

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Mouloud MAMMARI de Tizi-Ouzou
Faculté de Génie électrique et informatique
Département d'informatique



Mémoire de fin de cycle

De Master Professionnel

Domaine : Mathématiques et informatique

Filière : Informatique

Spécialité : Ingénierie des systèmes d'information

Thème

Classification d'images

A l'aide des réseaux de neurones convolutifs

Encadreur : M^{me} FELLAG Samia

Présenté par :

AIT TAHAR Sabrina

OUDIR Dyhia

Année universitaire : 2016/2017

Remerciements

Toute notre gratitude, grâce et remerciements vont au Dieu tout puissant qui nous a donné la force, la patience, le courage et la volonté de mener à terme notre humble travail. Nous n'aurions jamais pu mener à terme notre projet sans la présence de nombreuses personnes qui nous ont aidé et soutenu de près comme de loin.

A notre encadreur Mme Fellag Samia qui nous a assisté tout au long de la réalisation de ce projet, qu'elle trouve ici l'expression de notre profonde gratitude quant à son apport précieux et au temps qu'elle nous a consacré.

Nous tenons à remercier aussi l'université Mouloud Mammeri de nous avoir offert l'opportunité de vivre une expérience aussi enrichissante autant sur le plan humain que professionnel tout en gardant la possibilité de poursuivre les études.

Nos remerciements s'adressent également à tous nos enseignants que nous avons eu l'honneur d'avoir tout au long de notre formation pour tout ce qu'ils nous ont appris, leurs encouragements et leur grande inspiration.

Nous remercions vivement les membres du jury qui nous ont fait l'honneur d'examiner notre travail.

Enfin nous témoignons notre reconnaissance à nos familles et amis qui nous ont apporté leur soutien moral et intellectuel tout au long de notre projet.

Dédicaces

*A cœur vaillant rien d'impossible
A conscience tranquille tout est accessible
Quand il y a la soif d'apprendre
Tout vient à point à qui sait attendre
Malgré les obstacles qui s'opposent
En dépit des difficultés qui s'interposent
Les études sont avant tout
Notre unique et seul atout
Elles représentent la lumière de notre existence
L'étoile brillante de notre réjouissance
Aujourd'hui, ici rassemblés auprès des jurys,
Nous prions Dieu que cette soutenance
Fera signe de persévérance
Et que nous serions enchantés
Par notre travail honoré.*

Nous dédions ce mémoire à :

Nos chers parents, symbole de bonté par excellence, source de tendresse et d'encouragement. Aucune dédicace ne saurait être assez éloquente pour exprimer ce que vous méritez pour tous les sacrifices que vous n'avez cessés de nous donner, rien au monde ne vaut les efforts fournis jour et nuit pour notre éducation et bien être, ce travail est le fruit de vos sacrifices que vous avez consentis pour notre éducation et notre formation.

A notre promotrice Mme Fellag Samia :

Un remerciement particulier et sincère pour tous vos efforts fournis. Vous avez toujours été présente. Que ce travail soit un témoignage de notre gratitude et notre profond respect.

*A tous nous amis, qui d'une part ou d'une autre
Ont contribué à mener à terme notre projet.*

Résumé

Le data mining est l'ensemble des méthodes et techniques destinées à l'exploration et l'analyse des bases de données informatique, entre autre les bases de données « d'images » en vue de détecter dans ces données des structures particulières restituant l'essentiel de l'information utile.

Le data mining offre donc des moyens de traiter ces images car tout comme tout autre donnée, les images, véhiculent des informations, qui une fois exploitées, peuvent servir à une tâche quelconque.

Les réseaux de neurones convolutifs ont apporté d'énormes progrès, pour résoudre le problème de classification d'images et servent donc pour des tâches de reconnaissance visuelle difficiles.

Notre travail consiste à concevoir un système de classification d'images en nous aidant des réseaux de neurones convolutifs, qui appliquent un certain nombre de filtre sur une série d'images en entrées, puis extraire des connaissances utiles à cette classification.

Mots clefs : classification d'images, réseaux de neurones, réseaux de neurones convolutifs, tensorflow, inception, data mining

1 Table des matières

Introduction générale.....	9
1. Introduction	12
2. L'extraction de connaissances à partir de données.....	12
3. Etapes du processus d'ECD	13
3.1 Le nettoyage et l'intégration	14
3.2 Le prétraitement des données	14
3.3 La fouille de données.....	14
3.4 L'évaluation et la présentation	14
4. Définitions du data mining.....	15
5. Les domaines d'application.....	15
6. Données fouillées en data mining	16
6.1 Les fichiers plats.....	16
6.2 Les bases de données relationnelles	16
6.3 Les data warehouses	16
6.4 Les bases de données transactionnelles	16
6.5 Les bases de données multimédia.....	17
6.6 Les bases de données spatiales	17
6.7 Les bases de données de séries temporelles	17
6.8 Le World Wide Web.....	18
7. Tâches du data mining :	18
7.1 La classification :	18
7.2 L'estimation	19
7.3 La prédiction	19
7.4 Les règles d'association	19
7.5 Le clustering (ségrmentation)	19
7.6 La description.....	20
8. Les techniques de Data Mining.....	20
8.1 Analyse du panier de la ménagère	21

8.2	L raisonnement basé sur la mémoire	22
8.3	La détection automatique de clusters.....	22
8.4	Le raisonnement à partir de cas	22
8.5	Les arbres de décision	22
8.5.1	Structure d'un arbre de décision :.....	23
8.5.1	Algorithmes d'induction des arbres de décision.....	24
8.6	Les réseaux de neurones	25
9.	Structure d'un réseau de neurone.....	25
9.1	Les algorithmes génétiques	26
9.2	Les règles associatives	26
9.2.1	Algorithmes d'induction des règles associatives	27
9.3	Les réseaux bayésiens.....	28
9.4	Les séries temporelles	29
10.	Conclusion	29
Chapitre 2 : La tâche de classification		31
1.	Introduction	32
2.	Définition de la classification	32
3.	But de la classification :.....	33
4.	Etapas d'une classification :.....	34
5.	Domaines d'application :	34
6.	Critères pour une bonne classification :	35
7.	Mesures de similarités	36
7.1	Fonctions de similarité :.....	37
8.	Les différentes méthodes de Classification :	38
8.1	La classification automatique « Clustering » (non supervisé) :.....	39
8.1.1	Principe de fonctionnement de la classification automatique	39
8.2	METHODE SUPERVISEE « Classement » ou « DISCRIMINATION »	48
8.2.1	Principe de la méthode de classification supervisée	48
8.2.2	Les k plus proches voisins (K-PPV).....	48
8.2.3	La classification bayésienne	50
8.2.4	L'approche des réseaux de neurones.....	51
9.	Conclusion.....	53

Chapitre 3 : La classification d'images et les réseaux de neurones convolutifs	54
Introduction.....	55
1. Notions d'infographie	55
1.1 Définition d'une image	55
1.2 Caractéristiques d'une image	55
1.3 La numérisation	56
2.1 Classe d'information et classe spectrale	57
2.2 Méthodes supervisées/non supervisées	57
3.1 Perceptron simple	59
3.2 Définition des réseaux de neurones convolutionnels	59
3.3 Contribution des RNC dans la classification d'images.....	60
3.4 Architecture d'un RNC	60
3.4.1 Couches convolutionnelles.....	60
3.4.2 Assemblage des couches (pooling)	61
3.4.3 Couches denses (entièrement connectées)	61
3.5 Fonctionnement d'un CNN	63
3.5.1 Couche non linéaire.....	64
3.5.2 Couche de rectification	65
3.5.4 Couche entièrement connectée.....	67
3.5.5 La fonction SoftMax	68
Conclusion	68
Chapitre 4 : Réalisation et implémentation d'un système de classification d'images	69
Introduction.....	70
1. Architecture du système	70
1.2 Description des couches du système	70
1.2.1 La couche de convolution	70
1.2.2 La couche pooling.....	70
1.2.3 Couche entièrement connectée.....	71
2. Algorithmes de création des couches du réseau	73
2.1 Algorithme d'apprentissage	73
2.2 Algorithme de retro propagation:	74
3. Environnement de développement.....	75

3.1	Le langage python.....	75
3.2	Syntaxe du langage python	75
4	Description du Framework TensorFlow.....	75
4.1	TensorFlow dans la classification d'images.....	76
4.2	TensorFlow comment ça marche ?	76
5	Le modèle Inception de Google	77
6	Configuration de l'environnement	78
6.1	Installation de TensorFlow	78
6.2	Choix des données d'apprentissage	78
6.3	Test de fonctionnalité.....	79
	Conclusion	82
	Conclusion générale.....	84
	Références	86

Introduction générale

Le data mining couvre l'ensemble des outils et méthodes qui permettent l'extraction d'informations pertinentes, autrement dit utiles à la décision, à partir d'un grand volume de données.

Ce vaste domaine ne se limite pas qu'au traitement des données structurées sous forme de tables numériques, il offre également des moyens pour aborder, notamment, les images, car avec la numérisation croissante, l'émergence des réseaux sociaux et des objets connectés, les bases de données sont inondées d'images en tout genre.

Et les images tout comme tout autre document véhiculent des informations, et si ces informations sont exploitées elles peuvent servir à une tâche quelconque. C'est donc tout le travail de la fouille de données : extraire des connaissances à partir de données

Notre cerveau fait que la vision semble facile, Il ne prend aucun effort pour distinguer un lion d'un jaguar, lire un signe ou reconnaître le visage d'un humain. Mais ce sont ici des problèmes difficiles à résoudre pour un ordinateur.

Pour pouvoir, faire apprendre à l'ordinateur à distinguer entre une image et une autre, il existe différentes techniques, mais la plus utilisée et la plus efficace d'entre elles est l'approche neuronale, qui font partie de la catégorie des méthodes de classification.

Au cours des dernières années, d'énormes progrès ont été fait pour résoudre les problèmes classification d'images. En particulier, en utilisant les réseaux neurones convolutifs car cela peut réaliser des performances raisonnables sur des tâches de reconnaissance visuelle difficiles.

La classification d'images consiste à cet effet le fondement de notre travail. Une grande partie de notre mémoire lui sera consacrée.

Intitulé « La classification d'images à l'aide des réseaux de neurones convolutifs », ce mémoire tend ainsi à résoudre le problème de classification d'images grâce à la nouvelle technologie « TensorFlow » qui est un outil open source développé par Google pour optimiser les calculs numériques s'appuyant principalement sur les réseaux de neurones convolutifs.

Nous tentons aussi à démontrer que l'information contenu dans les images est une mine d'or, et que le data mining offre des techniques pour les extraire et en tirer des connaissances qui seront utile pour des utilisations futures.

C'est dans cette perspective que nous avons scindé notre travail en quatre chapitres, les trois premiers développent l'aspect théorique, le dernier sera dédié à la réalisation de notre travail.

Dans le premier chapitre intitulé « le data mining », nous évoquerons les principales tâches et techniques utilisées en data mining, dont la classification qui a pour méthode les réseaux de

neurones. Certains algorithmes utilisés dans la découverte de connaissance y sont aussi mentionnés.

Dans le deuxième chapitre, nous mettons en évidence la tâche de classification en général, en retraçant ses différentes étapes, ainsi que ses divers domaines d'application et les critères qu'il faut pour une bonne classification. Puis les méthodes de classification supervisées et non supervisées y sont mentionnées en détail.

Le troisième chapitre détaillera les méthodes que nous avons utilisé au cours de notre travail, à savoir, les méthodes de classification d'images, parmi elles les réseaux de neurones convolutifs, en spécifiant leur architecture et leur principe de fonctionnement.

Le quatrième et dernier chapitre sera dédié à la présentation de notre application, en retraçant les différentes démarches utilisées, ainsi que les principaux outils qui ont servi à sa

Chapitre 1 : Le data mining

1. Introduction

Dans notre société, il y a de plus en plus de capteurs mesurant n'importe quels changements de l'environnement. Par conséquent, une tâche importante est la compréhension de ces données. Et comme il est dit par Piatetsky-Shapiro, « [...] as long as the world keeps producing data of all kinds [...] at an ever increasing rate, the demand for data mining will continue to grow » [1], c'est à dire que tant que le monde continue de produire des données de toutes sortes, à un taux toujours croissant, la demande en data mining continuera de croître. D'où la fouille de données devient une nécessité, face à l'importance grandissante que prend l'information.

Dans ce présent chapitre, nous évoquons les concepts de base de cette fouille de données, où les différentes étapes du processus d'extraction de connaissances à partir des données sont décrites, les différentes tâches et techniques utilisés ainsi que certains algorithmes principaux.

2. L'extraction de connaissances à partir de données

L'extraction de connaissances à partir de données (ECD) est un processus itératif et interactif d'analyse d'un grand ensemble de données brutes afin d'en extraire des connaissances exploitables par un utilisateur analyste qui y joue un rôle central [2]

Un processus d'ECD est constitué de quatre phases qui sont : le nettoyage et intégration des données, le prétraitement des données, la fouille de données et enfin l'évaluation et la présentation des connaissances [3]

La figure suivante récapitule ces différentes phases ainsi que les enchainements possibles entre ces phases. [4]

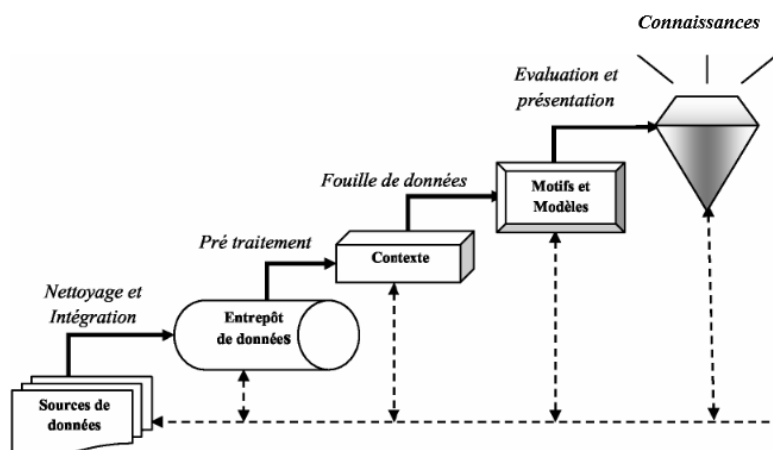


Figure 1 : Processus d'extraction de connaissances à partir de données

Chapitre 1 : Le data mining

3. Etapes du processus d'ECD

L'extraction de connaissance à partir de données comporte quatre étapes principales :

- Le nettoyage et l'intégration
- Le prétraitement des données
- La fouille de données
- L'évaluation et la présentation

Mais avant de présenter ces quatre étapes nous allons d'abord voir qu'est-ce qu'une donnée, une donnée bruitée, une donnée incomplète et une données incohérente.

- a. Donnée :** une donnée en data mining est un enregistrement au sens des bases de données, c'est-à-dire un ensemble de lignes caractérisées par un ensemble d'attributs.
- b. Donnée bruitée :** les données contiennent des enregistrements erronés ou des aberrations

Exemples :

- Distorsion de la voix transmise par téléphone dû à un problème matériel ou un problème de réseau
 - Un salaire inférieur à zéro
- c. Donnée incomplète :** Une donnée incomplète est une donnée ayant des valeurs ou des attributs manquants

Exemple : Ce tableau illustre un exemple d'enregistrement où certains attributs comportent des valeurs manquantes

Age	Statut	Genre	salaire
35	Marié	F	735
40	Marié	M	
20			546
7	Célibataire	F	100

Tableau : Exemple de données avec des valeurs manquantes

- d. Donnée incohérente :** Les données contiennent des enregistrements en conflits ou des contradictions (divergence entre attributs)

Exemple :

Si les données sont des notes d'étudiants, lorsqu'elles dépassent la plage de valeurs autorisées, alors elles ne sont pas comprises dans la fourchette correcte donc ce sont des données incohérentes.

Chapitre 1 : Le data mining

3.1 Le nettoyage et l'intégration

Le nettoyage des données consiste à retravailler ces données bruitées, soit en les supprimant, soit en les modifiant de manière à tirer le meilleur profit.

L'intégration est la combinaison des données provenant de plusieurs sources (bases de données, sources externes, etc.). Le but de ces deux opérations est de générer des entrepôts de données et/ou des magasins de données spécialisés contenant les données retravaillées pour faciliter leur exploitation future. [3]

3.2 Le prétraitement des données

Il peut arriver parfois que les bases de données contiennent à ce niveau un certain nombre de données incomplètes et/ou bruitées. Ces données erronées, manquantes ou inconsistantes doivent être retravaillées si cela n'a pas été fait précédemment. Dans le cas contraire, durant l'étape précédente, les données sont stockées dans un entrepôt. Cette étape permet de sélectionner et transformer des données de manière à les rendre exploitables par un outil de fouille de données.

Cette seconde étape du processus d'ECD permet d'affiner les données. Si l'entrepôt de données est bien construit, le prétraitement de données peut permettre d'améliorer les résultats lors de l'interrogation dans la phase de fouille de données. [3]

3.3 La fouille de données

La fouille de données (data mining en anglais), est le cœur du processus d'ECD. Il s'agit à ce niveau de trouver des pépites de connaissances à partir de données. Tout le travail consiste à appliquer des méthodes intelligentes dans le but d'extraire cette connaissance.

Il est possible de définir la qualité d'un modèle en fonction de critères comme les performances obtenues, la fiabilité, la compréhensibilité, la rapidité de construction et d'utilisation et enfin l'évolutivité.

Tout le problème de la fouille de données réside dans le choix de la méthode adéquate à un problème donné. Il est possible de combiner plusieurs méthodes pour essayer d'obtenir une solution optimale globale. [3]

3.4 L'évaluation et la présentation

Cette phase est constituée de l'évaluation, qui mesure l'intérêt des motifs extraits, et de la présentation des résultats à l'utilisateur grâce à différentes techniques de visualisation. Cette étape est dépendante de la tâche de fouille de données employée.

En effet, bien que l'interaction avec l'expert soit importante quelle que soit cette tâche, les techniques ne sont pas les mêmes. Ce n'est qu'à partir de la phase de présentation que l'on peut employer le terme de connaissance à condition que ces motifs soient validés par les experts du domaine. [3]

4. Définitions du data mining

Plusieurs définitions ont été proposées pour le data mining, notamment :

Le data mining est l'analyse de grandes ensembles de données observationnelles pour découvrir des nouvelles relations entre elles et de les reformuler afin de les rendre plus utilisables de la part des propriétaires. [5]

Le data mining est un domaine interdisciplinaire utilisant dans le même temps des techniques d'apprentissage automatiques, de reconnaissance des formes, des statistiques, des bases de données, et de visualisation pour déterminer les manières d'extraction des informations de très grandes bases de données. [6]

Généralement on s'accorde à définir le data mining comme étant l'ensemble des méthodes et techniques destinées à l'exploration et l'analyse de bases de données informatiques, de façon automatique ou semi-automatique, en vue de détecter dans ces données des règles, des associations, des tendances inconnues ou cachées, des structures particulières restituant l'essentiel de l'information utile tout en réduisant la quantité de données », [7]

5. Les domaines d'application

Le data mining est une spécialité transverse, elle regroupe un ensemble de théories et d'algorithmes ouverts à tout domaine susceptible de drainer une masse importante de données. Parmi ces domaines, on cite :

- **Sociétés commerciales** : Pour connaître par exemple la distribution et l'emplacement géographique des clients, analyser la fidélité des clients dans les opérations à base de crédit, ou encore évaluer la menace concurrentielle dans une région.
- **Grandes surfaces** : L'analyse du ticket de la supérette permet de déterminer les articles souvent achetés simultanément et organiser les promotions en conséquence ;
- **Secteur bancaire** : L'objectif premier des banques est de réduire le risque de prêts bancaires ; la création de modèles à partir des caractéristiques des clients permet de discriminer les clients à risque élevé ;
- **Secteur d'assurances** : Dans le secteur des assurances, le data mining est utilisé pour la détection de fraudes et leur anticipation ;
- **Secteur médical** : Dans le secteur médical, le data mining est naturellement répandu. On trouve par exemple la détermination de groupes de patients susceptibles d'être soumis à des protocoles thérapeutiques déterminés. Les études sur les associations de médicament permettent, entre autres, de révéler les anomalies de prescription.
- **Secteur de l'éducation** : Le data mining est employé dans les établissements scolaires pour améliorer la qualité d'enseignement.

Exemple :

Répartir les élèves ayant une grande capacité d'assimilation dans la même classe.

6. Données fouillées en data mining

Le Data Mining n'est pas spécifique à un seul type de médias ou de données. Il est applicable à n'importe quel type d'information. Il est utilisé et étudié pour les Bases de Données incluant :

6.1 Les fichiers plats

Les fichiers plats sont la source de données la plus commune pour les algorithmes du Data Mining et particulièrement dans le niveau de recherches. Les fichiers plats sont des fichiers de données simples dans le format texte ou binaire avec une structure connue par l'algorithme du Data Mining qui va y être appliqué. [8]

6.2 Les bases de données relationnelles

Les algorithmes du Data Mining appliqués sur des Bases de Données relationnelles sont plus polyvalents que les algorithmes spécifiquement faits pour les fichiers plats puisqu'ils peuvent profiter de la structure inhérente aux bases de données relationnelles. Le Data Mining peut profiter du SQL pour la sélection, la transformation et la consolidation.

6.3 Les data warehouses

Un Data Warehouse est un support de données rassemblées de multiples sources de données (souvent hétérogènes) et est destinée à être utilisée dans l'ensemble sous le même schéma unifié.

Exemple :

Supposons une entreprise OurVideoStore qui a des contrats d'exclusivité dans toute l'Amérique du Nord. La plupart des magasins de vidéos appartenant à l'entreprise OurVideoStore peuvent avoir des bases de données et des structures différentes.

Si le directeur de l'entreprise veut accéder aux données de tous les magasins pour prendre des décisions stratégiques, il serait plus approprié si toutes les données étaient stockées dans un seul emplacement avec une structure homogène qui permet l'analyse interactive des données.

Autrement dit, les données de différents magasins peuvent être chargées, nettoyées, transformées et intégrées ensemble. Pour faciliter la prise de décisions et les vues multidimensionnelles, les Data Warehouses sont souvent modélées par une structure de données multidimensionnelle. [9]

6.4 Les bases de données transactionnelles

En général, une Base de Données transactionnelle est un fichier où chaque enregistrement représente une transaction. Une transaction contient un identifiant unique de transaction

Chapitre 1 : Le data mining

(transactionID) et une liste d'items composant la transaction (les achats d'un client lors d'une visite).

Les bases de données transactionnelles peuvent contenir d'autres informations tels que la date de la transaction, l'identifiant du consommateur, l'identifiant de la personne qui a vendu, et ainsi de suite.

Pour le Data Mining sur ce type de données nous utilisons souvent les règles d'association (appelées aussi Analyse du panier de la ménagère) dans lesquelles les associations des items arrivant ensemble ou séquentiellement soient étudiées. [8]

6.5 Les bases de données multimédia

Les bases de données multimédia comportent des documents sonores, des vidéos, des images et des médias en textes et audio. Elles peuvent être stockées sur des bases de données orientées objets ou objets relationnelles ou simplement sur un fichier système.

Le multimédia est caractérisé par sa haute dimension ce qui rend le datamining sur ce type de données très difficile. Le data mining sur les supports des multimédias exige la vision par ordinateur, l'infographie, l'interprétation des images et les méthodologies de traitement de langages naturels [9] [10].

6.6 Les bases de données spatiales

Ce sont des bases de données, qu'en plus de leurs données usuelles, elles contiennent des informations géographiques comme les cartes et les positionnements mondiaux ou régionaux. De telles bases de données présentent de nouveaux défis aux algorithmes de data mining [15].

6.7 Les bases de données de séries temporelles

Les bases de données de séries temporelles contiennent des données relatives au temps, comme les données du marché boursier ou les activités enregistrées. Ces bases de données ont couramment un flux continu de nouvelles données entrantes, qui parfois rend l'analyse en temps réel un besoin exigeant.

Le data mining pour ce genre de bases de données est généralement l'étude des tendances et des corrélations entre les évolutions des différentes variables, aussi bien que la prédiction des tendances et des mouvements des variables par rapport au temps.

Exemple :

Par exemple, une base de données du trafic automobile qui stocke une description symbolique de séries temporelles de ce dernier, il sera possible de répondre à la requête : « définir les grands axes où le commerce est fluide le week-end [9] [11]

6.8 Le World Wide Web

Le World Wide Web est le support de données le plus hétérogène et le plus dynamique disponible. Un grand nombre d'auteurs et d'éditeurs contribuent sans arrêt à son accroissement et évolution, et chaque jour un énorme nombre d'utilisateurs accède à ses ressources.

Les données dans le World Wide Web sont organisées dans des documents interconnectés. Ces documents peuvent être des textes, audio, vidéos, données brutes et même des applications.

Conceptuellement, le World Wide Web est composé de trois grands composants : le contenu du Web, qui englobe les documents disponibles ; la structure du Web, qui garantit les hyperliens et les relations entre documents ; et l'usage du Web, en décrivant quand et comment les ressources seront accédées. Une quatrième dimension peut être ajoutée concernant la nature dynamique ou l'évolution des documents.

Le data mining pour le World Wide Web, ou le web mining, essaie d'aborder toutes ces questions et il est souvent divisé en contenu Web mining, la structure Web mining et l'usage Web mining. [9]

7. Tâches du data mining :

Pour pouvoir extraire les données pertinentes à une entreprise parmi leur abondance, différentes méthodes sont mises en œuvre. Ces dernières se basent sur l'identification de liens logiques entre différents motifs et tendances, afin d'établir des statistiques. Parmi elles, on cite :

7.1 La classification :

La classification a pour rôle d'affecter un objet à une classe prédéfinie selon une mesure de proximité

Les techniques de classification commencent par définir un « plan d'expérience » ou un ensemble de données d'apprentissage sur lequel on applique les méthodes de classification.

Exemples :

Voici quelques exemples de l'utilisation des tâches de classification :

- Attribuer ou non un prêt à un client dans une banque
- Établir un diagnostic médical
- Accepter ou refuser un retrait dans un distributeur de billets
- Attribuer un sujet principal à un article de presse [12]

7.2 L'estimation

L'estimation est similaire à la classification à part que la variable de sortie est numérique plutôt que catégorielle. Etant donné un enregistrement de données (une table avec attributs et valeurs), l'estimation consiste à compléter une valeur manquante dans un champ particulier.

Exemple :

On cherche à estimer la lecture de tension systolique d'un patient dans un hôpital, en se basant sur l'âge du patient, son genre, son indice de masse corporelle et le niveau de sodium dans son sang. La relation entre la tension systolique et les autres données vont fournir un modèle d'estimation. Et par la suite nous pouvons appliquer ce modèle dans d'autres cas [8] [13].

Souvent, la classification et l'estimation sont utilisés ensemble, comme quand le Data Mining est utilisée pour prévoir qui va probablement répondre à une offre de transfert d'équilibre (de solde) de carte de crédit et aussi évaluer la taille de l'équilibre (du solde) à être transféré.

7.3 La prédiction

La prédiction est semblable à la classification et l'estimation, sauf que pour la prévision, les résultats se situent dans le futur. Exemples de tâches de prédiction :

- Prédire, au vu de leurs actions passées, les départs de clients dans une banque
- Prévoir le champion de la coupe du monde en football en se basant sur la comparaison des statistiques des équipes. [13]

7.4 Les règles d'association

Les règles d'associations ou groupement par similitude consiste à déterminer quels attributs "vont ensemble". C'est la tâche la plus répandue dans le monde du business, où elle est appelée l'analyse d'affinité ou l'analyse du panier du marché, est l'association des recherches pour mesurer la relation entre deux et plusieurs attributs. Les règles d'associations sont de la forme "Si antécédent, alors conséquent".

Exemple :

Trouver dans un supermarché quels produits sont achetés ensemble et quels sont ceux qui ne s'achètent jamais ensemble, déterminer la proportion des cas dans lesquels un nouveau médicament peut générer des effets dangereux [9].

7.5 Le clustering (ségrégation)

Le clustering désigne le groupement des données, des observations ou des cas dans des classes d'objets similaires. Un cluster maximise la similarité des objets du même cluster et minimise la similarité des objets de clusters différents.

Chapitre 1 : Le data mining

En effet, il n'y a pas de variable cible pour le clustering. La tâche de clustering ne cherche pas à estimer ou à prédire la valeur d'une variable cible. Mais plutôt à segmenter l'ensemble des données en sous-groupes relativement homogènes à l'aide de mesures de distances.

La segmentation est une tâche d'apprentissage « non supervisée » car on ne dispose d'aucune autre information préalable que la description des exemples.

Après application de l'algorithme et donc lorsque les groupes ont été construits, d'autres techniques ou une expertise doivent dégager leur signification et leur éventuel intérêt.

Les algorithmes du clustering peuvent être appliqués dans des différents domaines, tel que : la découverte des groupes de clients ayant des comportements semblables.

La Classification des plantes et des animaux étant donnée leurs caractéristiques, et la segmentation des observations des épices pour identifier les zones dangereuses. [13]

7.6 La description

Parfois le but du Data Mining est simplement de décrire ce qui se passe sur une base de données compliquée en expliquant les relations existantes dans les données pour en premier lieu comprendre le mieux possible les individus, les produits et les processus présents sur cette base.

Une bonne description d'un comportement implique souvent une bonne explication de celui-ci [14]. Dans la société Américaine nous pouvons prendre comme exemple comment une simple description, « les femmes supportent le parti Démocrate plus que les hommes », peut provoquer beaucoup d'intérêt et promouvoir les études de la part des journalistes, sociologues, économistes et les spécialistes en politiques [15].

8. Les techniques de Data Mining

Les techniques ou les outils du Data Mining représentent une partie très importante dans la tâche de l'apprentissage machine (machine learning), nous allons citer quelques-unes afin de donner une description générale sans entrer dans le détail.

Ces techniques se divisent en connaissances dirigées et connaissances non dirigées, dans le cas des connaissances dirigées ou apprentissage supervisé, la robustesse de l'algorithme dépendra de la précision de son entraînement.

Un algorithme apprenant des contenus supervisés produit une carte interne qui permet sa réutilisation pour classifier de nouvelles quantités de données.

Exemple :

Chapitre 1 : Le data mining

Prenons l'exemple d'un algorithme qui détecte les visages, un utilisateur devra lui montrer ce qu'est un visage et ce qui ne l'est pas afin qu'il puisse apprendre et prédire si les prochaines photos le sont ou non.

En résumé, l'algorithme apprend sur des exemples, dans le cas de cette utilisation les exemples ont besoin d'être étiqueté afin d'assurer l'efficacité de son apprentissage.

Dans le cas de l'apprentissage non-supervisé, il n'y a pas besoin de l'intervention d'un être humain, car l'algorithme va par lui-même comprendre comment différencier un visage d'un paysage en cherchant leurs corrélations.

Étant donné qu'un algorithme ne peut pas tout simplement savoir ce qui constitue un visage, la méthode non-supervisé va donc partitionner et classer les données dans des groupes homogènes (clustrering)

Apprentissage Supervisé VS Non-supervisé	
Classification	Clustering
<ul style="list-style-type: none">- Nombre de classes connues- Sur entraînement- Utilisé pour classifier des données futures	<ul style="list-style-type: none">- Nombre de classes inconnues- Pas de connaissance préalable- Utilisé pour comprendre et explorer les données

Figure 2: Différence entre apprentissage supervisé et apprentissage non supervisé

La différence distincte entre l'apprentissage supervisé et l'apprentissage non-supervisé est le fait que l'apprentissage non-supervisé cherche à trouver des partitions de modèles par lui-même, l'extraction des données est donc descriptive.

L'apprentissage supervisé est utilisé quand l'utilisateur sait labelliser les informations, l'extraction des données est donc prédictive. [16]

8.1 Analyse du panier de la ménagère

L'analyse du panier de la ménagère est un moyen de trouver les groupes d'articles qui vont ensemble lors d'une transaction. C'est une technique de découverte de connaissances non dirigée (de type analyse de clusters) qui génère des règles et supporte l'analyse des séries temporelles (si les transactions ne sont pas anonymes).

Les règles générées sont simples, faciles à comprendre et assorties d'une probabilité, ce qui en fait un outil agréable et directement exploitable par l'utilisateur métier. [17]

Exemples :

Chapitre 1 : Le data mining

Le client qui achète de la peinture achète un pinceau

Le client qui achète un téléviseur achète un magnétoscope

8.2 L raisonnement basé sur la mémoire

Le raisonnement basé sur la mémoire (RBM) est une technique de prédiction et de classification utilisée dans le cadre de la découverte de connaissances dirigée. Elle peut être également utilisée pour l'estimation.

Pour chaque nouvelle instance présentée, le système recherche le(s) voisin(s) le(s) plus proche(s) et procède ainsi à l'affectation ou estimation. L'avantage du RBM est qu'il est facile à mettre en œuvre, très stable et supporte tout type de données. [17]

8.3 La détection automatique de clusters

La détection automatique de clusters est une technique de découverte de connaissances non dirigée (ou apprentissage sans supervision). Elle consiste à regrouper les enregistrements en fonction de leurs similitudes.

Chaque groupe représente un cluster. C'est une excellente technique pour démarrer un projet d'analyse ou de data mining. Les groupes de similitudes permettront de mieux comprendre les données et d'imaginer comment les utiliser au mieux. [17]

8.4 Le raisonnement à partir de cas

Le raisonnement à partir de cas (RàPC) est un paradigme de raisonnement qui utilise les expériences précédentes pour résoudre de nouveaux problèmes [18]. Par le terme "expériences précédentes", nous voulons dire : expériences d'analyse, de planification, de stratégie...

Le champ d'application du RàPC est très vaste aujourd'hui. Il est appliqué dans tous les domaines où on a besoin d'utiliser ou de synthétiser les expériences passées afin de proposer de nouvelles solutions.

8.5 Les arbres de décision

Les arbres de décisions sont des outils d'aide à la décision qui permettent selon des variables discriminantes de répartir une population d'individus en groupes homogènes en fonction d'un objectif connu.

Les arbres de décision sont des outils puissants et populaires pour la classification et la prédiction. Un arbre de décision permet à partir des données connues sur le problème de donner des prédictions par réduction, niveau par niveau, du domaine des solutions.

Chapitre 1 : Le data mining

8.5.1 Structure d'un arbre de décision :

Chaque nœud interne d'un arbre de décision permet de répartir les éléments à classer de façon homogène entre ses différents fils en portant sur une variable discriminante de ces éléments.

Les branches qui représentent les liaisons entre un nœud et ses fils sont les valeurs discriminantes de la variable du nœud. Et en fin, les feuilles d'un arbre de décision représentent les résultats de la prédiction des données à classer [19].

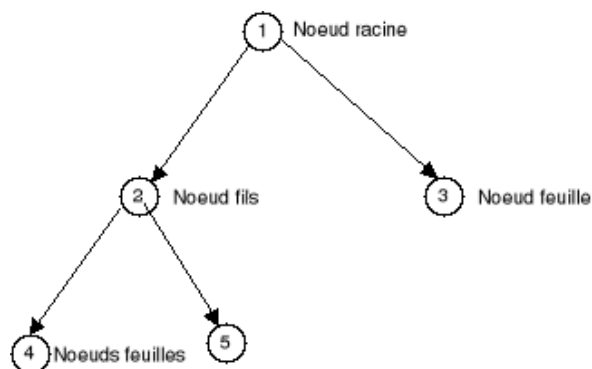


Figure 3: Structure d'un arbre de décision

Exemple :

On considère l'arbre de décision de la figure suivante. La population est constituée d'un ensemble de patients. Il y a deux classes : malade et bien portant. Les descriptions sont faites avec les deux attributs : Température qui est un attribut à valeurs décimales et gorge irritée qui est un attribut logique.

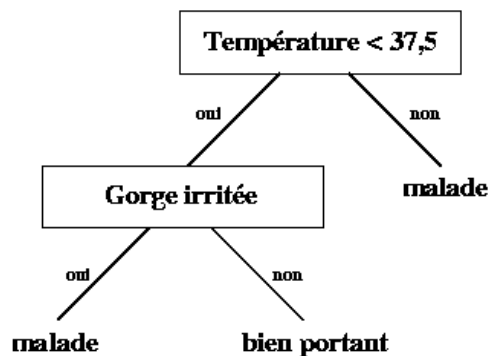


Figure 4: Exemple d'arbre de décision

Pour construire un tel arbre, plusieurs algorithmes existent : CAID, ID3, C4.5,... etc. On commence généralement par le choix d'un attribut puis le choix d'un nombre de critères pour son nœud.

Chapitre 1 : Le data mining

On crée pour chaque critère un nœud concernant les données vérifiant ce critère. L'algorithme continue d'une façon récursive jusqu'à obtenir des nœuds concernant les données de chaque même classe.

8.5.1 Algorithmes d'induction des arbres de décision

On dénombre plusieurs algorithmes pour construire des arbres de décision, parmi lesquels :

- **CHAID**

L'acronyme CHAID signifie en fait "*Chi-squared Automatic Interaction Detector*". Il s'agit de l'une des méthodes d'arbres de classification les plus anciennes. La méthode CHAID va "construire" des arbres de décision non-binaires (c'est-à-dire des arbres de décision dans lesquels nous pouvons avoir plus de deux branches connectées à un même nœud (racine)), en utilisant un algorithme assez simple qui est particulièrement bien adapté à l'analyse des fichiers de données les plus volumineux [20] [21].

- **ID3**

C'est un algorithme de classification supervisé, c'est-à-dire qu'il se base sur des exemples déjà classés dans un ensemble de classes pour déterminer un modèle de classification. Le modèle que produit ID3 est un arbre de décision. Cet arbre servira à classer de nouveaux échantillons.

Chaque exemple en entrée est constitué d'une liste d'attributs. Un de ces attributs est l'attribut « cible » et les autres sont les attributs « non cibles ». On appelle aussi cette "cible" la "classe".

En fait l'arbre de décision va permettre de prédire la valeur de l'attribut « cible » à partir des autres valeurs. Bien entendu, la qualité de la prédiction dépend des exemples : plus ils sont variés et nombreux, plus la classification de nouveaux cas sera fiable.

- **CART**

L'**algorithme CART** dont l'acronyme signifie « Classification And Regression Trees », s'attelle à construire un arbre de décision en classifiant un ensemble d'enregistrements. Cet arbre fournit un modèle pour classer de nouveaux échantillons. Il a été publié par Leo Breiman en 1984.

Ce dernier construit un arbre de décision d'une manière analogue à l'algorithme ID3. Contrairement à ce dernier, l'arbre de décision généré par CART est binaire (un nœud ne peut avoir que 2 fils) [10]

Chapitre 1 : Le data mining

- C4.5

L'**algorithme C4.5** est un algorithme de classification supervisé, publié par Ross Quinlan en 1993. Il est basé sur l'algorithme ID3 auquel il apporte plusieurs améliorations.

À partir d'un échantillon d'apprentissage composé d'une variable objectif ou variable prédite et d'au moins une variable d'apprentissage ou variables prédictives, C4.5 produit un modèle de type arbre de décision.

Ce modèle permet de prédire pour un individu la valeur estimée de la variable objectif en fonction des valeurs prise par les variables "prédictives". L'algorithme C4.5 se base sur une mesure de l'entropie dans l'échantillon d'apprentissage pour produire le modèle (graphe d'induction) [22].

8.6 Les réseaux de neurones

Les réseaux de neurones sont des outils très utilisés pour la classification, l'estimation, la prédiction et la segmentation [23]. Un réseau de neurones « ou réseau neuronal » a une architecture calquée sur celle du cerveau, organisée en neurones et synapses, et se présente comme un ensemble de nœuds « ou neurones formels, ou unités » connectés entre eux [24].

9. Structure d'un réseau de neurone

Un réseau est organisé en couches, il peut être constitué d'une unique couche, et est alors appelé perceptron. Chaque neurone de la couche reçoit toutes les informations entrées et renvoie son résultat vers un unique neurone de sortie. Ce qu'il renvoie est la sortie finale du réseau.

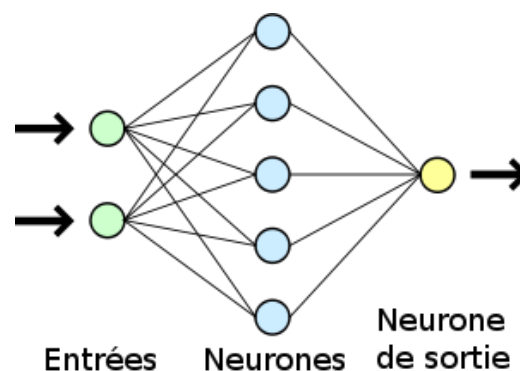


Figure 5: Structure d'un réseau de neurone à couche unique (perceptron)

Ces réseaux de neurones feront l'objet de notre travail, et seront donc plus détaillés dans le chapitre suivant.

Chapitre 1 : Le data mining

9.1 Les algorithmes génétiques

Ces algorithmes constituent parfois une alternative intéressante aux réseaux de neurones mais sont le plus souvent complémentaires.

▪ Principe de base des algorithmes génétiques

Le principe de fonctionnement d'un algorithme génétique est le suivant :

1. Codage du problème sous forme d'une chaîne binaire.
2. Génération aléatoire d'une population. Celle-ci contient un pool génétique qui représente un ensemble de solutions possibles.
3. Calcul d'une valeur d'adaptation pour chaque individu. Elle sera fonction directe de la proximité des différents individus avec l'objectif, on parle ici d'évaluation.
4. La Sélection des individus doit se reproduire en fonction de leurs parts respectives dans l'adaptation globale.
5. Croisement des génomes des parents.
6. Sur la base de ce nouveau pool génétique, on repart à partir du point 3.

9.2 Les règles associatives

Les règles associatives sont des règles extraites d'une base de données transactionnelles et qui décrivent des associations entre certains éléments.

Elles sont fréquemment utilisées dans le secteur de la distribution des produits où la principale application est l'analyse du panier de la ménagère (Market Basket Analysis) dont le principe est l'extraction d'associations entre produits sur les tickets de caisse.

Le but de la méthode est l'étude de ce que les clients achètent pour obtenir des informations sur qui sont les clients et pourquoi ils font certains achats.

La méthode recherche quels produits tendent à être achetés ensemble et peut être appliquée à tout secteur d'activité pour lequel il est intéressant de rechercher des groupements potentiels de produits ou de services : services bancaires, services de télécommunications, par exemple.

Exemples :

Les règles associatives peuvent être utilisées dans le secteur médical pour la recherche de complications dues à des associations de médicaments ou à la recherche de fraudes en recherchant des associations inhabituelles.

Une règle d'association est de la forme :

Si condition alors résultat. Dans la pratique, nous nous limitons généralement à des règles où la condition se présente sous la forme d'une conjonction d'apparition d'articles et le résultat se constitue d'un seul article.

Chapitre 1 : Le data mining

Exemple :

Une règle à trois articles sera de la forme : Si X et Y alors Z ; règle dont la sémantique peut être énoncée : Si les articles X et Y apparaissent simultanément dans un achat alors l'article Z apparaît [19] [25].

Les algorithmes des règles d'association utilisent les notions de support et de confiance pour déterminer la pertinence des associations.

Le support d'une règle est défini par :

$$Support(A) = \frac{count(A)}{|T|}$$

Où A est un item (un produit) et T une base de données transactionnelles dont est issu A.

La confiance d'une règle est définie par :

$$Confidence(A \Rightarrow B) = \frac{Support(A \Rightarrow B)}{Support(A)}$$

Les règles associatives générées sont retenues si elles ont un support supérieur à minSupp et une confiance supérieure à minConf. Ces deux constantes –dépendantes de la base de données- sont définies par l'utilisateur du système de façon empirique. [26] [27]

9.2.1 Algorithmes d'induction des règles associatives

Les règles associatives comptent plusieurs algorithmes, les plus connus sont :

- **L'algorithme Apriori**

Apriori est un algorithme classique de recherche de règles d'association créé par Agrawal, Imielinski et Swami en 1993. Comme tous les algorithmes de découvertes d'associations, il travaille sur des bases de données transactionnelles (des enregistrements de transactions).

Pour révéler la pertinence d'une règle on utilise deux concepts qui sont le support [30] et la confiance [27]. Afin d'être retenue, chaque règle devra avoir un support supérieur à minSupp et une confiance supérieure à minConf. Ces deux valeurs étant définies empiriquement par l'utilisateur du système.

- **Algorithme FP-Growth**

FP-Growth (Frequent-Pattern Growth) est un algorithme complètement innovant par rapport aux autres algorithmes de recherche de règles associatives presque tous basés sur Apriori.

Chapitre 1 : Le data mining

Créer par Han, Pei, Yin et Mao en 2000, l'algorithme utilise une structure de données compacte appelé Frequent-Pattern tree et qui apporte une solution au problème de la fouille de motifs fréquents dans une grande base de données transactionnelle [28].

- **Algorithme Eclat**

Eclat est le fondateur de la seconde grande famille d'algorithmes de recherche de règles d'associations avec l'algorithme Apriori. Développé par M. Zaki en 2000, l'algorithme démarre avec la liste des éléments fréquents puis l'algorithme est réitéré en rajoutant les ensembles fréquents à l'ensemble des candidats C jusqu'à ce que cet ensemble soit vide.

L'ensemble des transactions de D qui contiennent l'élément X est défini par :

$$D(X) = \{T \in D | X \subseteq T\}$$

Les relations sont filtrées selon l'équation suivante :

$$freq(C_0) = \{(x, D_x) | (x, D_x) \in C_0, |D_x| \geq \text{minsup}\}$$

9.3 Les réseaux bayésiens

Un réseau bayésien est un système représentant la connaissance et permettant de calculer des probabilités conditionnelles apportant des solutions à différentes sortes de problématiques.

Structure d'un réseau bayésien :

La structure de ce type de réseau est simple : un graphe dans lequel les nœuds représentent des variables aléatoires, et les arcs (le graphe est donc orienté) reliant ces dernières sont rattachées à des probabilités conditionnelles.

Notons que le graphe est acyclique : il ne contient pas de boucle. Les arcs représentent des relations entre variables qui sont soit déterministes, soit probabilistes.

Ainsi, l'observation d'une ou plusieurs causes n'entraîne pas systématiquement l'effet ou les effets qui en dépendent, mais modifie seulement la probabilité de les observer.

L'intérêt particulier des réseaux bayésiens est de tenir compte simultanément de connaissances à priori d'experts (dans le graphe) et de l'expérience contenue dans les données.

Pour résumer, un réseau bayésien est un modèle probabiliste graphique permettant d'acquérir, de capitaliser et d'exploiter des connaissances, né du besoin de créer des systèmes experts à base de probabilités. [19]

Chapitre 1 : Le data mining

Les réseaux bayésiens sont à la fois des modèles de représentation des connaissances et des machines à calculer les probabilités conditionnelles.

Exemple :

Cette figure illustre un réseau bayésien représentant les dépendances entre cinq variables, l'absence d'un arc allant de X_1 à X_5 signifie que la saison n'a pas une influence directe sur l'état glissant ou pas du chemin. Egalement un chemin mouillé peut être dû à un arrosage ou la pluie.

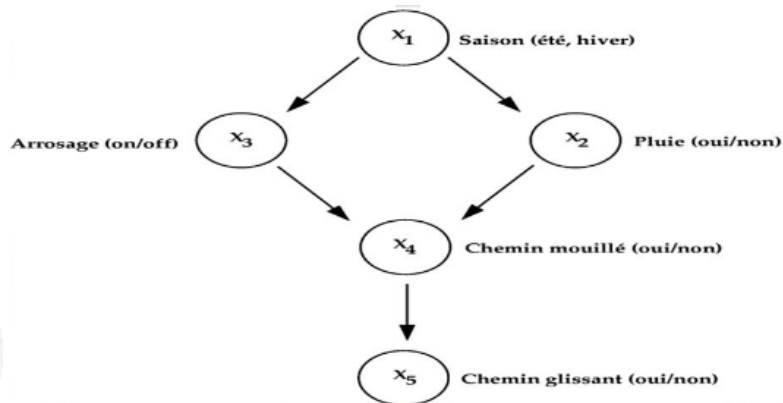


Figure 6: Exemple d'un réseau bayésien

9.4 Les séries temporelles

Les séries temporelles représentent la mesure de certaines caractéristiques en fonction du temps. Les statisticiens ont établi ce concept et ses propriétés depuis plus d'un siècle [20] ; mais leurs techniques s'avèrent insuffisantes pour les énormes quantités de données des séries temporelles réelles.

En tant qu'outil de Data-Mining les 'grandes' séries temporelles sont traitées selon plusieurs aspects. Elles sont d'abord représentées de telle sorte à ce que l'on puisse définir une mesure de similarité entre séries.

10. Conclusion

Dans ce chapitre, nous avons exposé le processus ECD et ses différentes étapes en général, la fouille de données et les techniques utilisées pour extraire l'information utile et les différentes sortes de données travaillées.

Dans notre travail nous nous intéressons aux techniques de la classification automatique, nous avons vu qu'elle permet de regrouper des objets (individus ou variables) en groupes ou classes. Les détails, feront l'objet du chapitre suivant.

Chapitre 2 : La tâche de classification

1. Introduction

La problématique de la classification consiste à affecter les objets d'un ensemble de données à des catégories ou classes prédéfinies. Ce type de question fait partie des problèmes rencontrés lors de la phase du groupement et la classification de données.

Dans ce chapitre nous présenterons un panorama des méthodes de classification les plus connues et qui font référence à l'existence de groupes ou classes de données,

2. Définition de la classification

La classification est une discipline reliée de près ou de loin à plusieurs domaines, elle est connue aussi sous des noms variés (classification, clustering, segmentation, ...) selon les objets qu'elle traite et les objectifs qu'elle vise à atteindre.

Pour attribuer une définition au terme « classification », il faudrait d'abord définir ses racines, ça vient du verbe “classer” qui désigne plus une action qu'un domaine, ou plutôt une série de méthodes qu'une théorie unifiée.

En mathématique, On appelle classification, la catégorisation algorithmique d'objets. Elle consiste à attribuer une classe ou catégorie à chaque objet (ou individu) à classer, en se basant sur des données statistiques.

Elle fait couramment appel aux méthodes d'apprentissage et est largement utilisée en reconnaissance de formes.

Il est important de noter qu'il ne faut pas confondre entre ces deux termes : « classification » et « classement », au fait le mot classification en anglais signifie une chose, alors que le même mot en français ait une autre signification (utilité).

Dans un classement on affecte les objets à des groupes préétablis, c'est le but de l'analyse discriminante qui est de fixer des règles pour déterminer la classe des objets.

La classification est donc, en quelque sorte, le travail préliminaire au classement, à savoir la recherche des classes "naturelles" dans le domaine étudié, en anglais « Cluster Analysis ».

Cette collision entre les termes peut se résumer comme suit :

D'une manière générale en vertu de ces définitions, la classification se définit alors comme une méthode mathématique d'analyse de données, pour faciliter l'étude d'une population

Chapitre 2 : La tâche de classification

d'effectif important, généralement des bases d'observations caractérisant un domaine particulier (animaux, plantes, maladies, gènes, ... etc.), où on les regroupe en plusieurs classes,

3. But de la classification :

L'objectif d'une classification est de répartir l'échantillon en groupes d'observations homogènes, chaque groupe étant bien différencié des autres.

Le plus souvent, cependant, cet objectif est plus raffiné ; on veut, en général, obtenir des sections à l'intérieur des groupes principaux, puis des subdivisions plus petites de ces sections, et ainsi de suite.

En bref, on désire avoir une hiérarchie, c'est à dire une suite de partitions "emboîtées", de plus en plus fines, sur l'ensemble d'observations initial.

Une telle hiérarchie peut avantageusement être résumée par un arbre hiérarchique (figure 1) dont les nœuds (m, n, p, q) symbolisent les diverses subdivisions de l'échantillon ; les éléments de ces subdivisions étant les objets (a, b, c, d, e), placés à l'extrémité inférieure des branches qui leur sont reliées.

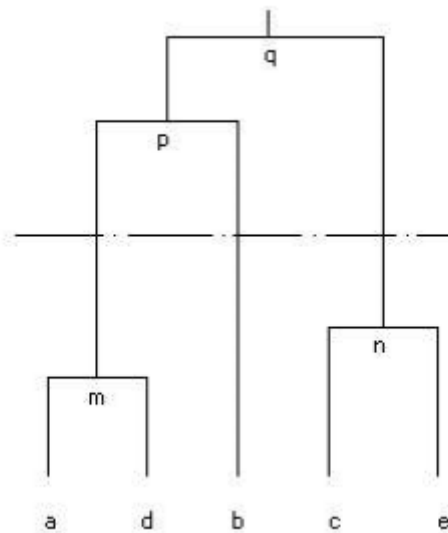


Figure 7: Exemple d'arbre hiérarchique

Chapitre 2 : La tâche de classification

Cet arbre porte sur cinq objets a, b, c, d, e. Les points m, n, p, q sont les nœuds de l'arbre. Le trait horizontal mixte indique un niveau de troncature définissant une partition en trois classes.

Le niveau des nœuds, qui est le plus souvent chiffré, est sensé indiquer un degré de ressemblance entre les objets correspondants. Ainsi, sur notre figure 1, les objets a et d se ressemblent plus que les objets c et e.

Remarquons, en passant, que si on coupe cet arbre à un niveau intermédiaire entre n et p, on obtient une partition en trois classes de l'ensemble étudié, à savoir les parties {a, d}, {b}, {c, e}. En faisant varier ce niveau de troncature on obtient les diverses partitions constituant la hiérarchie. [29]

4. Etapes d'une classification :

1. **Choix des données :** Il faut sélectionner les individus à classer et les variables qui serviront pour critère de classification. Si par exemple on veut faire une classification selon le milieu de résidence, il faut stratifier le fichier initial en deux sous-fichiers.

On sélectionne les individus qui résident en milieu urbain et ceux en milieu rural puis lancer la classification sur chacun de ces deux sous-fichiers.

2. **Calcul des similarités entre les n individus à partir des données initiales :** On choisit une distance ou un indice d'écart entre paires d'individus. La distance généralement utilisée dans les algorithmes de classification hiérarchique est la distance euclidienne.

3. Choix d'un algorithme de classification et exécution.

4. L'interprétation des résultats :

- évaluation de la qualité de la classification,
- description des classes obtenues.

5. Domaines d'application :

Chapitre 2 : La tâche de classification

La classification comme dit préalablement joue un rôle dans presque toutes les sciences et techniques qui font appel à la statistique multidimensionnelle.

A titre d'exemple les sciences biologiques : botanique, zoologie, écologie, ... qui utilisent le terme "taxinomie" pour désigner l'art de la classification. Ainsi que les sciences de la terre et des eaux : géologie, pédologie, géographie, étude des pollutions, font grand usage de classification.

Une autre forte utilité des techniques de classification dans les sciences de l'homme : psychologie, sociologie, linguistique, archéologie, histoire, etc....et sans oublier les techniques dérivées comme les enquêtes d'opinion, le marketing, etc.

Ces dernières emploient parfois les mots de "typologie" et "segmentation" pour désigner la classification, Citons encore la médecine [30], l'économie, l'agronomie ... etc.

Dans toutes ces disciplines la classification peut être employée comme un domaine particulier ; mais elle est souvent vue comme une méthode complémentaire à d'autres méthodes statistiques.

Elle est très largement utilisée à l'interprétation des graphiques d'analyse factorielle, ou bien déterminer des groupes d'objets homogènes, préalablement à une régression linéaire multiple.

6. Critères pour une bonne classification :

L'objectif principal des techniques de classification est de trouver une partition où les objets d'une classe devraient être semblables (entre eux), les objets de différentes classes devraient être différents. Une bonne classification devrait accomplir différents critères :

6.1 Validité : Elle peut se définir par :

Chaque classe d'une partition doit être homogène : Les objets qui appartiennent à la même classe doivent être semblables. Les classes doivent être isolées entre elles : Les objets de différentes classes doivent être différents. La classification doit s'adapter aux données : La classification doit pouvoir expliquer la variation des données. [31]

6.2 Interprétabilité :

Les classes doivent avoir une interprétation substantive c'est-à-dire qu'il est possible de donner des noms aux classes, dans le meilleur des cas les noms doivent correspondre aux types déduits d'une certaine théorie. [31]

6.3 Stabilité :

Les classes doivent être stable ça veut dire que les petites modifications dans les données et dans les méthodes ne doivent pas changer les résultats. [31]

Chapitre 2 : La tâche de classification

6.4 D'autres critères :

Parfois la taille et le nombre de classes sont employés en tant que critères additionnels : le nombre de classes doit être aussi petit que possible, et la taille des classes ne doit pas être trop petite. [31]

7. Mesures de similarités

Pour comparer l'homogénéité ou le ressemblance, la similarité entre deux objets (points, images, classes, phonème ..), il faut pouvoir mesurer la similarité (ou la dissimilarité) entre eux.

Nous allons décrire maintenant des mesures de similarité pour prouver la similarité entre les objets, «tout système ayant pour but d'analyser ou d'organiser automatiquement un ensemble de données ou de connaissances doit utiliser, sous une forme ou une autre, un opérateur de similarité dont le but est d'établir les ressemblances ou les relations qui existent entre les informations manipulées».

Donc la similarité est une partie importante de la définition d'une méthode de clustering, elle consiste en effet à définir et formaliser une mesure de similarité adaptée aux caractéristiques des données.

Si les composantes des vecteurs de données d'instance sont toutes dans les mêmes unités physiques alors il est possible que la distance euclidienne est suffisante pour réussir à grouper les données similaires. Cependant, même dans ce cas, la distance euclidienne peut parfois être trompeuse. [32]

Vocabulaire :

Il est à noter qu'il y a deux concepts pour exprimer la notion de proximité entre les objets à classifier :

1. Mesure de dissimilarité DM : plus la mesure est faible plus les points sont similaires (distance).

2. Mesure de similarité SM : plus la mesure est grande, plus les points sont similaires.

On parle souvent de « distances » en désignant une mesure de similarité, lorsque ces mesures ont les propriétés de non-négativité, réflexivité, symétrie (la distance entre l'objet A à B est la même que la distance de B à A) et qui respectent l'inégalité triangulaire.

Il existe un grand nombre de mesures de similarité, dans ce qui suit, nous présentons quelques unes des fonctions entre deux objets $d(X_1; X_2)$.

Chapitre 2 : La tâche de classification

7.1 Fonctions de similarité :

1. La distance euclidienne : (aussi appelée la distance à vol d'oiseau) Un rapport de clusters analysis en psychologie de la santé a conclu que la mesure de la distance la plus courante dans les études publiées dans ce domaine de recherche est la distance euclidienne ou la distance au carré euclidienne.

$$d^2(x_1, x_2) = \sum_i (x_{1i} - x_{2i})^2 = (x_1 - x_2)(x_1 - x_2)'$$

1. **La distance de Manhattan** : (appelée aussi taxi-distance)

$$d^2(x_1, x_2) = \sum_i |x_{1i} - x_{2i}|$$

2. **La distance de Mahalanobis** : corrige les données pour les différentes échelles et des corrélations dans les variables, L'angle entre deux vecteurs peut être utilisés comme mesure de distance quand le regroupement des données est de haute dimension. Voir l'espace produit scalaire

$$d^2(x_1, x_2) = (x_1 - x_2) C^{-1} (x_1 - x_2)'$$

(C=covariance)

3. **La distance de Sebestyen:**

$$d^2(x_1, x_2) = (x_1 - x_2) W (x_1 - x_2)'$$

(W=matrice diagonale de pondération)

4. **La distance de Hamming** : mesure le nombre minimum de substitutions nécessaires pour charger un membre dans un autre. Elle permet ainsi , de quantifier la différence entre deux séquences de symboles, généralement utilisée dans le cas des valeurs discrètes (vecteurs)

$$d(a, b) = \sum_{i=0}^{n-1} (a_i \neq b_i)$$

Exemple : Considérons les suites binaires suivantes :

a=(0 0 0 1 1 1 1) et b=(1 1 0 1 0 1 1) alors $d=1+1+0+0+1+0+0$

La distance entre a et b est égale à 3 car 3 bits diffèrent.

$$x^2(\vec{x}, \vec{y}) = \sum_{i=1}^n \frac{(x_i - y_i)^2}{x_i + y_i}$$

Cependant que pour les données de grandes dimensions, il ya une distance spécifique très utilisée :

Chapitre 2 : La tâche de classification

5. **La métrique Minkowski:** Pour les données dimensionnelles, c'est la mesure populaire

$$d_p(x_i, x_j) = \left(\sum_{k=1}^d |x_{ik} - x_{jk}|^p \right)^{\frac{1}{p}}$$

où d est la dimensionnalité des données. La distance euclidienne est un cas particulier où $p = 2$, alors que Manhattan $p = 1$.

Néanmoins, il n'existe pas de directives générales théoriques pour la sélection d'une mesure à une application donnée. Une autre question, est de savoir comment mesurer la distance entre 2 classes $D(C1; C2)$? Pour cela il ya certaines fonctions qui permettent de mesurer cette distance comme : d_p

Plus proche voisin :

$$\min(d(i, j), i \in C_1, j \in C_2)$$

Diamètre maximum :

$$\max(d(i, j), i \in C_1, j \in C_2)$$

Distance moyenne : $\sum_{i,j} \frac{d(i,j)}{n_1 n_2}$

Distance des centres de gravité :

$$d(\mu_1, \mu_2)$$

8. Les différentes méthodes de Classification :

On peut grouper les méthodes classificatoires en deux grandes familles, cette fois-ci, on prend en considération l'intervention ou non d'un « attribut classe » au fur et à mesure du processus de la classification, ces deux types sont : « supervisée (Classement) » et « non supervisée (Classification, Clustering) ».

1. **Non supervisé (classification) :** on ne connaît pas de groupe.
2. **Supervisé (classement) :** groupes fixés, exemples d'objets de chaque groupe.

Cependant, Il existe d'autres types de classification qui s'appuient sur d'autres types de méthodes d'apprentissages comme « l'apprentissage semi-supervisé » et « l'apprentissage par renforcement ».

Chapitre 2 : La tâche de classification

En effet, l'apprentissage semi-supervisé est un bon compromis entre les deux types d'apprentissage « supervisé » et « non-supervisé », car il permet de traiter un grand nombre de données sans avoir besoin de toutes les étiqueter, et il profite des avantages des deux types mentionnés.

Alors que L'apprentissage par renforcement est fort utilisé dans le cas d'apprentissage interactif.

8.1 La classification automatique « Clustering » (non supervisé) :

Le Clustering aussi connu sous le nom de (Segmentation) est un regroupement en classes homogènes consistant à représenter un nuage de points d'un espace quelconque en un ensemble de groupes appelés Cluster.

Un « Cluster » est donc une collection d'objets qui sont « similaires » entre eux et qui sont « dissemblables » par rapport aux objets appartenant à d'autres groupes. On peut voir cette définition clairement graphiquement dans l'exemple suivant :

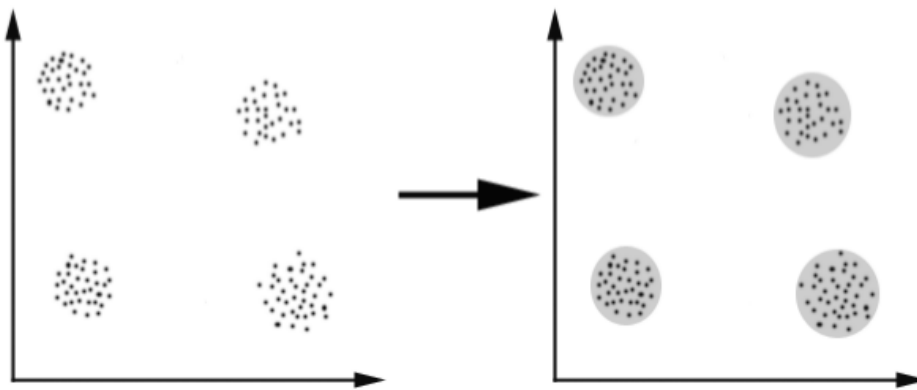


Figure 8: Illustration de regroupement en clusters

Dans ce cas, il est très facile pour une personne d'identifier 4 Clusters dans lesquels les données (nuage des points) peuvent être divisées, le critère de similarité est la distance : deux ou plusieurs objets appartiennent au même cluster s'ils sont « proches », bien sûr cela dépend d'une distance donnée (dans ce cas la distance géométrique).

8.1.1 Principe de fonctionnement de la classification automatique

Chapitre 2 : La tâche de classification

Contrairement à la classification (méthodes supervisées), on ne possède pas des connaissances apriori sur les classes prédéfinies des éléments. Donc La division des objets dans les différents groupes (clusters) se procède en se basant sur le calcul de similarité entre les éléments.

Alors que l'objectif des méthodes du Clustering est de grouper des éléments proches dans un même groupe de manière à ce que deux données d'un même groupe soient le plus similaires possible et que deux éléments de deux groupes différents soient le plus dissemblables possible [33].

Exemple

On utilise souvent ce type de classification en traitement d'images pour fixer les divers objets qu'elles contiennent (segmentation) : routes, villes, rues, des organes humaines (pour les images médicales) ...

Les types de Clustering :

Il existe deux grands types de clustering :

A/ le clustering hiérarchique : d'agglomération (« bottom-up »)

B/ le clustering non-hiérarchique : de division (« top-down »)

Dans le premier cas, on décompose l'ensemble d'individus en une arborescence de groupes.

Dans le 2ème, on décompose l'ensemble d'individus en K groupes, les algorithmes de ce type peuvent aussi être utilisés comme algorithmes de division dans le clustering hiérarchique.

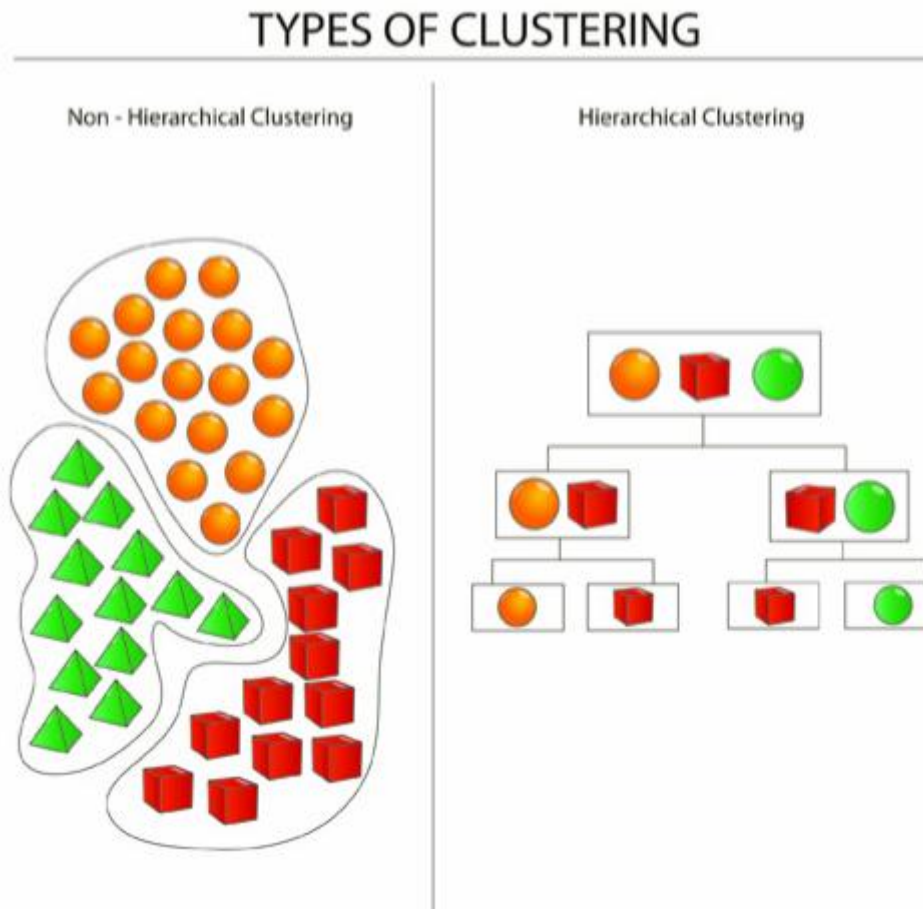


Figure 9: Les deux types de clustering non-hiérarchique/hiérarchique

8.1.1.1 Méthodes non hiérarchiques :

Ce sont des méthodes qui produisent directement une partition en un nombre fixé de classes.

Parmi ces méthodes, nous retrouvons :

1. Méthode de leader

Cette méthode considère chaque objet une seule fois. Lorsque le premier objet arrive, on lui attribue la première classe et il devient le leader de celle-ci.

Ensuite, chaque fois qu'un nouvel objet se présente, on calcule sa distance par rapport aux leaders de chacune des classes existantes à cet instant, et on compare cette distance à un seuil.

Chapitre 2 : La tâche de classification

Si cette distance est inférieure au seuil fixé, on attribue au nouvel objet la classe du premier leader trouvé (pour lequel la distance calculée est inférieure au seuil), sinon une nouvelle classe est créée et le nouvel objet devient le leader de cette classe [34].

L'algorithme est le suivant :

1. Soit E l'ensemble à classer.
2. Soit m une dissimilarité maximale donnée.
3. Soit $k=1$.
4. Soit $i_1=1$.
5. Soit $i_2= 1$.
6. Tant que $i_1 < |E|$
7. Tant que $\text{diss}(e[i_1],e[i_2])<m$.
8. Ajoute $e[i_2]$ à la classe $C[k]$.
9. $i_2 := i_2 + 1$.
10. FinTantQue
11. $i_1:=i_2$.
12. $k:= k +1$.
13. Fin Tant que

Cette méthode dépend de l'ordre de présentation des objets. Lorsque cet ordre n'est pas optimal, le nombre de classes augmente sensiblement.

Par ailleurs, pour définir des nouveaux leaders, cette méthode utilise des distances, ce qui nous ramène au problème de la définition des métriques.

3. Méthode de k-means

L'algorithme k-means mis au point par McQueen en 1967[35], un des plus simples algorithmes d'apprentissage non supervisé, appelée algorithme des centres mobiles [36] ; il attribue chaque point dans un cluster dont le centre (centroïde) est le plus proche.

Le centre est la moyenne de tous les points dans le cluster, ses coordonnées sont la moyenne arithmétique pour chaque dimension séparément de tous les points dans le cluster c'est à dire chaque cluster est représentée par son centre de gravité.

L'idée principale est de définir les k centroïdes arbitraires c_1, c_2, \dots, c_k (k le nombre de clusters fixé a priori, chaque c_i représente le centre d'une classe), Ces centroïdes doivent être placés dans des emplacements différents.

Chapitre 2 : La tâche de classification

Donc, le meilleur choix est de les placer le plus possible éloignés les uns des autres. La prochaine étape est de prendre chaque point appartenant à l'ensemble de données et l'associer au plus proche centroïde.

C'est à dire chaque classe S_i sera représentée par un ensemble d'individus les plus proches de centre c_i . Les nuées dynamiques sont une généralisation de ce principe, où chaque cluster est représenté par un noyau mais plus complexe qu'une moyenne.

Algorithme :

Choisir k moyennes c_1, c_2, \dots, c_k initiales (par exp au hasard)

1. Répéter :

affectation de chaque point à son cluster le plus proche :

$$S_i^{(t)} = \left\{ x_j : \|x_j - m_i^{(t)}\| \leq \|x_j - m_{i^*}^{(t)}\| \text{ for all } i^* = 1, \dots, k \right\}$$

mettre à jour la moyenne de chaque cluster

$$m_i^{(t+1)} = \frac{1}{|S_i^{(t)}|} \sum_{x_j \in S_i^{(t)}} x_j$$

2. Jusqu'à : atteindre la convergence quand il n'y a plus de changement.

Fin.

Cette méthode est convergente et surtout avantageuse du point de vue calcul mais elle dépend essentiellement de la partition initiale.

Il existe donc un risque d'obtenir une partition qui ne soit pas optimale mais seulement meilleure que la partition initiale. De plus, la définition de la classe se fait à partir de son centre, qui pourrait ne pas être un individu de l'ensemble à classer, d'où le risque d'obtenir des classes vides.

3. Méthode des nuées dynamiques

Cette méthode a été proposée par **DID** [37]. Elle peut être considérée comme une généralisation de la méthode des centres mobiles.

Le principe de la méthode est le suivant : on tire au hasard k noyaux parmi une famille de noyaux (chaque noyau contient un sous-ensemble d'individus).

Chapitre 2 : La tâche de classification

Puis chaque point de l'ensemble d'apprentissage est affecté au noyau dont il est plus proche. On obtient ainsi une partition en k classes dont on calcule les noyaux.

On recommence le processus avec les nouveaux noyaux et ainsi de suite jusqu'à ce que la qualité de la partition ne s'améliore plus.

Algorithme :

1. Soit E l'ensemble à classer.
2. Soit f une fonction qui détermine un noyau d'une classe donnée.
3. Soit g une fonction qui détermine une classe autour d'un noyau donné II

Noyau : c'est l'ensemble d'éléments qui agit comme un centre.

4. Soit W une fonction qui mesure l'homogénéité des classes d'une partition donnée et un ensemble de nœuds donné.
5. Soit K le nombre de classes à créer.
6. Choisir K noyaux dans E .
7. Tant que W n'est pas satisfaisant
8. Utilise g pour déterminer une classe autour de chaque noyau.
9. Utilise f pour déterminer les noyaux de ces classes.
10. Fin Tant que

Cette méthode a l'avantage de traiter rapidement de grands ensembles d'individus. Elle fournit une solution dépendant de la configuration initiale et nécessite le choix du nombre de classes.

En général le nombre de classes est fixé par l'utilisateur et l'initialisation est faite par un tirage au hasard. Pour comparer l'individu avec les noyaux, cette méthode utilise des distances, ce qui a l'inconvénient d'établir des métriques.

En conclusion, les méthodes non hiérarchiques permettent de traiter rapidement de grands ensembles d'individus, mais elles supposent que le nombre des classes est fixé au départ.

Si le nombre de classes n'est pas connu ou si ce nombre ne correspond pas à la configuration véritable de l'ensemble d'individus (d'où le risque d'obtenir des partitions de valeurs douteuses), il faut presque toujours tester diverses valeurs de k , ce qui augmente le temps de calcul.

C'est pourquoi, lorsque le nombre des individus n'est pas trop élevé, on préfère utiliser les méthodes hiérarchiques.

Chapitre 2 : La tâche de classification

8.1.1.2 Méthodes hiérarchiques

Le processus basique des méthodes hiérarchiques a été donné par **Johnson [37]** **Lance & Williams [38]**, Ce type de clustering consiste à effectuer une suite de regroupements en Clusters de moins en moins fines en agrégeant à chaque étape les objets (simple élément) ou les groupes d'objets (un Cluster-partition-) les plus proches.

Ce qui nous donne une arborescence de clusters. Cette approche utilise la mesure de similarité pour refléter l'homogénéité ou l'hétérogénéité des classes.

La classification hiérarchique consiste à effectuer une suite de regroupements en classes de moins en moins fines en agrégeant à chaque étape les objets ou les groupes d'objets les plus proches.

Elle fournit ainsi un ensemble de partitions de l'ensemble d'objets[38]. Cette approche utilise la notion de distance, qui permet de refléter l'homogénéité ou l'hétérogénéité des classes. Ainsi, on considère qu'un élément appartient à une classe s'il est plus proche de cette classe que de toutes les autres.

1.Principe d'une méthode hiérarchique de classification

Son principe est simple, initialement chaque individu forme une classe, soit n classes, donc on cherche à réduire ce nombre de classe $n_{\text{newnbrclss}} < n$ itérativement de sorte que dans chaque étape on fusionne deux classes ensemble (Les deux classes choisies pour être fusionnées sont celles qui sont les plus "proches" en fonction de leur dissimilarité) ou ajouter un nouveau élément à une classe (un élément appartient à une classe s'il est plus proche de cette classe que de toutes les autres).

La valeur de dissimilarité est appelée indice d'agrégation qui commence dans la première itération faible, et croît d'itération en itération.

Parmi les algorithmes les plus connus de ce type :La classification ascendante hiérarchique(CHA) où le mot ascendante est utilisé pour désigner qu'elle part d'une situation dont tous les individus représentent des clusters à part entière ,puis on cherche à les rassembler en classes de plus en plus grandes.

Ainsi Le qualificatif "hiérarchique" désigne le fait qu'elle produit une hiérarchie, (une amélioration a été proposée en 2002 par **P. Bertrand**, appelée Classification Ascendante 2-3 Hiérarchique).

La figure 10 est une illustration du principe des méthodes hiérarchiques. Dans cette figure, on représente la suite des partitions d'un ensemble $\{a, b, c, d, e\}$:

Chapitre 2 : La tâche de classification

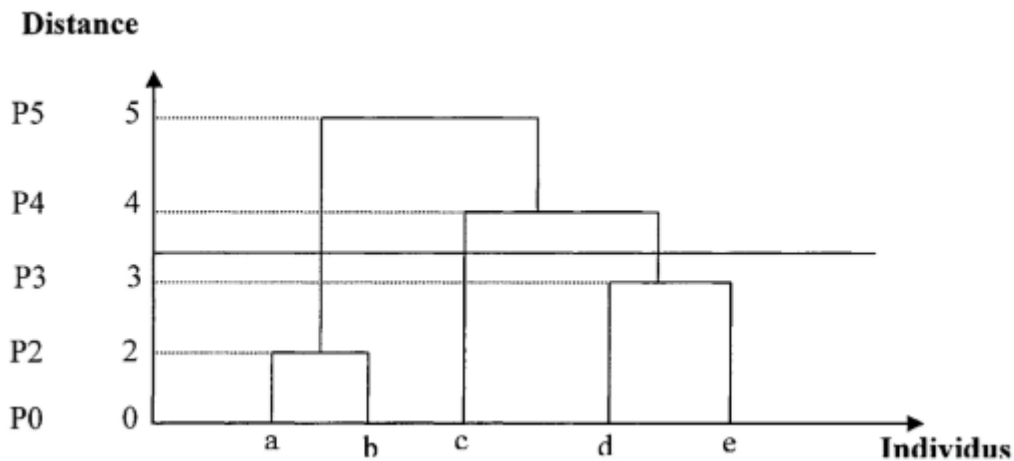


Figure 10: La partition hiérarchique.

Les différentes partitions représentées dans la figure 10 sont:

$P_0 = \{\{a\}, \{b\}, \{c\}, \{d\}, \{e\}\}$ correspond à la distance $d = 0$;

$P_1 = \{\{a, b\}, \{c\}, \{d\}, \{e\}\}$ correspond à la distance $d = 2$;

$P_2 = \{\{a, b\}, \{c\}, \{d, e\}\}$ correspond à la distance $d = 3$;

$P_3 = \{\{a, b\}, \{c, d, e\}\}$ correspond à la distance $d = 4$;

$P_4 = \{\{a, b, c, d, e\}\}$ correspond à la distance $d = 5$.

À chaque partition correspond une valeur numérique représentant le niveau auquel ont lieu les regroupements.

Les partitions sont définies en coupant l'arbre à un certain niveau en regardant les branches qui tombent. Dans l'exemple de la figure 11.1, si on coupe l'arbre à une valeur 3.5 on aura la partition suivante: $P_2 = \{\{a, b\}, \{c\}, \{d, e\}\}$.

La principale difficulté présentée par cette méthode est la définition du critère de regroupement de deux classes, c'est-à-dire la détermination d'une distance entre les classes.

Il en existe un bon nombre d'algorithmes pour les méthodes hiérarchiques, on se limite à présenter l'algorithme de la méthode directe, elle est fondée sur un critère d'agrégation, dont la dissimilarité de classes dépend uniquement sur la dissimilarité entre objets.

L'algorithme est le suivant:

1. Soit E l'ensemble à classifier.
2. Calcule et écrit le tableau T de la dissimilarité $diss$.
3. Soit H l'ensemble des singletons.
4. Tant que T a plus d'une colonne.

Chapitre 2 : La tâche de classification

5. Détermine la valeur minimale m de T , soit la position $C1, C2$.
 6. Ajoute $(C1 \vee C2)$ à H avec $v(C1 \vee C2) = m$.
 7. Agrège les colonnes et lignes de $C1$ et $C2$ en mettant pour toute classe C la valeur $D(C, C1 \vee C2)$
// Cela revient simplement au maximum ou au minimum des valeurs anciennes, selon le choix de D .
 8. Soit T ce nouveau tableau.
 9. Fin Tant que
 10. Ajoute E à H avec $v(E) = \text{valeur de } T$.
- II. H est l'hierarchie.

2. Discussion

la CAH ne nécessite pas de connaître le nombre de clusters a priori. De plus, il n'y a pas de fonction d'initialisation, ainsi une seule construction d'un cluster (équivalent à une itération pour les méthodes de partitionnement).

En ce qui concerne généralement les méthodes hiérarchiques le problème qu'on peut rencontrer réside dans la sélection d'une ultra-métrique (distance pour calculer la similarité entre clusters) soit la plus proche de la métrique utilisée pour les individus, car ces méthodes sont heuristiques, pour cela il y a plusieurs techniques qui permettent de le faire : Saut minimal (single linkage); Saut maximal (complete linkage); Saut moyen; Barycentre... une autre faiblesse est : la complexité de temps d'au moins $O(n^2)$, où n est le nombre d'objets au total, ainsi qu'on pourrait jamais défaire ce qui a été fait précédemment.

Il est difficile parfois d'apporter une justification aux méthodes hiérarchique (CAH, CDH..), Cependant, dans, une interprétation probabiliste de la CAH, basée sur une estimation par maximum de vraisemblance des modèles de mélange, est proposée comme solution pour mieux interpréter les résultats. [39]

Un autre inconvénient de ce type de méthode est que si une action est effectuée (fusion ou décomposition), elle ne peut être annulée. Cela permet de réduire le champ d'exploration, mais une telle astuce ne peut corriger une décision erronée.

Afin d'améliorer la qualité d'une classification hiérarchique, on peut profiter de deux techniques:

- analyser attentivement les liens entre objets à chaque étape améliorer la partition obtenue avec une méthode de deuxième type de clustering (partitionnement).

Chapitre 2 : La tâche de classification

8.2 METHODE SUPERVISEE « Classement » ou « DISCRIMINATION »

Le « classement » est une méthode supervisée qui consiste à définir une fonction qui attribue une ou plusieurs classes à chaque donnée.

Dans cette approche on suppose qu'un expert fournit auparavant les étiquettes pour chaque donnée, les étiquettes sont des classes d'appartenance.

La classification supervisée (appelée aussi classement ou classification inductive) a pour objectif « d'apprendre » par l'exemple.

Elle cherche à expliquer et à prédire l'appartenance de documents à des classes connues a priori. Ainsi c'est l'ensemble des techniques qui visent à deviner l'appartenance d'un individu à une classe en s'aidant uniquement des valeurs qu'il prend. [40]

8.2.1 Principe de la méthode de classification supervisée

La conception supervisée d'un classifieur à C classes (ensemble fini de classes c_i) est le fait de classifier supposés avoir été préalablement « étiquetés » par un « superviseur » en C ensembles qui forment un ensemble d'apprentissage.

Le superviseur n'est qu'un Classifieur en lequel on a confiance (expert humain , caractère répétitif , le système visuel humain ...) , donc notre système de classification supervisée va être conçu en basant sur les exemples du superviseur (l'ensemble d'apprentissage où pour tout exemple on connaît à priori sa classe.)

c'est-à-dire, on cherche à prédire si un objet (élément) « x_i » de la base de données, décrit par un ensemble de descripteurs « d », appartient ou non à une classe « c_j » parmi N Classes, pour le faire, on a un ensemble d'apprentissage décrit par :

$$A=(x_1,c_2), (x_2,c_4), (x_3,c_2)...(x_i,c_j) / x_i \in \mathbb{R}^d, c_j \in C$$

Donc pour chaque objet x_i de l'ensemble des données, on peut connaître sa classe a priori c_j . La classification supervisée tente de chercher, à partir des données de A,

Une fonction de décision Γ qui va associer à tout nouveau élément x_i de test une classe C_j , puis on compare ce que nous a donné cette fonction avec la classe connu a priori de cet élément, de sorte à minimiser les mauvais classements ($\Gamma(x_i) \neq c_j$). Donc l'objectif est de chercher à prédire la classe de toute nouvelle donnée.

8.2.2 Les k plus proches voisins (K-PPV)

La méthode des plus proches voisins (noté parfois k-PPV ou k-NN pour -Nearest-Neighbor) consiste à déterminer pour chaque nouvel individu que l'on veut classer, la liste des plus proches voisins parmi les individus déjà classés.

Chapitre 2 : La tâche de classification

L'individu est affecté à la classe qui contient le plus d'individus parmi ces plus proches voisins. Cette méthode nécessite de choisir une distance, la plus classique est la distance euclidienne, et le nombre de voisins à prendre en compte.

Cette méthode supervisée et non-paramétrique est souvent performante. De plus, son apprentissage est assez simple, car il est de type apprentissage par cœur (on garde tous les exemples d'apprentissage).

Cependant, le temps de prédiction est très long, car il nécessite le calcul de la distance avec tous les exemples, mais il existe des heuristiques pour réduire le nombre d'exemples à prendre en compte [40].

Algorithme :

1. initialisation, choix de :

- Nombre de classes, Valeur de k , exemples initiaux, mesure de similarité.

2. pour chaque vecteur d'objet à classer :

- mesurer la distance du vecteur avec tous les autres déjà classés
- déterminer la liste des k vecteurs les plus proches de lui (k -ppv)
- déterminer la classe la plus représentée dans la liste des k -ppv et affecter notre vecteur à cette classe.

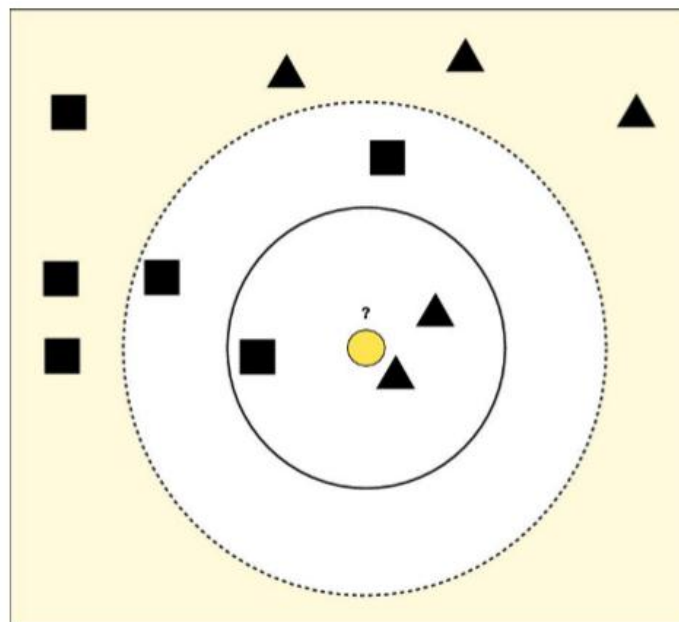


Figure 11: Illustration des K-PPV d'un point

Chapitre 2 : La tâche de classification

Ce qu'on peut remarquer dans cette méthode, c'est le coût de calculs qu'elle impose au fur et à mesure de ce processus de classification, car ce coût augmente avec chaque vecteur qu'on vient de classer, plus on ajoute de nouveaux vecteurs déjà classés, plus ce coût augmente, ce qui explique le temps d'exécution qu'elle prend pour classer.

En plus de la sensibilité de cet algorithme à l'initialisation des paramètres d'entrées (le choix de k , la distance utilisée..) alors il faut que lors de la sélection des paramètres d'entrées ces derniers respectent certaines contraintes (comme que k ne soit pas un multiple du nombre de classes pour éviter une surreprésentation d'une classe par rapport à une autre).

Malgré ces points, k -ppv reste une des méthodes les plus utilisées grâce à sa simplicité et robustesse et son caractère de généralisation à partir d'un nombre éminent de données d'apprentissage.

8.2.3 La classification bayésienne

Un classifieur probabiliste linéaire simple basée sur le théorème de Bayes suppose que les descripteurs (attributs) qui décrivent les objets de l'ensemble d'apprentissage sont indépendants.

8.2.3.1 Principe de la classification bayésienne

L'ensemble d'apprentissage «A» est connu et chaque objet est étiqueté par sa classe « C_k », l'objectif est de chercher à classer un nouvel objet « X_{new} » non encore étiqueté. Le Classifieur bayésien va choisir la classe « C_k » qui a la plus grande probabilité, on parle de règle MAP (maximum a posteriori)[41]

$$C_{MAP} = \underset{C_k \in C}{\operatorname{argmax}} P(C_k | X_{new}) = \underset{C_k \in C}{\operatorname{argmax}} \frac{P(X_{new} | C_k) P(C_k)}{P(X_{new})} = \underset{C_k \in C}{\operatorname{argmax}} P(X_{new} | C_k) P(C_k)$$

donc Il nous faut estimer les probabilités $P(C_k)$ et $P(X_{new} / C_k)$ à partir des données d'apprentissage. Les probabilités a priori des classes $P(C_k)$, peuvent être estimées facilement par :

$$P(C_k) = \frac{\text{nombre d'expression d'apprentissage dans la classe } C_k}{\text{le nombre totale de documents dans l'ensemble d'apprentissage}}$$

Maintenant pour estimer les valeurs de $P(X_{new} / C_k)$, puisque les descripteur(attributs) de « X_{new} » sont indépendants, alors on aura grâce aux théories d'indépendance bayésienne entre les variables[09]

Chapitre 2 : La tâche de classification

$$P(X_{\text{new}}|C_k)=P(f_1|C_k) P(f_2|C_k)\dots P(f_n|C_k)$$

Où les « fi » sont les attributs qui décrivent l'ensemble de données, sachant que :

$$p(C, f_1, \dots, f_n) = p(C) \prod_{i=1}^n p(f_i|C)$$

Et pour estimer les paramètres d'une loi de probabilité relative à une caractéristique précise, il est nécessaire de présupposer le type de la loi en question.

Le Classifieur naïf de Bayes est très performant même avec peu de données, car il fait souvent de bonnes hypothèses sur la distribution des données avec un peu de données d'entraînement afin d'estimer les paramètres nécessaires à la classification (moyennes et variances) [10], mais

Lorsque le nombre de descripteurs est grand, il est parfois impossible de construire ce modèle sur des tableaux de probabilités.

8.2.4 L'approche des réseaux de neurones

Les réseaux de neurones sont à l'origine d'une tentative de modélisation mathématique du cerveau humain. Le principe général consiste à définir des unités simples appelées neurones, chacune étant capable de réaliser quelques calculs élémentaires sur des données numériques.

On relie ensuite un nombre important de ces unités formant ainsi un outil de calcul puissant.

Un réseau de neurones est un système composé de plusieurs unités de calcul simples (nœuds) fonctionnant en parallèle, dont la fonction est déterminée par la structure du réseau et l'opération effectuée par les nœuds.

Le principe de fonctionnement est le suivant : On dispose initialement d'une base de connaissances constituée de couples de données (entrées / sorties) et on souhaite utiliser cette base de données pour entraîner un algorithme à reproduire les associations constatées entre les entrées et les sorties de l'échantillon.

L'exemple le plus simple de réseau de neurones est souvent donné par le “perceptron multicouches” qui est un cas particulier de réseau de neurones (Figure 12).

Chapitre 2 : La tâche de classification

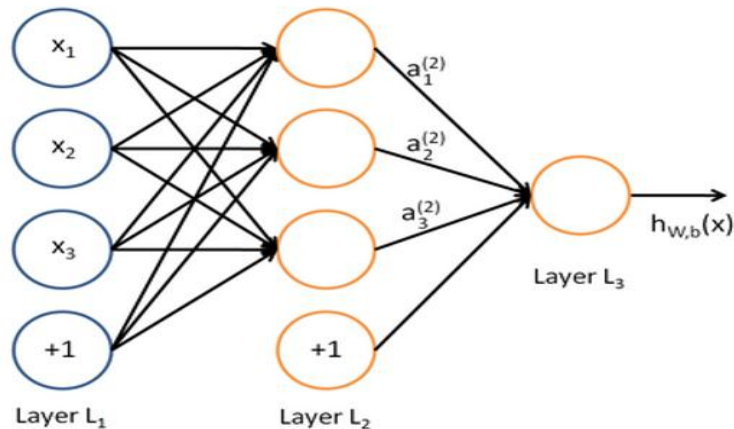


Figure 12: Perceptron à trois couches (schéma type)

Pour un réseau de neurones avec N nœuds d'entrée, notés $C(1), \dots, C(N)$, et N poids affectés aux liaisons notés $w(1), \dots, w(N)$ l'entrée d'un nœud de la couche suivante sera généralement une somme pondérée des valeurs de sortie des neurones précédents :

$$X = w(1)*C(1) + w(2)*C(2) + w(3)*C(3) + \dots + w(N)*C(N)$$

Les poids sont des paramètres adaptatifs, dont la valeur est à déterminer en fonction du problème via un algorithme d'apprentissage (propagation, rétro-propagation...) [41].

Les réseaux de neurones peuvent être utilisés pour effectuer une classification supervisée floue de la manière suivante : chaque nœud d'entrée correspond à un attribut de l'objet (autant de nœuds d'entrée que d'attributs).

On peut prendre un neurone de sortie par classe ; la valeur de sortie est la valeur de la fonction d'appartenance (probabilité que l'objet appartienne à cette classe) [42].

8.2.4.1 Caractéristiques des différentes méthodes

Quel que soit le type de la classification il ya Trois éléments qui permettent de caractériser les différentes méthodes :

1. La classification se déroule séquentiellement en regroupant les observations les plus 'semblables' (méthodes hiérarchiques) ou elle regroupe en k groupes toutes les observations simultanément (méthodes non-hiérarchiques).
2. Le critère de 'ressemblance' entre deux observations.

Chapitre 2 : La tâche de classification

3. Le critère de 'ressemblance' entre deux groupes ou entre une observation et un groupe.

Ces trois éléments permettent de définir le déroulement ainsi que le type de la méthode, le deuxième et le troisième caractère ont un point primordial dans la performance et la qualité du résultat attendu d'une méthode, car il y aura certainement une différence de calcul (précision) entre le fait d'utiliser la distance euclidien au lieu de la distance de Hamming (ie que la distance utilisée est prise en considération afin d'améliorer les résultats).[43][44]

9. Conclusion

Plusieurs méthodes sont proposées pour le problème général de la classification. Ils diffèrent par les mesures de proximités qu'ils utilisent, la nature des données qu'ils traitent et l'objectif final de la classification, chacune de ces méthodes possède ses points forts et ses points faibles.

Chapitre 3 : La classification d'images et les réseaux de neurones convolutifs

Introduction

Dans ce chapitre nous allons voir qu'est-ce que la classification d'images, qu'est-ce que les réseaux de neurones convolutifs et comment se fait la classification d'images grâce à ces réseaux.

Mais avant de voir tout cela, nous allons d'abord citer quelques notions d'infographie

1. Notions d'infographie

1.1 Définition d'une image

Une image est une représentation visuelle, voire mentale, de quelque chose (objet, être vivant et/ou concept) elle est composée de points appelés : pixels.

Un pixel (souvent abrégé px) est l'unité de base permettant de mesurer la définition d'une image. Son nom provient de la locution anglaise *picture element*, qui signifie « élément d'image ».

1.2 Caractéristiques d'une image

- a. **Résolution** : c'est le nombre de pixels contenus dans l'image par unité de longueur. Elle s'exprime le plus souvent en **ppp** (point par pouces) ou en **dpi** (dots per inch) en anglais parfois en **point par cm**.
- b. **Définition** : On appelle Définition d'une image, le nombre de pixels qui la compose.

Exemple :

Pour une image de 640 colonnes sur 240 ligne, l'image est composée de :

$$640 \times 240 = 153\ 600 \text{ pixels}$$

- c. **Taille** : On appelle taille d'une image, le produit de sa définition par le nombre d'octet par pixel.

Exemple :

Une image RGB de 640 colonnes sur 240lignes, la taille de l'image est de :

$$640 \times 240 \times 3 = 460\ 800 \text{ octets}$$

La qualité d'une image est meilleure si celle-ci est représentée avec plus de pixels par cm ou par pouce, comme on peut le remarquer dans les exemples ci-dessous.

d. Couleur

Une image est donc représentée par un tableau à deux dimensions dont chaque case est un pixel. Pour représenter informatiquement une image, il suffit donc de créer un tableau de pixels dont chaque case contient une valeur.

La valeur stockée dans une case est codée sur un certain nombre de bits déterminant la couleur ou l'intensité du pixel, on l'appelle profondeur de codage (parfois profondeur de couleur).

De nombreux systèmes de codage ayant leurs particularités propres existent, comme par exemple :

- RGB (ou RVB) : basé sur un mélange additif (combinaison de rayons lumineux) de trois couleurs primaires (Rouge, Vert, Bleu).
- CMYK ou (CMJN) : basé sur un mélange soustractif (combinaison de pigments colorés) de trois couleurs primaires (Cyan, Magenta, Jaune) et du noir.
- HLS (ou TLS): basé sur la perception physiologique de la couleur par l'œil humain (Teinte, Luminance, Saturation).

Le codage le plus répandu en traitement numérique des images est le codage RGB car il correspond directement aux systèmes d'affichage typiquement utilisé en informatique graphique (affichage sur un écran), et est plus simple à manipuler au niveau des opérations de combinaison de couleur.

1.3 La numérisation

C'est le processus qui permet de passer de l'état d'image physique (image optique par exemple) qui est caractérisée par l'aspect continu du signal qu'elle représente (une infinité de valeur dans l'intensité lumineuse par exemple).

C'est cette forme numérique qui permet une exploitation ultérieure par des outils logiciels sur ordinateur.

2 Classification d'images

Un analyste qui tente de classer les caractéristiques d'une image, utilise les éléments de l'interprétation visuelle pour identifier des groupes homogènes de pixels qui représentent des classes intéressantes de surfaces.

La classification numérique des images utilise l'information spectrale contenue dans les valeurs d'une ou de plusieurs bandes spectrales pour classer chaque pixel individuellement. Ce type de classification est appelé reconnaissance de regroupements spectraux.

Les deux façons de procéder (manuelle ou automatique) ont pour but d'assigner une classe particulière ou thème à chacun des pixels d'une image.

La "nouvelle" image qui représente la classification est composée d'une mosaïque de pixels qui appartiennent chacune à un thème particulier. Cette image est essentiellement une représentation thématique de l'image originale.

2.1 Classe d'information et classe spectrale

Lorsqu'on parle de classes, il faut faire la distinction entre des classes d'information et des classes spectrales. Les classes d'information sont des catégories d'intérêt que l'analyste tente d'identifier dans les images.

Les classes spectrales sont des groupes de pixels qui ont les mêmes caractéristiques (ou presque) en ce qui a trait à leur valeur d'intensité dans les différentes bandes spectrales des données.

L'objectif ultime de la classification est de faire la correspondance entre les classes spectrales et les classes d'information. Il est rare qu'une correspondance directe soit possible entre ces deux types de classes.

Des classes spectrales bien définies peuvent apparaître parfois sans qu'elles correspondent nécessairement à des classes d'information intéressantes pour l'analyse.

D'un autre côté, une classe d'information très large (par exemple la forêt) peut contenir plusieurs sous-classes spectrales avec des variations spectrales définies.

En utilisant l'exemple de la forêt, les sous-classes spectrales peuvent être causées par des variations dans l'âge, l'espèce, la densité des arbres ou simplement par les effets d'ombrage ou des variations dans l'illumination.

L'analyste a le rôle de déterminer de l'utilité des différentes classes spectrales et de valider leur correspondance à des classes d'informations utiles.

2.2 Méthodes supervisées/non supervisées

Les méthodes de classification les plus communes peuvent être séparées en deux grandes catégories : les méthodes de **classification supervisée** et les méthodes de **classification non supervisée**.

a. La méthode non supervisée

Lors de l'utilisation d'une méthode non supervisée. Les classes spectrales sont formées en premier, basées sur l'information numérique des données seulement. Ces classes sont ensuite associées, par un analyste, à des classes d'information utile (si possible).

Des programmes appelés algorithmes de classification sont utilisés pour déterminer les groupes statistiques naturels ou les structures des données. Habituellement, l'analyste spécifie le nombre de groupes ou classes qui seront formés avec les données.

De plus, l'analyste peut spécifier certains paramètres relatifs à la distance entre les classes et la variance à l'intérieur même d'une classe. Le résultat final de ce processus de classification itératif peut créer des classes que l'analyste voudra combiner, ou des classes qui devraient être séparées de nouveau.

Chacune de ces étapes nécessite une nouvelle application de l'algorithme. L'intervention humaine n'est donc pas totalement exempte de la classification non supervisée. Cependant, cette méthode ne commence pas avec un ensemble prédéterminé de classes comme pour la classification supervisée.

b. La méthode supervisée

La classification supervisée procède de la façon contraire, l'analyste identifie des échantillons assez homogènes de l'image qui sont représentatifs de différents types de surfaces (classes d'information).

Ces échantillons forment un ensemble de données-tests. Les informations numériques pour chacune des bandes et pour chaque pixel de ces ensembles sont utilisées pour que l'ordinateur puisse définir les classes et ensuite reconnaître des régions aux propriétés similaires à chaque classe.

Un algorithme est procédé afin de déterminer la "signature" numérique de chacune des classes. Plusieurs algorithmes différents sont possibles. Une fois que l'ordinateur a établi la signature spectrale de chaque classe à la classe avec laquelle il a le plus d'affinités.

Une classification supervisée commence donc par l'identification des classes d'information qui sont ensuite utilisées pour définir les classes spectrales qui les représentent.

3 Les réseaux de neurones convolutifs

Avant de voir qu'est-ce qu'un réseau de neurones convolutionnelles, nous allons d'abord commencer par définir un perceptron simple

3.1 Perceptron simple

Le perceptron est un type de réseau de neurones considéré comme le type de réseaux neuronal le plus simple, il se compose de deux couches de neurones : La première couche est composée de cellules d'entrée, la deuxième couche fournit la réponse comme il dispose d'un unique neurone de sortie.

Il est utilisé pour la classification des données en deux catégories, il est limité aux seuls problèmes linéairement séparables. L'impossibilité de traiter les problèmes non linéaires avec les réseaux de type Perceptron a été résolue par un réseau multicouche.

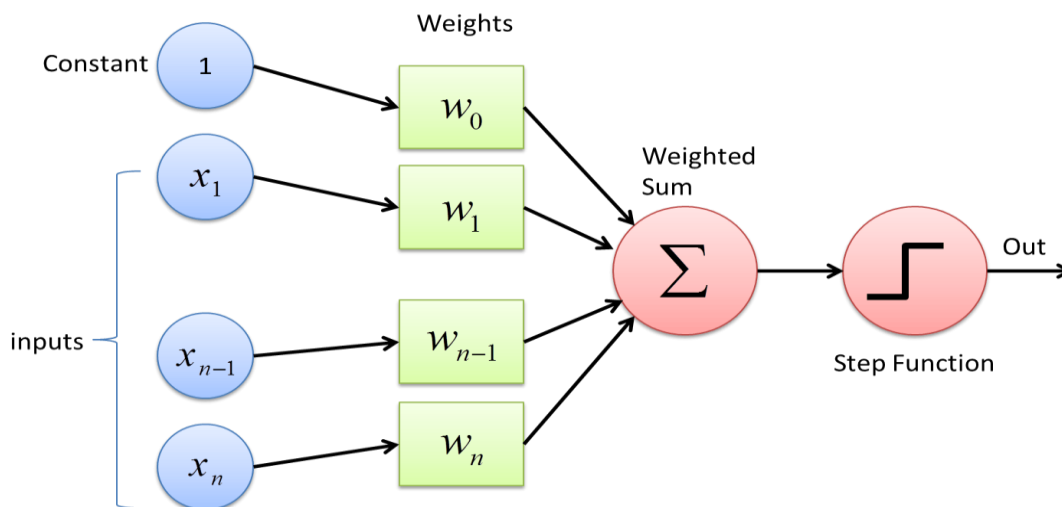


Figure 13: Perceptron simple

3.2 Définition des réseaux de neurones convolutifs

Les réseaux de neurones convolutifs (CNN pour Convolutional Neural Networks) sont des réseaux de neurones multicouches qui sont spécialisés dans des tâches de reconnaissance de forme.

Ils sont réputés pour leur faible taux de prétraitement nécessaires à leur fonctionnement et leur architecture repose sur plusieurs réseaux de neurones convolutionnels alternant entre couches de convolution et couches d'agrégation (pooling).

Le début de l'architecture consistant en une succession de couches de convolution et d'agrégation est dédiée à l'extraction automatique de caractéristiques, tandis que la seconde partie, composée de couches de neurones complètement connectés, est dédiée à la classification.

3.3 Contribution des RNC dans la classification d'images

Les réseaux neuronaux convolutionnels (CNN) sont l'architecture de modèle à la fine pointe de la technologie pour les tâches de classification d'image.

Les CNN appliquent une série de filtres aux données de pixels brutes d'une image pour extraire et apprendre des fonctionnalités de niveau supérieur, que le modèle peut alors utiliser pour la classification.

3.4 Architecture d'un RNC

Un réseau de neurones convolutif contient trois composantes :

3.4.1 Couches convolutionnelles

Pour reconnaître des objets sur une image, la première idée qui pourrait venir à l'esprit serait de comparer les images pixel par pixel, pour calculer un pourcentage de correspondance.

Cette méthode ne fonctionne pas, car un objet peut avoir plusieurs formes différentes, et qu'il peut avoir des orientations différentes. C'est pour cela qu'il est utilisé des couches de convolutions.

Une couche de convolution permet de reconnaître des formes de bas niveau sur une matrice, généralement avec des filtres de taille 3×3 pixels.

Tous les filtres sont appliqués sur la matrice d'entrée en "glissant chacun des filtres" sur cette matrice de droite à gauche et de haut en bas.

Plus précisément, pour chaque filtre, et pour tout carré de la taille du filtre sur la matrice d'entrée, les cases correspondantes de la matrice de sorties sont égales à la somme de toutes les valeurs du filtre, multipliées respectivement par ceux des carrés d'entrée.

Ainsi la taille de la matrice de sortie est égale à celle de l'entrée moins deux lignes et deux colonnes.

3.4.2 Assemblage des couches (pooling)

Ces couches abaissent les données d'image extraites par les couches convolutionnelles pour réduire la dimensionnalité de la carte des caractéristiques afin de diminuer le temps de traitement.

Une couche pooling permet de réduire la taille de la matrice d'entrée, d'augmenter la vitesse tout en gardant les informations les plus importantes. Il existe deux types de pooling :

- **Le Maximum Pooling** divise la taille de l'entrée par 2 en prenant le maximum des carrés de taille 2 x 2 pixels.
- **L'Average Pooling** divise la taille de l'entrée par 2 en calculant la moyenne des carrés de taille 2 x 2 pixels.

Les couches de **Maximum Pooling** sont essentielles après des couches de convolutions, afin de pouvoir reconnaître des objets plus grands, et donc plus complexes. Souvent le maximum pooling est le plus utilisé.

Un algorithme de mise en commun couramment utilisé est le regroupement maximal, qui extrait les sous-régions de la carte des fonctionnalités (par exemple, les carreaux de 2x2 pixels), conserve leur valeur maximale et supprime toutes les autres valeurs..

3.4.3 Couches denses (entièrement connectées)

Multiplie une matrice d'entrée par une autre, et lui ajoute un biais (ces deux matrices sont calculées pendant la phase d'apprentissage). Cette couche permet de redimensionner la taille de la sortie.

Ces couches sont généralement utilisées à la fin d'un réseau pour que la sortie soit un vecteur de probabilité que l'image représente un objet.

Par exemple, ce réseau pourrait retourner 4 probabilités correspondant chacune à ce que l'image passée en entrée soit un chien, un chat, un visage ou une voiture.

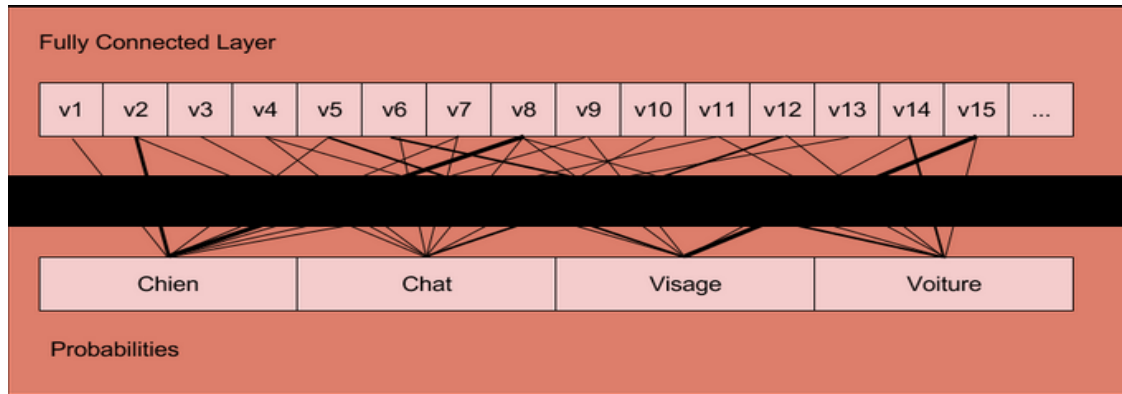


Figure 14: Exemple d'un réseau de neurones convolutionnelles

Typiquement, une CNN est composée d'une pile de modules convolutifs qui effectuent une extraction de fonctionnalités.

Chaque module se compose d'une couche convolutionnelle suivie d'une couche de mise en commun.

Le dernier module convolutif est suivi d'une ou plusieurs couches denses qui effectuent une classification.

La couche dense finale dans un CNN contient un seul nœud pour chaque classe cible dans le modèle (toutes les classes possibles que le modèle peut prédire), avec une fonction d'activation de softmax pour générer une valeur entre 0-1 pour chaque nœud (la somme de tous ces valeurs de softmax sont égales à 1).

Donc en résumé, cette couche entièrement effectue une classification sur les caractéristiques extraites par les couches convolutionnelles et diminuées par les couches de mise en commun.

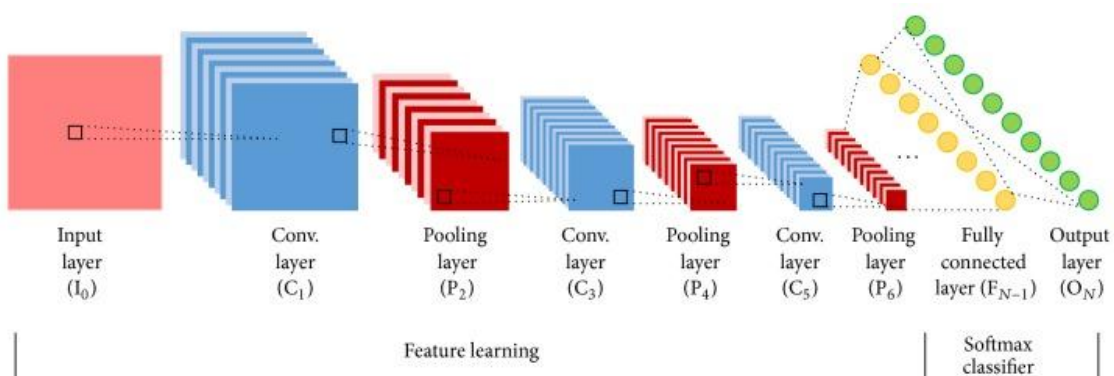


Figure 15: Architecture d'un réseau de neurone convolutionnel

Typiquement, un CNN est composée d'une pile de modules convolutifs qui effectuent une extraction de fonctionnalités. Chaque module se compose d'une couche convolutionnelle suivie d'une couche de mise en commun.

Le dernier module convolutif est suivi d'une ou plusieurs couches denses qui effectuent une classification.

La couche dense finale dans un CNN contient un seul nœud pour chaque classe cible dans le modèle (toutes les classes possibles que le modèle peut prédire), avec une fonction d'activation de softmax pour générer une valeur entre 0-1 pour chaque nœud (la somme de tous ces valeurs de softmax sont égales à 1).

3.5 Fonctionnement d'un CNN

La tâche principale de la couche convolutionnelle est de détecter les conjonctions locales des fonctionnalités de la couche précédente et de mapper leur apparence à une carte des fonctionnalités.

À la suite de la convolution dans les réseaux neuronaux, l'image est divisée en perceptrons, créant des champs réceptifs locaux et finalement comprimant les perceptrons dans les cartes caractéristiques de taille $m_2 \times m_3$.

Ainsi, cette carte stocke l'information où la fonctionnalité se produit dans l'image et la façon dont elle correspond au filtre. Par conséquent, chaque filtre est formé à l'espace par rapport à la position dans le volume auquel il est appliqué.

Dans chaque couche, il y a une ensemble de m_1 des filtres. Le nombre de combien de filtres sont appliqués dans une seule étape équivaut à la profondeur du volume des cartes des caractéristiques de sortie.

Chaque filtre détecte une caractéristique particulière à chaque emplacement de l'entrée. La sortie $y_i^{(l)}$ de la couche l se compose de $m_1^{(l)}$ caractéristiques de la taille $m_2^{(l)} \times m_3^{(l)}$.

La i ème carte de caractéristique, notée $y_i^{(l)}$, est calculé comme :

$$Y_i^{(l)} = B_i^{(l)} + \sum_{j=1}^{m_1^{(l-1)}} K_{i,j}^{(l)} * Y_j^{(l-1)}$$

où $B_i^{(l)}$ est une matrice de polarisation et $k_{i,j}^l$, j est le filtre de taille $2h_1^{(l)} + 1 \times 2h_2^{(l)} + 1$ reliant la j^{th} feature map dans la couche (l-1) avec i^{th} feature map.

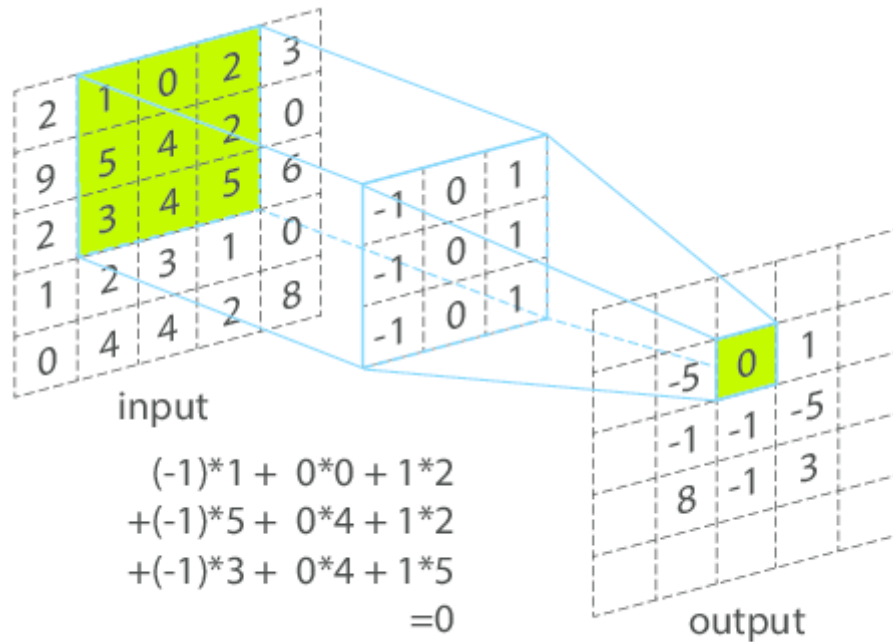


Figure 16: Convolution avec un filtre 3*3

3.5.1 Couche non linéaire

Une couche non linéaire dans un réseau neuronal convolutif consiste en une fonction d'activation qui prend la carte des caractéristiques générée par la couche convolutionnelle et crée la carte d'activation comme sortie.

La fonction d'activation est une opération élémentaire sur le volume d'entrée et donc les dimensions de l'entrée et de la sortie sont identiques.

$$Y_i^{(l)} = f(Y_i^{l(l-1)})$$

Semblable aux perceptrons multicouches, la fonction d'activation est généralement implémentée en tant que fonctions logistiques.

Cependant, des recherches plus récentes suggèrent que les unités linéaires rectifiées (ReLU) sont avantageuses par rapport aux fonctions d'activation traditionnelles, en particulier dans les réseaux neuronaux convolutionnels.

3.5.2 Couche de rectification

Une couche de rectification dans un réseau neuronal convolutif effectue une opération de valeur absolue sur le volume d'entrée (généralement le volume d'activation).

Laisse couche l être une couche de rectification, il prend le volume d'activation $Y^{(l-1)}$ à partir d'une couche non linéaire $(l-1)$ et génère le volume d'activation rectifié $Y_i^{(l)}$

$$Y_i^{(l)} = |Y_i^{(l-1)}|$$

Indépendamment de la simplicité générale de l'opération, elle joue un rôle clé dans la performance du réseau neuronal convolutif lors de l'utilisation d'une méthode de mise en commun, les valeurs négatives dans le volume d'activation sont susceptibles d'annuler les activations positives, ce qui réduit considérablement la précision du réseau.

Par conséquent, la rectification est appelée « élément crucial ».

3.5.3 Unités linéaires rectifiées (ReLU)

Les unités linéaires rectifiées (ReLU) sont une implémentation spéciale qui combine des couches de non-linéarité et de rectification dans les réseaux neuronaux convolutionnels.

Une unité linéaire rectifiée (c'est-à-dire un seuil à zéro) est une fonction linéaire par morceaux définie comme suit :

$$Y_i^{(l)} = \max (Y_i^{(l-1)}, 0)$$

Elles comprennent que des opérations simples en termes de calcul (principalement des comparaisons) et donc beaucoup plus efficaces à mettre en œuvre dans les réseaux neuronaux convolutionnels.

En raison de ses avantages et de ses performances, la plupart des architectures récentes des réseaux neuronaux convolutionnels n'utilisent que des couches unitaires linéaires rectifiées (ou leurs dérivées telles que les ReLU bruyantes ou étanches) en raison de ses avantages.

Pooling layer

La couche de mise en commun ou de descendante est responsable de réduire la taille spatiale des cartes d'activation.

En général, ils sont utilisés après plusieurs étapes d'autres couches (c'est-à-dire des couches de convolution et de non-linéarité) afin de réduire progressivement les exigences de calcul à travers le réseau ainsi que de minimiser la probabilité d'un dépassement.

Elle comporte deux hyperparamètres, l'étendue spatiale du filtre $F(l)$ et la foulure $S(l)$. Il faut un volume d'entrée de taille $m_1^{(l-1)} \times m_2^{(l-1)} \times m_3^{(l-1)}$ et fournit un volume de sortie de taille $m_1^{(l)} \times m_2^{(l)} \times m_3^{(l)}$ avec :

$$\begin{aligned}m_1^{(l)} &= m_1^{(l-1)} \\m_2^{(l)} &= (m_2^{(l-1)} - F(l)) / S(l) + 1 \\m_3^{(l)} &= (m_3^{(l-1)} - F(l)) / S(l) + 1\end{aligned}$$

Cette dernière fonctionne en définissant une fenêtre de taille $F(l) \times F(l)$ et la réduction des données dans cette fenêtre à une seule valeur.

La fenêtre est déplacée par les positions $S(l)$ après chaque opération de manière similaire à la couche convolutionnelle et la réduction est répétée à chaque position de la fenêtre jusqu'à ce que tout le volume d'activation soit réduit spatialement.

Les méthodes de réduction les plus courantes sont le regroupement maximal et la mise en commun moyenne. La mise en commun maximale fonctionne en trouvant la valeur la plus élevée dans la région de la fenêtre et en rejetant le reste des valeurs.

Le regroupement moyen, d'autre part, utilise plutôt la moyenne des valeurs dans la région. Les paramètres de mise en commun les plus couramment utilisés sont $F(l) = 2$, $S(l) = 2$ qui ne se chevauche pas.

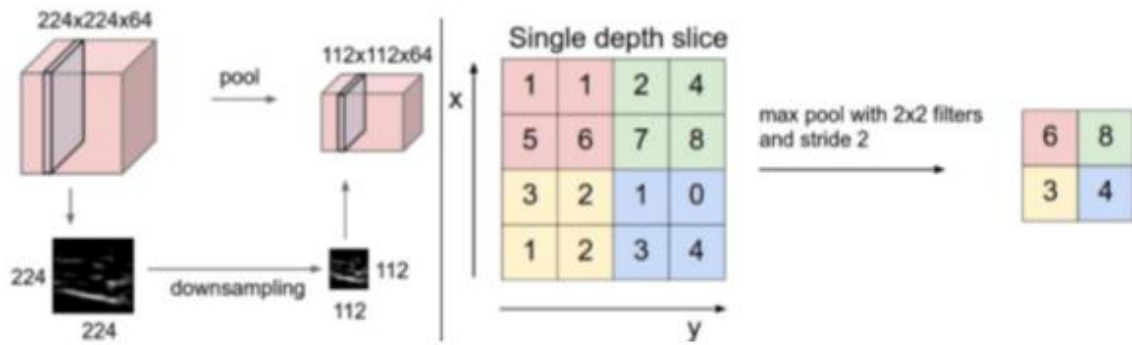


Figure 17: Max pooling avec filtre 2*2

3.5.4 Couche entièrement connectée

Les couches totalement connectées dans un réseau convolutif sont pratiquement un perceptron multicouches (généralement un MLP à deux ou trois couches) qui vise à cartographier $m_1^{(l-1)} \times m_2^{(l-1)} \times m_3^{(l-1)}$ le volume d'activation de la combinaison des couches différentes précédentes en une distribution de probabilité de classe.

Ainsi, la couche de sortie du perceptron multicouches aura $m_1^{(l-i)}$ sorties, c'est-à-dire des neurones de sortie où i indique le nombre de couches dans le perceptron multicouches.

La principale différence par rapport à un perceptron multicouches standard est la couche d'entrée où, au lieu d'un vecteur, un volume d'activation est pris comme entrée.

En conséquence, la couche entièrement connectée est définie comme suit :

Si (l-1) est une couche complètement connectée :

$$Y_i^{(l)} = f(z_i^{(l)}) \text{ avec } z_i^{(l)} = \sum_{j=1}^{m_1^{(l-1)}} w_{i,j}^{(l)} y_j^{(l-1)}$$

Le but de la structure complète entièrement connectée est de régler les paramètres de poids

$$W_{i,j}^{(l)}$$

Pour créer une représentation de probabilité stochastique de chaque classe en fonction des cartes d'activation générées par la concaténation de couches convolutives, de non-linéarité, de rectification et de mise en commun.

Les couches individuelles entièrement connectées fonctionnent de manière identique aux couches du perceptron multicouche, la seule exception étant la couche d'entrée.

3.5.5 La fonction SoftMax

La fonction softmax calcule la distribution des probabilités de l'évènement 'n' sur différents événements. D'une manière générale cette fonction calculera les probabilités de chaque classe cible sur toutes les classes possibles.

Plus tard les probabilités calculées seront utiles pour déterminer la classe cible pour les entrées données.

Dans notre cas, softmax calcule la probabilité qu'une image apparaitra a une classe déjà étiquetée.

Le principal avantage d'utiliser softmax est que la gamme sera entre 0 et 1, et la somme de toutes les probabilités sera égale a une. Si la fonction softmax utilisée pour le modèle multi classification renvoie les probabilités de chaque classe, la classe cible aura la plus élevée.

$$F(x_i) = \frac{\exp(x_i)}{\sum_{j=0}^k \exp(x_j)}$$

Avec : x_i le vecteur d'entrée de l'image a classifier

Et x_j : vecteur de classe j

Conclusion

Dans ce chapitre nous avons vu comment les réseaux de neurones interviennent dans la classification d'images, le chapitre suivant présentera notre système de classification d'images, son architecture et la description de ses différentes couches de fonctionnement.

Chapitre 4 : Réalisation et implémentation d'un système de classification d'images

Introduction

Après avoir vu la classification en général, et la classification d'images en particulier avec les réseaux de neurones convolutifs, nous passons à la dernière partie qui décrira l'architecture de notre système, et expliquera les différentes couches qui le compose

1. Architecture du système

L'architecture de notre système se divise en 3 composantes. Une couche convolutionnelle, une couche de pooling et une couche entièrement connectée. Ces composantes travaillent ensemble pour apprendre une représentation caractéristique dense d'une entrée

Cette architecture est illustrée dans la figure 19

1.2 Description des couches du système

1.2.1 La couche de convolution

Une convolution se compose d'un noyau, également appelé filtre, qui est appliqué dans une forme de fenêtre coulissante pour extraire des éléments de l'entrée.

Ce filtre est décalé après chaque opération sur l'entrée d'un montant appelé strides. À chaque opération, une matrice de multiplication du noyau et de la région actuelle d'entrée est calculée.

Les filtres peuvent être empilés pour créer des représentations à haute dimension de l'entrée.

Enfin, une fonction d'activation (généralement une ReLU) est appliquée pour donner la non-linéarité de la convolution.

1.2.2 La couche pooling

La mise en commun ou pooling est une opération visant à réduire la dimensionnalité. Il applique une fonction résumant les informations voisines.

Deux fonctions communes sont le max pooling et average pooling. En calculant le maximum d'une région d'entrée, la sortie résume l'intensité des valeurs environnantes

Les couches de pooling ont également un noyau, un padding et sont déplacées en strides (foulées).

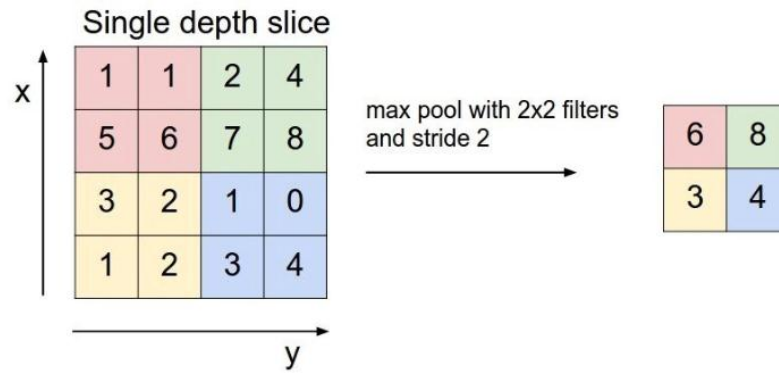


Figure 18: Exemple de fonctionnement de la couche pooling avec un filtre 2x2

1.2.3 Couche entièrement connectée

Effectue une classification sur les caractéristiques extraites par les couches convolutionnelles et diminuées par les couches de mise en commun. Dans une couche dense, chaque noeud de la couche est connecté à chaque noeud de la couche précédente.

En résumé :

En passant l'entrée par une convolution, on extrait des fonctionnalités hautement dimensionnelles. Pooling résume l'information spatiale et réduit la dimensionnalité. Enfin, cette représentation de la fonction passe par des couches entièrement connectées à un classificateur.

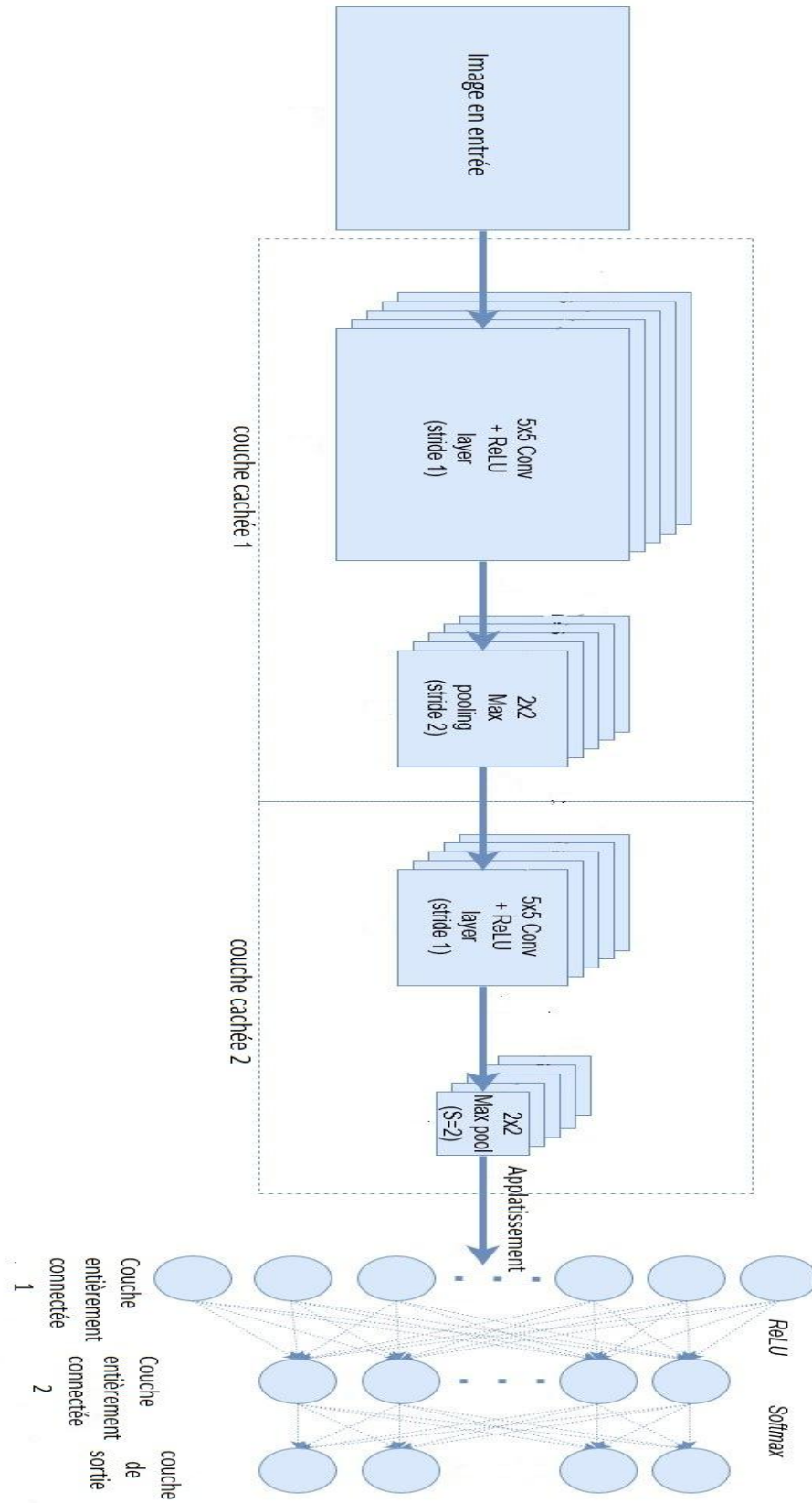


Figure 19: Architecture du système de classification d'images

2. Algorithmes de création des couches du réseau

2.1 Algorithme d'apprentissage

```
Function perceptron return a perceptron hypothesis
Input : example image with input  $X=x_1,x_2,\dots,x_n$  and output  $y$ 
Network : a perceptron with weight  $w_j, j=0,\dots,n$  and activation
function  $g$ .
Repeat
  For each  $e$  in example do
     $in := \sum_{j=0}^n w_j x_j (e)$ 
     $Err := y[e] - g[in]$ 
     $w_j := w_j + a * Err * g'(in) * x_j (e)$ 
  Until some stopping condition is satisfied.
Return neural-net hypotheses (network).
```

2.2 Algorithme de retro propagation:

Algorithme de retro propagation:

Function Back-propagation return a neural network

Input : example image with input $X=x_1, x_2, \dots, x_n$ and output vectory y network a multilayers network with l layer, weight $w_{j,l}$, activation function g

Repeat

For each e in example do

For each node j in the input layer do

$$a_i := x_i(e)$$

For $l=2$ to M do

$$in_i := \sum_{j=0}^n w_{j,l}$$

$$a_i := g(in_i)$$

For each node i in the output layer l do

$$\nabla_i := f(in_i) * (y_k(e) - a_i)$$

For $l=M-1$ to l do

For each node i in the output layer l do

$$\nabla_j := g'(in_j) \sum_l w_{j,l} * \nabla_i$$

For each node i in the output layer $l+1$ do

$$w_{j,l} := w_{j,l} + \alpha * \nabla_i$$

For $l=M-1$ to l do

$$\nabla_j := g'(in_j) \sum_l w_{l,l} \nabla_i$$

Until some stopping condition is satisfied

Return neural net hypothesis network.

3. Environnement de développement

3.1 Le langage python

Python est un langage de programmation open source, de haut niveau et multiplateformes, à la fois puissant et facile à maîtriser.

Python est un langage interprété, c'est à dire qu'il n'est pas nécessaire de le compiler avant de l'exécuter

3.2 Syntaxe du langage python

Python a été conçu pour être un langage lisible. Il vise à être visuellement épuré. Par exemple, il possède moins de constructions syntaxiques que de nombreux langages structurés tels que C, Perl, ou Pascal. Les commentaires sont indiqués par le caractère croisillon (#).

Les blocs sont identifiés par l'indentation, au lieu d'accolades comme en C ou C++ ; ou de begin ... end comme en Pascal ou Ruby. Une augmentation de l'indentation marque le début d'un bloc, et une réduction de l'indentation marque la fin du bloc courant. Par convention, l'indentation est habituellement de quatre espaces en Python

Fonction factorielle en C	Fonction factorielle en Python
<pre>int factorielle(int n) { if (n < 2) { return 1; } else { return n * factorielle(n - 1); } }</pre>	<pre>def factorielle(n): if n < 2: return 1 else: return n * factorielle(n - 1)</pre>

4 Description du Framework TensorFlow

TensorFlow, une bibliothèque logicielle Open Source dédiée à l'apprentissage automatique développé par Google en novembre 2015, elle prenait alors la décision de le plaçant sous licence Apache 2.0. Il était question de prendre de l'avance dans le domaine de l'intelligence artificielle, et plus particulièrement dans les architectures de type neuronal, utilisées pour gérer de grandes quantités de données, l'apprentissage profond (ou automatique) ainsi que certains calculs spécifiques.

TensorFlow est aujourd'hui particulièrement utilisé pour le Deep Learning, et donc les réseaux de neurones. Son nom est notamment inspiré du fait que les opérations courantes sur des réseaux de neurones sont principalement faites via des tables de données multidimensionnelles, appelées Tenseurs (Tensor) et un Tensor à deux dimensions est l'équivalent d'une matrice.

Aujourd'hui, les principaux produits de Google sont basés sur TensorFlow : Gmail, Google Photos, Reconnaissance vocal, etc.

TensorFlow est populaire pour de nombreuses raisons :

- Multiplateformes (Linux, Mac OS, et même Android et iOS !)
- APIs en Python, C++, Java et Go (l'API Python est plus complète cependant, c'est sur celle-ci que nous allons travailler)
- Temps de compilation très courts dû au backend en C/C++
- Supporte les calculs sur CPU, GPU
- Une documentation extrêmement bien fournie avec de nombreux exemples et tutoriels
- Google a annoncé avoir migré la quasi-totalité de ses projets liés au Deep Learning en TensorFlow : Gmail, Google Photos, Reconnaissance vocal, etc

4.1 TensorFlow dans la classification d'images

TensorFlow permet aux chercheurs, aux ingénieurs et à tous les passionnés de construire un système de classification d'images en utilisant un modèle d'apprentissage en profondeur déjà formé et connu pour classer une image dans 1000 catégories fournies par la compétition académique ImageNet avec un taux d'erreur qui a porté sur la performance humaine

4.2 TensorFlow comment ça marche ?

La particularité de TensorFlow est qu'il représente les calculs sous la forme d'un graphe d'exécution : chaque nœud représente une Opération à réaliser, et chaque lien représente un Tensor. Une Opération peut aller d'une simple addition à une fonction complexe de différenciation matricielle.

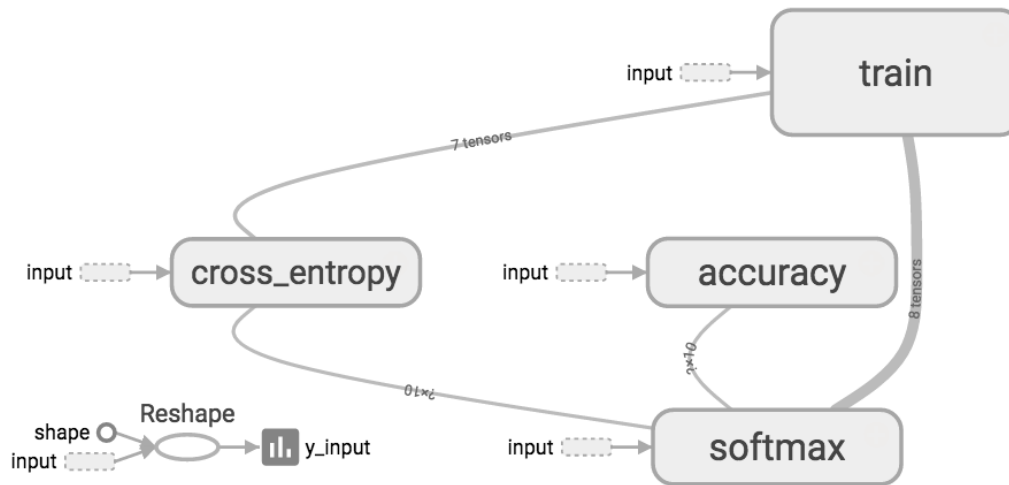


Figure 20: Architecture du Framework TensorFlow

Chaque Opération prend en entrée zéro, un ou plusieurs Tensor, effectue un calcul, et retourne zéro, un ou plusieurs Tensor. Un exemple typique de Tensor est un batch d'images. Un batch d'images est représenté par un Tensor à 4 dimensions : taille du batch (nombre d'images dans le batch), hauteur, largeur et nombre de canaux de représentation (3 pour une image en couleurs représentée en RGB).

La création du graphe est automatiquement gérée par TensorFlow une fois les Tensor et Opérations implémentés et instanciés. Cela permet une optimisation et parallélisation du code et de l'exécution lors du lancement.

TensorFlow possède de plus un support très vaste pour la création d'opérations spécifiques au Deep Learning, et il devient donc facile de construire un réseau de neurones et d'utiliser les opérations mathématiques couramment associées pour l'entraîner avec les bons optimiseurs.

5 Le modèle Inception de Google

Inception V3 est un modèle d'apprentissage en profondeur de TensorFlow publié par Google, son architecture peut être visualisé par ce modèle schématique

C'est un modèle de reconnaissance d'image qui se compose de deux parties :

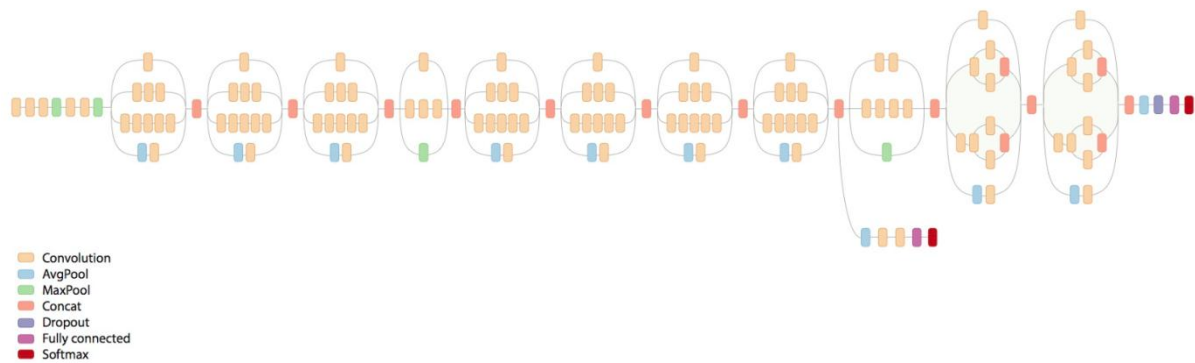


Figure 21: Le modèle Inception V3

6 Configuration de l'environnement

6.1 Installation de TensorFlow

Comme cité précédemment, TensorFlow est basé principalement sur le langage python, donc pour pouvoir l'installer et l'utiliser, il faudra d'abord installer une version de python adaptée, et qui est la version 3.5. et n'oubliant pas bien sur de configurer la variable d'environnement de Windows.

Ensuite avec l'invite de commande il suffit simplement de taper la commande :

```
C : \> Pip3 install -upgrade tensorflow
```

6.2 Choix des données d'apprentissage

Les données de formation sont celles que nous souhaitons que notre ordinateur apprenne, afin de reconnaître les objets dans les images. Nous avons besoin d'au moins deux cents images pour chaque classe.

Nous avons utilisé le plugin de Google appelé " Fatkun Batch Download Image " pour télécharger une quantité d'images à la fois

Nous avons choisi d'entraîner notre modèle sur cinq catégories d'images, nous avons entre autre :

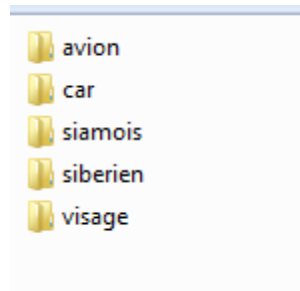


Figure 22: Dossier d'images d'entraînement

Tous les dossiers contiennent exactement le même nombre d'images : 420 images de tailles différentes.

Les dossiers siamois et siberien contiennent des images de chats de ces deux races, et les dossiers car et avion, contiennent respectivement des images de voitures et d'avions de tout type

6.3 Test de fonctionnalité

Tout d'abord nous allons faire appel au fichier « retrain.py » qui se trouve dans le dossier de TensorFlow précédemment installé, et lui spécifier certains paramètres :

```
python {$dossier_de_travail}/retrain.py
--bottleneck_dir=/{$ dossier_de_travail }/bottlenecks
--how_many_training_steps 500
--model_dir=/{$ dossier_de_travail }/inception
--output_graph=/{$ dossier_de_travail }/retrained_graph.pb
--output_labels=/{$ dossier_de_travail }/retrained_labels.txt
--image_dir /{$ dossier_de_travail }/{$dossier_d'images_d'entraînement}
```

```
Microsoft Windows [version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Tous droits réservés.

C:\Users\Sabrina>python D:\tensorflow_work\tensorflow\tensorflow\examples\image_
retraining\retrain.py --bottlenecks_dir=D:\tensorflow_work\bottlenecks --how_m
any_training_steps 500 --model_dir=D:\tensorflow_work\inception-2015-12-05 --outpu
t_graph=D:\tensorflow_work\retrained_graph.pb --output_labels=D:\tensorflow_work
\retrained_labels.txt --image_dir D:\tensorflow_work\training_data
```

Après exécution de ce morceau de code, nous verrons apparaître de nouveaux dossiers dans notre espace de travail, tel que le dossier « bottlenecks » qui contient une instance des dossiers de nos images, sauf que cette fois-ci, ce dossier ne contient pas des images mais des fichiers avec l'extension .txt contenant des valeurs numériques comprises entre 0 et 1, et donc ce sont les caractéristiques extraites de nos images

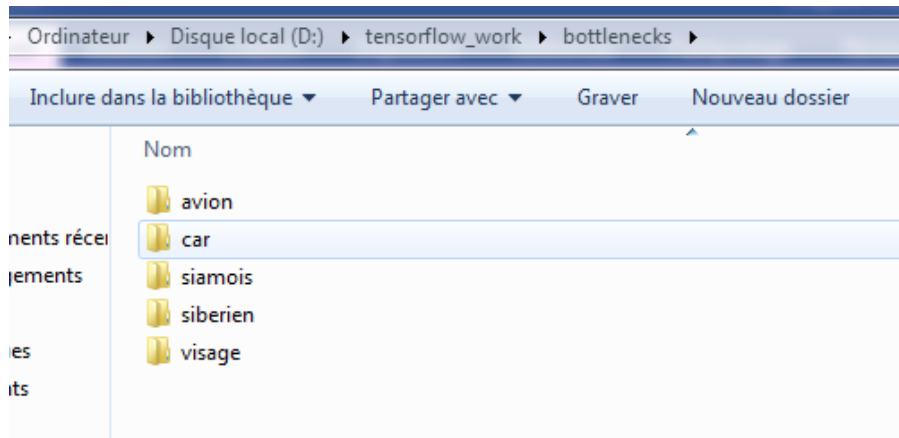


Figure 23: Contenu du dossier bottlenecks

Le dossier « bottlenecks » qui crée une instance des dossiers de nos images, sauf que cette fois-ci, ce dossier ne contient pas des images mais des fichiers avec l'extension .txt contenant des valeurs numériques comprises entre 0 et 1, et donc ce sont les caractéristiques extraites de nos images

D'autres fichiers sont aussi créés et mis dans le même dossier de travail :

Ensuite il faut créer un fichier qu'on nommera label_image.py et l'ajuster en fonction de nos dossiers

```
label_image.py x
1 import tensorflow as tf, sys
2 image_path = sys.argv[1]
3 # Read in the image_data
4 image_data = tf.gfile.GFile(image_path, 'rb').read()
5 # Loads label file, strips off carriage return
6 label_lines = [line.rstrip() for line
7                 in tf.gfile.GFile("D:/tensorflow_work/retrained_labels.txt")]
8 # Unpersists graph from file
9 with tf.gfile.GFile("D:/tensorflow_work/retrained_graph.pb", 'rb') as f:
10     graph_def = tf.GraphDef()
11     graph_def.ParseFromString(f.read())
12     _ = tf.import_graph_def(graph_def, name='')
13 # Feed the image_data as input to the graph and get first prediction
14 with tf.Session() as sess:
15     softmax_tensor = sess.graph.get_tensor_by_name('final_result:0')
16     predictions = sess.run(softmax_tensor,
17                            {'DecodeJpeg/contents:0': image_data})
18     # Sort to show labels of first prediction in order of confidence
19     top_k = predictions[0].argsort() [-len(predictions[0]):] [::-1]
20     for node_id in top_k:
21         human_string = label_lines[node_id]
22         score = predictions[0][node_id]
23         print('%s (score = %.5f)' % (human_string, score))
```

Figure 24 : Contenu du fichier Label_image.py

Maintenant nous allons écrire une requête, et spécifier à l'intérieur le chemin d'une image que nous avons choisi, notre image se situe ici dans le répertoire :

F:\images_test

Ensuite nous ferons appel au fichier « label_image.py »

```
C:\Windows\system32\cmd.exe
Microsoft Windows [version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Tous droits réservés.

C:\Users\Sabrina>python D:\tensorflow_work\label_image.py F:\images_test\c.jpg
```

Le résultat de la classification, nous montre l'appartenance de notre image à un score de 0.92 à la classe des chats siberien

Chapitre 4 : Réalisation et implémentation d'un système de classification d'images

```
C:\Windows\system32\cmd.exe
Microsoft Windows [version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Tous droits réservés.

C:\Users\Sabrina>python D:\tensorflow_work\label_image.py F:\images_test\c.jpg
2017-09-24 20:40:17.413201: W C:\tf_jenkins\home\workspace\rel-win\M\windows\PY\
35\tensorflow\core\platform\cpu_feature_guard.cc:45] The TensorFlow library wasn
't compiled to use AVX instructions, but these are available on your machine and
could speed up CPU computations.
2017-09-24 20:40:19.035293: W C:\tf_jenkins\home\workspace\rel-win\M\windows\PY\
35\tensorflow\core\framework\op_def_util.cc:333] Op BatchNormWithGlobalNormaliza
tion is deprecated. It will cease to work in GraphDef version 9. Use tf.nn.batch
_normalization().
siberien (score = 0.92003)
siamois (score = 0.05261)
visage (score = 0.01687)
avion (score = 0.00676)
car (score = 0.00373)
```

Figure 25: Test du fonctionnement du système

Voici une image qui récapitule ce qui se passe dans notre système lorsqu'une nouvelle image lui est soumise en vue de la classer

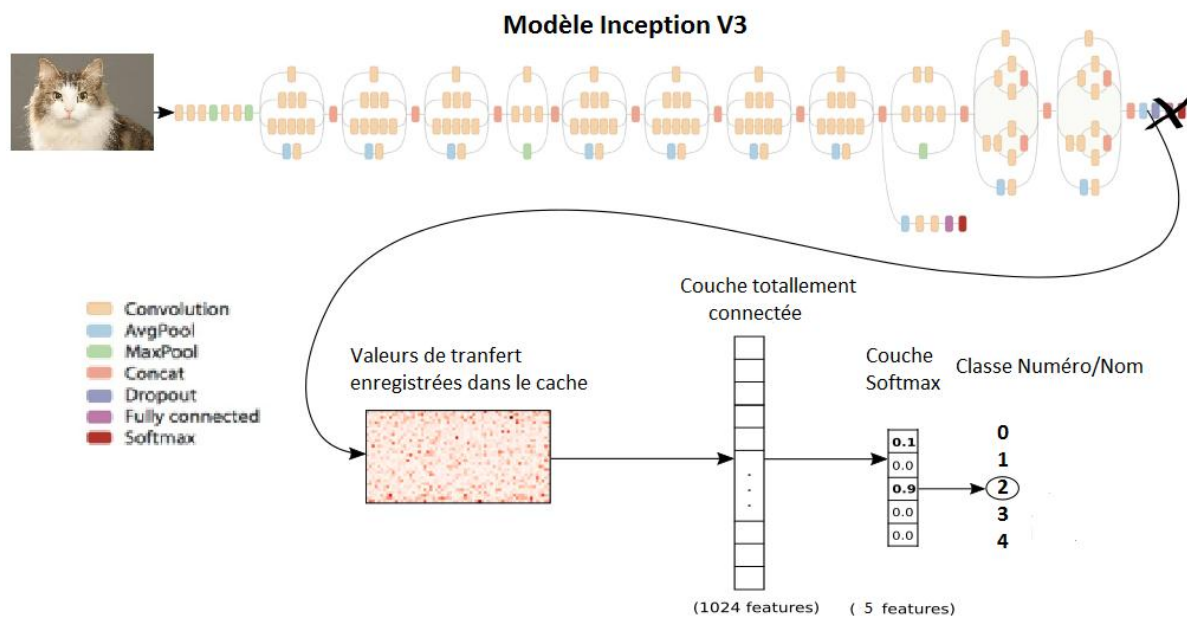


Figure 26 : Démarche de fonctionnement du système

Conclusion

Chapitre 4 : Réalisation et implémentation d'un système de classification d'images

Pour résumer, nous avons ensemble d'images réparties en 5 dossiers, contenant chacun 420 images d'entraînement, après exécution du réseau de neurone, des caractéristiques sont extraites de chaque image grâce aux couches convolutionnelle et pooling,

La couche de convolution élimine les caractéristiques inutiles, jusqu'à n'en garder que les caractéristiques significatives, pour produire des cartes de caractéristiques.

Vers la fin de la phase d'extraction de caractéristique ou d'entraînement, chaque dossier d'images formera un vecteur de caractéristiques (Tensor).

Donc lorsque viendra une nouvelle image, le même traitement lui ai attribué, ensuite après avoir obtenu sa bottleneck, il est d'abord normalisé en un vecteur de poids avec la fonction maxpool, puis comparé avec les autres vecteurs de nos dossiers.

Une fois la comparaison faite, un vecteur de probabilité est créé, c'est-à-dire les probabilités d'appartenance de cette nouvelle image à une de nos classes.

Conclusion générale

Au cours de notre mémoire nous avons répondu à la problématique concernant la classification d'images en utilisant les réseaux de neurones convolutifs, pour réaliser une application en nous aidant d'une bibliothèque Open Source « TensorFlow », qui nous a permis de créer notre propre modèle de réseau de neurones, et l'entraîner sur nos propres données d'images.

Les réseaux de neurones convolutifs sont à ce jour les modèles les plus performants pour classer des images. Ils appliquent une série de filtres aux données de pixels brutes d'une image pour extraire et apprendre des fonctionnalités de niveau supérieur, que le modèle peut alors utiliser pour la classification.

Avec l'étape d'extraction des caractéristiques dans les images on obtient des connaissances qui seront par la suite utilisées pour la classification.

Ce qui fait donc la réunion entre les deux domaines : le data mining et l'apprentissage machine. ET c'est le data mining qui est au service de cet apprentissage machine, en lui fournissant un ensemble de connaissances extraites des images qui serviront par la suite à la classification de nouvelles données d'images.

L'application que nous avons élaborée permet de classer une image en entrée dans cinq différentes catégories, en la faisant passer par un certain nombre de filtre pour extraire des caractéristiques significatives, qui seront par la suite utilisées pour générer un vecteur de probabilité.

Notre système pourrait être amélioré pour accueillir plusieurs images en entrée à la fois et non pas qu'une seule, il pourrait aussi être une entrée pour un autre système d'analyse de séquences vidéos par exemple, pour pouvoir distinguer des objets dans les séquences d'images qui composent la vidéo.

Ce travail nous a permis d'approfondir nos connaissances sur les méthodes de classifications d'images, et plus précisément sur les réseaux de neurones du type convolutifs.

Nous espérons que ce mémoire éveillera une passion et un intérêt chez les promotions à venir et à toute personne intéressé par ce sujet afin qu'elle puisse persévérer dans ce domaine et améliorer ce modeste travail que nous avons entamé.

Table des figures

Figure 1 : Processus d'extraction de connaissances à partir de données.....	12
Figure 2: Différence entre apprentissage supervisé et apprentissage non supervisé.....	21
Figure 3: Structure d'un arbre de décision.....	23
Figure 4: Exemple d'arbre de décision	23
Figure 5: Structure d'un réseau de neurone à couche unique (perceptron).....	25
Figure 6: Exemple d'un réseau bayésien.....	29
Figure 7: Exemple d'arbre hiérarchique	33
Figure 8: Illustration de regroupement en clusters	39
Figure 9: Les deux types de clustering non-hiérarchique/hiérarchique	41
Figure 10: La partition hiérarchique.....	46
Figure 11: Illustration des K-PPV d'un point	49
Figure 12: Perceptron à trois couches (schéma type).....	52
Figure 13: Perceptron simple	59
Figure 14: Exemple d'un réseau de neurones convolutionnelles	62
Figure 15: Architecture d'un réseau de neurone convolutionnel.....	62
Figure 16: Convolution avec un filtre 3*3	64
Figure 17: Max pooling avec filtre 2*2	67
Figure 18: Exemple de fonctionnement de la couche pooling avec un filtre 2x2.....	71
Figure 19: Architecture du système de classification d'images	72
Figure 21: Architecture du Framework TensorFlow.....	77
Figure 22: Le modèle Inception V3.....	78
Figure 23: Dossier d'images d'entraînement	79
Figure 24: Contenu du dossier bottlenecks	80
Figure 25 : Contenu du fichier Label_image.py	81
Figure 26: Test du fonctionnement du système	82
Figure 27 : Démarche de fonctionnement du système.....	82

Références

- [1] G. PIATETSKY-SHAPIRO, Data mining and knowledge discovery 1996 to 2005: overcoming the hype and moving from «university» to «business» and «analytics», *Data mining and Knowledge Discovery*, 15(1), 99-105.
- [2] D. A. Zighed, G. Duru, and J. P. Auray, “Sipina, Méthode et Logiciel,” Lacassagne, 2000
- [3] F. U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth, “From data mining to knowledge discovery in databases,” Dans *aimag KDD overview*, pp 1–34, 1996
- [4] Y. Kodratoff, “techniques et outils de l’extraction de connaissances à partir des données,” *Signaux*, vol. 92, pp 38–43, Mars 1998
- [5] Site web damacosy, de la gestion des données à la maîtrise de l’information-data mining
- [6] P. CABENA, P. HADJINIA, R. STADLER, J. VERHEES et A. ZANASI, *Discovering data mining : from concept to implementation*, prentice hall, upper saddle river, NJ, 1998
- [7] M. KANTARDZIC “data mining-concepts, Models, Methods and algorithms” IEEE press, piscataway, NJ, USA, 2003
- [8] CHAMI Djazia, Une plateforme orientée agent pour le data mining, Université HADJ LAKHDAR – BATNA, 2009-2010
- T. MENOUEUR, M. DERMOUCHE, *Application des techniques de data mining pour la classification automatique des données et la recherche d’associations*, Ecole Nationale Supérieure d’Informatique, 2009/2010
- [9] BRAHIMI Belgacem, *Extraction de connaissances à partir de données incomplètes et imprécises*, Université de M’sila, 2011
- [10] AMROUNE Naceredine, *Fouille de données multi-stratégie pour l’extraction de connaissances à partir d’images médicales*, Université de M’sila,
- [11] O. R. ZAÏANE, *Principles of Knowledge Discovery in Databases*, CMPUT690, University of Alberta, 1999.
- [12] Wikipédia, Encyclopédie en ligne, les bases de données multimédia
- [13] G. HEBRAIL, *Transformation de longues séries temporelles en descriptions symboliques*, ENST Paris, LTCI-UMR 5141 CNRS, Département Informatique et Réseaux, France.
- [14] B. AGARD, A. KUSIAK, *Exploration Des Bases De Données Industrielles À L’aide Du Data Mining – Perspectives*, 9ème Colloque National AIP PRIMECA, avril 2005.
- [15] *Outil de reporting décisionnel : DATA WAREHOUSE - Cube OLAP : Présentation conviviale des données chiffrées*, Rapporté de <http://www.siom.fr/sycube.pdf>
- [16] site web quora.com, *Quelle est la différence entre l’apprentissage supervisé et l’apprentissage non supervisé dans l’apprentissage machine*
- [17] BENAMAR HOUMADI, *mémoire, étude exploratoire d’outils pour le data mining*
- [18] A. Aamodt and E. Plaza. *Case-based reasoning: Foundational issues, methodological variations, and system approaches*. In IOS press, 1994.
- [19] G. CALAS, *Études des principaux algorithmes de data mining*, Spécialisation Sciences Cognitives et Informatique Avancée, France.
- [20] Kass, G. V.; *An Exploratory Technique for Investigating Large Quantities of Categorical Data*, *Journal of Applied Statistics*, Vol. 29, No. 2 (1980)
- [21] Nisbet, Robert; Elder, John; Miner, Gary; *Handbook for Statistical Analysis And Data Mining*, Academic Press, 2009
- [22] J. Ross Quinlan, *Machine Learning*, 1986, « Induction of decision trees »
- [23] Jeffrey C. Schlimmer and Douglas H. Fisher. *A case study of incremental concept induction*. In *AAAI*, pages 496–501, 1986.
- [24] Quinlan, J. R. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers, 1993
- [25] S. Tufféry, *Data mining et statistique décisionnelle, l’intelligence dans les bases de données*, Groupe bancaire Français, Universités de Rennes 1 et parisDauphine, 2005.
- [26] S. Tufféry, *Data mining et scoring, Bases de données et gestion de la relation client*, Groupe bancaire Français, Universités de Rennes 1 et paris-Dauphine, 2002

- [27] Mohammed J. Zaki. Scalable algorithms for association mining. *IEEE Transactions on Knowledge and Data Engineering*, 12 :372–390, 2000.
- [28] Mohammed J. Zaki. Scalable algorithms for association mining. *IEEE Transactions on Knowledge and Data Engineering*, 12 :372–390, 2000.
- [29] : Livre :ALGORITHMES DE CLASSIFICATION Marseille, Juin 2006.Maurice ROUX Professeur émérite Université Paul Cézanne Marseille, France.2006
- [30] [Jamouille,&al,2000]:MarcJamouille,MichelRoland,JacquesHumbert,Jean-François Brûlet.Traitementdel’informationmédicaleparlaClassificationinternationaledesoins primaires, deuxième version : CISP-2. Care Edition, Bruxelles, 2000.
- [31] Mounzer BOUBOU : "contribution aux méthodes de classification non supervisée via des approches prétopologiques et d’agrégations d’opinion”, thèse de doctorat, université Claud Bernard –Lyon1, 2007.
- [32] [Peter, 2001] : Richard O. Duda, Peter E. Hart, David G. Stork, *Pattern classification*, Wiley-interscience, 2001.
- [33] [Bisson, 2000] Bisson, G. , *La similarité : une notion symbolique/numérique*. Chap. XX of : *Apprentissage symbolique-numérique (tome 2)*. Editions CEPADUES.2002.
- [34] [Govaert, 2003] : Gérard. Govaert, *Analyse des données*. IC2(série Traitement du signal et de l’image), Lavoisier.2003
- [35] [Berrani & al., 2002] : Berrani, S.-A., Amsaleg, L., & Gros, P. Recherche par similarités dans les bases de données multidimensionnelles : panorama des techniques d’indexation. *Ingénierie des systèmes d’information (RSTI série ISI-NIS)*, 7(5-6), pp 65-90.2002.
- [36] [Ball et Hall, 1967] : Ball, G. H. et Hall, D. J. ISODATA, an Iterative Method of Multivariate Analysis and Pattern Recognition. *Behavior Science*, 153, 1967.
- [37] [Bailey & Elkan, 1995] : T. L. Bailey and C. Elkan , “Unsupervised learning of multiple motifs in biopolymers using expectation maximization,” *Machine Learning*, vol. 21, pp. 51–80, 1995.
- [38] [Rennie & al. , 2003] : Rennie J, Shih L, Teevan J, and Karger D. Tackling The Poor Assumptions of Naive Bayes Classifiers. In *Proceedings of the Twentieth International Conference on Machine Learning (ICML)*. 2003.
- [39]. Fabien Moutarde. “Brève introduction aux arbres de décision”. Cours en ligne. Centre de Robotique (CAOR), Ecole des Mines de Paris. 2008. (Consulté le 13.01.2010). Disponible sur : www.ensmp.fr/~moutarde/slides_AD.pdf
- [40] [Saporta, 1990] : Saporta G. , *probabilités, analyse des données et statistiques*. Technip , paris, 1990.
- [41] [Lebart & al. 200] : Lebart L., Morineau A. & piron M.*statistique exploratoire multidimensionnelle*. Dunod, 3ème édition, paris, 2000.
- [42] [kohonen, 1982] : Kohonen T. self-organized formation of topologically correct feature maps. *Biological cybernetics* no 43, pp59-69, reprinted in Anderson & Rosenfeld , Eds, *Neurocomuting : foundations of research*, MIT press, Cambridge Ma, 1988.
- [43] [Hartigans, 1975] : J. Hartigans. *clustering algorithms*. John Wiley and Sons, Inc., 1975.
- [44][SPA 80] Spath .R, " Cluster Analysis Algorithms for data reduction and classification of objects". Ellis Rorwood, Willy & Sons, New York, 1980.