'ensemble de données dans espace d'apprentissage faible, il existe plusieurs techniques de cette méthodes on cite : isomap, les autoencodeurs multi-couches, les Kernel PCA …

La dernière méthode citée est l'analyse en composantes principales à noyaux (*en anglais* «Kernel Principal Componant Analysis»), c'est une technique basée sur les noyaux et sa performance dépend du choix de la fonction noyaux, ce genre de technique à encourager l'émergence de plusieurs méthodes on prend comme exemple les SVM.

Les Kernels PCA s'appliquent avec succès surtout pour la reconnaissance vocale. [Extraction.pdf]

4.2.2.2. Méthodes locales

Ce genre de méthode préserve les propriétés des données dans leur voisinage, en pensant que si on protège la propriété locale de chaque point alors la propriété globale de l'ensemble est automatiquement protéger, et ceci en utilisant les kernels PCA à l'aide d'un noyau local spécifique. [Extraction.pdf]

5. Conclusion

Dans ce chapitre nous avons vu la classification et ses trois types : supervisée, non supervisée et semi-supervisée; ainsi que leur rôle avec les données de grande dimension afin de discuter le problème de fléau de dimensionnalité qui nécessite une réduction de dimension, cette dernière possède deux techniques : la sélection d'attributs et l'extraction des attributs, ces deux dernière ont été entamées et expliquées.

Chapitre II
L’aprentissage profond

1. Introduction

L'apprentissage profond est un domaine de recherche en croissance rapide. Ce chapitre va présenter l'apprentissage profond ainsi que ses différentes architectures, est détaillera les plus réputées et les plus utilisées comme les DNNs et leurs classes.

2. L'apprentissage automatique

2.1. Définition

L'apprentissage automatique (*en anglais : Machine Learning*) est une référence de développement et d'implémentation de l'évolution d'une machine via un processus d'apprentissage afin de remplir des taches difficile à effectuer par les algorithmes classiques.
Arthur Samuel a dit que l'apprentissage automatique est de donner la capacité aux machines d'apprendre sans les programmer explicitement.
Selon Tom Mitchell, un programme qui apprend est un programme qui tire profit d'une expérience par rapport à une famille de tache pour une mesure d'efficacité, si cette dernière s'améliore avec l'expérience. **[Ben Ayed, 2015]**

2.2. Les réseaux de neurone

Les réseaux de neurones sont la modélisation mathématique du cerveau humain, et aussi la représentation graphique en associant des objets élémentaires connus sous le nom de neurones formels en réseaux qui se distinguent par l'architecture du graphe et son niveau de complexité, le type de neurone et l'objectif visé. **[Hopkins, 1982]**

2.3. Limitation des algorithmes classiques

La performance correcte des algorithmes classiques d'apprentissage automatique repose fortement sur des représentations de données appelées fonctionnalités.
Cependant, les fonctionnalités sont généralement conçues par des ingénieurs humains possédant une vaste expertise dans le domaine, et l'identification des fonctionnalités les plus appropriées à la tâche donnée reste difficile.

3. Apprentissage profond

L'apprentissage profond ou "Deep Learning" est une branche de l'apprentissage automatique qui a récemment émergé sur les bases de données volumineuses, la puissance de l'informatique parallèle et distribuée ainsi que les algorithmes sophistiqués.

Il a surmonté les limitations précédentes et a rapidement augmenté l'intérêt académique, et a beaucoup contribué à l'avancement d'ultime domaines que la communauté de l'intelligence artificielle a lutté plusieurs années sans pouvoir donner de bonnes résultats comme le domaine de reconnaissance vocale ou d'image, traitement de langage naturel et la traduction.

Les succès de l'apprentissage profond sont construits sur une base de détails algorithmiques significatifs et peuvent généralement être compris en deux parties : la construction et la formation d'architectures d'apprentissage profond.**[Seonwoo, Byunghan et Sungroh]**

3.1. Groupe d'architectures connues

3.1.1. Réseaux de neurones profonds

Les réseaux de neurones profonds ou DNNs (Deep Neural Network) possèdent une structure basic : une couche d'entrée, plusieurs couches cachées et une couche de sortie.

Quand les données d'entrée sont fournies aux DNN, les valeurs de sorties sont calculées séquentiellement tout au long des couches du réseau.

A chaque couche, le vecteur d'entrée comprenant les valeurs de sortie de chaque unité dans la couche inférieure est multiplié par le vecteur de pondération pour chaque unité de la couche courante afin de produire la somme pondérée, Ensuite, une fonction non linéaire, telle qu'une tangente sigmoïde, hyperbolique ou linéaire rectifiée, est appliquée à la somme pondérée pour calculer les valeurs de sortie de la couche.**[Seonwoo, Byunghan et Sungroh]**

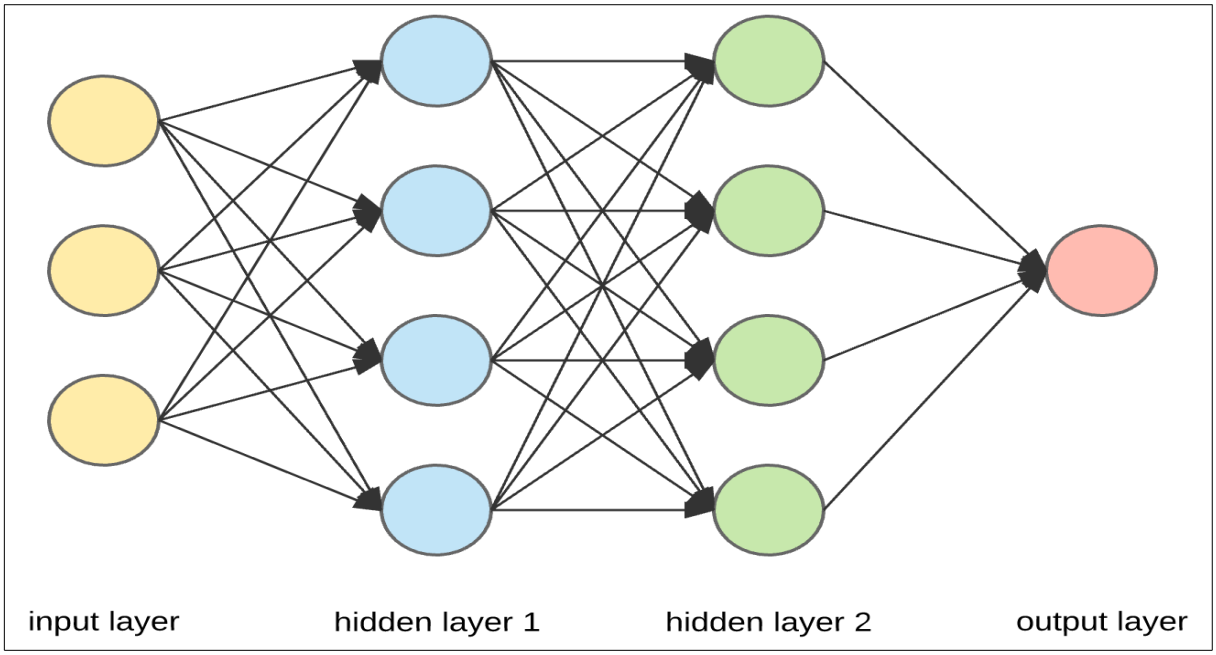


Figure II.1. *Représentation d'un réseau de neurones profond.*

Le calcul dans chaque couche transforme les représentations de la couche inférieure en représentations légèrement plus abstraites. En fonction des types de couches utilisées dans les DNN et de la méthode d'apprentissage correspondante, les DNN peuvent être classés en :

- Perceptron multicouche.
- Auto-encodeur empilé.
- Réseau de croyance profond…etc.

3.1.1.1. Perceptron Multicouche

Le perceptron multicouche ou MLP (MultiLayer Perceptron) possède une structure similaire aux réseaux de neurones habituels tout en contenant plus de couches empilées, sa technologie est utilisée en cas de présence de grands nombre de données étiquetées (apprentissage supervisé).

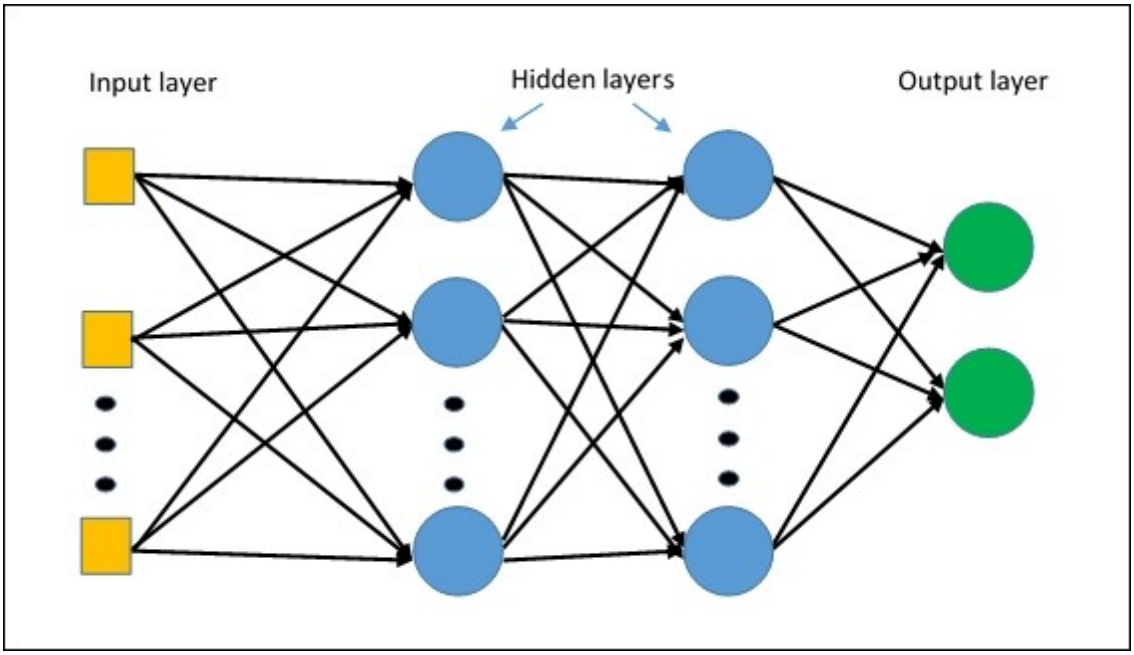


Figure II.2. *Illustration d'un perceptron multi-couches*

3.1.1.2. Réseau de croyance profond (DBN)

Un DBN (Deep Belief Network) est un réseau de neurones profond qui se comporte comme un modèle génératif dans lequel les couches de neurones sont entraînées les unes après les autres et chacune d'entre elles essaye de modéliser la distribution de ses entrées.
Pendant l'entrainement d'une couche il est considéré qu'elle forme une machine de Boltzmann restreinte avec la couche des unités visibles c'est-à-dire la couche qui précède la couche courante celle des unités cachées.
Chaque machine de Boltzmann donne des poids par divergence contrastive d'une façon non-supervisée, la couche entraînée est maintenant considérée comme une fonction déterministe qui peut calculer l'espérance d'activation de ses propres unités ainsi que celles de la couche précédente.

Les espérances sont les représentations des exemples d'entrée, la représentation devient de plus en plus abstraite en montant dans les couches, donc on aura une représentation complexe mais qui arrive quand même à expliquer la réalité qu'elle représente.
L'apprentissage des poids remplace la phase d'initialisation aléatoire des poids qu'on trouve dans les réseaux de neurones ordinaires.**[Lamblin,2007]**

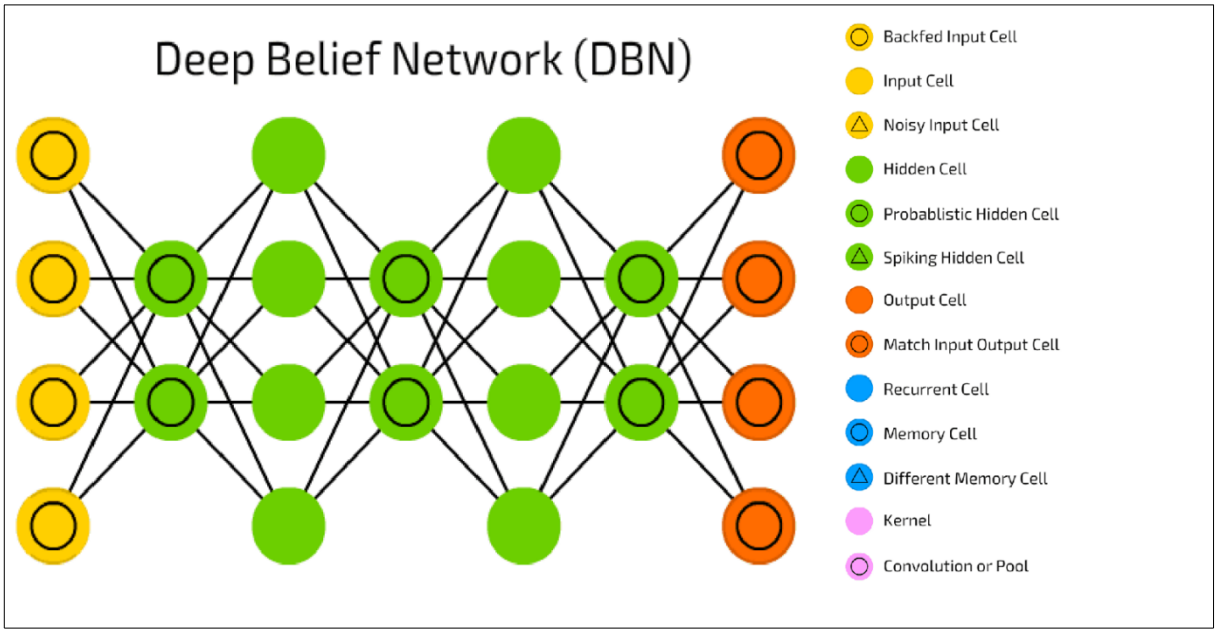


Figure II.3. *Un schéma représentant la forme d'un réseau de croyance profond*

On remarquer qu'un DBN possède le même nombre de nœuds dans la couche d'entrée et celle de sortie.

- Machine de Boltzmann restreinte

Les machines de Boltzmann restreintes connues en anglais sous l'abréviation RBM (Restricted Boltzmann Machine) sont des extracteurs de fonctionnalités utiles, ils sont principalement utilisés pour initialiser les réseaux de neurones profonds à action directe.

Les machines Boltzmann offrent une flexibilité cadre pour la modélisation des données avec latente variables.**[Hugo, 2009]**

3.1.1.3. Auto-encodeur empilé (SAE)

Les Stacked Auto-Encoder et DBN utilisent respectivement des Auto-Encodeur et des RBM en tant que blocs de construction des architectures.
La principale différence entre ceux-ci et MLP est que l'entraînement est exécuté en deux phases : une préformation non supervisée puis une mise au point supervisée. Tout d'abord, dans une préformation non supervisée, les couches sont empilées séquentiellement et

entraînées par une seule couche à la fois, tout en utilisant des données non marquées. Ensuite, dans un réglage fin supervisé, une couche de classificateur de sortie est empilée, et le réseau de neurones entier est optimisé en recyclant avec des données étiquetées.
Puisque la SAE et la DBN exploitent des données non étiquetées, ils peuvent aider à éviter le problème de sur-apprentissage.

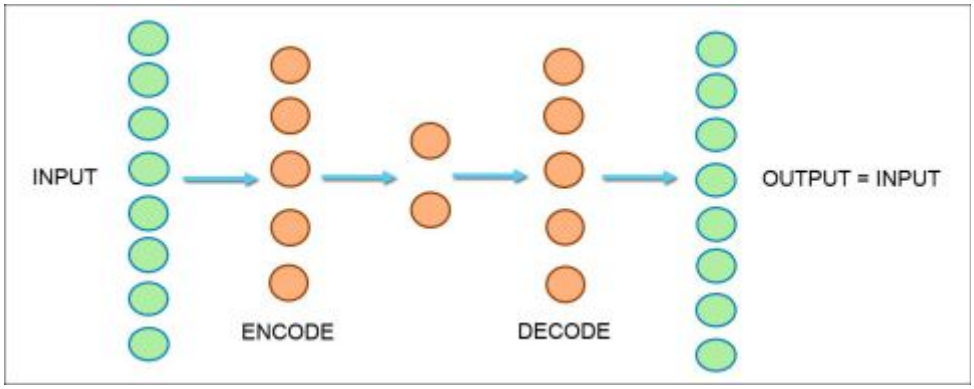


Figure II.4. *Mode de fonctionnement d'un auto-encodeur empilé*

Les DNNs possèdent une bonne aptitude pour analyser les données de grande dimension et ceci est prometteur pour la recherche dans différents domaines, ils peuvent mieux comprendre la nature des données grâce à leur méthode d'apprentissage par représentation hiérarchique, cette qualité peut être considéré comme limite au même temps car les caractéristiques hiérarchiques seule sont pris comme données brute tant dis que les caractéristiques conçues par l'homme sont fournies comme données d'entrée et non pas des données brutes.**[Seonwoo, Byunghan et Sungroh]**

3.1.2. Réseaux de neurones convolutionnels

Les CNNs (Convolutionnal Neuronal Network) traitent divers types de données en particuliers les images bidimensionnelles, ils sont inspirés par le cortex visuel du cerveau qui contient des cellules simples et d'autres complexes, les premières réagissent aux modèles primitifs dans les sous régions du stimuli-visuel, tant que les cellules complexes synthétisent l'information via les cellules simples afin d'identifier des formes plus complexes.
Les codes CNN sont appliqués pour imiter trois idées clés :

- La connectivité locale
- L'invariance à l'emplacement
- L'invariance de la transition locale.

La structure de base des réseaux de neurones convolutionnels est constituée de couches de convolution, de couches non linéaires et de couches de regroupement, afin d'utiliser des sous-régions de données hautement corrélées, des cartes de caractéristiques qui sont des groupes de sommes pondérées localement, sont obtenus à chaque couche de convolution en calculant des circonvolutions entre des correctifs locaux et des vecteurs de poids appelés filtres.

En outre, les filtres sont appliqués de manière répétée sur tout le jeu de données, pour améliorer l'efficacité de l'entraînement et réduire le nombre de paramètres à apprendre, et les couches non linéaires augmentent les propriétés non linéaires des cartes de caractéristiques.
On effectue un sous-échantillonnage maximal ou moyen des régions non chevauchantes dans les mappages d'entités pour chaque couche de groupement, ce sous-échantillonnage sans chevauchement permet aux réseaux de neurones convolutionnels de gérer des entités quelque peu différentes mais sémantiquement similaires et donc d'agréger des entités locales pour identifier des entités plus complexes.**[Seonwoo, Byunghan et Sungroh]**

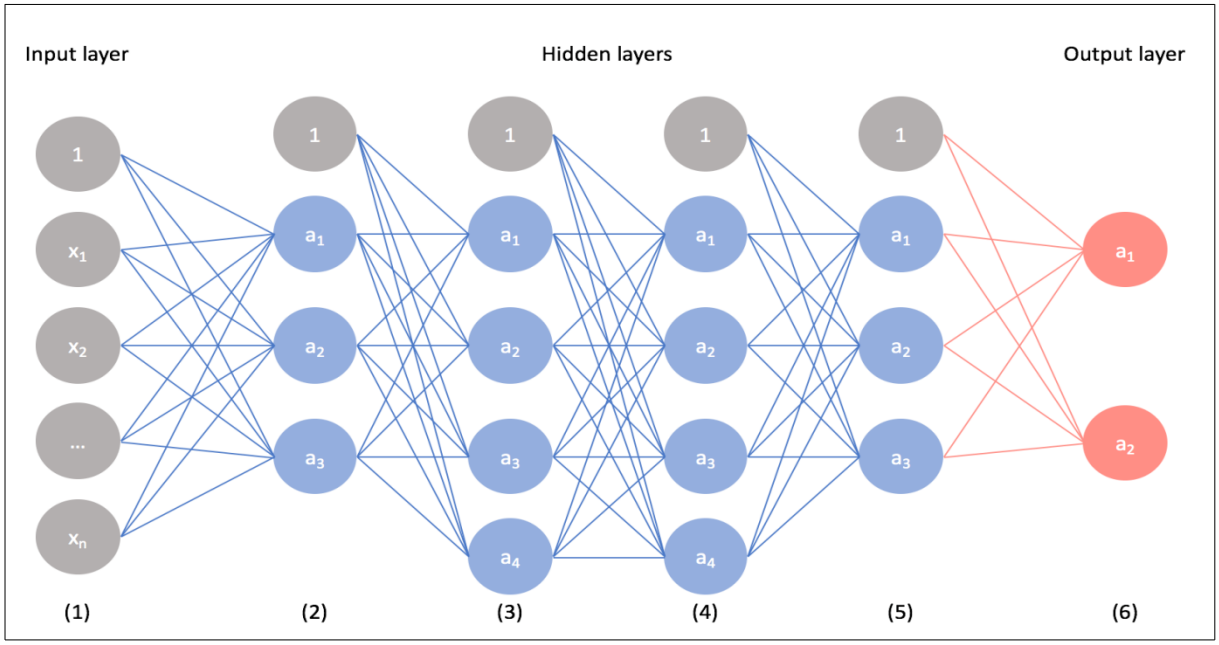


Figure II.5. *Réseau de neurones convolutionnel*

Grace à leur capacité exceptionnelle à analyser l'information spatiale et leur développement dans le domaine de la reconnaissance des objets, les CNNs sont les architectures d'apprentissage profond les plus réussies.

3.1.3. Réseaux de neurone récurrent

Les RNNs (Récurrent Neuronal Network) possèdent une structure avec une connexion cyclique les données d'entrée sont traitées séquentiellement et le calcul récurrent est effectué dans les unités cachées c'est-à-dire la ou la connexion est cyclique, donc les informations passées sont implicitement stockées dans des vecteurs d'état qui ne sont que des unités cachées, en utilisant ces vecteurs d'états on obtient par calcul la sortie pour l'entrée courante.
Afin d'éviter que la sortie de l'entrée courante ne soit affecté par une entrée passée ou futur, des RNNs bidirectionnels ont été conçus et utilisés.**[Seonwoo, Byunghan et Sungroh]**

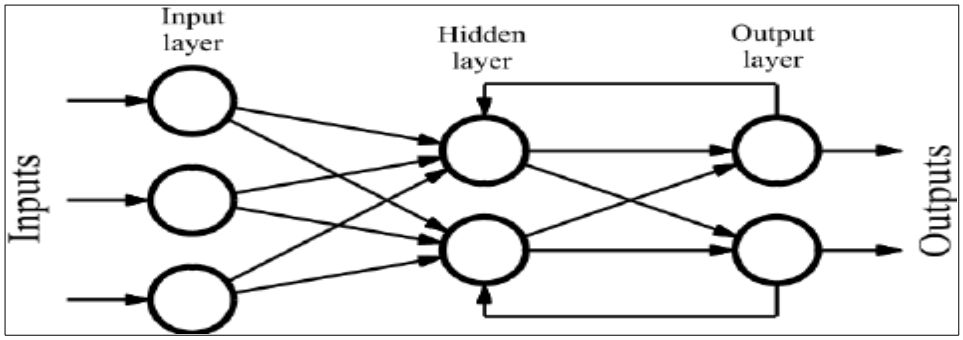


Figure II.6. *Graphe de réseau de neurones récurrent*

3.1.3.1. LSTM

Un LSTM (Long Short-Term Memory) est un type particulier de RNN qui élimine le défaut de disparition ou le problème de gradient explosif du RRN et cela rend LSTM adapté à apprendre de l'histoire à classer, traiter et prévoir les séries chronologiques quand il y a très temps long et inconnu entre les événements importants.

Les LSTM aident à préserver l'erreur qui peut être rétro-propagée à travers le temps et les couches, en maintenant une erreur plus constante, ils permettent aux filets récurrents de continuer à apprendre au fil du temps.**[Rana, epps et Jurdak, 2014]**

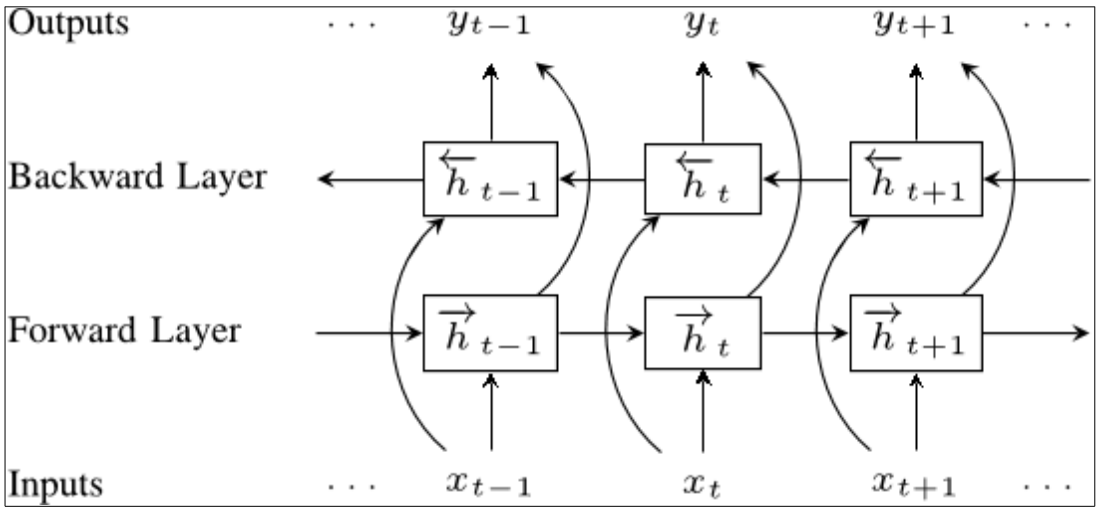


Figure II.7. *LSTM bidirectionnel*

3.1.3.2. GRU

Le Gated Recurrent Unit est un peu plus simplifié que la LSTM, Il combine les portes d'oubli et d'entrée dans une seule "porte de mise à jour" et possède une "porte de réinitialisation" supplémentaire, le modèle final est plus simple que les modèles LSTM standard et est devenu de plus en plus populaire.**[Rana, epps et Jurdak, 2014]**

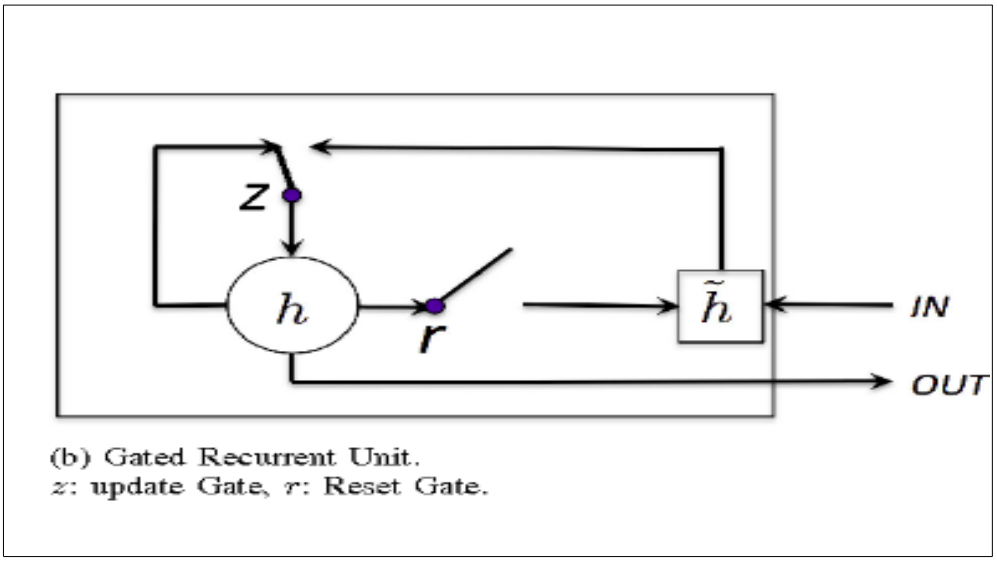


Figure II.8. *Gated Reccurent Unit*

Les GRU et LSTM ont des points communs et aussi des différences qui sont illustré dans le tableau suivant :

Tableau II.1. *Différence entre un GRU et un LSTM*

	LSTM	GRU
Exposition à la mémoire contrôlée	La quantité de mémoire vue par les autres unités du réseau est contrôlée par la sortie Gate.	Toute la mémoire est exposée au réseau.
Nouveau calcul de mémoire	Aucun contrôle séparé pour le montant de flux d'informations de l'étape de temps précédente.	Contrôle le flux d'informations de l'activation précédente.
Complexité vs performance	il est susceptible d'avoir une plus grande complexité Avec une porte supplémentaire	A moins de paramètres et peut donc s'entraîner comparativement plus vite ou besoin de moins de données pour généraliser

Les RNNs ne semblent pas être profonds comme DNN ou CNN en termes de nombre de couches mais ils peuvent être considérés comme une structure encore plus profonde s'ils sont déroulés dans le temps car ils ont été utilisés avec succès dans de nombreux domaines, y compris le traitement du langage naturel et la traduction de la langue.

Même si les RNNs ont été explorés moins que les DNN et les CNN, ils fournissent encore des méthodes d'analyse très puissantes pour l'information séquentielle.**[Seonwoo, Byunghan et Sungroh]**

3.2. Nouvelles Architectures

3.2.1. Réseaux de neurone temporel spatial-profond

La structure de base des DST-NN (Deep Spatio-Temporal Neuronal Network) est constituée de couches cachées multidimensionnelles, les caractéristiques spatiales se réfèrent aux entrées d'origine pour l'ensemble du DST-NN est sont utilisées de la même manière dans chaque couche, tant dis que les caractéristiques temporelles, elles sont progressivement modifiées de façon à progresser vers les couches supérieures mise à part la première couches. **[Seonwoo, Byunghan et Sungroh]**

3.2.2. Réseaux de neurones récurent multidimensionnel

Les MD-RNNs (Multi Dimensional Recurrent Neuronal Network) ont pour but d'appliquer les capacités des RNNs sur des données multidimensionnelles non séquentielles et ceci en les traitant comme des groupes de donnée séquentielles.
Les unités cachées sont connectées à une seule couche de sortie, et les résultats finaux sont calculés en tenant compte de tous les contextes possibles.**[Seonwoo, Byunghan et Sungroh]**

3.2.3. Les Auto-encodeur convolutionnels

Les CAE (Convolutionnel Auto-Encoded) sont conçus pour bénéficier des AE et CNNs afin d'apprendre de bonnes représentations hiérarchiques de données qui reflètent des informations spatiales et être bien régularisée par une formation non-supervisée.
Le codeur CNN et le décodeur constitué de couche de déconvolution et de désolidarisation sont intégrés pour former un CAE et sont formés de la même manière que dans AE. **[Seonwoo, Byunghan et Sungroh]**

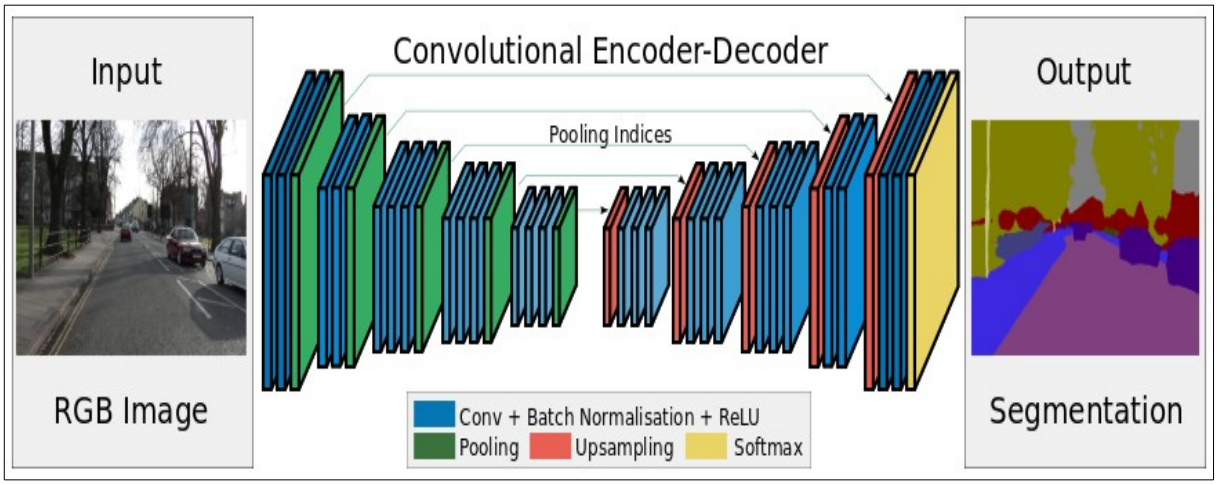


Figure II.9. *Autoencodeur covolutionnel appliqué sur une image RGB*

4. Conclusion

Les Réseaux de neurones sont un domaine de recherche très actif, il ne cesse d'augmenter pour améliorer les performances des résultats, ce chapitre a présenté les architectures des réseaux de neurones les plus connues et les plus utilisées ainsi que les nouvelles architectures qui semble avoir un avenir prometteur dans différents domaines d'application.

Chapitre III
Méthodes de Conception

1. Introduction

Ce chapitre va présenter d'une façon détaillée la conception de notre méthode proposée pour la réduction de la dimensionnalité et la classification des données de grande dimension, qui consiste à faire dans un premier temps une réduction de dimension de la base de données qui contient un grand nombre d'attributs et ceci en utilisant les méthodes le plus répandues dans le domaine de réduction de dimension qui sont bien évidemment : la sélection et l'extraction d'attributs via deux algorithmes importants : l'algorithme de gain d'information et un réseau de neurones profond de type auto-encodeur empilé, ensuite pourprédire les classes des nouvelles instances qui contiennent beaucoup moins d'attributs, cela dit que les attributs qui ont été réduits sont moins pertinents que ceux qui ont été gardés.
La classification sera faite avec deux méthodes, la première est une technique d'apprentissage automatique profond et la deuxième sera par l'utilisation d'un algorithme de classification très connu qui est le RBF-SVM.

2. Méthode proposée pour la réduction de dimensionnalité

La classification des instances rencontre plusieurs problèmes tels que le grand nombre des attributs qui caractérise les instances ou les exemples.Afin d'améliorer la précision de la classification et réduire les temps de calcul, divers méthodes de réduction de dimension ont été conçues dans ce but. Dans ce travail nous proposons une méthode pour la réduction de la dimensionnalité des données pour la classification composée de deux étapes : étape de sélection d'attributs et une autre étape d'extraction d'attributs.
Voici les étapes de la méthode proposée pour la réduction de la dimensionnalité des bases de données volumineuses qui est schématisée dans la figure III.1 :

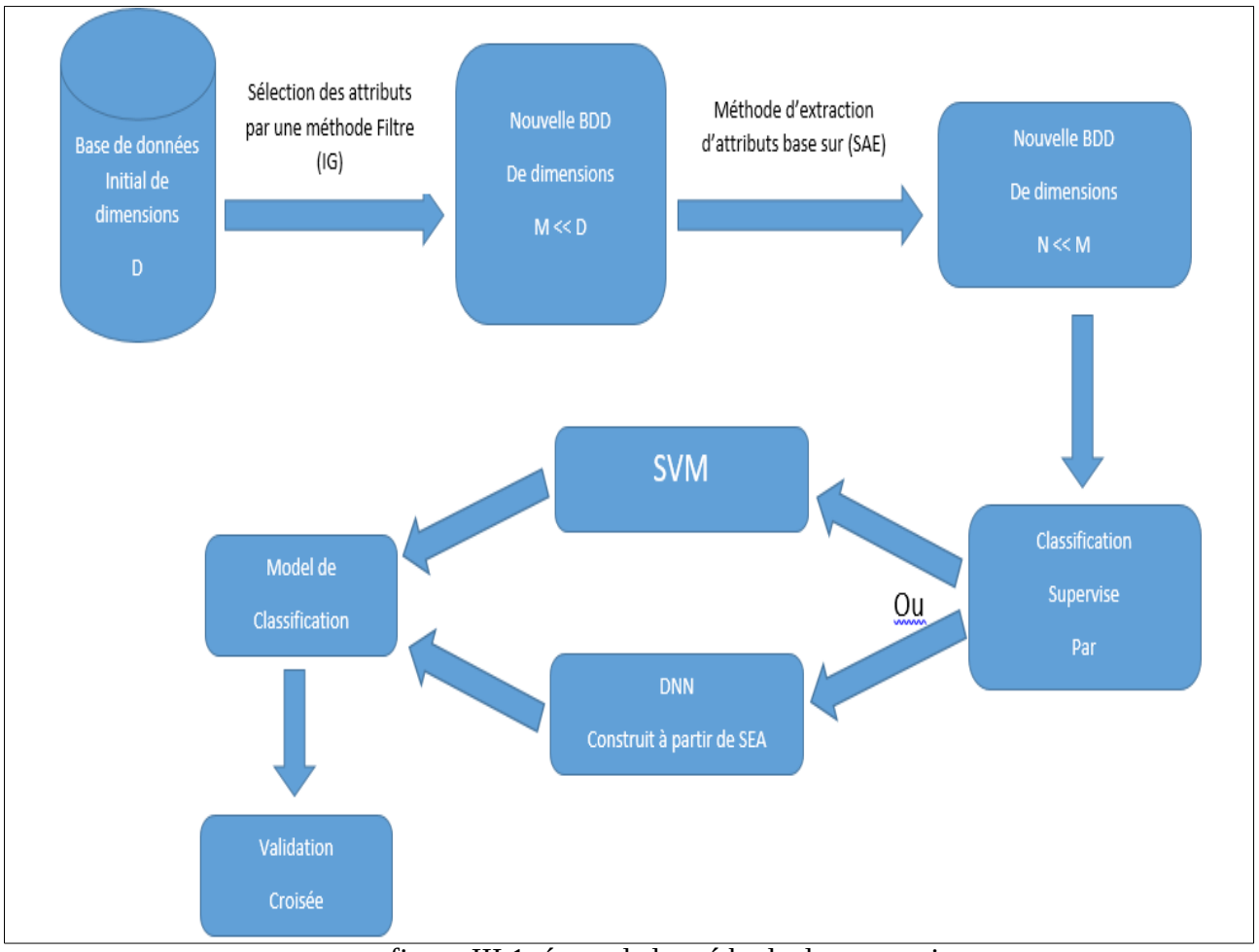


figure III.1. étape de la méthode de conception

2.1. La sélection d'attributs avec une méthode de filtrage

la sélection d'attributs est de loin la technique la plus répandue dans le domaine de réduction de dimensionnalité,elle est fréquemment utilisée comme une étape de prétraitement pour l'apprentissage automatique en sélectionnant un sous-ensemble de caractéristiques d'origine afin que l'espace des fonctionnalités soit réduit de manière optimale en fonction d'un certain critère d'évaluation.

La sélection d'attributs est un domaine fertile de la recherche et du développement depuis les années 1970 et il s'est avérée efficace pour :

- Eliminer les caractéristiques inutiles et redondantes
- Accroître l'efficacité des tâches d'apprentissage
- Améliorer les performances d'apprentissage telles que l'exactitude prédictive
- Améliorer la compréhensibilité des résultats acquis.

La sélection des fonctionnalités devient très nécessaire pour les tâches d'apprentissage automatique face aux données de grande dimension de nos jours afin d'éviter les problèmes qui peuvent être causés par les données volumineuses. [Lei et Liu,2003]

La mesure de pertinence associée aux méthodes de sélection d'attributs sont souvent basées sur des heuristiques calculant l'importance individuelle de chaque attribut dans le modèle obtenu, ces heuristiques sont de différentes natures : statistique, probabiliste, information mutuelle ou celle qui mesure l'indépendance ou la vraisemblance entre les attributs.

2.1.1. Gain d'information

le gain d'information est souvent utilisé comme critère de qualité dans le domaine de l'apprentissage automatique, et ceci en mesurant le nombre de bit d'information obtenus pour la prédiction tout en connaissant la présence ou l'absence d'un terme dans un document.[Yiming et pedersan,2003]

L'entropie ou le gain d'information est le taux oumoyen auquel l'information est produite par une source de données stochastiques.

On calcule le gain d'informations pour chaque attribut, les attributs qui apportent plus d'informations auront une valeur de gain d'information plus élevée et peuvent être sélectionnés, tandis que ceux qui n'en ajoutent pas beaucoup auront un score inférieur et pourront être supprimés.

2.1.2. Information mutuelle

L'information mutuelle est un critère généralement utilisé dans la modélisation statistique des langages d'association de mots et d'application connexe, elle possède un taux de complexité de temps similaire à celui du gain d'information.

Dans le cas de caractérisation des termes la faiblesse de l'information mutuelle est que le score du terme est fortement influencé par les probabilités marginales de terme.[Yiming et pedersan,2003]

2.1.3. Le X^2 statistique

LeX^2 statistique mesure le manque d'indépendance entre un termeet sa catégorie ou un attribut et sa classe et peut être comparée à la X^2 distribution avec un degré de liberté pour juger les extrémités.

Le X^2 statistique est aussi connu sous le terme de "le calcul des scores CHI", il possède une complexité quadratique similaire à celle du gain d'information et l'information mutuelle, la différence entre cette dernière et le X^2 statistique est que ce dernier est une valeur normalisée, par conséquent les valeurs seront comparer dans la même catégorie.

La normalisation est rompue si une cellule de tableaux de fréquence du terme est peu peuplée c'est à dire si sa valeur est vraiment faible par rapport à d'autre ce qui rend le X^2 statistique non fiable.

Il faut mentionner que la méthode de jugement de qualité de la performance de sélection d'attribut qui est la plus pratique pour nos bases est bien celle du "*gain d'information*", c'est pour ça qu'on l'avais utilisée pour réduire le nombre d'attributs de nos base de données d'une façon remarquable.

Remarque : avant d'entamer la deuxième phase de réduction de dimension, une étape très nécessaire a était effectuer sur toute les bases de données génétique qui est : "*la normalisation des valeurs des attributs"*, c'est à dire toutes les valeurs ont été mises dans un intervalle de 0 et 1 afin de faciliter la prochaine réduction et améliorer la prédiction des classes.

2.2. Réduction de dimension via l'extraction d'attributs

la méthode de réduction de dimension utilisé n'est d'autre que celle des réseaux de neurones profonds et plus précisément les autoencodeur-empilé (*en anglais : Stacked-Autoencoder*) .

Les auoencodeurs sont des réseauxde neurones composés de trois types de couches :

- La couche des entrées
- les couches cachées
- la couche des sorties

Les couches cachées relient la couche d'entrée avec celle de la sortie et elles sont partagées pour former l'encodeur avec la couche d'entrée, et le décodeur avec la couche de sortie.

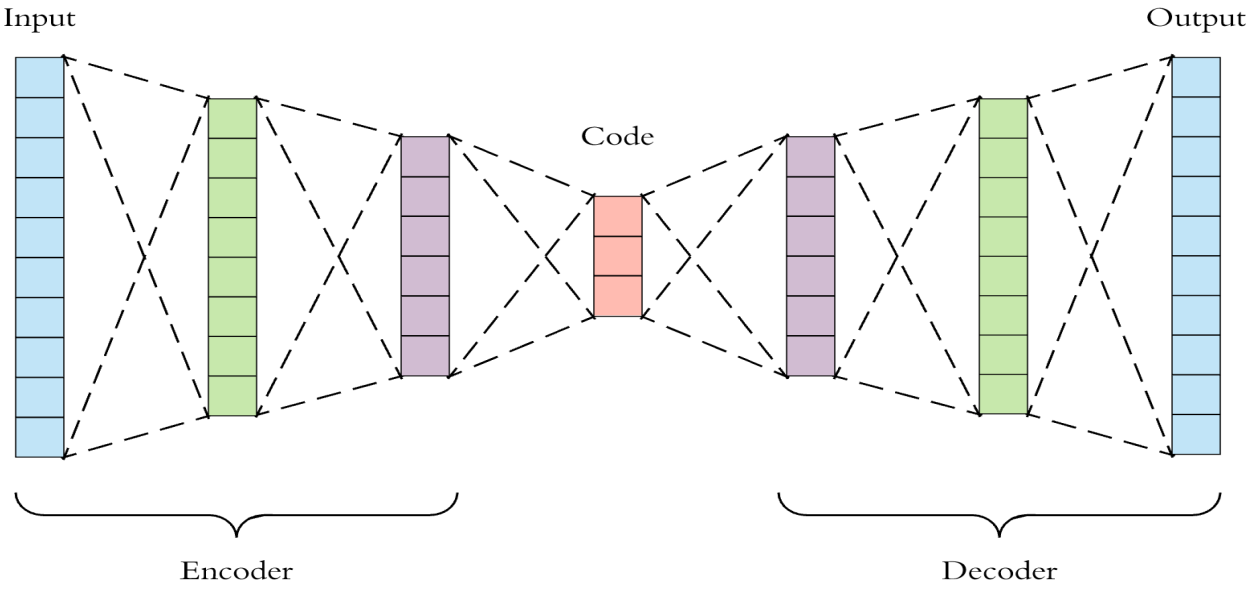


Figure III.2. Autoencodeur avec plusieurs couches cachées

Afin d'extraire les attributs les plus fort ou bien les plus importants pour une bonne classification, l'autoencodeur passe par trois phases principales :

2.2.1. L'encodage

Cette étape se passe comme noté précédemment entre la couche d'entrée et les couches cachées, les nœuds de la première ne sont d'autres que les attributs initiales de nos bases de données qui seront combinés aléatoirement pour avoir de nouveaux attributs avec peut-être plus de précision. Le nombre de neurones cachés a été initialisé au début de l'apprentissage pour chaqu'une des couches cachées.Dans notre architecture proposée ont adeux puis trois couches cachées, pour l'étape d'encodage des attributs, les poids seront mis automatiquement a fur et à mesure de l'encodage, couche par couche et non pas dans le début d'un seul coup.

Il faut préciser que la dernière couche cachée de cette étape est la première pour l'étape suivante c'est à dire le décodage.

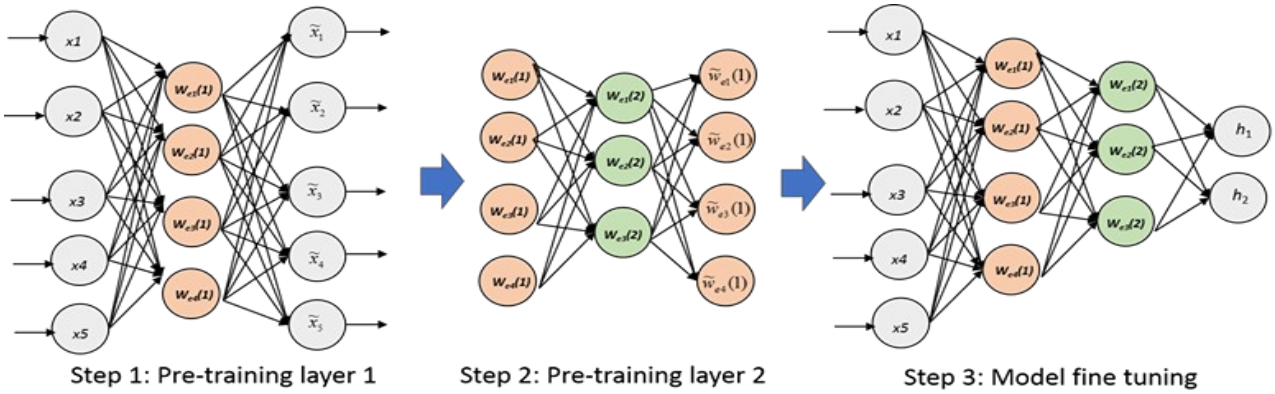


Figure III.3. Illustration d'encodage avec trois couches cachées

L'encodeur calcule à partir de X, le vecteur h de taille m qui est le nombre de neurones cachéstel que :

$$\mathbf{h = !(W^{(1)}x + b^{(1)})}$$

-$W^{(1)}$ est une matrice de taille m*n

-$b^{(1)}$ un vecteur de biais de taille m

- !(.) est une fonction d'activation de type sigmoïde hyperbolique définie comme suite :

$$\mathbf{f(y)=1/(1-exp^{(-x)})}$$

2.2.2. Le décodage

Le décodage se fait entre les couches cachées et la couche des sortie qui possède le même nombre de nœuds de la couche d'entrée, les couches cachées dans cette étapes sont symétriques aux couches cachées de l'étape précédente avec une en commun (celle du milieu).

L'auto-encodeur empilé tante d'avoir à partir de la couche caché du milieu la représentation la plus proche des attributs initiales (ceux de la couche d'entrée).

Dans la phase de modification des poids de la partie décode, une descente de gradient est effectuée, ce qui oblige une fonction dérivable comme sigmoïde a êtreexécutée et c'est ce que nous avons effectués dans notre travail, les poids des nœuds de la partie décode sont les même de la partie encode on a fait une sorte de transposé des poids en utilisant la fonction "sigmoïde" dans les deux partie (encode et décode), et ceci afin de réduire le taux d'erreur qui sera calculé dans la troisième et dernière étapes pour valider l'extraction.

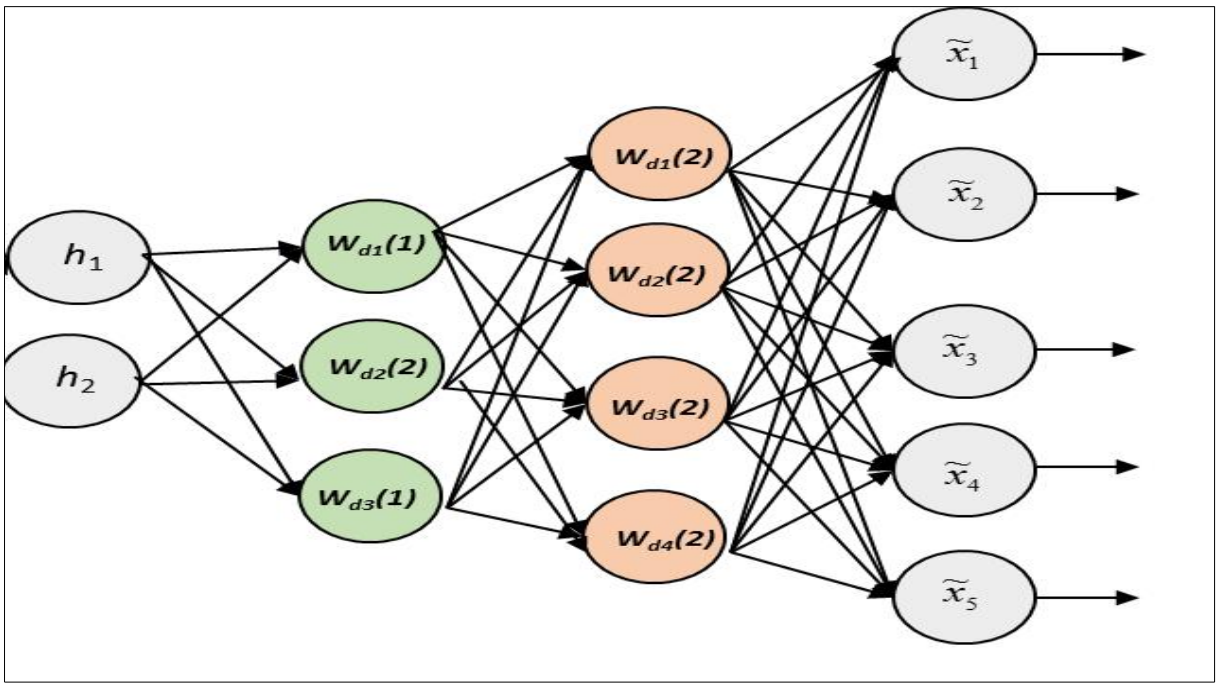


Figure III.4. Illustration de décodage du réseau précédent

Le décodeur cherche à reconstruire le vecteur x à partir de la couche cachée h**,** le résultat de cette reconstruction est le vecteur $\overline{X}$ tel que :

$$\overline{X} = (W^{(2)}h + b^{(2)})$$ où :

-$\overline{X}$ est un vecteur de taille m
-$W^{(2)}$ est une matrice de poids de taille n*m
-$b^{(2)}$ est un vecteur de biais de taille n.

2.2.3. La reconstruction

Une erreur quadratique est calculée entre les nœuds de la couche d'entrée et ceux de la couche de sortie à la fin du décodage et ceci afin de trouver l'erreur minimale entre elles, pour bien choisir l'ensemble de nœuds le plus représentatif réduit de l'ensemble initial.

Cette ensemble n'est d'autre que celui de la couche cachée qui se trouve au milieu du grand réseau de neurone c'est à dire la fameuse couche cachée commune entre l'étape d'encodage et celle du décodage.

Durant l'apprentissage, l'auto-encodeur tente de réduire une erreur de reconstruction entre X et $\overline{X}$, Il utilise pour ceci l'erreur quadratique moyenne de manière à minimiser l'erreur de reconstruction totale avec l'ensemble de paramètres ✓ = {$W^{(1)}$, $b^{(1)}$, $W^{(2)}$, $b^{(2)}$}.

Plusieurs essais ont été effectuer sur les multiples base de tests afin de trouver la meilleure représentation réduite, les essais ont diversifié entre le changement de nombre de nœuds dans les

couches cachées ou bien en changeant carrément le nombre de couche cachée utilisée car on peut bien évidemment les changer selon notre besoin pour améliorer les résultats.

3. Classification des données

Après avoir réduit d'une manière remarquable le nombre d'attributs de la base de données originale, il est temps maintenant de prédire les classes de nouvelles instances, pour cela deux méthodes ont été utilisés et qui sont :

- La prédiction de la classe via un réseau de neurones profond (DNN).
- La prédiction de la classe via l'algorithme de classification Support Vector Machine (SVM).

3.1. Méthode de réseau de neurones profond (DNN)

Après avoir choisi la meilleure représentation réduite des données de la base initiale, l'algorithme de classification ne prendra que les couches de la partie encode, il ajoutera par la suite une seule couche contenant deux nœuds seulement, ceci est spécifique à nos bases qui sont des bases binaires, c'est à dire elles ne possèdent que deux classes, pour les autres types de bases le nombre de nœuds dans la dernière couche changera selon le nombre de classes de chaque base.
Le type de fonction d'activation adéquat a notre problème de classification est bien le type "non linéaire" et on a choisi la fonction "softmax" qui quand on l'utilise la somme des sorties doit être égale à 1 ce qui signifie que la procédure donnera en fin d'exécution 0 pour un des nœuds et forcément 1 pour le deuxième, le nœuds prenant 1 est donc la classe prédite pour cette série d'attributs. La figure III.4 explique la construction du réseau de neurones profond utilisé.
Une évaluation de performance de ce système sera effectuée afin de juger le taux de précision des classes prédites par rapport aux classes originales.

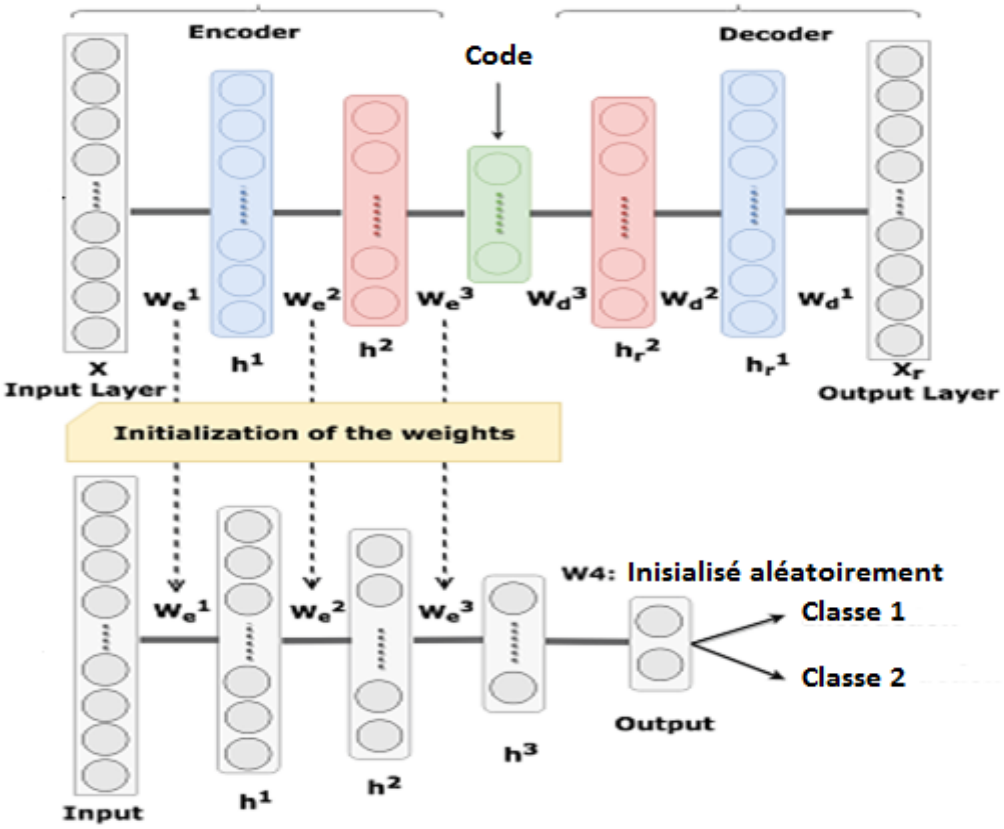


Figure III.5. Construction de DNN pour la classification

3.2. Les Séparateurs à Vaste Marge

Les Séparateurs à Vaste Marge (*en anglais : Support Vector Machines*) sont des techniques largement répandues en apprentissage statistique conçues dans un premier temps pour la classification binaire, et afin d'éviter ou de limiter le phénomène du sur-apprentissage (*en anglais : overfitting*), son principe consiste à intégrer une estimation de complexité durant l'apprentissage.
Les séparateurs à vaste marge reposent sur deux idées clés :

- la notion de marge maximale.
- la notion de fonction noyau.

Le cas qui nous intéresse est celui de la fonction noyaux car il s'agit d'apprentissage automatique des données. Cependant, dans ce travail on a utilisé les séparateurs à vaste marge avec le noyau RBF (*en anglais : Kernel Radial Basis Function*) ce dernier est une fonction à valeur réelle dont la valeur ne dépend que de la distance par rapport à l'origine, la norme est généralement la distance euclidienne, bien que d'autres fonction de distancesoient également possibles.

3.2.1. Notions de base : Hyperplan, marge et support vecteur

Pour deux classes d'exemples donnés, le but de SVM est de trouver un classificateur qui va séparer les données et maximiser la distance entre ces deux classes. Avec SVM, ce classificateur est un classificateur linéaire appelé hyperplan.
Dans le schéma qui suit, on détermine un hyperplan qui sépare les deux ensembles de points.Les points les plus proches, qui seuls sont utilisés pour la détermination de l'hyperplan, sont appelés vecteurs de support.

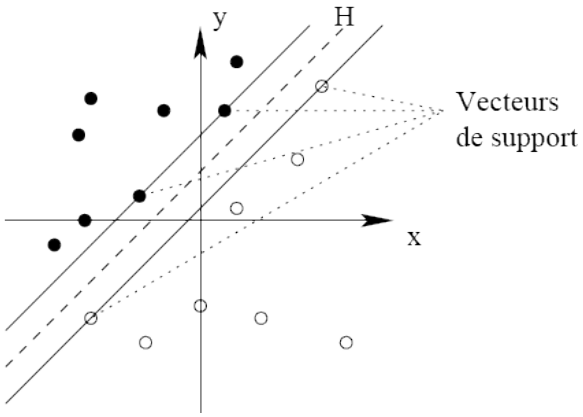


Figure III.6. un hyperplan

Il est évident qu'il existe une multitude d'hyperplan valide mais la propriété remarquable des SVM est que cet hyperplan doit être optimal. Nous allons donc en plus chercher parmi les hyperplans valides, celui qui passe « au milieu » des points des deux classes d'exemples. Intuitivement, cela revient à chercher l'hyperplan le « plus sûr ». En effet, supposons qu'un exemple n'ait pas été décrit parfaitement, une petite variation ne modifiera pas sa classification si sa distance à l'hyperplan est grande. Formellement, cela revient à chercher un hyperplan dont la distance minimale aux exemples d'apprentissage est maximale. On appelle cette distance « marge » entre l'hyperplan et les exemples. L'hyperplan séparateur optimal est celui qui maximise la marge. Comme on cherche à maximiser cette marge, on parlera de séparateurs à vaste marge.

3.2.2. Linéarité et non-linéarité

Parmi les modèles des SVM, on constate les cas linéairementséparable et les cas non linéairement séparable. Les premiers sont les plus simples de SVM car ils permettent de trouver facilement le classificateur linéaire. Dans la plupart des problèmes réels il n'y a pas de séparation

linéaire possible entre les données, le classificateur de marge maximale ne peut pas être utilisé car il fonctionne seulement si les classes de données d'apprentissage sont linéairement séparables.

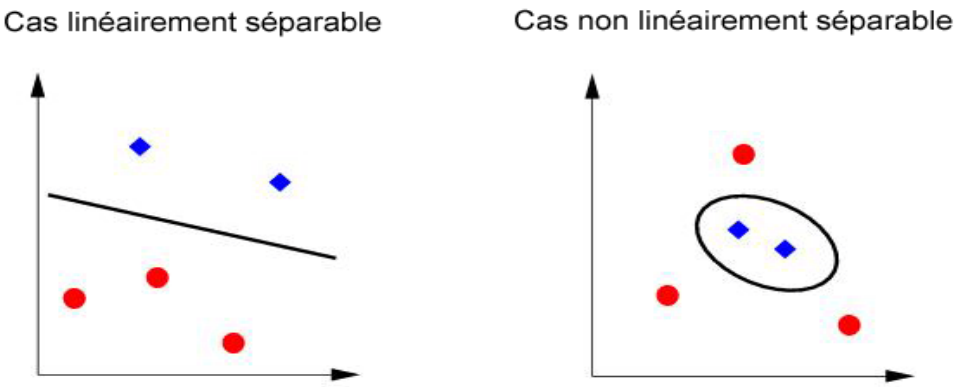


Figure III.7. Linéarité et non-linéarité

Pour surmonter les inconvénients des cas non linéairement séparable, l'idée des SVM est de changer l'espace des données. La transformation non linéaire des données peut permettre une séparation linéaire des exemples dans un nouvel espace. On va donc avoir un changement de dimension. Cette nouvelle dimension est appelé « espace de re-description ». En effet, intuitivement, plus la dimension de l'espace de re-description est grande, plus la probabilité de pouvoir trouver un hyperplan séparateur entre les exemples est élevée.

On a donc une transformation d'un problème de séparation non linéaire dans l'espace de représentation en un problème de séparation linéaire dans un espace de re-description de plus grande dimension. Cette transformation non linéaire est réalisée via une fonction noyau. En pratique, quelques familles de fonctions noyau paramétrables sont connues et il revient à l'utilisateur de SVM d'effectuer des tests pour déterminer celle qui convient le mieux pour son application. On peut citer les exemples de noyaux suivants : polynomiale, gaussien, sigmoïde et Laplacien.

Le cas qui nous intéresse est celui de la fonction noyaux car il s'agit d'apprentissage automatique des données. Cependant, dans ce travail on a utilisé les séparateurs à vaste marge avec le noyau RBF (*en anglais : Kernel Radial Basis Function*) ce dernier est une fonction à valeur réelle dont la valeur ne dépend que de la distance par rapport à l'origine, la norme est généralement la distance euclidienne, bien que d'autres fonction de distancesoient également possibles.

3.3. Procédure de validation

Afin de tester la performance du système la méthode de validation croisée a été employée, c'est une méthode statistique utilisée pour estimer les compétences des modèles d'apprentissage automatique pour comparer et sélectionner un modèle pour un problème de modélisation prédictif donné, on l'avait choisi car elle a beaucoup d'avantage on cite :

- Facile à comprendre
- Facile à mettre en œuvre

- Donne des estimations de compétences généralement moins biaisées que les autres méthodes.

La procédure a un seul paramètre appelé k qui fait référence au nombre de groupes dans lesquels un échantillon de données doit être divisé. En tant que telle, la procédure est souvent appelée validation croisée par k-fold, elle consiste à utiliser un échantillon limité afin d'estimer comment le modèle est censé fonctionner en général, lorsqu'il est utilisé pour faire des prédictions sur des données non utilisées lors de la formation du modèle.

La procédure générale est la suivante:

- Mélangez le jeu de données de manière aléatoire.
- Diviser le jeu de données en k groupes
- Pour chaque groupe unique:
  - Prenez le groupe comme un ensemble de données de test ou d'attente
  - Prendre les groupes restants comme un ensemble de données d'entraînement
  - Ajustez un modèle sur l'ensemble d'entraînement et évaluez-le sur l'ensemble de test
  - Conserver le score d'évaluation et rejeter le modèle
- Résumer les compétences du modèle en utilisant l'échantillon de scores d'évaluation du modèle

Noter bien que Si l'on choisit une valeur pour k qui ne divise pas de manière égale l'échantillon de données, un groupe contiendra un reste des exemples. Il est préférable de diviser l'échantillon de données en k groupes avec le même nombre d'échantillons, de sorte que l'échantillon des scores de compétence du modèle soit équivalent.

Voici une illustration de 10-fold :

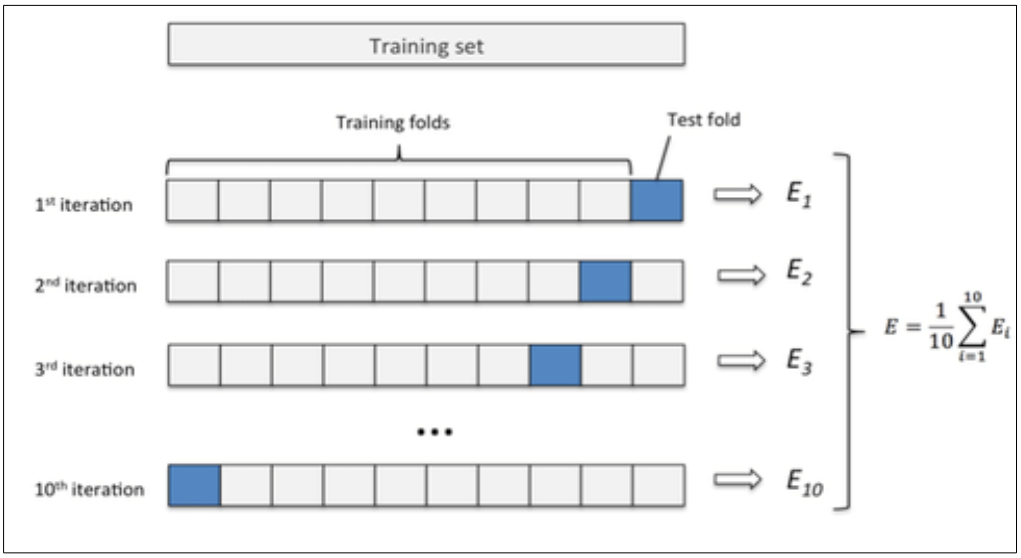


Figure III.8. Schéma de validation croisée avec 10-fold

4. Conclusion

Ce chapitre a détaillé toutes les étapes de la méthodeproposée dans ce travail afin de réduire la dimension des différentes bases de tests, et pour pouvoir mieux classifier ou prédire les classes de nouvelles instances.

Chapitre IV
Implémentation et Résulats

1. introduction

Dans ce chapitre nous allons voir les différents outils utilisé pour la réalisation de notre travail, ainsi que le langage de programmation et les bibliothèques correspondantes et enfin, les résultats obtenue on appliquant les méthodes cité dans le chapitre précédent.

2. Description des bases de données utilisées

Nous donnons ci-dessous les caractéristiques de jeux de données utilisé dans la phase expérimentale, ces bases de données concernent des problèmes de reconnaissance de cancers et ou de prévision de diagnostic en oncologie et qui sont utilisés dans de nombreux travaux concernant la classification des données de puces à ADN.

1- Leucémie (Lukemia)

Le jeu de données appelé dans la suite “données de la leucémie” est constitué de 72 échantillons représentant deux types de leucémie aigüe. 47 tissus sont du type Leucémie lymphoblastique aiguë (ALL) et 25 sont du type Leucémie myéloïde aiguë(AML). Pour chaque échantillon, on a relevé les niveaux d’expression de 7129 gènes.

2- Cancer du colon

Ce jeu de données qui concerne le cancer du colon, est constitué de 62 échantillons dont 40 sont des tissus tumoraux et 22 sont des tissus sains ou normaux. Les expériences ont été menées avec des puces relevant les valeurs d’expression pour plus de 6500 gènes humains mais seuls les 2000 gènes ayant les plus fortes intensités minimales ont été retenus. La matrice des niveaux d’expression comporte donc 2000 colonnes et 62 lignes.

3- Tumeurs du cerveau (CNS)

Cette base de données relative aux tumeurs embryonnaires du système nerveux central ou CNS, elle contient les échantillons obtenus de 60 patients qui ont été traités pour des médulloblastomes. Parmi ces 60 patients, 39 sont survivants et pour les 21 autres le traitement est un échec. Chaque échantillon est décrit par 7129 gènes.

4- Cancer de la prostate

Dans ce jeu de données, le niveau d’expression de 12600 gènes est mesuré sur 136 tissus. L’objectif initial est de distinguer les tissus normaux (77) des tissus cancéreux (59).

Le tableau suivant résume les caractéristiques des 4 bases de données utilisées dans ce travail.

Toutes ces bases de test sont issues des données de puce à ADN et qui sont caractérisées par :

- Une grande dimensionnalité due au nombre élevé de descripteurs (gènes)
- Un nombre limité d'échantillons

Tableau IV.1 . *Caractéristiques des bases de test*

Nom de la base	Nombre d'attributs	Nombre d'instance	distribution
leucémie	7129	72	25 aigue 47 lucémie ymphoblastique aiguë
Cancer du colon	2000	62	40 tumoraux 22 sains
Tumeur de cerveau	7129	60	39 survivant 21 morts
Cancer de la prostate	12600	136	77 normaux 59 cancéreux

3. Selection d'attributs

La selection ou le filtrage des attributs a été effectué avec le logiciel *weka qui* est une suite de logiciels d'apprentissage automatique écrite en Java et développée à l'université de Waikato en Nouvelle-Zélande.

Weka est un logiciel libre disponible sous la Licence publique générale GNU.

Voici la fiche du logiciel :

- **Type :** Structure logicielle d'apprentissage automatique
- **version:**3.8.1 (23 janvier 2017)
- **Environnements :** plateforme Java

3.1. Etapes de filtrage de donnée avec weka

Dans cette partie on va donner les différentes étapes qui ont été effectuées afin de filtrer les attributs non pertinents ou peu significatifs de nos différentes bases de test, sachant que toute les bases utilisé sont des classes binaire, elles varient entre le nombre d'instance ainsi que le nombre d'attributs, chaque base traite un genre de cancer précis et les classes désigne existence ou l'absence d'un cancer.

Chaque étape sera sous-forme d'une image capturée directement lors de l'exécution de cette phase.

1. lancer le logiciel :

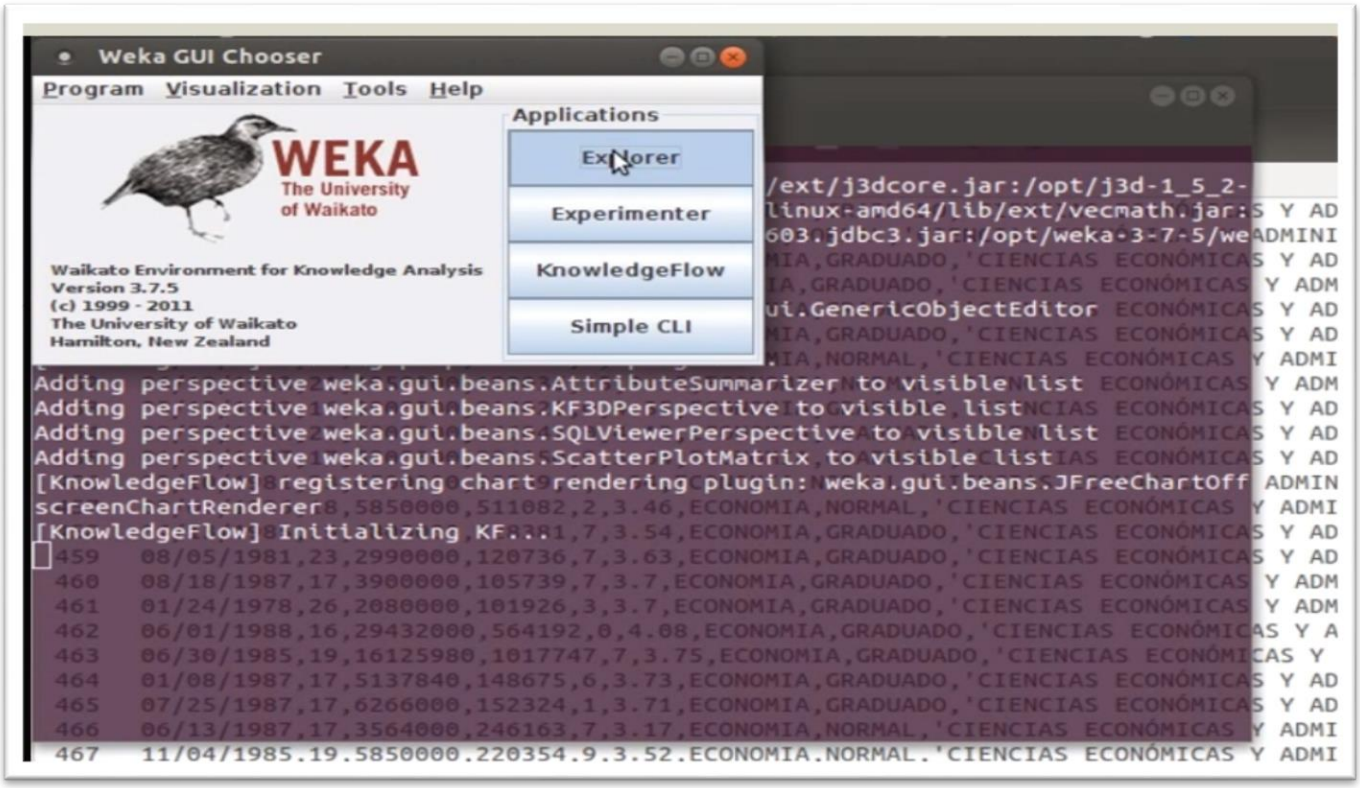


Figure IV.1. *Lancer Weka*

2. Ouvrir le fichier de données qu'on veut filtrer :

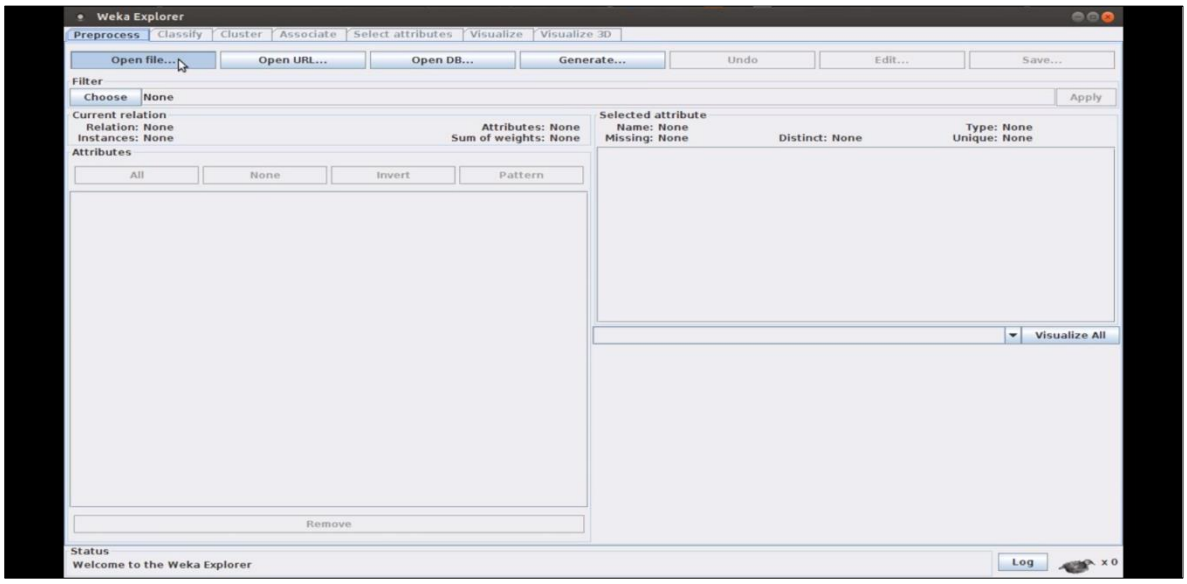


Figure IV.2. *Ouverture d'un fichier*

3. Choisir l'algorithme de sélection :

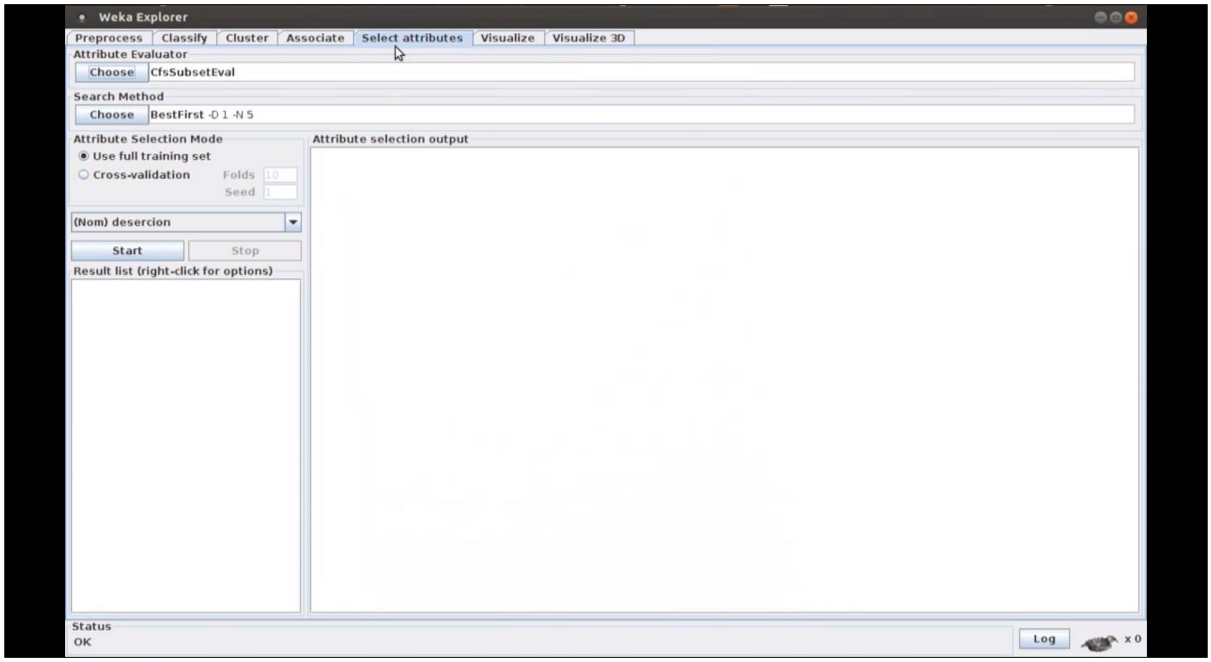


Figure IV.3. *Choix de l'algorithme voulu*

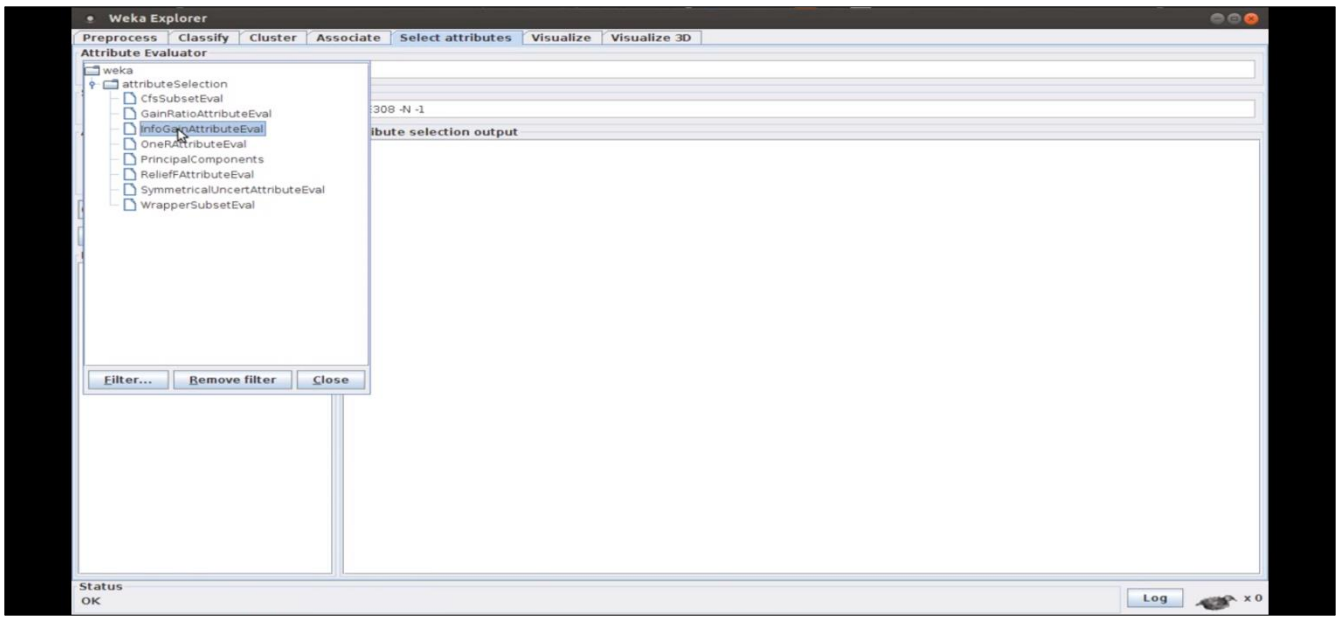


Figure IV.4. *Choix de l'algorithme*

4. Sélectionner la méthode d'évaluation

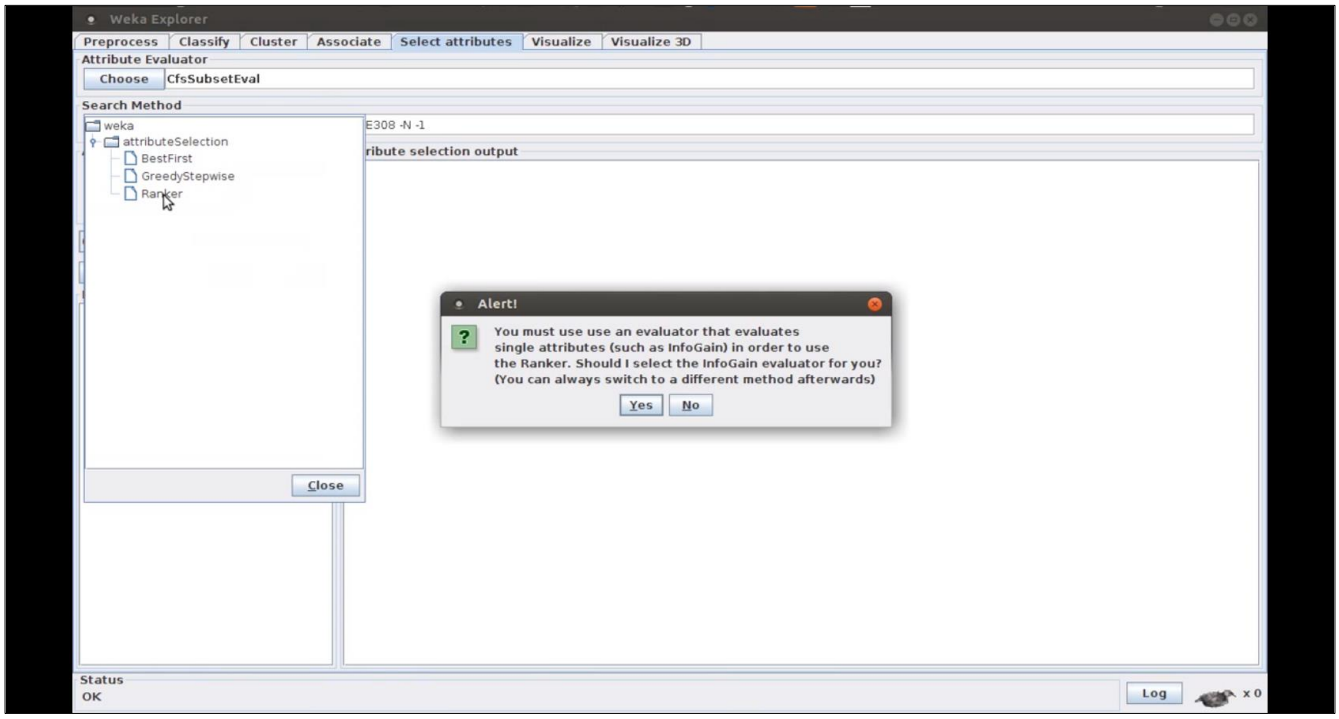


Figure IV.5. *Méthode d'évaluation*

3.2. Résultat de filtrage

Le tableau suivant donne les résultats obtenus en appliquant les étapes précédentes sur quatre bases de données à classe binaire :

Tableau IV.2. *Résultats de la sélection sur les bases de test*

nom fichier	prostate	Leukemia	colon	cns
nombre d'attributs originaux	12600	7129	2000	7129
nombre d'attributs après filtrage	2658	940	263	74
Nombre d'instances	136	72	62	60

4. Extraction des attributs et classification

Concernant la deuxième phase de réduction de dimension, on a choisi de faire l'extraction des attributs via un Autoencodeur empilé. Deux types d'architectures d'autoencodeur empilé ont été créés. La première architecture est composée de 5 couches et la deuxième architecture est composée de 7 couches. Pour le développement de ces architectures on a utilisé le langage python qui est un langage de programmation orienté objet, multi-paradigme et multiplateformes, il favorise la programmation impérative structurée, fonctionnelle et orientée objet, la version python utilisée est 2.7.0

Plusieurs bibliothèques ont été utilisées dans ce travail, mais la plus importante est bien *Tensorflow* qui est une bibliothèque de logiciels open source pour le calcul numérique de haute performance.

Son architecture flexible permet un déploiement facile du calcul sur diverses plates-formes (processeurs, processeurs graphiques, TPU), et des ordinateurs de bureau aux clusters de serveurs, aux périphériques mobiles et périphériques.

La bibliothèque *Tensorflow*[31] a été développé à l'origine par des chercheurs et des ingénieurs de l'équipe Google Brain au sein de l'organisation AI de Google, elle prend en charge l'apprentissage automatique et l'apprentissage en profondeur et le cœur de calcul numérique flexible est utilisé dans de nombreux autres domaines scientifiques.

Version utilise : 1.4.0 CPU

Commande d'installation : sudo pip install tensorflow 1.4.0

4.1. Résultat d'extraction

Cette section va donner les résultats obtenue après l'exécution d'un code python, qui consiste à réduire le nombre d'attributs des bases filtrés dans la phase précédente puis dans une deuxième étape tente de prédire les classes des instances avec les nouveaux attributs combinés.

Les résultats suivant ont été récoltés avec un taux d'apprentissage de 0,01 et le max_epoche est fixé à 1000 epoche.

Taux de précision = (nombre d'instances bien classer / nombre total des instances)*100

[3] Définition de tensorflow consulté le 12 juillet 2018 récupérée de [https://www.tensorflow.org/]

Tableau IV.3. *Extraction et classification pour la base de cancer de prostate*

prostate	5 couches	7 couches
Nombre d'attribut avant extraction	2658	2658
Nombre d'attribut aprés extraction	800	400
Nombre de noeuds dans la première couche	1500	1500
Nombre de noeuds dans la deuxième couche	800	800
Nombre de noeuds dans la troisième couche	/	400
Instance bien classer	125	120
Instance mal classser	11	16
Taux de précision	91.91%	88.23%

Tableau IV.4. *Extraction et classification pour la base de cancer de colon*

colon	5 couches	7 couches
Nombre d'attribut avant extraction	263	263
Nombre d'attribut aprés extraction	90	50
Nombre de noeuds dans la première couche	150	150
Nombre de noeuds dans la deuxième couche	90	90
Nombre de noeuds dans la troisième couche	/	50
Instance bien classer	59	57
Instance mal classser	3	5
Taux de précision	95.16%	91.93%

Tableau IV.5. *Extraction et classification pour la base de leukemia*

leukemia	5 couches	7 couches
Nombre d'attribut avant extraction	940	940
Nombre d'attribut aprés extraction	200	100
Nombre de noeuds dans la première couche	500	500
Nombre de noeuds dans la deuxième couche	200	200
Nombre de noeuds dans la troisième couche	/	100
Instance bien classer	66	65
Instance mal classser	06	7
Taux de précision	91.66%	90.27%

Tableau IV.6. *Extraction et classification pour la base de CNS*

CNS	5 couches	7 couches
Nombre d'attribut avant extraction	74	74
Nombre d'attribut aprés extraction	15	8
Nombre de noeuds dans la première couche	35	35
Nombre de noeuds dans la deuxième couche	15	15
Nombre de noeuds dans la troisième couche	/	8
Instance bien classer	55	51
Instance mal classser	5	9
Taux de précision	91.67%	85.0%

4.2. Analyse des résultats d'extraction

Comme on peut le remarquer le système de prédiction est très bon car il nous a permis d'avoir au moins 85% comme taux de précision, c'est à dire que malgré le nombre d'attributs vraiment réduit par rapport au nombre d'attributs original il a quand même put faire la prédiction de classe et avoir des résultats très bons par rapport au classes originales

5. prédiction des classes par SVM et tests de performance avec la validation croisée

Scikit-learn est une bibliothèque libre de python dédiée à l'apprentissage automatique. Elle est développée par de nombreux contributeurs notamment dans le monde académique par des instituts français d'enseignement supérieur et de recherche comme INRIAet Télécom ParisTech.
Elle comprend notamment des fonctions pour estimer des forêts aléatoires, des régressions logistiques, des algorithmes de classification, et les machines à vecteurs de support. Elle est conçue pour s'harmoniser avec d'autres bibliothèques libre de python, notamment NumPyet SciPy.

- La version de Scikit-learn qu'on a utilisé est : 0.19.1
- La commande d'installation : sudo pip install Scikit-learn

5.1. Resultats de prédiction par SVM

Dans cette partie nous allons voir les résultats de prédiction obtenue en appliquant l'algorithme SVM et en jugant la performance du système de prédiction par la méthode de validation croisée qui a été appliquer 20 fois pour chaque base.
La précision maximale et minimale indiqué dans les tableaux ci-dessous sont données avec l'itération le rang de la validation croisée correspondante, on peut avoir une seule précision dans plusieurs itérations.
Les paramètre C et Gamma ont été calculé par un code python spécialisé pour donner la meilleure approximation correspondante à chaque base, C'est le paramètre de la fonction de coût qui contrôle l'influence de chaque vecteur de support individuel, ce processus implique une erreur de négociation pour la stabilité ; tant dis que Gamma est le paramètre libre de la fonction de base radiale gaussienne.

- Ecart type est une mesure de la dispersion des valeurs autour de leur moyenne arithmétique. Cette dernière se calcule avec la formule suivante :

$$\overline{x} = \frac{1}{n}\sum_{i} x_i$$

Dans notre cas I = 20 et les X_i sont les précisions obtenues dans chaque itération tant dis que l'écart type se calcule de la façon suivante :

$$\sqrt{\frac{1}{n}\sum_i (x_i - \bar{x})^2}$$

Tableau IV.7. *Résultats de prédiction par SVM pour la base de cancer de prostate*

prostate	Réduction par 5 couches	Réduction par 7 couches
Nombre d'instance	136	136
Nombre d'attribut	800	400
C	256	16384
gamma	1.479925	0.26358
Précision maximale	65.44% validation croisée 7	70.59% validation croisée 7,9 et 12
Précision minimale	33.82 validation croisée 2	32.35% validation croisée 3
Moyenne de 20 exécution	61.46%	66.18%
Ecart_Type	7.92%	9.08%

Tableau IV.8. *résulats de prédiction par SVM pour la base de leukemia*

leukemia	Réduction par 5 couches	Réduction par 7 couches
Nombre d'instance	72	72
Nombre d'attribut	200	100
C	1024	32768
gamma	0.25	0.00061
Précision maximale	98.61% validation croisée 2,3 et11	94.44% validation croisée 6 et 9
Précision minimale	97.22% validation croisée 4,5,6,7,8 et 9	90.28% validation croisée 2
Moyenne de 20 exécution	97.44%	92.99%
Ecart_Type	0.51%	0.84%

5.2. Analyses des résultats de SVM

Les résultats de la première base ne sont pas mieux que ceux des résultats obtenus avec la méthode d'extraction, mais on peut quand même les juger de bons résultats car le pourcentage de 65 et 70 % n'est pas faible.

Tant que pour les autres bases, les résultats de SVM sont encore meilleurs que ceux de l'extraction, on a pu avoir un résultat parfait pour la base de CNS car le nombre d'attributs est vraiment faible.

De cela on peut en déduire que la classification par Autoencodeur est fiable lorsqu'on possède un bon nombre d'instance mais dans le cas contraire les SVMs sont beaucoup plus fiable

Tableau IV.9. *Résultats de prédiction par SVM pour la base de cancer de colon*

colon	Réduction par 5 couches	Réduction par 7 couches
Nombre d'instance	62	62
Nombre d'attribut	90	50
C	1024	1024
gamma	0.000244	1.33571
Précision maximale	90.32% validation croisée 5,8,9,10,11 et 13	100% validation croisée de 2 à 21
Précision minimale	82.26% validation croisée 4	100% validation croisée 2 à 21
Moyenne de 20 exécution	88.54%	100%
Ecart_Type	2.02%	0.0%

Tableau IV.10. *résulats de prédiction par SVM pour la base de CNS*

CNS	Réduction par 5 couches	Réduction par 7 couches
Nombre d'instance	60	60
Nombre d'attribut	15	8
C	16	4096
gamma	0.169179	0.270130
Précision maximale	100% validation croisée de 2 à 21	100% validation croisée de 2 à 21
Précision minimale	100% validation croisée 2 à 21	100% validation croisée de 2 à 21
Moyenne de 20 exécution	100%	100%
Ecart_Type	0.0%	0.0%

Voila les résultats d'exécution (moyenne de taux de classification) de la méthode SVM sur les bases originales

Tableau IV.11. *Résultats par SVM sur les bases originales*

méthode	cns	Prostate	Leukemia	colon
SVM avec des paramètres par défaut	65	62.57	72.5	80.27
SVM avec des paramètres optimisés	68.61	91.64	98.29	87.37

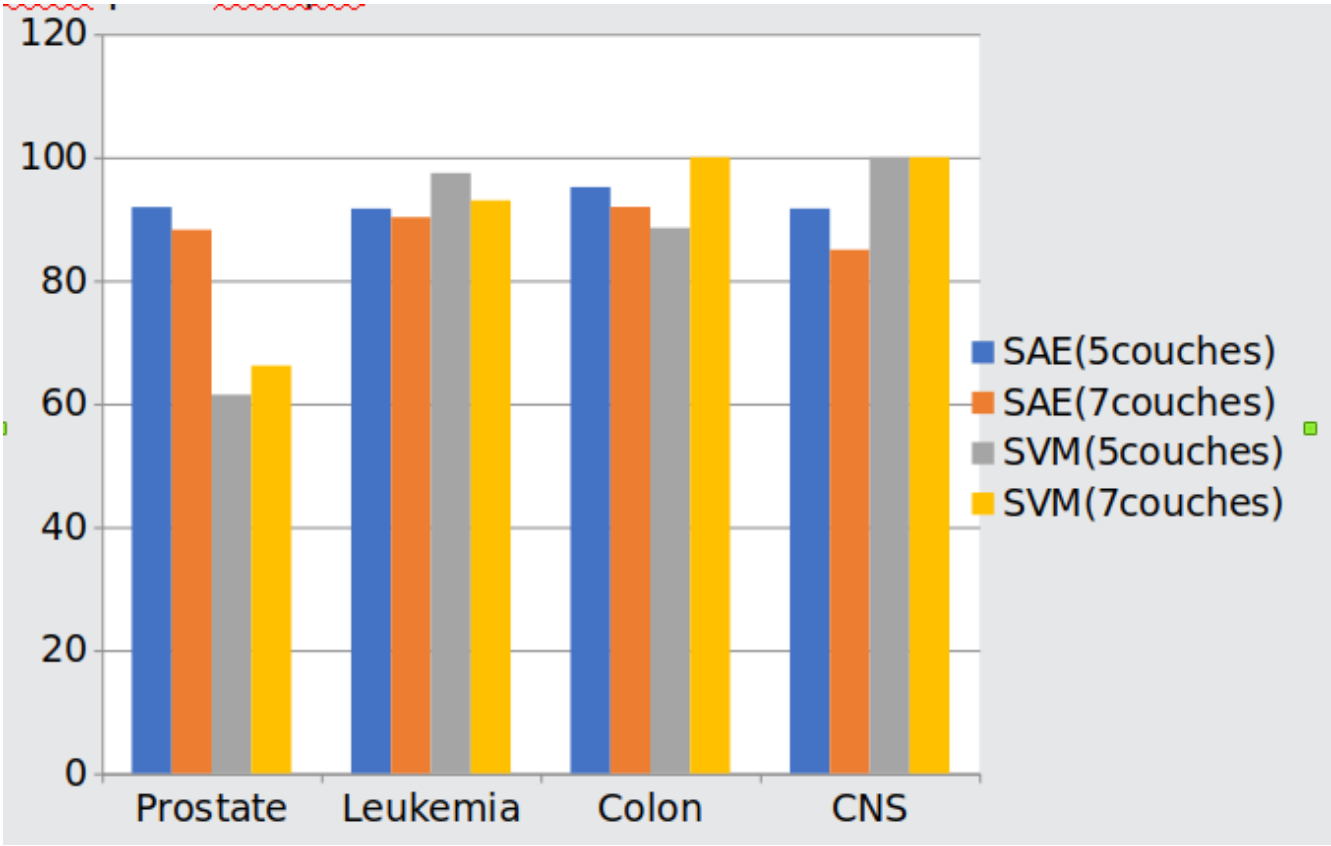


Figure IV.6. *Les taux de précisions des deux méthodes de classification*

6. Conclusion

Ce dernier chapitre décrit les outils utiliser pour réaliser le travail expérimental et donne les résultats obtenus avec un analyse et une comparaison des résultats de prédiction des deux méthodes utilisées.

Conclusion Générale

Dans notre projet de fin d'étude on a atteint les objectifs visés dans le début du travail , dans un premier temps nous avons proposé une méthode pour la réduction de la dimensionnalité de bases de données volumineuses, la réduction est faite en deux phases ; la première en utilisant un outils très efficace dans divers applications ainsi que le filtrage de données qui est le logiciel WEKA, la deuxième phase est faite avec un programme python qui construit et traine un réseau de neurones profond, il s'agit d'un autoencodeur empilé.

Le deuxième objectif visé est la classification des nouvelles instances avec beaucoup moins d'attributs, de même que la réduction, la prédiction des classe a était faite avec deux façon aussi, la première méthode est une suite de la deuxième phase de réduction, s'est à dire en aussi utiliser un réseau de neurones profond pour prédire la classes de nouvelles instances de données.
Les mêmes bases résultantes de la réduction de dimensions ont été entrainées dans un code python exécutant l'algorithme SVM, afin de prédire une autre fois les classes de nouvelles instances de données.

Ces méthodes ont étés tester sur quatre bases de données génétiques à grande dimensions concernant la maladie de cancer, et elles ont donné des résultats très bonnes, donc on peut juger le système de fiabilité et efficacité pour les résultats satisfaisantes qu'il a donné, il y avait quand même une petite différence entre les résultats des autoencodeurs et celles de SVM, et on a constaté que SVM était meilleur que les autoencodeurs empilés car il a pu a trois reprise donner des résultats parfait.

Le travail sur notre projet nous permis d'approfondir notre connaissance dans le système d'exploitation Linux et ses diverses fonctionnalités professionnelles, il nous a permis aussi de connaître de nouvelles bibliothèques python et de savoir le mode de fonction de plusieurs algorithmes de l'apprentissage automatique tels que les différentes architectures de deeplearning et la méthode de classification SVM, sans oublier les connaissances acquits sur le logiciel Weka.

Bien que les objectifs visés ont été atteint, nous espérons l'amélioration de notre travail en utilisant des bases beaucoup plus volumineuses et avec une diversité au niveau de nombre classes et non pas seulement des bases binaires. Et aussi l'intégration de d'autres techniques de sélection et d'extraction d'attributs.

Bibliographie

Références bibliographiques

[Belahcen, 2013]

Belahcen mohamed, Méthodes de réduction et classification, Université mohamed Khider de Biskra, 2013.

[Ben Ayed,2015]

Salma Jamoussi ben Ayed, l'apprentissge profond pour le TALN,Sousse Tunisie,23-25 Mars 2015.

[El ferchichi, 2013]

Sabra El FERCHICHI, Sélection et extraction d'attributs pour les problèmes de classification, 2013.

[Hopkins,1982]

Hopkins,Reseaux de Neurons,1982.

[Hugo, 2009]

Hugo Larochelle,au dela de la machine de boltzmann restreinte,University of toronto, 2009.

[Lamblin, 2007]

Pascal Lamblin, Entrainement de reseaux de neurones profonds,19 avril 2007.

[lei et liu , 2003]

Lei yu et Huan liu, Feature Selection for High-Dimensional Data:A Fast Correlation-Based Filter Solution, arizona state university, tempe, usa, 2003.

[Rana,epps et Jurdak, 2014]

Rana,epps et Jurdak,Gated Recurrent Unit(GRU)for emotion classification from noisy speech,University of southern queensland,Spingfield, 2014.

[Samb, Camara et Ndiaye, 2014]

Samb et Camara et Ndiaye, Approche de séléction d'attributs pour la classification ,Université cheikh anta diop de dakar SENEGAL, 2014.

[Seonwoo et Sungroh]

Seonwoo et sungroh, Deep Learning in bioinformatics, Seoul national University.

[Verleysen,2003]

Verleysen,M.(2003).Learning high-dimensional data. in proceedings of limitations and future trends in neural computation, pages 141-162 IOS press.

[yiming et pedersan, 2003]

yiming et jan pedersan, a comparative study on feature selection in text categorization, rose sunnyvale CA 94089 USA.